Interdomain routing with BGP4

Part 2/5

Olivier Bonaventure

Department of Computing Science and Engineering
Université catholique de Louvain (UCL)
Place Sainte-Barbe, 2, B-1348, Louvain-la-Neuve (Belgium)

URL : *http://www.info.ucl.ac.be/people/OBO*

November 2004

Outline

- Organization of the global Internet
- BGP basics
  - Routing policies
  - The Border Gateway Protocol
  - How to prefer some routes over others
- BGP in large networks
- Interdomain traffic engineering with BGP
- BGP-based Virtual Private Networks

Interdomain routing

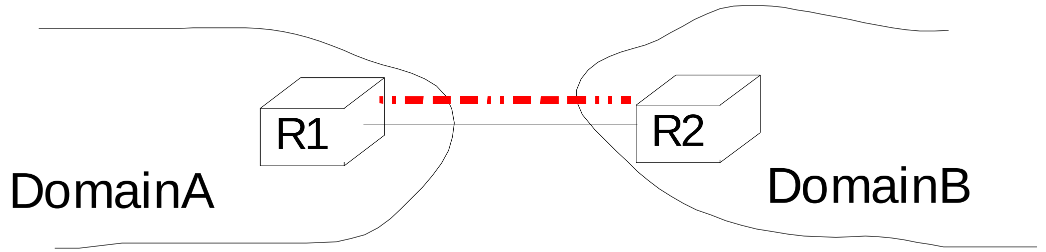
- Goals
  - Allow to transmit IP packets along the best path towards their destination through several transit domains while taking into account the routing policies of each domain without knowing the detailed topology of those domains
    - From an interdomain viewpoint, best path often means *cheapest* path
    - Each domain is free to specify inside its routing policy the domains for which it agrees to provide a transit service and the method it uses to select the best path to reach each destination

Domains versus Autonomous Systems

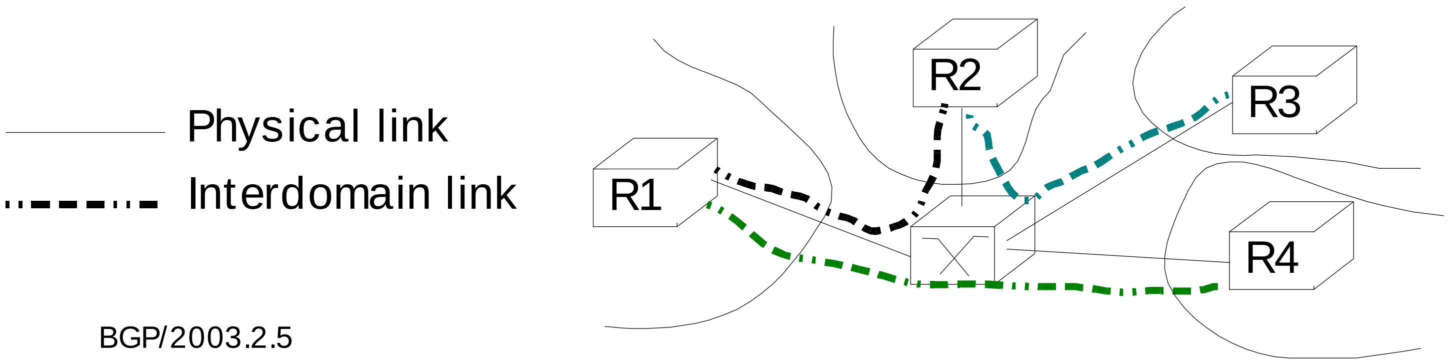
- The BGP interdomain routing protocol deals with Autonomous Systems (AS)
  - An AS is defined as <<*a set of routers under a single technical administration ... that presents a consistent picture of what destinations are reachable through it.*>>
  - Each AS is identified by its AS number
- In practice
  - A domain is often equivalent to an AS
  - A domain may be composed of several ASes
    - Ex: Worldcom uses AS701, AS702, ...
  - Many domains do not have an AS number
    - Ex: small networks connected to one provider without using BGP

Types of interdomain links

- Two types of interdomain links
  - Private link
    - Usually a leased line between two routers belonging to the two connected domains



  - Connection via a public interconnection point
    - Usually Gigabit or higher Ethernet switch that interconnects routers belonging to different domains

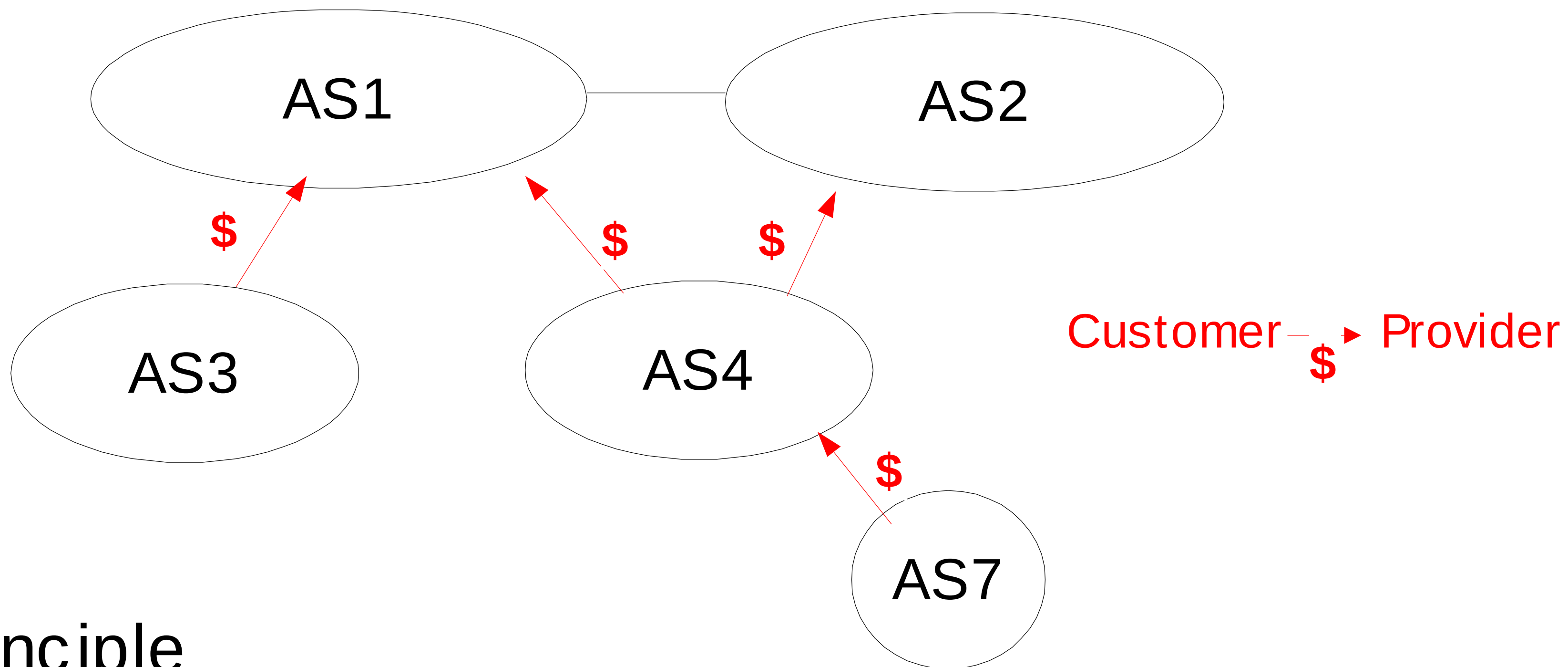


Routing policies

- In theory BGP allows each domain to define its own routing policy...

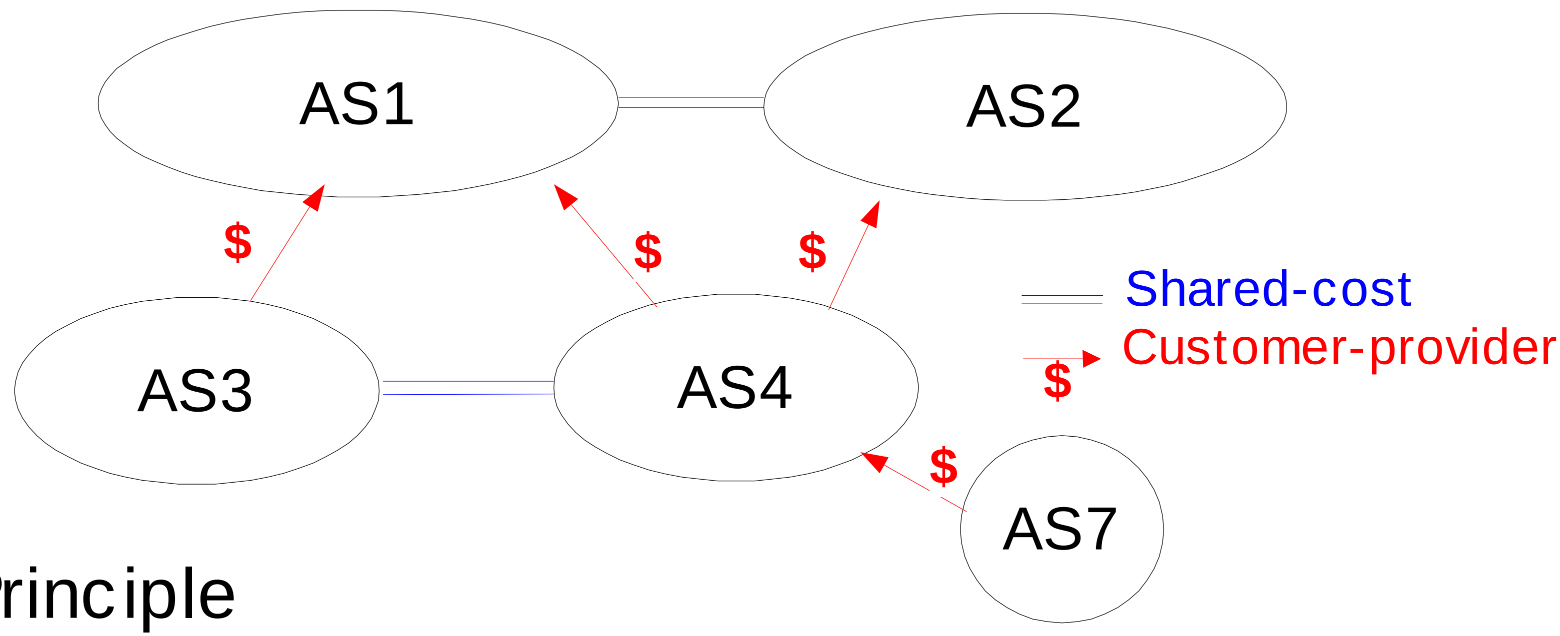
- In practice there are two common policies
  - customer-provider peering
    - **Customer c** buys Internet connectivity from **provider P**
  - shared-cost peering
    - **Domains x** and **y** agree to exchange packets by using a direct link or through an interconnection point

Customer-provider peering



- Principle
  - Customer sends to its provider its internal routes and the routes learned from its own customers
    - Provider will advertise those routes to the entire Internet to allow anyone to reach the Customer
  - Provider sends to its customers all known routes
    - Customer will be able to reach anyone on the Internet

Shared-cost peering



- Principle
  - PeerX sends to PeerY its internal routes and the routes learned from its own customers
    - PeerY will use shared link to reach PeerX and PeerX's customers
    - PeerX's providers are not reachable via the shared link
  - PeerY sends to PeerX its internal routes and the routes learned from its own customers
    - PeerX will use shared link to reach PeerY and PeerY's customers
    - PeerY's providers are not reachable via the shared link

Routing policies

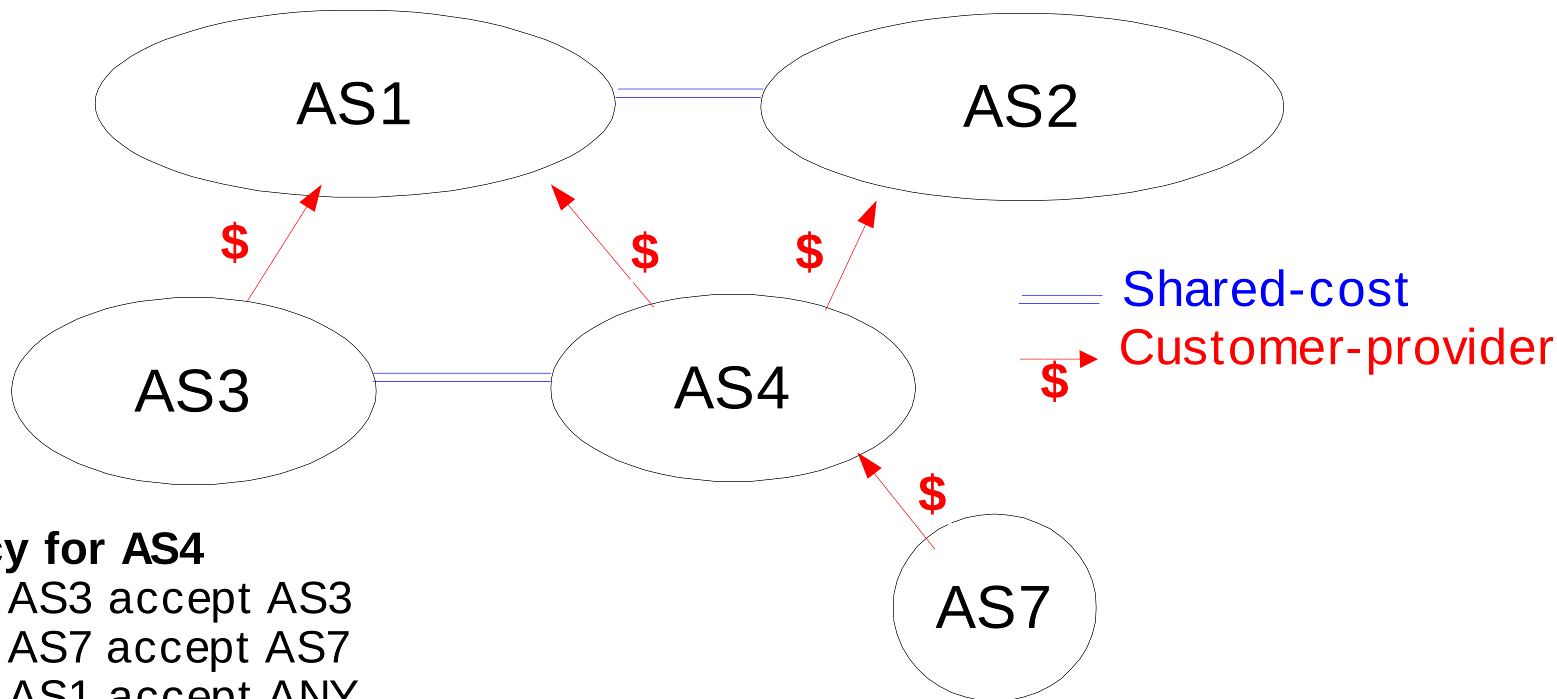
- A domain specifies its routing policy by defining on each BGP router two sets of filters for each peer
  - Import filter
    - Specifies which routes can be accepted by the router among all the received routes from a given peer
  - Export filter
    - Specifies which routes can be advertised by the router to a given peer
- Filters can be defined in RPSL
  - Routing Policy Specification Language

RPSL

- Simple import policies
  - Syntax
    - `import: from AS# accept list_of_AS`
  - Examples
    - `Import: from Belgacom accept Belgacom WIN`
    - `Import: from Provider accept ANY`

- Simple export policies
  - Syntax
    - `Export: to AS# announce list_of_AS`
  - Example
    - `Export: to Customer announce ANY`
    - `Export: to Peer announce Customer1 Customer2`

Routing policies
Simple example with RPSL



Import policy for AS4
Import: from AS3 accept AS3
import: from AS7 accept AS7
import: from AS1 accept ANY
import: from AS2 accept ANY

Export policy for AS4
export: to AS3 announce AS4 AS7
export: to AS7 announce ANY
export: to AS1 announce AS4 AS7
export: to AS2 announce AS4 AS7

Import policy for AS7
Import: from AS4 accept ANY

Export policy for AS4
export: to AS4 announce AS7

Scalable routing policies with RPSL

- How to specify policies of large domains ?
  - Define one `route` object for each advertised prefix
    - `route:` prefix
    - `descr:` human-readable description
    - `origin:` AS# advertising the prefix
  - Define one as-set for all the clients of a given AS
    - `as-set`: macro name
    - `descr:` human-readable description
    - `members:` list of clients AS#
  - Specify the routing policies by using `as-sets` instead of AS numbers whenever possible

Scalable routing policies with RPSL (2)

- Example

```
aut-num:    AS20965
as-name:    GEANT
descr:      The GEANT IP Service
...
import:     from AS2611 action pref=100;accept AS-BELNET
...
export:     to AS2611 announce AS-GEANTNRN ...
```

```
as-set:     AS-BELNET
descr:      BELNET AS Macro
members:    AS2611, AS15383, AS9208, AS2111
```

```
route:      130.104.0.0/16
descr:      NET-UCLOUVAIN
origin:     AS2611
...
route:      138.48.0.0/16
descr:      FUNDP-AC-BE
origin:     AS2611
```

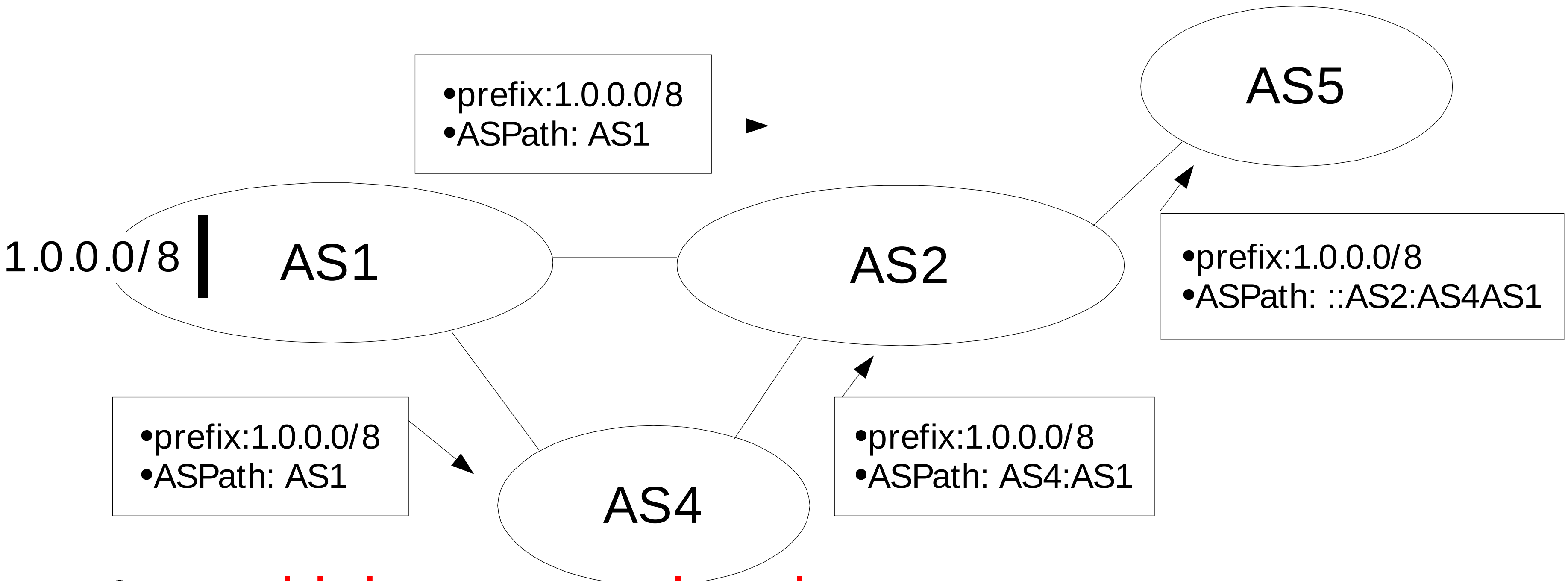
```
route:      81.19.48.0/20
descr:      IST-ATRIUM-EXP-20030212
origin:     AS2111
```

Outline

- Organization of the global Internet
- BGP basics
  - Routing policies
  - The Border Gateway Protocol
  - How to prefer some routes over others
- BGP in large networks
- Interdomain traffic engineering with BGP
- BGP-based Virtual Private Networks

The Border Gateway Protocol

- Principle
  - Path vector protocol
    - BGP router advertises its best route to each destination



  - ... with incremental updates
    - Advertisements are only sent when their content changes

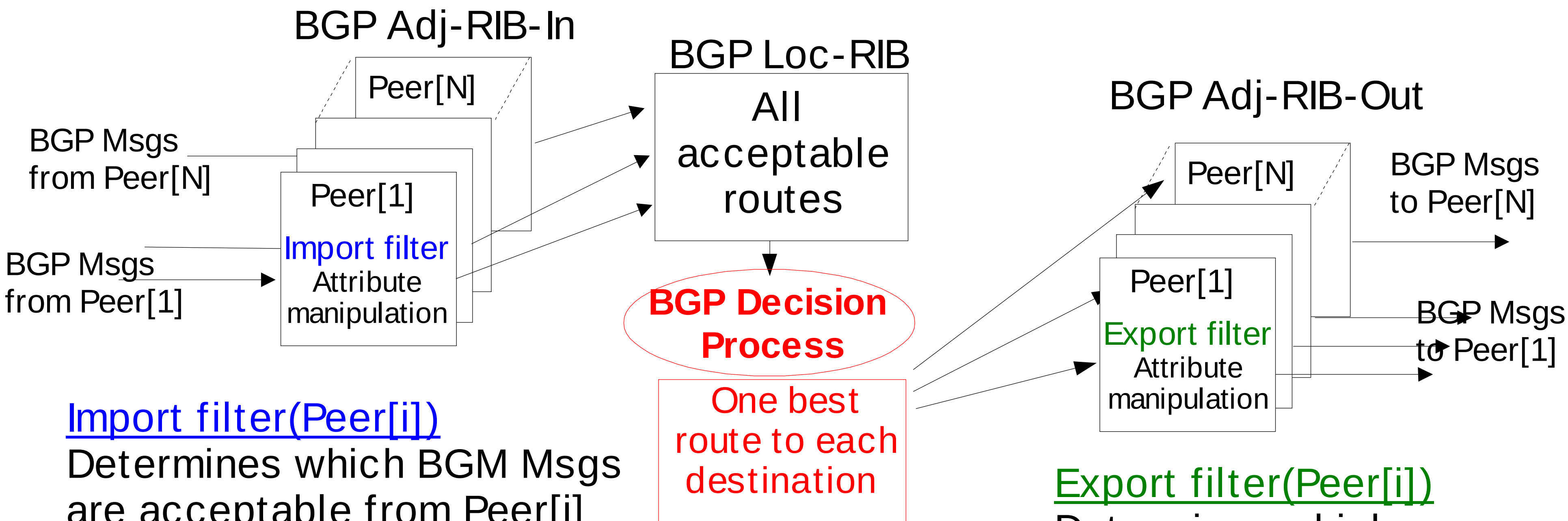
"Origin" of the routes announced by BGP

- Where do the routes announced by a BGP router come from ?
  - Learned from other BGP routers
    - BGP router only propagates the received routes
  - Static configuration
    - BGP router is configured to advertise some prefixes
    - Drawback : requires manual configuration
    - Advantage : Stable set of advertised prefixes
  - Learned from an Interior Gateway Protocol
    - The prefixes received from the IGP are advertised by the BGP router usually as an aggregate
    - Advantage
      - BGP advertisements follow network state, prefix is automatically withdrawn by BGP it is not reachable via IGP
    - Drawback
      - BGP announcements will be unstable if IGP is unstable...

Policies and BGP

- Two mechanisms to support policies in BGP
  - Each domain defines itself which is the best route to reach each destination based on the routes learned from its peers
    - The chosen best route is not necessarily the "shortest" route as with IGPs
    - Only the best route towards each destination can be announced to external peers
  - Each domain determines, on its own, which routes can be advertised to each peer
    - An AS does not necessarily advertise to all its neighbors all the routes that it knows

Conceptual model of a BGP router



<u>Import filter(Peer[i])</u>
Determines which BGM Msgs
are acceptable from Peer[i]

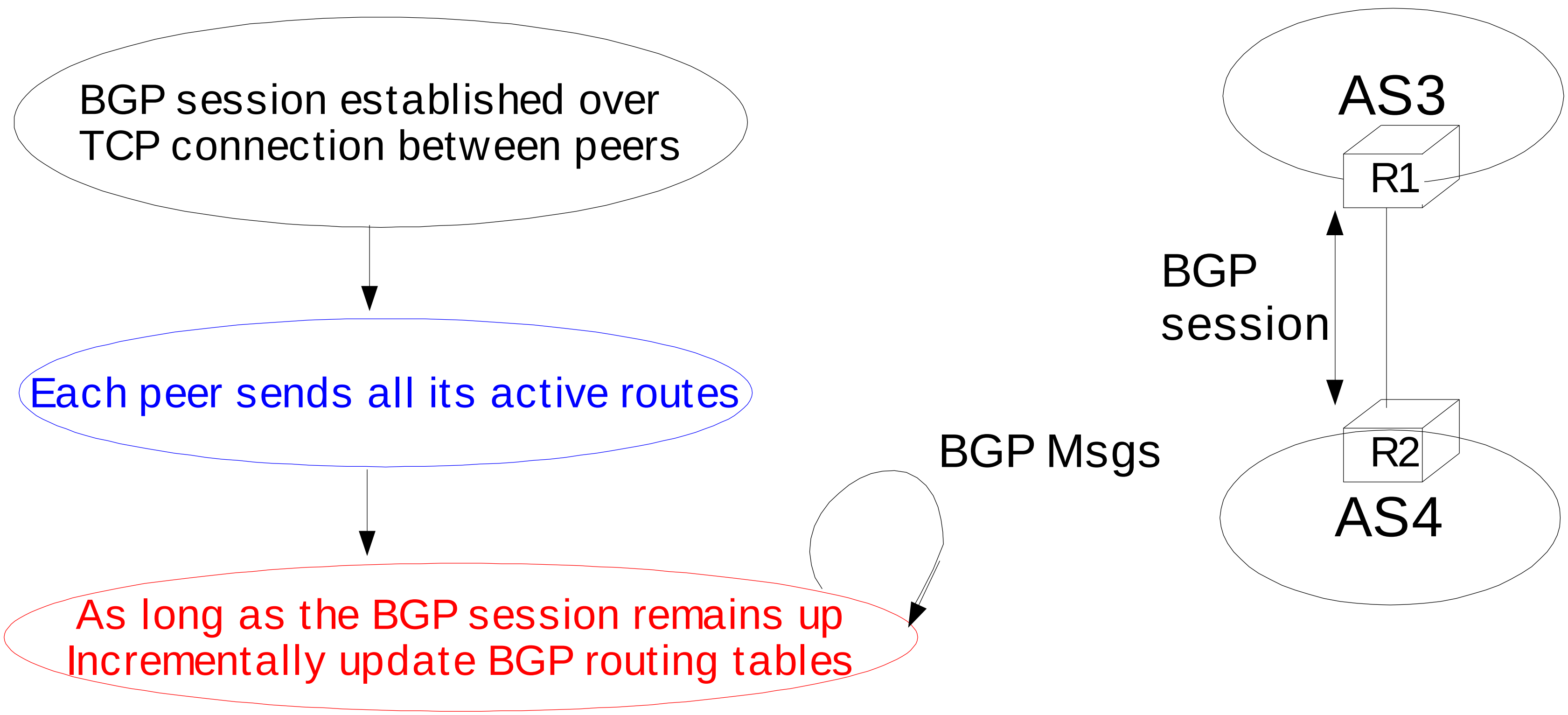
<u>Export filter(Peer[i])</u>
Determines which
routes can be sent to Peer[i]

<u>BGP Routing Information Base</u>
Contains all the acceptable routes
learned from all Peers + internal routes

- BGP decision process selects
  the best route towards each destination

BGP : Principles of operation

- Principles
  - BGP relies on the incremental exchange of path vectors



BGP : Principles of operation (2)

- Simplified model of BGP
  - 2 types of BGP path vectors
  - UPDATE
    - Used to announce a route towards one prefix
    - Content of UPDATE
      - Destination address/prefix
      - Interdomain path used to reach destination (AS-Path)
      - Nexthop (address of the router advertising the route)
  - WITHDRAW
    - Used to indicate that a previously announced route is not reachable anymore
    - Content of WITHDRAW
      - Unreachable destination address/prefix

BGP : Session Initialization

```
Initialize_BGP_Session(RemoteAS, RemoteIP)
{ /* Initialize and start BGP session */
/* Send BGP OPEN Message to RemoteIP on port 179*/
/* Follow BGP state machine */

/* advertise local routes and routes learned from peers*/
foreach (destination=d inside BGP-Loc-RIB)
 {
  B=build_BGP_UPDATE(d);
  S=apply_export_filter(RemoteAS,B);
  if (S<>NULL)
     { /* send UPDATE message */
       send_UPDATE(S,RemoteAS, RemoteIP)
     }
 }
/* entire RIB was sent */
/* new UPDATE will be sent only to reflect local or distant
   changes in routes */
...
}
```

Events during a BGP session

1. Addition of a new route to RIB
   - A new internal route was added on local router
     - static route added by configuration
     - Dynamic route learned from IGP
   - Reception of UPDATE message announcing a new or modified route
2. Removal of a route from RIB
   - Removal of an internal route
     - Static route is removed from router configuration
     - Intradomain route declared unreachable by IGP
   - Reception of WITHDRAW message
3. Loss of BGP session
   - All routes learned from this peer removed from RIB

Export and Import filters

```
BGPMsg Apply_export_filter(RemoteAS, BGPMsg)
{ /* check if Remote AS already received route */
if (RemoteAS isin BGPMsg.ASPath)
   BGPMsg==NULL;
/* Many additional export policies can be configured : */
/* Accept or refuse the BGPMsg */
/* Modify selected attributes inside BGPMsg */
}

BGPMsg apply_import_filter(RemoteAS, BGPMsg)
{ /* check that we are not already inside  ASPath */
 if (MyAS isin BGPMsg.ASPath)
   BGPMsg==NULL;
/* Many additional import policies can be configured : */
/* Accept or refuse the BGPMsg */
/* Modify selected attributes inside BGPMsg */
}
```

BGP : Processing of UPDATES

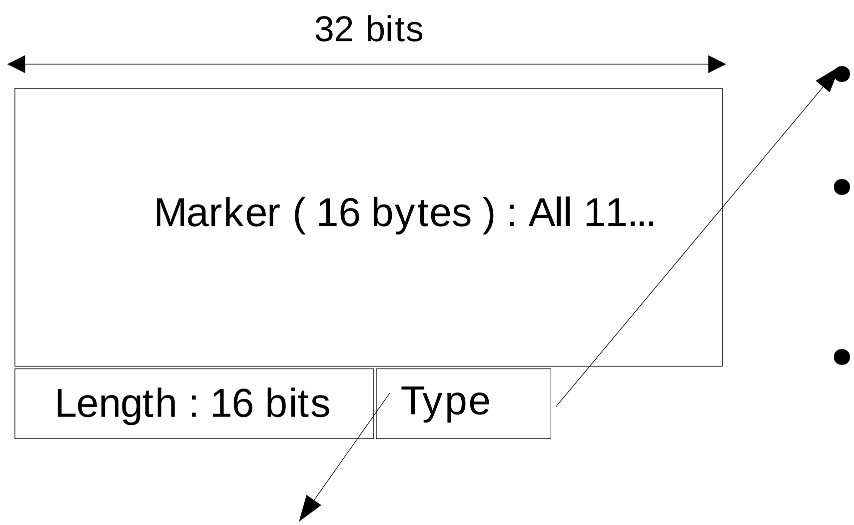
```
Recvd_BGPMsg(Msg, RemoteAS)
{
 B=apply_import_filer(Msg,RemoteAS);
 if (B==NULL) /* Msg not acceptable */
     exit();
 if IsUPDATE(Msg)
 {
  Old_Route=BestRoute(Msg.prefix);
  Insert_in_RIB(Msg);
  Run_Decision_Process(RIB);
  if (BestRoute(Msg.prefix)<>Old_Route)
  { /* best route changed */
    B=build_BGP_Message(Msg.prefix);
    S=apply_export_filter(RemoteAS,B);
    if (S<>NULL) /* announce best route */
     send_UPDATE(S,RemoteAS);
    else if (Old_Route<>NULL)
     send_WITHDRAW(Msg.prefix);
 } ...
```

BGP : Processing of WITHDRAW

```
Recvd_Msg(Msg, RemoteAS)
...
if IsWITHDRAW(Msg)
 {
  Old_Route=BestRoute(Msg.prefix);
  Remove_from_RIB(Msg);
  Run_Decision_Process(RIB);
  if (Best_Route(Msg.prefix)<>Old_Route)
  { /* best route changed */
    B=build_BGP_Message(d);
    S=apply_export_filter(RemoteAS,B);
    if (S<>NULL) /* still one best route */
       send_UPDATE(S,RemoteAS, RemoteIP);
    else if(Old_Route<>NULL)/* no best route anymore */
       send_WITHDRAW(Msg.prefix,RemoteAS,RemoteIP);
  }
 }
}
```

The BGP messages

- Variable length messages
  - With fixed size header

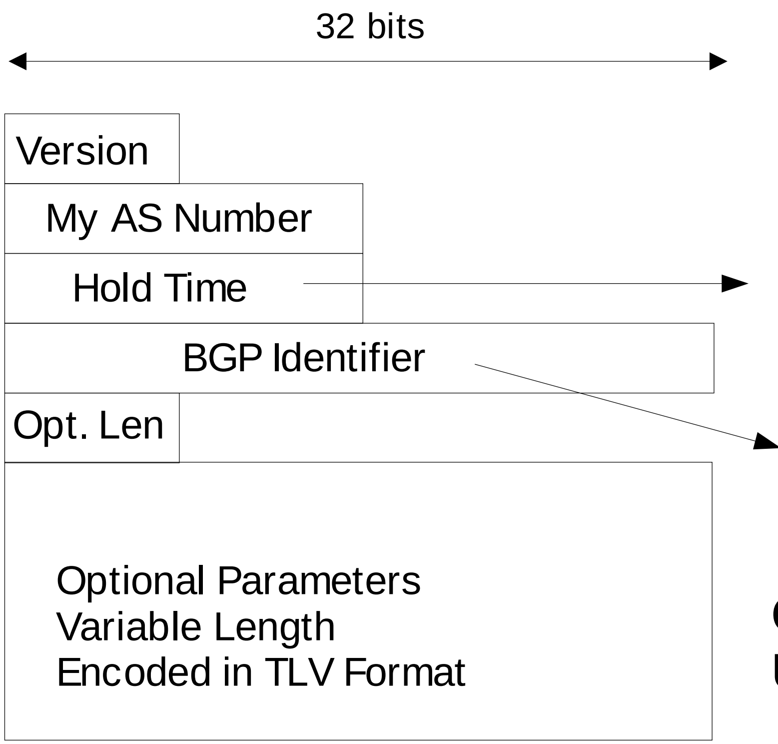


Max length of BGP messages : **4096** bytes

- OPEN
  used to establish BGP session
- UPDATE
  used to send new routes and to remove unusable routes
- NOTIFICATION
  used to inform the remote peer of an error
  BGP session is closed upon transmission or reception of NOTIFICATION message
- KEEPALIVE
  one message must be sent at least every 30 seconds on each BGP session
- ROUTE_REFRESH
  used to support graceful restart

The OPEN message

- Used to establish a BGP session between two BGP peers



Currently version 4

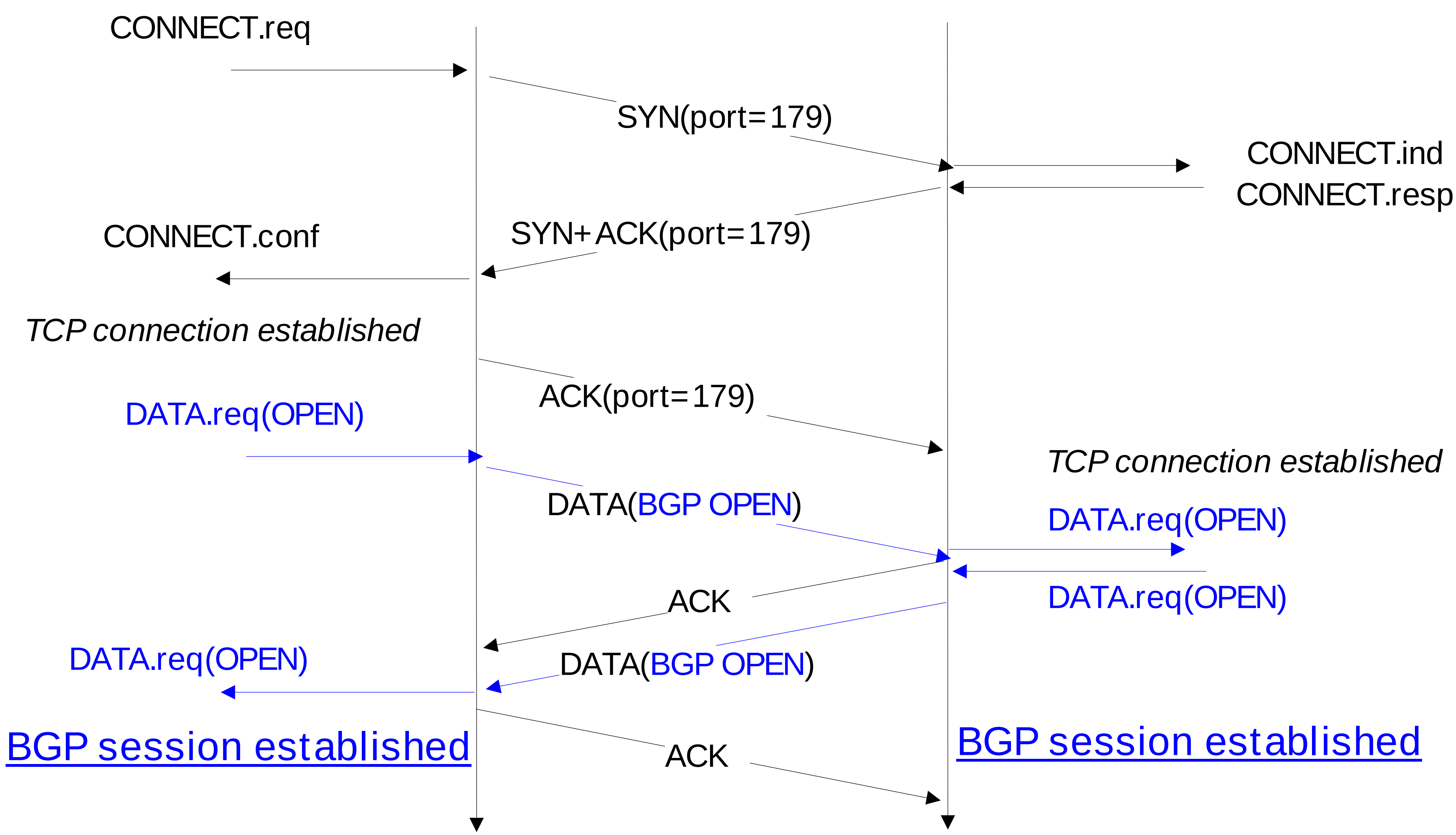
AS # of the BGP peer sending the message

Hold Time : maximum delay between successive
KEEPALIVE, and/or UPDATE messages

BGP Id : Usually IP v4 loopback address
of BGP peer

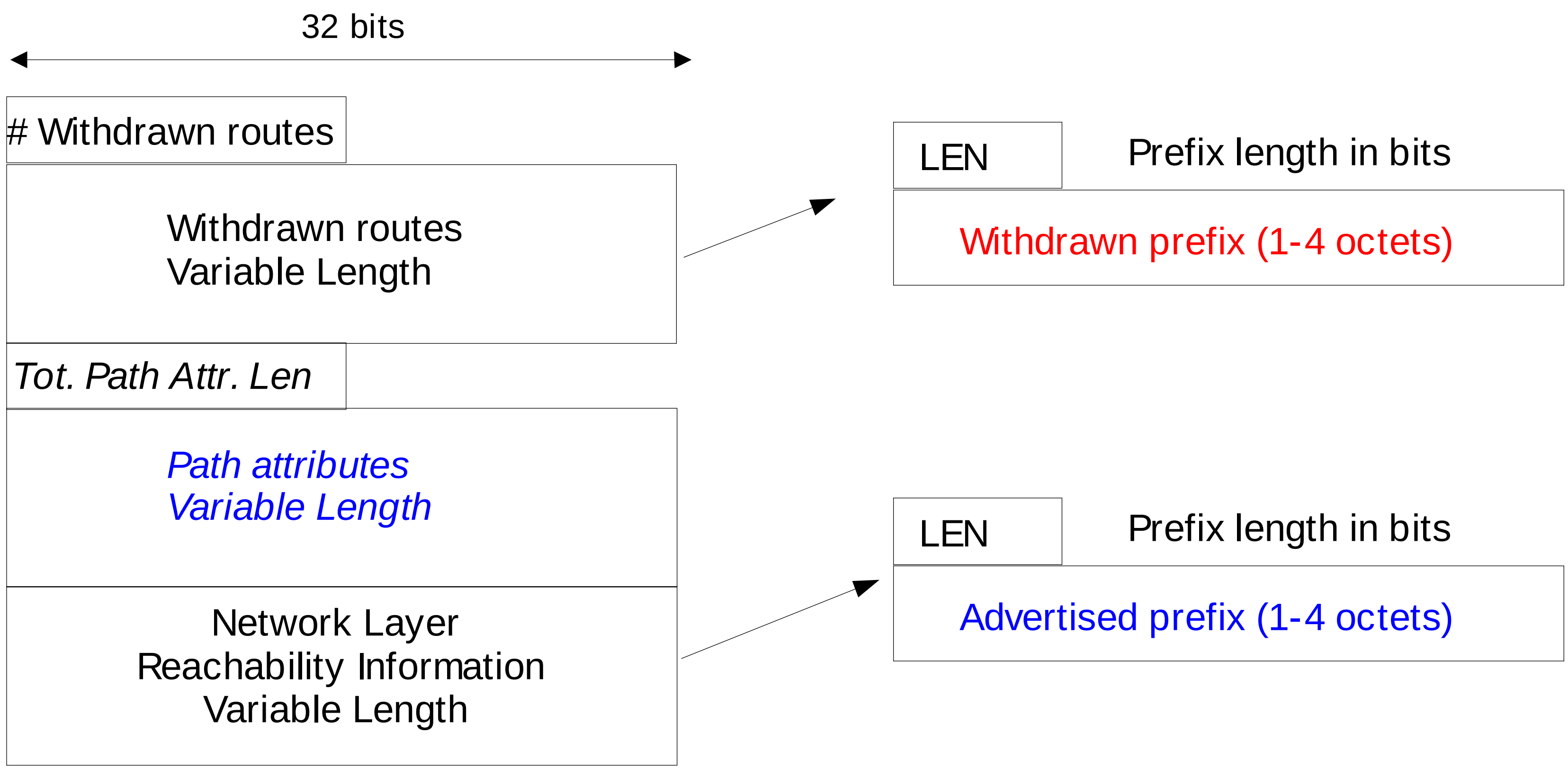
Optional field :
Used notably for capabilities negotiation

Establishment of a BGP session



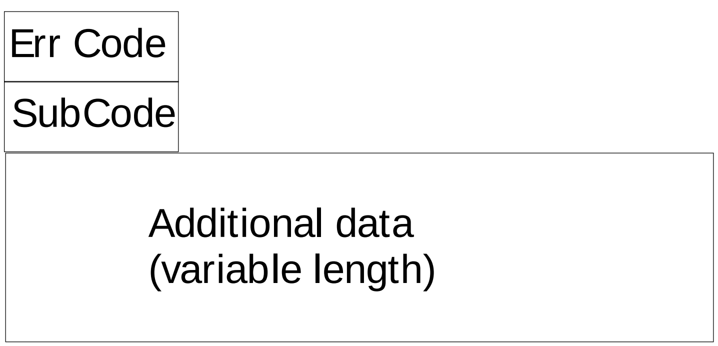
The UPDATE message

- Single message type used to carry both IP v4 route announcements and route withdrawals



The KEEPALIVE and NOTIFICATION messages

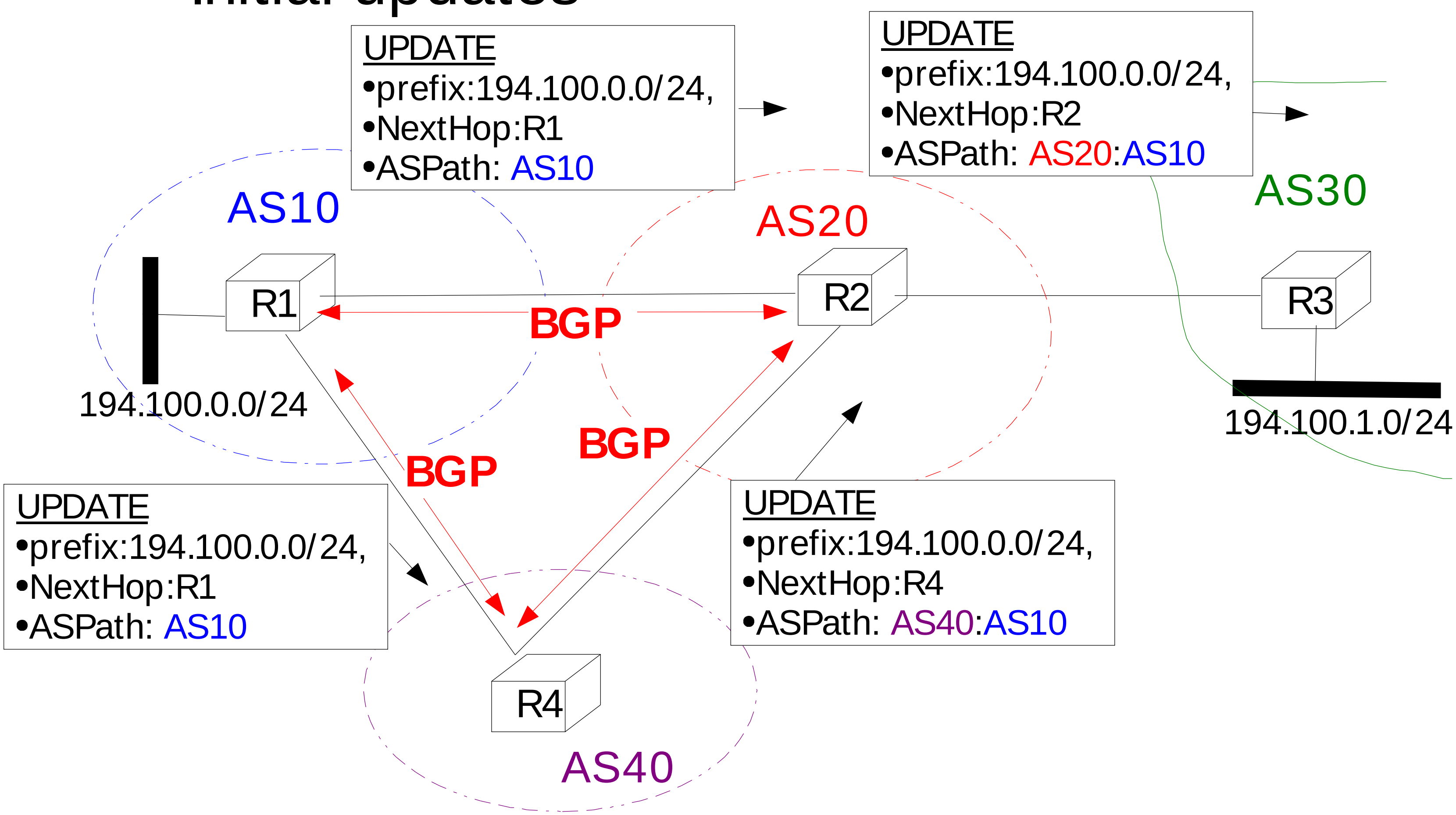
- The KEEPALIVE message
  - BGP Message containing only the default header
  - Every HoldTime/3 seconds, send a KEEPALIVE message if no recent BGP message was sent
- The NOTIFICATION message
  - indicates problem in processing of BGP message
    - BGP session is released upon transmission/reception of NOTIFICATION



- Example errors :
- 2 : OPEN Message Error
  - Unsupported Version, Unsupported Optional Parameter, ...
- 3 : UPDATE Message Error
  - Malformed Attribute List, ...
- 4 Hold Timer Expired
- 5 Finite State Machine Error
- 6 Cease

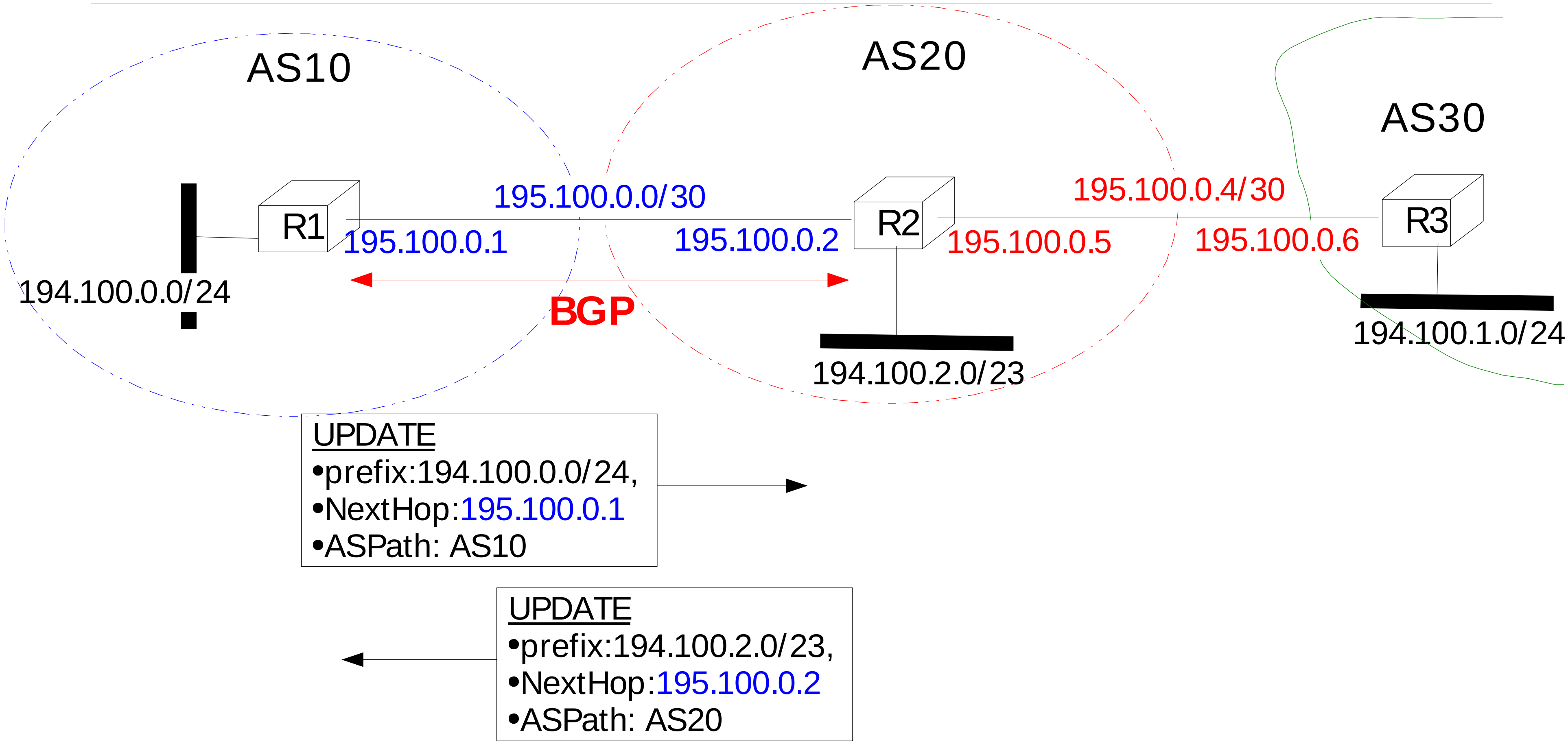
BGP and IP
A first example

- Initial updates



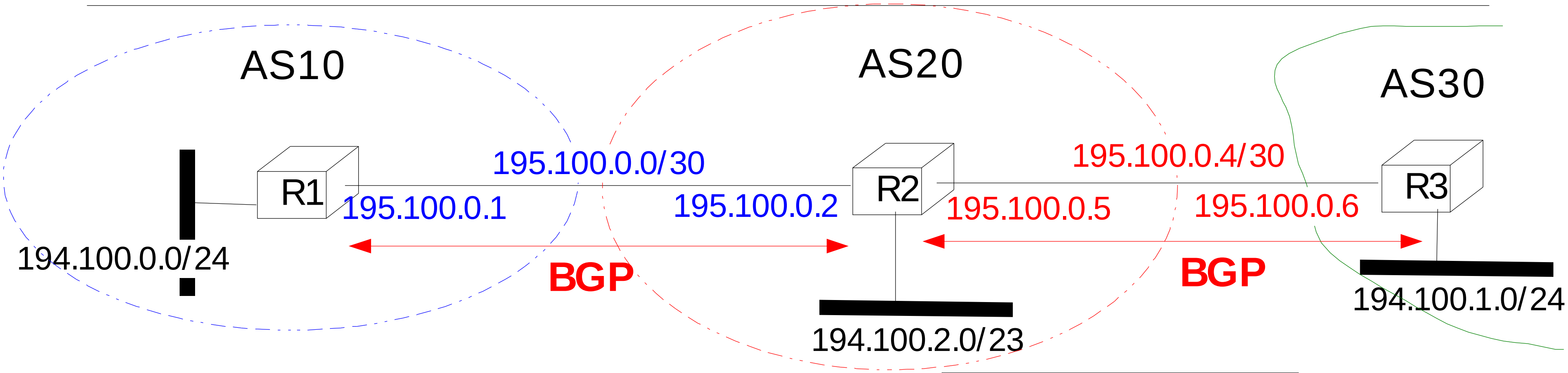
- What happens if link AS10-AS20 goes down ?

BGP and IP
A second example



- Main Path attributes of UPDATE message
  - NextHop : IP address of router used to reach destination
  - ASPath : Path followed by the route advertisement

BGP and IP
A second example (2)



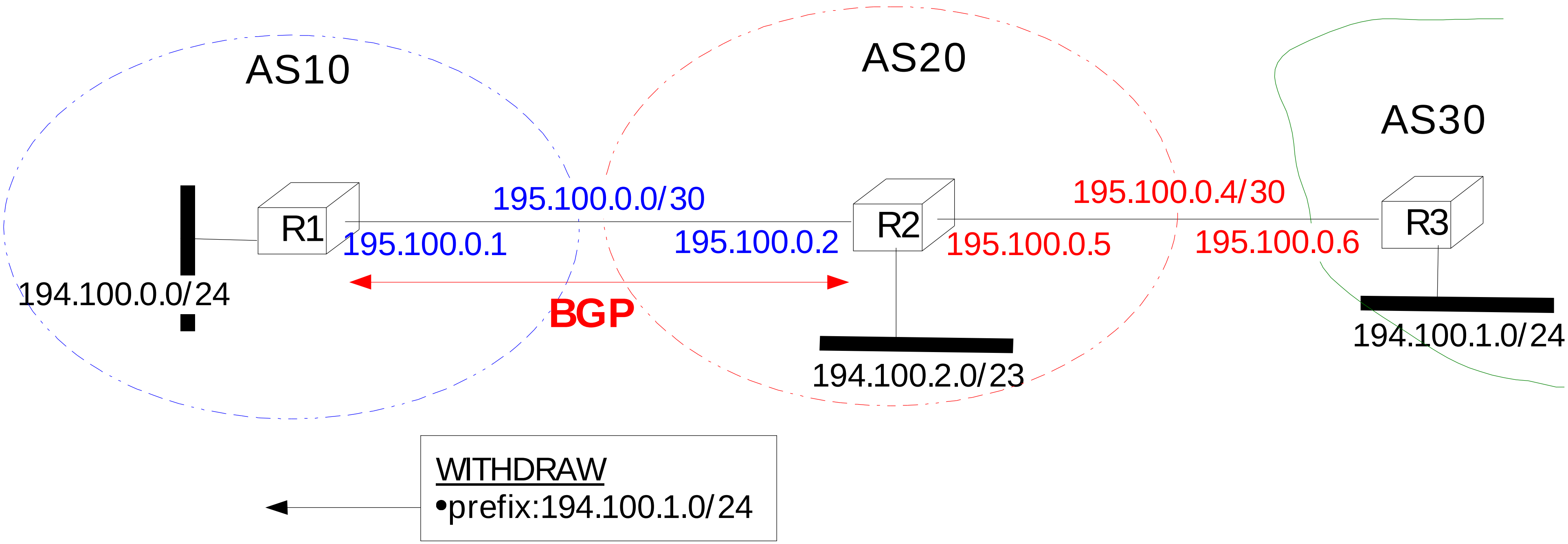
<u>UPDATE</u>
- prefix:194.100.0.0/24
- NextHop:195.100.0.5
- ASPath: AS20:AS10

<u>UPDATE</u>
- prefix:194.100.2.0/23
- NextHop:195.100.0.5
- ASPath: AS20

<u>UPDATE</u>
- prefix:194.100.1.0/24,
- NextHop:195.100.0.2
- ASPath: AS20;AS30

<u>UPDATE</u>
- prefix:194.100.1.0/24,
- NextHop:195.100.0.6
- ASPath: AS30

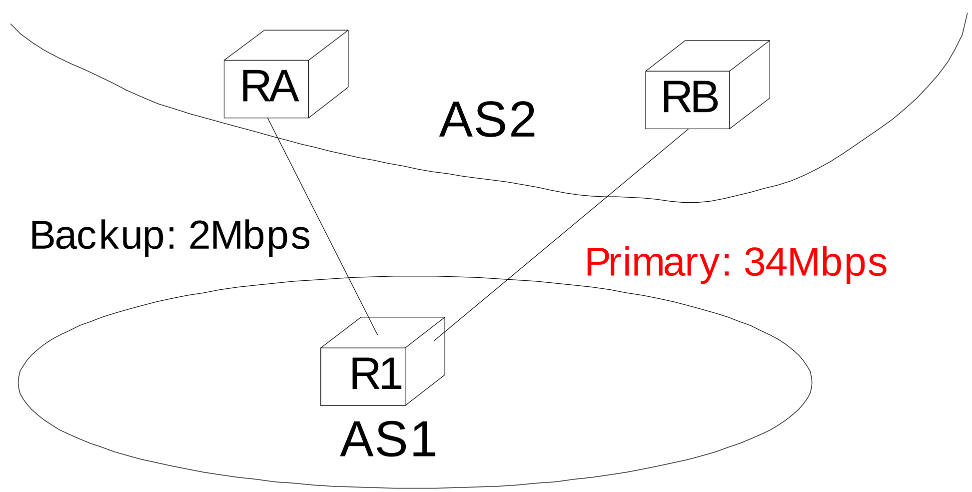
BGP and IP
A second example (3)



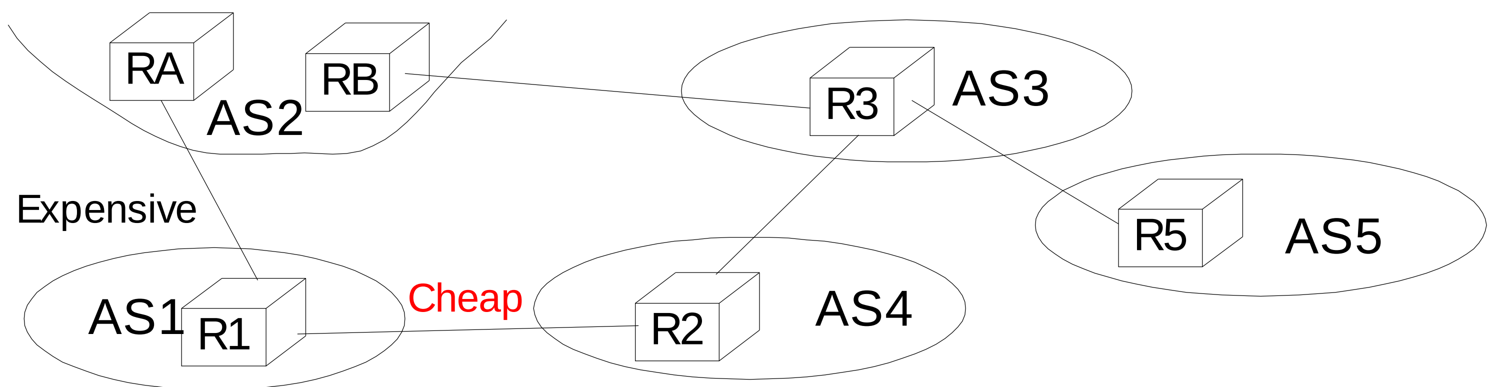
Outline

- Organization of the global Internet
- BGP basics
  - Routing policies
  - The Border Gateway Protocol
  - How to prefer some routes over others
- BGP in large networks
- Interdomain traffic engineering with BGP
- BGP-based Virtual Private Networks

How to prefer some routes over others ?

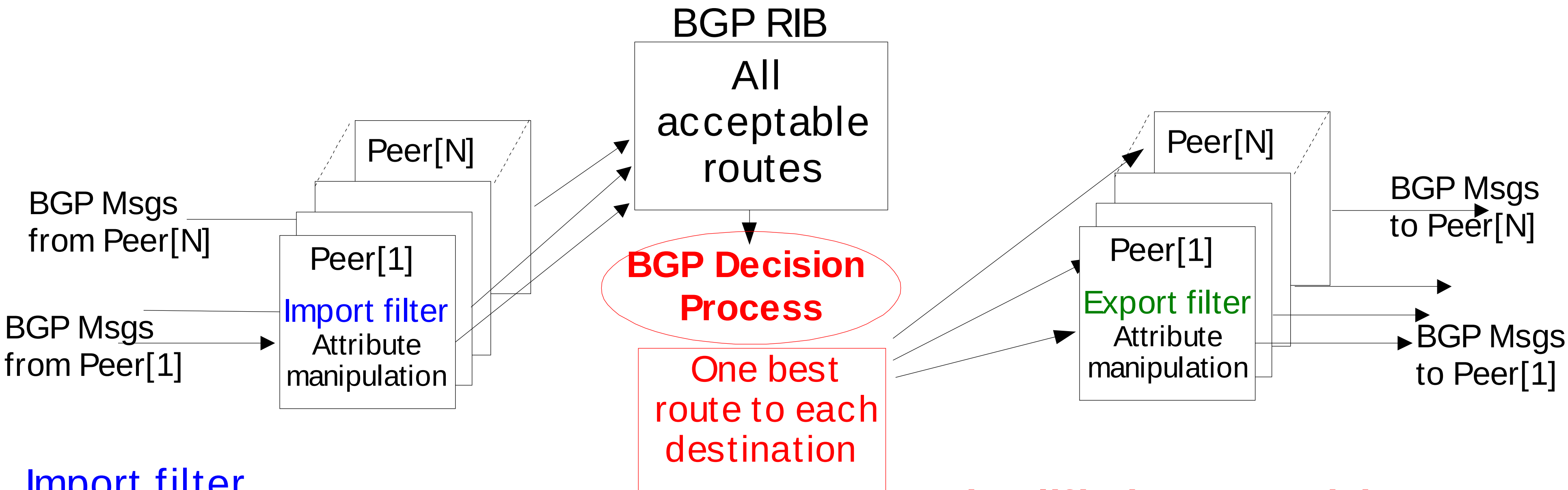


- How to ensure that packets will flow on primary link ?



- How to prefer cheap link over expensive link ?

How to prefer some routes over others (2) ?



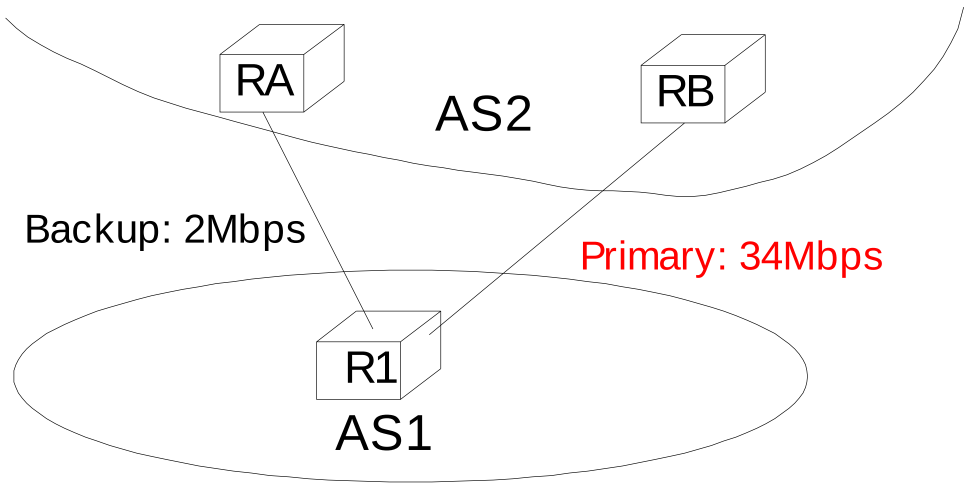
Import filter

- Selection of acceptable routes
- Addition of `local-pref` attribute inside received BGP Msg
  - Normal quality route : `local-pref=100`
  - Better than normal route :`local-pref=200`
  - Worse than normal route :`local-pref=50`

Simplified BGP Decision Process

- Select routes with highest `local-pref`
- If there are several routes, choose routes with the shortest ASPath
- If there are still several routes tie-breaking rule

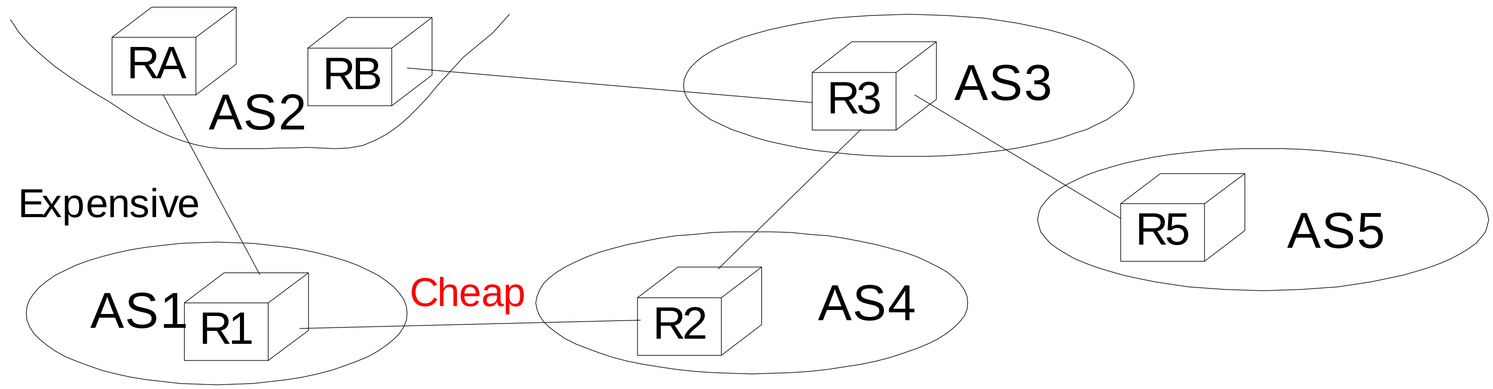
How to prefer some routes over others (3) ?



RPSL-like policy for AS1
aut-num: AS1
import: from AS2 RA at R1 set localpref=100;
from AS2 RB at R1 set localpref=200;
accept ANY
export: to AS2 RA at R1 announce AS1
to AS2 RB at R1 announce AS1

RPSL-like policy for AS2
aut-num: AS2
import: from AS1 R1 at RA set localpref=100;
from AS1 R1 at RB set localpref=200;
accept AS1
export: to AS1 R1 at RA announce ANY
to AS2 R1 at RB announce ANY

How to prefer some routes over others (4) ?



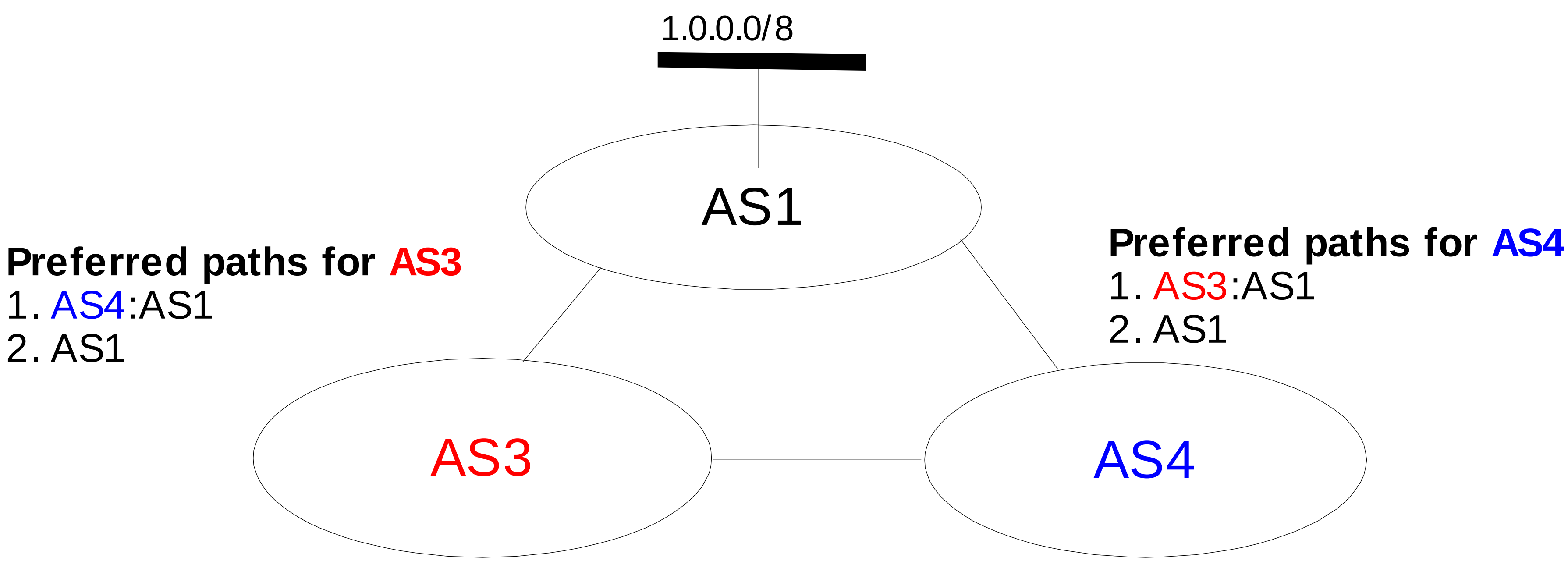
RPSL policy for AS1

aut-num: AS1
import: from AS2 RA at R1 set localpref=100;
from AS4 R2 at R1 set localpref=200;
accept ANY
export: to AS2 RA at R1 announce AS1
to AS4 R2 at R1 announce AS1

- AS1 will prefer to send packets over the cheap link
- But the flow of the packets destined to AS1 will depend on the routing policy of the other domains

Limitations of `local-pref`

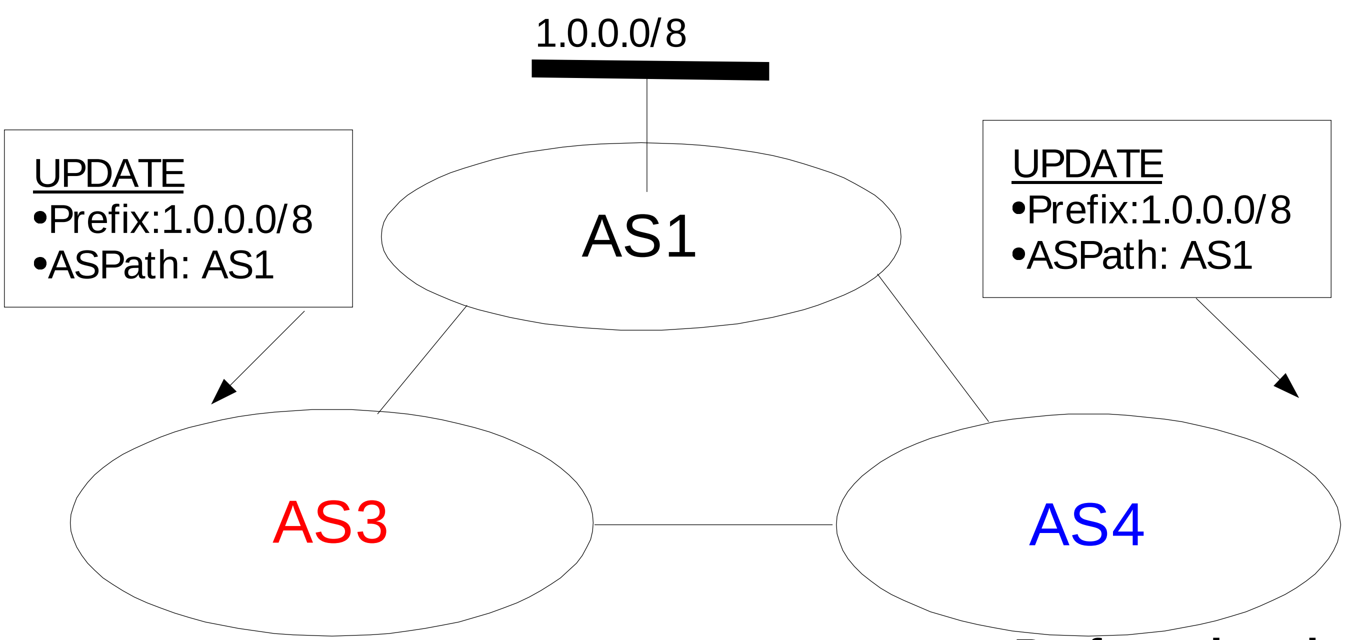
- In theory
  - Each domain is free to define its order of preference for the routes learned from external peers



  - How to reach 1.0.0.0/8 from AS3 and AS4 ?

Limitations of `local-pref` (2)

- AS1 sends its UPDATE messages ...



Preferred paths for AS3
1. AS4:AS1
2. AS1

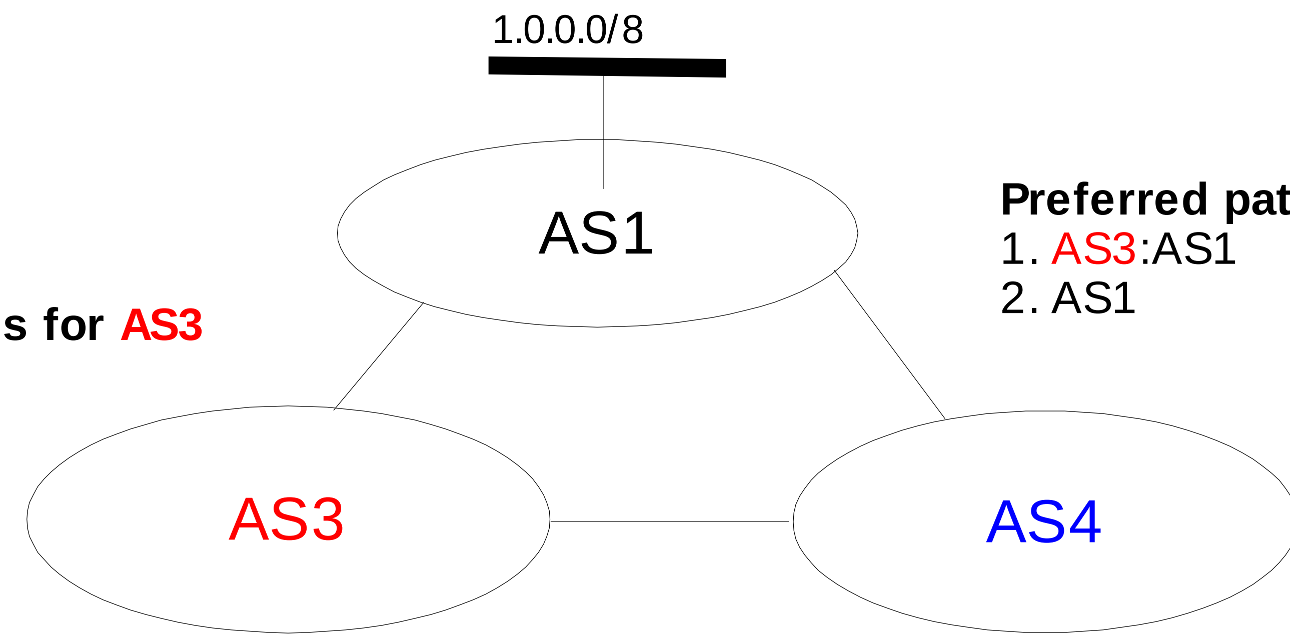
Preferred paths for AS4
1. AS3:AS1
2. AS1

Routing table for AS3
1.0.0.0/8 ASPath: AS1 (best)

Routing table for AS4
1.0.0.0/8 ASPath: AS1 (best)

Limitations of `local-pref` (3)

- First possibility
  - AS3 sends its UPDATE first...



Preferred paths for AS4
1. AS3:AS1
2. AS1

Preferred paths for AS3
1. AS4:AS1
2. AS1

Routing table for AS3
1.0.0.0/8 ASPath: AS1 (best)

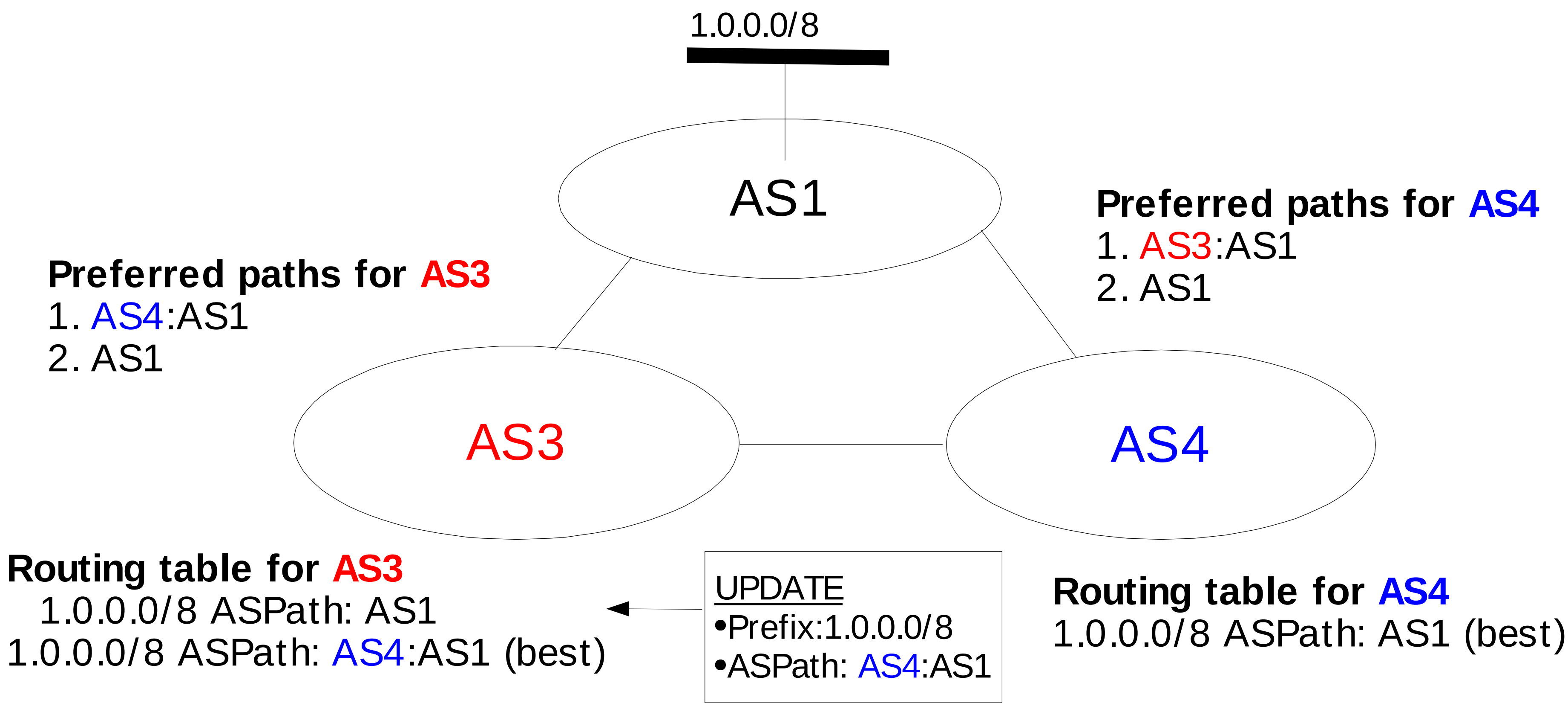
UPDATE
•Prefix:1.0.0.0/8
•ASPath: AS3:AS1

Routing table for AS4
1.0.0.0/8 ASPath: AS1
1.0.0.0/8 ASPath:AS3:AS1 (best)

- Stable route assignment

Limitations of `local-pref` (4)

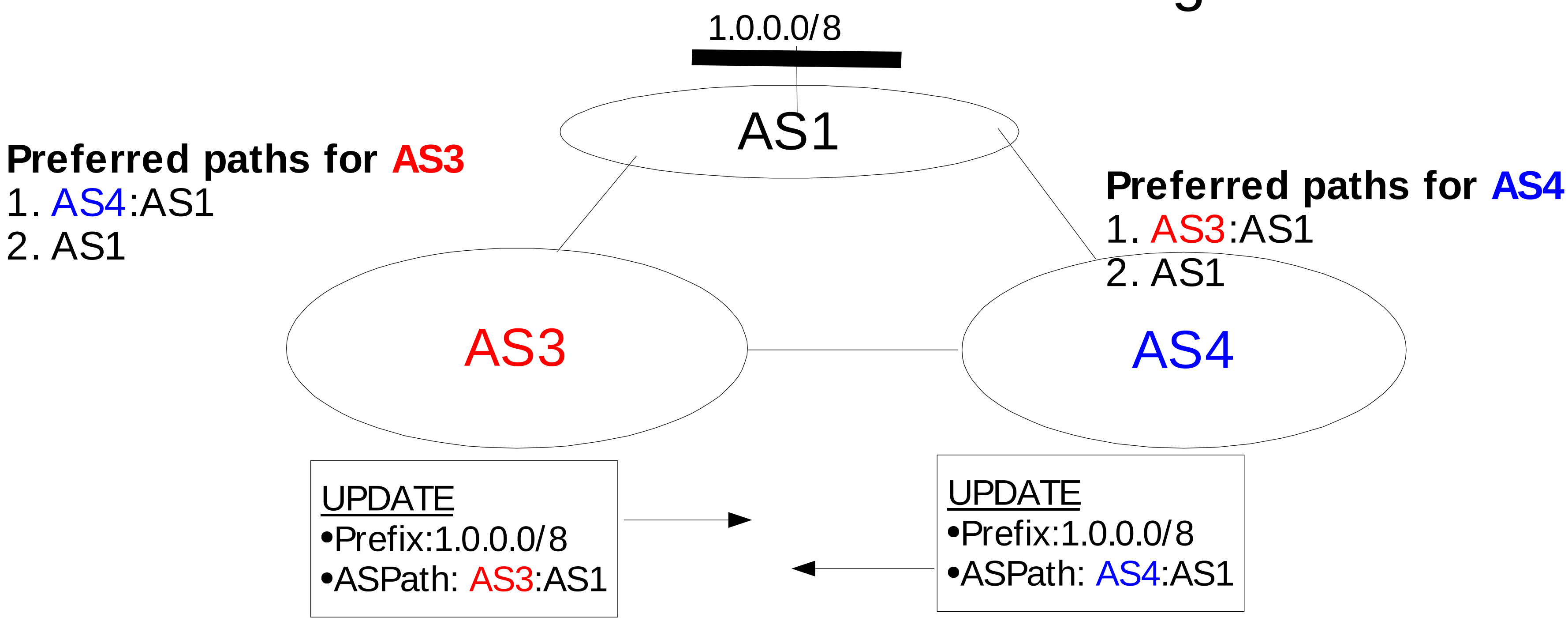
- Second possibility
  - AS4 sends its UPDATE first...



- Another (but different) stable route assignment

Limitations of `local-pref` (5)

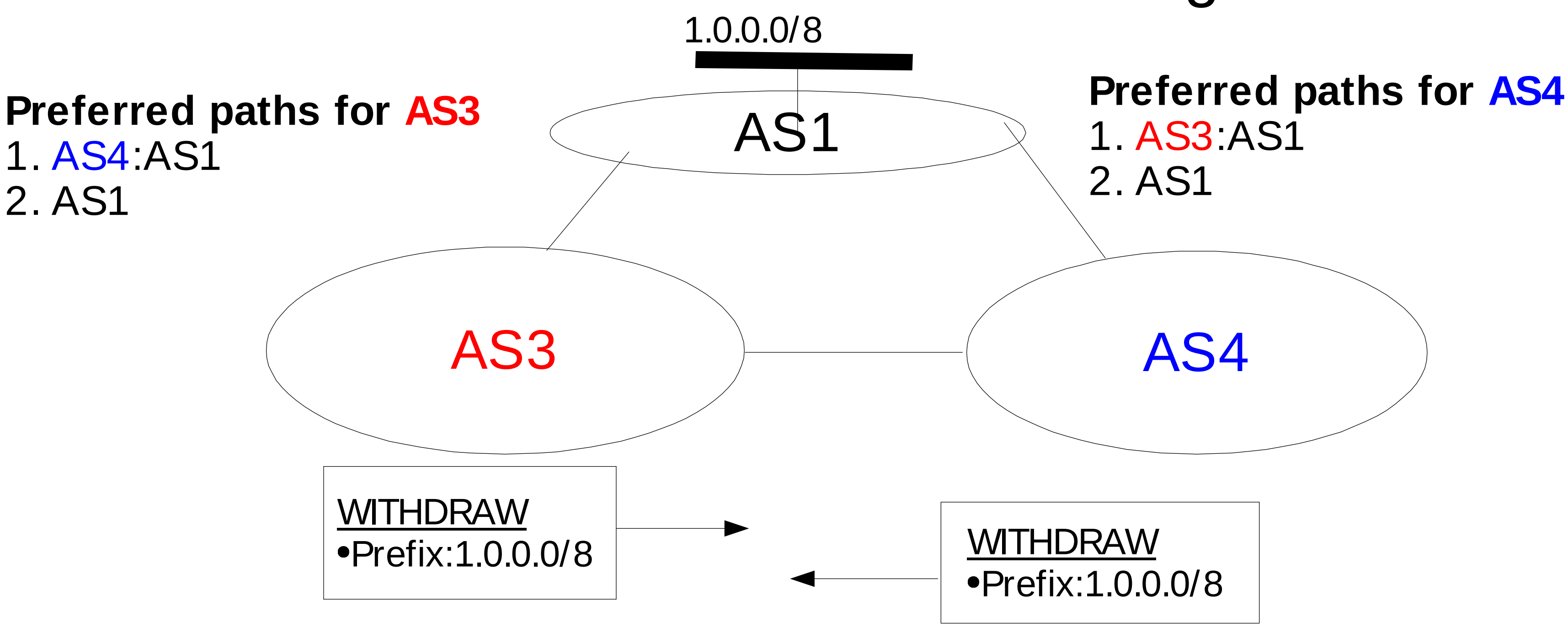
- Third possibility
  - AS3 and AS4 send their UPDATE together...



- ◆ AS3 prefers the indirect path and will thus send withdraw since the chosen best path is via AS4
- ◆ AS4 prefers the indirect path and will thus send withdraw since the chosen best path is via AS3

Limitations of `local-pref` (6)

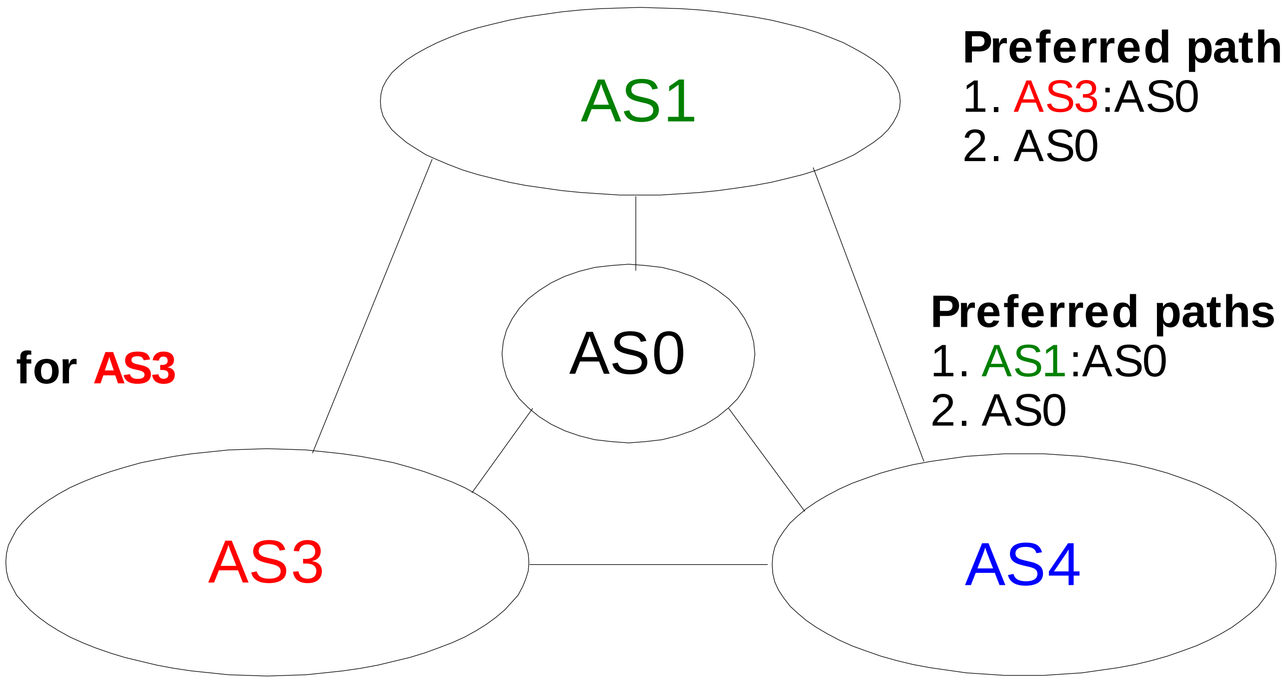
- Third possibility (cont.)
  - AS3 and AS4 send their UPDATE together...



- AS3 learns that the indirect route is not available anymore
  - AS3 will reannounce its direct route...
- AS4 learns that the indirect route is not available anymore
  - AS4 will reannounce its direct route...

More limitations of `local-pref`

- Unfortunately, interdomain routing may not converge at all in some cases...



Preferred paths for AS1
1. AS3:AS0
2. AS0

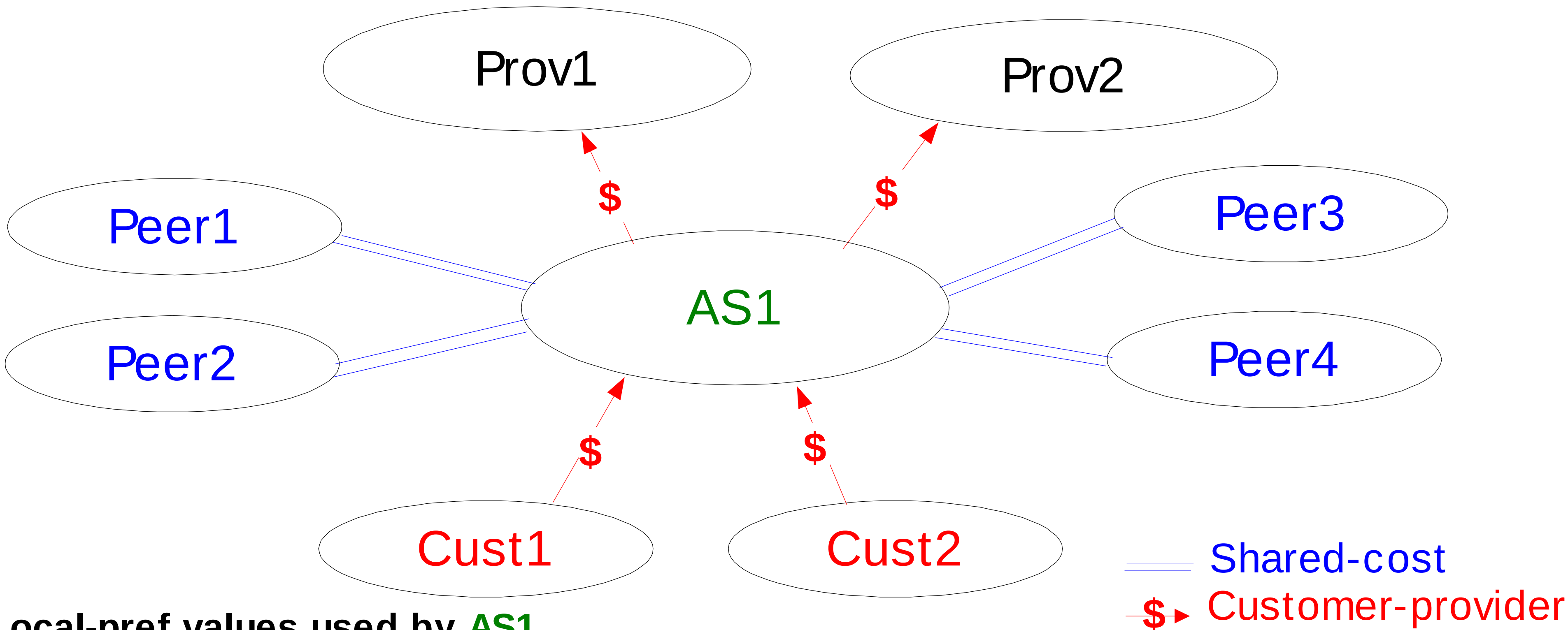
Preferred paths for AS4
1. AS1:AS0
2. AS0

Preferred paths for AS3
1. AS4:AS0
2. AS0

  - How to reach a destination inside AS0 in this case ?

`local-pref` and economical relationships

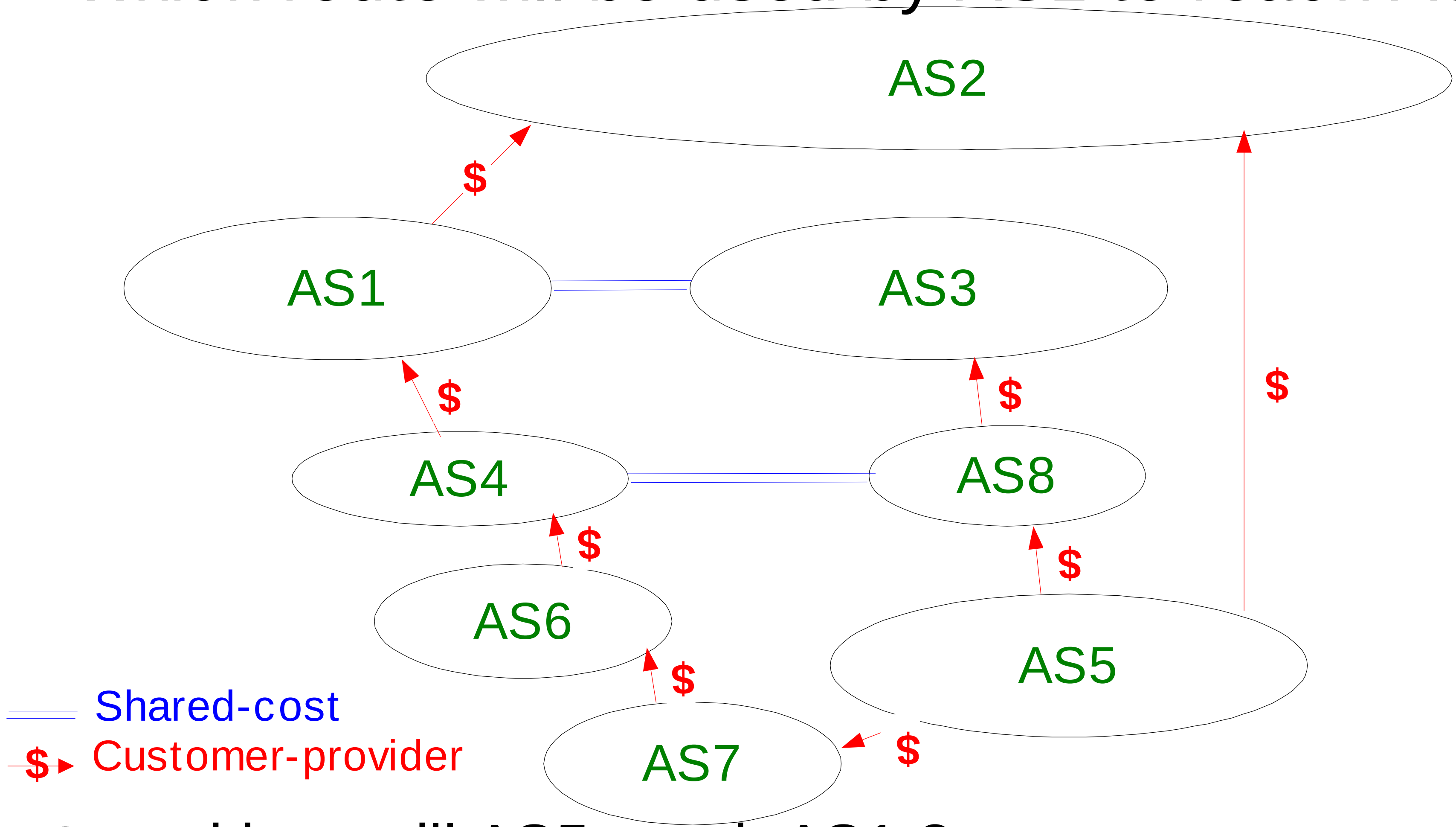
- In practice, `local-pref` is often used to enforce economical relationships



Local-pref values used by AS1
> 1000 for the routes received from a Customer
500 – 999 for the routes learned from a Peer
< 500 for the routes learned from a Provider

Consequence of this utilization of `local-pref`

- Which route will be used by AS1 to reach AS5 ?

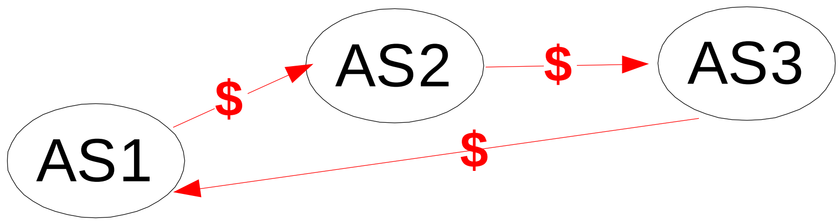


- and how will AS5 reach AS1 ?

Internet paths are often asymmetrical

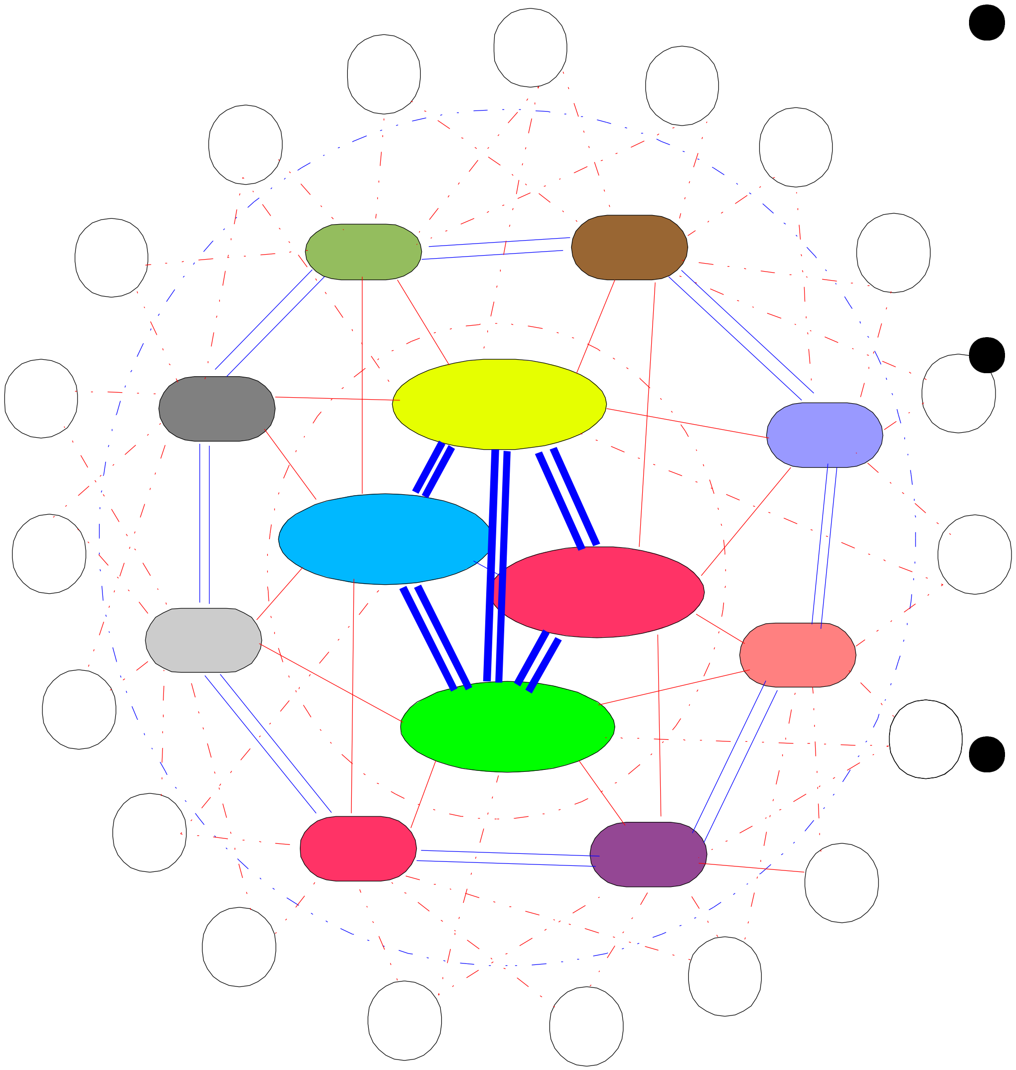
Guidelines for a safe utilization of `local-pref`

- The directed graph composed of the customer->provider links is loop-free
  - An AS cannot be a customer of a provider of its providers



  - An AS always prefer a route via a customer over a route via a provider or a peer
    - With some restrictions on the graph composed of peer-to-peer relationships, it is also possible to allow an AS to give the same preference to a route via a customer or via a peer

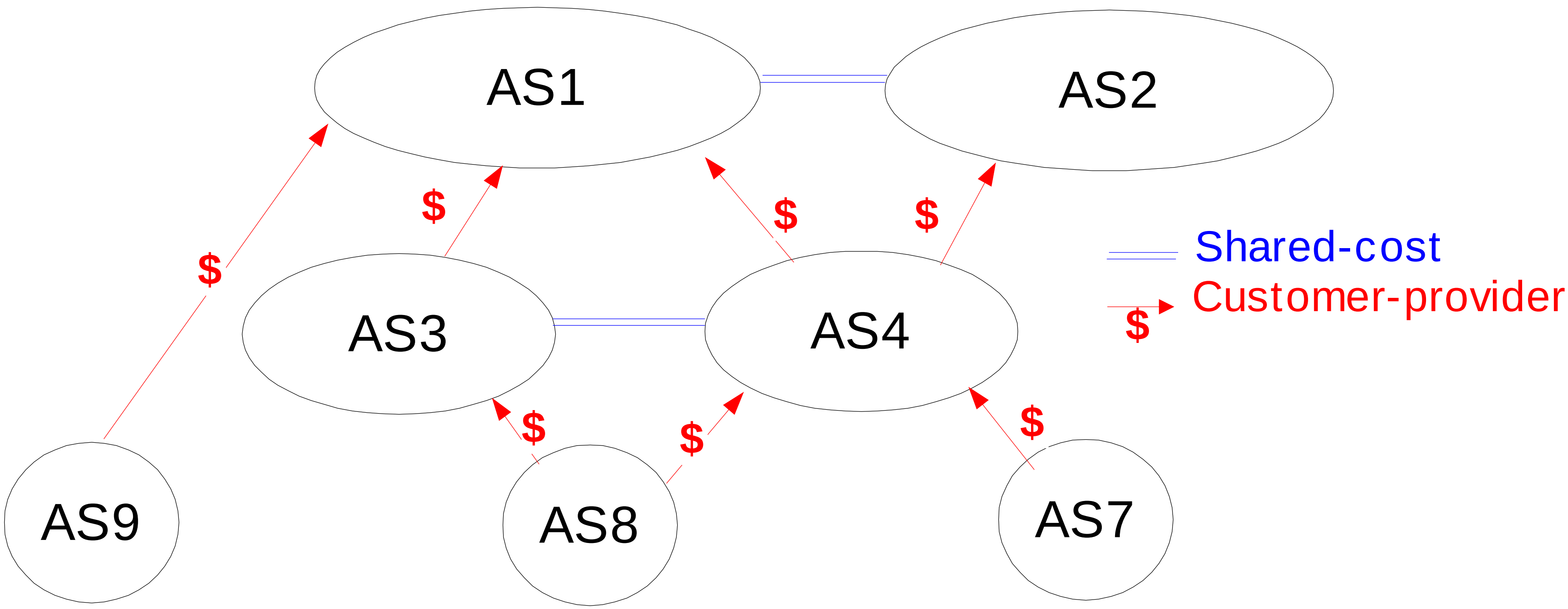
The Organization of the Internet



- Tier-1 ISPs
  - Dozen of large ISPs interconnected by shared-cost
  - Provide transit service
    - Uunet, Level3, OpenTransit, ...
- Tier-2 ISPs
  - Regional or National ISPs
  - Customer of T1 ISP(s)
  - Provider of T2 ISP(s)
  - shared-cost with other T2 ISPs
    - France Telecom, BT, Belgacom
- Tier-3 ISPs
  - Smaller ISPs, Corporate Networks, Content providers
  - Customers of T2 or T1 ISPs
  - shared-cost with other T3 ISPs

Composition of Internet paths

- Most Internet paths contain a sequence of
  - 0 or more Customer->Provider relationships
  - 0 or 1 Peer-to-Peer relationships
  - 0 or more Provider->Customer relationships



Summary

- Routing policies
  - Two main routing policies
    - Customer-Provider relationship
    - Peer-to-Peer relationship
- The Border Gateway Protocol
  - Path vector protocol with incremental updates
  - Import and export filters to implement routing policies
  - Utilization of local-pref
    - Influence BGP decision process
    - Prefer some routes over others
    - Be careful with possible oscillations due to bad setting