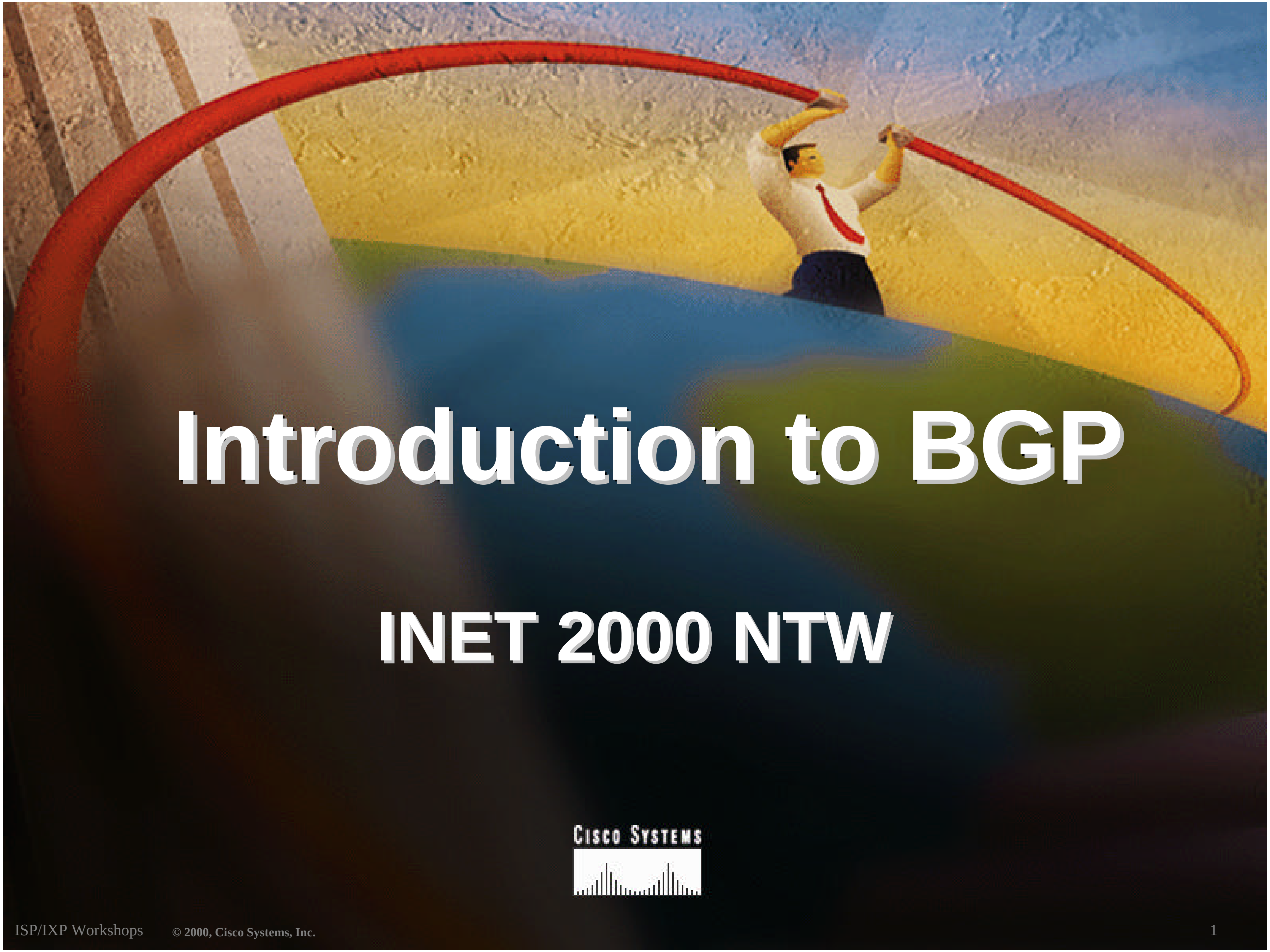
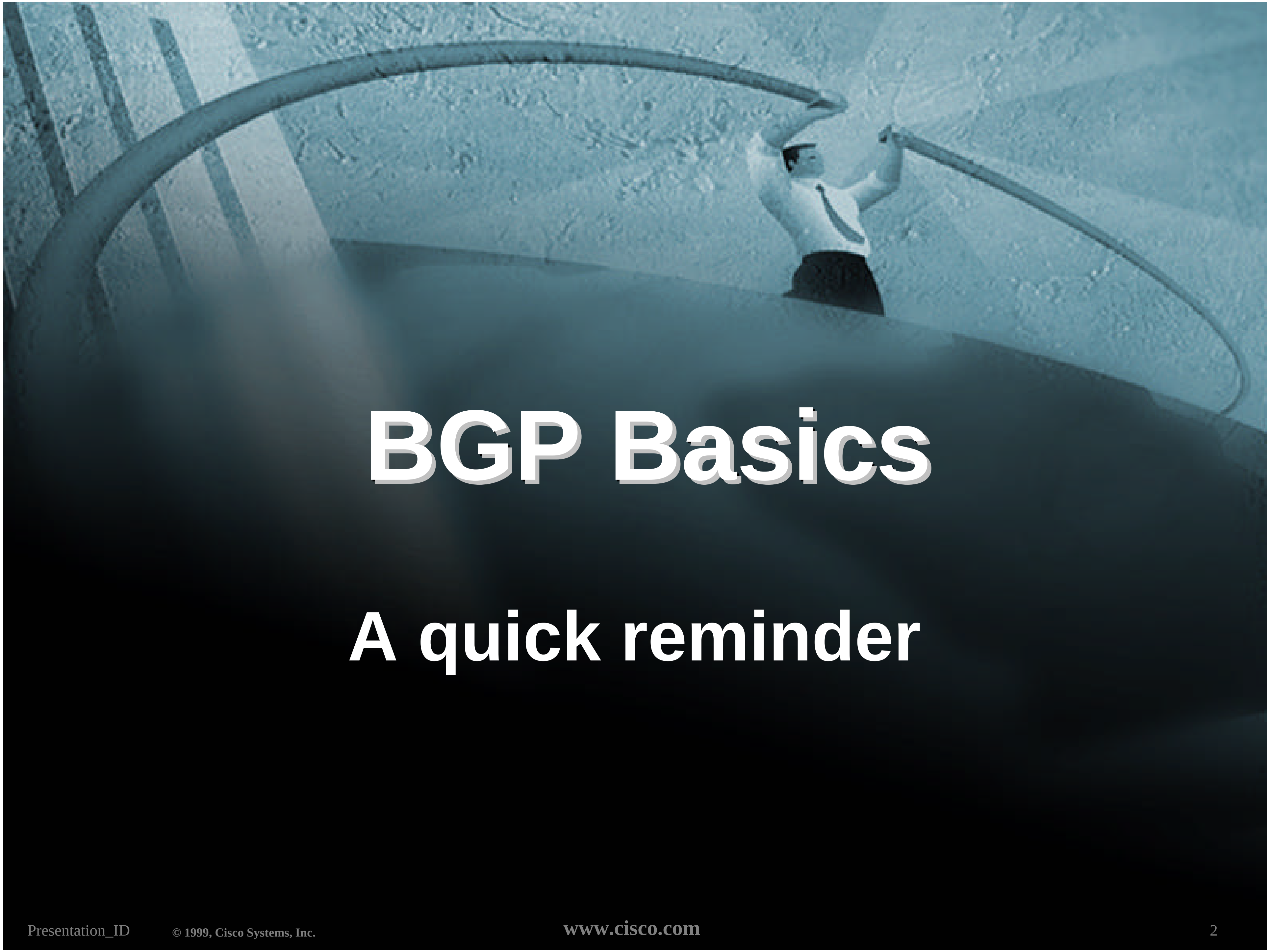


Introduction to BGP

INET 2000 NTW

Cisco Systems



BGP Basics

A quick reminder

Border Gateway Protocol

- Routing Protocol used to exchange routing information between networks

  exterior gateway protocol

- RFC1771

  work in progress to update

  `draft-ietf-idr-bgp4-10.txt`

- Currently Version 4
- Runs over TCP

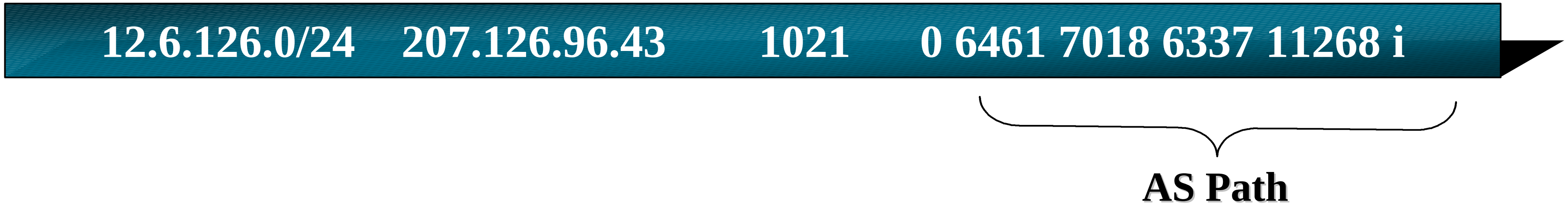
BGP

- Path Vector Protocol
- Incremental Updates
- Many options for policy enforcement
- Classless Inter Domain Routing (CIDR)
- Widely used for Internet backbone
- Autonomous systems

Path Vector Protocol

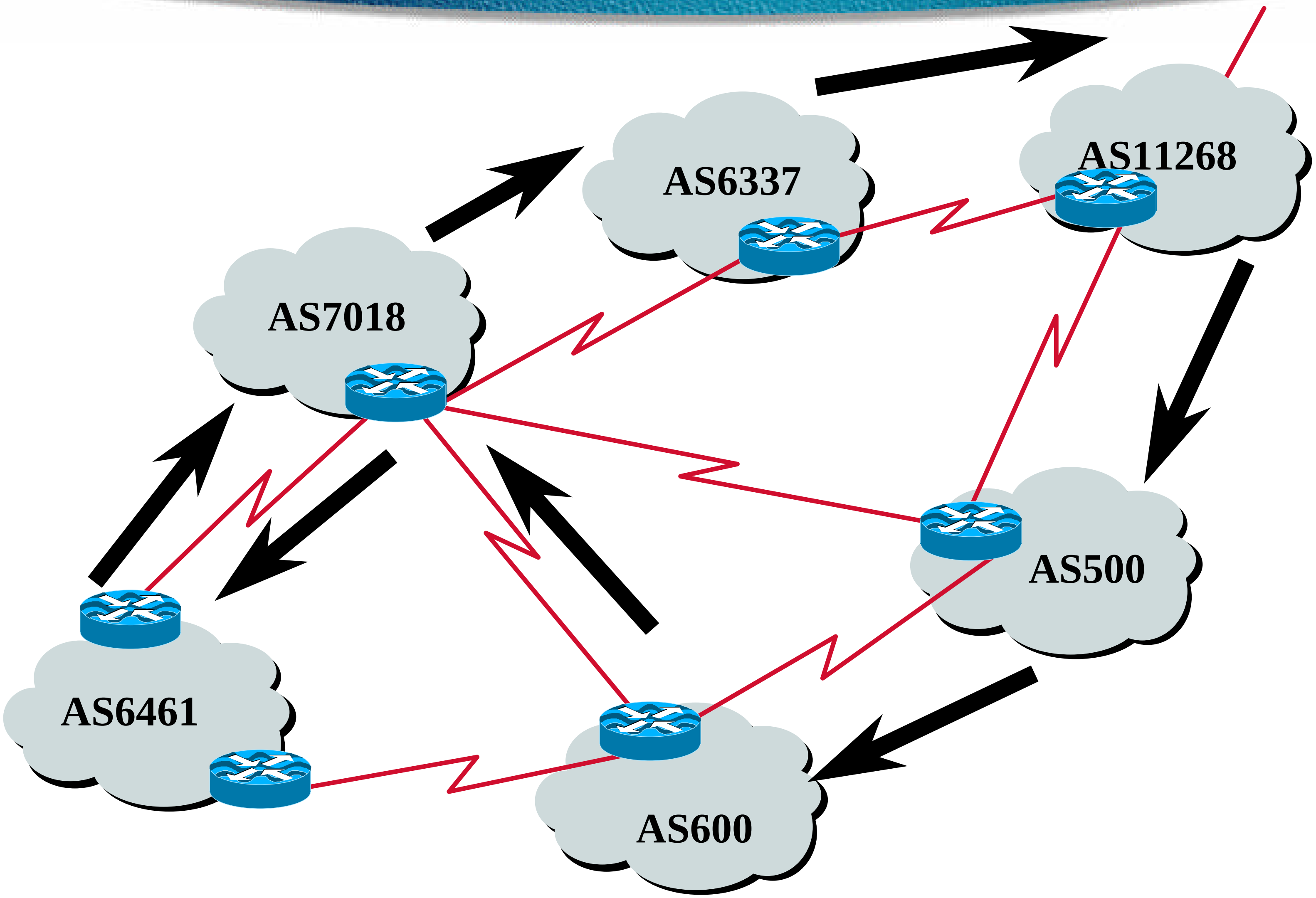
- **BGP is classified as a *path vector* routing protocol** (see RFC 1322)

A path vector protocol defines a route as a pairing between a destination and the attributes of the path to that destination.



12.6.126.0/24 207.126.96.43 1021 0 6461 7018 6337 11268 i

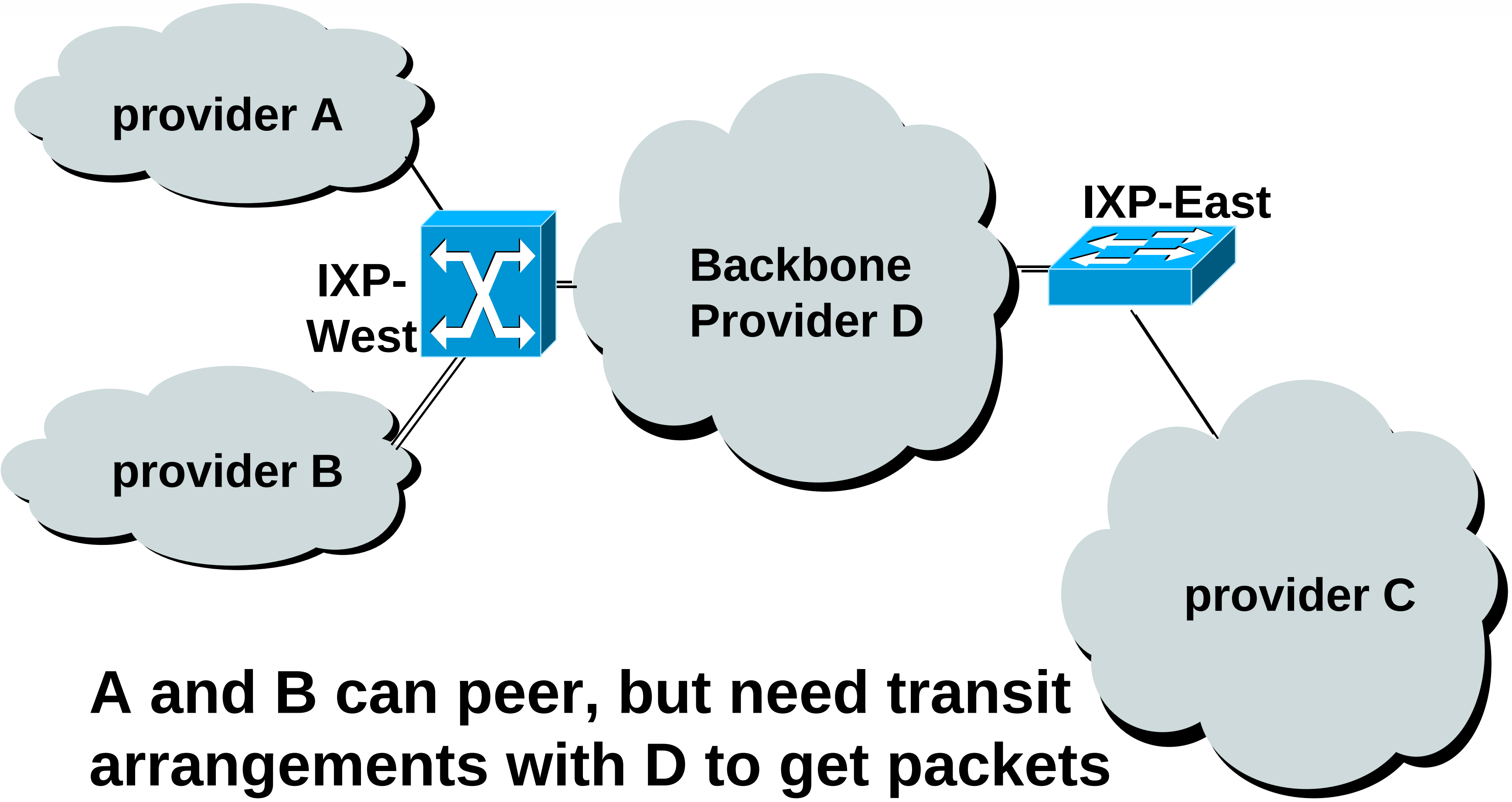
Path Vector Protocol



Definitions

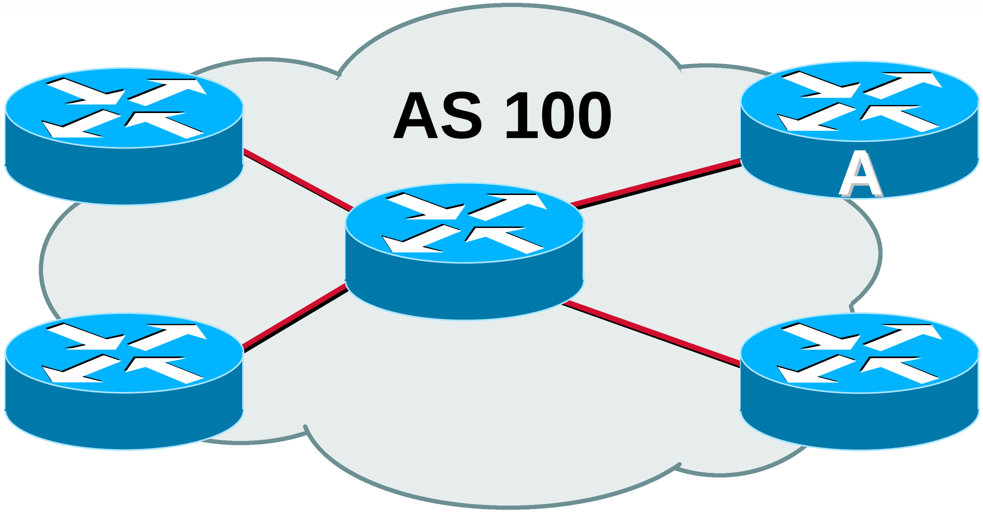
- **Transit** - carrying traffic across a network, usually for a fee
- **Peering** - exchanging routing information and traffic
- **Default** - where to send traffic when there is no explicit match is in the routing table

Peering and Transit example



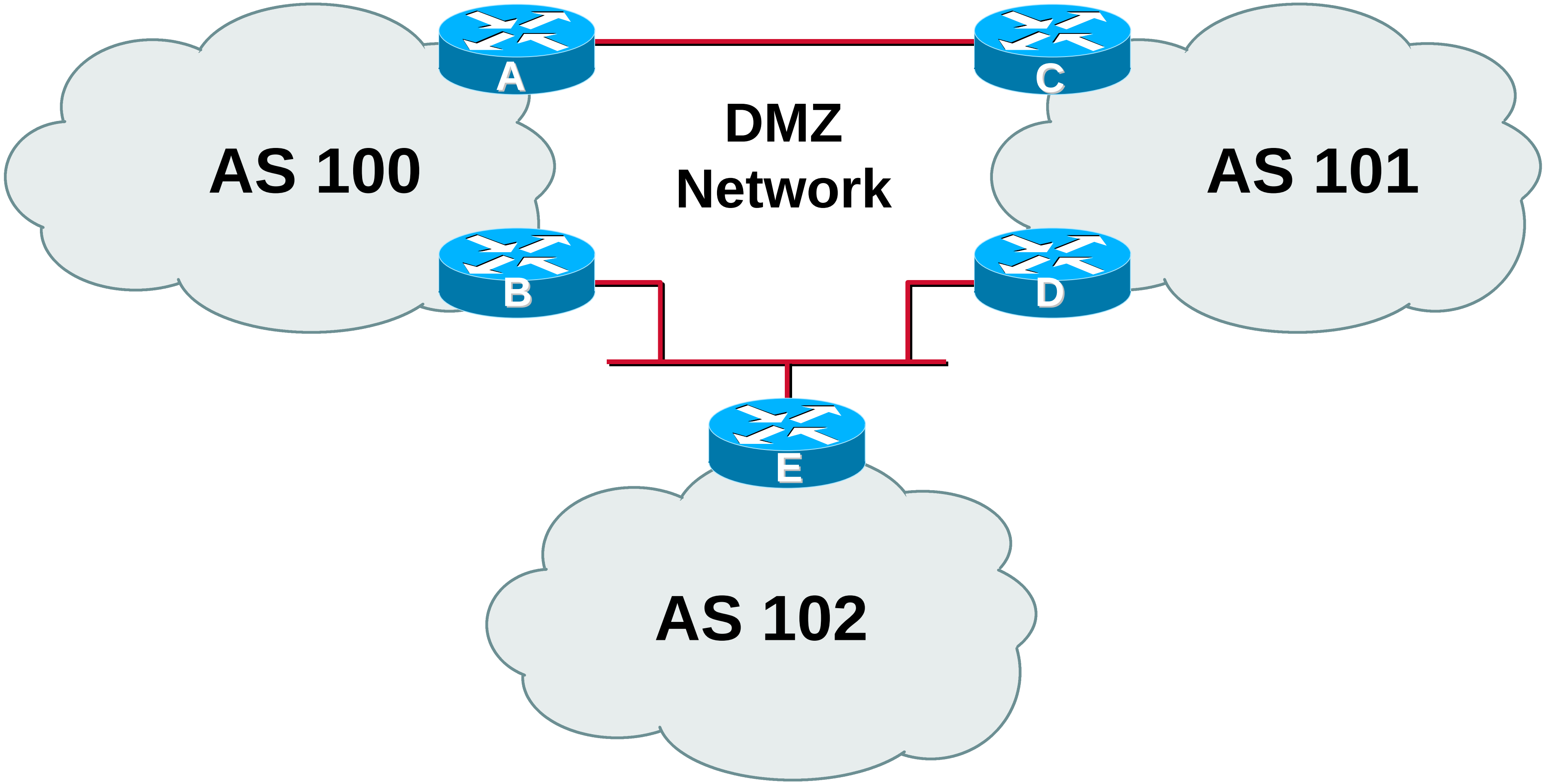
A and B can peer, but need transit arrangements with D to get packets to/from C

Autonomous System (AS)



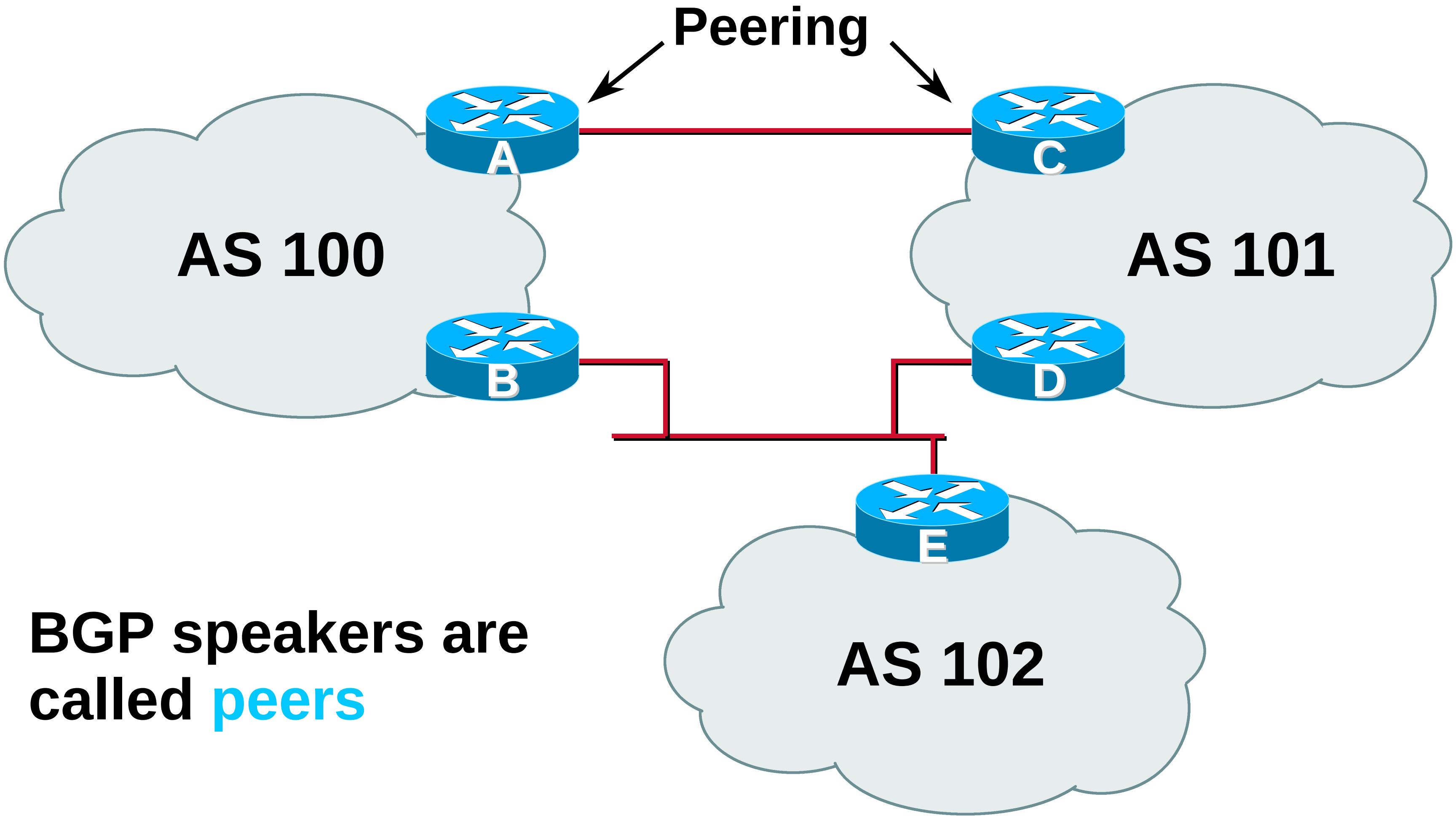
- Collection of networks with same routing policy
- Single routing protocol
- Usually under single ownership, trust and administrative control

Demarcation Zone (DMZ)



- **Shared network between ASes**

BGP Basics



BGP speakers are called peers

BGP General Operation

- Learns multiple paths via internal and external BGP speakers
- Picks the best path and installs in the forwarding table
- Policies applied by influencing the best path selection

Constructing the Forwarding Table

- BGP “in” process

  receives path information from peers

  results of BGP path selection placed in the BGP table

  “best path” flagged

- BGP “out” process

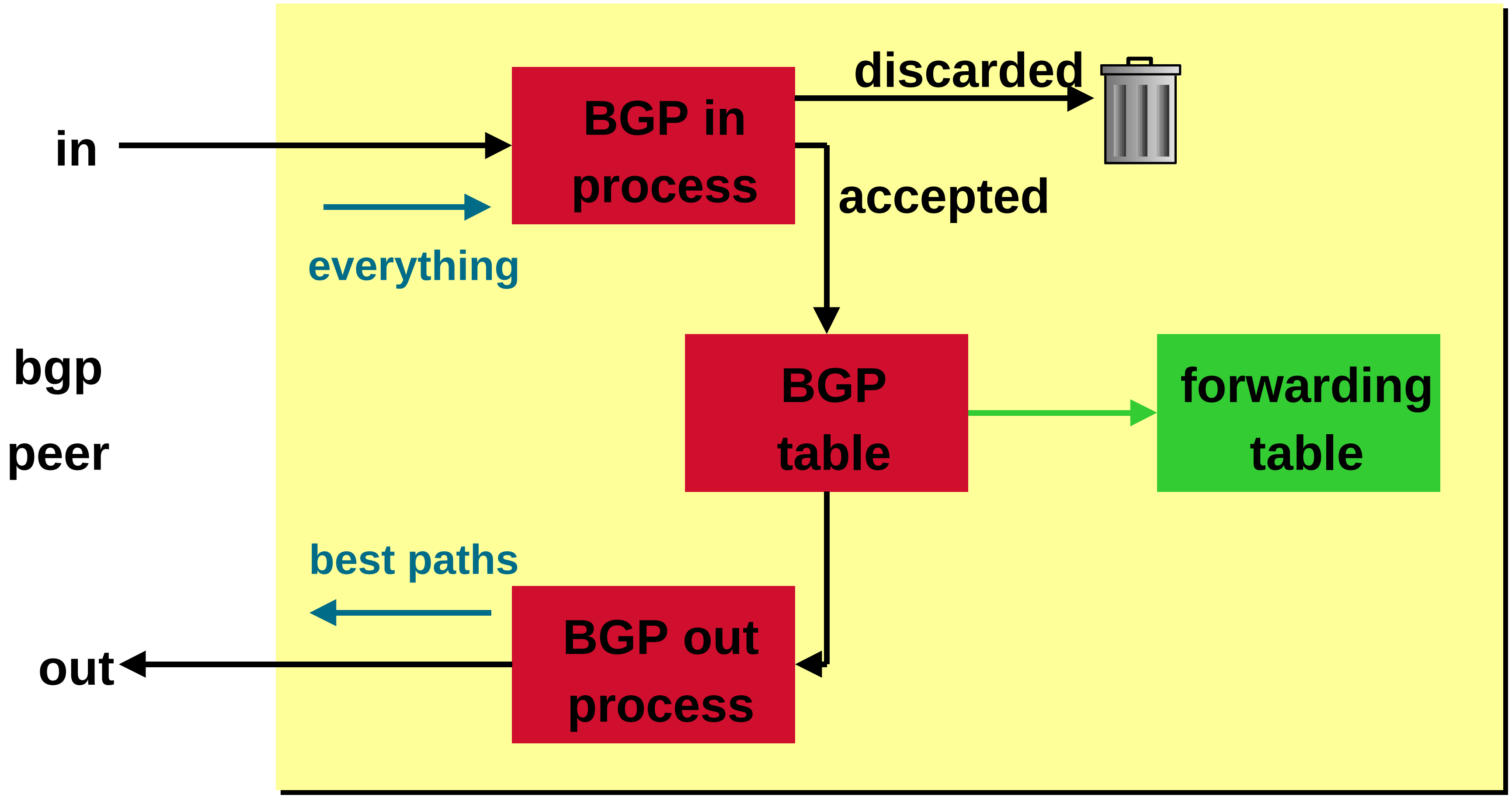
  announces “best path” information to peers

- Best paths installed in forwarding table if:

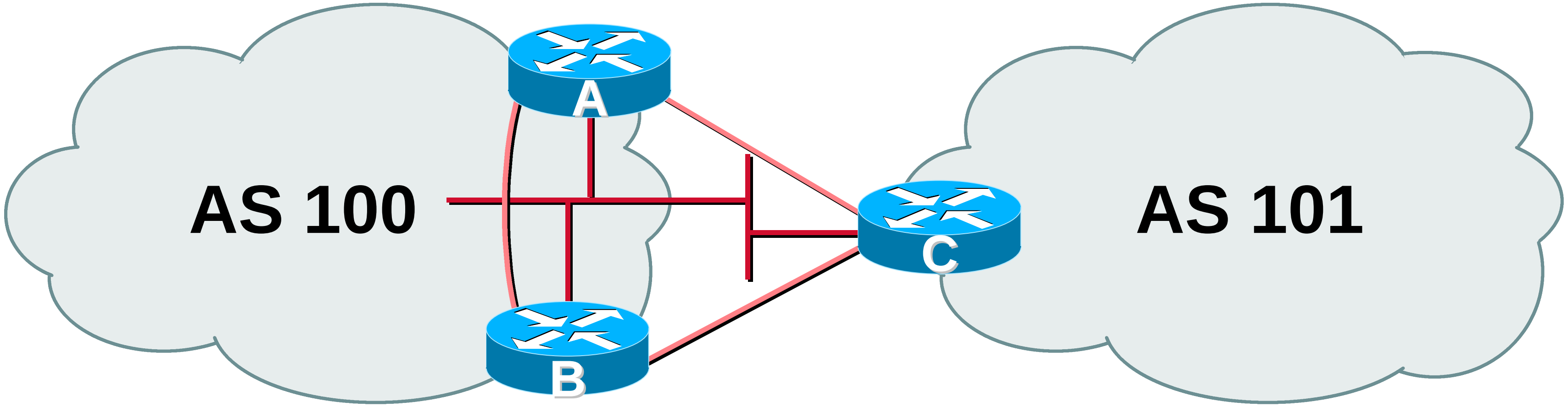
  prefix and prefix length are unique

  lowest “protocol distance”

Constructing the Forwarding Table



External BGP Peering (eBGP)



- Between BGP speakers in different AS
- Should be directly connected
- Do not run an IGP between eBGP peers

Configuring External BGP (Cisco IOS)

Router A in AS100

```
interface ethernet 5/0
ip address 222.222.10.2 255.255.255.240
router bgp 100
 network 220.220.8.0 mask 255.255.252.0
 neighbor 222.222.10.1 remote-as 101
 neighbor 222.222.10.1 prefix-list RouterC in
 neighbor 222.222.10.1 prefix-list RouterC out
```

Router C in AS101

```
interface ethernet 1/0/0
ip address 222.222.10.1 255.255.255.240
router bgp 101
 network 220.220.16.0 mask 255.255.240.0
 neighbor 222.222.10.2 remote-as 100
 neighbor 222.222.10.2 prefix-list RouterA in
 neighbor 222.222.10.2 prefix-list RouterA out
```

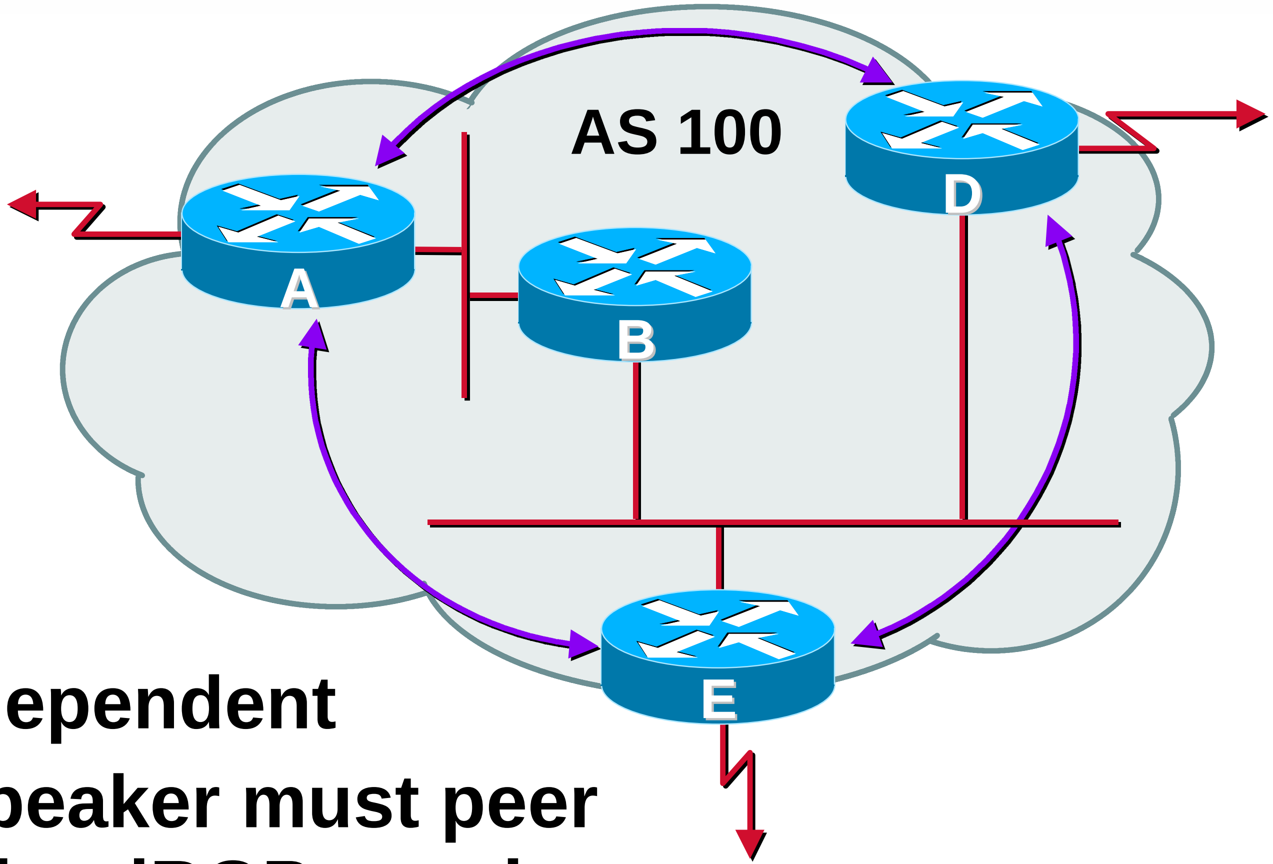
Internal BGP (iBGP)

- BGP peer within the same AS
- Not required to be directly connected
- iBGP speakers need to be fully meshed

  they originate connected networks

  they do not pass on prefixes learned from other iBGP speakers

Internal BGP Peering (iBGP)

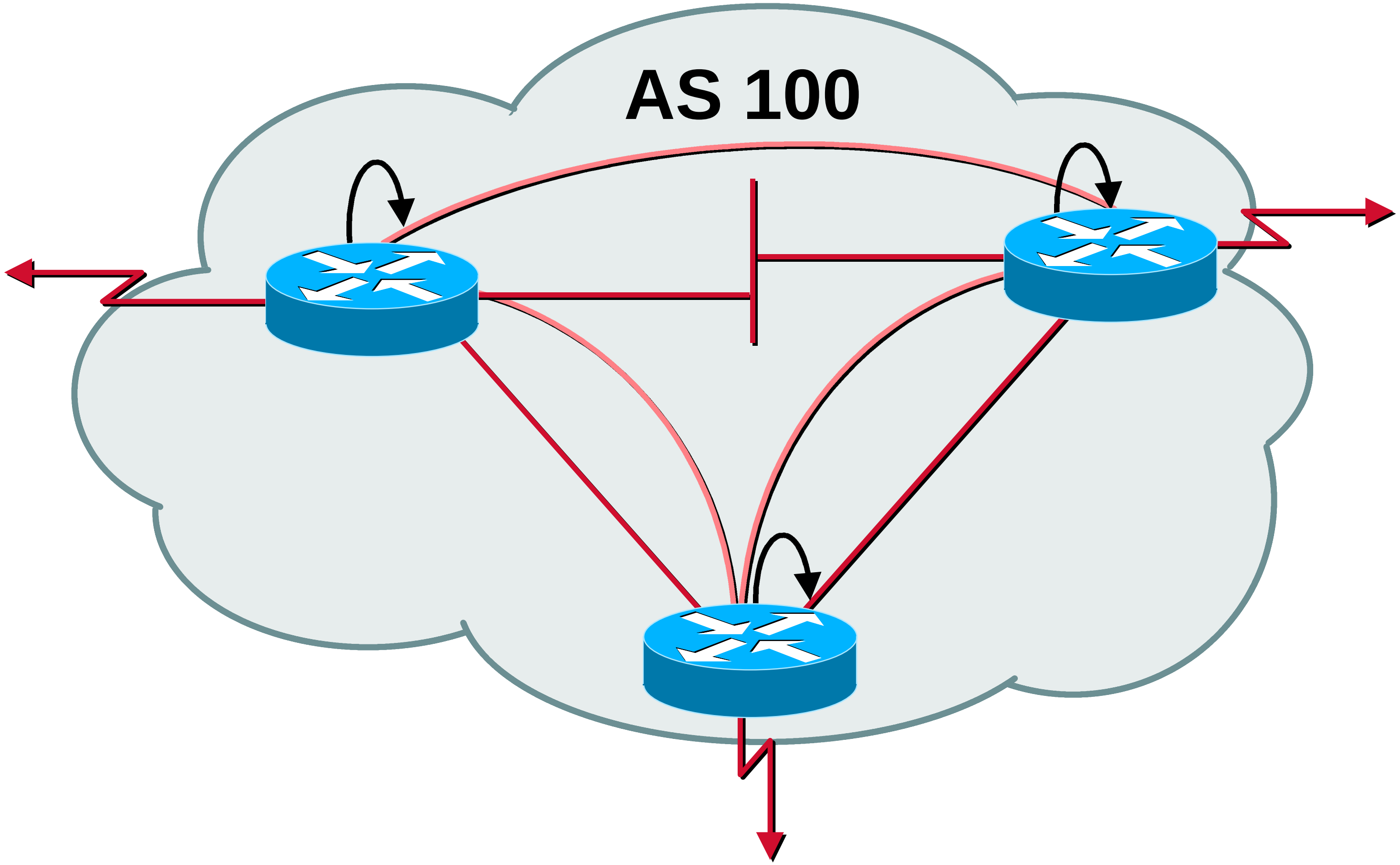


- Topology independent
- Each iBGP speaker must peer with every other iBGP speaker in the AS

Stable iBGP Peering

- Peer with loop-back address
- iBGP session is not dependent on state of a single interface
- iBGP session is not dependent on physical topology
- Loop-back interface does not go down - ever!

Peering to Loop-Back Address



Configuring Internal BGP (Cisco IOS)

Router A

```
interface loopback 0
ip address 215.10.7.1 255.255.255.255
router bgp 100
  network 220.220.1.0
  neighbor 215.10.7.2 remote-as 100
  neighbor 215.10.7.2 update-source loopback0
  neighbor 215.10.7.3 remote-as 100
  neighbor 215.10.7.3 update-source loopback0
```

Router B

```
interface loopback 0
ip address 215.10.7.2 255.255.255.255
router bgp 100
  network 220.220.5.0
  neighbor 215.10.7.1 remote-as 100
  neighbor 215.10.7.1 update-source loopback0
  neighbor 215.10.7.3 remote-as 100
  neighbor 215.10.7.3 update-source loopback0
```

Inserting prefixes into BGP - network command

- Configuration Example

```
router bgp 109
 network 198.10.4.0 mask 255.255.254.0
ip route 198.10.0.0 255.255.254.0 serial0
```

- A matching route must exist in the routing table before the network is announced
- Forces origin to be “IGP”

Configuration Aggregation - Network Command

- Configuration Example

```
router bgp 109
 network 198.10.0.0 mask 255.255.0.0
ip route 198.10.0.0 255.255.0.0 null0 250
```

- A matching route must exist in the routing table before the network is announced
- Easiest and best way of generating an aggregate

Configuring Aggregation - aggregate-address command

- Configuration Example

```
router bgp 109
 network 198.10.4.0 mask 255.255.252.0
 aggregate-address 198.10.0.0 255.255.0.0 [ summary-only ]
```

- Requires more specific prefix in routing table before aggregate is announced
- {summary-only} keyword

  optional keyword which ensures that only the summary is announced if a more specific prefix exists in the routing table

Auto Summarisation

- Cisc IOS automatically summarises subprefixes to the classful network.

  Example:

  ```
  61.10.8.0/22 --> 61.0.0.0/8
  ```

- Must be turned off for any Internet connected site using BGP.

  ```
  router bgp 109
   no auto-summary
  ```

Synchronisation

- **In Cisco IOS, BGP does not advertise a route before all routers in the AS have learned it via an IGP**
- **Disable synchronisation if:**

  AS doesn’t pass traffic from one AS to another, or

  All transit routers in AS run BGP, or

  iBGP is used across backbone

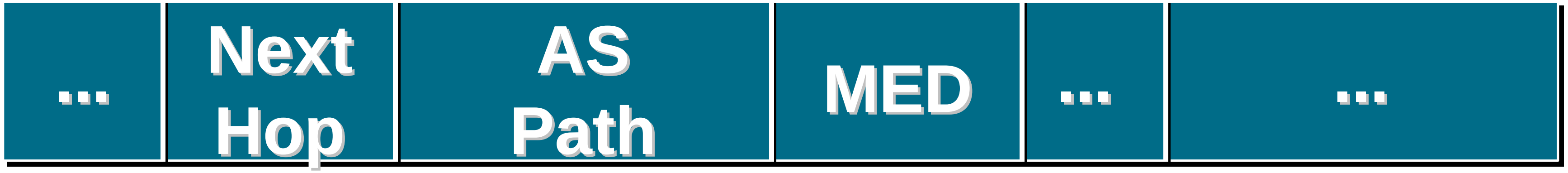
```
router bgp 109
 no synchronization
```

Summary

- BGP4 - distance vector protocol
- iBGP versus eBGP
- stable iBGP - peer with loopbacks
- announcing prefixes & aggregates
- no synchronization & no auto-summary

BGP Attributes

What Is an Attribute?

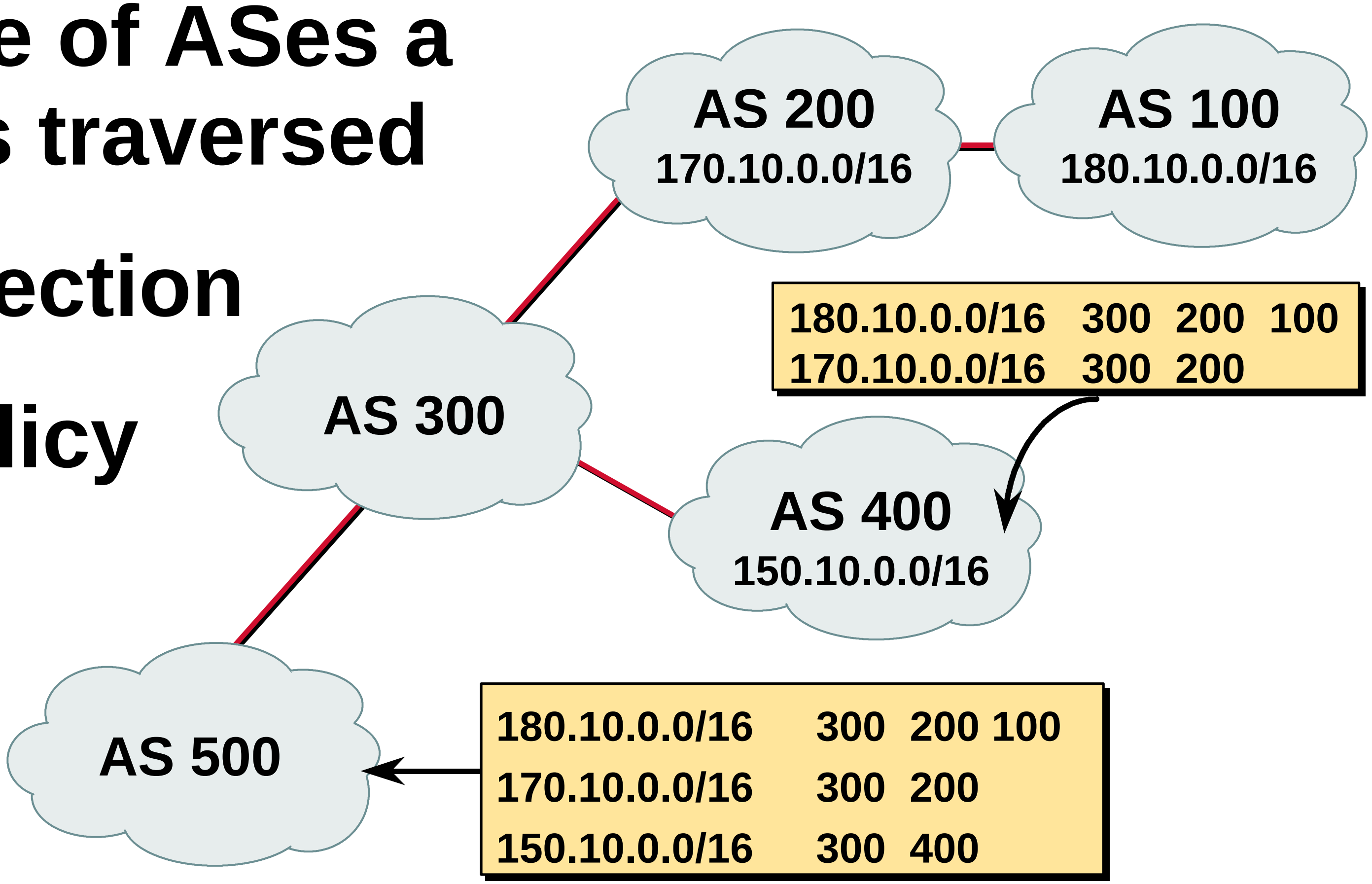


...	Next Hop	AS Path	MED	...	...

- Describes the characteristics of prefix
- Transitive or non-transitive
- Some are mandatory

AS-Path

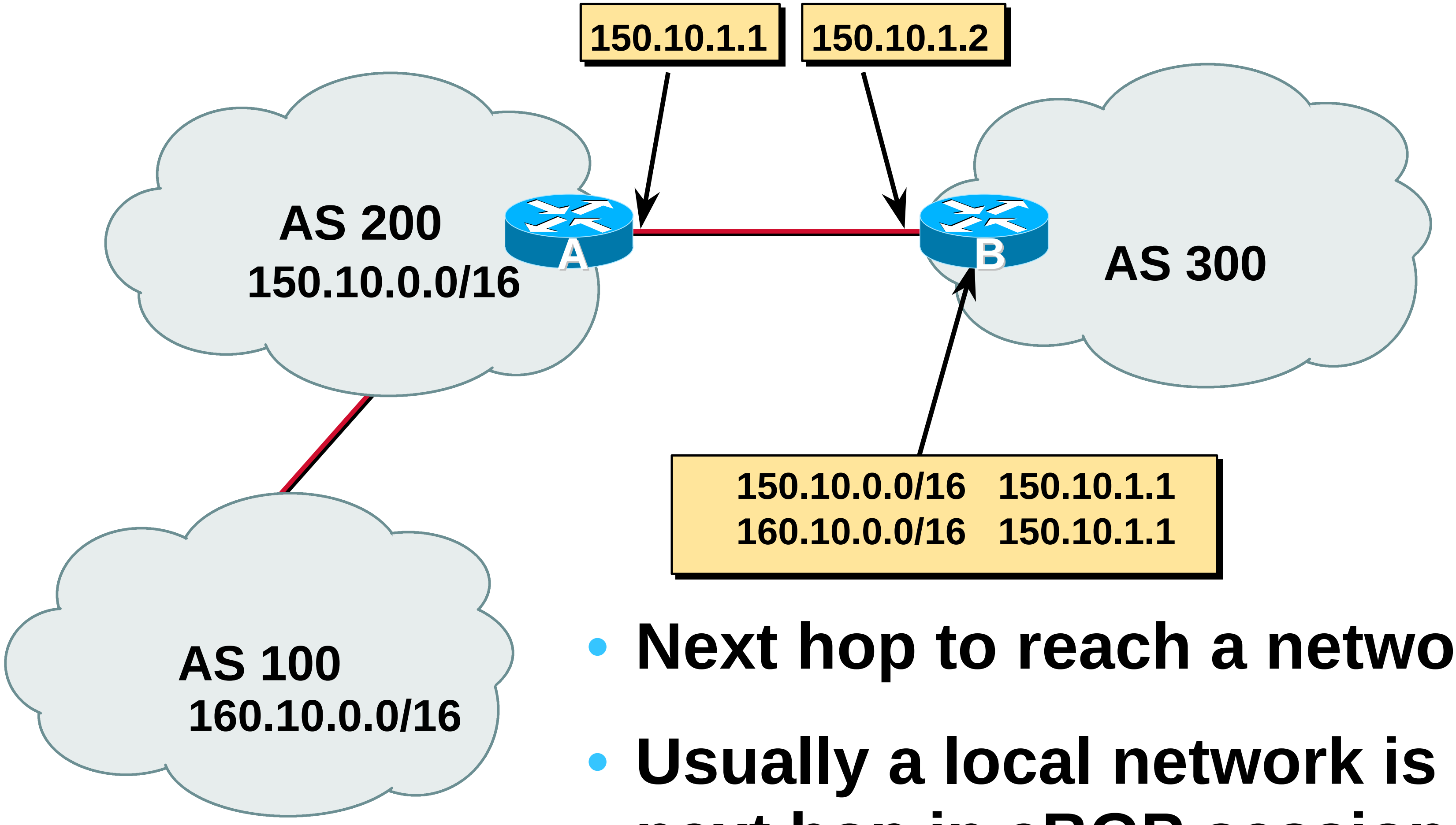
- Sequence of ASes a route has traversed
- Loop detection
- Apply policy



180.10.0.0/16	300 200 100
170.10.0.0/16	300 200

180.10.0.0/16	300 200 100
170.10.0.0/16	300 200
150.10.0.0/16	300 400

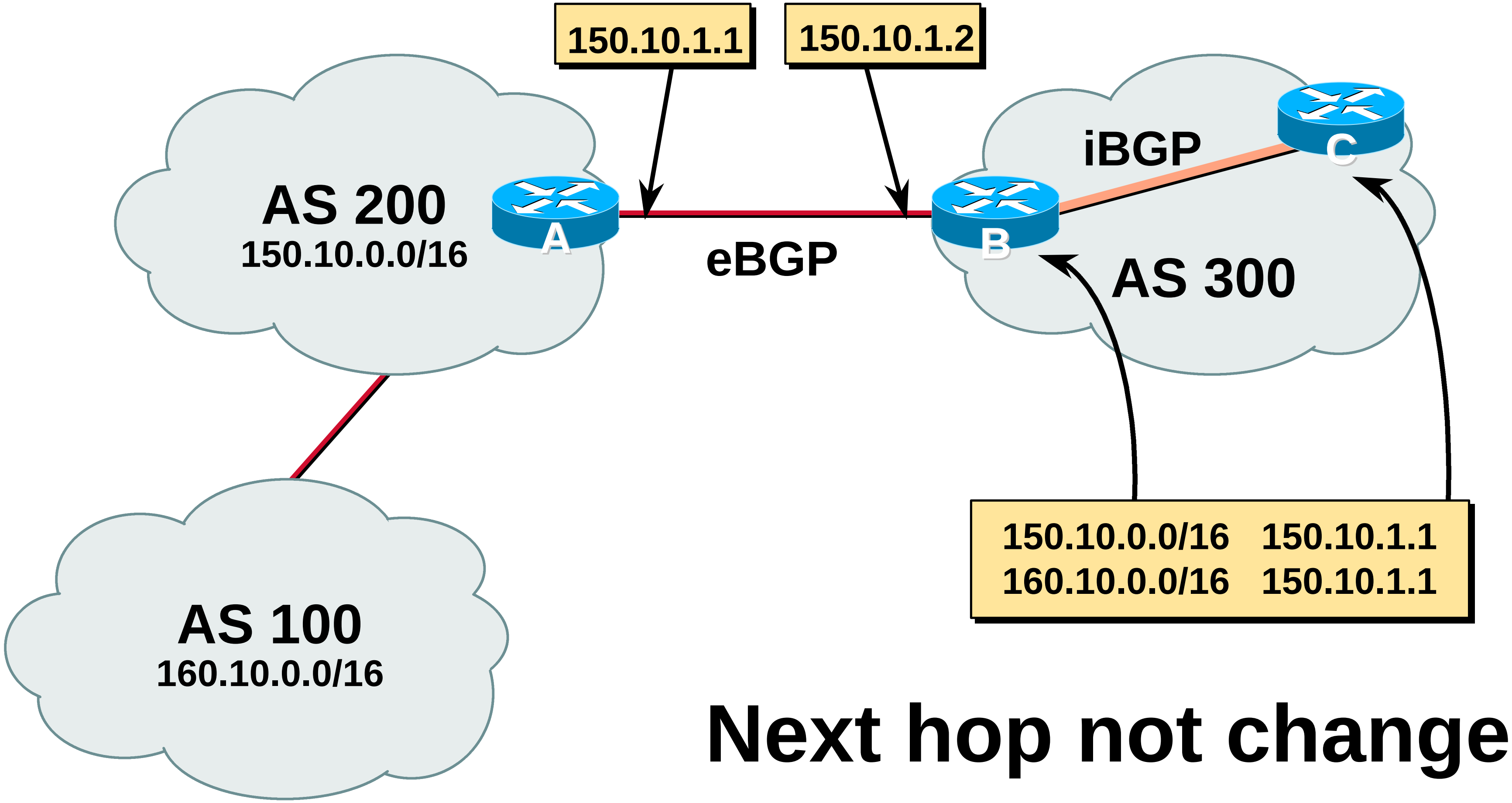
Next Hop



150.10.0.0/16 150.10.1.1
160.10.0.0/16 150.10.1.1

- Next hop to reach a network
- Usually a local network is the next hop in eBGP session

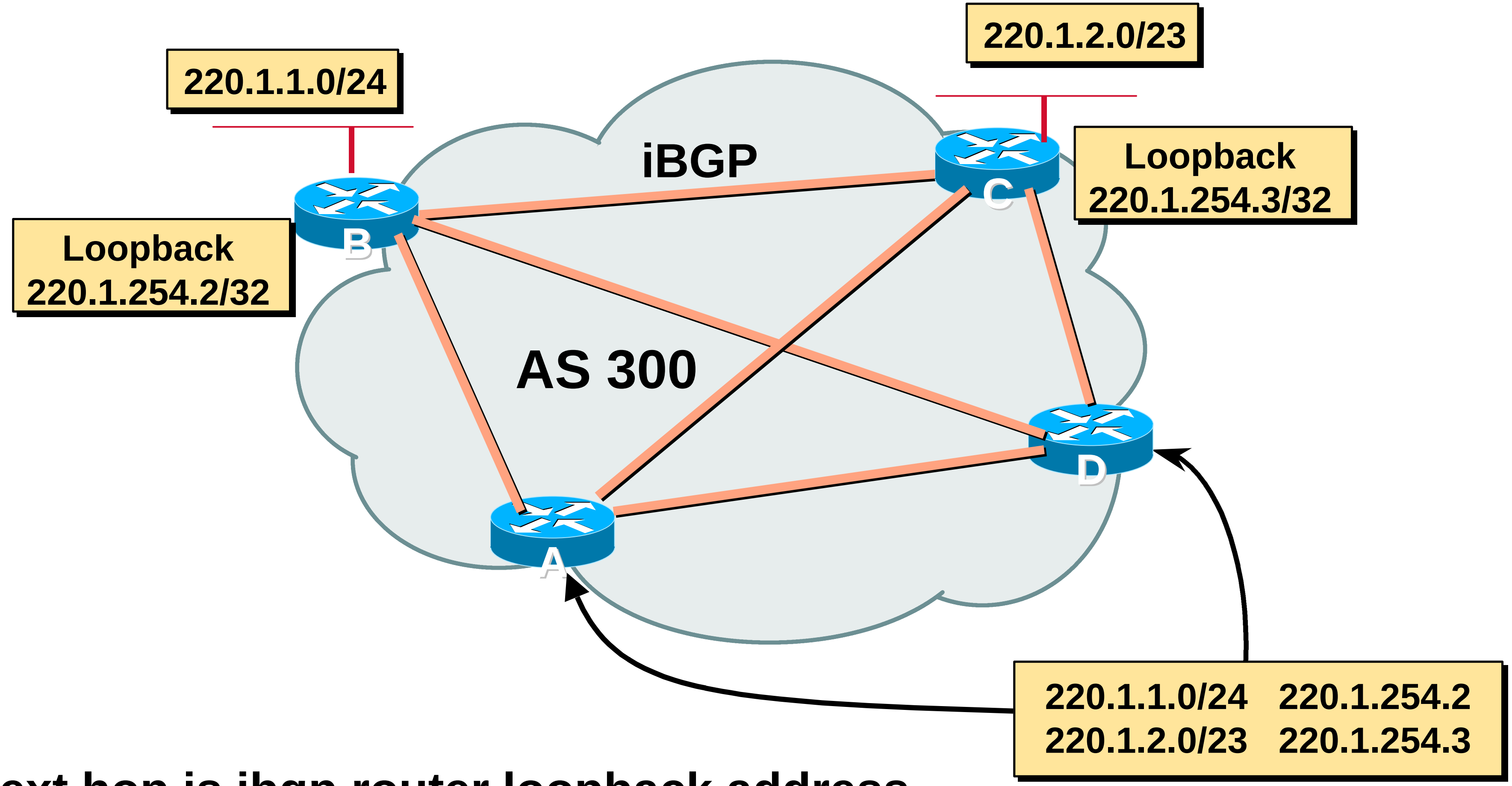
Next Hop



150.10.0.0/16	150.10.1.1
160.10.0.0/16	150.10.1.1

Next hop not changed

iBGP Next Hop



220.1.1.0/24	220.1.254.2
220.1.2.0/23	220.1.254.3

Next hop is ibgp router loopback address

Recursive route look-up

Next Hop (summary)

- IGP should carry route to next hops
- Recursive route look-up
- Unlinks BGP from actual physical topology
- Allows IGP to make intelligent forwarding decision

Origin

- Conveys the origin of the prefix
- Influence best path selection
- Three values - IGP, EGP, incomplete

  IGP - generated from BGP network statement

  EGP - generated from EGP

  incomplete - generated by “redistribute” action

Aggregator

- Useful for debugging purposes
- Conveys the IP address of the router/BGP speaker generating the aggregate route
- Doesn’t influence path selection

Local Preference

- Local to an AS - non-transitive

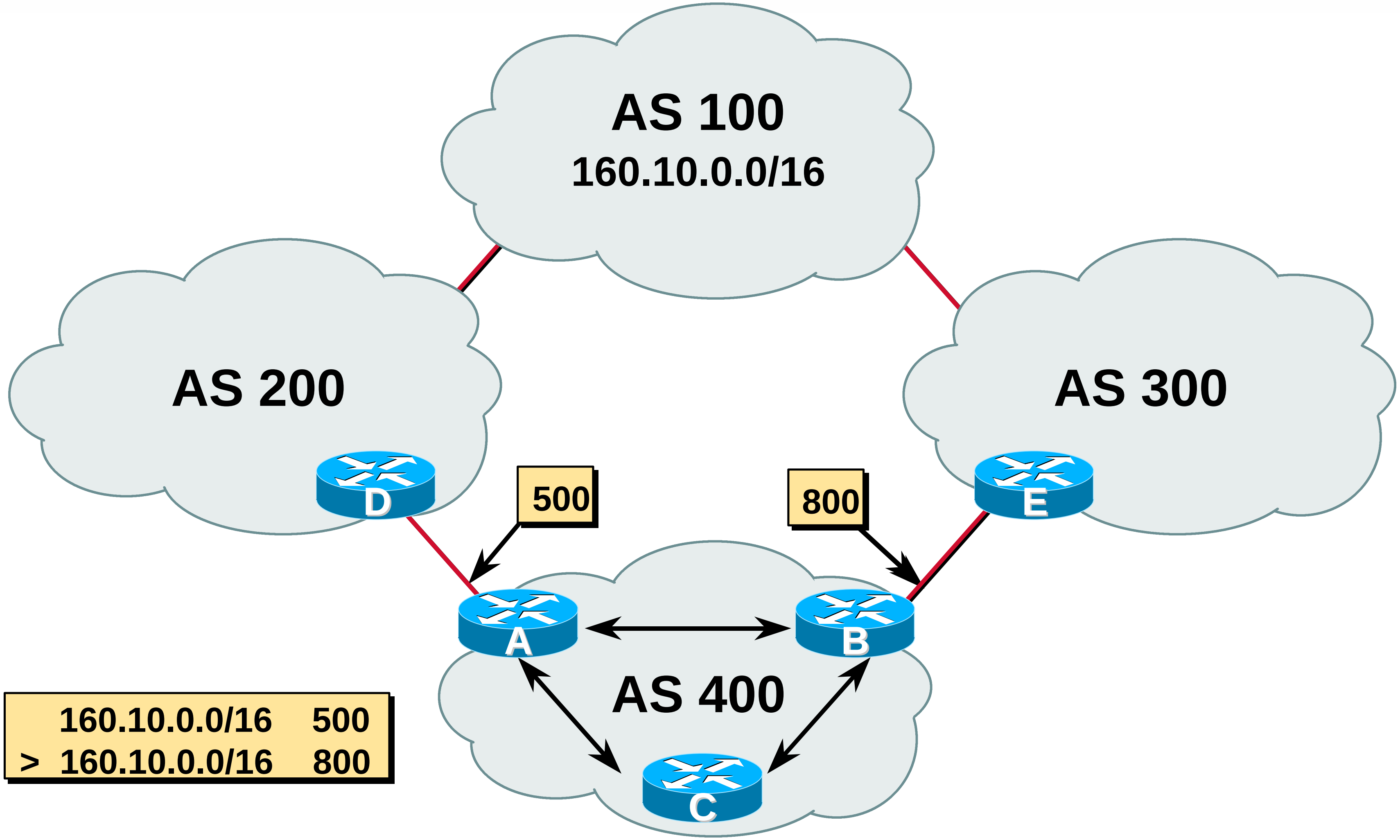
  local preference set to 100 when heard from neighbouring AS

- Used to influence BGP path selection

  determines best path for outbound traffic

- Path with highest local preference wins

Local Preference



```
   160.10.0.0/16    500
>  160.10.0.0/16    800
```

Local Preference

- Configuration of Router B:

```
router bgp 400
 neighbor 220.5.1.1 remote-as 300
 neighbor 220.5.1.1 route-map local-pref in
!
route-map local-pref permit 10
 match ip address prefix-list MATCH
 set local-preference 800
!
ip prefix-list MATCH permit 160.10.0.0/16
ip prefix-list MATCH deny 0.0.0.0/0 le 32
```

Multi-Exit Discriminator

- Inter-AS - non-transitive

  metric reset to 0 on announcement to next AS

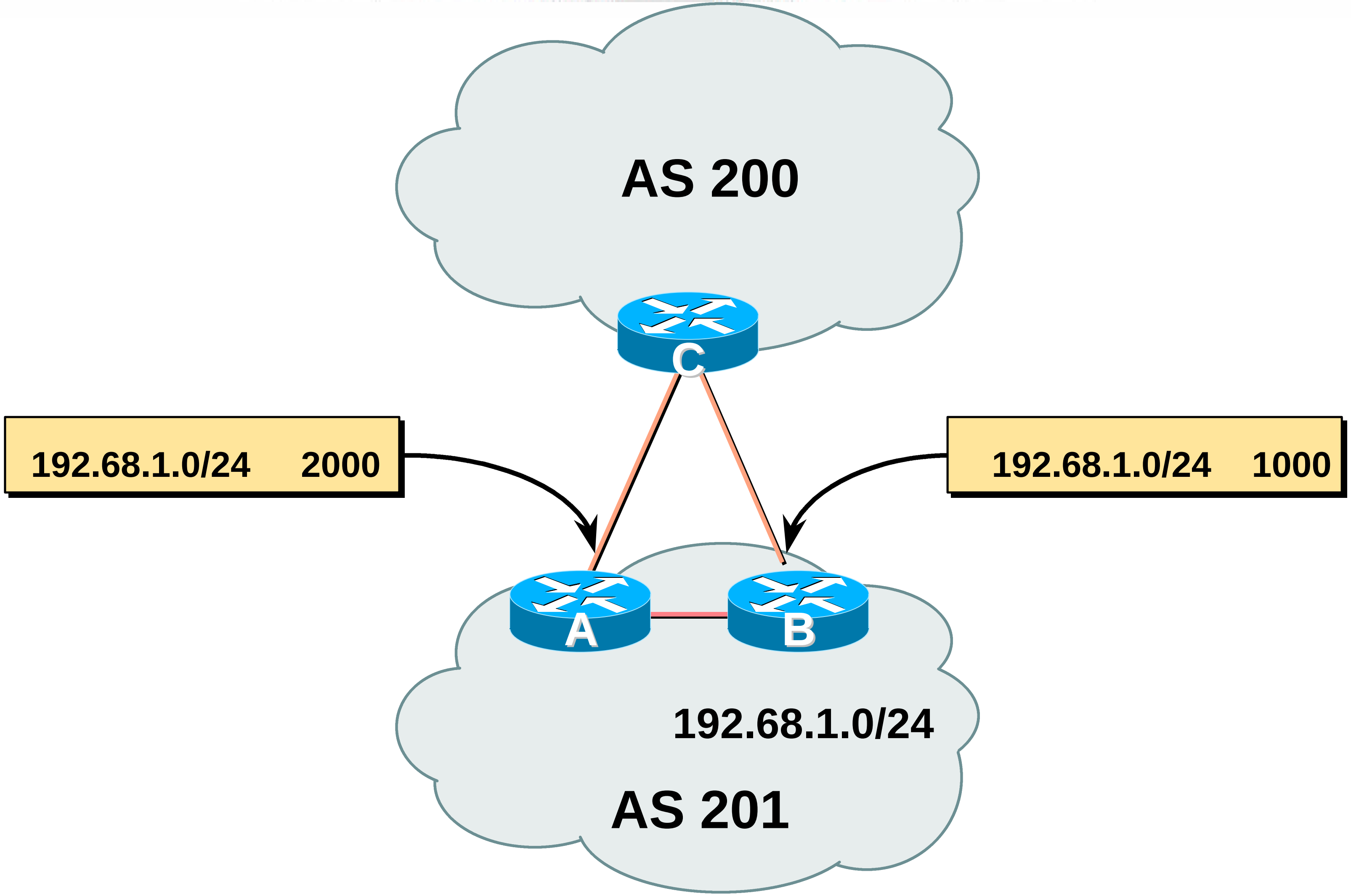
- Used to convey the relative preference of entry points

  determines best path for inbound traffic

- Comparable if paths are from same AS

- IGP metric can be conveyed as MED

Multi-Exit Discriminator (MED)



Multi-Exit Discriminator

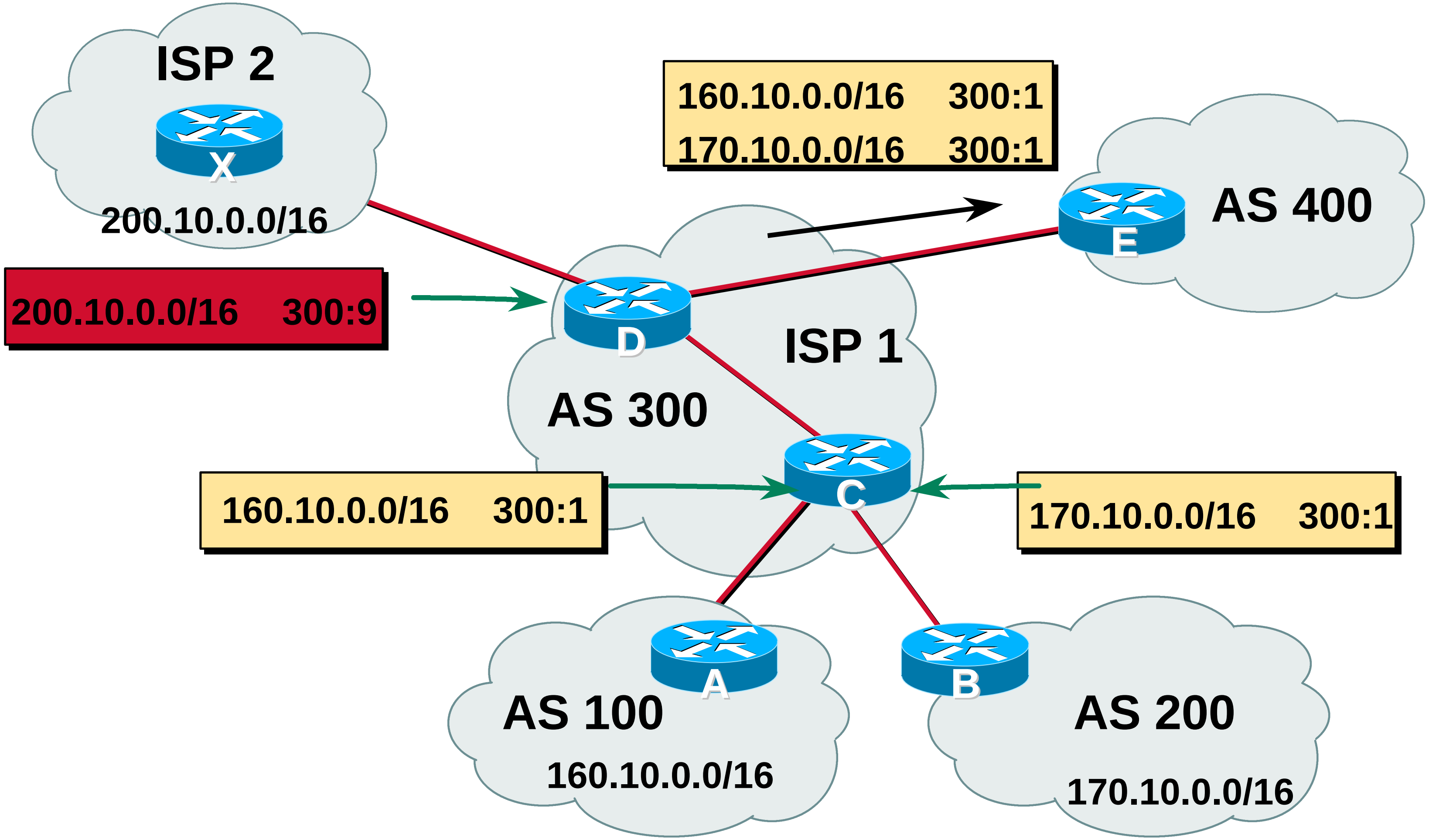
- Configuration of Router B:

```
router bgp 400
 neighbor 220.5.1.1 remote-as 200
 neighbor 220.5.1.1 route-map set-med out
!
route-map set-med permit 10
 match ip address prefix-list MATCH
 set metric 1000
!
ip prefix-list MATCH permit 192.68.1.0/24
ip prefix-list MATCH deny 0.0.0.0/0 le 32
```

Community

- BGP attribute
- Used to group destinations
- Represented as two 16bit integers
- Each destination could be member of multiple communities
- Community attribute carried across AS’s
- Useful in applying policies

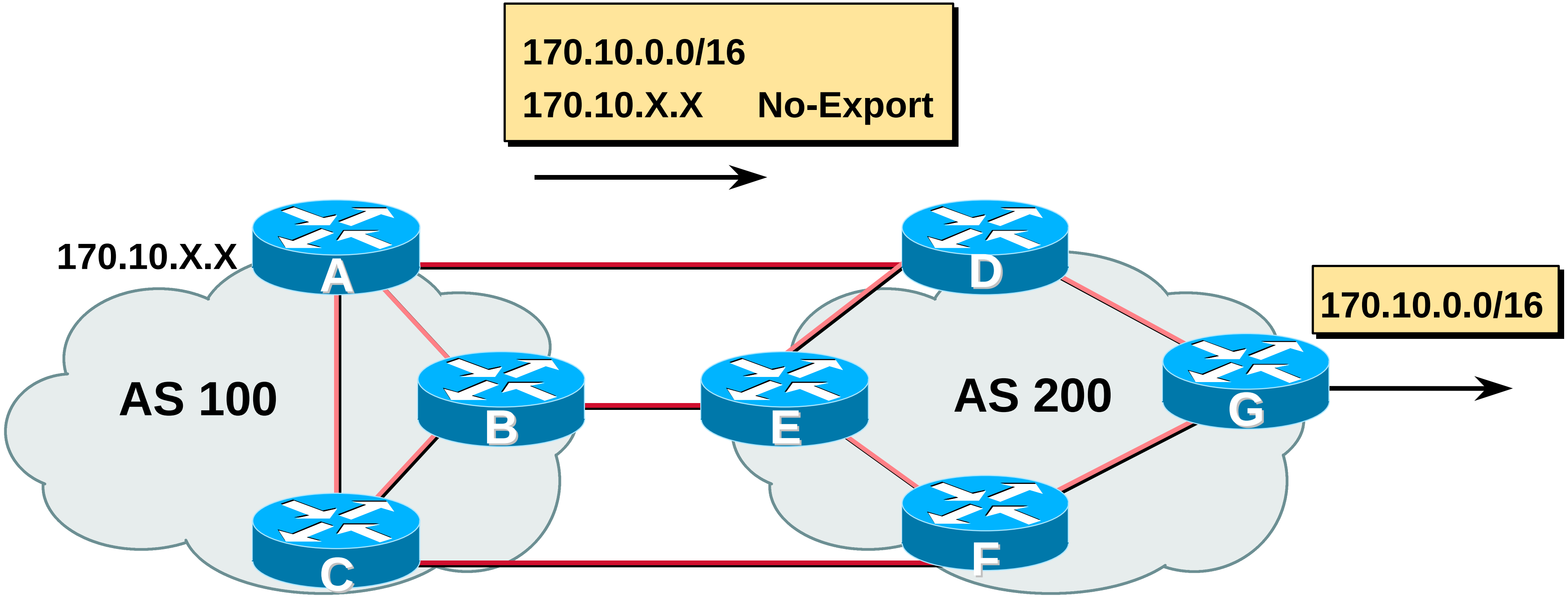
Community


ISP 2
X
200.10.0.0/16
200.10.0.0/16 300:9
160.10.0.0/16 300:1
170.10.0.0/16 300:1
AS 400
E
D
ISP 1
AS 300
C
160.10.0.0/16 300:1
170.10.0.0/16 300:1
A
B
AS 100
160.10.0.0/16
AS 200
170.10.0.0/16

Well-Known Communities

- internet = all routes are members of this community
- no-export = do not advertise to eBGP peers
- no-advertise = do not advertise to any peer
- local-AS = do not advertise outside local AS (used with confederations)

No-Export Community



BGP Path Selection Algorithm

Why is this the best path?

BGP Path Selection Algorithm

- Do not consider iBGP path if not synchronised
- Do not consider path if no route to next hop
- Highest weight (local to router)
- Highest local preference (global within AS)
- Shortest AS path

BGP Path Selection Algorithm (continued)

- Lowest origin code

  IGP < EGP < incomplete

- Multi-Exit Discriminator

  Considered only if paths are from same AS

- Prefer eBGP path over iBGP path
- Path with shortest next-hop metric wins
- Lowest router-id

Applying Policy with BGP

The BGP Toolkit

Applying Policy with BGP

- Policy-based on AS path, community or the prefix
- Rejecting/accepting selected routes
- Set attributes to influence path selection
- Tools:

  Prefix-list (filters prefixes)

  Filter-list (filters AS paths)

  Route-maps and communities

Policy Control - Prefix List

- Per neighbour prefix filter

  incremental configuration

- High performance access-list
- Inbound or Outbound
- Based upon network numbers (using familiar IPv4 address/mask format)

Prefix Lists - Examples

- Deny default route

```
ip prefix-list EG deny 0.0.0.0/0
```

- Permit the prefix 35.0.0.0/8

```
ip prefix-list EG permit 35.0.0.0/8
```

- Deny the prefix 172.16.0.0/12

```
ip prefix-list EG deny 172.16.0.0/12
```

- In 192/8 allow up to /24

```
ip prefix-list EG permit 192.0.0.0/8 le 24
```

This allows all prefix sizes in the 192.0.0.0/8 address block, apart from /25, /26, /27, /28, /29, /30, /31 and /32.

Prefix Lists - Examples

- In 192/8 deny /25 and above

```
ip prefix-list EG deny 192.0.0.0/8 ge 25
```

This denies all prefix sizes /25, /26, /27, /28, /29, /30, /31 and /32 in the address block 192.0.0.0/8.

It has the same effect as the previous example

- In 192/8 permit prefixes between /12 and /20

```
ip prefix-list EG permit 193.0.0.0/8 ge 12 le 20
```

This denies all prefix sizes /8, /9, /10, /11, /21, /22, … and higher in the address block 193.0.0.0/8.

- Permit all prefixes

```
ip prefix-list EG permit 0.0.0.0/0 le 32
```

Policy Control - Prefix List

- Example Configuration

```
router bgp 200
 network 215.7.0.0
 neighbor 220.200.1.1 remote-as 210
 neighbor 220.200.1.1 prefix-list PEER-IN in
 neighbor 220.200.1.1 prefix-list PEER-OUT out
!
ip prefix-list PEER-IN deny 218.10.0.0/16
ip prefix-list PEER-IN permit 0.0.0.0/0 le 32
ip prefix-list PEER-OUT permit 215.7.0.0/16
ip prefix-list PEER-OUT deny 0.0.0.0/0 le 32
```

Policy Control - Filter List

- Filter routes based on AS path
- Inbound or Outbound
- Example Configuration:

```
router bgp 100
 network 215.7.0.0
 neighbor 220.200.1.1 filter-list 5 out
 neighbor 220.200.1.1 filter-list 6 in
!
ip as-path access-list 5 permit ^200$
ip as-path access-list 6 permit ^150$
```

Policy Control - Regular Expressions

- Like Unix regular expressions

.	Match one character
*	Match any number of preceding expression
+	Match at least one of preceding expression
^	Beginning of line
$	End of line
_	Beginning, end, white-space, brace
|	Or
()	brackets to contain expression

Policy Control - Regular Expressions

- Simple Examples

.*	Match anything
.+	Match at least one character
^$	Match routes local to this AS
_1800$	Originated by 1800
^1800_	Received from 1800
_1800_	Via 1800
_790_1800_	Passing through 1800 then 790
_(1800_)+	Match at least one of 1800 in sequence
_\(65350\)_	Via 65350 (confederation AS)

Policy Control - Regular Expressions

- Not so simple Examples

^[0-9]+$	Match AS_PATH length of one
^[0-9]+_[0-9]+$	Match AS_PATH length of two
^[0-9]*_[0-9]+$	Match AS_PATH length of one or two
^[0-9]*_[0-9]*$	Match AS_PATH length of one or two
^[0-9]+_[0-9]+_[0-9]+$	Match AS_PATH length of three
_(701|1800)_	Match anything which has gone through AS701 or AS1800
_1849(_.+_)12163$	Match anything of origin AS12163 and passed through AS1849

Policy Control - Route Maps

- Example Configuration - route map and prefix-lists

```
ip prefix-list HIGH-PREF permit 10.0.0.0/8
ip prefix-list HIGH-PREF deny 0.0.0.0/0 le 32
ip prefix-list LOW-PREF permit 20.0.0.0/8
ip prefix-list LOW-PREF deny 0.0.0.0/0 le 32
!
route-map infilter permit 10
 match ip address prefix-list HIGH-PREF
 set local-preference 120
!
route-map infilter permit 20
 match ip address prefix-list LOW-PREF
 set local-preference 80
!
router bgp 100
 neighbor 1.1.1.1 route-map infilter in
```

Policy Control - Route Maps

- Example Configuration - route map and filter lists

```
router bgp 100
 neighbor 220.200.1.2 remote-as 200
 neighbor 220.200.1.2 route-map filter-on-as-path in
!
route-map filter-on-as-path permit 10
 match as-path 1
 set local-preference 80
!
route-map filter-on-as-path permit 20
 match as-path 2
 set local-preference 200
!
ip as-path access-list 1 permit _150$
ip as-path access-list 2 permit _210_
```

Policy Control - Route Maps

- Example configuration of AS-PATH prepend

```
router bgp 300
 network 215.7.0.0
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 route-map SETPATH out
!
route-map SETPATH permit 10
 set as-path prepend 300 300
```

- Standard practice implements two occurrences of the ASN when prepending

Policy Control - Matching Communities

- Example Configuration

```
router bgp 100
 neighbor 220.200.1.2 remote-as 200
 neighbor 220.200.1.2 route-map filter-on-community in
!
route-map filter-on-community permit 10
 match community 1
 set local-preference 50
!
route-map filter-on-community permit 20
 match community 2 exact-match
 set local-preference 200
!
ip community-list 1 permit 150:3 200:5
ip community-list 2 permit 88:6
```

Policy Control - Setting Communities

- Example Configuration

```
router bgp 100
 network 215.7.0.0
 neighbor 220.200.1.1 remote-as 200
 neighbor 220.200.1.1 send-community
 neighbor 220.200.1.1 route-map set-community out
!
route-map set-community permit 10
 match ip address prefix-list NO-ANNOUNCE
 set community no-export
!
route-map set-community permit 20
 match ip address prefix-list EVERYTHING
!
ip prefix-list NO-ANNOUNCE permit 172.168.0.0/16 ge 17
ip prefix-list EVERYTHING permit 0.0.0.0/0 le 32
```

BGP Summary

- Attributes
- Path Selection Process
- Policy Control Tools

- Any questions?