CS551
External v.s. Internal BGP

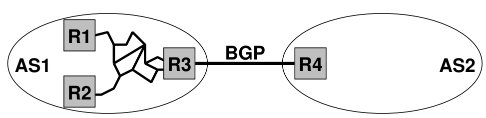
Bill Cheng

http://merlot.usc.edu/cs551-f12

EGP vs. IGP

- Exterior vs. Interior
- World vs. me
- Little control vs. complete ***administrative control***
- ***BGP*** (and GGP, Hello, EGP) vs. (RIP, ***OSPF***, IS-IS, IGRP, EIGRP)

Learning Routes

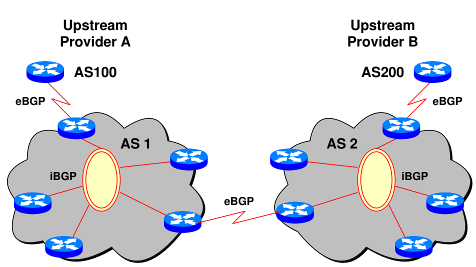


- BGP can be used by R3 and R4 to learn routes.
- How do R1 and R2 learn routes? (How does R3 pass on the routes that is has learned to R1 and R2?)
- Option 1: Inject routes in IGP (such as OSPF)
  - only works for small routing tables
- Option 2: Use I-BGP

Why BGP as an IGP?

- I-BGP has mechanisms to forward BGP policy directives across an AS
- Often use I-BGP with *some* other IGP (such as OSPF) that does internal routing

I-BGP



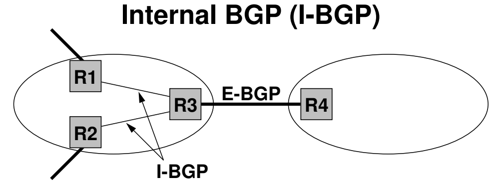
E-BGP vs. I-BGP

- E-BGP connects AS's (*external* GP)
- I-BGP is *intra*-AS (*internal* GP)
- Differences in operation
  - direct vs. indirect connections
  - different failure modes
  - special attributes for internal use

Internal BGP (I-BGP)

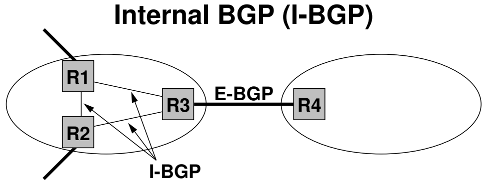
- Same message types, attribute types, and state machine as E-BGP
- Different rules about re-advertising prefixes:
  - prefix learned from E-BGP can be advertised to I-BGP neighbor and vice-versa, but
  - prefix learned from one I-BGP neighbor *cannot* be advertised to another I-BGP neighbor
  - reason: no AS-PATH within the same AS and thus danger of looping

Internal BGP (I-BGP)



- R3 can tell R1 and R2 prefixes from R4
- R3 can tell R4 prefixes from R1 and R2
- R3 cannot tell R2 prefixes from R1

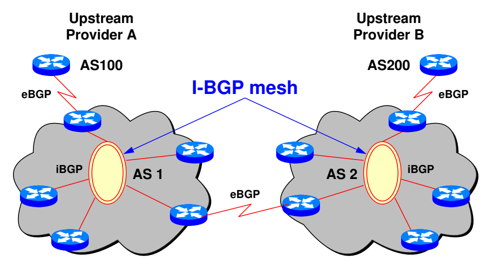
Internal BGP (I-BGP)



- R3 can tell R1 and R2 prefixes from R4
- R3 can tell R4 prefixes from R1 and R2
- R3 cannot tell R2 prefixes from R1

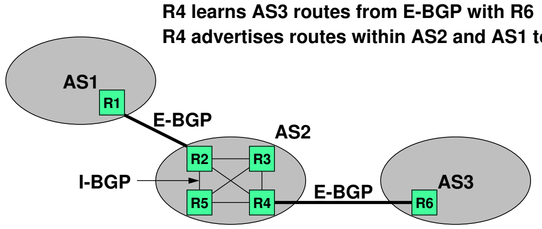
R2 can only find these prefixes through a direct connection to R1
Result: I-BGP routers must be *fully connected* (via TCP)!
- contrast with E-BGP sessions that map to physical links

I-BGP



BGP Example

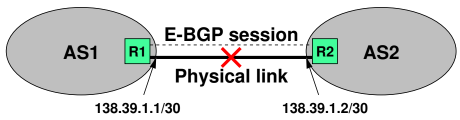
R1 advertises routes within AS1 to R2
R2 advertises routes within AS2 and AS3 to R1
R2 learns AS3 routes from I-BGP with R4
R4 learns AS3 routes from E-BGP with R6
R4 advertises routes within AS2 and AS1 to R6



Link Failures

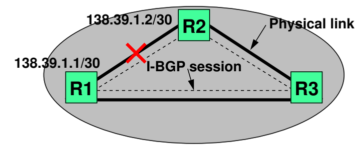
- Two types of link failures:
  - failure on an E-BGP link
  - failure on an I-BGP Link
- These failures are treated completely different in BGP
- Why?

Failure on an E-BGP Link



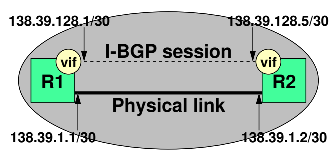
- Note that 138.39.1.1 and 138.39.1.2 are on the *same* network
- If the link R1-R2 goes down, then the TCP connection breaks and so does the E-BGP connection; BGP routes are removed
- This is the desired behavior

Failure on an I-BGP Link



- If physical link R1-R2 goes down, the 138.39.1.0/30 network becomes unreachable, connection between R1 and R2 is lost
- R1 and R2 should, in theory, still be able to exchange traffic, i.e., the indirect path through R3 should be used
  - given the above configuration, it would not work!
  - thus, E-BGP and I-BGP must use *different conventions* with respect to TCP endpoints
- Note: I-BGP often does *not* go over a *physical link*

Virtual Interfaces (VIFs, a.k.a. Loop-back Interfaces)



- Note that 138.39.128.1 and 138.39.128.5 are on *different* networks here!
- A VIF is not associated with a physical link or hardware interface
- How do routers learn of VIF addresses?
  - use IGP

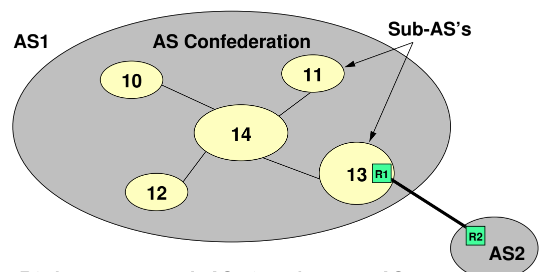
Scaling the I-BGP Mesh

- Two methods:
  - *BGP confederations*
    - scale by adding hierarchy to AS (sub-AS)
  - *Route reflectors*
    - scale by adding hierarchical IBGP route forwarding

AS Confederation

- Subdivide a single AS into multiple, internal sub-AS's to reduce I-BGP mesh size
  - simple hierarchy
  - but only one level
- Still advertises a single AS to external peers
  - internally use *sub-AS's*

An AS Confederation

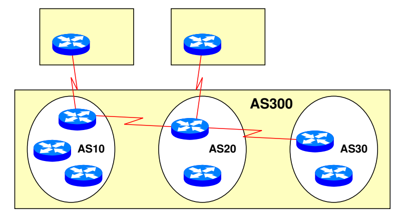


- R2 does not see sub-AS 10-14, but sees AS1

Confederations

- BGP sessions between sub-AS's are like regular E-BGP but with some changes:
  - *local-pref* attribute remains meaningful within confederation (E-BGP ignores it)
  - *next-hop* attribute traverses sub-AS boundaries (assumes single IGP running - everyone has same route to *next-hop*)
  - AS-PATH now includes AS-CONFED-SET and AS-CONFED-SEQUENCE to avoid loops

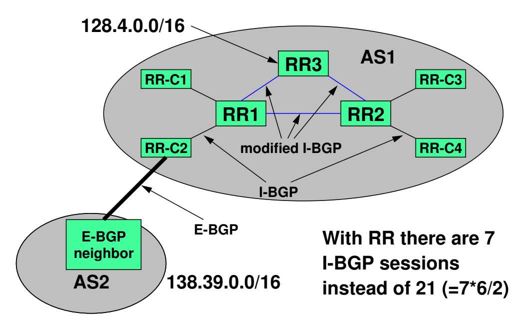
BGP Confederation



Route Reflectors

- *Route Reflector (RR):* router whose BGP implementation allows re-advertisement of routes between I-BGP neighbors
  - RR runs modified I-BGP
- *Route Reflector Client (RRC):* router that depends on RR to re-advertise its routes to entire AS. It also depends on RR to learn routes from the rest of the network
  - RRC runs normal I-BGP

RR Example



With RR there are 7 I-BGP sessions instead of 21 (=7*6/2)

Rules for Route Reflectors

- Reflectors advertise routes learned from clients into the I-BGP mesh
  - RR1 advertises 138.39.0.0/16 learned from RRC2 into I-BGP
- Reflectors do not re-advertise routes between non-clients
  - RR1 will not re-advertise 128.4.0.0/16 learned from RR3 to RR2