Border Gateway Protocol - An Introduction

Network Routing class

Jim Binkley

outline

- overview/theory
  - history/topologies/2 kinds of BGP/basic idea as DV protocol/important ideas
- protocol
- database, IBGP issues, policy tricks, Cisco config minimal intro
- problems including flapping/security

bibliography

- rfc 1771, “A Border Gateway Protocol 4”, Yakov Rekhter, and Tony Li, 1995
- rfcs 1772-1774 related, other BGP rfcs exist
- Books:
  - Moy’s OSPF has a very good overview chapter
  - “**Internet Routing Architectures**”, Halabi, Cisco Press, title should be “Fun with BGP”
    - » entire book about BGP basically
  - IP Routing Protocols - U. Black, has a chapter
  - Huitema of course and Perlman, 2nd edition for a little contrarian thinking

more RFCs (which can be state of the art in this case)

- 1657 - BGP MIB (SMIv2) - 1994
- 2385 - Protection of BGP Sessions Via the TCP/MD5 Signature Option, Heffernan, 1998
- 2595 - Use of BGP-4 Multi-protocol Extensions for IPv6 Inter-Domain Routing
- 2858 - Multi-protocol Extensions for BGP-4, Bates, et. al, June 2000
- route reflection/confederation/communities/ flapping as well + probably something else
- RFC 3221 - recent experience (growth of table/s)

history

- GGP - gateway to gateway (you knew that?) - DV IGP used in ARPANET
  - had 2 out of 4 echo to learn if peer existed
  - explicit ACK of update
- EGP - an EGP!, NSFNET time period
  - net had to be strictly hierarchical, no loops
  - metric-less since there could not be 2 paths
- IDRP - “i drip, you drip, we all drip”, OSI BGP equivalent, had influence on BGP
- and one more ... (next slide)

history, cont

- IDPR - Martha Steenstrup, RFC 1479
  - LS EGP, competition for awhile with BGP
    - » again with IPv6, deja vu all over again
  - not hop by hop, but source route
  - initial router determines path to other side
  - can thus enforce arbitrary policies
    - » go to X, then Y, then turn left, you are at Grandma's
  - call this "flow setup" :-> ?
  - considering MPLS, there may be some irony here

BGP history

- some EGP problems drove BGP design
  - needed to tolerate multiple paths and choose
  - early policy experiments aided evolution
- BGP-4 as BGP-3 did not speak CIDR
- multi-protocol BGP recently introduced
  - can deliver IPv6 info
  - can deliver multicast group info and perform RPF function for “uber” PIM/SM

basic idea “use TCP”

- we use TCP between BGP peers, call the peers speakers (2 peers), port 179
  - BGP is vc oriented, pt./pt. pair-wise, unicast
- TCP handles many of the error problems, hence BGP can be simpler
  - and stream data
  - don’t need our own reliable protocol, etc
  - can be multi-hop if that makes sense

two kinds of BGP

- **External BGP**, **EBGP** - exterior BGP connection between two separate AS
  - typically have direct link connection
  - over a T1, T3, OC-xyzzy, Ethernet segment
  - since two AS/two admins collide, this may take
    - » lawyers, and contracts, and money
- **Internal BGP**, **IBGP** - internal to AS
  - may be multi-hop
  - may need to send BGP updates across the AS

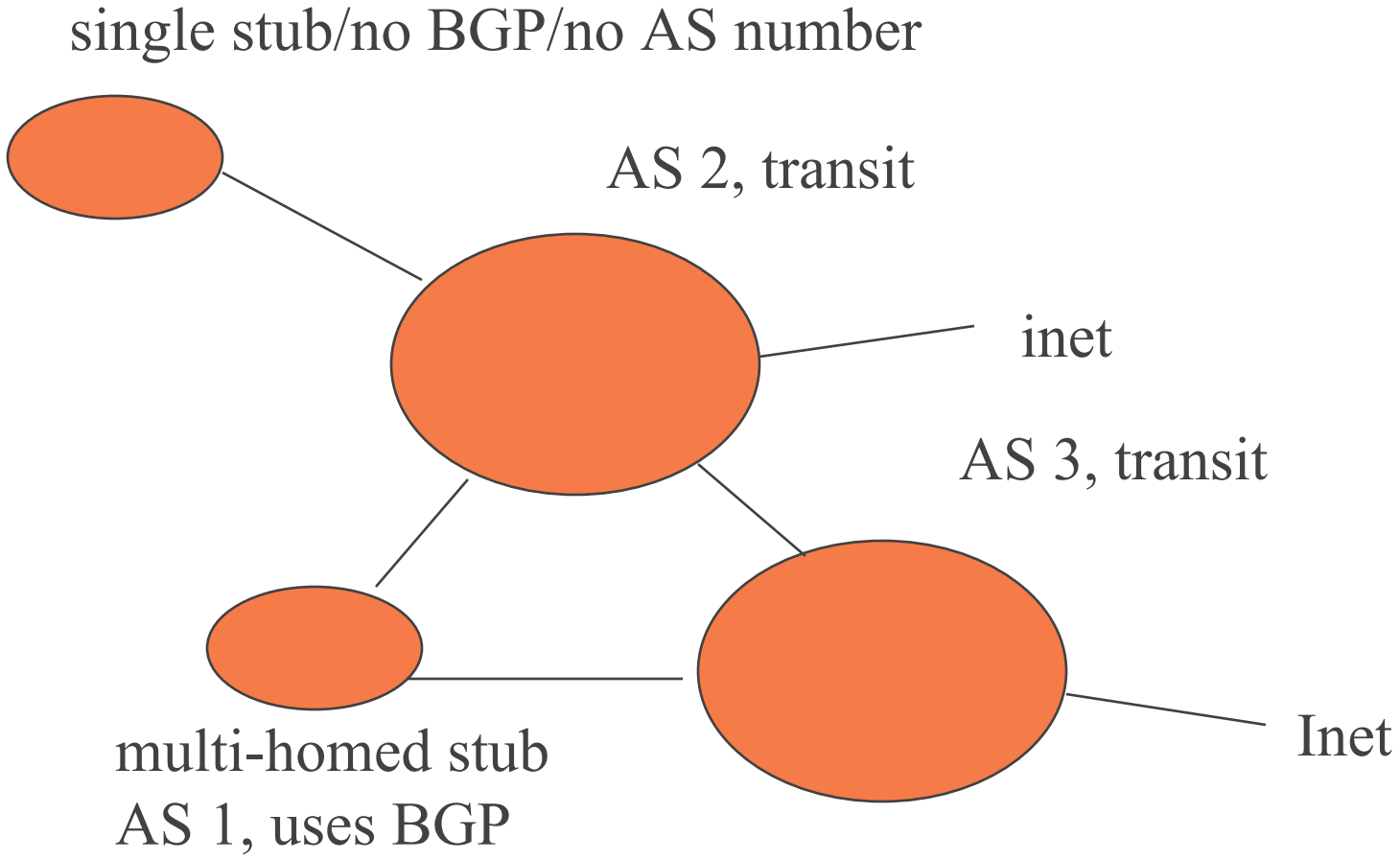
how do we get reachability?

- external BGP - usually same link
  - manually configured on some telco links
  - if same ethernet segment, ARP will do it for us
- internal BGP - may be multi-hop
  - if so, rely on IGP to get the job done
    - » note: BGP control and routed packets (data)
  - of course, that could include static routing
  - IGP/EGP convergence problem - touch on this later

topologies

- **transit network** - packets are routed thru it, may not source/sink
  - multiple external and internal BGP peers
  - likely to have full Inet routing table (>=75k)
- **multi-homed stub**
  - stub does not have transit packets, src/sink only
  - > one way out - may be for redundancy
  - needs AS number
- **single stub** - one way out only
  - doesn't need AS or BGP for that matter

topo picture



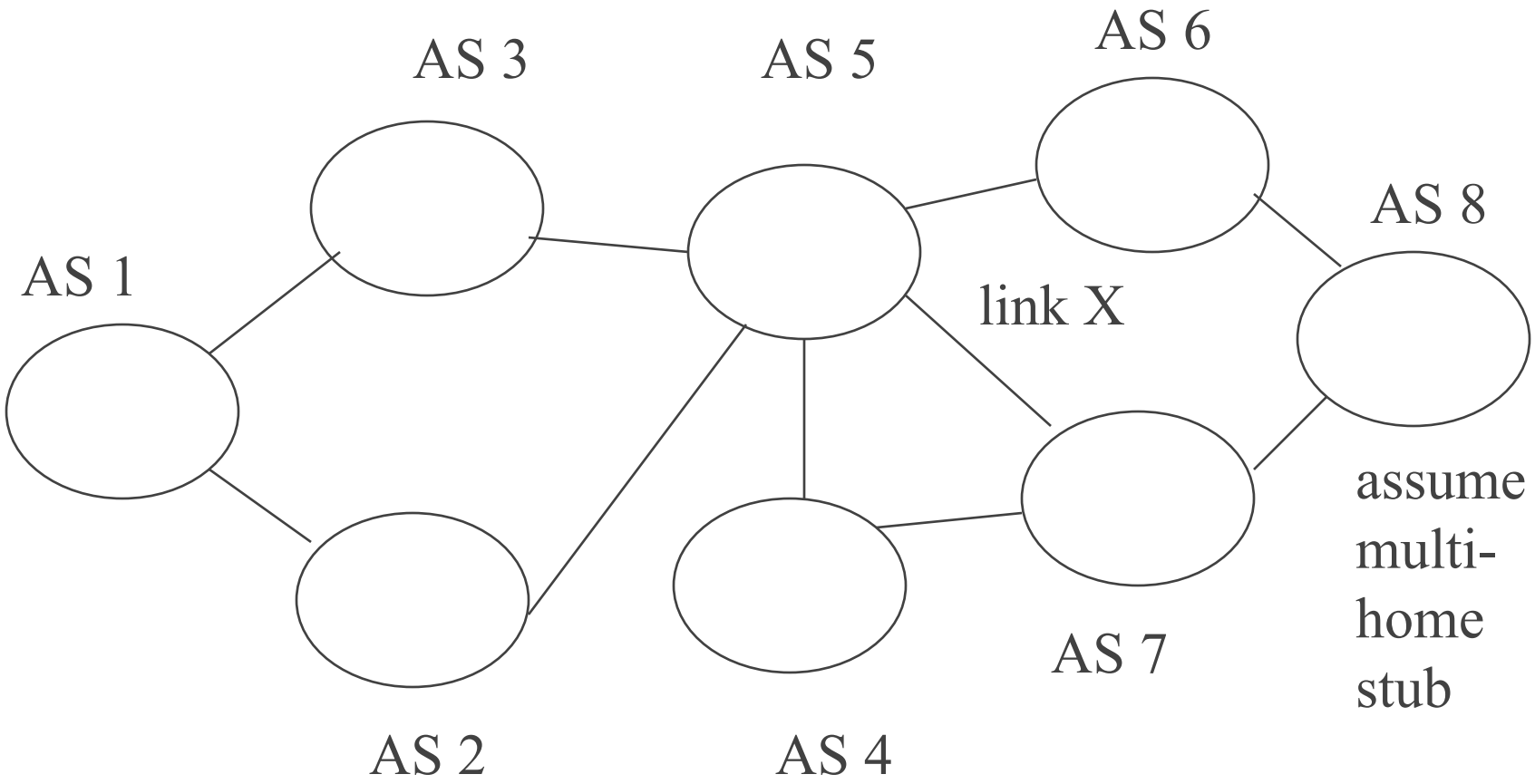
stub routing (no need for BGP)

- 1. simply use static route
- 2. get default route dynamically using IGP, RIP, OSPF, whatever from ISP/transit
- 3. use BGP (training-wheel version)
  - likely to have fake AS, private AS numbers exist, and ISP/transit system can simply not advertise them, instead make stub appear as part of its AS routing space

BGP as routing protocol

- Distance-Vector with a twist
- basic BGP logical update consists of:
  - (ip network(D), subnet mask, "attributes")
  - this is oversimplified, deal with this later in protocol
- we make routing decisions based on **attributes** (multiple) + manual configuration, however
- one attribute is the Vector; i.e., the **AS path** expressed as a complete source route of AS
- (to net 111.0.0.0, via AS 1,2,3,4,5)

BGP AS path



assume Net number and AS number the same

A7 - BGP routing database for A1 then:

- **1. to N1, via AS5, AS3, AS1, next hop IP, etc. (3 AS hops)**
- 2. to N1, via AS5, AS2, AS1, etc
- 3. to N1, via AS4, AS5, AS3, AS1, etc
- 4. to N1, via AS4, AS5, AS2, AS1, etc.
- default policy may be to choose least hop count, therefore choose #1 above
- what happens if link X goes away?
- we can choose route #3, thru AS4, 4 hops

route UPDATES

- note that as route is forwarded, one's own AS is **prepended**
- e.g., AS3 update about AS1 to AS5
  - input AS1, output AS3, AS1
- this gives us a metric and it helps us remain loop-free at layer 3
  - and handle loops at layer 2
- simple rule: **if you see yourself in the AS path, that's a loop, (and an error)**

BGP is not RIP

- does not send entire routing table every N seconds
  - **sends full routing table at boot** (good thing about TCP)
  - only sends updates upon change (new or **withdrawals**)
- does not do count to infinity
  - stores multiple paths in database (RIB) and can choose new one if available
  - and know topology because of AS path (can't fool me)
- routing updates may be chosen on best hop count in terms of # of AS, a default metric therefore exists (more on policy in a bit)

e.g., back at AS path picture

- if using RIP, AS2 might be told by AS1, AS3, is one hop, therefore AS2 might tell AS1, AS3 is two hops
  - but mean two hops thru A1 ?!
- with BGP, AS1 sends route AS1, AS3
  - AS1 will not accept AS2, AS1, AS3 from AS2

however - regarding policy routing

- routing choices may be made on basis of "policy"
  - policy mechanism not as flexible as arbitrary src routing, as a simplification for now, you can:
  - ignore routes or some routes from A
  - send all or some routes to B (or none)
  - policy based on **IP address**, **AS number/path** or **Communities** (sets of routes), and/or BGP attributes
  - and manual configuration choices about same

NSFNET

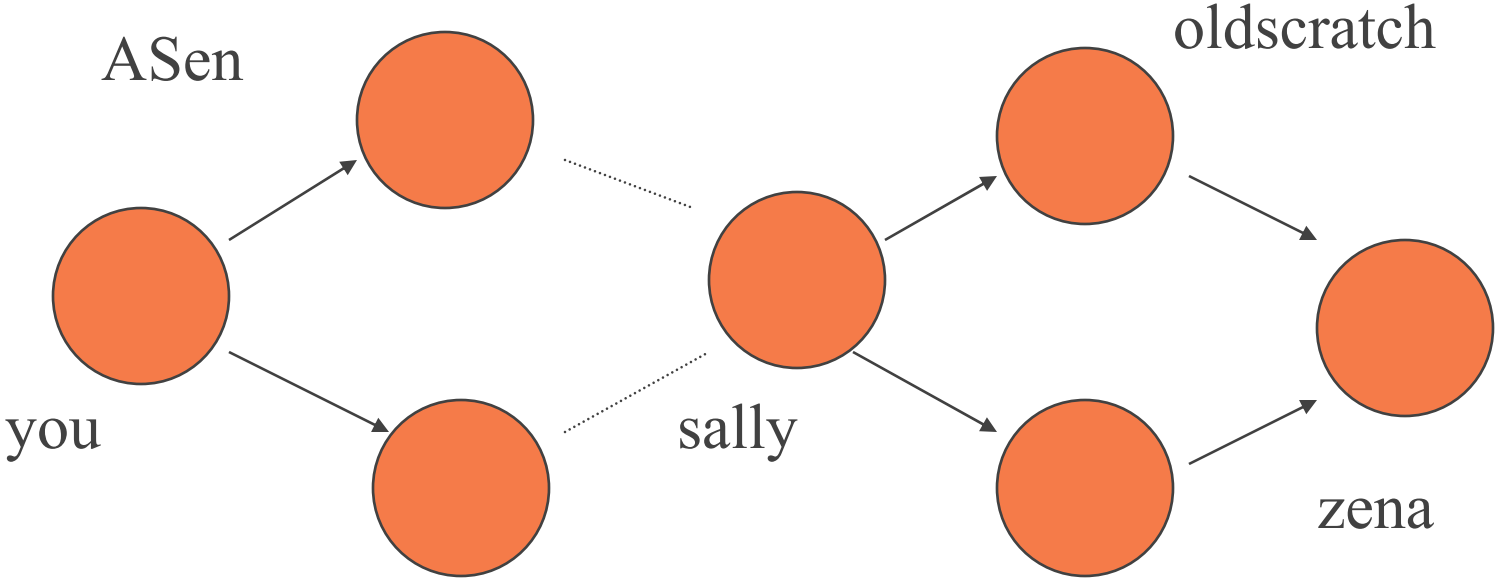
- as sole Inet backbone
- way back when, got us thinking about this
- Acceptable Use Policy:
  - not ok for business to use govt. funded net
  - therefore business had to somehow tunnel around it
- another possibility: don’t make silly rules

policy routing and BGP

- we might distinguish **policy-in-the-large** and **policy-in-the-small**
- e.g., IDPR was after end to end policies
  - not clear how to administer though (more lawyers)
- BGP can't do that, so let's admit it and move on
- your policy affects this router or your set of routers in your AS
  - you can only hack at other people's policies ...
- essentially **manual** and **locally** configured

BGP policy is hop-by-hop (mostly)

- an example of something you can't do



sally can choose to not advertise you r routes to sweetangel or just have a static route to oldscratch for zena ... you cannot control sally

sweetangel

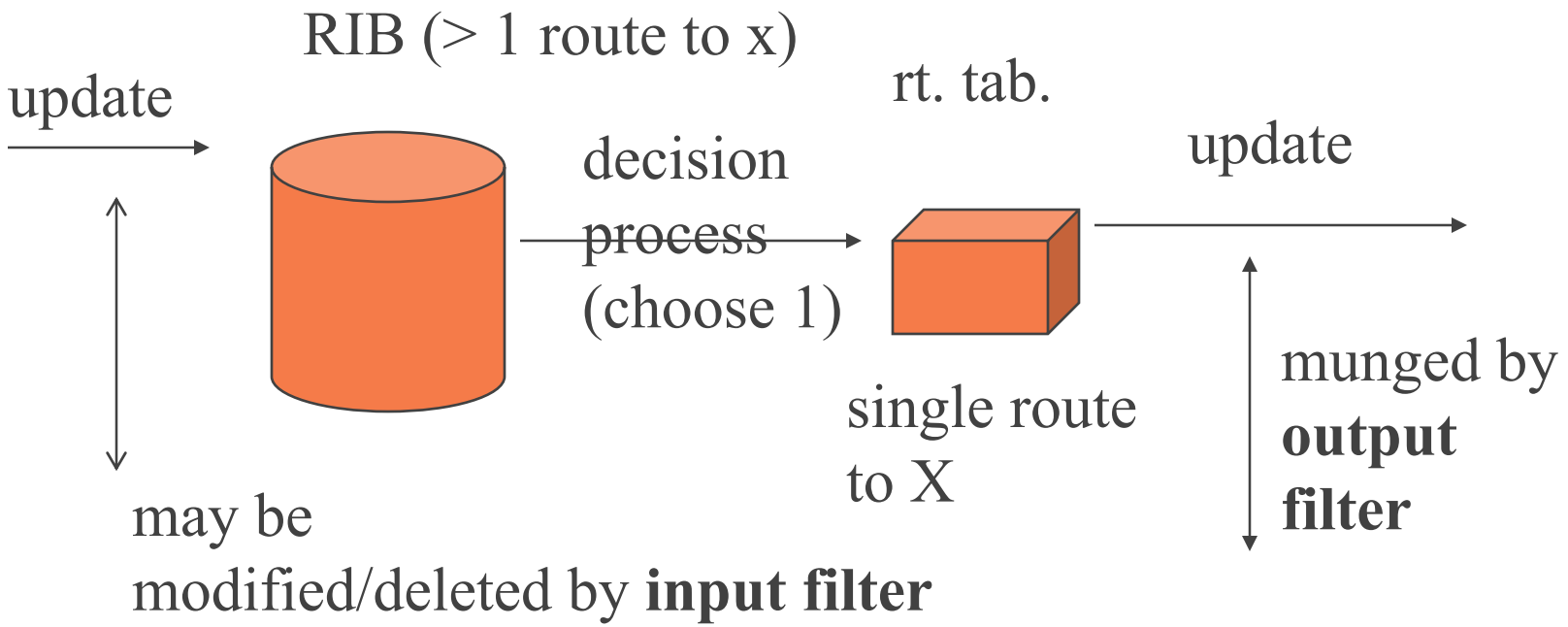
one other little item - asymmetric routes

- in the preceeding slide we wanted to route thru sweetangel to zena
- but got routed thru oldscratch
- zena might have a default route thru sweetangel
- thus paths could be asymmetric
- this is not unusual

Cisco scheme for how BGP routing proceeds (overview)

- we get UPDATES (new or withdrawals)
  - we subject them to input policy configuration
- survivors are stored in routing database
  - IETF term is **Routing Info Base (RIB-IN)**
- **decision process** chooses "best" (acc to policy)
- puts chosen best route in routing table
  - in theory, BGP routing table
  - subject these routes to output policy config
- advertise those routes put in routing table to peers

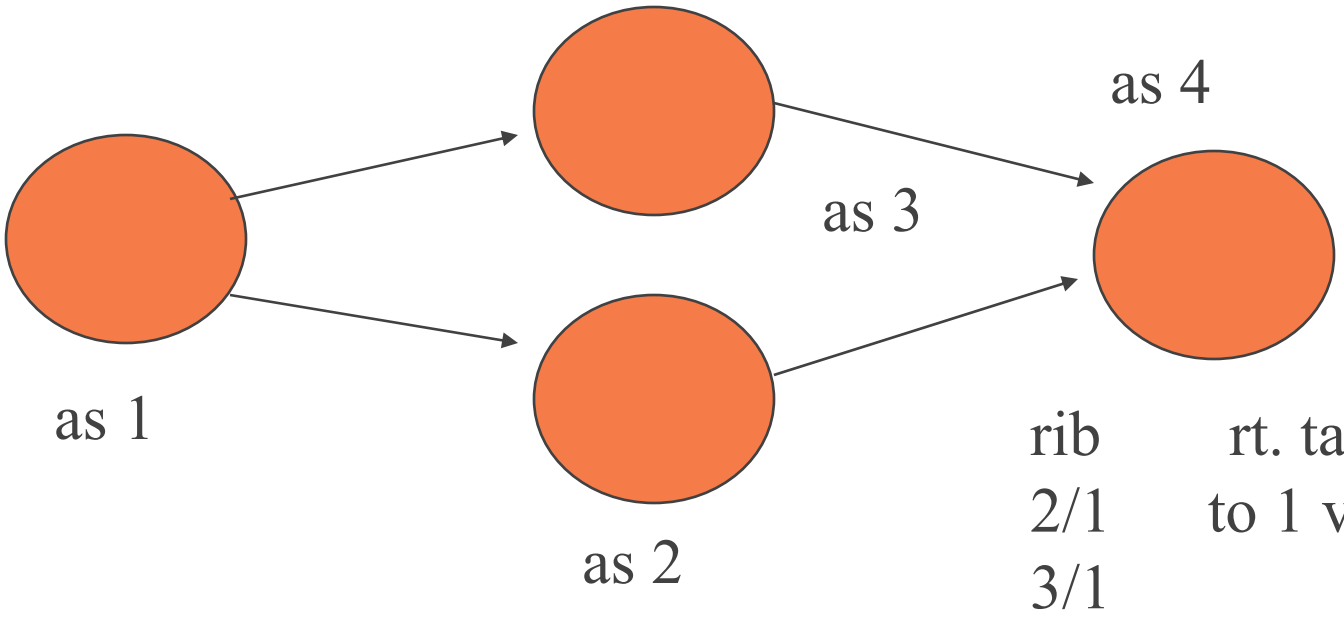
picture of BGP router process



important principle

- BGP does hop by hop routing, therefore
- **we only advertise what we use**
- if we put it in our routing table
  - we MAY advertise it, depending upon output filtering
- if we receive a routing withdrawal and it is
  - in our RIB only, what do we do?
  - in RIB and routing table, what do we do?

assume as4, lose as2 or as3



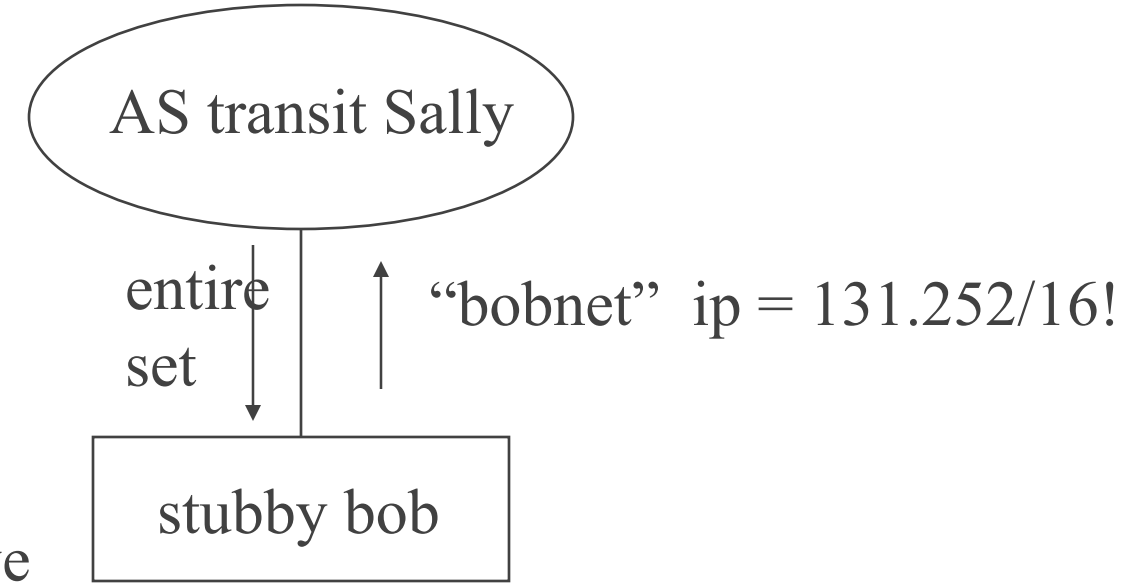
1. if as2 lost, we don't change routing table, no update
2. if as 3 lost, we have 2/1 in rib, change routing table, to 1 via 2, send update

convergence with BGP means what?

- not all RIBs are the same for sure
  - (different vectors, and other attributes)
- same set of IP dsts, with at least one path, and one routing table entry
  - which may differ from R to R
  - important assumption: policy does not lead to partition of Internet (has happened)
- policy can cause differences of course
- **flapping** - route goes up/down at high frequency, leads to mucho BGP updates

stupid BGP mistake

if sally sends us full Inet routing table, what should we send her? what should we not send her?

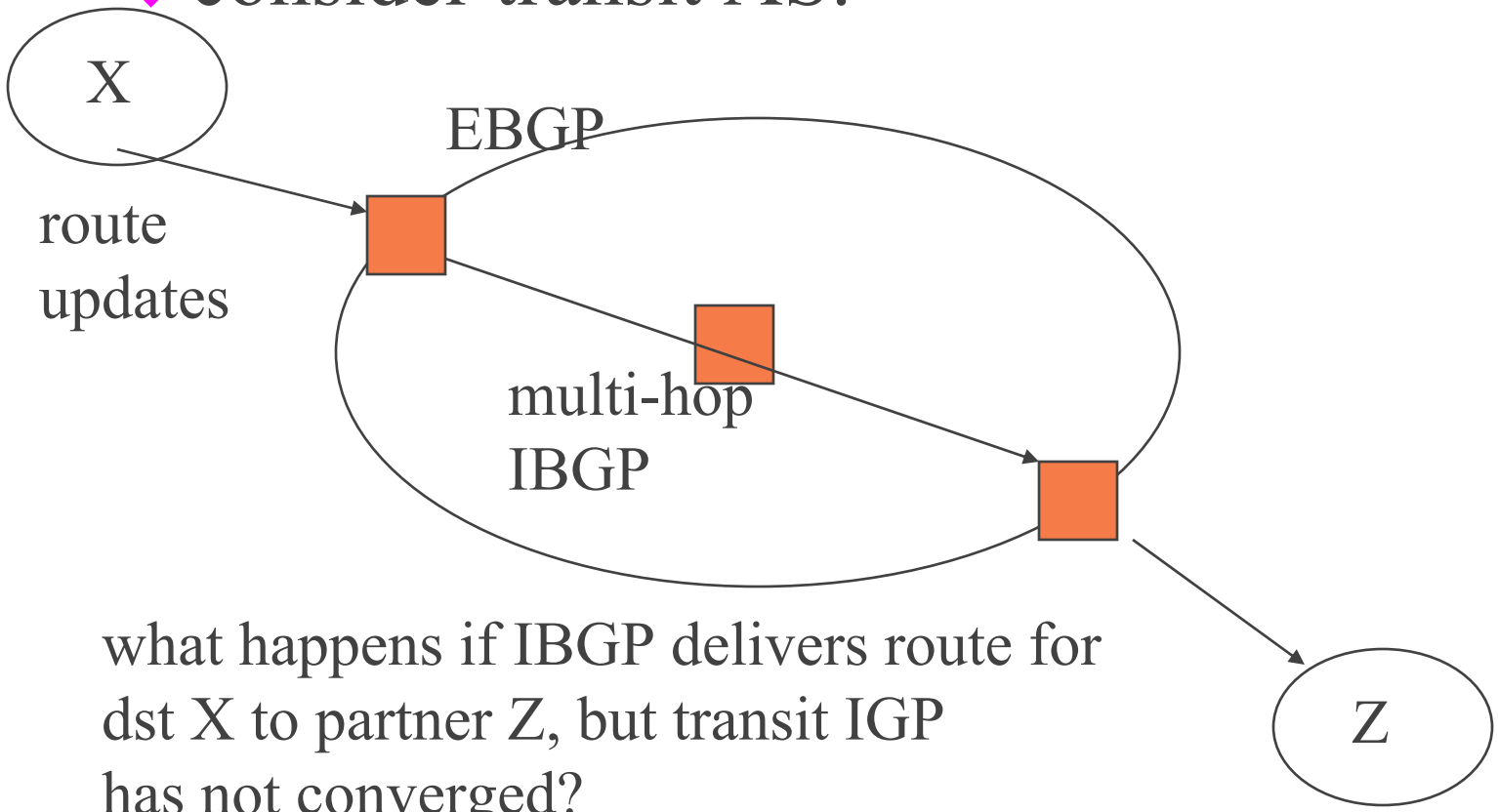


summary: some update rules

- we only advertise what we put in our routing table
- updates are not refreshed
  - RIB entries do not time out
- BGP only talks when something changes
  - updates are adds or withdrawals or some other change based on attributes
- any RIB change drives the decision process
- we exchange routing tables at boot
- all of above subject to policy configuration, in/out

IBGP/IGP issues: 1. synchronization

- consider transit AS:



what happens if IBGP delivers route for dst X to partner Z, but transit IGP has not converged?

IGP sync:

- answer: we must somehow make sure that the IGP has converged before
- EBGP is advertised to Z
  - remember I send you routes, you send me data
- why? because IBGP is multi-hop, and interior router might not know path to X
  - black hole ...
- in general: don't send route until you can forward ...

how do we solve this problem?

- 1. we could wait for **IGP synchronization**
  - e.g., EBGP router to Z can't advertise until IGP "route tag" shows up and
  - local IGP routing table shows path to X
- acc. to Moy, transit AS do not want to dump full Inet routing table into IGP
  - e.g., OSPF on all routers does SPF calculation over and over again during route flap
  - you have >= 150k routes == ouch

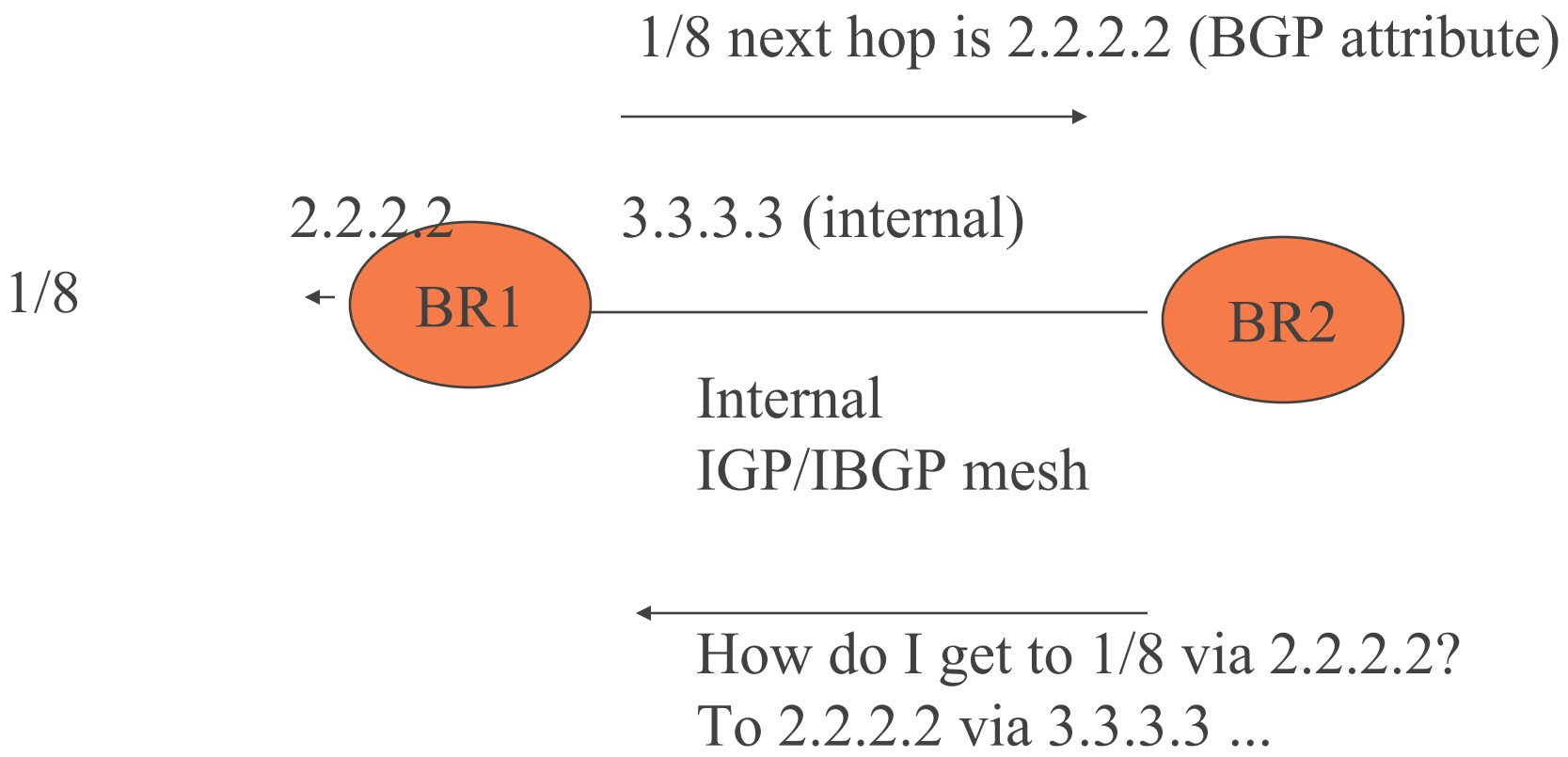
plan B, C, D, etc.

- 2. all internal routers use IBGP (aka use BGP ...)
  - with no synchronization
  - IBGP is IGP (deal with it ...)
  - IGP basically gets you to next hop
  - wait: we have a potential N**2 problem …
- 3. or possibly default route plus a few IGP routes leaked in (if possible)
- 4. or route recursion …
- 5. or simply tunnel over internal routers – can use logical circuits courtesy of MPLS or possibly vlans courtesy of Ethernet (or ATM circuits)

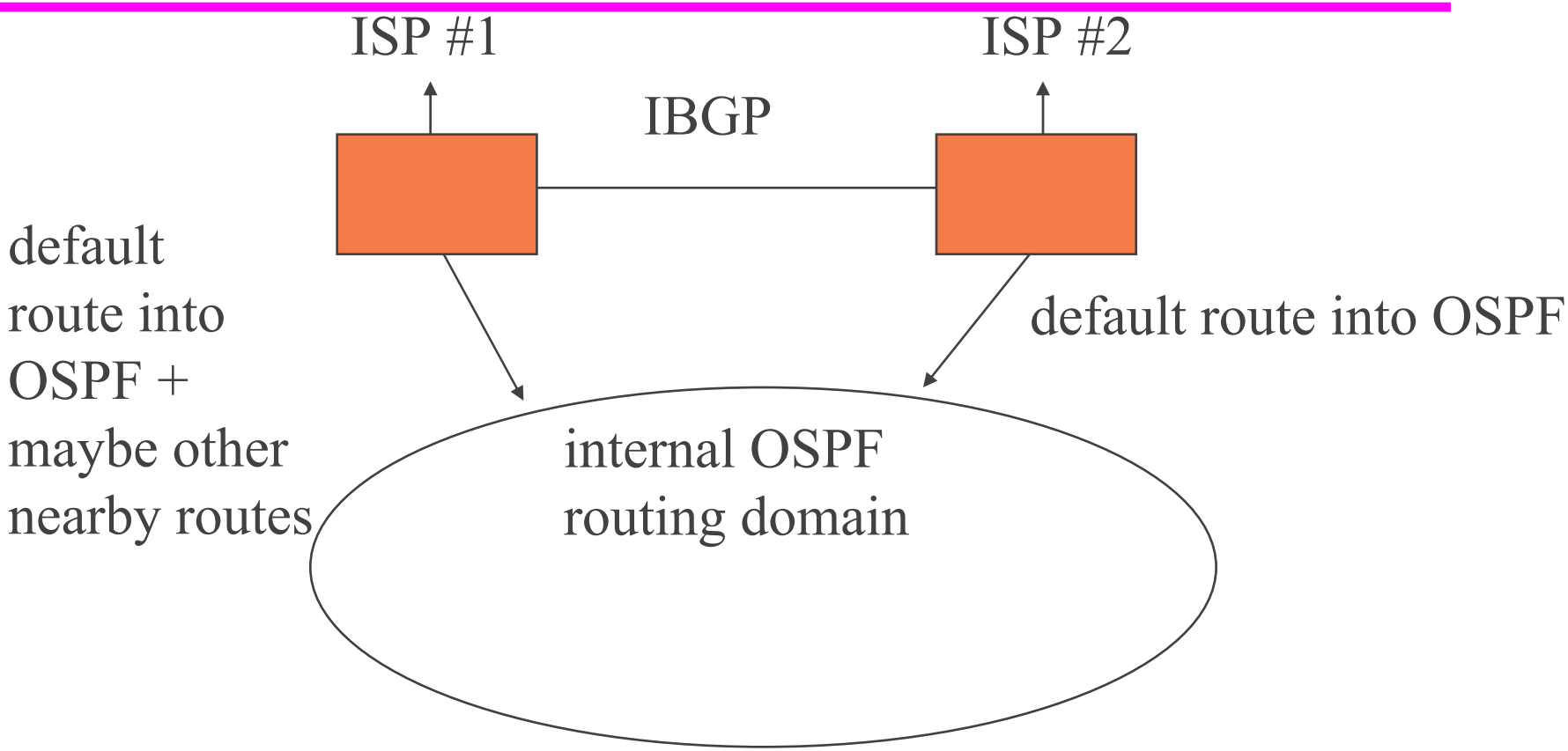
common implementation idea

- combine next-hop bgp attribute with
- recursive routing table lookup
  - (similar to an IPIP tunnel but not the trick)
- control: next-hop for ip X is router Y
- routing back: next-hop is NOT directly connected router, therefore must “tunnel” back to Y

recursive lookup picture



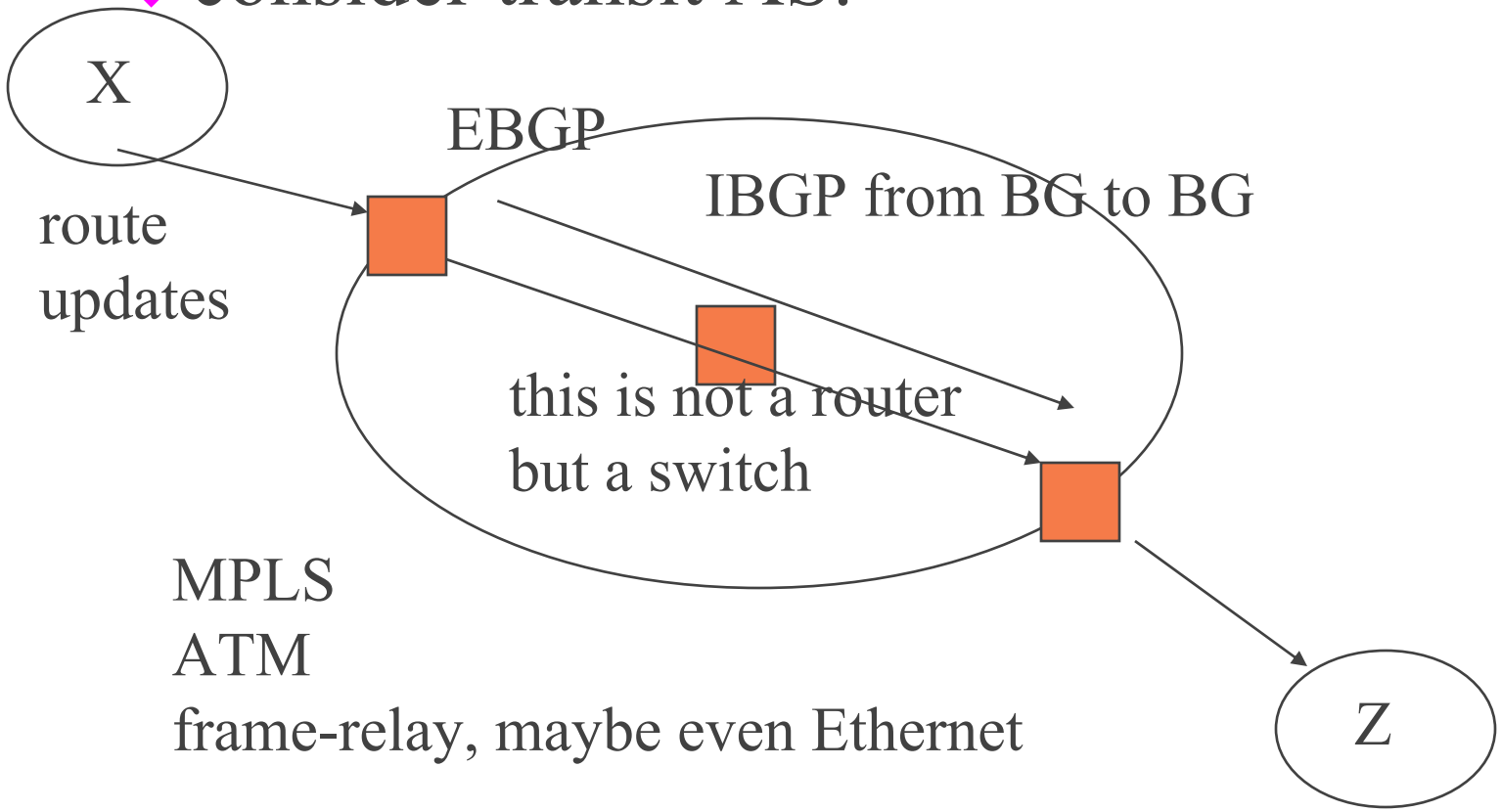
stub AS might be implemented like so:



therefore default routes can help out in this case

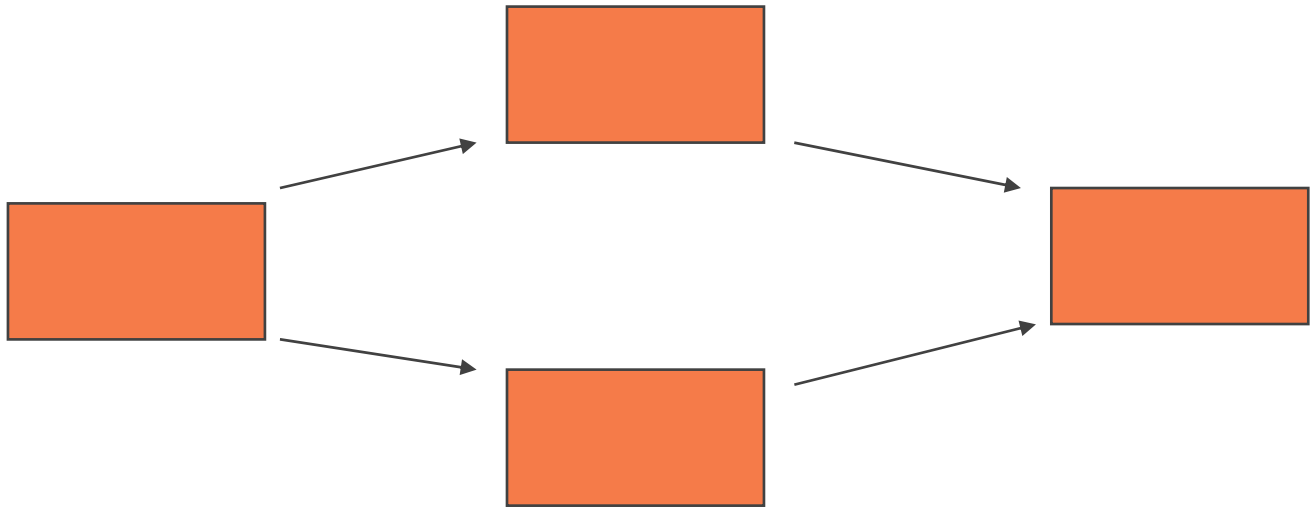
circuit or logical circuit

- consider transit AS:



MPLS – very short intro

- ATM allows circuits across switches



multiplexing and circuit paths

based on tags (small ints) in cells

setup manually or dynamically (signaling protocol)

Multi-protocol layer switching

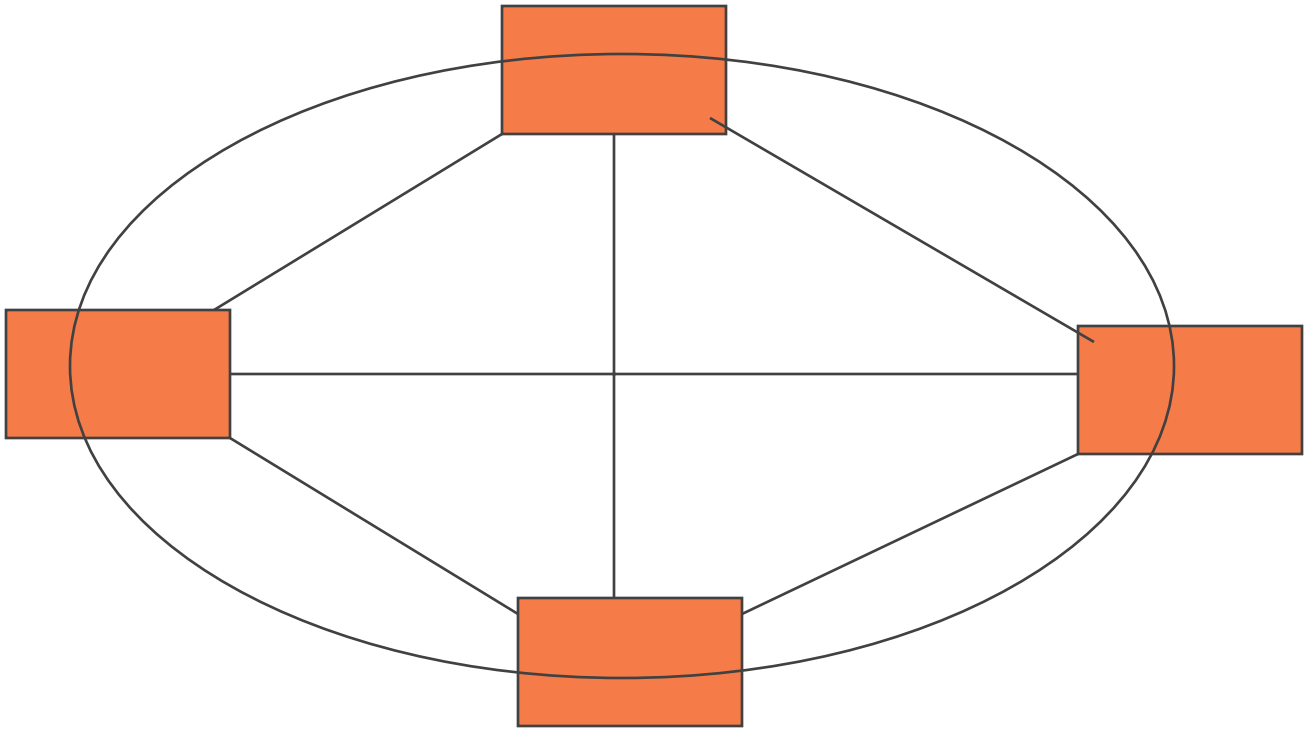
- logically between L2 and L3
- not L2 specific
- can setup signal path
- basically “tunnel” across a domain
- offers possibilities for traffic shaping, QOS, VPNs, and more or less making L2 link go further
- and has tags like ATM

another IBGP issue

- in-order to remain loop free, **all AS internal routers must peer**
  - same AS, we can't add it as a prefix
- call this **full-mesh IBGP**
- in large AS, this leads to manual configuration nightmare
  - all those TCP connections, N**2 more or less
- thus notions of route reflectors, route confederations to improve intra-AS scalability

full meshed IBGP

must have peer connection for all peers



mechanisms exist for making IBGP mesh more scalable

- route confederation notion:
  - break single AS up into multiple internal AS
  - tie together with EBGP connection
  - to outside still appears as one AS
  - each internal group must have fulled meshed IBGP
  - next-hop, MED, and local preference attributes important

route reflector

- in addition to confederation, we may have **route reflector** (internal route server)
- AS divided into clusters
- each cluster has route reflector
- route reflector “reflects” updates to internal cluster peers, thus no full mesh in cluster
- clusters have IBGP connection between them - need complete connections here

note re IBGP and attributes

- AS_PATH is NOT incremented,
  - therefore must manually prevent loops
- NEXT_HOP is not touched either.
  - it's the way out of the AS with IBGP
  - need recursive lookup to send pkt in direction of next-hop

the protocol

- open/close state machine as virtual circuit
- TCP, port 179
- TCP pros
  - we don't have to resend or be reliable
  - don't care about fragments/resends/loss, TCP job
  - we can be message-based, variable length
    - » BGP is TLV protocol design more or less
  - hence updates can be incremental
- BGP is stateful due to TCP and RIB both

TCP cons

- we need our own keepalive as we cannot rely on TCP keepalive
  - or assume all link hw has up/down indication
- TCP might slow-down due to congestion control
  - doesn't make sense to have BGP as control slow-down in the face of "real video" ???
- BGP level security would not prevent TCP level attacks
  - e.g., you have authenticated BGP, you face TCP sequence number spoofing

BGP message types

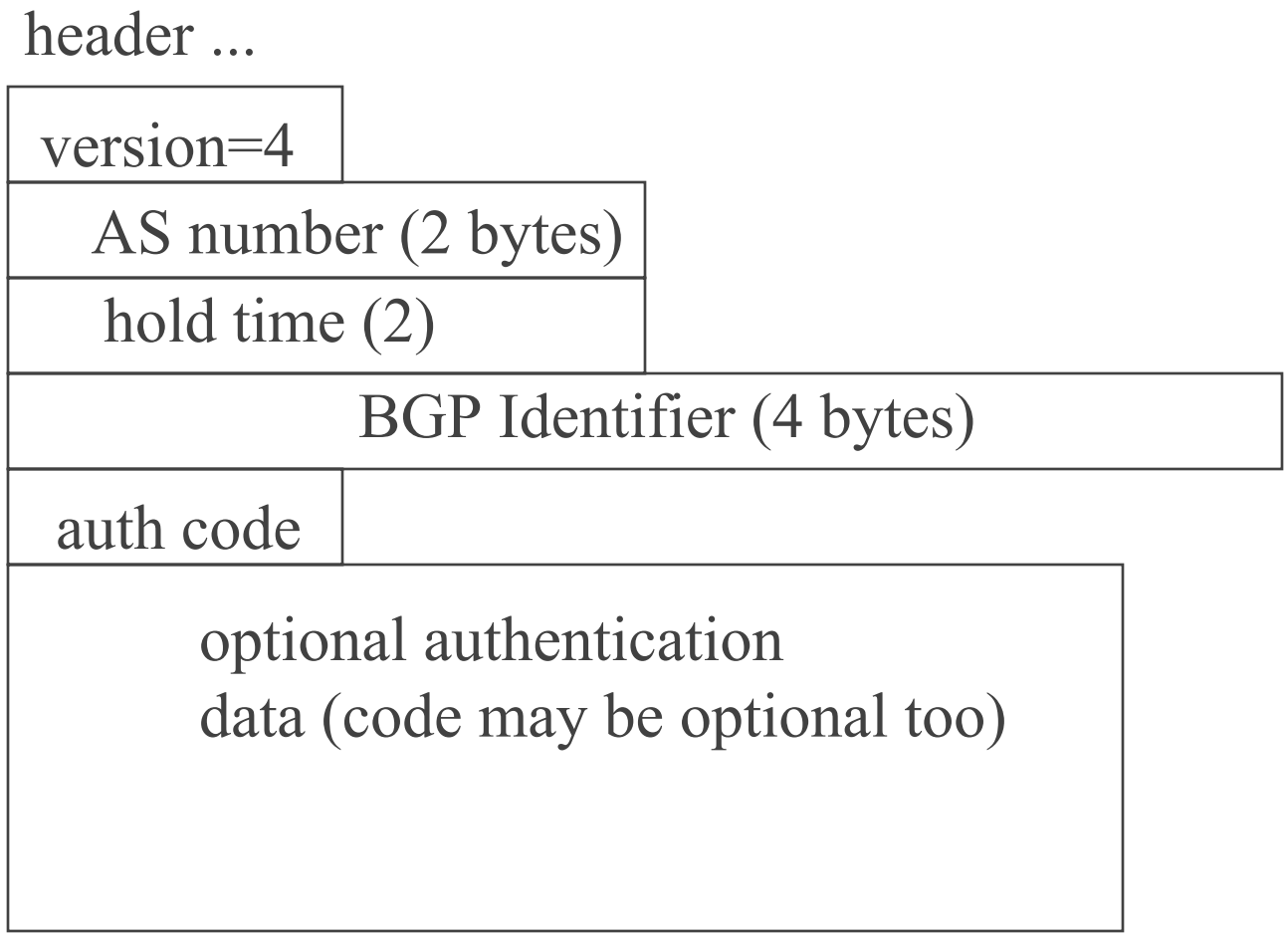
- 1 **OPEN** - start of connection
- 2 **UPDATE** - set of route withdrawals or new routes
- 3 **NOTIFICATION** - fatal error or close
- 4 **KEEPALIVE** - I'm still here partner
- all messages have common header
- messages overlayed on TCP byte stream

header

- all BGP messages started with 19-byte fixed length header
- marker can be used for checksum (e.g., MD5) or simply as framing/redundancy check (must have expected value).
- e.g., if no authentication, then marker is all 1s.
- length, acc. to RFC 1771, 19 to 4096

16 byte marker	
length (2 bytes)	type (1 byte) (1,2,3,4 for values)

open message



open

- post connect, 1st send OPEN, get KEEPALIVE back if OK, else NOTIFICATION
- hold time - sender states in seconds time in which peer must send keepalives
  - or updates, but if no updates, then keepalives
- ID is a local IP address
- it is possible that both BGPs will connect at the same time
  - if so, one connection closed, winner has higher IP in ID

multi-protocol BGP note

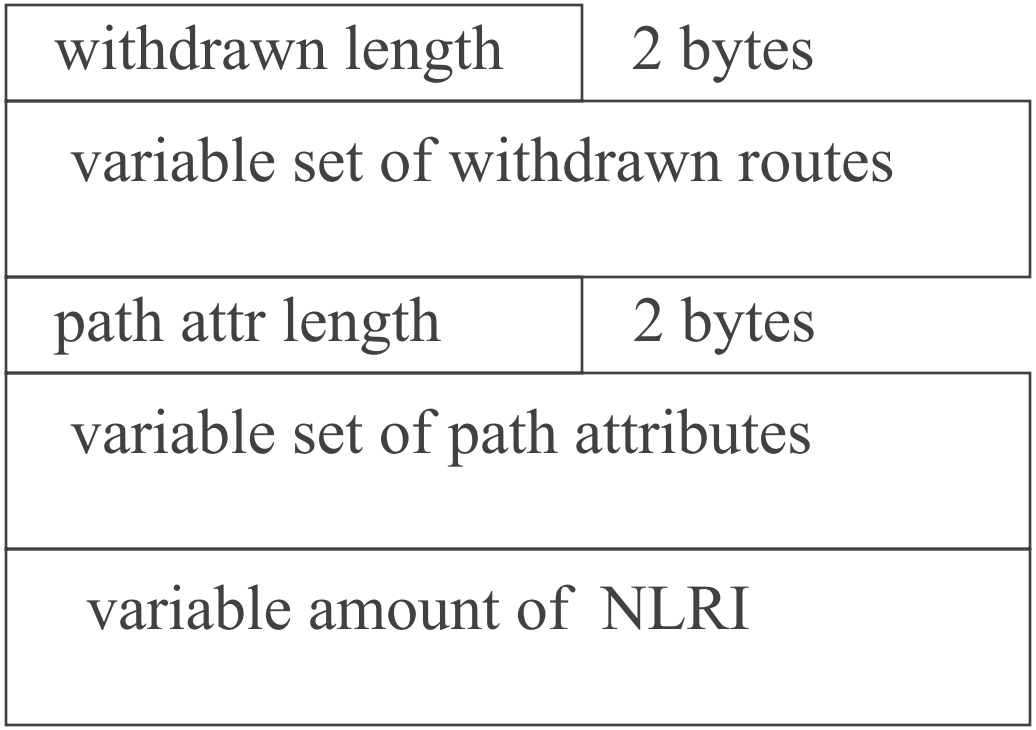
- note that open takes options
- multiprotocol BGP can thus be negotiated with these options:
  - capabilities negotiated at OPEN
  - includes MPLS, Multicast, IPv6
  - attributes for multicast NLRI also exist
- this allows BGP to do more than IPv4

updates

- contain two parts (either of which may not exist), more or less:
- (withdrawn IP nets (possibly > 1), one path)
- however one path consists of
- (path attribute length, attributes, NLRI)
- the path is in the attributes
- NLRI - network layer reachability information
  - set of possibly > 1 IP addr/masks (lengths really)
  - therefore these NLRI share the attributes

update

header ...



update, cont.

- withdrawn, aka unfeasible
  - if len = 0, there are none
- routes expressed in length/prefix form
  - length is 1 byte long, comes first
  - e.g., 8/64 would be 64.0.0.0/8
  - netmask, but actually contiguous prefix
  - both withdrawals, and NLRI like this
- withdrawn routes - routes to toss out of RIB
  - may or may not affect routing table

path attributes are complex part

- encoded as triple (type, length, value)
- type actually (flags as byte, type code)
- flags =
  - optional - else mandatory (msg must contain it)
  - transitive - pass it along, even if unrecognized
  - partial - set to 1 if unrecognized transitive anywhere in path
  - extended - used to indicate length 0..N

path attributes thus have 4 categories

- 1. well-known and mandatory
  - well-known, all implementations must do it
- 2. well-known and discretionary
- 3. optional transitive
- 4. optional non-transitive
- thus we can have attributes that may not be known to all implementations AND passed on or dropped (non-transitive)

before we nerd out on attributes

- bottom line: attributes are one more input for policy
- therefore policy is a function of
  - **attributes** in BGP updates
  - **local rules** about things like IP dst (NLRI), AS paths (one attribute among many), communities (another attribute)
  - and other possible manual config items, e.g., you can ignore an attribute

attribute types

- **ORIGIN/ mandatory**
- **AS_PATH mandatory**
- **NEXT_HOP mandatory**
- **MULTI_EXIT_DISC** (aka MED)
- **LOCAL_PREF**
- ATOMIC_AGGREGATE
- **AGGREGATOR transitive**
- **COMMUNITY transitive**
- ORIGINATOR_ID
- CLUSTER_LIST
  - about 2 for route reflection
- DPA **transitive**
- ADVERTISER
- RCID_PATH
  - above 2 for route server
- more may be defined
- note: not all explained here!!!

attributes explained

- ORIGIN may be {IGP, EGP, or INCOMPLETE)
  - historically used to indicate EGP origin during EGP to BGP transition
  - IGP means BGP injected route
  - INCOMPLETE means route redirection
    - » static or OSPF or something
  - created by route originator
  - can make policy decisions, (IGP better than INCOMPLETE)

attributes, more

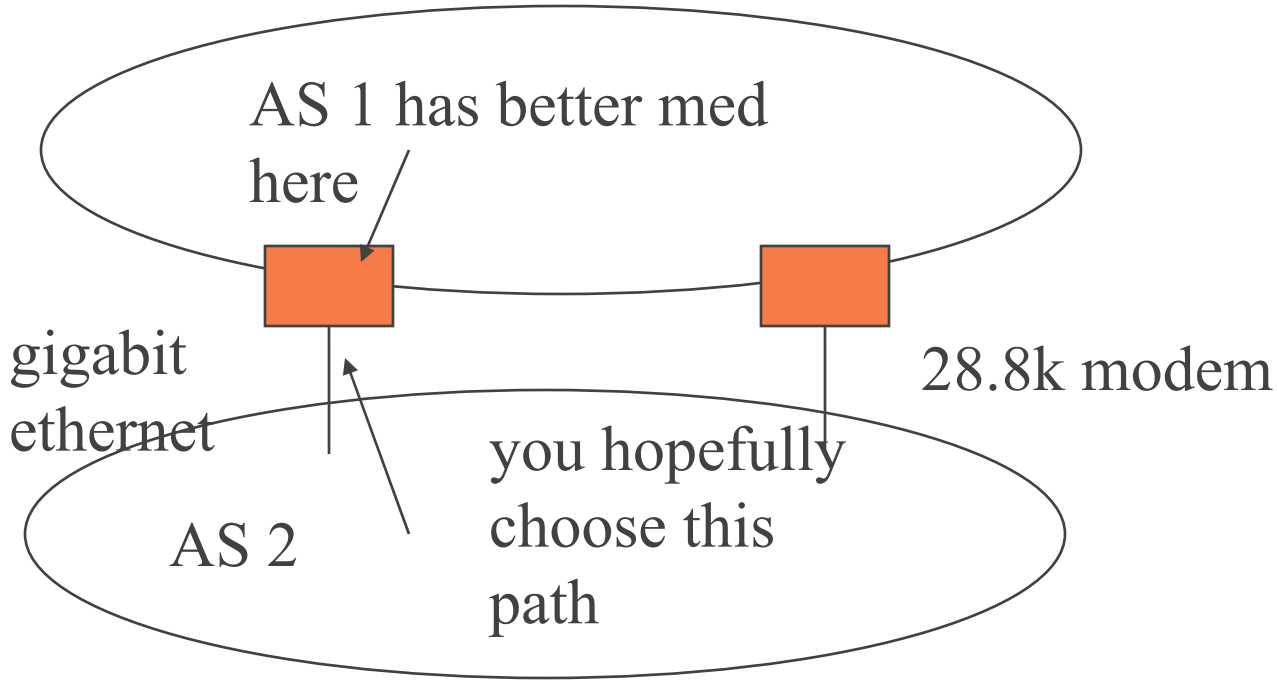
- **AS_PATH** is required
- if IBGP, then NULL, else prepend own AS
- path is a list of segments (ASen) expressed as TLV
- Tag is either
  - AS_SET - unordered, i.e., not a sequence
  - AS_SEQUENCE, ordered
- aggregation can muddy the path; e.g.,
  - 1, as_set = 2,3 as path is 1,2 or 1,3

attributes never end

- **NEXT_HOP**, router A on this link suggests using router B as next hop instead of A
- **MED** - AS 1 has two points of attachment to you, the MED indicates preferred path
  - it is a weight
  - lower value win
- **LOCAL_PREF** BGP uses this to tell IBGP peer/s that it is best way to outside X
  - higher value wins

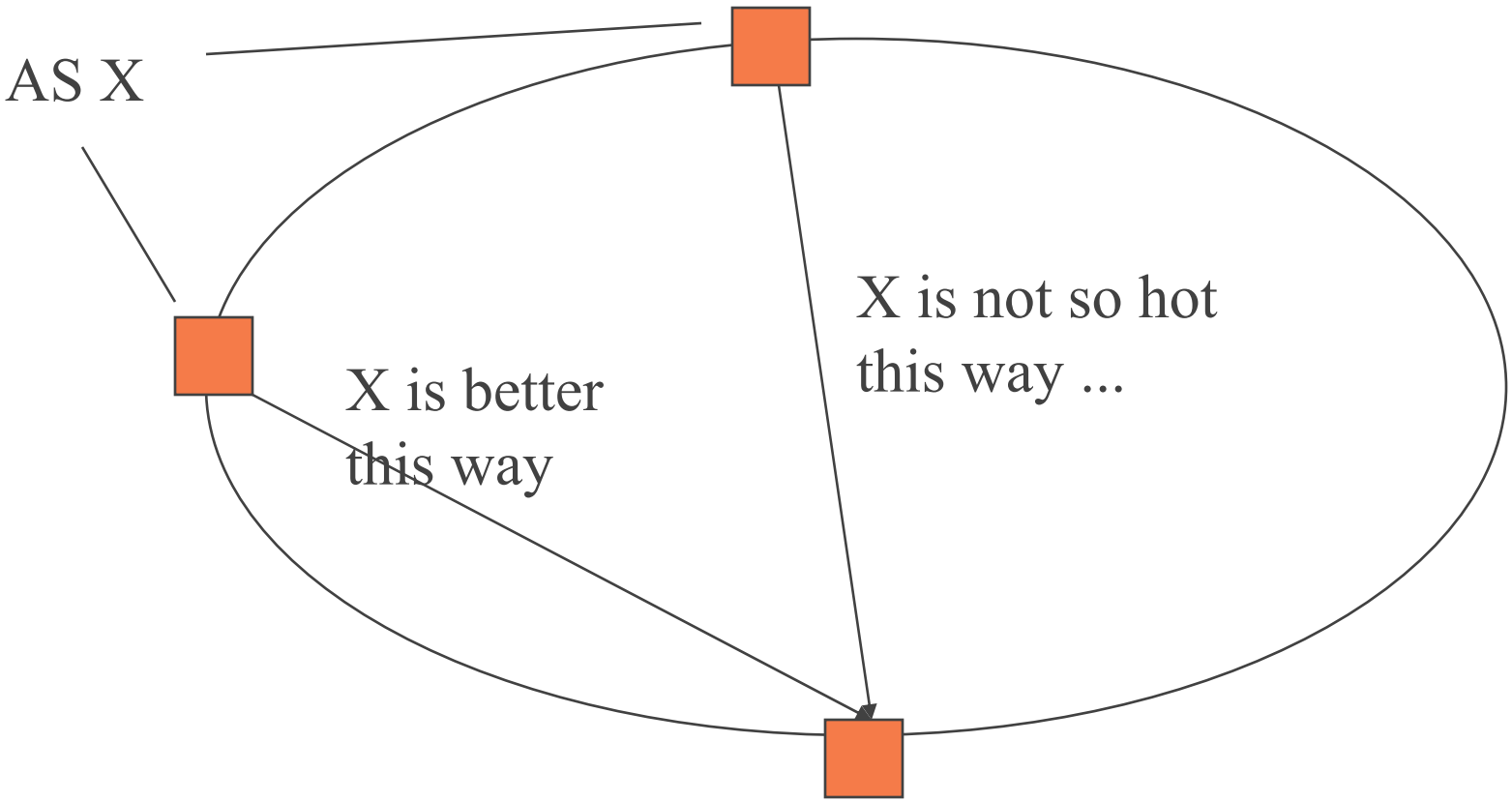
MED picture

note: this is near-local attempt to influence another AS



AS 1 uses MED to tell AS2 what local link to use

LOCAL_PREF



more attributes

- AGGREGATOR - info only, AS X committed aggregation on this path
- COMMUNITY - arbitrary routes grouped together as a set ... call it a **route-bundle**
  - useful for policy (I will forward the state of Kansas, but not the state of Missouri)
  - often stripped at AS boundaries, even though transitive
  - allows you to use tags as opposed to addressing info

community

- predefined attributes include:
  no-export - do not send this to EBGP peers
  no-advertise - do not send this to anyone
  internet - send this to everyone (the uber-bundle)
- E.g, an AS might distinguish between routes from UUNET, I2, and routes internal to itself, and tell its own customers which is which

Cisco weight attribute

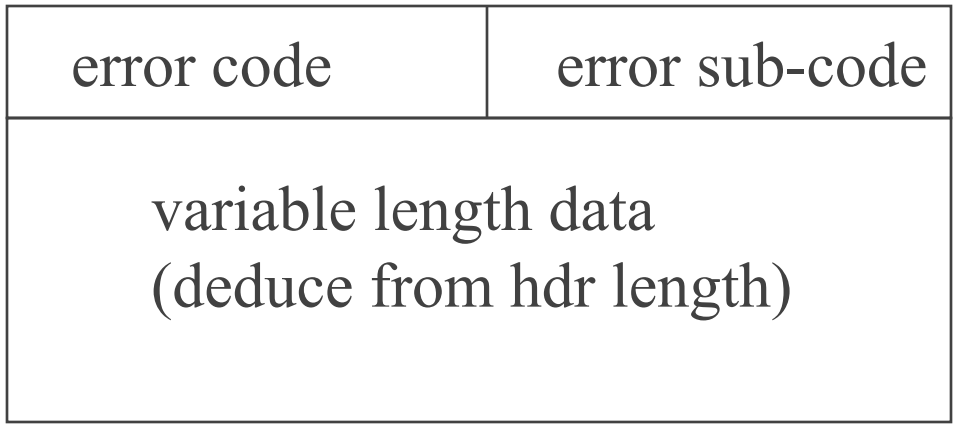
- cisco-defined and local to a router, not BGP protocol
- R1 recvs route X from R2 and R3
- if from R2, weight is 50
- if from R3, weight is 100
- bigger weight is put in routing table

summary: attributes/plus Cisco weight

- MED
- LOCAL_PREF
- Cisco admin. weight
- COMMUNITY
- AS_PATH
- ORIGIN
- NEXT_HOP

notification

header (marker, length, type=NOTIFICATION)



notification protocol

- when?
  - error
  - e.g., holddown elapsed
  - or graceful close (on purpose)
- result is peer connection is closed
  - errors are fatal
- and hopefully log message ...
  - oh admin - things are bad here ...

notification error codes (major, minor) codes

- 1 - message header errors
  - (error = 1, sub-code=1), connection not synchronized
  - (1,2) - bad message length
  - (1,3) - bad message type
- 2 - open message
  - (2,1) - bad version number
  - (2,2) - bad AS
  - (2,3) - bad ID
  - (2,4) - unsupported optional parameter

notification errors, cont.

- 3 - update message error
  - quite a few ... problems with attributes
  - note (3,7), AS routing loop
- 4 - hold timer expired
- 5 - finite state machine error
- 6 - cease (close ... not really an error)

keepalive from 1000 miles up

- BGP messages only occur if there are routing topology changes
- keepalives on link are how we learn about link failure
  - and are rather important
  - we may not be able to trust a specific kind of link to tell us (keepalive is sw fix on flaky hw)
  - we may not be able to trust TCP keepalive, therefore BGP does not use

keepalive

- nothing but (marker, length, type=KEEPALIVE)
- in order to avoid connection failure
- must send message or KEEPALIVE
  - within holddown time
- zero holddown means no KEEPALIVES needed
  - perhaps we want to avoid link charges
- keep in mind transport is TCP, therefore delay an be unpredictable
  - keepalive frequency > holddown time is good idea

routing decision process

- we have RIB (database) paths and other attributes
- we must process them into routing table entries
- the decision process is the algorithm here
- logically we do the following (acc to 1771)
  - 1. choose routes to advertise to IBGP peers
  - 2. choose routes to advertise to EBGP peers
  - 3. route aggregation and route information reduction
- some function is applied to all possible candidate routes for IP dst X, highest preference wins

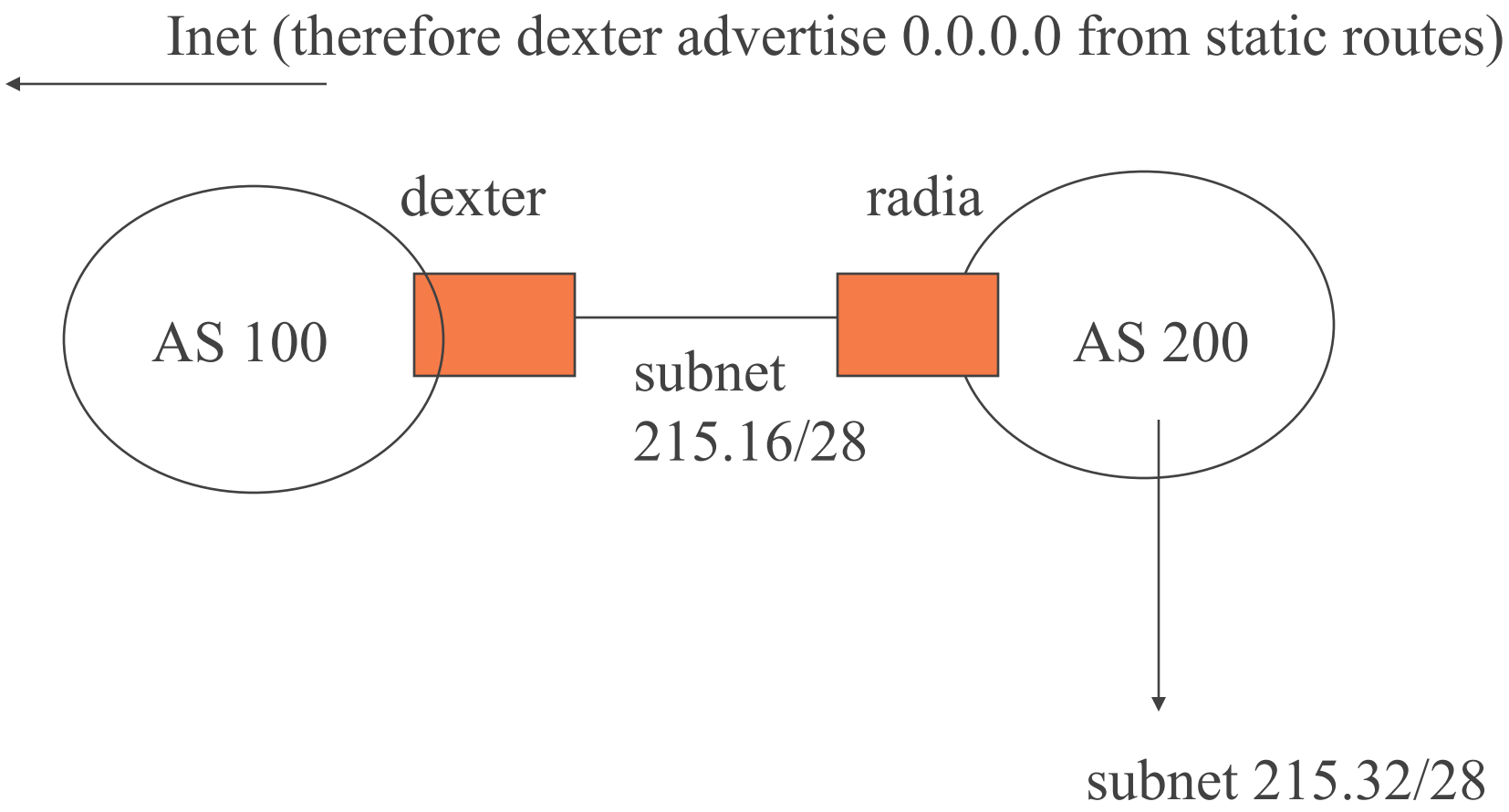
condensed cisco algorithm

- next-hop route must exist (may need IGP to provide it)
- consider larger administrative weights first (Cisco weight)
- prefer route with largest local preference, else if same prefer local originated
- if none of above, choose shortest AS_PATH
- prefer IGP over EGP (ORIGIN)
  - IGP better than EGP better than INCOMPLETE (which appear because of route redistribution)
- prefer lowest MED metric
- if MEDS same prefer EBGP over IBGP
- else if tie, prefer lowest BGP ID

basic Cisco setup

- router bgp <as-number>
  - network <network-number> mask <mask number> [route-map route-map-name]
- Note: network injects local network into BGP, but does not specify which IP addr to use for peer connection
  - neighbor <ip-address> remote-as <number>
- Note: neighbor specifies peer and peer AS

logical network layer - 2 EBGP peers



simple example - dexter

- router bgp 100
  - network 131.252.215.16 mask 255.255.255.240
  - redistribute static
  - neighbor 131.252.215.18 remote-as 200
  - default-information originate

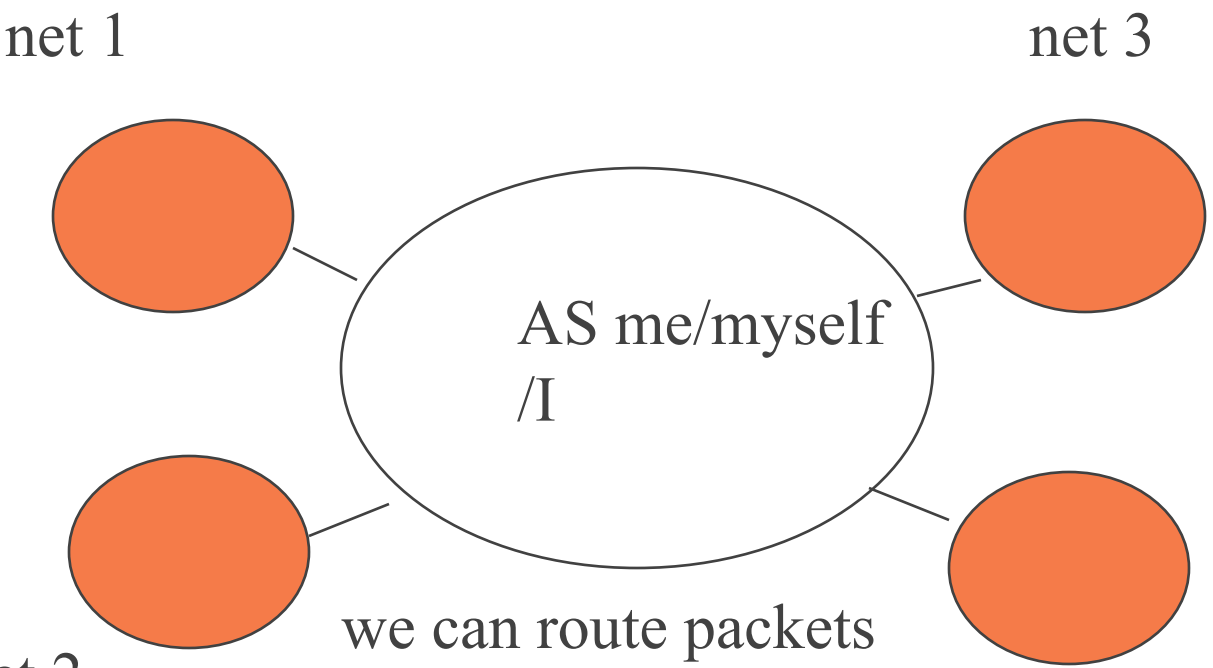
simple example - radia

- router bgp 200
  - network 131.252.215.32 mask 255.255.255.240
  - neighbor 131.252.215.17 remote-as 100

- note: radia has IP address 215.18 and dexter has ip address 215.17 on shared 215.16/28 subnet

some bgp tricks (cisco code not included)

- 1. routing by input src



we can route packets
from net 1 to net 4,
from net 2 to net 3
based in IP src address mapping

ip src addr mapping

- questions about previous slide:
- why is such a routing policy “not normal”?
- can you perform this trick for the AS “outer mongolia”; i.e., an AS arbitrarily far away?

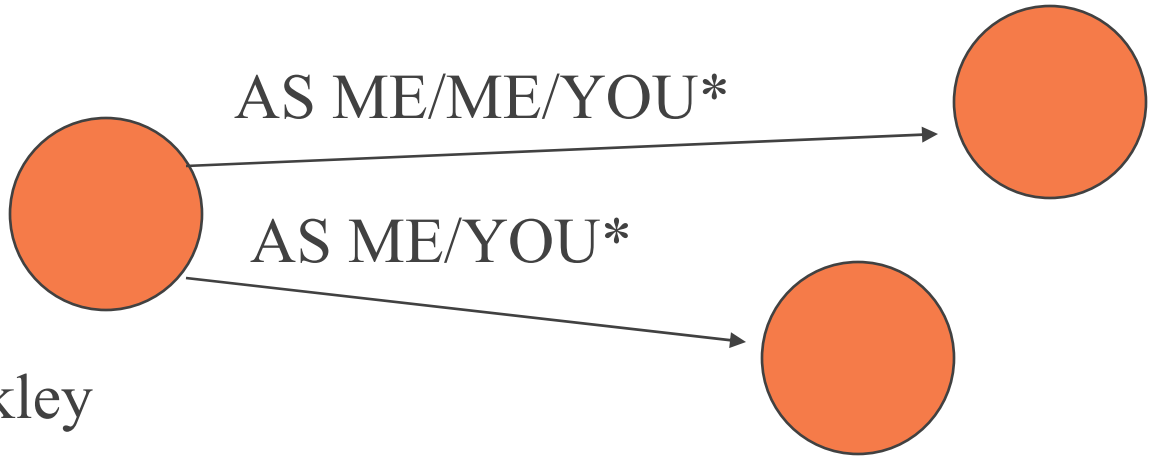
review 2 1-way paths

- inbound traffic - depends on routes YOU SEND
- outbound traffic - depends on routes YOU RECEIVE
- it may not be that hard to advertise NET1 over LINK1
  - and thus cause asymmetric routing as a form of load balancing

AS_PATH manipulation

- one possible way to influence an AS farther away
- prepend your own AS > 1 time to a path you send out
  what is consequence of this routing-wise?

- **what is consequence of this**?



load balancing ?!

- **see Halabi for his discussion**
- **define here as multiple paths at layer 3 to dst X**
- general remarks
  - possible, but remember two things
  - BGP is hop by hop - you have less knowledge of net farther from home (ahem. KISS may apply)
  - routing is two 1-way problems
  - Asymmetric routing may/may not be ok - your call
- you cannot load balance without redundancy - and asymmetric routing may be part of picture

Cisco routers

- automatically load-balance if
  - same router, two links to same IP prefix
  - what can you say about the nature of those two links? (similar bandwidth pro)
  - this info is not extended into IBGP, i.e., only one route is forwarded
  - use **maximum-paths** BGP command

hot-potato routing

- in decision process, (after EBGP over IBGP)
- we can prefer IGP (OSPF) shortest path
- this means data packet goes shortest path internally to get OUTSIDE of us
- hot-potato -> in some sense spit packets out of AS the fastest possible way

some BGP problems

- scalability of transit system with IBGP
  - and IGP issues therein
  - we covered this one already (confed/reflector)
- flapping (up/down/up/down ...)
- misconfigured junior partner
  - howzabout “routed -g” globally?
- congestion leads to TCP backoff
- security

flapping

- small fraction of routes have been known to cause many updates to "flood" BGP net
- call this "**route flap**"
  - route UP, then DOWN, then UP, DOWN, etc.
- basic idea: if path changes too fast, we will suppress sending updates about
  - aka holddown technique
  - a path may have a weight associated with it, penalized over time for more flapping
- Cisco calls anti-flapping config **route dampening**

BGP misconfigurations

- small AS could simply announce that it is MIT (BGP equivalent of routed -g ...)
  - and suck local MIT packets towards it
- April 1997, small Virginia ISP more or less announced it was Inet Center (it wasn't)
- such incidents have led to desire to sanity check and/or globally list policy
- btw: you can always use ACLs and MAPS to sanity check your (small) neighbors

Inet Routing Registry effort (www.irr.net)

- global registry in multiple distributed databases
- continues earlier RADB (www.radb.net) effort
- RIPE-181 policy language evolved now into RPSL - Routing Policy Spec. Language
  - (see RFC 2650 for examples)
- policy language describes routes/AS #s send/received by a given AS number
  - as well as POC (point of contact)
  - import from AS1 accept ANY
  - import from AS2 accept only AS2

criticisms

- garbage-in, garbage-out
  - admins may not keep up
- accept ANY isn't terribly useful
  - big ASs can however enforce check on small AS
- Bates/Bush/Rekhter/Li have suggested that routing policy be made available in DNS tree
  - could be administered locally
  - DNS could be made secure with signatures

BGP congestion and other problems

- 1997 SIGCOMM/Labovitz paper found
  - more Inet updates in BGP than needed
  - many were due to bugs in hw/sw
- 1998 study repeat found improvement but
  - possible problems due to congestion
  - TCP would backoff
  - causing BGP timer failures, reboots, lost packets, BGP update spikes, cascading failures

BGP security

- in theory, BGP marker designed for MD like MD5 or the like
- but, attack could be aimed at underlying TCP therefore we must protect TCP too
  - spoof TCP sequence number and do what?
  - DOS - send RESETs
  - or inject fake route info for MIM attack?
- protection schemes therefore?

possible fixes

- RFC 2385 - TCP option using MD5 signature
  - point is sign both TCP and BGP data
- another possibility - use IPSEC
  - possibly with AH only
  - end to end between the two peer routers, not tunnel mode

BGP and AS numbers

- how do you find AS info? e.g., using ARIN
  - # **whois -h whois.arin.net "a <number>"**
  - **note: whois –h whois.arin.net ?**
- e.g., PSU AS number
- Portland State University (ASN-PDXNET)
  - Autonomous System Name: PDXNET
  - Autonomous System Number: 6366
- as found in ARIN
- query -- see if you can find OGI AS #?
- query #2 -- what if you have an AS_PATH ... see if you can decode it; e.g., 3701/14262/11964

BGP and Inet exchange connectivity

- upstream connectivity may be defined as follows:
  - **transit** – you buy full connectivity from an ISP
    - » therefore you are an end customer usually
  - **public peering** – ISP1 and ISP2 give each other BGP info about their own customers
    - » not the Inet as a whole
    - » probably done in a public way at an exchange/NAP
  - **private peering** – at an exchange, or NAP two ISPs have a private circuit and exchange whatever they exchange

ISP Tiers

- Tier 1 – the big ISP players
  - national backbone
  - does not purchase transit
- Tier 2 –
  - national backbone
  - BUT does purchase some transit
- Tier 3
  - regional or local network
  - mostly transit, may have some peering

this implies various levels then for exchanges

- private peering in 8 US locations called the "default-free" zone
- NY, Wash DC, Atlanta, Chicago, Dallas, LA, Seattle, San Jose
- nevertheless there exist MAEs, IXPs, NAPs
  - metropolitan area exchanges
  - inet exchange points and network access points
  - these are in some sense public peering points

general study question

- BGP peering means exchange of AS information
  - large want to charge small for this of course
  - can involve lawyers, contracts, etc.
- see what you can find out about peering on the Inet
  - including structures of NATs/MAEs
  - how would you design a large peering network? (never mind the lawyers ...)

more picky study questions

- what kinds of BGP protocol messages exist?
- what are the pros/cons of using TCP as a transport?
- what security mechanisms can be used with BGP?
- explain BGP and policy - how can an AS control route dissemination?
- what is the MED attribute? what is it good for?
- what does hot-potato routing mean? really?
- why does AS_PATH protect BGP against looping?