Today's Lecture

- Scalable Addressing
  - Sub-netting
  - Super-netting (CIDR)
  - Route Aggregation Examples

- BGP
  - Global Internet routing
  - BGP protocol outline
How Is a Packet Sent

• A host bootup: use DHCP to get its own IP address, default router IP address, domain name, DNS server, etc.
• To send a packet to:” www.winlab.rutgers.edu”
  – DNS to resolve the IP address and then form the packet (more on higher layers later)
  – Check if the dest. is in the same network once an IP packet passed to the network layer
    • If the dest belongs to the same network, send to dest. directly (nexthop = dest)
    • Otherwise, send to the default router (nexthop = default router)
  – Discover nexthop’s layer 2 address (Ethernet MAC address) using ARP
  – Put IP packet in Ethernet frame (add Ethernet header)
    • dest. Ethernet addr = nexthop MAC addr, source Ethernet addr = sender MAC addr
    • No change to IP, Dest. IP addr = original IP dest., source IP addr. = original IP source
  – Ethernet switch forwards the frame toward “nexthop”, (dest host or default router) according to dest. Ethernet addr using switch forwarding table
    • Switch forwarding table is established by learning and Spanning Tree Protocol
  – In the default router, remove Ethernet header and pass the packet to IP layer, decide which output port and next hop to send based on dest IP addr using the IP forwarding table
    • Router IP forwarding table is established by Distance Vector or Link State routing protocol

| Phy layer Header | Ethernet Header | IP Header | TCP/UDP Header | Data Payload | Ethernet Trailer | Phy layer Trailer |
Scalable IP Routing
Internet Structure

Recent Past

- Stanford
- BARRNET regional
- Berkeley
- PARC
- NCAR
- UA
- UNM
- UNL
- ISU
- KU
- MidNet regional
- NSFNET backbone
- Westnet regional
Internet Structure

Today

Backbone service provider

Large corporation

"Consumer" ISP

Small corporation

Peering point

"Consumer" ISP

Large corporation

Peering point
IP Address

“class-full” addressing:

class

A 0network host 1.0.0.0 to 127.255.255.255
B 1network host 128.0.0.0 to 191.255.255.255
C 110network host 192.0.0.0 to 223.255.255.255
D 1110multicast address 224.0.0.0 to 239.255.255.255

32 bits
Distance Vector vs. Link State

Distance Vector
- A node exchanges routing info only with its directly connected neighbors
- Exchanged routing info: distance to all nodes in its routing table (everything this node has learned)
- Route computation: Distributed Bellman-Ford

Link State
- A node floods its link-state advertisement to all the nodes in the network
- Exchanged routing info: the state of the links to its directly connected links
- Route computation: Dijkstra’s algorithm
How to Make Routing Scale

- Flat versus Hierarchical Addresses
- Inefficient use of Hierarchical Address Space
  - class C with 2 hosts \((2/255 = 0.78\% \text{ efficient})\)
  - class B with 256 hosts \((256/65535 = 0.39\% \text{ efficient})\)

- Still Too Many Networks
  - routing tables do not scale
  - route propagation protocols do not scale
Subnetting

- Add another level to address/routing hierarchy: *subnet*
- *Subnet masks* define variable partition of host part
- Subnets visible only within site

<table>
<thead>
<tr>
<th>Network number</th>
<th>Host number</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111111111111111111111111</td>
<td>00000000</td>
</tr>
</tbody>
</table>

Class B address

Subnet mask (255.255.255.0)

<table>
<thead>
<tr>
<th>Network number</th>
<th>Subnet ID</th>
<th>Host ID</th>
</tr>
</thead>
</table>

Subnetted address
Subnet Example

Forwarding table at router R1

<table>
<thead>
<tr>
<th>Subnet Number</th>
<th>Subnet Mask</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.96.34.0</td>
<td>255.255.255.128</td>
<td>interface 0</td>
</tr>
<tr>
<td>128.96.34.128</td>
<td>255.255.255.128</td>
<td>interface 1</td>
</tr>
<tr>
<td>128.96.33.0</td>
<td>255.255.255.0</td>
<td>R2</td>
</tr>
</tbody>
</table>
Super-netting (CIDR)

- Class addressing doesn’t match real needs:
  - Class C is 255 addresses, too small
  - Class B is 64K addresses, too big
- Need method of allocating addresses in multiple sizes
- Assign block of contiguous network numbers to nearby networks
- Called CIDR: Classless Inter-Domain Routing
Supernetting (CIDR)

• Assign block of contiguous network numbers to nearby networks
• Called CIDR: Classless Inter-Domain Routing
• Protocol uses a (length, value) pair
  \[
  \text{length} = \# \text{ of bits in network prefix}
  \]
• Use CIDR bit mask to identify block size
• All routers must understand CIDR addressing
• Routers can aggregate routes with a single advertisement -> use longest prefix match
Supernetting (CIDR)

- Routers can aggregate routes with a single advertisement -> use longest prefix match
- Hex/length notation for CIDR address:
  - C4.50.0.0/12 denotes a netmask with 12 leading 1 bits, i.e. FF.F0.0.0
- Routing table uses “longest prefix match”
  - 171.69 (16 bit prefix) = port #1
  - 171.69.10 (24 bit prefix) = port #2
  - then DA=171.69.10.5 matches port #2
  - and DA = 171.69.20.3 matches port#1
Classless Inter Domain Routing (CIDR)

- **Problem:** Class B addresses are running out
- **Solution:** Allocate multiple Class C addresses
- **Problem:** Random allocation of Class C addresses need multiple routing table entries
- **Solution:** Allocate “contiguous” Class C addresses
- **Routing entry:** [IP Address of Network and Net Mask]

**IP Address:** 195.201.3.5 = 11000011 11001001 00000011 00000101
**Net Mask:** 254.0.0.0 = 11111110 00000000 00000000 00000000

**Network IP:** 194.0.0.0 = 11000010 00000000 00000000 00000000
Route Aggregation with CIDR

Border gateway (advertises path to 11000000000000000001)

Regional network

Corporation X (11000000000000001000001)

Corporation Y (110000000000000001000000)
 CIDR (continued)

- How many Class C addresses?

<table>
<thead>
<tr>
<th>Organization’s Requirements</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fewer than 256 addresses</td>
<td>1 Class C network</td>
</tr>
<tr>
<td>Fewer than 512 addresses</td>
<td>2 Contiguous Class C networks</td>
</tr>
<tr>
<td>Fewer than 1024 addresses</td>
<td>4 Contiguous Class C networks</td>
</tr>
<tr>
<td>Fewer than 2048 addresses</td>
<td>8 Contiguous Class C networks</td>
</tr>
<tr>
<td>Fewer than 4096 addresses</td>
<td>16 Contiguous Class C networks</td>
</tr>
<tr>
<td>Fewer than 8192 addresses</td>
<td>32 Contiguous Class C networks</td>
</tr>
<tr>
<td>Fewer than 16384 addresses</td>
<td>64 Contiguous Class C networks</td>
</tr>
</tbody>
</table>

- Contiguous Class C network addresses allow a “single” entry in the routing table for all the above organizations.
## Coordinated Address Allocation

- **Address aggregation using Geographic scope**

<table>
<thead>
<tr>
<th>Region</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-regional</td>
<td>192.0.0.0 -- 193.255.255.255</td>
</tr>
<tr>
<td>Europe</td>
<td>194.0.0.0 -- 195.255.255.255</td>
</tr>
<tr>
<td>Others</td>
<td>196.0.0.0 -- 197.255.255.255</td>
</tr>
<tr>
<td>North America</td>
<td>198.0.0.0 -- 199.255.255.255</td>
</tr>
<tr>
<td>Central/South America</td>
<td>200.0.0.0 -- 201.255.255.255</td>
</tr>
<tr>
<td>Pacific Rim</td>
<td>202.0.0.0 -- 203.255.255.255</td>
</tr>
<tr>
<td>Others</td>
<td>204.0.0.0 -- 205.255.255.255</td>
</tr>
<tr>
<td>Others</td>
<td>206.0.0.0 -- 207.255.255.255</td>
</tr>
</tbody>
</table>

- European networks will have a single entry in routing tables of routers in other continents: 
  
  
  [Network IP =194.0.0.0; mask = 254.0.0.0] 
  
  194.0.0.0 = 10110010 00000000 00000000 00000000 
  195.255.255.255 = 10110011 11111111 11111111 11111111 

  *Same 7 high-order bits implies*  
  
  Mask = 11111110 00000000 00000000 00000000 = 254.0.0.0
# Route Aggregation Examples

**Q:** How does *network* get network part of IP addr?

**A:** gets allocated portion of its provider ISP’s address space

<table>
<thead>
<tr>
<th>ISP's block</th>
<th>11001000 00010111 00010000 00000000</th>
<th>200.23.16.0/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization 0</td>
<td>11001000 00010111 00010000 00000000</td>
<td>200.23.16.0/23</td>
</tr>
<tr>
<td>Organization 1</td>
<td>11001000 00010111 00010010 00000000</td>
<td>200.23.18.0/23</td>
</tr>
<tr>
<td>Organization 2</td>
<td>11001000 00010111 00010100 00000000</td>
<td>200.23.20.0/23</td>
</tr>
<tr>
<td>...</td>
<td>….</td>
<td>….</td>
</tr>
<tr>
<td>Organization 7</td>
<td>11001000 00010111 00011110 00000000</td>
<td>200.23.30.0/23</td>
</tr>
</tbody>
</table>

Note: The table includes the IP address block for each organization, with the network part highlighted. The address space is divided into blocks, and organizations are assigned specific bits for their network addresses.
Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:

- Organization 0
  - 200.23.16.0/23
- Organization 1
  - 200.23.18.0/23
- Organization 2
  - 200.23.20.0/23
- Organization 7
  - 200.23.30.0/23

Fly-By-Night-ISP

ISPs-R-Us

"Send me anything with addresses beginning 200.23.16.0/20"

"Send me anything with addresses beginning 199.31.0.0/16"

Internet
Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1

Organization 0
- 200.23.16.0/23

Organization 2
- 200.23.20.0/23

Organization 7
- 200.23.30.0/23

Organization 1
- 200.23.18.0/23

Fly-By-Night-ISP

ISPs-R-Us

Internet

“Send me anything with addresses beginning 200.23.16.0/20”

“Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23”
Address Matching in CIDR

- Routing table uses “longest prefix match”
  - 171.69 (16 bit prefix) = routing table entry #1
  - 171.69.10 (24 bit prefix) = routing table entry #2
  - then DA=171.69.10.5 matches routing table entry #2
  - and DA = 171.69.20.3 matches routing table entry #1
CIDR (Summary)

- Continuous block of $2^N$ addresses
- [Base address, Mask]
- Lookup algorithm:
  - Masks destination address against mask in routing table entry
  - Match means route is found
  - May be multiple matchings!
  - Longest mask breaks “ties” (longest prefix match)
IP addressing (Summary)

- Classful addressing:
  - inefficient use of address space, address space exhaustion
    - e.g., class B net allocated enough addresses for 65K hosts, even if only 2K hosts in that network

- **CIDR: Classless InterDomain Routing**
  - network portion of address of arbitrary length
  - address format: `a.b.c.d/x`, where `x` is # bits in network portion of address

```
11001000 00010111 00010000 00000000
```
```
200.23.16.0/23
```
IPv6
IP Version 6

• Features
  – 128-bit addresses (classless)
  – multicast
  – real-time service
  – authentication and security
  – autoconfiguration
  – end-to-end fragmentation
  – protocol extensions

• Header
  – 40-byte “base” header
  – extension headers (fixed order, mostly fixed length)
    • fragmentation
    • source routing
    • authentication and security
    • other options
## IPv6 Technology Scope

<table>
<thead>
<tr>
<th>IP Service</th>
<th>IPv4 Solution</th>
<th>IPv6 Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addressing Range</td>
<td>32-bit, Network Address Translation</td>
<td>128-bit, Multiple Scopes</td>
</tr>
<tr>
<td>Autoconfiguration</td>
<td>DHCP</td>
<td>Serverless, Reconfiguration, DHCP</td>
</tr>
<tr>
<td>Security</td>
<td>IPSec</td>
<td>IPSec Mandated, works End-to-End</td>
</tr>
<tr>
<td>Mobility</td>
<td>Mobile IP</td>
<td>Mobile IP with Direct Routing</td>
</tr>
<tr>
<td>IP Multicast</td>
<td>IGMP/PIM/Multicast BGP</td>
<td>MLD/PIM/Multicast BGP, Scope Identifier</td>
</tr>
</tbody>
</table>
### IPv4 & IPv6 Header Comparison

<table>
<thead>
<tr>
<th>IPv4 Header</th>
<th>IPv6 Header</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>Version</td>
</tr>
<tr>
<td>IHL</td>
<td>Traffic Class</td>
</tr>
<tr>
<td>Type of Service</td>
<td>Flow Label</td>
</tr>
<tr>
<td>Total Length</td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td>Payload Length</td>
</tr>
<tr>
<td>Flags</td>
<td>Next Header</td>
</tr>
<tr>
<td>Fragment Offset</td>
<td>Hop Limit</td>
</tr>
<tr>
<td>Time to Live</td>
<td>Source Address</td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
</tr>
<tr>
<td>Header Checksum</td>
<td></td>
</tr>
<tr>
<td>Source Address</td>
<td></td>
</tr>
<tr>
<td>Destination Address</td>
<td></td>
</tr>
<tr>
<td>Options</td>
<td></td>
</tr>
<tr>
<td>Padding</td>
<td></td>
</tr>
</tbody>
</table>

**Legend**
- field’s name kept from IPv4 to IPv6
- fields not kept in IPv6
- Name & position changed in IPv6
- New field in IPv6
IPv6 Addressing

• IPv6 Addressing rules are covered by multiples RFC’s
  – Architecture defined by RFC 2373

• Address Types are :
  – Unicast : One to One (Global, Link local, Site local, Compatible)
  – Anycast : One to Nearest (Allocated from Unicast)
  – Multicast : One to Many
  – Reserved

• A single interface may be assigned multiple IPv6 addresses of any type (unicast, anycast, multicast)
  – No Broadcast Address -> Use Multicast
IPv6 Address Representation

- 16-bit fields in case insensitive colon hexadecimal representation
  - 2031:0000:130F:0000:0000:09C0:876A:130B

- Leading zeros in a field are optional:
  - 2031:0:130F:0:0:9C0:876A:130B

- Successive fields of 0 represented as ::, but only once in an address:
  - 2031:0:130F::9C0:876A:130B
  - 2031::130F::9C0:876A:130B
  - 0:0:0:0:0:0:1 => ::1
  - 0:0:0:0:0:0:0 => ::

- IPv4-compatible address representation
  - 0:0:0:0:0:192.168.30.1 = ::192.168.30.1 = ::C0A8:1E01
IPv6 Addressing

• Prefix Format (PF)Allocation
  – PF = 0000 0000 : Reserved
  – PF = 001 : Aggregatable Global Unicast Address
  – PF = 1111 1110 10 : Link Local Use Addresses (FE80::/10)
  – PF = 1111 1110 11 : Site Local Use Addresses (FEC)::/10)
  – PF = 1111 1111 : Multicast Addresses (FF00::/8)
  – Other values are currently Unassigned (approx. 7/8th of total)

• All Prefix Formats have to support EUI-64 bits Interface ID setting
  – But Multicast
Aggregatable Global Unicast Addresses

- Aggregatable Global Unicast addresses are:
  - Addresses for generic use of IPv6
  - Structured as a hierarchy to keep the aggregation
- See draft-ietf-ipngwg-addr-arch-v3-07
Address Allocation

The allocation process is under reviewed by the Registries:

- IANA allocates 2001::/16 to registries
- Each registry gets a /23 prefix from IANA
- Formerly, all ISP were getting a /35
- With the new proposal, Registry allocates a /36 (immediate allocation) or /32 (initial allocation) prefix to an IPv6 ISP
- Policy is that an ISP allocates a /48 prefix to each end customer
Hierarchical Addressing & Aggregation

- Larger address space enables:
  - Aggregation of prefixes announced in the global routing table.
  - Efficient and scalable routing.

ISP
2001:0410::/32

Customer no 1
2001:0410:0001::/48

Customer no 2
2001:0410:0002::/48

Only announces the /32 prefix

IPv6 Internet
2001::/16
**Link-Local & Site-Local Unicast Addresses**

- Link-local addresses for use during auto-configuration and when no routers are present:

  - \[1111\ 1110\ 10\ 0\] interface ID

- Site-local addresses for independence from Global Reachability, similar to IPv4 private address space:

  - \[1111\ 1110\ 11\ 0\] SLA* interface ID
Multicast Addresses (RFC 2375)

- low-order flag indicates permanent / transient group; three other flags reserved
- scope field:  1 - node local
  - 2 - link-local
  - 5 - site-local
  - 8 - organization-local
  - B - community-local
  - E - global
  -(all other values reserved)
more on IPv6 Addressing

<table>
<thead>
<tr>
<th>80 bits</th>
<th>16 bits</th>
<th>32 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000..............................0000</td>
<td>0000</td>
<td>IPv4 Address</td>
</tr>
</tbody>
</table>

IPv6 Addresses with Embedded IPv4 Addresses

<table>
<thead>
<tr>
<th>80 bits</th>
<th>16 bits</th>
<th>32 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000..............................0000</td>
<td>FFFF</td>
<td>IPv4 Address</td>
</tr>
</tbody>
</table>

IPv4 mapped IPv6 address
IPv6 Addressing Examples

LAN: 3ffe:b00:c18:1::/64

Ethernet0

interface Ethernet0
ipv6 address 2001:410:213:1::/64 eui-64

MAC address: 0060.3e47.1530

router# show ipv6 interface Ethernet0
Ethernet0 is up, line protocol is up
   IPv6 is enabled, link-local address is FE80::260:3EFF:FE47:1530
Global unicast address(es):
   2001:410:213:1::/64, subnet is 2001:410:213:1::/64
Joined group address(es):
   FF02::1:FF47:1530 FF02::1 FF02::2
MTU is 1500 bytes
Global IP Routing (BGP)
Routing in the Internet

• The Global Internet consists of Autonomous Systems (AS) interconnected with each other:
  – Stub AS: small corporation: one connection to other AS’s
  – Multihomed AS: large corporation (no transit): multiple connections to other AS’s
  – Transit AS: provider, hooking many AS’s together

• Two-level routing:
  – Intra-AS: administrator responsible for choice of routing algorithm within network
  – Inter-AS: unique standard for inter-AS routing: BGP
Internet AS Hierarchy

Inter-AS border (exterior gateway) routers

Intra-AS (interior gateway) routers
Intra-AS Routing

- Also known as **Interior Gateway Protocols (IGP)**
- Most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)
Hierarchical OSPF

- **Two-level hierarchy: local area, backbone.**
  - Link-state advertisements only in area
  - Each node has detailed area topology; only know direction (shortest path) to nets in other areas.
- **Area border routers:** “summarize” distances to nets in own area, advertise to other Area Border routers.
- **Backbone routers:** run OSPF routing limited to backbone.
- **Boundary routers:** connect to other AS’s.
Internet inter-AS routing: BGP

- **BGP (Border Gateway Protocol):** *the de facto standard*
- **Path Vector** protocol:
  - similar to Distance Vector protocol
  - each Border Gateway broadcast to neighbors (peers) *entire path* (i.e., sequence of AS’s) to destination
  - **BGP routes to networks (ASs), not individual hosts**
  - E.g., Gateway X may send its path to dest. Z:

  \[
  \text{Path} \ (X,Z) = X, Y_1, Y_2, Y_3, ..., Z
  \]
**Internet inter-AS routing: BGP**

*Suppose:* gateway X send its path to peer gateway W

- W may or may not select path offered by X
  - cost, policy (don’t route via competitors AS), loop prevention reasons.

- If W selects path advertised by X, then:
  
  \[ \text{Path (W,Z)} = w, \text{ Path (X,Z)} \]

- Note: X can control incoming traffic by controlling its route advertisements to peers:
  - e.g., don’t want to route traffic to Z -> don’t advertise any routes to Z
Q: What does a BGP router do?

- Receiving and filtering route advertisements from directly attached neighbor(s).
- Route selection.
  - To route to destination X, which path (of several advertised) will be taken?
- Sending route advertisements to neighbors.
BGP Operations (Simplified)

Establish session on TCP port 179

Exchange all active routes

Exchange incremental updates

While connection is ALIVE exchange route UPDATE messages
Inter-AS routing in the Internet: BGP

AS1 (RIP intra-AS routing)

AS2 (OSPF intra-AS routing)

AS3 (OSPF intra-AS routing)

R1

R2

R3

R4

R5
Choices

• Link state or distance vector?
  – no universal metric - policy decisions

• Problems with distance-vector:
  – Bellman-Ford algorithm may not converge

• Problems with link state:
  – metric used by routers not the same - loops
  – LS database too large - entire Internet
  – may expose policies to other AS’s
Solution: Path Vectors

• Each routing update carries the entire path

• Loops are detected as follows:
  – when AS gets route check if AS already in path
  – if yes, reject route
  – if no, add self and advertise route further

• Advantage:
  – metrics are local - AS chooses path, protocol ensures no loops
BGP Path

- Advertise the reachability of a network (length, value) in a sequence of AS’s the path traverse

- Policy
  - BGP provides capability for enforcing various policies
  - Policies are not part of BGP: they are provided to BGP as configuration information
  - BGP enforces policies by choosing paths from multiple alternatives and controlling advertisement to other AS’s
BGP Example

- Speaker for AS2 advertises reachability to P and Q
  - networks 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS2

- Speaker for backbone advertises
  - networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path (AS1, AS2).
- Speaker can cancel previously advertised paths
Path Suboptimality

3 hop red path vs 2 hop green path
Examples of BGP policies

- A multihomed AS refuses to act as transit
  - limit path advertisement
- A multihomed AS can become transit for some AS’s
  - only advertise paths to some AS’s
- An AS can favor or disfavor certain AS’s for traffic transit from itself
  - Use AS X to reach prefix p iff AS Z does not advertises reachability to p, but AS X advertises reachability to this prefix
BGP: controlling who routes to you

- A, B, C are provider networks
- X, W, Y are customer (of provider networks)
- X is dual-homed: attached to two networks
  - X does not want to route from B via X to C
  - ... so X will not advertise to B a route to C
A advertises to B the path AW
B advertises to X the path BAW
Should B advertise to C the path BAW?
  – No way! B gets no “revenue” for routing CBAW since neither W nor C are B’s customers
  – B wants to force C to route to w via A
  – B wants to route *only* to/from its customers!
Assume Y forbids T’s traffic
T cannot reach X, but X can reach T!
In fairness: could you do this “right” and still scale? Exporting internal state would dramatically increase global instability and amount of routing state.
**Interior BGP peers**

- IGP cannot propagate all the information required by BGP
- External routers in an AS use interior BGP (IBGP) connections to communicate
- External routers agree on routes and inform IGP
Interconnecting BGP peers

• BGP uses TCP to connect peers
• Advantages:
  – BGP much simpler
  – no need for periodic refresh
  – incremental updates
• Disadvantages
  – congestion control on a routing protocol?
**Hop-by-hop model**

- BGP advertises to neighbors only those routes that it uses
  - consistent with the hop-by-hop Internet paradigm
  - e.g., AS1 cannot tell AS2 to route to other AS’s in a manner different than what AS2 has chosen (need source routing for that)
Four Types of BGP Messages

- **Open**: Establish a peering session.
- **Keep Alive**: Handshake at regular intervals.
- **Notification**: Shuts down a peering session.
- **Update**: Announcing new routes or withdrawing previously announced routes.

announcement = prefix + attributes values
BGP common header

Types: OPEN, UPDATE, NOTIFICATION, KEEPALIVE
**BGP OPEN message**

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marker (security and message delineation)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Length</th>
<th>Type: open</th>
<th>version</th>
</tr>
</thead>
<tbody>
<tr>
<td>My autonomous system</td>
<td>Hold time</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BGP identifier</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Parameter length</th>
</tr>
</thead>
</table>

**Optional parameters** `<type, length, value>`

- **My AS**: id assigned to that AS
- **Hold timer**: max interval between KEEPALIVE or UPDATE messages
- **BGP ID**: address of one interface (same for all messages)
BGP UPDATE message

UPDATE message reports information on a **SINGLE** path, but can report multiple withdrawn routes.
Network Level Reachability Information

- list of IP address prefixes encoded as follows:

| Length (1 byte) | Prefix (variable) |
BGP NOTIFICATION message

Used for error notification
BGP KEEPALIVE message

Marker (security and message delineation)

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Type: KEEPALIVE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sent periodically to peers to ensure connectivity
If hold_time is zero, messages are not sent
Route Selection Summary

Highest Local Preference

Enforce relationships

Shortest AS Path
Lowest MED
i-BGP < e-BGP
Lowest IGP cost to BGP egress

traffic engineering

Lowest router ID

Throw up hands and break ties
Today’s Homework

• Peterson & Davie, Chap 4, 4th ed
  4.31, 33, 37, 40, 45

5th ed: 4.68, 4.72, 5.1, 5.3, and use Unix
*traceroute* utility to determine how many
hops it is to a few selected hosts in the
Internet.

Download and browse IPv6 and BGP RFC’s
Sources

• RFC1771: main BGP RFC
• RFC1772-3-4: application, experiences, and analysis of BGP
• RFC1965: AS confederations for BGP
• http://www.academ.com/nanog/feb1997/BGP Tutorial/sld022.htm (Cisco tutorial)