Chapter 4: Network Layer

Chapter goals:
- understand principles behind network layer services
- network layer service models
- forwarding versus routing
- basic routing models
- routing (path selection)
- dealing with scale
- advanced topics: IPv6
- instantiation, implementation in the Internet

Key Network-Layer Functions
- forwarding: move packets from router's input to appropriate router output
- routing: determine route taken by packets from source to dest.
- Routing algorithms

Interplay between routing and forwarding
Connection setup

- 3rd important function in some network architectures:
  - ATM, frame relay, X.25
- Before datagrams flow, two hosts and intervening routers establish virtual connection
- Routers get involved
- Network and transport connect service:
  - Network: between two hosts
  - Transport: between two processes

Network service model

- Q: What service model for "channel" transporting datagrams from sender to receiver?
  - Example services for individual datagrams:
    - Guaranteed delivery
    - Guaranteed delivery with less than 40 msec delay
  - Example services for flow of datagrams:
    - In-order datagram delivery
    - Guaranteed minimum bandwidth to flow
    - Restrictions on changes in inter-packet spacing

Network layer service models:

<table>
<thead>
<tr>
<th>Network Architecture Model</th>
<th>Congestion Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet bad effort</td>
<td>none</td>
</tr>
<tr>
<td>ATM CBR</td>
<td>none</td>
</tr>
<tr>
<td>ATM VBR</td>
<td>none</td>
</tr>
<tr>
<td>ATM ABR</td>
<td>none</td>
</tr>
<tr>
<td>ATM UBR</td>
<td>none</td>
</tr>
</tbody>
</table>

Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IP header addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

Network layer connection and connection-less service

- Datagram network provides network-layer connectionless service
- VC network provides network-layer connection service
- Analogous to the transport-layer services, but:
  - Service: host-to-host
  - No choice: network provides one or the other
  - Implementation: in the core

Virtual circuits

- "Source-to-dest path behaves much like telephone circuit"
  - Performance-wise
  - Network acts like source-to-dest path
- Call setup, teardown for each call before data can flow
- Each packet carries VC identifier (not destination host address)
- Every router on source-dest path maintains "state" for each outgoing connection
- Link, network resources (bandwidth, buffers) may be directed to VC
**VC implementation**

A VC consists of:
1. Path from source to destination
2. VC numbers, one number for each link along path
3. Entries in forwarding tables in routers along path
4. Packet belonging to VC carries a VC number.
5. VC number must be changed on each link.
6. New VC number comes from forwarding table.

**Virtual circuits: signaling protocols**

- Used to setup, maintain tear-down VC
- Used in ATM, frame-relay, X.25
- Not used in today's Internet

**Forwarding table**

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000-00001011 00010000</td>
<td>D11001000-00001011 00010000</td>
</tr>
<tr>
<td>11001000-00001011 00010000</td>
<td>D11001000-00001011 00010000</td>
</tr>
<tr>
<td>11001000-00001011 00010000</td>
<td>D11001000-00001011 00010000</td>
</tr>
<tr>
<td>11001000-00001011 00010000</td>
<td>D11001000-00001011 00010000</td>
</tr>
<tr>
<td>11001000-00001011 00010000</td>
<td>D11001000-00001011 00010000</td>
</tr>
<tr>
<td>otherwise</td>
<td>D11001000-00001011 00010000</td>
</tr>
</tbody>
</table>

**Datagram networks**

- No call setup at network layer
- Routers: no state about end-to-end connections
- Network-level concept of "connection"
- Packets forwarded using destination host address
- Packets between same source-dest pair may take different paths

**Longest prefix matching**

<table>
<thead>
<tr>
<th>Prefix Match</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000-00010000 00010000</td>
<td>D11001000-00010000 00010000</td>
</tr>
<tr>
<td>11001000-00010000 00010000</td>
<td>D11001000-00010000 00010000</td>
</tr>
<tr>
<td>11001000-00010000 00010000</td>
<td>D11001000-00010000 00010000</td>
</tr>
<tr>
<td>otherwise</td>
<td>D11001000-00010000 00010000</td>
</tr>
</tbody>
</table>

**Examples**

- D.A 11001000 00010110 00010000 10100001 Which interface?
- D.A 11001000 00010110 00010000 10100001 Which interface?
**Datagram or VC network: why?**

- Internet
  - data exchange among computers
  - "elastic" service, no strict timing
  - "smart" end systems (computers)
  - can adapt, perform control, error recovery
- ATM
  - evolved from telephony
  - human conversation
  - strict timing, reliability requirements
  - need for guaranteed service

- "dumb" end systems
  - simple inside network
  - complexity of "edge"

- many link types
- different characteristics
- uniform service difficult

**Chapter 4: Network Layer**

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP Internet Protocol
- 4.5 Routing algorithms

**The Internet Network layer**

Host, router network layer functions:

<table>
<thead>
<tr>
<th>Transport layer: TCP, UDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP datagram format</td>
</tr>
</tbody>
</table>

**IP datagram format**

- Protocol version (4 bits)
- Header length (4 bits)
- Type of service (8 bits)
- Total length (16 bits)
- Identification (16 bits)
- Flags (3 bits)
- Fragment offset (13 bits)

**IP Fragmentation & Reassembly**

- Networks lose the MTU (maximum transfer unit)
- Fragmentation
- Reassembly
- Each fragment has
  - original IP header
  - original IP options

**Chapter 4: Network Layer**

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP Internet Protocol
- 4.5 Routing algorithms
- 4.5 Link state protocols
- 4.6 Distance Vector
- 4.7 Broadcast and multicast routing
IP Fragmentation and Reassembly

Example

- 1400-byte datagram
- MTU = 1500 bytes

1400-byte data field

offset: MTU/8

Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What’s inside a router
- 4.4 IP Internet Protocol
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

IP Addressing: Introduction

- IP address: 32-bit identifier for host, router, interface, connection between host/router and physical link.
- Hosts typically have multiple interfaces.
- IP addresses associated with each interface.

Subnets

Recipe

- To determine the subnets, detach each interface from its host or router, creating islands of isolated networks. Each isolated network is called a subnet.

Subnets

How many?
IP addressing: C IDR

C IDR: Classless InterDomain Routing
- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is N bits in subnet portion of address

<table>
<thead>
<tr>
<th>subnet part</th>
<th>host part</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00001011 00000000 00000000</td>
<td>200.23160/23</td>
</tr>
</tbody>
</table>

IP addresses: how to get one?

Q: How does host get IP address?

- hard-coded by system admin in a file
  - Windows: control-panel-network-configuration-\ip\properties
  - UNIX: /etc/rc.config
  - DHCP: Dynamic Host Configuration Protocol
- dynamically get address from as server
- "plug and play"
- (more in next chapter)

IP addresses: how to get one?

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

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

<table>
<thead>
<tr>
<th>ISP's block</th>
<th>subnet part</th>
<th>host part</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00001011 00000000 00000000</td>
<td>200.23160/23</td>
<td></td>
</tr>
<tr>
<td>Organization 0</td>
<td>11001000 00001011 00000000 00000000</td>
<td>200.23116/23</td>
</tr>
<tr>
<td>Organization 1</td>
<td>11001000 00001011 00000000 00000000</td>
<td>200.23160/23</td>
</tr>
<tr>
<td>Organization 2</td>
<td>11001000 00001011 00000000 00000000</td>
<td>200.23160/23</td>
</tr>
<tr>
<td>Organization 3</td>
<td>11001000 00001011 00000000 00000000</td>
<td>200.23160/23</td>
</tr>
<tr>
<td>Organization 4</td>
<td>11001000 00001011 00000000 00000000</td>
<td>200.23160/23</td>
</tr>
<tr>
<td>Organization 5</td>
<td>11001000 00001011 00000000 00000000</td>
<td>200.23160/23</td>
</tr>
<tr>
<td>Organization 6</td>
<td>11001000 00001011 00000000 00000000</td>
<td>200.23160/23</td>
</tr>
</tbody>
</table>

Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information

Hierarchical addressing: more specific routes

ISP & Us has a more specific route to Organization 1

IP addressing: the last word...

Q: How does an ISP get block of addresses?

- ICANN: Internet Corporation for Assigned Names and Numbers
  - allocates addresses
  - manages DNS
  - assigns domain names, resolves disputes
NAT: Network Address Translation

Outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to NAT IP address, new port #.
   - remote clients/servers will respond to NAT IP address, new port # of destination.

Incoming datagrams: replace (NAT IP address, new port #) in data fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table.

NAT: Network Address Translation

Implementation: NAT router must:
   - outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #).
   - remote clients/servers will respond using (NAT IP address, new port #) at destination.
   - remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair.

Chapter 4: Network Layer

4.1 Introduction
4.2 Virtual circuit and datagram networks
4.3 What's inside a router
4.4 IP: Internet Protocol
   - Datagram format
   - IP address
   - ICMP
   - IPv6
4.5 Routing algorithms
   - Link state
   - Distance Vector
   - Hierarchical routing
4.6 Routing in the Internet
   - RIP
   - OSPF
   - BGP
4.7 Broadcast and multicast routing
**ICMP: Internet Control Message Protocol**

Used by hosts & routers to communicate network-level information:
- IP Error Reporting
  - Unreachible host, network, port protocol
  - Echo request/replica (used by ping)
- Network-layer "data" in IP datagram:
  - ICMP message type, code, plus first 8 bytes of IP datagram (routing error)

**Traceroute and ICMP**

1. Source sends series of UDP segments to host:
   - First has TTL-1
   - Second has TTL-2, etc.
   - Unrecognized
2. Destination replies with:
   - Unreachable packet
   - Destination unreachable ICMP message (type 3, code 3)
3. When traceroute detects no reply, sends next UDP segment.
4. When ICMP message arrives, source calculates RTT.
5. Inside router 3:
   - Trace route has 3 hops.

**IPv6**

1. Initial motivation: 32-bit address space soon to be completely allocated.
2. Additional motivation:
   - Header format allows fast processing/forwarding
   - Header changes to facilitate QoS
   - Fixed-length 40-byte header
   - No fragmentation allowed

**Chapter 4: Network Layer**

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router?
- 4.4 IP Internet Protocol
  - Datagram format
  - IP header
  - ICMP
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIPv2
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

**IPv6 Header (Cont.)**

- Priority: identifies priority among datagrams in flow
- Flow Label: identifies datagrams in same "flow" (concept of "flow" not well defined)

**Other Changes from IPv4**

- Checksum: removed entirely to reduce processing time at each hop
- Options allowed, but outside of header, indicated in "Next Header" field
- ICMPv6: new version of ICMP
  - Additional message types, e.g. "Packet Too Big"
  - Multicast group management functions
Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneously
- No "fog打工"
- How will the network operate with mixed IPv4 and IPv6 routers?
- Tunneling: IPv6 carried as payload in IPv4 datagram among IPv4 routers

Tunneling

Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
- 4.5 Routing algorithms
- 4.6 Routing in the Internet
- 4.7 Broadcast and multicast routing

Interplay between routing, forwarding

Graph abstraction
Graph abstraction: costs

\[ c(x) = \text{cost of link } (x, y) \]
\[ c(x) = 5 \]

Routing Algorithm classification

Global or decentralized information?

Global:
1. All routers have complete topology, link costs, and link cost changes.
2. "Link state" algorithms
3. "Distance vector" algorithms

Static or dynamic?

Static:
1. Routes change slowly over time
2. Routes change quickly
3. Periodic updates
4. In response to link cost changes

Routing algorithm: algorithm that finds least-cost path

Chapter 4: Network Layer

4.1 Introduction
4.2 Virtual circuit and datagram networks
4.3 What’s inside a router?
4.4 IP: Internet Protocol
4.4.1 Datagram format
4.4.2 Addressing
4.4.3 ICMP
4.4.4 IP
4.5 Routing algorithms
4.5.1 Distance Vector
4.5.2 Link state
4.5.3 Hierarchical routing
4.6 Routing in the Internet
4.6.1 RIP
4.6.2 OSPF
4.6.3 IS-IS
4.7 Broadcast and multicast routing

A Link-State Routing Algorithm

Dijkstra’s algorithm

Initialization:
1. \( N = 0 \)
2. For all nodes, \( d(v) = D(v) = \infty \)
3. For all nodes, \( p(v) = \) undefined
4. Add node 0 to \( N \)
5. Set \( d(0) = D(0) = 0 \)
6. Set \( p(0) = 0 \)

Loop:
1. Add all nodes in \( N \) such that \( D(v) = \min(\{D(w) + c(w, v) : w \notin N \}) \)
2. Update \( D(v) \) for all \( v \) adjacent to \( w \) and not in \( N \)
3. Update \( p(v) = w \) if new cost to \( w \) is either old cost to \( w \) or known cost to \( v \)
4. Add new path cost to \( w \) plus cost from \( w \) to \( v \)
5. Add all nodes to \( N \)

Dijkstra’s algorithm: example
Distance Vector Algorithm

Distance Vector Equation (Dynamic programming)

D(v) = cost of least-cost path from x to y

where mink is taken over all neighbors y of x.

Distance Vector Algorithm

For each neighbor y, maintain:
- The distance vector: D = [D(y), y, N]
- Node x knows cost to each neighbor y:
- Node x maintains D = [D(y), y, N]
- For each neighbor y, maintain:
- D(y) = estimate of least-cost from x to y

Dijkstra's algorithm, discussion

Algorithm complexity is n^2 log n. At each iteration, need to check all n - 1 neighbors. More efficient implementations are possible. (O(log n))

Dijkstra's algorithm example (2)

Bellman-Ford example

Clearly, d(2, 5) = 3, d(4, 5) = 3, d(5, 3) = 3.

5 * 3 = 4

Chapter 4: Network Layer
Distance vector algorithm (4)

Basic idea:
- Each node periodically sends its own distance vector estimate to neighbors
- When a node $x$ receives new DV estimate from neighbor, it updates its own DV using BFS equation:
  \[ D_{ij}(y) = \min \{d(x,y) + D_{ij}(y) \} \text{ for each node } y \]
- Under minor, natural conditions, the estimate $D_{ij}(y)$ converge to the actual least cost $d_{ij}(y)$

Distance Vector Algorithm (5)

Iterative, asynchronous:
- Each local thread calculates:
- 1. Local link cost change
- 2. Update routing info, recalculates distance vector
Distributed:
- Each node notifies neighbors only when its DV changes
- Neighbors then notify their neighbors if necessary

Distance Vector: link cost changes

Link cost changes:
- node detects local link cost change
- node updates routing info, recalculates DV
- if DV changes, notify neighbors

At time $t_x$, y detects link cost change, updates its DV, and informs its neighbors.

At time $t_y$, y receives the update from y and updates its table. It computes a new least cost to x and sends it to neighbors not in DV.

At time $t_y$, y receives x's update and updates its distance table. y's least cost do not change hence y does not send any message to y.

Comparison of LS and DV algorithms

Message complexity:
- LS: With $n$ nodes, $O(n^2)$ messages
- DV: Exchange between neighbors only

Robustness: what happens if router malfunctions?
- LS: No change
- DV: Node may advertise incorrect cost

Convergence time:
- LS: $O(n^2)$ algorithm, requires: $O(n^2)$ hops
- DV: Convergence time varies

Speed of Convergence:
- LS: LS has algorithm that requires: $O(\log n)$ hops
- DV: DV convergence times vary

Each node computes only its own table
- LS: May have failures
- DV: May have failures
- LS: Causes infinite problem
- DV: Causes unknown problem
- LS: Information propagates only through network
- DV: Information propagates through network
Chapter 4: Network Layer

Hierarchical Routing

Hierarchical Routing

Interconnected ASes

Example: Setting forwarding table in router Id
Example: Choosing among multiple ASes

New suppose AS1 learns from the inter-AS protocol that subnet x is reachable from AS3 and from AS2.

To configure forwarding table, router 1 must determine which gateway it should forward packets for data x.

This is also the job on inter-AS-routing protocol.

Hot potato routing: send packet towards closest of two routers.

Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP Internet Protocol
  - Datagram format
  - IP addressing
  - ICMP
  - IPv4
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

Intra-AS Routing

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

Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP Internet Protocol
  - Datagram format
  - IP addressing
  - ICMP
  - IPv4
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

RIP (Routing Information Protocol)

- Distance vector algorithm
- Included in BSD UNIX distribution in 1982
- Distance metric: # of hops (max = 15 hops)

RIP advertisements

- Distance vectors: exchanged among neighbors every 30 sec via Response Message (also called advertisement)
- Each advertisement: list of up to 25 destination nets within AS
RIP: Example

```
          .          .          .
         /           /           /
        /A          /D          /
       /           /           /
      /x          /B          /
     /           /           /
    /           /           /
   /           /           /
  /           /           /
 /           /           /
,           ,           ,

Destination Network  Next Router  Num. of hops to dest.
,                      A          2
z                      B          2
x                      B          7
w                      A          1

Routing table in D
```

RIP: Example

```
          .          .          .
         /           /           /
        /A          /D          /
       /           /           /
      /x          /B          /
     /           /           /
    /           /           /
   /           /           /
  /           /           /
 /           /           /
,           ,           ,

Destination Network  Next Router  Num. of hops to dest.
,                      A          2
z                      B          2
x                      B          7
w                      A          1

Routing table in D
```

RIP: Link Failure and Recovery

- If no advertisement heard after 180 sec --> neighbor/link declared dead
- Routes via neighbor invalidated
- New advertisements sent to neighbors
- Neighbors in turn send out new advertisements (if tables changed)
- Link failure info quickly propagates to entire net
- Poison reverse used to prevent ping pong loops (infinite distance = 16 hops)

RIP Table processing

- RIP routing tables managed by application-level process called route-d (daemon)
- Advertisements sent in UDP packets, periodically repeated

Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IP address
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

OSPF (Open Shortest Path First)

- "open" publically available
- Uses Link State algorithm
  - LS packet dissemination
  - Topology map of each node
  - Route computed using Dijkstra's algorithm
- OSPF advertisements carried one entry per neighbor
- Advertisements disseminated to entire AS (via flooding)
  - Carried in OSPF messages directly over IP (rather than TCP or UDP)
OSPF "advanced" features (not in RIP)

- Security: all OSPF messages authenticated (to prevent malicious intrusion)
- Multiple same-cost paths allowed (only one path in RIP)
- For each link, multiple cost metrics for different TOS (e.g., satellite link cost set "low" for best effort, high for real-time)
- Integrated uni- and multicast support:
  - Multicast OSPF (MOSPF) uses same topology data base as OSPF
  - Hierarchical OSPF in large domains.

Hierarchical OSPF

- Two-level hierarchy: local area, backbone.
- Link state advertisements only in areas.
- 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): de-facto standard
- BGP provides each AS a means to:
  1. Obtain subnet reachability information from neighboring ASes
  2. Propagate the reachability information to all routers internal to the AS.
  3. Determine "good" routes to subnets based on reachability information and policy.
- Allows a subnet to advertise its existence to rest of the Internet: "I am here!"

Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP Internet Protocol
  - Internet address format
  - IP addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

BGP basics

- Pair of routers (BGP peers) exchange routing info over semi-permanent TCP connections: BGP sessions.
- Notes that BGP sessions do not correspond to physical links.
- When AS2 advertises a prefix to AS1, AS2 is premising it will forward any datagrams destined to that prefix towards the prefix.
- AS2 can aggregate prefixes in its advertisement.
Distributing reachability info

- With BGP peering between 3a and 1c, AS 3a sends prefix reachability info to AS 1c.
- 1c can then use BGP to distribute this new prefix reach info to all routers in AS 1c.
- 1b can then re-advertise the new reach info to AS 2 over the 1b-to-3a eBGP session.
- When router learns about a new prefix, it creates an entry for the prefix in its forwarding table.

Path attributes & BGP routes

- When advertising a prefix, advert includes BGP attributes.
  - prefix * attributes = "route"
- Two important attributes:
  - AS-PATH: contains the ASes through which the advert for the prefix passed AS 3757 AS D
  - NEXT-HOP: Indicates the specific internal-AS router to next-hop AS. (There may be multiple paths from current AS to next-hop AS.)
- When gateway router receives route advert, uses import policy to accept/decline.

BGP route selection

- Router may learn about more than 1 route to same prefix. Router must select route.
- Elimination rules:
  1. Local preference value attribute: policy decision
  2. Shortest AS-PATH
  3. Closest NEXT-HOP router: hot potato routing
  4. Additional criteria

BGP messages

- BGP messages exchanged using TCP.
- BGP messages:
  - OPEN: opens TCP connection to peer and authenticates sender
  - UPDATE: advertises new path (or withdraws old)
  - KEEPALIVE: keeps connection alive in absence of UPDATES: also ACKs open-request
  - NOTIFICATION: reports errors in previous msg; also used to close connection

BGP routing policy

- A, B, C are provider networks
- X, Y, Z 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

BGP routing policy (2)

- 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 "return" for routing CBAW since neither W nor C are B's customers
  - B wants to force C to route to via A
  - B wants to route to A from B's customers
Why different Intra- and Inter-AS routing?

Policy:
- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed.
- Hierarchical routing saves table size, reduced update traffic.

Performance:
- Intra-AS: can focus on performance.
- Inter-AS: policy may dominate over performance.