Network Layer

This material is provided only for the use of students in CSE5231 at FIT. Many of the slides were provided with the text Computer Networking by Kurose and Ross, Addison Wesley. Classroom use is generously supported but further use or distribution may be a violation of existing copyright protection.
Definitions

- Internetwork or internet
  - An arbitrary collection of “networks” that are interconnected to provide a packet delivery service.

- Network
  - Either a direct-link network as studied earlier or a switched network that (for now) uses one type of technology (ATM, Frame Relay, Ethernet, Token Ring, etc.) to forward cells or frames.

- The Internet is the global network based on use of the Internet Protocol to route datagrams.
Simple Internetworking

Networks

TCP  IP  ETH

TCP  IP  ETH

IP  ETH  FDDI

IP  FDDI  ATM

IP  ATM  ETH
## Network layer service models:

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Service Model</th>
<th>Guarantees?</th>
<th>Congestion feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Bandwidth</td>
<td>Loss</td>
</tr>
<tr>
<td>Internet</td>
<td>best effort</td>
<td>none</td>
<td>no</td>
</tr>
<tr>
<td>ATM</td>
<td>CBR</td>
<td>constant rate</td>
<td>yes</td>
</tr>
<tr>
<td>ATM</td>
<td>VBR</td>
<td>guaranteed rate</td>
<td>yes</td>
</tr>
<tr>
<td>ATM</td>
<td>ABR</td>
<td>guaranteed minimum</td>
<td>no</td>
</tr>
<tr>
<td>ATM</td>
<td>UBR</td>
<td>none</td>
<td>no</td>
</tr>
</tbody>
</table>

- Internet model being implemented: Intserv, Diffserv
Datagram networks: the Internet model

- datagram forwarding is sometimes called “routing”
- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of “connection”
- packets typically routed using destination host ID
  - packets between same source-dest pair may take different paths
IP Addressing: introduction

- **IP address**: 32-bit identifier for host, router interface

- **interface**: connection between host, router and physical link
  - router’s typically have multiple interfaces
  - host may have multiple interfaces (usually one)
  - IP addresses associated with interface, not host, router

223.1.1.1 = 11011111 00000001 00000001 00000001

Network Analysis: Internetworking 4 - 6
**IP Addressing**

- **IP address:**
  - network part (high order bits)
  - host part (low order bits)

- **What's a network?**
  (from IP address perspective)
  - device interfaces having same network part of IP address
  - can physically reach each other without intervening router

network consisting of 3 IP networks
(for IP addresses starting with 223, first 24 bits are network address)
IP Addressing

How to find the networks?

- Detach each interface from router, host
- Create "islands of isolated networks"

Interconnected system consisting of six networks
ICANN: Internet Corp for Assigned Names and Numbers

Where network numbers originate:
- Assigns network IDs based on organization size.
- **Class A**: Large Organizations
- **Class B**: Medium-sized Organizations
- **Class C**: Everyone else
ICANN Domain Names

To reach another person on the Internet you have to type an address into your computer - a name or a number. That address has to be unique so computers know where to find each other. ICANN coordinates these unique identifiers across the world. Without that coordination we wouldn't have one global Internet. ICANN is responsible for the global coordination of the Internet's system of unique identifiers like domain names (like .org, .museum and country codes like .uk) and the addresses used in a variety of Internet protocols that help computers reach each other over the Internet.

So, what has ICANN done in the past decade?

- Back in 1998, there was a single registrar, charging $50 a year for domain names; now there are over 900 ICANN-accredited registrars and a domain costs from just $6.
- Helped the domain name system grow from roughly three million domains a decade ago to over 160 million today.
- Expanded the Internet's generic top-level domains from three (dotcom, dotnet and dotorg) to 16, including .info, .biz, .cat, .asia, .mobi and .name.
- Seen over 35,000 domains go through the Uniform Dispute Resolution Process, a faster, cheaper and more efficient alternative to the law courts for ownership disputes.
- Developed policies with the full involvement of governments, business, the technical community and individual Net users that make the Internet's addressing system able to adapt to the radical new uses that the network is put to every year.
IP Addresses

given notion of “network”, let’s examine IP addresses:

class

A 0network host 1.0.0.0 to 127.255.255.255
B 10 network host 128.0.0.0 to 191.255.255.255
C 110 network host 192.0.0.0 to 223.255.255.255
D 1110 multicast address 224.0.0.0 to 239.255.255.255

32 bits
Base Number of Available Host IDs

- **Class A:** \(2^{24} - 2 = 16,777,214\).
- **Class B:** \(2^{16} - 2 = 65,534\).
- **Class C:** \(2^8 - 2 = 254\).
How routers determine next hop

- Determine best match in the routing table (longest prefix match)
- If router does not have an interface directly on the (sub)network, it forwards packet to next router.
- If router does have an interface directly on the destination (sub)network, it forwards packet directly to destination host.
- Router uses “subnet mask” to determine which bits it should use in matching the destination address to the routing table entry.
Subnets and Supernets

- **Subnets:**
  - A network is divided into several smaller subnets
  - Each subnet has its own subnet address.

- **Supernets:**
  - Organization may combine several class C network addresses to create a supernetwork
  - Permits a larger number of host addresses
Subnetting

- Classes A, B, C designed with two level hierarchy in mind
  - Network ID
  - Host ID
- Consider UCF with Class B Network address: 132.170.0.0
  - Without subnet idea only a single flat network with at most $2^{16} - 2 = 65,534$ host addresses.
- Imagine the router tables!

- To fix this problem we introduce a 3-level hierarchical IDs:
  - Network ID
  - Subnet ID
  - Host ID
- Use “subnet mask” of all 1’s covering Network ID plus Subnet ID
  - network or subnet address results from bit-wise “and” operation on subnet mask plus IP address.
Subnet Example
Routing Table Comparisons

Table Entry

Destination Address
- In this example, the route table entry/mask does not match destination address and router examines next entry.
- If dest@ = 140.252.1.5, the router will determine there is a match.
- Router will forward based on the longest such match.
- If the router has an interface directly on the (sub)network, the table entry must instruct the router to send through that interface directly to the destination.
- If the router has no interface directly on the (sub)network, the table entry must instruct the router to send to the next router.
- There MUST be a matching entry (or default entry) in the table.
Getting a datagram from source to dest.

IP datagram:

- datagram remains unchanged, as it travels source to destination
- addr fields of interest here
Getting a datagram from source to dest.

Starting at A, given IP datagram addressed to B:
- look up net. address of B
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
  - B and A are directly connected

<table>
<thead>
<tr>
<th>Dest. Net.</th>
<th>next router</th>
<th>Nhops</th>
</tr>
</thead>
<tbody>
<tr>
<td>223.1.1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>223.1.2</td>
<td>223.1.1.4</td>
<td>2</td>
</tr>
<tr>
<td>223.1.3</td>
<td>223.1.1.4</td>
<td>2</td>
</tr>
</tbody>
</table>

misc fields | 223.1.1.1 | 223.1.1.3 |

data

misc fields | 223.1.1.1 | 223.1.1.3 |

data
### Getting a datagram from source to dest.

Starting at A, dest. E:
- look up network address of E
- E on *different* network
  - A, E not directly attached
- routing table: next hop router to E is 223.1.1.4
- link layer sends datagram to router 223.1.1.4 inside link-layer frame
- datagram arrives at 223.1.1.4
- continued....

<table>
<thead>
<tr>
<th>misc fields</th>
<th>223.1.1.1</th>
<th>223.1.2.2</th>
<th>data</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Dest. Net.</th>
<th>next router</th>
<th>Nhops</th>
</tr>
</thead>
<tbody>
<tr>
<td>223.1.1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>223.1.2</td>
<td>223.1.1.4</td>
<td>2</td>
</tr>
<tr>
<td>223.1.3</td>
<td>223.1.1.4</td>
<td>2</td>
</tr>
</tbody>
</table>

Network Analysis: Internetworking
Getting a datagram from source to dest.

Arriving at 223.1.4, destined for 223.1.2.2

- look up network address of E
- E on *same* network as router’s interface 223.1.2.9
  - router, E directly attached
- link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9
- datagram arrives at 223.1.2.2!!! (hooray!)
How routers determine next hop

- Determine best match in the routing table (longest prefix match)
- If router does not have an interface directly on the (sub)network, it forwards packet to next router.
- If router does have an interface directly on the destination (sub)network, it forwards packet directly to destination host.
- Router uses “subnet mask” to determine which bits it should use in matching the destination address to the routing table entry.
Create a routing table for the router shown below. The table should contain only 3 rows that include network address, subnet mask, and next. It should be sufficient so that each workstation shown can communicate with all the other workstations shown.
# Routing Table

<table>
<thead>
<tr>
<th>Network</th>
<th>Subnet Mask</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>130.25.18.0</td>
<td>255.255.255.0</td>
<td>Eth0</td>
</tr>
<tr>
<td>40.0.0.0</td>
<td>255.0.0.0</td>
<td>Eth1</td>
</tr>
<tr>
<td>130.46.0.0</td>
<td>255.255.0.0</td>
<td>Eth2</td>
</tr>
</tbody>
</table>
Assign IP addresses to Eth0-Eth2. Using these and the IP addresses shown in the diagram, create a routing table for the upper left router. The routing table must contain exactly five rows with IP address, subnet mask, next and be sufficient to support communication among all IP addresses shown.
# Routing Table

<table>
<thead>
<tr>
<th>Network</th>
<th>Subnet Mask</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.5.10.0</td>
<td>255.255.255.0</td>
<td>Eth0/100.5.10.1</td>
</tr>
<tr>
<td>100.5.20.0</td>
<td>255.255.255.0</td>
<td>Eth1/100.5.20.1</td>
</tr>
<tr>
<td>130.5.0.0</td>
<td>255.255.0.0</td>
<td>Eth2/130.5.10.1</td>
</tr>
<tr>
<td>90.60.1.0</td>
<td>255.255.255.0</td>
<td>Eth3</td>
</tr>
<tr>
<td>92.0.0.0</td>
<td>255.0.0.0</td>
<td>90.60.1.1</td>
</tr>
</tbody>
</table>
Subnets NOT on byte boundary

Example: Class C Subnet

- An organization with a Class C address has one physical network available with 254 hosts.
- What if it needs 6 subnetworks?
  - Use 3 bits to cover subnet ID in the subnet mask (enables 6 subnets)
  - Leaves 5 bits for hosts per subnet or 30 host IDs.
- Example: Network address: 223.50.20.0
  - subnet mask 255.255.255.224 or 11111111 11111111 11111111 11100000
  - preferred subnet IDs: 32, 64, 96, 128, 160, 192
  - preferred subnet addresses:
    - 223.50.20.32 (host @: 223.50.20.33 – 223.50.20.62)
    - 223.50.20.64 (host @: 223.50.20.65 – 223.50.20.94)
    - 223.50.20.96 (host @: 223.50.20.97 – 223.50.20.126)
    - 223.50.20.128 (host @: 223.50.20.129 – 223.50.20.158)
    - 223.50.20.160 (host @: 223.50.20.161 – 223.50.20.190)
    - 223.50.20.192 (host @: 223.50.20.193 – 223.50.20.222)
## Determining subnet IDs (cont.)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>Avoid</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td>96</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>128</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td>160</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td>192</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>Avoid</td>
</tr>
</tbody>
</table>
Supernets

- Mostly needed because Class A and Class B addresses exhausted
- Can still get Class C but 254 IDs for subnet/host may not suffice.
- Solution: Supernetting
  - Organization gets a block of 4 Class C addresses
  - Uses these in one “supernetwork.”
- Example: Assigned Class C’s 223.50.20.x - 223.50.23.x
  - supernet mask: 11111111 11111111 11111100 00000000
    (255.255.252.0)
  - This gives organization 1024 - 2 = 1022 host IDs internally.
- Router may list only 223.50.20.0 in its table:
  - If destination address matches 223.50.20 after applying the mask, then destination is in the supernet.
- Example: 223.50.22.87
  - Last two bytes 00010110 01011011
  - After mask 00010100 00000000 (match 223.50.20.0)
  - Allows HostID to be interpreted as 10 01011011 = 599.
- Routing technique is called Classless Inter-Domain Routing (CIDR).
Alternative address format

address format: \texttt{a.b.c.d/x}, where \( x \) is \# bits in network portion of address
IP addresses: how host gets one?

Hosts (host portion):
- hard-coded by system admin in a file
- **DHCP**: Dynamic Host Configuration Protocol:
  - dynamically get address: “plug-and-play”
    - host broadcasts “DHCP discover” msg
    - DHCP server responds with “DHCP offer” msg
    - host requests IP address: “DHCP request” msg
    - DHCP server sends address: “DHCP ack” msg
IP addresses: how to get one?

Network (network portion):

- get allocated portion of 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>
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

"Send me anything with addresses beginning 200.23.16.0/20"

ISPs-R-Us

"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

"Send me anything with addresses beginning 200.23.16.0/20"

ISPs-R-Us

"Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23"

Internet
IP addressing: the last word...

Q: How does an ISP get block of addresses?
A: ICANN: Internet Corporation for Assigned Names and Numbers
- allocates addresses
- manages DNS
- assigns domain names, resolves disputes
**ARP Example: Local Destination**

- Host E1 wants to send to mary@eagle.cs.uni.edu
- Lookup done by DNS to get 192.31.65.5.
- E1 broadcasts only on local LAN the ARP request: "Who owns 192.31.65.5?"
- Each NIC will accept the broadcast on MAC and check it's IP address.
- Only E2 responds with its MAC @.
- E1 then puts the original IP datagram inside Ethernet Frame addressed to MAC @ of E2.
- Optimizations: (a) E1 stores info for a time. (b) E1 includes own IP/MAC map in ARP and all stations can copy. (c) Each machine can ARP itself when boots and all stations can copy.
ARP Example (distant destination)

- Host E1 wants to send to E6.
- **ARP PROXY:**
  - CS Router E3 configured to respond to all ARP requests for 192.31.63.0 with its own IP @ 192.31.65.1.
  - E1 then caches 192.31.65.1/E6.
- **Simpler:**
  - E1 sees that E6 on distant subnet and it is configured to send all such traffic to E3.
- **Either way:** IP packet is sent to E3 and E3 looks up the destination IP address.
  - Routing table says send to 192.31.60.7. E3 will ARP on FDDI if necessary to get MAC @.
  - Router E4 receives and looks up destination address in routing table to get hostid. Note that E4 can also send ARP request on the EE Ethernet if necessary to get E6 Mac @.
### IP datagram format

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP protocol version number</td>
<td>Version of the IP protocol.</td>
</tr>
<tr>
<td>Header length</td>
<td>Length of the header in 32-bit words.</td>
</tr>
<tr>
<td>“Type” of data</td>
<td>Type of data being transmitted.</td>
</tr>
<tr>
<td>Max number remaining hops</td>
<td>Number of hops remaining before discarding the datagram.</td>
</tr>
<tr>
<td>Time to live</td>
<td>Time remaining before discarding the datagram.</td>
</tr>
<tr>
<td>Upper layer</td>
<td>Layer of the upper protocol.</td>
</tr>
<tr>
<td>Internet checksum</td>
<td>Checksum for the datagram.</td>
</tr>
<tr>
<td>32 bit source IP address</td>
<td>Source IP address for the datagram.</td>
</tr>
<tr>
<td>32 bit destination IP address</td>
<td>Destination IP address for the datagram.</td>
</tr>
<tr>
<td>Options (if any)</td>
<td>Additional options for the datagram.</td>
</tr>
<tr>
<td>Data</td>
<td>Payload data (variable length, typically a TCP or UDP segment).</td>
</tr>
</tbody>
</table>

**Notes:**
- **Ver**: 32 bits
- **Header length**: 32 bits
- **Total datagram + hdr length (bytes)**
- **Type of service**
- **Fragmentation/reassembly**
- **Upper layer protocol**
- **E.g. timestamp, record route taken, specify list of routers to visit.**
IP Fragmentation & Reassembly

- network links have MTU (max.transfer unit) - max data that a frame can carry
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments

fragmentation:
in: one large datagram
out: 3 smaller datagrams
### IP Fragmentation and Reassembly

One large datagram becomes several smaller datagrams

<table>
<thead>
<tr>
<th>length</th>
<th>ID</th>
<th>fragflag</th>
<th>offset</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>1480</td>
</tr>
<tr>
<td>1500</td>
<td>x</td>
<td>1</td>
<td>185</td>
<td>1480</td>
</tr>
<tr>
<td>1040</td>
<td>x</td>
<td>1</td>
<td>370</td>
<td>1020</td>
</tr>
</tbody>
</table>

MTU = 1500

Total length Of datagram. Data = 3980.

185 × 8 = 1480

370 × 8 = 2960
The Internet Network layer

Host, router network layer functions:

- **Routing protocols**
  - path selection
  - RIP, OSPF, BGP

- **IP protocol**
  - addressing conventions
  - datagram format
  - packet handling conventions

- **ICMP protocol**
  - error reporting
  - router “signaling”
ICMP: Internet Control Message Protocol

- used by hosts, routers, gateways to communication network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer “above” IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>echo reply (ping)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>dest. network unreachable</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>dest host unreachable</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>dest protocol unreachable</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>dest port unreachable</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>dest network unknown</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>dest host unknown</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>redirect (host change route)</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>echo request (ping)</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>route advertisement</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>router discovery</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>TTL expired</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>bad IP header</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>traceroute</td>
</tr>
</tbody>
</table>
Routing in the Internet

- The Global Internet consists of Autonomous Systems (AS) interconnected with each other:
  - **Stub AS**: small corporation
  - **Multihomed AS**: large corporation (no transit)
  - **Transit AS**: provider

- Two-level routing:
  - **Intra-AS**: administrator is responsible for choice
  - **Inter-AS**: unique standard
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 IGPs:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco propr.)
- IGPs use a routing metric to choose an optimal path: administrative cost, hop count, throughput, delay, ...
Next-hop Routing Table Basics

IP Routing Tables Usually Have 7 Fields:
1. **Mask**: applied to dest IP@ to find network/subnet address of dest.
2. Either destination host-specific or network-specific address.
3. **Next hop address**: IP@ of next-hop router.
4. **Flags**:
   - **U** - router is up
   - **G** - destination is another network (not this one)
   - **H** - destination is a host-specific address
   - **D** - destination added by ICMP redirection
   - **M** - destination modified by ICMP redirection
5. **Reference count**: number of users using route at this moment
6. **Use**: Number of packets transmitted to destination by this router
7. **Interface**: Name of the network interface
Routing Table example (continued)

- Three attached class C networks (LANs)
- Router only knows routes to attached LANs
- Default router used for other destinations
- Route multicast address: 224.0.0.0
- Loopback interface (for debugging)
Routing

Routing protocol

Goal: determine “good” path (sequence of routers) thru network from source to dest.

Graph abstraction for routing algorithms:

- graph nodes are routers
- graph edges are physical links
  - link cost: delay, $ cost, or congestion level

“good” path:
  - typically means minimum cost path
  - other def’s possible
Routing Algorithm classification

Global or decentralized information?

Global:
- all routers have complete topology, link cost info
- “link state” algorithms

Decentralized:
- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- “distance vector” algorithms

Static or dynamic?

Static:
- routes change slowly over time

Dynamic:
- routes change more quickly
  - periodic update
  - in response to link cost changes
Routing

Routing protocol

Goal: determine “good” path (sequence of routers) thru network from source to dest.

Graph abstraction for routing algorithms:

- graph nodes are routers
- graph edges are physical links
  - link cost: delay, $ cost, or congestion level

“good” path:
  - typically means minimum cost path
  - other def’s possible
Major Routing Algorithm Types

- Link-State Routing
- Distance-Vector Routing
- Policy-Based Routing
A Link-State Routing Algorithm

Dijkstra’s algorithm
- net topology, link costs known to all nodes
  - accomplished via “link state broadcast”
  - all nodes have same info
- computes least cost paths from one node (‘source”) to all other nodes
  - gives routing table for that node
- iterative: after k iterations, know least cost path to k dest.’s

Notation:
- $c(i,j)$: link cost from node i to j. cost infinite if not direct neighbors
- $D(v)$: current value of cost of path from source to dest. V
- $p(v)$: predecessor node along path from source to v, that is next v
- $N$: set of nodes whose least cost path definitively known
### Dijkstra's algorithm: example

<table>
<thead>
<tr>
<th>Step</th>
<th>start N</th>
<th>D(B),p(B)</th>
<th>D(C),p(C)</th>
<th>D(D),p(D)</th>
<th>D(E),p(E)</th>
<th>D(F),p(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A</td>
<td>2, A</td>
<td>5, A</td>
<td>1, A</td>
<td>infinity</td>
<td>infinity</td>
</tr>
<tr>
<td>1</td>
<td>AD</td>
<td>2, A</td>
<td>4, D</td>
<td></td>
<td>2, D</td>
<td>infinity</td>
</tr>
<tr>
<td>2</td>
<td>ADE</td>
<td>2, A</td>
<td>3, E</td>
<td></td>
<td></td>
<td>4, E</td>
</tr>
<tr>
<td>3</td>
<td>ADEB</td>
<td></td>
<td>3, E</td>
<td></td>
<td></td>
<td>4, E</td>
</tr>
<tr>
<td>4</td>
<td>ADEBC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4, E</td>
</tr>
<tr>
<td>5</td>
<td>ADEBCF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note after selecting node D in step 0, do not bring its entry down to step 1.
Dijkstra’s Algorithm, Discussion

Algorithm complexity: n nodes

- Max of n-1 iterations (add one node to N each iteration)
  - Less than n-1 compares to choose smallest label each iteration.
- Max of one addition + 1 compare (=2) to update each label implies max of 2[(n-1)+...+1] operations to update all labels not in N. (n)(n-1) operations.

\[ \Rightarrow O \left( n^2 \right) \]

Oscillations possible:
- e.g. link cost = amount of carried traffic

![Diagram](initially) initially... recompute routing... recompute... recompute
Distance Vector Routing

Each router maintains a table of:
- Best known distance to each destination router
- Which link to use to go there

Router updates table by exchanging information with its neighbors
Each sends a vector of distances; hence, "distance vector" routing
- How often?
- Which metric? (queue length, delay, hops)
- Must metric be measured?
Distance Vector Routing Algorithm

iterative:
- continues until no nodes exchange info.
- *self-terminating*: no "signal" to stop

asynchronous:
- nodes need *not* exchange info/iterate in lock step!

distributed:
- each node communicates *only* with directly-attached neighbors

Distance Table data structure
- each node has its own row for each possible destination column for each directly-attached neighbor to node
- example: in node X, for dest. Y via neighbor Z:
Distance Vector Table at Rtr J

(a) Router

(b) Vectors received from J's four neighbors

New estimated delay from J

<table>
<thead>
<tr>
<th>To</th>
<th>A</th>
<th>I</th>
<th>H</th>
<th>K</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>24</td>
<td>20</td>
<td>21</td>
<td>8</td>
</tr>
<tr>
<td>A</td>
<td>12</td>
<td>36</td>
<td>31</td>
<td>28</td>
<td>20</td>
</tr>
<tr>
<td>B</td>
<td>25</td>
<td>18</td>
<td>19</td>
<td>36</td>
<td>28</td>
</tr>
<tr>
<td>C</td>
<td>40</td>
<td>27</td>
<td>8</td>
<td>24</td>
<td>20</td>
</tr>
<tr>
<td>D</td>
<td>14</td>
<td>7</td>
<td>30</td>
<td>22</td>
<td>17</td>
</tr>
<tr>
<td>E</td>
<td>23</td>
<td>20</td>
<td>19</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>F</td>
<td>18</td>
<td>31</td>
<td>6</td>
<td>31</td>
<td>18</td>
</tr>
<tr>
<td>G</td>
<td>17</td>
<td>20</td>
<td>0</td>
<td>19</td>
<td>12</td>
</tr>
<tr>
<td>H</td>
<td>21</td>
<td>0</td>
<td>14</td>
<td>22</td>
<td>10</td>
</tr>
<tr>
<td>I</td>
<td>21</td>
<td>11</td>
<td>7</td>
<td>22</td>
<td>6</td>
</tr>
<tr>
<td>J</td>
<td>24</td>
<td>22</td>
<td>9</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>K</td>
<td>29</td>
<td>33</td>
<td>9</td>
<td>9</td>
<td>15</td>
</tr>
</tbody>
</table>

New routing table for J

Network Analysis: Internetworking 4 - 57
Distance Vector Routing: overview

Iterative, asynchronous:
- each local iteration caused by:
  - local link cost change
  - message from neighbor: its least cost path change from neighbor

Distributed:
- each node notifies neighbors only when its least cost path to any destination changes
  - neighbors then notify their neighbors if necessary

Each node:
1. wait for (change in local link cost or msg from neighbor)
2. recompute distance table
3. if least cost path to any dest has changed, notify neighbors
Comparison of LS and DV algorithms

Message complexity
- **LS:** with \(n\) nodes, \(E\) links, \(O(nE)\) msgs sent each
- **DV:** exchange between neighbors only
  - convergence time varies

Speed of Convergence
- **LS:** \(O(n^{\text{**2}})\) algorithm requires \(O(nE)\) msgs
  - may have oscillations
- **DV:** convergence time varies
  - may be routing loops
  - count-to-infinity problem

Robustness: what happens if router malfunctions?
- **LS:**
  - node can advertise incorrect *link* cost
  - each node computes only its *own* table

- **DV:**
  - DV node can advertise incorrect *path* cost
  - each node's table used by others
    - error propagate thru network
Hierarchical Routing

Our routing study thus far - idealization
- all routers identical
- network “flat”
... not true in practice

scale: with 50 million destinations:
- can’t store all dest’s in routing tables!
- routing table exchange would swamp links!

administrative autonomy
- internet = network of networks
- each network admin may want to control routing in its own network
Hierarchical Routing

- aggregate routers into regions, “autonomous systems” (AS)
- routers in same AS run same routing protocol
  - “intra-AS” routing protocol
  - routers in different AS can run different intra-AS routing protocol

gateway routers

- special routers in AS
- run intra-AS routing protocol with all other routers in AS
- also responsible for routing to destinations outside AS
  - run inter-AS routing protocol with other gateway routers
Intra-AS and Inter-AS routing

Gateways:
- perform inter-AS routing amongst themselves
- perform intra-AS routers with other routers in their AS

inter-AS, intra-AS routing in gateway A.c

network layer
link layer
physical layer

Routing Table:
- DL
- PHY
- to/from A.b
- to/from A.d
- to/from B.a
Intra-AS and Inter-AS routing

- We’ll examine specific inter-AS and intra-AS Internet routing protocols shortly
RIP (Routing Information Protocol)

- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops)
- Distance vectors: exchanged every 30 sec via Response Message (also called advertisement)
- Each advertisement: route to up to 25 destination nets
RIP (Routing Information Protocol)

<table>
<thead>
<tr>
<th>Destination Network</th>
<th>Next Router</th>
<th>Num. of hops to dest.</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>y</td>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>z</td>
<td>B</td>
<td>7</td>
</tr>
<tr>
<td>x</td>
<td>--</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Routing information 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
RIP Table processing

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

- **Core**
  - Switching
  - Full Routing Information

- **Distribution Layer**
  - Aggregation
  - Summarization
  - Partial Routing Information

- **Access Layer**
  - Network Entry Control
  - Edge Services
  - Minimal Routing Information
The Right Topology: 3 Layers

- Network core forwards packets at high speeds
- Distribution layer summarizes routes and aggregates traffic
- Access layer feeds traffic into the network and performs other edge services

Guiding principles:
- The area affected by a topology change should be as small as possible.
- Routers should contain the least information possible.
Address Summarizing

TABLE BEFORE
...
10.0.10.0/24 Wic/0
10.0.20.0/24 Wic/0
...

TABLE AFTER
...
10.0.20.0/22 Wic/0
...

Distribution Router

Access Router

Address changes

10.0.10.10/24
10.0.10.0/24
10.0.10.20/24
10.0.20.10/24
10.0.20.0/24
10.0.20.10/24
10.0.20.20/24
10.0.20.20/24
Packet to 10.0.21.20 Arrives at Distribution Router

- Apply subnet mask of
  11111111.11111111.11111100.00000000
  to destination address
  00001010.00000000.00010101.00010100
  - Obtain 10.0.20.0
- Matches distribution router table entry for Wic/0
- Access router routes properly using
  24-bit subnet mask.
OSPF (Open Shortest Path First)

- “open”: publicly available
- Uses Link State algorithm
  - LS packet dissemination
  - Topology map at each node
  - Route computation using Dijkstra’s algorithm

- OSPF advertisement carries one entry per neighbor router
- Advertisements disseminated to entire AS (via flooding)
OSPF “advanced” features (not in RIP)

- **Security**: all OSPF messages authenticated (to prevent malicious intrusion); TCP connections used
- **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 database as OSPF
- **Hierarchical** OSPF in large domains.
Hierarchical OSPF (single AS)
Hierarchical OSPF

- Two-level hierarchy: local area, backbone.
  - Link-state advertisements only in area
  - Each node has detailed area topology; only knows 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 ASs.
IGRP (Interior Gateway Routing Protocol)

- CISCO proprietary; successor of RIP (mid 80s)
- Distance Vector, like RIP
- several cost metrics (delay, bandwidth, reliability, load etc)
- uses TCP to exchange routing updates
- Loop-free routing via Distributed Updating Alg. (DUAL) based on diffused computation
Inter-AS routing
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 ASs) to destination
  - 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 sends 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:
  \[
  \text{Path (W,Z)} = \text{Path(W,X)}, \text{Path (X,Z)}
  \]
- Note: X can control incoming traffic by filtering its route advertisements to peers:
  - e.g., don’t want to route traffic to Z -> don’t advertise any routes to Z
Internet inter-AS routing: BGP

- 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
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

Scale:
- Hierarchical routing saves table size, reduced update traffic

Performance:
- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance
IP over ATM: Two Methods

- Classical IP over ATM
  - Relies on an ARP server (which every node must be able to find) to resolve IP addresses to ATM addresses.

- LAN Emulation
  - Relies on servers that provide a broadcast service on what is really a point-to-point network.
Classical IP over ATM

- Standards define the Logical IP Subnet (LIS).
  - One large ATM network may be divided into several smaller LISes.
  - All nodes on same LIS have same IP network number (network portion of IP address).
  - Two nodes in same LIS communicate directly over the ATM network.
  - Two nodes in different LISes communicate using a router.
ATM ARP Server

- Each node in a LIS must be configured with the ATM address of an ATMARP server.
- When node boots, it creates a vc to the ATMARP server.
- Booting node then registers its ATM and IP addresses with the server.
- Connection within LIS? Get ATM address from ARP server and connect directly.
- Connection outside LIS? Forward to a router that is on the LIS.
LAN Emulation over ATM

- Recall that the original ARP over Ethernet relied on the use of the Ethernet broadcast to resolve its IP addresses to Mac addresses.
- A number of computer companies decided that it would ease the transition to ATM if ATM could emulate Ethernet and Token Ring interfaces including the broadcast mechanisms.
Implementing LANE

- Participating ATM devices must have both ATM addresses and 48 bit MAC addresses.
  - Devices connecting to the ATM network are called Lan emulation clients or LECs.
- Three servers are required:
  - Lan emulation configuration server (LECS)
  - Lan emulation server (LES)
  - The broadcast and unknown server (BUS)
**LANE Configuration**

- Each LE client must be configured with the atm address of the LECS and connect to it when it boots.
- **Client gives its ATM address to the LECS.**
- LECS provides type of LAN emulated, max packet size, ATM address of the LES. One LECS may manage several LANEs.
- **Client registers its ATM and MAC addresses with the LES and LES gives client the ATM address of the BUS.**
Data delivery in LANE

- Client sends any broadcast traffic to the BUS.
  - In particular it can now ARP as usual to obtain MAC addresses that correspond to destination IP address (or router address).

- For unicast traffic client sends first packet to BUS along with an ATM address resolution request to the LES asking for ATM address that corresponds to destination MAC address.

- LES returns the dest ATM address and subsequent traffic is sent on a VC set up directly to the destination.
  - Care must be taken to avoid out of order delivery with first frame sent to BUS.
  - VCs are maintained for some period of time using a caching algorithm.
**IPv6**

- **Initial motivation:** 32-bit address space completely allocated by 2008.

- **Additional motivation:**
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS
  - new “anycast” address: route to “best” of several replicated servers

- **IPv6 datagram format:**
  - fixed-length 40 byte header
  - no fragmentation allowed
V4 and V6 Address Comparison

- **IPV4 address:** 203.178.141.220
  - Supports about 4 billion IP addresses
  - Envisioned for computers

- **IPV6 address (128 bits):**
  - Supports about $3.4 \times 10^{38}$ addresses
  - Cellphones, cars, home appliances

- **Observation:** Per Cent of IPV4 address space:
  - North America - 72%
  - Europe - 17%
  - Asia/Pacific - 9%
**IPv6 Header (Cont)**

*Priority*: identify priority among datagrams in flow

*Flow Label*: identify datagrams in same “flow.”

(concept of “flow” not well defined).

*Next header*: identify upper layer protocol for data

<table>
<thead>
<tr>
<th>ver</th>
<th>pri</th>
<th>flow label</th>
<th>payload len</th>
<th>next hdr</th>
<th>hop limit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>source address (128 bits)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>destination address (128 bits)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>data</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Other Changes from IPv4

- **Checksum**: removed entirely to reduce processing time at each hop
- **Options**: allowed, but outside of header, indicated by "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 "flag days"
  - How will the network operate with mixed IPv4 and IPv6 routers?

- Two proposed approaches:
  - Dual Stack: some routers with dual stack (v6, v4) can "translate" between formats
  - Tunneling: IPv6 carried as payload in IPv4 datagram among IPv4 routers
Dual Stack Approach
Tunneling

IPv6 inside IPv4 where needed
VLANs

- Defn: VLANs define broadcast domains in a layer 2 network.
- Legacy networks: broadcast domain boundaries were determined by router interfaces.
- Layer 2 switches: administrator tells switch how far it can propagate the broadcast.
  - What other ports allowed to receive it?
  - Flood to all ports or a subset?
Broadcast Domains in Legacy Networks
Broadcast Domains in Switched Networks
VLANs in Switched Networks
Why VLANs?

- **Security:** VLANs enable you to place all functionally related users in the same broadcast domain - isolated from others.

- **Broadcast Distribution:** VLANs restrict broadcasts (ARP, etc.)

- **Bandwidth Utilization:** LAN Switches + VLANs limit the amount of bandwidth sharing

- **Router Latency:** VLANs can be used to take routers OUT of the source-dest path.

- **Workstation Moves:** Simplifies moves of manually configured workstations.
Simple LAB VLAN

VLAN1

RSM

VLAN2

Routers 1 through 6

Campus Router
Hierarchical Network Design

(Every node and link in network is a single point of failure.)

Core
• Switching
• Full Routing Information

Distribution Layer
• Aggregation
• Summarization
• Partial Routing Information

Access Layer
• Network Entry Control
• Edge Services
• Minimal Routing Information
Redundancy

- **Defn.** Providing alternate paths around potential failure points to reduce the probability of a loss of service.

- **Caution.** Redundancy complicates routing, weakens hierarchy, may reduce stability.
Mathematical Advantage

- Assume (workstation) H-R-R-R-H (server)
- Assume each node/link has MTTF of 1000 hours and that failures are independent.
- Estimate end-to-end MTTF:

  Probability of at least one failure in an hour
  
  \[ P[\text{at least one element fails}] = 1 - P[\text{all 7 elements working}] \]
  
  \[ = 1 - (1 - .001)^7 = 1 - (0.999)^7 = 0.007. \]

  System MTTF = \[ \frac{1}{0.007} = 142.9 \text{ hours.} \]
Add a Redundant Path

- Assume \[ R - R - R \]
  \[ H - R - R - R - H \]
- Probability of failure (per hour) is probability that top path fails AND bottom path fails or \[ 0.007 \times 0.007 = 0.000049 \].
  - Because failures are independent
- MTTF is 20,408 hours!
Series System Reliability

- A series system is one in which all components are so interrelated that the entire system will fail if any one of its components fails.

- In this case define the reliability of component $i$ as

  $$ R_i = P(A_i) = P[\text{component } i \text{ is functioning properly}] $$

- Then system reliability is:

  $$ R_s = P(A_1 \cap A_2 \cap \ldots \cap A_n) = \prod_{i=1}^{n} R_i. $$
Parallel System Reliability

- A parallel system is one in which all components are so interrelated that the entire system will fail only if all of its components fail. (We assume components are mutually independent.)

- With same definitions, probability of system failure is:

$$P[\overline{A_1} \cap \overline{A_2} \cap \ldots \cap \overline{A_n}] = \prod_{i=1}^{n} (1 - P(A_i))$$

$$= \prod_{i=1}^{n} (1 - R_i).$$
Series and Parallel Reliability

- In summary, assuming independence of components, the reliability of a system in series is:
  \[ R_s = P(A_1 \cap A_2 \cap \ldots \cap A_n) \]
  \[ = \prod_{i=1}^{n} R_i. \]
- and the reliability of a system in parallel is:
  \[ R_p = 1 - \prod_{i=1}^{n} (1 - R_i) \]
Simple Example:

\[ R_{sp} = R_1 \cdot R_2 \cdot \left[ 1 - (1 - R_3)^3 \right] \cdot \left[ 1 - (1 - R_4)^2 \right] \cdot R_5 \]
**OSPF (Open Shortest Path First)**

- “open”: publicly available
- Uses Link State algorithm
  - LS packet dissemination
  - Topology map at each node
  - Route computation using Dijkstra’s algorithm

- OSPF advertisement carries one entry per neighbor router
- Advertisements disseminated to entire AS (via flooding)
Hierarchical OSPF (single AS)
Hierarchical OSPF

- **Two-level hierarchy**: local area, backbone.
  - Link-state advertisements only in area
  - Each node has detailed area topology; only knows 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 ASs.
Enabling OSPF on the router involves the following two steps in config mode:

1. Enabling an OSPF process using the router ospf <process-id> command.

2. Assigning areas to the interfaces using the network <network or IP address> <mask> <area-id> command.

The OSPF process-id is a numeric value local to the router. It does not have to match process-ids on other routers. It is possible to run multiple OSPF processes on the same router, but is not recommended as it creates multiple database instances that add extra overhead to the router.

The network command is a way of assigning an interface to a certain area. The mask is used as a shortcut and it helps putting a list of interfaces in the same area with one configuration line. The mask contains wildcard bits where 0 is a match and 1 is a "do not care" bit, e.g. 0.0.255.255 indicates a match in the first two bytes of the network number. Backwards mask!

The area-id is the area number we want the interface to be in. The area-id can be an integer between 0 and 4294967295 or can take a form similar to an IP address A.B.C.D.
Example:

![Network Diagram]

**Commands:**

*RTA#*

interface Ethernet0
ip address 192.213.11.1

interface Ethernet1
ip address 192.213.12.2

interface Ethernet2
ip address 128.213.1.1

**Commands:**

*RTA#*

router OSPF 100
network 192.213.0.0 0.0.255.255 area 0.0.0.0
network 128.213.1.1 0.0.0.0 area 23
Redundant Network Design

Network Analysis: Internetworking
Example: Enable OSPF on Router 4

Define interfaces as usual then…
config t
router ospf 100
network 10.5.4.0 0.0.0.255 area 3
network 10.0.30.0 0.0.0.255 area 8
network 10.6.4.0 0.0.0.255 area 6
exit
write
Definition: Router ID (RID)

- Used when unique reference to router required.
- Typically the largest IP address on the router interfaces.
- Example, router 5 has 4 IP addresses:
  - 10.0.100.5
  - 10.5.2.1
  - 10.5.3.1
  - 10.5.4.1
- RID is 10.5.4.1
Virtual Links

- Used for two purposes:
  - Linking an area that does not have a physical connection to the backbone (area 0)
  - Patching the backbone in case discontinuity of area 0 occurs.

- Ex. Router 4 connects to areas 3, 6, 8 and area 8 does not connect directly to area 0. Define two virtual links to area 0 from area 8. One through area 3 and one through area 6.
Creating Virtual Links

in router 4:
config
tarea 3 virtual-link 10.5.4.1 (RID of router 5)
tarea 6 virtual-link 10.6.4.1 (RID of router 6)
exit
write

NOTE: Virtual links must be entered in pairs. If a virtual link is defined from router A to router B, one must be defined from router B to router A.
Enabling OSPF
(Summary for each router)

- Enable OSPF process using `router OSPF <pid>` command.
- Define all router interfaces as usual.
- Assign areas to interfaces using `network` command with “backward” mask.
- Set up any required virtual links to connect all areas to backbone (or join split backbone).
  - Use RID in virtual-link command.
  - Define virtual links in router pairs.
Redundant Network Design

Network Analysis: Internetworking
Review: Manual Routing from 10.0.20.20

- Need at least the following:
  - 10.0.20.0 255.255.255.0 hme0
  - Default 10.0.20.1
In Router 2

- Default goes to router 5
  - 0.0.0.0 0.0.0.0 10.5.2.1

- Others
  - 10.6.2.0 255.255.255.0 Serial0/1
  - 10.0.20.0 255.255.255.0 FastEthernet0/1
In Router 5

- Default
  - 0.0.0.0 0.0.0.0 10.0.100.100

- Others
  - 10.0.100.0 255.255.255.0 FastEthernet0/0
  - 10.5.2.0 255.255.255.0 Serial0/0
  - 10.5.3.0 255.255.255.0 Serial0/1
  - 10.5.4.0 255.255.255.0 FastEthernet0/1
  - 10.0.20.0 255.255.255.0 10.5.2.2
  - 10.0.30.0 255.255.255.0 10.5.3.2
In RSM

- Default is out of lab via VLAN2
- 10.0.100.0 255.255.255.0 VLAN1
- 10.5.0.0 255.255.0.0 10.0.100.5
- 10.6.0.0 255.255.0.0 10.0.100.6
- 10.0.20.0 255.255.255.0 10.0.100.5
- 10.0.30.0 255.255.255.0 10.0.100.6
Network Address Translation

- Networks can be allowed to use private addresses within some limited scope.
  - Other nets may use same addresses elsewhere.
  - These addresses may not be used globally.
  - 10.*.*.* commonly used.

- Global access provided through a NAT box that translates between private address and a global addressed available to the NAT box.
  - NAT may manage a pool of global addresses and assign them as needed.
  - PAT (port address translation) can reuse a single global address among several hosts on the internal (10.*) net.