Chapter 4: Network Layer

Chapter goals:
- understand principles behind network layer services
- instantiation, implementation in the Internet
Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Datagram networks
- 4.3 What’s inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 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

- transport segment from sending to receiving host
- on sending side, encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in every host, router
- Router examines header fields in all IP datagrams passing through it
Two 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**

**analogy:**
- **routing**: process of planning trip from source to dest
- **forwarding**: process of getting through single interchange

Interplay between routing and forwarding

<table>
<thead>
<tr>
<th>Routing Algorithm</th>
<th>Local Forwarding Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Header Value</td>
</tr>
<tr>
<td></td>
<td>0100</td>
</tr>
<tr>
<td></td>
<td>0101</td>
</tr>
<tr>
<td></td>
<td>0111</td>
</tr>
<tr>
<td></td>
<td>1001</td>
</tr>
</tbody>
</table>

Value in arriving packet’s header
Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Datagram networks
- 4.3 What’s inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 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

Datagram networks

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

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00010000 00000000 through 11001000 00010111 11111111</td>
<td>0</td>
</tr>
<tr>
<td>11001000 00010111 00011000 00000000 through 11001000 00010111 11111111</td>
<td>1</td>
</tr>
<tr>
<td>11001000 00010111 00011001 00000000 through 11001000 00010111 11111111</td>
<td>2</td>
</tr>
<tr>
<td>otherwise through 11001000 00010111 11111111</td>
<td>3</td>
</tr>
</tbody>
</table>

Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Datagram networks
- 4.3 What’s inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 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
Router Architecture Overview

Two key router functions:
- run routing algorithms/protocol (RIP, OSPF, BGP)
- forwarding datagrams from incoming to outgoing link

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
  - IPv4 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
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"

Transport layer: TCP, UDP

<table>
<thead>
<tr>
<th>Forwarding table</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP protocol</td>
</tr>
<tr>
<td>ICMP protocol</td>
</tr>
</tbody>
</table>

Link layer

physical layer

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
  - IPv4 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
**IP datagram format**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP protocol version number</td>
<td>32 bits</td>
</tr>
<tr>
<td>header length (bytes)</td>
<td>16-bit identifier</td>
</tr>
<tr>
<td>&quot;type&quot; of data</td>
<td>16-bit identifier</td>
</tr>
<tr>
<td>max number remaining hops</td>
<td>time to live</td>
</tr>
<tr>
<td>remaining hops (decremented at each router)</td>
<td>upper layer</td>
</tr>
<tr>
<td>upper layer protocol to deliver payload to</td>
<td>checksum</td>
</tr>
<tr>
<td>how much overhead with TCP?</td>
<td>Options (if any)</td>
</tr>
<tr>
<td>❑ 20 bytes of TCP</td>
<td>data (variable length, typically a TCP or UDP segment)</td>
</tr>
<tr>
<td>❑ 20 bytes of IP</td>
<td></td>
</tr>
<tr>
<td>❑ = 40 bytes + app layer overhead</td>
<td></td>
</tr>
<tr>
<td>total datagram length (bytes)</td>
<td></td>
</tr>
<tr>
<td>for fragmentation/reassembly</td>
<td></td>
</tr>
</tbody>
</table>

**IP Fragmentation & Reassembly**

- network links have MTU (max transfer size) - largest possible link-level frame.
  - 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

reassemble
IP Fragmentation and Reassembly

**Example**
- 4000 byte datagram
- MTU = 1500 bytes

1480 bytes in data field

offset = 1480/8

One large datagram becomes several smaller datagrams

---

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
  - IPv4 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
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 typically has one interface
  - IP addresses associated with each interface

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Binary Representation</th>
<th>Octal Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>223.1.1.1</td>
<td>11011111 00000001 00000001 00000001</td>
<td>223 1 1 1</td>
</tr>
<tr>
<td>223.1.1.2</td>
<td>11011111 00000001 00000001 00000000</td>
<td>223 1 1 0</td>
</tr>
<tr>
<td>223.1.1.3</td>
<td>11011111 00000001 00000001 00000000</td>
<td>223 1 1 0</td>
</tr>
<tr>
<td>223.1.1.4</td>
<td>11011111 00000001 00000001 00000000</td>
<td>223 1 1 0</td>
</tr>
<tr>
<td>223.1.2.1</td>
<td>11011111 00000000 00000000 00000001</td>
<td>223 1 0 1</td>
</tr>
<tr>
<td>223.1.2.2</td>
<td>11011111 00000000 00000000 00000000</td>
<td>223 1 0 0</td>
</tr>
<tr>
<td>223.1.2.9</td>
<td>11011111 00000000 00000000 00001001</td>
<td>223 1 0 1</td>
</tr>
<tr>
<td>223.1.3.1</td>
<td>11011111 00000001 00000000 00000000</td>
<td>223 1 1 0</td>
</tr>
<tr>
<td>223.1.3.2</td>
<td>11011111 00000001 00000000 00000000</td>
<td>223 1 1 0</td>
</tr>
<tr>
<td>223.1.3.27</td>
<td>11011111 00000001 00000000 01100101</td>
<td>223 1 1 1</td>
</tr>
</tbody>
</table>

Subnets

- **IP address**:
  - Subnet part (high order bits)
  - Host part (low order bits)

- **What's a subnet?**
  - Device interfaces with same subnet part of IP address
  - Can physically reach each other without intervening router

Network consisting of 3 subnets
**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.

Subnet mask: /24

**Subnets**

How many?
IP addressing: CIDR

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

![Subnet and host parts](examplesubnet.png)

IP addresses: how to get one?

Q: How does host get IP address?

- hard-coded by system admin in a file
  - Wintel: control-panel->network->configuration->tcp/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>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:

"Send me anything with addresses beginning 200.23.16.0/20"
"Send me anything with addresses beginning 199.31.0.0/16"
**Hierarchical addressing: more specific routes**

ISP-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 receives a specific route to Organization 1

Less specific routes:

- 200.23.16.0/23
- 200.23.18.0/23
- 200.23.20.0/23
- 200.23.30.0/23

**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
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
  - IPv4 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

ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate 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>4</td>
<td>0</td>
<td>source quench (congestion control - not used)</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>
</tbody>
</table>
Traceroute and ICMP

- Source sends series of UDP segments to dest
  - First has TTL = 1
  - Second has TTL = 2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router & IP address
- When ICMP message arrives, source calculates RTT
- Traceroute does this 3 times

Stopping criterion
- UDP segment eventually arrives at destination host
- Destination returns ICMP "host unreachable" packet (type 3, code 3)
- When source gets this ICMP, stops.

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
  - IPv4 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
**IPv6**

- **Initial motivation:** 32-bit address space soon to be completely allocated.
- **Additional motivation:**
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS
- **IPv6 datagram format:**
  - fixed-length 40 byte header
  - no fragmentation allowed

---

**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
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
  - IPv4 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

**Interplay between routing, forwarding**

[Diagram showing routing algorithm and forwarding process]

Value in arriving packet’s header

Local forwarding table:

<table>
<thead>
<tr>
<th>Header Value</th>
<th>Output Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>3</td>
</tr>
<tr>
<td>0101</td>
<td>2</td>
</tr>
<tr>
<td>0111</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
</tr>
</tbody>
</table>
Graph abstraction

Graph: $G = (N, E)$

$N =$ set of routers = { u, v, w, x, y, z }

$E =$ set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Remark: Graph abstraction is useful in other network contexts

Example: P2P, where N is set of peers and E is set of TCP connections

Graph abstraction: costs

- $c(x,x') =$ cost of link $(x,x')$
  - e.g., $c(w,z) = 5$

- cost could always be 1, or inversely related to bandwidth, or inversely related to congestion

Cost of path $(x_1, x_2, x_3, ..., x_p) = c(x_1,x_2) + c(x_2,x_3) + ... + c(x_{p-1},x_p)$

Question: What's the least-cost path between u and z ?

Routing algorithm: algorithm that finds least-cost path
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

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
  - IPv4 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
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 forwarding table for that node
- iterative: after k iterations, know least cost path to k dest.’s

Notation:
- \( c(x,y) \): link cost from node x to y: \( \infty \) 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
- \( N' \): set of nodes whose least cost path definitively known

Dijsktra’s Algorithm

1. Initialization:
   2. \( N' = \{u\} \)
   3. for all nodes v
   4. if v adjacent to u
   5. then \( D(v) = c(u,v) \)
   6. else \( D(v) = \infty \)
   7. 

8. Loop
   9. find w not in \( N' \) such that \( D(w) \) is a minimum
   10. add w to \( N' \)
   11. update \( D(v) \) for all v adjacent to w and not in \( N' \) :
   12. \( D(v) = \min( D(v), D(w) + c(w,v) ) \)
   13. /* new cost to v is either old cost to v or known
   14. shortest path cost to w plus cost from w to v */
   15. until all nodes in \( N' \)
**Dijkstra's algorithm: example**

<table>
<thead>
<tr>
<th>Step</th>
<th>N'</th>
<th>D(v),p(v)</th>
<th>D(w),p(w)</th>
<th>D(x),p(x)</th>
<th>D(y),p(y)</th>
<th>D(z),p(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>u</td>
<td>2,u</td>
<td>5,u</td>
<td>1,u</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>ux</td>
<td>2,u</td>
<td>4,x</td>
<td>2,x</td>
<td>∞</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>uxy</td>
<td>2,u</td>
<td>3,y</td>
<td>4,y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>uxyv</td>
<td>3,y</td>
<td></td>
<td>4,y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>uxyvw</td>
<td></td>
<td></td>
<td></td>
<td>4,y</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>uxyvwz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Resulting shortest-path tree from u:

Resulting forwarding table in u:

<table>
<thead>
<tr>
<th>destination</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>(u,v)</td>
</tr>
<tr>
<td>x</td>
<td>(u,x)</td>
</tr>
<tr>
<td>y</td>
<td>(u,x)</td>
</tr>
<tr>
<td>w</td>
<td>(u,x)</td>
</tr>
<tr>
<td>z</td>
<td>(u,x)</td>
</tr>
</tbody>
</table>
**Dijkstra’s algorithm, discussion**

**Algorithm complexity:** $n$ nodes
- each iteration: need to check all nodes, $w$, not in $N$
- $n(n+1)/2$ comparisons: $O(n^2)$
- more efficient implementations possible: $O(n\log n)$

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

```
initially ... recompute routing ... recompute ... recompute
```

**Distance Vector: link cost changes**

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

---

"good news travels fast"

At time $t_0$, $y$ detects the link-cost change, updates its DV, and informs its neighbors.

At time $t_1$, $z$ receives the update from $y$ and updates its table. It computes a new least cost to $x$ and sends its neighbors its DV.

At time $t_2$, $y$ receives $z$’s update and updates its distance table. $y$’s least costs do not change and hence $y$ does not send any message to $z$. 
Distance Vector: link cost changes

Link cost changes:
- good news travels fast
- bad news travels slow - “count to infinity” problem!
- 44 iterations before algorithm stabilizes: see text

Poisoned reverse:
- If Z routes through Y to get to X:
  - Z tells Y its (Z’s) distance to X is infinite (so Y won’t route to X via Z)
- will this completely solve count to infinity problem?

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
  - IPv4 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
Hierarchical Routing

Our routing study thus far - idealization

- all routers identical
- network "flat"
... not true in practice

scale: with 200 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 router

- Direct link to router in another AS
Interconnected ASes

- Forwarding table is configured by both intra- and inter-AS routing algorithm
  - Intra-AS sets entries for internal dests
  - Inter-AS & Intra-AS sets entries for external dests

Inter-AS tasks

- Suppose router in AS1 receives datagram for which dest is outside of AS1
  - Router should forward packet towards one of the gateway routers, but which one?

AS1 needs:
1. to learn which dests are reachable through AS2 and which through AS3
2. to propagate this reachability info to all routers in AS1

Job of inter-AS routing!
**Example: Setting forwarding table in router 1d**

- Suppose AS1 learns (via inter-AS protocol) that subnet $x$ is reachable via AS3 (gateway 1c) but not via AS2.
- Inter-AS protocol propagates reachability info to all internal routers.
- Router 1d determines from intra-AS routing info that its interface $I$ is on the least cost path to 1c.
- Puts in forwarding table entry $(x,I)$.

**Example: Choosing among multiple ASes**

- Now suppose AS1 learns from the inter-AS protocol that subnet $x$ is reachable from AS3 and from AS2.
- To configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest $x$. 
Example: Choosing among multiple ASes

- Now suppose AS1 learns from the inter-AS protocol that subnet \( x \) is reachable from AS3 and from AS2.
- To configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest \( x \).
- This is also the job on inter-AS routing protocol!
- Hot potato routing: send packet towards closest of two routers.

| Learn from inter-AS protocol that subnet \( x \) is reachable via multiple gateways | Use routing info from intra-AS protocol to determine costs of least-cost paths to each of the gateways | Hot potato routing: Choose the gateway that has the smallest least cost | Determine from forwarding table the interface \( I \) that leads to least-cost gateway. Enter \((x, I)\) in forwarding 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
  - IPv4 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
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)

---

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
  - IPv4 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
RIP (Routing Information Protocol)

- Distance vector algorithm

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
  - IPv4 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
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

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
  - IPv4 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
Internet inter-AS routing: BGP

- **BGP (Border Gateway Protocol):** *the de facto standard*
- **BGP provides each AS a means to:**
  1. Obtain subnet reachability information from neighboring ASes.
  2. Propagate reachability information to all AS-internal routers.
  3. Determine “good” routes to subnets based on reachability information and policy.
- **allows subnet to advertise its existence to rest of Internet: “I am here”**

BGP basics

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

- With eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
- 1c can then use iBGP to distribute this new prefix reachability info to all routers in AS1.
- 1b can then re-advertise new reachability info to AS2 over 1b-to-2a eBGP session.
- When router learns of new prefix, creates entry for prefix in its forwarding 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
  - IPv4 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
**Broadcast Routing**

- Deliver packets from source to all other nodes
- Source duplication is inefficient:

  ![Network Diagram]

  - Source duplication: how does source determine recipient addresses?

**In-network duplication**

- Flooding: when node receives brdcst pckt, sends copy to all neighbors
  - Problems: cycles & broadcast storm
- Controlled flooding: node only brdcsts pkt if it hasn’t brdcst same packet before
  - Node keeps track of pckt ids already brdcsted
  - Or reverse path forwarding (RPF): only forward pckt if it arrived on shortest path between node and source
- Spanning tree
  - No redundant packets received by any node
### Spanning Tree

- First construct a spanning tree
- Nodes forward copies only along spanning tree

(a) Broadcast initiated at A
(b) Broadcast initiated at D

### Multicast Routing: Problem Statement

- **Goal:** find a tree (or trees) connecting routers having local mcast group members
  - **tree:** not all paths between routers used
  - **source-based:** different tree from each sender to rcvrs
  - **shared-tree:** same tree used by all group members

Shared tree  
Source-based trees
Chapter 4: summary

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What’s inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 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