Chapter 4
Network 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
Network 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.

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

*routing algorithms*
Interplay between routing and forwarding

Routing algorithm

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>

Value in arriving packet’s header
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
Network layer connection and connection-less service

- datagram network provides network-layer connectionless service
- VC network provides network-layer connection service
Virtual circuits

“source-to-dest path behaves much like telephone circuit”

- performance-wise
- network actions along source-to-dest path

- each packet carries VC identifier (not destination host address)
- every router on source-dest path maintains “state” for each passing connection
- link, router resources (bandwidth, buffers) may be allocated to VC (dedicated resources = predictable service)
### Forwarding table

**Forwarding table in northwest router:**

<table>
<thead>
<tr>
<th>Incoming interface</th>
<th>Incoming VC #</th>
<th>Outgoing interface</th>
<th>Outgoing VC #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>3</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>63</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>1</td>
<td>97</td>
<td>3</td>
<td>87</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Routers maintain connection state information!
Virtual circuits: signaling protocols

- used in ATM, frame-relay, X.25
- not used in today’s Internet

1. Initiate call
2. incoming call
3. Accept call
4. Call connected
5. Data flow begins
6. Receive data
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 00010111 11111111</td>
<td>0</td>
</tr>
<tr>
<td>11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111</td>
<td>1</td>
</tr>
<tr>
<td>11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111</td>
<td>2</td>
</tr>
<tr>
<td>otherwise 11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111</td>
<td>3</td>
</tr>
</tbody>
</table>

4 billion possible entries
Longest prefix matching

<table>
<thead>
<tr>
<th>Prefix Match</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00010</td>
<td>0</td>
</tr>
<tr>
<td>11001000 00010111 00011000</td>
<td>1</td>
</tr>
<tr>
<td>11001000 00010111 00011</td>
<td>2</td>
</tr>
<tr>
<td>otherwise</td>
<td>3</td>
</tr>
</tbody>
</table>

Examples

DA: 11001000 00010111 00010110 [0110 10100001]  Which interface?

DA: 11001000 00010111 00011000 [1000 10101010]  Which interface?
Router Architecture Overview

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

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</td>
<td>Number of the internet protocol version</td>
</tr>
<tr>
<td>header length</td>
<td>Length of the IP header (in bytes)</td>
</tr>
<tr>
<td>“type” of data</td>
<td>Protocol type, e.g., IPv4, IPv6</td>
</tr>
<tr>
<td>max number remaining</td>
<td>Maximum number of hops the packet can travel before being discarded</td>
</tr>
<tr>
<td>remaining hops</td>
<td>(Decrement at each router)</td>
</tr>
<tr>
<td>time to live</td>
<td>Time left for the packet to be received</td>
</tr>
<tr>
<td>upper layer</td>
<td>Protocol used to deliver the data to the destination</td>
</tr>
<tr>
<td>32 bit source IP address</td>
<td>Source IP address of the packet</td>
</tr>
<tr>
<td>32 bit destination IP address</td>
<td>Destination IP address of the packet</td>
</tr>
<tr>
<td>Options (if any)</td>
<td>Additional options for the packet</td>
</tr>
<tr>
<td>data</td>
<td>Payload data, typically a TCP or UDP segment</td>
</tr>
<tr>
<td>total datagram length</td>
<td>Total length of the datagram (in bytes)</td>
</tr>
<tr>
<td>for fragmentation/</td>
<td>Field used for fragmentation and reassembly</td>
</tr>
<tr>
<td>reassembly</td>
<td></td>
</tr>
</tbody>
</table>

Network Layer 4-19
IP Fragmentation & Reassembly

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

- **4000 byte datagram**
  - ID = x
  - fragflag = 0
  - offset = 0

- **MTU = 1500 bytes**
  - ID = x
  - fragflag = 1
  - offset = 0

- **1480 bytes in data field**
  - offset = 1480/8

- **1500 bytes**
  - ID = x
  - fragflag = 1
  - offset = 185

- **1500 bytes**
  - ID = x
  - fragflag = 1
  - offset = 370

- **1040 bytes**
  - ID = x
  - fragflag = 0
  - offset = 370
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

*Example IP addresses and binary representation*:

- 223.1.1.1 = 11011111 00000001 00000001 00000001
- 223.1.2.2 = 11010101 00000001 00000001 00000010
- 223.1.3.1 = 11010011 00000001 00000001 00000001
- 223.1.3.2 = 11010011 00000001 00000001 00000011
- 223.1.3.27 = 11010011 00000001 00000001 10000011

Network Layer 4-23
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 diagram showing three subnets](image-url)
Subnets

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: \texttt{a.b.c.d/x}, where \( x \) is \# bits in subnet portion of address

\[
\begin{array}{c}
11001000 \\
00010111 \\
00010000 \\
00000000
\end{array}
\]

200.23.16.0/23
IP addresses: how to get one?

Q: How does a host get IP address?

- hard-coded by system admin in a file
  - Windows: 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”
**DHCP: Dynamic Host Configuration Protocol**

**Goal:** allow host to *dynamically* obtain its IP address from network server when it joins network

- Allows reuse of addresses
DHCP client-server scenario

DHCP server: 223.1.2.5

DHCP discover
src: 0.0.0.0, 68
dest: 255.255.255.255, 67
yiaddr: 0.0.0.0
transaction ID: 654

DHCP offer
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 654
Lifetime: 3600 secs

DHCP request
src: 0.0.0.0, 68
dest: 255.255.255.255, 67
yiaddr: 223.1.2.4
transaction ID: 655
Lifetime: 3600 secs

DHCP ACK
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 655
Lifetime: 3600 secs
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
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 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

ISPs-R-Us

- “Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23”

Fly-By-Night-ISP

- “Send me anything with addresses beginning 200.23.16.0/20”

Internet
NAT: Network Address Translation

All datagrams **leaving** local network have **same** single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)
**NAT: Network Address Translation**

- **Motivation:** local network uses just one IP address as far as outside world is concerned:
  - range of addresses not needed from ISP: just one IP address for all devices
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security plus).
NAT: Network Address Translation

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

S: 138.76.29.7, 5001
D: 128.119.40.186, 80

3: Reply arrives dest. address: 138.76.29.7, 5001

S: 128.119.40.186, 80
D: 10.0.0.1, 3345

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345
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)

<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>
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

![IPv6 Header Diagram]

<table>
<thead>
<tr>
<th>ver</th>
<th>pri</th>
<th>flow label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>payload len</th>
<th>next hdr</th>
<th>hop limit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

source address
(128 bits)

destination address
(128 bits)

data

32 bits
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?

- **Tunneling**: IPv6 carried as payload in IPv4 datagram among IPv4 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
   - 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**

Routing algorithm

**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>

Value in arriving packet’s header

Network Layer 4-46
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 \( (x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p) \)

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. **Loop**
   8. find \( w \) not in \( N' \) such that \( D(w) \) is a minimum
   9. add \( w \) to \( N' \)
   10. update \( D(v) \) for all \( v \) adjacent to \( w \) and not in \( N' \) :
        \[
        D(v) = \min( D(v), D(w) + c(w,v) )
        \]
   11. /* new cost to \( v \) is either old cost to \( v \) or known
   12. shortest path cost to \( w \) plus cost from \( w \) to \( v \) */
   13. 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></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The network diagram represents the connections and the steps involved in the algorithm. The table shows the updated distances and predecessors at each step.
### 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>4,y</td>
<td>4,y</td>
</tr>
<tr>
<td>3</td>
<td>uxyv</td>
<td>2,u</td>
<td>3,y</td>
<td>4,y</td>
<td>4,y</td>
<td>4,y</td>
</tr>
<tr>
<td>4</td>
<td>uxyvw</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>uxyvzw</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Graph representation:**

![Graph](image)
Dijkstra’s algorithm: example (2)

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>
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
Distance Vector Algorithm

Bellman-Ford Equation (dynamic programming)

Define
\[ d_x(y) := \text{cost of least-cost path from } x \text{ to } y \]

Then
\[ d_x(y) = \min_v \{ c(x,v) + d_v(y) \} \]

where \( \min \) is taken over all neighbors \( v \) of \( x \)
Bellman-Ford example

Clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

B-F equation says:

$$d_u(z) = \min \{ c(u,v) + d_v(z),
               c(u,x) + d_x(z),
               c(u,w) + d_w(z) \}$$

$$= \min \{2 + 5, 1 + 3, 5 + 3\} = 4$$

Node that achieves minimum is next hop in shortest path ➔ forwarding table
Distance Vector Algorithm

- $D_x(y) = \text{estimate of least cost from } x \text{ to } y$
- Node $x$ knows cost to each neighbor $v$: $c(x,v)$
- Node $x$ maintains distance vector $D_x = [D_x(y) : y \in N]$
- Node $x$ also maintains its neighbors’ distance vectors
  - For each neighbor $v$, $x$ maintains $D_v = [D_v(y) : y \in N]$
Distance vector algorithm (4)

**Basic idea:**

- From time-to-time, each node sends its own distance vector estimate to neighbors.
- Asynchronous.
- When a node $x$ receives new DV estimate from neighbor, it updates its own DV using B-F equation:
  $$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \quad \text{for each node } y \in N$$
- Under minor, natural conditions, the estimate $D_x(y)$ converge to the actual least cost $d_x(y)$.
\[ D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} \]
\[ = \min\{2+0, 7+1\} = 2 \]

\[ D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} \]
\[ = \min\{2+1, 7+0\} = 3 \]
\[ D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} = \min\{2+0, 7+1\} = 2 \]

\[ D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} = \min\{2+1, 7+0\} = 3 \]
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
Inter-AS tasks

- Suppose router in AS1 receives datagram destined outside of AS1:
  - Router should forward packet to gateway router, but which one?

AS1 must:
1. Learn which dests are reachable through AS2, which through AS3
2. Propagate this reachability info to all routers in AS1

Job of inter-AS routing!
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)
RIP (Routing Information Protocol)

- distance vector algorithm
- included in BSD-UNIX Distribution in 1982
- distance metric: # of hops (max = 15 hops)

From router A to subnets:

<table>
<thead>
<tr>
<th>destination</th>
<th>hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>1</td>
</tr>
<tr>
<td>v</td>
<td>2</td>
</tr>
<tr>
<td>w</td>
<td>2</td>
</tr>
<tr>
<td>x</td>
<td>3</td>
</tr>
<tr>
<td>y</td>
<td>3</td>
</tr>
<tr>
<td>z</td>
<td>2</td>
</tr>
</tbody>
</table>
RIP advertisements

- *distance vectors*: exchanged among neighbors every 30 sec via Response Message (also called *advertisement*)

- each advertisement: list of up to 25 destination subnets within AS
RIP: Example

<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/Forwarding table in D
RIP: Example

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next</th>
<th>hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>x</td>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>z</td>
<td>B A</td>
<td>7 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Routing/Forwarding table in D

Advertisement from A to D

Destination Network | Next Router | Num. of hops to dest.
---------------------|-------------|----------------------
 w                   | A           | 2                    |
 y                   | B           | 2                    |
 z                   | B A         | 7 5                  |
 x                   | --          | 1                    |
   ....              | ....        | ....                 |

Network Layer 4-73
RIP: Link Failure and Recovery

If no advertisement heard after 180 sec -->
neighbor/link declared dead
  m routes via neighbor invalidated
  m new advertisements sent to neighbors
  m neighbors in turn send out new advertisements (if
tables changed)
  m link failure info propagates quickly to entire net
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 “advanced” features (not in RIP)

- **security**: all OSPF messages authenticated (to prevent malicious intrusion)
- **multiple same-cost paths** allowed (only one path in RIP)
- integrated uni- and **multicast** support:
  - **Multicast OSPF (MOSPF)** uses same topology database as OSPF
- **hierarchical** OSPF in large domains.
Hierarchical OSPF
Hierarchical OSPF

- **two-level hierarchy**: local area, backbone.
  - Link-state advertisements only in area
  - Each node has detailed area topology; only know direction (shortest path) to nets in other areas.

- **area border routers**: “summarize” distances to nets in own area, advertise to other Area Border routers.

- **backbone routers**: run OSPF routing limited to backbone.

- **boundary routers**: connect to other AS’s.
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 ASs.
  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 *promises* it will forward datagrams towards that prefix.
  - AS2 can aggregate prefixes in its advertisement
BGP route selection

- router may learn about more than 1 route to some prefix. Router must select route.
- elimination rules:
  1. local preference value attribute: policy decision
  2. shortest AS-PATH
  3. closest NEXT-HOP router
  4. additional criteria
**BGP routing policy**

- **A, B, C** are provider networks
- **X, W, Y** are customer (of provider networks)
- **X** is *dual-homed*: attached to two networks
  - X does not want to route from B via X to C
  - .. so X will not advertise to B a route to C
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:

  - source duplication: how does source determine recipient addresses?
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](image1)

![Source-based trees](image2)

**Shared tree**

**Source-based trees**
Chapter 4: summary

r  4.1 Introduction
r  4.2 Virtual circuit and datagram networks
r  4.3 What’s inside a router
r  4.4 IP: Internet Protocol
   m  Datagram format
   m  IPv4 addressing
   m  ICMP
   m  IPv6
r  4.5 Routing algorithms
   m  Link state
   m  Distance Vector
   m  Hierarchical routing
r  4.6 Routing in the Internet
   m  RIP
   m  OSPF
   m  BGP
r  4.7 Broadcast and multicast routing