Chapter 4: Network Layer

Chapter goals:
- Understand principles behind network layer
- Instantiation, Implementation in the Internet
- Advanced topics: IPv6, mobility
- Dealing with scale
- Routing (path selection)
- How a router works
- Forwarding versus routing
- Network layer service models

Services:
passing through it
fields in all IP datagrams
router examines header
in every host, router
layer segments to transport
on receiving side, delivers
into datagrams
encapsulates segments
on sending side
sending to receiving host
transport segment from

ci

Chapter 4: Network Layer
Two Key Network-Layer Functions

- Forwarding: move packets from router's input to appropriate router output
- Analogies:
  - Routing: process of planning trip from source to dest.
  - Forwarding: process of getting through single interchange

Routing Algorithms:
- Determine the route taken by packets from source
- Input to appropriate packets from router's output
- Analogies:
- Routing: determine
- Forwarding: move

Interplay between Routing and Forwarding
Connection setup

ATM, Frame Relay, X.25

3rd important function in some network architectures:

Network service model

Example services for a flow of datagrams:

- Guaranteed delivery
- Guaranteed minimum bandwidth to flow
- Guaranteed delay with less than 40 msec

Example services for individual datagrams:

- In-order datagram delivery
- Guaranteed delivery

Network layer 4-7

- Network: between two hosts (may also involve intervening routers)
- Transport: between two processes in case of VCs

4-8

Q: What service model for "channel" transporting datagrams from sender to receiver?
Network layer service models:

- ATM
  - VBR
  - ABR
  - CBR
  - UBR

- Internet
  - ATM
  - IP
  - ICMP
  - IPv4 addressing
  - Datagram format
  - Protocol
  - RIP
  - OSPF
  - BGP

- Internet Routing in the Network Layer
  - 4.5 Routing algorithms
  - Link state
  - Distance vector
  - Link state
  - Hierarchical 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
  - Link state
  - Hierarchical routing

4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP

4.7 Broadcast and multicast routing

Network layer service models:

<table>
<thead>
<tr>
<th>Congestion Feedback</th>
<th>Bandwidth</th>
<th>Loss</th>
<th>Order</th>
<th>Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Internet</td>
<td>best effort</td>
<td>none</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>VBR</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>ABR</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>UBR</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>CBR</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>none</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
Virtual circuits

"source-to-dest path behaves much like telephone circuit"

- call setup, tear down for each call before data can flow
- network actions along source-to-dest path
- link, router resources (bandwidth, buffers) may be allocated to VC (dedicated resources = predictable service)
- every router on source-dest path maintains "state" for each passing VC identifier (not destination host address)
- performance-wise no choice: host-to-host service
- network provides one or the other
- implementation: in network core

Network layer connection and connection-less service

- datagram network provides network-layer connection-less service
- VC network provides network-layer connection service but: analogous to the transport-layer services,
A VC consists of:

1. A path from source to destination
2. VC numbers, one number for each link along the path
3. Entries in forwarding tables in routers along the path

A packet belonging to a VC carries VC number (rather than dest address)

VC number can be changed on each link.

Routing table in the 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>
</tbody>
</table>

Routers maintain connection state information!
Virtual circuits: signaling protocols

- used to setup, maintain, and teardown VC
- used in ATM, frame-relay, X.25
- not used in today's Internet

Datagram networks

- no call setup at network layer
- 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

Datagram networks

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

Virtual circuits: signaling protocols
### Forwarding Table

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00010000 00000000 through 0</td>
<td>otherwise</td>
</tr>
<tr>
<td>11001000 00010111 00010111 11111111</td>
<td>4 billion possible entries</td>
</tr>
<tr>
<td>11001000 00010111 00011000 00000000 through 1</td>
<td>4 billion possible entries</td>
</tr>
<tr>
<td>11001000 00010111 00011111 11111111</td>
<td>4 billion possible entries</td>
</tr>
<tr>
<td>11001000 00010111 00011001 00000000 through 2</td>
<td>4 billion possible entries</td>
</tr>
<tr>
<td>11001000 00010111 00011111 11111111</td>
<td>4 billion possible entries</td>
</tr>
</tbody>
</table>

### Examples

<table>
<thead>
<tr>
<th>Prefix Match</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>11000 11101000 00010011</td>
</tr>
<tr>
<td>2</td>
<td>00000000 10011000 11101000 00010011</td>
</tr>
<tr>
<td>1</td>
<td>00000000 00011000 11101000 00010011</td>
</tr>
<tr>
<td>0</td>
<td>00000000 00001000 11101000 00010011</td>
</tr>
</tbody>
</table>

### Longest Prefix Matching

<table>
<thead>
<tr>
<th>Network Layer 4-18</th>
<th>Which interface?</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA: 11001000 00010111 00011000 10101010</td>
<td>otherwise</td>
</tr>
<tr>
<td>DA: 11001000 00010111 00010110 10100001</td>
<td>Which interface?</td>
</tr>
</tbody>
</table>
Datagram or VC network: why?

Internet (datagram)

- "elastic" service, no strict timing req.
- "smart" end systems (computers)
- can adapt, perform control, error recovery
- simple inside network, complexity at "edge"
- need for guaranteed service, strict timing, reliability
- human conversation: evolved from telephony
- need for guaranteed service
- BGP
- OSPF
- RIP
- Link State
- Distance Vector
- Link State
- IP
- ICMP
- IPv4 addressing
- Datagram format
- Protocol

ATM (VC)

- "smart" service, no strict timing req.
- evolved from telephony
- data exchange among "elastic" computers
- hierarchical routing
- network
- complexity inside
- telephones
- dumb end systems service
- requirements
- Internet

Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.4 IP: Internet Protocol

4.5 Routing algorithms

4.6 Routing in the Network Layer

4.7 Broadcast and multicast routing

4.8 RIP

4.9 OSPF

4.10 BGP

4.11 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

Input Port Functions

Physical layer:
- Bit-level reception

Data link layer:
- Ethernet

Network Layer 4-21

Decentralized switching:
- Given datagram dest., lookup output port using forwarding table in input port memory
- Queueing: If datagrams arrive faster than forwarding rate, complete input port processing at line speed
- Goal: Complete input port processing at line speed

Forwarding Rate into Switch Fabric

Queuing: If datagrams arrive faster than forwarding rate, complete input port processing at line speed

Routing Processor

Forwarding datagrams from incoming to outgoing link

Two Key Router Functions:
Three types of switching fabrics

Network Layer 4-23

Switching Via Memory

First generation routers:
- Traditional computers with switching under direct control of CPU
- Packet copied to system's memory
- Speed limited by memory bandwidth (2 bus crossings per datagram)

Network Layer 4-24

Crossbar

Three types of switching fabrics
Switching Via a Bus

Datagram from input port memory to output port memory via a shared bus

Network Layer 4-25

Bus contention: switching speed limited by bus bandwidth

Bus

Switching Via An Interconnection Network

Banyan networks, other interconnection nets

Advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.

Cisco 12000: switches 60 Gbps through the interconnection network

Overcome bus bandwidth limitations

Network Layer 4-26

Cisco 12000: switches 60 Gbps through the interconnection network

Overcome bus bandwidth limitations

Advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.

Network Layer 4-26

Bus contention: switching speed limited by bus bandwidth

Bus

to output port memory via a shared bus
datagram from input port memory

Network Layer 4-25

Switching speed for access and enterprise

32 Gbps bus, Cisco 5600: sufficient

Network Layer 4-25
Output Port Queuing

- Buffer overflow
- Queueing (delay) and loss due to output port
- Output line speed
- Buffering when arrival rate via switch exceeds

Datagrams for transmission

Scheduling discipline chooses among queued

Fabric faster than the transmission rate
• Buffering required when datagrams arrive from

Output Ports
How much buffering?

**RFC 3439 rule of thumb:** average buffering equal to "typical" RTT (say 250 msec) times link capacity $C$.

- $N$ flows, $C = 10$ Gbps link: 2.5 Gbit buffer
- $N$ flows, $C = 10$ Gbps link: 2.5 Gbit buffer

**Recent recommendation:** with $N$ flows, $C = 10$ Gbps link: 2.5 Gbit buffer

- $N$ flows, $C = 10$ Gbps link: 2.5 Gbit buffer

**Input Port Queuing**

- Fabric slower than input ports combined -> queuing delay and loss due to input buffer overflow!
- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward
- May occur at input queues

- Fabric slower than input ports combined - queuing delay and loss due to input buffer overflow!
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 Network Layer
  - RIP
  - OSPF
  - BGP

4.7 Broadcast and multicast routing

The Internet Network Layer

Host, router network layer functions:

- Routing protocols: RIP, OSPF, BGP
- IP protocol: addressing conventions, datagram format, packet handling conventions
- Transport layer: TCP, UDP

Network Layer Forwarding Table

- Physical layer
- Link layer
- Transport layer: TCP, UDP
Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.4 IP: Internet Protocol

4.5 Routing algorithms

- Link state
- Distance Vector
- Link State
- Hierarchical routing
- Hybrid
- Hybrid

4.6 Routing in the router

- RIP
- OSPF
- RIP
- BGP
- IGP
- BGP
- EGP
- BGP
- EGP
- IGP
- BGP
- EGP
- IGP

IP Datagram Format

- Version (4 bits)
- Header Length (4 bits)
- Type of Service (8 bits)
- Total Length (16 bits)
- Identification (16 bits)
- Flags (3 bits)
- Fragment Offset (13 bits)
- Time to Live (8 bits)
- Protocol (8 bits)
- Header Checksum (16 bits)
- Source Address (32 bits)
- Destination Address (32 bits)
- Options (if any)

IP Datagram Format

- Data (variable length, typically a TCP or UDP segment)
- Upper layer protocol to deliver payload to upper layer protocol
- Each router decrements remaining hops
- Max number of hops

IPv4

- Version
- Header Length
- Type of Service
- Total Length
- Identification
- Flags
- Fragment Offset
- Time to Live
- Protocol
- Header Checksum
- Source Address
- Destination Address
- Options

IPv6

- Version
- Header Length
- Next Header
- Payload Length
- Source Address
- Destination Address
- Options

IPv4 addressing

IPv6 addressing

ICMP

Datagram Format

Protocol

Internet

Multicast routing

Broadcast and Multicast

IPv4

IPv6

Routing in the Internet

4.7 Broadcast and Multicast

OSPF

RIP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP

EGP

IGP
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 network:

-One datagram becomes several datagrams

"reassembled" only at final destination

IP header bits used to identify, order related fragments

Example:

MTU = 1500 bytes

One large datagram becomes several smaller datagrams

4000 byte datagram

MTU = 1500 bytes

One data field (4000 bytes) - largest

MTU bytes in network links have MTU

ID = x

offset = 0

fragflag = 0

length = 4000

ID = x

offset = 185

fragflag = 1

length = 1500

ID = x

offset = 370

fragflag = 0

length = 1040

1480 bytes in data field

offset = 1480/8

Network layer 4-35

IP Fragmentation and Reassembly

Network layer 4-36

IP Fragmentation & Reassembly
Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.4 IP: Internet Protocol

4.5 Routing algorithms

4.6 Routing in the router

4.7 Broadcast and multicast routing

IPv6
ICMP
IPv4 addressing
Protocol

4.4.4 IP: Internet router

4.3 What's inside a datagram network

4.2 Virtual circuit and introduction

IP Addressing: Introduction

Interface: Associated with each
IP addresses
Interface
different
Interface
host typically has one
Multiple interfaces
routers typically have
and physical link
between host/router
Interface: connection
Router interface
Identifier for host,
IP address: 32-bit
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.

Recipe

To determine the subnet mask:

Detach each device's interface from its host or router, creating islands of isolated networks. Each isolated network is called a subnet.

Subnet mask: /24

Subnets

network consisting of 3 subnets

What's a subnet?

- Address
- Device interfaces with same subnet part of IP
- Host part (low order)
- Subnet part (high order bits)
IP addressing: CIDR

CIDR: Classless Interdomain Routing

Subnet portion of address

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

o Subnet portion of address of arbitrary length

How many?
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-Netwrok-Protocol
- UNIX: /etc/rc.config
- DHCP: Dynamic Host Configuration Protocol
  - Support for mobile users who want to join network (more shortly)
  - on"
  - "plug-and-play"

DHCP overview:
- Host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- Host requests IP address: "DHCP request" msg [optional]
- DHCP server sends address: "DHCP ack" msg
- Host requests IP address: "DHCP request" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- Host broadcasts "DHCP discover" msg [optional]

Goal: Allow host to dynamically obtain its IP address from network server when it joins network and can renew its lease on address when it re-joins network.

DHCP: Dynamic Host Configuration Protocol

IP addresses: how to get one?
DHCP server: 223.1.2.5 arriving

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: 223.1.3.2
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
DHCP: DHCP server receives Ethernet frame (broadcast) on LAN.

DHCP: DHCP server encapsulates in Ethernet, in UDP, encapsulated in IP, Ethernet demuxed to IP.

DHCP: use DHCP server for:
- hop router, address of DNS
- IP address, address of first-hop router for client
- connecting laptop needs its portion of address
- network mask (indicating network versus host)
- name and IP address of DNS server

DHCP can return more than just allocated IP address.
IP addresses: how to get one?

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

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

Hierarchical addressing allows efficient advertisement of routing information:

- Hierarchical addressing: route aggregation

ISP's block

199.31.0.0/16

Network Layer 4-52

Internet

Fly-By-Night-ISP

ISP's-R-Us

Organization 0

Organization 1

Organization 2

Organization 7

...
IP addressing: more specific routes

ISP-R-Us has a more specific route to Organization 1:

"Send me anything with addresses 200.23.16.0/23"

Fly-By-Night-ISP

Organization 7

ISP-R-Us

Organization 1

ISP-R-Us has a more specific route to Organization 1:

"Send me anything with addresses beginning 200.23.18.0/23"

"Send me anything with addresses beginning 200.23.16.0/20"

Q: How does an ISP get block of addresses?
A: ICANN: Internet Corporation for Assigned Names and Numbers

ICANN allocates addresses
ICANN manages DNS
ICANN assigns domain names, resolves disputes

IP addressing: the last word...
NAT: Network Address Translation

Motivation:
- Devices inside local network visible by outside world (a security plus).
- Devices inside local net not explicitly addressable.
- Can change IP without changing addresses of devices in local network.
- Can change ISP without notifying outside world.
- Range of addresses not needed from ISP: just one IP address for all devices.
- Network layer 4-55.

NAT: Network Address Translation

Network layer 4-55

Source, destination (as usual) have 10.0.0/24 address for network in this network.
Datagrams with source or destination in this network have different source port numbers.
NAT IP address: 138.76.29.7
NAT: Network Address Translation

NAT: Network Address Translation

Motivation: Local network uses just one IP address as far as outside world is concerned: devices inside local network not explicitly addressable, visible by outside world (a security plus).
NAT: Network Address Translation

Implementation:

NAT router must:

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

2. Remember (in NAT translation table) every (source IP address, new port #) to (NAT IP address, port #) translation pair.

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

4. Outgoing datagrams: replace (source IP address, port #) of incoming datagrams using (NAT IP address, new port #).
NAT: Network Address Translation

- 16-bit port-number field:
- NAT is controversial:
  - Routers should only process up to Layer 3
  - Network Layer 4-59
  - Violates end-to-end argument
  - NAT is controversial:
  - LAN-side address
  - 60,000 simultaneous connections with a single
  - 16-bit port-number field:

IPv6
- Address shortage should instead be solved by
  - Designers, e.g., P2P applications
  - NAT possibility must be taken into account by app
  - NAT controversy
NAT traversal problem

Solution 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATted host to:
- Learn public IP address (138.76.29.7)
- Add/remove port mappings
- Automate static NAT port map configuration (with lease times)
- Learn public IP address
- NATted host to:
  - Device (IGD) Protocol: Allows UPnP (Internet Gateway Plug and Play)

Solution 3: Relaying (used in Skype)
Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.5 Routing algorithms

Link state
Distance Vector
Hierarchical routing

4.6 Routing in the Network Layer

RIP
OSPF
BGP

4.7 Broadcast and multicast routing

IPv6
ICMP
IP addressing
Datagram Format
Routing Protocol
IPv4
Internet Protocol
Internet Router
IPv4 Datagram Networks
IPv4 What's Inside a Datagram Network
IPv4 Virtual Circuit and Link State
IPv4 Introduction
Traceroute and ICMP

Source sends series of UDP segments to destination host. 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 and IP address of that router.

Stopping criterion:
- Tracertoute does this 3 times.
- When ICMP message arrives, source calculates RTT
- UDP segment eventually arrives at destination host.
- Destination returns ICMP “host unreachable” packet.
- Tracertoute messaging stops.
- When source gets this (type 3, code 3) ICMP message, source calculates RTT. IPv6.

Chapter 4: Network Layer

4.1 Introduction
4.2 Virtual circuit and datagram networks
4.3 What’s inside a router
4.4 IP: Internet Protocol
4.5 Routing algorithms
- Link state
- Distance Vector
- Hierarchical routing

4.6 Routing in the Internet
- RIP
- OSPF
- BGP

4.7 Broadcast and multicast routing

IPv4 addressing
ICMP
IPv6
Protocol
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
- No option field

IPv6 Header (Cont):
- Priority: Identify priority among datagrams in flow
- Flow label: Identify datagrams in same "flow"
- Next header: Identity upper layer protocol for data

Network Layer 4-67
Other Changes from IPv4

Checksum: Removed entirely to reduce processing time at each hop

Options: Allowed, but outside of header, processing time at each hop

ICMPv6: New version of ICMP

Multicast Group Management Functions

Transition From IPv4 to IPv6

Not all routers can be upgraded simultaneously

Tunneling: IPv6 carried as payload in IPv4
Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

4.6 Routing in the router

- RIP
- OSPF
- BGP

4.4 IP: Internet Protocol

- Datagram format
- Addressing
- ICMP
- IPv6

4.7 Broadcast and multicast routing

Internet

- Local forwarding table
- Header value output link
- packet's header value

Interplay between routing, forwarding

Network Layer 4-74

Network Layer 4-73
Graph abstraction

Graph: $G = (N, E)$

- $N = \text{set of routers} = \{ u, v, w, x, y, z \}$
- $E = \text{set of links} = \{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

Remark: Graph abstraction is useful in other network contexts.

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

Remark: Graph abstraction is useful in other network contexts.

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

Routing algorithm:

- Finds least-cost path

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

Cost could always be 1, or inversely related to bandwidth, or inversely related to congestion, or inversely related to some other factor. For example, $c(w,z) = 5$.

Cost of link $(x,x') = c(x,x')$.
Routing Algorithm classification

Global or decentralized

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

Static or dynamic?

Static:
- routes change slowly over time

Dynamic:
- routes change more quickly
- periodic update

Decentralized:
- router knows physically-connected neighbors, link costs to neighbors
- “distance vector” algorithms

Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What’s inside a datagram network?

4.4 Routing in the Network Layer

4.5 Routing algorithms

4.6 Routing in the Internet

4.7 Broadcast and multicast routing

4.8 Routing in the Internet router

Internet Protocol
- IPv4 addressing
- Datagram format
- RIP
- BGP
- OSPF
- IPv6
- ICMP
- IPv6 addressing
- Datagram format

Routing Algorithms classification

Global or decentralized

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

Static or dynamic?

Static:
- routes change slowly over time

Dynamic:
- routes change more quickly
- periodic update

Decentralized:
- router knows physically-connected neighbors, link costs to neighbors
- “distance vector” algorithms

Internet:
- RIP
- OSPF
- BGP

Broadcast and multicast:
- broadcast and multicast routing
A Link-State Routing Algorithm

Dijkstra's algorithm accomplishes "link state broadcast" all nodes have same info

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 \( v \)
- \( p(v) \): predecessor node along path from source to \( v \)
- \( \Delta \): adjacent to \( v \)
- \( N' \): set of nodes whose least cost path definitively known

Dijsktra's Algorithm

1. Initialization:
   - \( N' = \{ u \} \)
   - for all nodes \( v \):
     - if \( v \) adjacent to \( u \) then \( D(v) = c(u,v) \)
     - else \( D(v) = \infty \)

2. Loop
   - find \( w \) not in \( N' \) such that \( D(w) \) is a minimum

3. for all nodes \( v \) adjacent to \( w \) and not in \( N' \):
   - \( D(v) = \min(D(v), D(w) + c(w,v)) \)

4. else if \( w \) adjacent to \( u \) then \( D(w) = \infty \)

5. \( N' = N' \cup \{ w \} \)

6. else \( D(w) = \min(D(w), D(v) + c(v,w)) \)

7. \( N' = N' \cup \{ v \} \)

8. until all nodes in \( N' \)

9. shortest path cost to \( v \) plus cost of path from \( v \) to \( v \)

10. new cost to \( v \) is either old cost to \( v \) or known

11. update \( D(v) \) for all \( v \) adjacent to \( w \) and not in \( N' \):

12. "known" shortest path to \( k \) dest.'s

13. after \( k \) iterations, know least cost path to \( k \) dest.'s

14. after \( k \) iterations, know least cost path to \( k \) dest.'s

15. give forwarding table to all other nodes

16. computes least cost paths from one node (source) to all other nodes

17. all nodes have same info

18. state broadcast" accomplished via "link state broadcast"

19. known to all nodes

20. net topology, link costs
Dijkstra's algorithm: example (2)

Resulting forwarding table in u:

<table>
<thead>
<tr>
<th>Link</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>z</td>
<td>(x' n)</td>
</tr>
<tr>
<td>m</td>
<td>(x' n)</td>
</tr>
<tr>
<td>y</td>
<td>(x' n)</td>
</tr>
<tr>
<td>x</td>
<td>(x' n)</td>
</tr>
<tr>
<td>v</td>
<td>(x' n)</td>
</tr>
</tbody>
</table>

Resulting shortest-path tree from u:

Dijkstra's algorithm: example
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
- More efficient implementations possible: \( O(\text{delay}) \)
- \( n(n+1)/2 \) comparisons: \( O(n^2) \)
- Each iteration: need to check all nodes, \( w \), not in \( N \)

Algorithm complexity: \( n \) nodes

Chapter 4: Network Layer

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

- IPv6
- ICMP
- IPv4 addressing
- Datagram format
- Protocol
- 4.4 IP: Internet Router
- 4.3 WHAT'S INSIDE A datagram networks
- 4.2 Virtual Circuit and
  - 4.1 Introduction

Discussion

Dijkstra's algorithm, discussion
Define $d_x(y)$ := cost of least-cost path from $x$ to $y$

Bellman-Ford example

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

Bellman-Ford equation says:

$$
\min \{ (z)^m p + (m')n, (z)^x p + (x')n, (z)^w p + (w')n \} = (z)^n
$$

Then

$\min$ is taken over all neighbors $v$ of $x$

Network Layer 4-86
Distance Vector Algorithm

\[ d^x(y) = \text{estimate of least cost from } x \text{ to } y \]

Node \( x \) knows cost to each neighbor \( y \):

\[ c(x,y) \]

Node \( x \) maintains distance vector \( D^x \):

\[ D^x = \{ d^x(y) : y \in N \} \]

For each neighbor \( v \) of \( x \), maintain:

\[ d^v(y) \rightarrow \text{min}(c(x,y) + d^v(y)) \]

When a node \( x \) receives new DV estimate from neighbor, it updates its own DV estimate from distance vector estimate to neighbors.

From time-to-time, each node sends its own distance vector estimate to neighbors.

Basic idea:

Distance Vector Algorithm (4)
Distance Vector Algorithm (5)

Iterative, asynchronous:
- Each local iteration caused by:
  - Local link cost change
  - DV update message from neighbor

Each node:
- Network Layer 4-89

Distributed:
- Each node notifies neighbors only when its DV changes:
  - x, y, z neighbors then notify their neighbors if necessary

Network Layer 4-90

\[
D_x(z) = \min\{c(x, z) + \min\{c(y, z), c(y', z)\}, c(x', y') + \min\{c(x, z), c(x', z)\}\}
\]

Cost from
\[
x \quad x \quad z
\]
\[
0 \quad 2 \quad 7
\]

Time

\[
D_x(y) = \min\{c(x, y) + \min\{c(y, z), c(y', z)\}, c(x', y') + \min\{c(x, y), c(x', y')\}\}
\]

Wait for (change in local link cost to)
- DV to any dest has changed, notify neighbors

Each node:
- DV update message from neighbor
- Local link cost change
- Diagnose only when its DV changes:
  - Each node notifies neighbors of their neighbors if necessary

Iterative, asynchronous: (G)
Distance Vector: Link cost changes

At time $t_0$, $y$ detects the link-cost change and updates its distance table.

It computes a new least cost to $x$ and sends its neighbors its $DV$. At time $t_1$, $x$ receives the update from $y$ and updates its table. At time $t_2$, $z$ receives the update from $x$ and updates its distance table. At time $t_3$, $y$ detects the link-cost change, updates its $DV$, and informs its neighbors.

If $DV$ changes, notify neighbors.

Distance vector updates routing info, recalculates node detects local link cost change:

**Distance Vector:**


table

"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 updates routing info, recalculates node detects local link cost change:

**Distance Vector:**


table

"good news travels fast"
Distance Vector: link cost changes

- Distance Vector:
  - 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
  - If Z routes through Y to
- Text
  - Algorithm stabilizes: see
  - 44 iterations before

Comparison of LS and DV algorithms

**Message complexity**
- LS:
  - Each node computes only
  - Incorrect link cost
  - Node can advertise:
  - DV:
  - Each node computes only
  - Incorrect path cost
  - Node can advertise

**Robustness:**
- What happens if router malfunctions?

**Speed of convergence**
- LS:
  - Convergence time varies
  - Node can advertise:
  - DV:
  - Exchange between neighbors only

**Network Layer 4-93**

Distance Vector: link cost changes

network
- Error propagation through others
  - Each node’s table used by
    - Incorrect path cost
  - DV node can advertise

DV:
- It’s own table
  - Each node computes only
  - Incorrect link cost
  - Node can advertise

LS:
- If router malfunctions?

Count to infinity problem:
- Will this completely solve route to X via Z
  - Z tells Y its (Z’s) distance
  - Get to X:
  - If Z routes through Y to

Poisoned reverse:
- Text
  - Algorithm stabilizes: see
  - 44 iterations before

"Count to infinity" problem:
- Bad news travels slow
  - Good news travels fast

Link cost changes: 0
Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.4 IP: Internet Protocol

4.5 Routing algorithms

- Link state
- Distance Vector
- RIP
- OSPF
- BGP

4.6 Routing in the Internet

- RIP
- OSPF
- BGP

4.7 Broadcast and multicast routing

- IPv4
- IPv6
- ICMP

Our routing study thus far - idealization

Hierarchical Routing

... not true in practice

Internet = network of networks

Each network admin may want to control routing in its own network

Administrative autonomy

Routing table exchange

with 200 million destinations:

- Can't store all destinations in routing table
- Routing table exchange would swamp links

Scale: 200 million destinations

... not true in practice
Hierarchical Routing

aggregate routers into regions, "autonomous systems" (AS) routers in same AS run same routing protocol

Gateway router

Direct link to router in another AS

Interconnected ASes

Forwarding table

Network Layer 4-97

Forwarding algorithm

Inter-AS routing

Intra-AS routing

Inter-AS & Intra-AS sets entries for external dests

Intra-AS sets entries for internal dests

configured by both intra- and inter-AS routing algorithm

AS1

AS2

AS3

Inter-AS routing protocol can run different Intra-AS protocols in different AS

"Intra-AS" routing

same routing protocol

routers in same AS run

systems (AS)

regions, "autonomous systems"

aggregate routers into Hierarchical Routing
Inter-AS routing tasks

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

Job of inter-AS routing!

Example: Setting forwarding table entry

Suppose AS1 learns (via inter-AS protocol) that subnet x is reachable via AS3 (gateway 1c) but not via AS2.

Router 1d determines from intra-AS routing info that its interface I is on the least cost path to 1c.

Installs forwarding table entry (x, I).

Network Layer 4-99

Job of inter-AS routing!

which one?

Gateway router, but forward packet to router should:

AS1: Destined outside of AS1

AS1 must:

AS1 receives datagram

Destination outside AS1

Which AS are reachable through AS2, through AS3?

which through

Receive through AS2 which dests are

AS1 receives datagram

Destination outside AS1

Which AS are reachable through AS2, through AS3?

which through

Receive through AS2 which dests are

AS1 must:

AS1 receives datagram

Destination outside AS1

Which AS are reachable through AS2, through AS3?

which through

Receive through AS2 which dests are

AS1 must:

AS1 receives datagram

Destination outside AS1

Which AS are reachable through AS2, through AS3?

which through

Receive through AS2 which dests are

AS1 must:

AS1 receives datagram

Destination outside AS1

Which AS are reachable through AS2, through AS3?

which through

Receive through AS2 which dests are

AS1 must:

AS1 receives datagram

Destination outside AS1

Which AS are reachable through AS2, through AS3?

which through

Receive through AS2 which dests are

AS1 must:

AS1 receives datagram

Destination outside AS1

Which AS are reachable through AS2, through AS3?

which through

Receive through AS2 which dests are

AS1 must:

AS1 receives datagram

Destination outside AS1

Which AS are reachable through AS2, through AS3?

which through

Receive through AS2 which dests are

AS1 must:

AS1 receives datagram

Destination outside AS1

Which AS are reachable through AS2, through AS3?

which through

Receive through AS2 which dests are

AS1 must:

AS1 receives datagram

Destination outside AS1

Which AS are reachable through AS2, through AS3?

which through

Receive through AS2 which dests are

AS1 must:
Example: Choosing among multiple ASes

- Suppose AS1 learns from inter-AS protocol that subnet x is reachable from AS3 and AS2.
- To configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest x.
- This is also job of inter-AS routing protocol.
- Hot potato routing: send packet towards closest of multiple gateways.
- Determine towards which gateway it should forward.
- Choose the gateway that has the smallest least cost.
- Use routing info from intra-AS protocol to determine costs of least-cost paths to each gateway.
- Choose the gateway with the smallest least cost.
- Determine from forwarding table the interface I that leads to least-cost gateway.
- Enter (x, I) in forwarding table.

Example: Choosing among multiple ASes

- Now suppose AS1 learns from inter-AS protocol that subnet x is reachable from AS3 and AS2.
- Hot potato routing: send packet towards closest of two routers.
Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.4 IP: Internet Protocol
   - Datagram format
   - IPv4 addressing
   - ICMP
   - IPv6

4.5 Routing algorithms
   - Link state
   - RIP
   - Distance Vector
   - Interior Gateway Routing Protocol (IGRP)
   - Open Shortest Path First (OSPF)
   - Routing Information Protocol (RIP)

4.6 Routing in the Internet
   - RIP
   - OSPF
   - BGP

4.7 Broadcast and multicast routing
   - 4.7 Broadcast and multicast routing
   - BGP
   - OSPF
   - RIP
   - Interior Gateway Protocol (IGP)

Intra-AS routing
   - Also known as Interior Gateway Protocols (IGP)
   - Most common Intra-AS routing protocols:
   - RIP
   - OSPF
   - IGRP

4.8 Internet routing
   - Hierarchical routing
   - Distance Vector
   - Link State
   - Routing algorithms
   - 4.5 Routing algorithms
   - 4.6 Routing in the Internet
   - 4.7 Broadcast and multicast routing

4.9 Network Layer

Network.
Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.4 IP: Internet Protocol

4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

4.6 Routing in the Network Layer

4.7 Broadcast and multicast routing

RIP (Routing Information Protocol)

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

From router A to subnets:

- Network Layer 4-106
- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops)

RIP (Routing Information Protocol)
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</th>
<th>Network</th>
<th>Next Router</th>
<th>Num. of hops to dest.</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>A</td>
<td>B</td>
<td>x</td>
</tr>
<tr>
<td>y</td>
<td>B</td>
<td>A</td>
<td>z</td>
</tr>
<tr>
<td>z</td>
<td>B</td>
<td>A</td>
<td>y</td>
</tr>
</tbody>
</table>

Routing/Forwarding table in D

ASes (also called autonomous systems) exchange distance vectors every 30 sec via Response Message.

RIP advertisements

- Each advertisement lists up to 25 destination subnets within the AS.
RIP: Link Failure and Recovery

If no advertisement heard after 180 sec -->

1. Neighbor/link declared dead
2. Routes via neighbor invalidated
3. New advertisements sent to neighbors
4. Neighbors in turn send new advertisements

If tables changed:

- Poison reverse used to prevent ping-pong loops
- Link failure info quickly propagates to entire net

(Infinite distance = 16 hops)

Example:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Network</th>
<th>Next Router</th>
<th>Num. of hops to dest.</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>A</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>y</td>
<td>B</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>z</td>
<td>B</td>
<td>A</td>
<td>7</td>
</tr>
<tr>
<td>w</td>
<td>A</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>z</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w</td>
<td>A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Routing/Forwarding Table in D

Advertise message from A to D

Network Layer 4-110
RIP Table processing

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

Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.4 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing

4.5 Routing algorithms

4.6 Routing in the Network Layer

4.7 Broadcast and multicast routing

4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6

4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP

4.4.2 Virtual circuit and datagrm networks

4.7 Broadcast and multicast routing

RIP Table processing

Repeated advertisements sent in UDP packets, periodically

Process called route-d (daemon)
OSPF (Open Shortest Path First)

- Hierarchical OSPF in large domains.
- Multiicast OSPF (MOSPF) uses same topology data.
- Security: all OSPF messages authenticated (to prevent malicious intrusion).
- Multiple same-cost paths allowed (only one path in RIP).
- Multiple link cost metrics for different TOS (e.g., satellite link cost set "low" for best effort; high for real time).
- For each link, multiple cost metrics for different TOS (via AS path, etc).

OSPF advertisements carry the following path data:

- Carried in OSPF messages directly over IP (rather than TCP flooding).
- Advertisements disseminated to entire AS (via router).
- OSPF advertisement carries one entry per neighbor.
- Route computation using Dijkstra's algorithm.
- Topology map at each node.
- LS packet dissemination uses link state algorithm.
- "open": publicly available.

OSPF advertisement carries one entry per neighbor.
- Route computation using Dijkstra's algorithm.
- Topology map at each node.
- LS packet dissemination uses link state algorithm.
- "open": publicly available.
Hierarchical OSPF

Network Layer 4-115

- Two-level hierarchy: Local area, Backbone
  - Backbone routers: Run OSPF routing limited to
    - Backbone
    - In own area, advertise to other area
  - Area border routers: "Summarize" distances to nets
    - Each node has detailed area topology; only know direction (shortest path) to nets in other areas.
    - Each node has detailed area topology; only know direction (shortest path) to nets in other areas.
  - Area border routers: "Summarize" distances to nets to other area border routers.
  - Boundary routers: Connect to other ASs.

Internet Inter-AS Routing: BGP

BGP (Border Gateway Protocol): the de facto standard

- Allows subject to advertise its existence to rest of Internet: "I am here"
- Propagates reachability information and policy
- Determines "good" routes to subnets based on internal routers
- Propagates reachability information to all ASs
- Obtains subject reachability information from neighboring ASs

BGP provides each AS a means to:

1. Obtain subject reachability information from neighboring ASs.
2. Propagate reachability information to all ASs.
3. Determine "good" routes to subnets based on reachability information and policy.

Internet Inter-AS Routing: BGP

- IPv6
- ICMP
- IPv4 Addressing
- Datagram Format
- Protocol
- Router
- What's inside a datagram network
- 4.1 Virtual Circuit and
- 4.2 Virtual Circuit and
- 4.3 What's inside a datagram network
- 4.4 IP: Internet
- 4.5 Routing Algorithms
- 4.6 Routing Algorithms
- 4.7 Broadcast and
- Broadcast and
- BGP
- OSPF
- RIP
- Multicast Routing
- Hierarchical Routing
- Distinct Vector
- Link State
- 4.6 Routing in the Internet
- 4.7 Broadcast and
- Multi-cast Routiing
- 4.1 Introduction
- 4.2 Virtual Circuit and
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.

AS1 promises it will forward datagrams towards prefix.

When AS2 advertises a prefix to AS1:

AS2 can aggregate prefixes in its advertisement

AS2 promises it will forward datagrams towards that prefix.

When router learns of new prefix, it creates entry for prefix in its forwarding table.

BGP sessions

Distributing reachability info

AS3

AS1

AS2

AS3

Distributing reachability info

AS3

AS1

AS2

AS3

AS1
BGP route selection

Path attributes & BGP routes

- advertised prefix includes BGP attributes.

prefix + attributes = "route"

- two important attributes:
  - AS-PATH: contains ASs through which prefix advertisement has passed: e.g., AS 67, AS 17
  - NEXT-HOP: indicates specific internal-AS router to next-hop AS.

When gateway router receives route, uses import policy to accept/decline.

BGP route selection

- router may learn about more than 1 route to same prefix. Router must select route.

elimination rules:

- local preference value attribute:
- shortest AS-PATH
- closest NEXT-HOP router: hot potato routing decision

4. additional criteria

- shortest AS-PATH
- closest NEXT-HOP router: hot potato routing
- local preference value attribute: policy
- additional criteria

Network Layer 4-121
BGP messages exchanged using TCP.

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

**BGP routing policy**

- A, B, C are provider networks
- X, 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
- X does not want to route from B via X to C

Network Layer 4.124
BGP routing policy (2)

Network Layer 4-125

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

Performance:
- Traffic: hierarchical routing saves table size, reduced update
- Scale: Inter-AS: single admin, so no policy decisions needed

Routing: Inter-AS: routed, who routes through its net. Intra-AS: admin wants control over how its traffic

Why different Intra- and Inter-AS routing?

Legend:
- Customer network:
- Provider network:

B wants to route only to/from its customers!
- B wants to force C to route to W via A
- Since neither W nor C are B's customers
- No way! B gets no "revenue" for routing C AW to C

Should B advertise path BAW to C?
- B advertises path BAW to X
- A advertises path AW to B

Network Layer 4-126

Policy:
- Inter-AS: admin wants control over how its traffic
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
  - Hierarchical routing
  - Distance vector
  - RIP
  - OSPF
  - BGP

4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP

4.7 Broadcast and multicast routing
  - Broadcast and multicast routing
  - BGP
  - OSPF
  - RIP

4.8 Internet:

4.9 What’s inside a datagram network
  - IPv4
  - Protocol
  - Router

4.10 Introduction
In-network duplication

Flooding: when a node receives a broadcast packet, it sends a copy to all its neighbors.

Problems: cycles and broadcast storms.

Controlled flooding: a node only broadcasts a packet if it has not already broadcast it.

Nodes keep track of packet IDs already broadcast. OR Reverse Path Forwarding (RPF): only forward packets if the reverse path is non-redundant.

First construct a spanning tree.

Spanning tree

Nodes forward copies only along spanning tree.

No redundant packets received by any node.

Controlled flooding: node only forwards a packet if it arrived on the shortest path between the node and the source.

Problems: cyclic broadcast storms.

In-network duplication
Spanning Tree: Creation

- Each node sends unicast join message to center node
- Message forwarded until it arrives at a node already belonging to spanning tree
- Goal: find a tree (or trees) connecting routers having local multicast group members

Multicast Routing: Problem Statement

- Goal: find a tree (or trees) connecting routers having local multicast group members
- Shared tree: same tree used by all group members
- Source-based: different tree from each sender to receivers
- Shared trees: not all paths between routers used
- Source-based trees: not all paths between routers used

Network Layer 4-131
Approaches for building mcast trees

**Approaches:**
- **Source-based tree:** one tree per source
  - Shortest path trees
  - Reverse path forwarding
- **Group-shared tree:** group uses one tree
  - Minimal spanning (Steiner)
  - Center-based trees

**Shortest Path Tree**

---

Approaches for building multicast trees

---

protocols adopting these approaches.

...we first look at basic approaches, then specific...

- Center-based trees
- Minimal spanning (Steiner)
- Group-shared tree: group uses one tree
- Reverse path forwarding
- Source-based tree: one tree per source
- Shortest path trees

**Legend**
- S: source

**Diagram Notes**
- Link used for forwarding, i indicates order link
- Router with attached group member
- Router with no attached group member
- Link used for forwarding, i indicates order link added by algorithm
Reverse Path Forwarding relies on the router's knowledge of the unicast shortest path from it to the sender.

Each router has a simple forwarding behavior:

- If (multicast datagram received on an incoming link)
  - then flood the datagram onto all outgoing links on the shortest path back to the center
  - else ignore the datagram

- Rely on the router's knowledge of unicast shortest paths from it to the sender.

Reverse Path Forwarding example:

```
then flood datagram onto all outgoing links
else ignore datagram
```
Reverse Path Forwarding: pruning

- pruning messages sent upstream by router with no downstream group members

**Legend**

- downstream group members
- prune messages sent upstream by router with no group members
- no need to forward datagrams down subtree
- forwarding tree contains subtrees with no group members
- computational complexity
  - not used in practice
  - excellent heuristics exists
- problem is NP-complete

**Steiner Tree:** minimum cost tree connecting all routers with attached group members

**Shared-Steiner Tree**
Center-based trees

Suppose R6 chosen as center:

**Center-based Trees: an Example**

Legend:

- **Group member** with no attached group member
- **Path order in which join messages generated**
  - Path taken by join-msg becomes new branch of tree for this router
  - Join-msg either hits existing tree branch for this center, or arrives at center
  - Join-msg "processed" by intermediate routers and forwarded towards center
  - Edge router sends unicast join-msg addressed to center
  - Single delivery tree shared by all one router identified as "center" of tree
Internet Multicasting Routing: DVMRP

- Distance vector multicast routing protocol, RFC1075
- Flood and prune: Reverse path forwarding, RFC1075
- Source-based tree
- Routes not wanting group: Send upstream prune everywhere via RPF
- Initial datagram to mcast group flooded
- No assumptions about underlying unicast
- Mbone routing done using DVMRP
- Commonly implemented in commercial routers
- Soft state: DVMRP router periodically (1 min.) "forgets" branches are pruned:
  - RPF tree prunes downstream: DVMRP routes
  - RPF tree based on DVMRP's own routing tables
  - Source-based tree
  - Reprune or else continue to receive data
  - Downstream router: Reprune or else continue to receive data
  - Following IGMP join: Prune to free tree
  - Routers not wanting group: Send upstream prune

DVMRP: continued...
Tunneling

Q: How to connect “islands” of multicast routers in a “sea” of unicast routers?

mcast datagram encapsulated inside “normal” (non-multicast-addressed) datagram

normal IP datagram sent thru “tunnel” via regular IP unicast to receiving mcast router

receiving mcast router unencapsulates to get mcast datagram

PIM: Protocol Independent Multicast

not dependent on any specific underlying unicast routing algorithm (works with all)

two different multicast distribution scenarios:

Dense:
- group members densely packed, in “close” proximity.
- bandwidth plentiful
- group members with # networks interconnect

Sparse:
- group members “widely” packed, in “close” proximity.
- bandwidth not plentiful
- # networks with group members

Dense: sparse

Routing algorithm (works with all)

PIM: Protocol Independent Multicast

Logical topology

Physical topology

Tunneling

Q: How to connect “islands” of multicast routers?
Consequences of Sparse-Dense Dichotomy:

**Dense**
- group membership by routers assumed until routers explicitly prune data-driven construction on mcast tree (e.g., RPF)
- bandwidth and non-group-router processing profligate

**Sparse**
- no membership until routers explicitly join
- group membership by consensus of Sparse-Dense Dichotomy:

PIM-Dense Mode

- flood-and-prune RPF, similar to DVMRP
- less complicated (less efficient) downstream construction of mcast tree (e.g., center-based)
- less complicated (less efficient) downstream for incoming datagram
- underlying unicast protocol provides RPF info

PIM-Dense Mode has protocol mechanism for router to detect it underlying routing algorithm

- flood than DVMRP reduces reliance on underlying unicast protocol provides RPF info
- flood than DVMRP reduces reliance on underlying routing algorithm

is a leaf-node router

- has protocol mechanism for router to detect it underlying routing algorithm

- has protocol mechanism for router to detect it underlying routing algorithm

- has protocol mechanism for router to detect it underlying routing algorithm
PIM - Sparse Mode

Sender(s):
- Unicast data to RP, which distributes down RP-rooted tree
- RP can extend multicast tree upstream to source
- RP can send stop msg if no attached receivers
  "no one is listening!"
- RP can extend multicast which distributes down unicast data to RP,
  all data multicast from rendezvous point

Intermediate routers:
- Update state and forward join after joining via RP,

Increased performance:
- Less concentration,
- Shorter paths,
- Source-specific tree

Center-based approach:
- RP sends join msg to rendezvous point
Chapter 4: Summary

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What’s inside a router

4.5 Routing algorithms

4.6 Routing in the Network Layer

4.4 IP: Internet Protocol

4.7 Broadcast and multicast routing

4.6 Routing in the Internet

4.3 What’s inside a router

4.2 Virtual circuit and datagram networks

IPv6

ICMP

IPv4 addressing

Datagram format

Protocol

RIP

OSPF

BGP

Hierarchical routing

Distance Vector

Link State

Routing algorithms

Broadcast and multicast routing

4.7 Broadcast and multicast routing

4.2 Virtual circuit and datagram networks

4.1 Introduction