Network Layer II

Dmitri Loguinov
Texas A&M University

November 2, 2023
Chapter 4: Roadmap

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
Router Architecture Overview

Two key router functions:
• Run routing algorithms/protocols (RIP, OSPF, BGP)
• Forward datagrams from incoming to outgoing link
  - Terminology: port = interface capable of sending/receiving

![Diagram showing router architecture with ports and switch fabric hardware]
Input Port (Queue) Functions

Decentralized switching:
- Given datagram destination, look up output port using forwarding table in input port memory
- **Goal**: complete input port processing at “line speed”
- **Queuing**: if datagrams arrive faster than forwarding rate into switch fabric

Physical layer:
- bit-level reception

Data link layer:
- (e.g., Ethernet, ATM, Token Ring, 802.11b): see ch. 5
Switching Via Memory

First generation routers (1960s-mid 1980s):

- Traditional computers with switching under direct control of CPU
- Packet copied to system memory
- Speed limited by CPU, memory latency/bandwidth, and bus bandwidth (two bus crossings per datagram)
- Honeywell 316 (1969)
Switching Via a Bus

- Datagram from input port memory to output port memory via a shared bus
- **Bus contention**: switching speed limited by bus bandwidth
- 1 Gbps bus in Cisco 1900: sufficient speed for access and small enterprise networks (not ISPs)
Switching Via An Interconnection Network

- Overcomes bus bandwidth limitations
  - Crossbar: packets transmitted in parallel as long as they do not occupy the same horizontal or vertical bus
- Cisco 12000 (1996): uses an interconnection network
  - CRS-X (2013): 1600 lbs, 84” rack, 7.6 KWatt, 800 Gbps/slot
  - 16 slots/rack = 12.8 Tbps
  - Up to 72 racks (922 Tbps)
Output Ports

• **Buffering/queuing** required when datagrams arrive from fabric faster than the transmission rate

• **Scheduling discipline** chooses among queued datagrams for transmission
  - Customer traffic: single FIFO drop-tail queue
  - ISP traffic: multiple queues with WRR or priority queuing
Output Port Queuing

- Buffering when arrival rate via switch fabric exceeds output line speed
  - Queuing delay and loss due to output buffer overflow
- Switch fabric often faster than individual ports
  - Produces large bursts of arrivals into output queues
Input Port Queuing

- Reasons for input-port queuing:
  - Head-of-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward

- Queuing delay and loss due to input buffer overflow!
  - How likely is this compared to output port queuing/loss?
Chapter 4: Roadmap

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
The Internet Network Layer

Host and router network layer functions:

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

- **IP protocol**
  - Datagram format
  - Addressing conventions

- **ICMP protocol**
  - Error reporting
  - Ping, traceroute

- **IGMP protocol**
  - Multicast

- **Transport layer: TCP, UDP**

- **Network layer**

- **Link layer**

- **Physical layer**
Chapter 4: Roadmap

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
4.6 Routing in the Internet
4.7 Broadcast and multicast routing
**IPv4 Datagram Format**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP protocol version number</td>
<td>The IP protocol version number.</td>
</tr>
<tr>
<td>Header length</td>
<td>The header length in 4-byte words.</td>
</tr>
<tr>
<td>QoS requested</td>
<td>QoS (Quality of Service) requested.</td>
</tr>
<tr>
<td>Max number remaining hops</td>
<td>The maximum number of remaining hops, decremented at each router.</td>
</tr>
<tr>
<td>Upper layer protocol</td>
<td>The upper layer protocol to deliver payload to.</td>
</tr>
<tr>
<td>How much overhead with TCP?</td>
<td>The overhead with TCP is 40 bytes. Annual: 20 bytes of TCP + 20 bytes of IP = 40 bytes.</td>
</tr>
<tr>
<td>Total datagram length</td>
<td>The total length of the datagram in bytes.</td>
</tr>
<tr>
<td>For fragmentation/reassembly</td>
<td>Options (if any) E.g. timestamp, record route taken, specify list of routers to visit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ver</td>
<td>IP version number.</td>
</tr>
<tr>
<td>hdr len</td>
<td>Header length.</td>
</tr>
<tr>
<td>type of service</td>
<td>Type of service.</td>
</tr>
<tr>
<td>length</td>
<td>Total length of the datagram in bytes.</td>
</tr>
<tr>
<td>16-bit identifier</td>
<td>The 16-bit identifier field.</td>
</tr>
<tr>
<td>flags</td>
<td>Flags field.</td>
</tr>
<tr>
<td>fragment offset</td>
<td>Fragment offset.</td>
</tr>
<tr>
<td>time to live</td>
<td>Time to live.</td>
</tr>
<tr>
<td>upper layer</td>
<td>Upper layer.</td>
</tr>
<tr>
<td>Internet checksum</td>
<td>Internet checksum.</td>
</tr>
<tr>
<td>32 bit source IP address</td>
<td>32-bit source IP address.</td>
</tr>
<tr>
<td>32 bit destination IP address</td>
<td>32-bit destination IP address.</td>
</tr>
<tr>
<td>Options (if any)</td>
<td>Options (if any), such as timestamp, record route taken.</td>
</tr>
<tr>
<td>data</td>
<td>Data field (variable length, typically a TCP or UDP segment)</td>
</tr>
</tbody>
</table>

**Example:**

- **IPv4 Datagram Format**
- **Header length** (in 4-byte words)
- **QoS requested**
- **Max number remaining hops** (decremented at each router)
- **Upper layer protocol** to deliver payload to
- **How much overhead with TCP?**
  - 20 bytes of TCP
  - 20 bytes of IP
  - = 40 bytes
IP Fragmentation & Reassembly

• Network links have varying MTUs (maximum transmission units) – largest possible link-level frames
  - Different link types, different MTUs (most common 1500)
• 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

fragmentation:
in: one large datagram
out: 3 smaller datagrams

reassembly
## IP Fragmentation and Reassembly

### Example
- 4000 byte datagram (including IP header)
- MTU = 1500 bytes

1480 bytes in payload

offset is in 8-byte words: 185 = 1480/8

### Diagram

<table>
<thead>
<tr>
<th></th>
<th>length</th>
<th>ID</th>
<th>fragflag</th>
<th>offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4000</td>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1500</td>
<td>x</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1500</td>
<td>x</td>
<td>1</td>
<td>185</td>
</tr>
<tr>
<td>4</td>
<td>1040</td>
<td>x</td>
<td>0</td>
<td>370</td>
</tr>
</tbody>
</table>

One large datagram becomes several smaller datagrams

Chapter 4: Roadmap

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
4.6 Routing in the Internet
4.7 Broadcast and multicast routing
**IP Addressing: Introduction**

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

- **Interface**: connection between host/router and physical link
  - Also called a *port*
  - Routers have many interfaces

- Can hosts have multiple interfaces?
  - Yes, it’s called *multi-homing*
Subnets

- **IP address:**
  - Subnet prefix: \( k \) bits
  - Host suffix: \( 32-k \) remaining bits

- **What’s a subnet (LAN)?**
  - Network composed of devices with the same subnet prefix of IP address
  - Can physically reach each other without intervening router
Subnets

Recipe

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

Subnet mask:
- 255.255.255.0
- or /24
Subnets

How many?
In the early Internet, only subnets with 8, 16, or 24 bit prefixes were allowed (“class A, B, C” networks)

This was inflexible and wasteful as well

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

```
11001000  00010111
00010000  00000000
```

**200.23.16.0/23**
IP Addresses: How to Get One?

Q: How does a host get an IP address?

• Either hard-coded by system admin in a file
  - Windows: Control-panel → network → configuration → tcp/ip → properties
  - Linux: /etc/rc.config

• Or dynamically assigned by DHCP (Dynamic Host Configuration Protocol)
  - “Plug-and-play” (more in Chapter 5)
**IP Addresses: How to Get One?**

**Q:** How does a *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>

- **Task:** split this ISP into one /21, three /23, and eight /26
Hierarchical Addressing: Route Aggregation

Hierarchical addressing allows efficient advertisement of routing information:

Organization 0
200.23.16.0/23

Organization 1
200.23.18.0/23

Organization 2
200.23.20.0/23

Organization 7
200.23.30.0/23

ISP-A

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

ISP-B

“Send me anything with addresses beginning with 199.31.0.0/16”

Internet
Hierarchical Addressing: More Specific Routes

ISP-B 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

ISP-A

ISP-B

“Send me anything in 200.23.16.0/20”

“Send me anything in 199.31.0.0/16 or 200.23.18.0/23”

Internet
Q: How does an ISP get a block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers assigns IPs to regional registries
   - These are ARIN (North/South America), RIPE (Europe), APNIC (Asia-Pacific), and AfriNIC (Africa)

• These registries process ISP and user requests for subnet space
   - Also manage DNS and resolve disputes

• Quiz #3 covers
   - Chapter 3: P7-9, 22-24, 26-28, 31-37, 40-41, 43-49
   - Chapter 4: P1-17 (including today’s lecture)