What is the class about?

- **Course description and syllabus:**
  - [http://www.nyu.edu/classes/jcf/csci-ga.2262-001/](http://www.nyu.edu/classes/jcf/csci-ga.2262-001/)
  - [http://cs.nyu.edu/courses/spring15/CSCI-GA.2262-001/index.html](http://cs.nyu.edu/courses/spring15/CSCI-GA.2262-001/index.html)

- **Textbooks:**
    James F. Kurose, Keith W. Ross
    Addison Wesley
Course Overview

- Computer Networks and the Internet
- Application Layer
- Fundamental Data Structures: queues, ring buffers, finite state machines
- Data Encoding and Transmission
- Local Area Networks and Data Link Control
- Wireless Communications
- Packet Switching
- OSI and Internet Protocol Architecture
- Congestion Control and Flow Control Methods
- Internet Protocols (IP, ARP, UDP, TCP)
- Network (packet) Routing Algorithms (OSPF, Distance Vector)
- IP Multicast
- Sockets
Course Approach

- Introduction to Basic Networking Concepts (Network Stack)
- Origins of Naming, Addressing, and Routing (TCP, IP, DNS)
- Physical Communication Layer
- MAC Layer (Ethernet, Bridging)
- Routing Protocols (Link State, Distance Vector)
- Internet Routing (BGP, OSPF, Programmable Routers)
- TCP Basics (Reliable/Unreliable)
- Congestion Control
- QoS, Fair Queuing, and Queuing Theory
- Network Services – Multicast and Unicast
- Extensions to Internet Architecture (NATs, IPv6, Proxies)
- Network Hardware and Software (How to Build Networks, Routers)
- Overlay Networks and Services (How to Implement Network Services)
- Network Firewalls, Network Security, and Enterprise Networks
Networks Part 1 Session in Brief

- Understand principles behind network layer services:
  - Network layer service models
  - Forwarding versus routing
  - How a router works
- Instantiation, implementation in the Internet
- Conclusion
Icons / Metaphors

Information

Common Realization

Knowledge/Competency Pattern

Governance

Alignment

Solution Approach
Agenda

1. Session Overview
2. Networks Part 1
3. Summary and Conclusion
Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
Switching Networks

- Long distance transmission is typically done over a network of switched nodes
- Nodes not concerned with content of data
- End devices are stations
  - Computer, terminal, phone, etc.
- A collection of nodes and connections is a communications network
- Data routed by being switched from node to node
Two different switching technologies
  - Circuit switching
  - Packet switching
Circuit Switching

- Dedicated communication path between two stations (during conversation)
- Three phases
  - Establish
  - Transfer
  - Disconnect
- Must have switching capacity and channel capacity to establish connection
- Must have intelligence to work out routing
Circuit Switching - Issues

- Circuit switching is inefficient (designed for voice)
  - Resources dedicated to a particular call
  - Much of the time a data connection is idle
  - Data rate is fixed
    - Both ends must operate at the same rate
- Set up (connection) takes time
- Once connected, transfer is transparent
Packet Switching – Basic Operation

- Data transmitted in small packets
  - Typically 1000 octets
  - Longer messages split into series of packets
  - Each packet contains a portion of user data plus some control info

- Control info
  - Routing (addressing) info

- Packets are received, stored briefly (buffered) and passed on to the next node
  - Store and forward
Use of Packets

Application data

control information (packet header)

packet

Packet-Switching Network
- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in every host, router
- router examines header fields in all IP datagrams passing through it
Two Key Network-Layer Functions

- **forwarding**: move packets from router’s input to appropriate router output
- **routing**: determine route taken by packets from source to dest.

  » **routing algorithms**

  **analogy:**

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

Routing algorithm

<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

Diagram showing the routing and forwarding process.
- 3rd important function in some network architectures:
  - ATM, frame relay, X.25
- before datagrams flow, two end hosts and intervening routers establish virtual connection
  - routers get involved
- network vs transport layer connection service:
  - network: between two hosts (may also involve intervening routers in case of VCs)
  - transport: between two processes
Q: What *service model* for “channel” transporting datagrams from sender to receiver?

Example services for individual datagrams:
- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

Example services for a flow of datagrams:
- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing
## Network layer service models

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Service Model</th>
<th>Guarantees?</th>
<th>Bandwidth</th>
<th>Loss</th>
<th>Order</th>
<th>Timing</th>
<th>Congestion feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>best effort</td>
<td>none</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no (inferred via loss)</td>
</tr>
<tr>
<td>ATM</td>
<td>CBR</td>
<td>constant rate</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no congestion</td>
</tr>
<tr>
<td>ATM</td>
<td>VBR</td>
<td>guaranteed rate</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no congestion</td>
</tr>
<tr>
<td>ATM</td>
<td>ABR</td>
<td>guaranteed minimum</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ATM</td>
<td>UBR</td>
<td>none</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
Network layer connection and connection-less service

- datagram network provides network-layer connectionless service
- VC network provides network-layer connection service
- analogous to the transport-layer services, but:
  - service: host-to-host
  - no choice: network provides one or the other
  - implementation: in network core
Virtual circuits

“source-to-dest path behaves much like telephone circuit”
  » performance-wise
  » network actions along source-to-dest path

- call setup, tear down for each call *before* data can flow
- 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)
a VC consists of:

1. path from source to destination
2. VC numbers, one number for each link along path
3. entries in forwarding tables in routers along path
   - packet belonging to VC carries VC number (rather than dest address)
   - VC number can be changed on each link.
     » New VC number comes from 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 to setup, maintain, teardown VC
- used in ATM, frame-relay, X.25
- not used in today's Internet
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
<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 00011001 11111111</td>
<td>2</td>
</tr>
<tr>
<td>otherwise</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 10100001  **Which interface?**
- **DA:** 11001000 00010111 00011000 10101010  **Which interface?**
Datagram or VC network: why?

Internet (datagram)
- data exchange among computers
  - “elastic” service, no strict timing req.
- “smart” end systems (computers)
  - can adapt, perform control, error recovery
  - simple inside network, complexity at “edge”
- many link types
  - different characteristics
  - uniform service difficult

ATM (VC)
- evolved from telephony
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- “dumb” end systems
  - telephones
  - complexity inside network
Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
Two key router functions:

- run routing algorithms/protocol (RIP, OSPF, BGP)

- for

output

input port

switching fabric

routing processor

output port
Input Port Functions

Decentralized switching:
- given datagram dest., lookup 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
(see Textbook Chapter 5)
Three types of switching fabrics
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)
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
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers
Switching Via An Interconnection Network

- overcome bus bandwidth limitations
- Banyan networks, other interconnection nets initially developed to connect processors in multiprocessor
- advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000: switches 60 Gbps through the interconnection network
- **Buffering** required when datagrams arrive from fabric faster than the transmission rate
- **Scheduling discipline** chooses among queued datagrams for transmission
Output port queueing

- buffering when arrival rate via switch exceeds output line speed
- queueing (delay) and loss due to output port buffer overflow!
How much buffering?

- RFC 3439 rule of thumb: average buffering equal to “typical” RTT (say 250 msec) times link capacity C
  - e.g., C = 10 Gps link: 2.5 Gbit buffer
- Recent recommendation: with N flows, buffering equal to
  \[
  \frac{\text{RTT} \cdot C}{\sqrt{N}}
  \]
- Fabric slower than input ports combined -> queueing may occur at input queues
- **Head-of-the-Line (HOL) blocking**: queued datagram at front of queue prevents others in queue from moving forward
- *queueing delay and loss due to input buffer overflow!*

![Diagram showing input port queuing and HOL blocking](image)
Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
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

Network layer

physical layer
Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
### IP datagram format

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP protocol version</td>
<td>Number of IP protocol version.</td>
</tr>
<tr>
<td>length</td>
<td>Length of IP header (bytes).</td>
</tr>
<tr>
<td>“type” of data</td>
<td>Type of data, typically a TCP or UDP segment.</td>
</tr>
<tr>
<td>16-bit identifier</td>
<td>Unique identifier for datagram fragmentation.</td>
</tr>
<tr>
<td>flgs</td>
<td>Flags for fragmentation/ reassembly.</td>
</tr>
<tr>
<td>fragment offset</td>
<td>Offset for fragmentation.</td>
</tr>
<tr>
<td>time to live</td>
<td>Time remaining for datagram.</td>
</tr>
<tr>
<td>upper layer</td>
<td>Upper layer protocol to deliver payload to head.</td>
</tr>
<tr>
<td>header length</td>
<td>Length of header (bytes).</td>
</tr>
<tr>
<td>header length</td>
<td>Length of header (bytes).</td>
</tr>
<tr>
<td>header checksum</td>
<td>Checksum for header.</td>
</tr>
<tr>
<td>source IP address</td>
<td>32-bit source IP address.</td>
</tr>
<tr>
<td>destination IP address</td>
<td>32-bit destination IP address.</td>
</tr>
<tr>
<td>Options (if any)</td>
<td>Options for datagram, e.g., timestamp, record route taken.</td>
</tr>
<tr>
<td>data</td>
<td>Variable length data, typically a TCP or UDP segment.</td>
</tr>
</tbody>
</table>

#### How much overhead with TCP?
- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead
IP Fragmentation & Reassembly

- Network links have MTU (max. transfer size) - largest possible link-level frame.
  - Different link types, different MTUs
- Large IP datagram divided ("fragmented") within net
  - One datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments
Example

- 4000 byte datagram
- MTU = 1500 bytes

1480 bytes in data field

Offset = 1480/8

One large datagram becomes several smaller datagrams

<table>
<thead>
<tr>
<th>length</th>
<th>ID</th>
<th>fragflag</th>
<th>offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1500</td>
<td>x</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1500</td>
<td>x</td>
<td>1</td>
<td>185</td>
</tr>
<tr>
<td>1040</td>
<td>x</td>
<td>0</td>
<td>370</td>
</tr>
</tbody>
</table>
Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- **IP: Internet Protocol**
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
**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

223.1.1.1 = 11011111 00000001 00000001 00000001

```
223 1 1 1 1
```
- **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
Recipe

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

Subnet mask: /24
Subnets

How many?
CIDR: Classless InterDomain Routing

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

```
11001000  00010111  00010000  00000000
```

200.23.16.0/23
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”
**Goal:** allow host to *dynamically* obtain its IP address from network server when it joins network
   - Can renew its lease on address in use
   - Allows reuse of addresses (only hold address while connected an “on”)
   - Support for mobile users who want to join network (more shortly)

**DHCP overview:**
- host broadcasts “DHCP discover” msg [optional]
- DHCP server responds with “DHCP offer” msg [optional]
- host requests IP address: “DHCP request” msg
- DHCP server sends address: “DHCP ack” msg
DHCP client-server scenario

DHCP server

arriving DHCP client needs address in this network
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
DHCP can return more than just allocated IP address on subnet:

» address of first-hop router for client
» name and IP address of DNS sever
» network mask (indicating network versus host portion of address)
DHCP: example

- connecting laptop needs its IP address, addr of first-hop router, addr of DNS server: use DHCP
- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet
- Ethernet frame broadcast (dest: FFFFFFFF) on LAN, received at router running DHCP server
- Ethernet demux’ed to IP demux’ed, UDP demux’ed to DHCP
- DCP server formulates DHCP ACK containing client’s IP address, IP address of first-hop router for client, name & IP address of DNS server

- encapsulation of DHCP server, frame forwarded to client, demux’ing up to DHCP at client

- client now knows its IP address, name and IP address of DNS server, IP address of its first-hop router
DHCP: wireshark output
(home LAN)

Message type: **Boot Request (1)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Request
Option: (61) Client identifier
  Length: 7; Value: 010016D323688A;
Hardware type: Ethernet
  Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
  Length: 11; Value: 010F03062C2E2F1F21F92B
  1 = Subnet Mask; 15 = Domain Name
  3 = Router; 6 = Domain Name Server
  44 = NetBIOS over TCP/IP Name Server
  ......
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:

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

ISPs-R-Us has a more specific route to Organization 1

- **Organization 0**: 200.23.16.0/23
- **Organization 2**: 200.23.20.0/23
- **Organization 7**: 200.23.30.0/23
- **Organization 1**: 200.23.18.0/23

Fly-By-Night-ISP

- “Send me anything with addresses beginning 200.23.16.0/20”
- “Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23”

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

**A:** ICANN: Internet Corporation for Assigned Names and Numbers

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes
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)
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).
Implementation: NAT router must:

» **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 #) as destination addr.

» **remember (in NAT translation table)** every (source IP address, port #) to (NAT IP address, new port #) translation pair

» **incoming datagrams: replace** (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table
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

NAT translation table

<table>
<thead>
<tr>
<th>WAN side addr</th>
<th>LAN side addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>138.76.29.7, 5001</td>
<td>10.0.0.1, 3345</td>
</tr>
<tr>
<td>……</td>
<td>……</td>
</tr>
</tbody>
</table>

3: Reply arrives dest. address: 138.76.29.7, 5001

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345
NAT: Network Address Translation

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, eg, P2P applications
  - address shortage should instead be solved by IPv6
client wants to connect to server with address 10.0.0.1

- server address 10.0.0.1 local to LAN (client can’t use it as destination addr)
- only one externally visible NATted address: 138.76.29.7

solution 1: statically configure NAT to forward incoming connection requests at given port to server

- e.g., (123.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000
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 (with lease times)

i.e., automate static NAT port map configuration
solution 3: relaying (used in Skype)
  » NATted client establishes connection to relay
  » External client connects to relay
  » relay bridges packets between connections
Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
ICMP: Internet Control Message Protocol

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>echo reply (ping)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>dest. network unreachable</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>dest host unreachable</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>dest protocol unreachable</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>dest port unreachable</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>dest network unknown</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>dest host unknown</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>source quench (congestion control - not used)</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>echo request (ping)</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>route advertisement</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>router discovery</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>TTL expired</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>bad IP header</td>
</tr>
</tbody>
</table>
Traceroute and ICMP

- Source sends series of UDP segments to dest
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number

- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router& IP address

- When ICMP message arrives, source calculates RTT

- Traceroute does this 3 times

Stopping criterion

- UDP segment eventually arrives at destination host
- Destination returns ICMP “host unreachable” packet (type 3, code 3)
- When source gets this ICMP, stops.
Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - 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
**Priority:** identify priority among datagrams in flow

**Flow Label:** identify datagrams in same “flow.”
(concept of “flow” not well defined).

**Next header:** identify upper layer protocol for data

<table>
<thead>
<tr>
<th>ver</th>
<th>pri</th>
<th>flow label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>payload len</td>
<td>next hdr</td>
<td>hop limit</td>
</tr>
<tr>
<td>source address</td>
<td>(128 bits)</td>
<td></td>
</tr>
<tr>
<td>destination address</td>
<td>(128 bits)</td>
<td></td>
</tr>
<tr>
<td>data</td>
<td></td>
<td>32 bits</td>
</tr>
</tbody>
</table>
Other Changes from IPv4

- **Checksum**: removed entirely to reduce processing time at each hop
- **Options**: allowed, but outside of header, indicated by “Next Header” field
- **ICMPv6**: new version of ICMP
  - additional message types, e.g. “Packet Too Big”
  - multicast group management functions
Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneous
  - 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
Tunneling (1/2)

Logical view:

Physical view:
Tunneling (2/2)

Logical view:

Physical view:

Flow: X
Src: A
Dest: F
data

Flow: X
Src: A
Dest: F
data

Flow: X
Src: A
Dest: F
data

A-to-B: IPv6

B-to-C: IPv6 inside IPv4

E-to-F: IPv6

IPv6 inside IPv4
Agenda

1. Session Overview
2. Networks Part 1
3. Summary and Conclusion
Summary

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
Assignments & Readings

- Readings
  - Chapter 4
- Assignments #7 and 8