

CS 580 Client-Server Programming
Fall Semester, 2012
Doc 2 Intro to Client-Server
Aug 30, 2012

Copyright ©, All rights reserved. 2012 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

Reading

Java

Java Network Programming, Harold 3rd Ed,

Chapter 2 - Network Basics

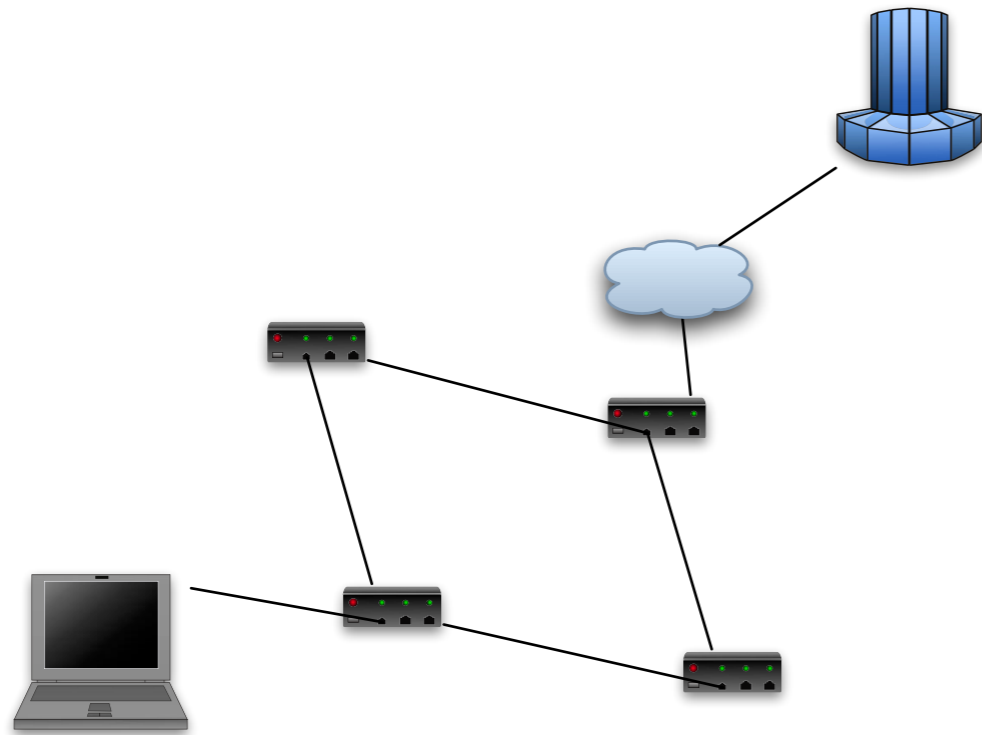
Chapter 4 Streams

Chapter 9 Sockets for Clients

References

Wikipedia, various articles, explicit references on individual slides

Network Overview



Messages divided into packets

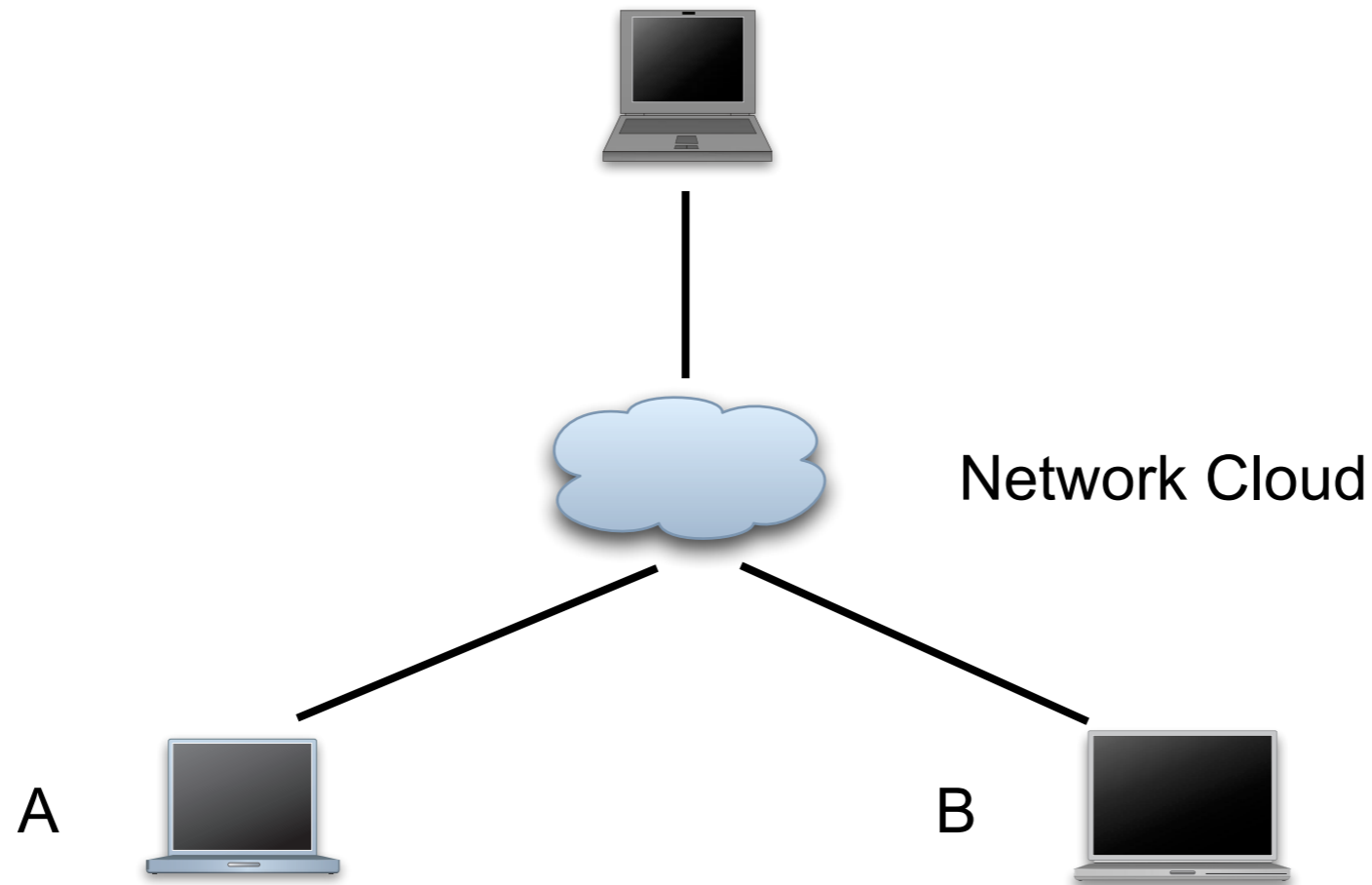
Each packet routed separately

Routing Issues

Overhead issues

Send Message To Machine A

This is just a sample message that one might send on a network to another machine.



Sending

This is just a sample message that one might send on a network to another machine.



A:1:This is just a samp

A:2:le message that one

A:3: might send on a ne

A:4:twork to another ma

A:5:chine.



Network Cloud

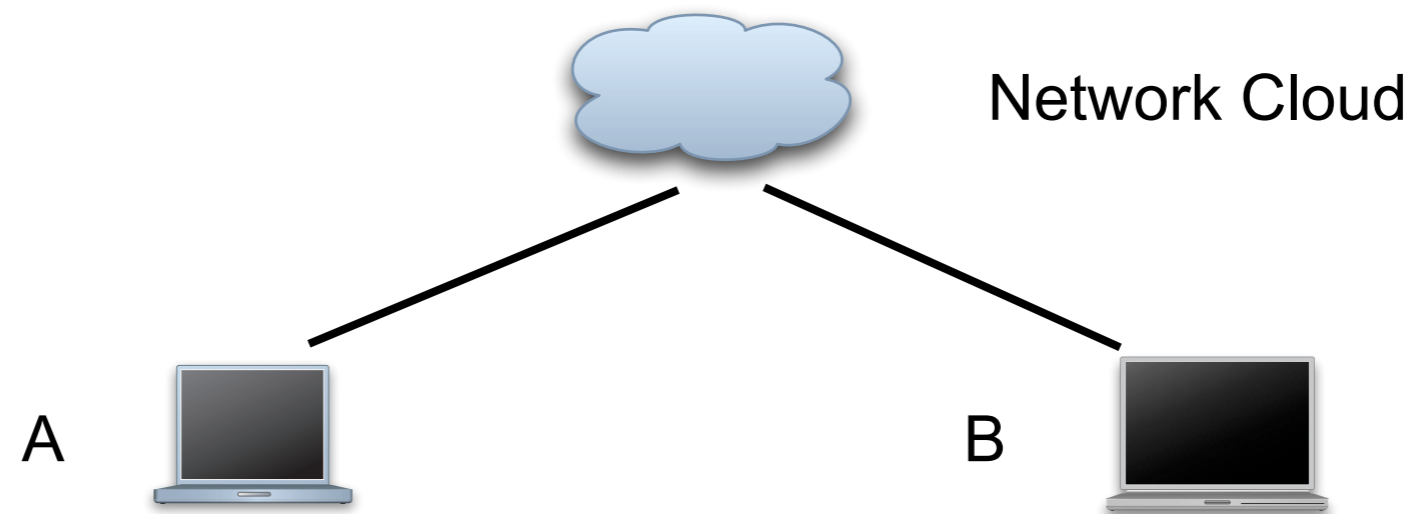
A



B



Receiving



A:2:le message that one

A:1:This is just a samp

A:5:chine.

A:4:twork to another ma

A:3: might send on a ne

This is just a sample message that one might send on a network to another machine.

Issues

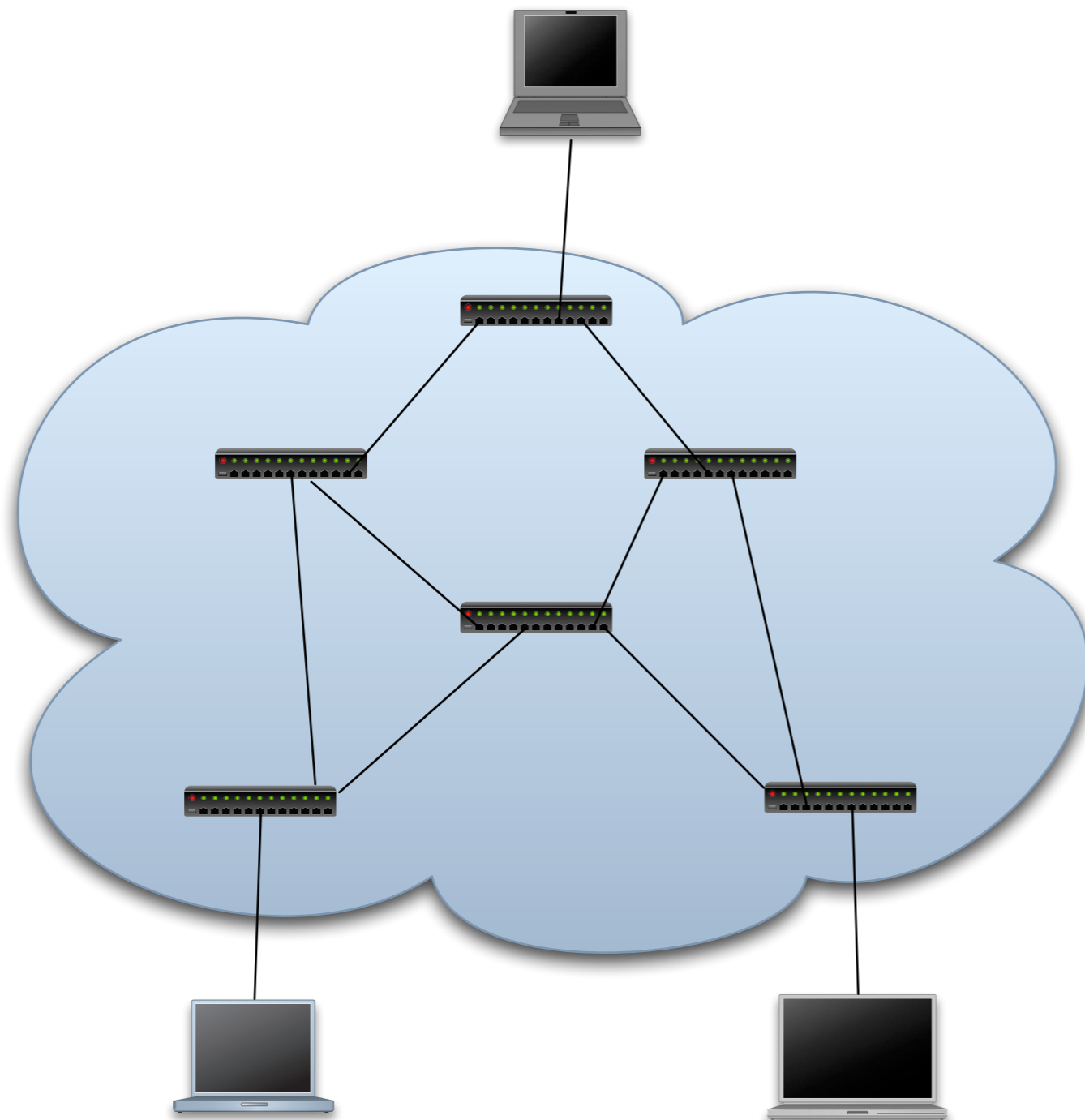
How does the message get to A

How does the message get to the correct program on A

How do packets get lost

How do packets get out of order

Routers



Bandwidth & Latency

Bandwidth

bit rate or throughput of a network connection

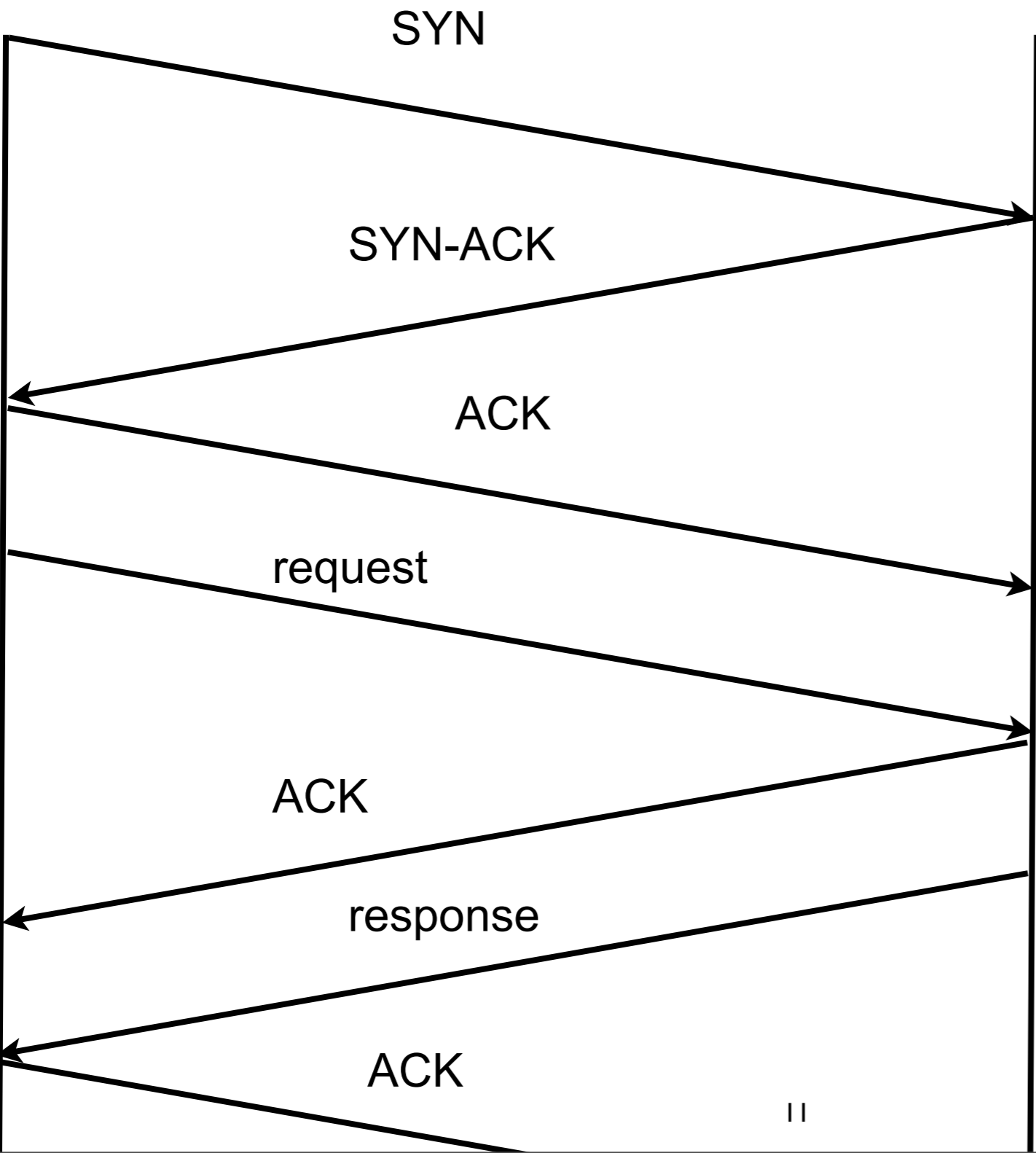
Latency

Time (delay) it takes a packet to reach its destination

Bandwidth, Latency & TCP

Client

Server

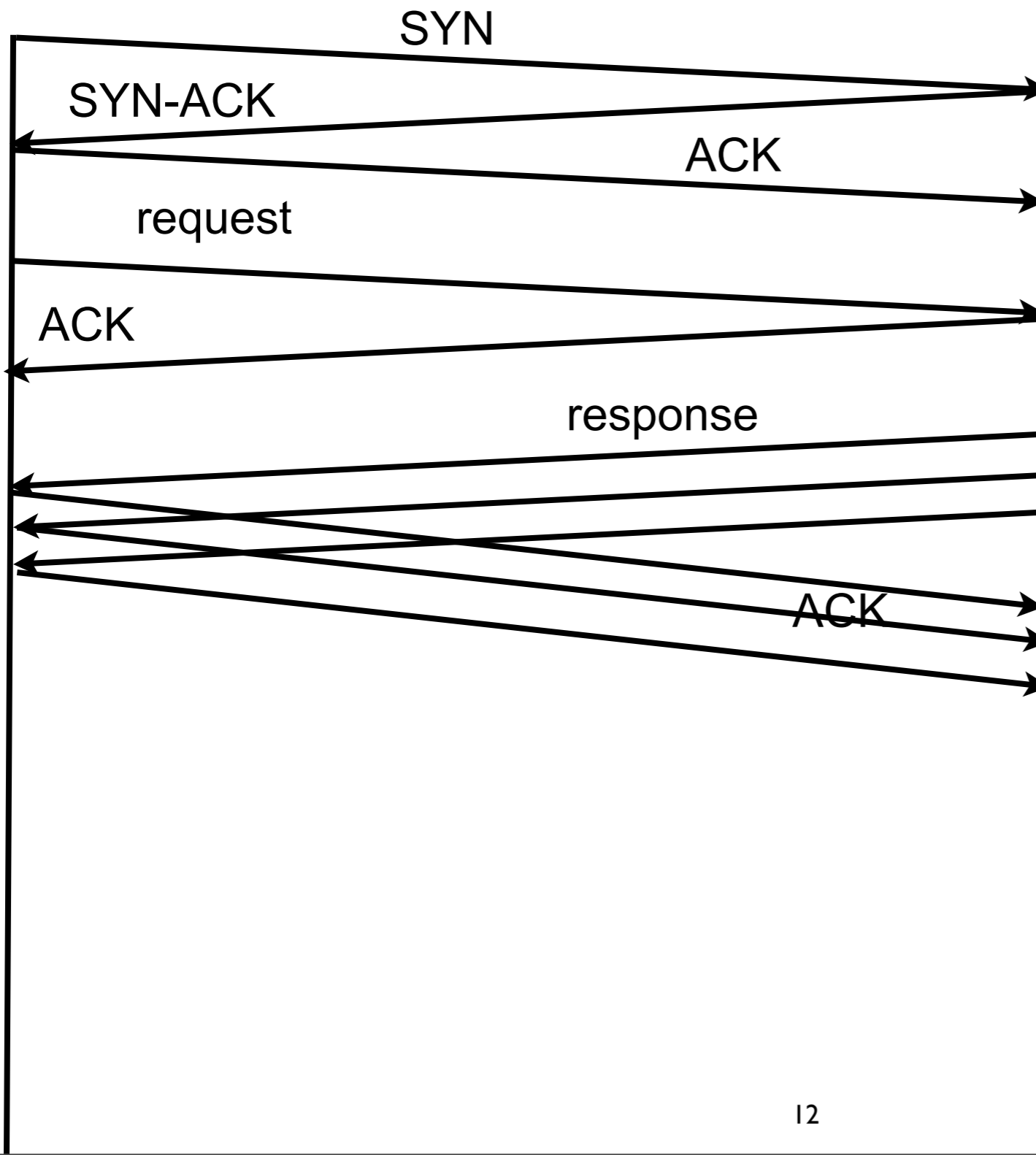


||

Bandwidth, Latency & TCP Window

Client

Server



TCP & Bandwidth

Start with small window

Each time send data increase size of window
Until detect packet loss

If have packet loss decrease size of window

How do packets get lost

Buffer overflow in routers

Wireless connections

How do packets get out of order

Different routes

Different wait times in router buffers

Some Useful Programs

netstat

Show status of network connections on machine

lsof

list open files (& pipes & sockets)

traceroute

Show the route to remote machine

tcpdump

Shows packets sent/received

netstat

Windows, Unix/Linux

Al pro 14->netstat

Active Internet connections

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp4	17680	0	10.0.1.192.60840	kusc-pc-stream2..irdmi	ESTABLISHED
tcp4	0	0	10.0.1.192.60627	208.43.202.32-st.http	ESTABLISHED
tcp4	0	0	10.0.1.192.60623	adsl-68-20-22-55.28205	ESTABLISHED
tcp4	0	0	localhost.26164	localhost.60431	ESTABLISHED
tcp4	0	0	localhost.60431	localhost.26164	ESTABLISHED
tcp4	0	0	10.0.1.192.afpovertcp	10.0.1.200.53611	ESTABLISHED
tcp4	0	0	localhost.26164	localhost.53896	ESTABLISHED
tcp4	0	0	localhost.53896	localhost.26164	ESTABLISHED
tcp4	0	0	localhost.26164	localhost.51153	ESTABLISHED
tcp4	0	0	localhost.51153	localhost.26164	ESTABLISHED
tcp4	0	0	localhost.26164	localhost.49164	ESTABLISHED
tcp4	0	0	localhost.49164	localhost.26164	ESTABLISHED
tcp4	37	0	10.0.1.192.49163	174.36.30.66-sta.https	CLOSE_WAIT
tcp4	37	0	10.0.1.192.49162	174.36.30.67-sta.https	CLOSE_WAIT
tcp4	0	0	10.0.1.192.60862	WAREHOUSE-THREE-.55510	TIME_WAIT
tcp4	0	0	10.0.1.192.60861	dhcp128036163075.55165	TIME_WAIT
etc					

netstat -s

```
Al pro 13->netstat -s
```

```
tcp:
```

```
1290678 packets sent
```

```
169 data packets (52330 bytes) retransmitted
```

```
686952 ack-only packets (8670 delayed)
```

```
1530460 packets received
```

```
332351 acks (for 114678930 bytes)
```

```
8686 duplicate acks
```

```
760 completely duplicate packets (611003 bytes)
```

```
27337 out-of-order packets (34890076 bytes)
```

```
11654 connection requests
```

```
104 connection accepts
```

```
48 bad connection attempts
```

```
8 listen queue overflows
```

```
etc.
```

Lsof

Unix/Linux

list open files

disk files, pipes and network sockets

Air 18->lsof -i

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
SystemUIS	194	whitney	9u	IPv4	0x3da8f40	0t0	UDP	*:*
adb	1141	whitney	5u	IPv4	0x9f28270	0t0	TCP	localhost:5037 (LISTEN)
Safari	3234	whitney	56u	IPv6	0x3dac19c	0t0	TCP	localhost:59088->localhost:59087 (TIME_WAIT)
Safari	3234	whitney	71u	IPv4	0x623366c	0t0	TCP	146.244.205.67:50097->a198-189-255-145.deploy.akamaitechnologies.com:http (CLOSE_WAIT)
Safari	3234	whitney	74u	IPv4	0x61fb270	0t0	TCP	146.244.205.67:50099->a198-189-255-145.deploy.akamaitechnologies.com:http (CLOSE_WAIT)

traceroute

tracpath - Linux

tracert - Windows

Al pro 15->traceroute www.sdsu.edu

traceroute to www.sdsu.edu (130.191.8.198), 64 hops max, 40 byte packets

```
1 10.0.1.1 (10.0.1.1) 0.679 ms 0.192 ms 0.174 ms
2 ip68-8-224-1.sd.sd.cox.net (68.8.224.1) 8.317 ms 6.879 ms 7.574 ms
3 fed1sysc01-gex0915.sd.sd.cox.net (68.6.10.106) 15.600 ms 8.736 ms 11.449 ms
4 fed1sysc10-get0005.sd.sd.cox.net (68.6.8.78) 10.456 ms 10.895 ms 8.740 ms
5 dt1xaggc01-get0701.sd.sd.cox.net (68.6.8.49) 12.298 ms 23.956 ms 7.625 ms
6 sdscCBSf01-fex0301.cox-sd.net (209.242.135.150) 7.831 ms 8.904 ms 8.057 ms
7 sdg-agg3.cenic.net (198.17.46.176) 12.655 ms 230.550 ms 264.670 ms
8 dc-sdg-agg1--sdg-agg3-ge.cenic.net (137.164.46.16) 15.624 ms 14.830 ms 14.923 ms
9 dc-sd-csu-egm--sdg-dc1.cenic.net (137.164.41.138) 13.097 ms 13.603 ms 15.632 ms
10 * * *
11 * * *
```

Internet Protocol (IP or TCP/IP)

Application Layer

DHCP, DNS, FTP, HTTP, SSH, Telnet, (more)

Transport Layer

TCP, UDP, (more)

Internet Layer

IPv4, IPv6, (more)

Link Layer

Ethernet, DSL, ISDN, FDDI, (more)

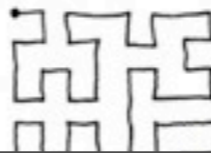
MAP OF THE INTERNET

THE IPv4 SPACE, 2006



THIS CHART SHOWS THE IP ADDRESS SPACE ON A PLANE USING A FRACTAL MAPPING WHICH PRESERVES GROUPING-- ANY CONSECUTIVE STRING OF IPs WILL TRANSLATE TO A SINGLE COMPACT, CONTIGUOUS REGION ON THE MAP. EACH OF THE 256 NUMBERED BLOCKS REPRESENTS ONE /8 SUBNET (CONTAINING ALL IPs THAT START WITH THAT NUMBER). THE UPPER LEFT SECTION SHOWS THE BLOCKS SOLD DIRECTLY TO CORPORATIONS AND GOVERNMENTS IN THE 1990's BEFORE THE RIRs TOOK OVER ALLOCATION.

0	1	14	15	16	19	→
3	2	13	12	17	18	
4	7	8	11			



 = UNALLOCATED BLOCK

TCP

Handles lost packets

Handles packet order

TCP has delays

- Starting of connection

- Closing of connection

- Resending packets

Client & Sever don't have to deal with

- Packet order

- Packet loss

TCP Header

Bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Source port																Destination port															
32	Sequence number																															
64	Acknowledgment number																															
96	Data offset	Reserved						C W R	E C E	U R G	A C K	P S H	R S T	S S Y N	F I N	Window Size																
128	Checksum																Urgent pointer															
160	Options (if Data Offset > 5)																															
...	...																															

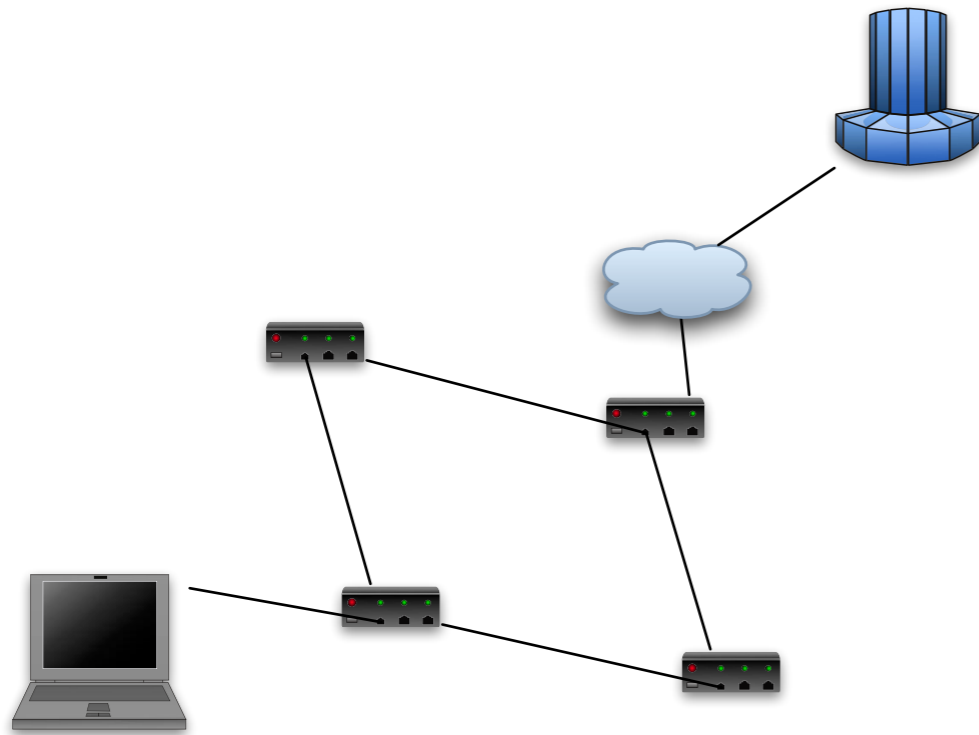
IP Header

bit offset	0–3	4–7	8–15	16–18	19–31
0	Version	Header length	Differentiated Services	Total Length	
32	Identification			Flags	Fragment Offset
64	Time to Live	Protocol		Header Checksum	
96	Source Address				
128	Destination Address				
160	Options (if Header Length > 5)				
160 or 192+	Data				

Ethernet Frame

Preamble	Start-of-Frame-Delimiter	MAC destination	MAC source	802.1Q header (optional)	Ethertype/Length	Payload (Data and padding)	CRC32	Interframe gap
7 octets of 10101010	1 octet of 10101011	6 octets	6 octets	(4 octets)	2 octets	46–1500 octets	4 octets	12 octets
		64–1522 octets						
72–1530 octets								
84–1542 octets								

UDP



Fast

Packets are treated individually
Packets may arrive out of order
Packets may be lost

Client & Server must handle resulting problems

Used by:

Games

NFS

UDP Header

bits	0 - 15	16 - 31
0	Source Port	Destination Port
32	Length	Checksum
64	Data	

IP Addresses

IP address is currently a 32-bit number

130.191.3.100 (Four 8 bit numbers)

IPv6 uses 128 bit numbers for addresses

105.220.136.100.0.0.0.0.0.0.18.128.140.10.255.255

69DC:8864:0:0:0:1280:8C0A:FFFF

69DC:8864::1280:8C0A:FFFF

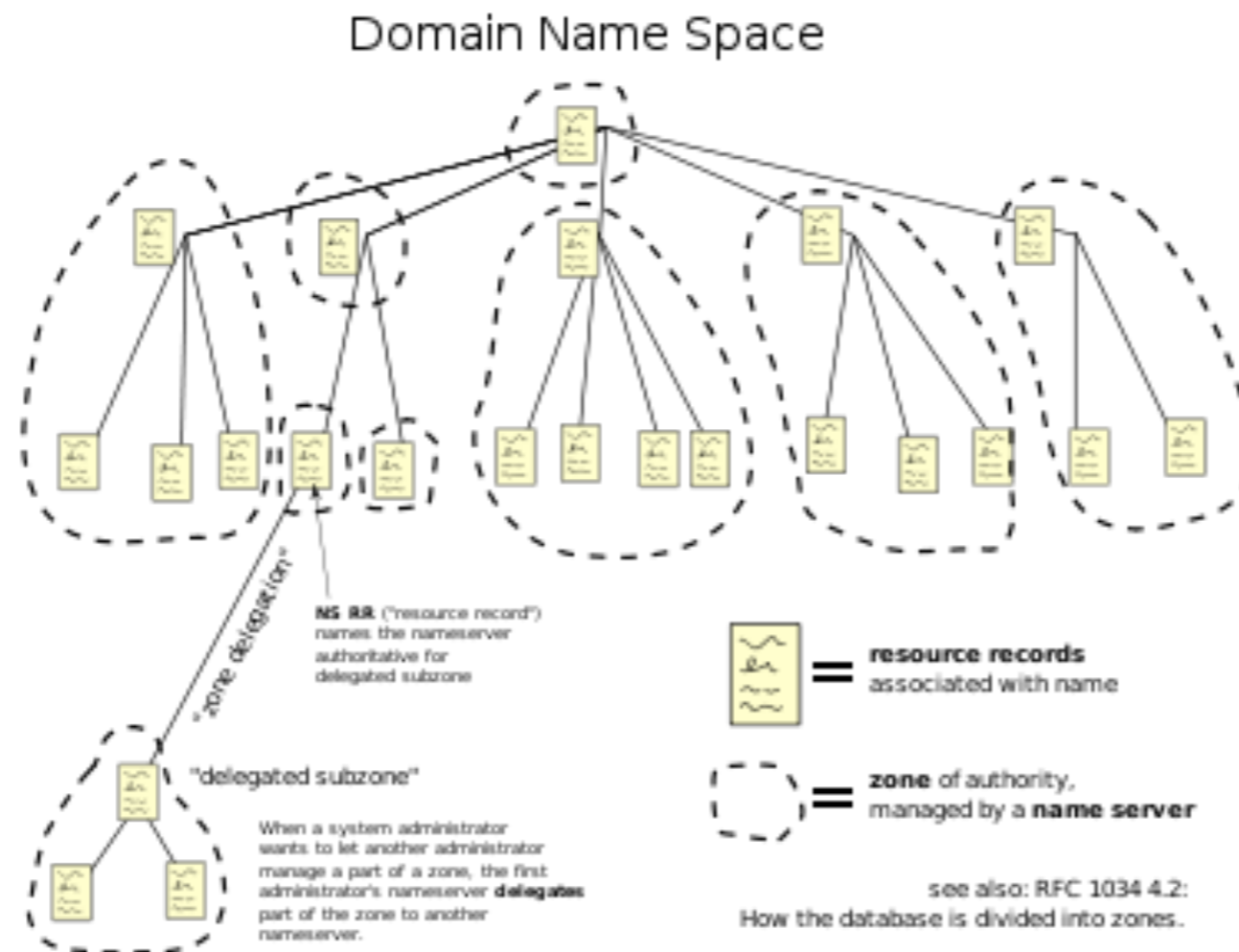
Machines on a network need a unique IP address

What is the difference between
MAC address
IP address

Domain Name System (DNS)

Maps machine names to IP addresses

Internet Corporation for Assigned Names and Numbers (ICANN <http://www.icann.org/>) oversees assigning TLDs



Unix "host" command

Shows mapping between machine names and IP address

```
->host rohan.sdsu.edu
```

```
rohan.sdsu.edu has address 130.191.3.100
```

```
->host 130.191.3.100
```

```
100.3.191.130.IN-ADDR.ARPA domain name pointer rohan.sdsu.edu
```


Ports

TCP/IP supports multiple logical communication channels called ports

Ports are numbered from 0 - 65535

A connection between two machines is uniquely defined by:

- Protocol (TCP or UDP)

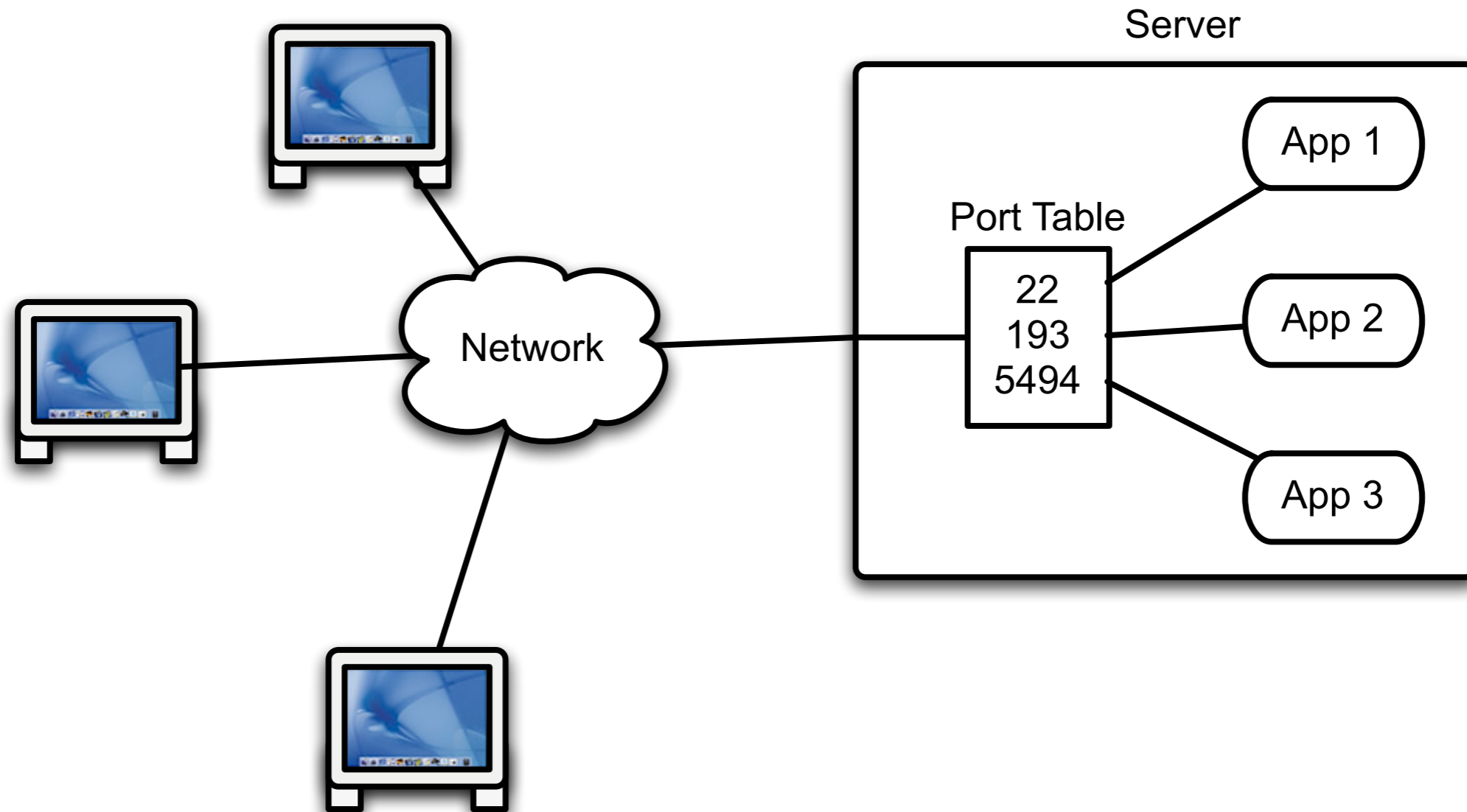
- IP address of local machine

- Port number used on the local machine

- IP address of remote machine

- Port number used on the remote machine

How Ports Work



Some Port Numbers

Well known Ports	1-1023
Registered Ports	1024-49151
Dynamic/Private Ports	49152-65535

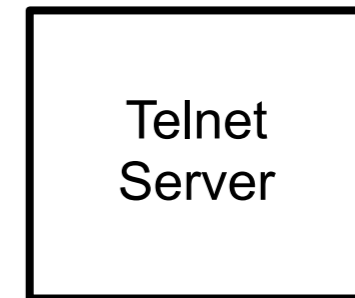
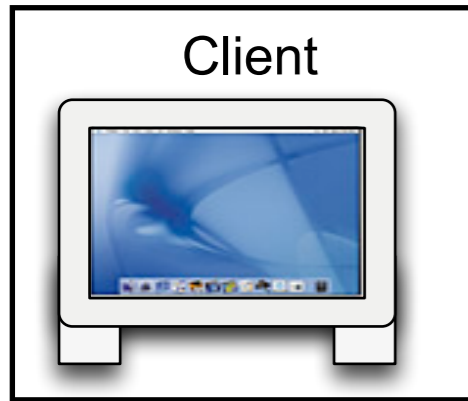
For a local list of services
`file://rohan.sdsu.edu/etc/services`

For a complete list see:
<http://www.iana.org/assignments/port-numbers>

See IANA numbers page <http://www.iana.org/numbers.html> for more information about protocol numbers and assignment of services

Service	Port
echo	7
discard	9
ftp	21
ssh	22
telnet	23
smtp	25
time	37
http	80
pop	110
https	443
doom	666
mysql	3306
postgresql	5432
gnutella	6346 6347

What is Telnet?



Protocol

Send text between client & server

Server

Requests login

Sends text to shell to be executed

Returns result of commands

Client

Transfers text between user and server

Telnet & Other Text-based Protocols

```
rohan 37 -> telnet www.eli.sdsu.edu 80
GET /courses/spring06/cs580/index.html HTTP/1.0 <CR>
<CR>
```

Note <CR> indicates where you need to hit return

```
rohan 38->telnet cs.sdsu.edu 110
Trying 130.191.226.116...
Connected to cs.sdsu.edu.
Escape character is '^]'.
+OK QPOP (version 3.1.2) at sciences.sdsu.edu starting.
USER whitney
+OK Password required for whitney.
PASS typeYourPasswordHere
+OK whitney has 116 visible messages (0 hidden) in 640516 octets.
```

Simple Date Example - Protocol

Client Commands	Server Response
"date" ended by line feed "date\n"	current date ended by line feed "January 30, 2007\n"
"time" ended by line feed "time\n"	Current time ended by line feed "6:58 pm\n"

Server listens for an incoming request

On request

- reads command
- returns response
- closes connection

On client errors - action not specified

Beware

Can only send bytes across network

Client & server maybe different hardware platforms

What is a newline?

End-of-file indicates connection is closed

Sample Java Client

```
import java.io.*;
import java.net.Socket;

class DateClient {
    String server;
    int port;

    public DateClient(String serverAddress, int port) {
        server = serverAddress;
        this.port = port;
    }

    public String date() {
        return send("date\n");
    }

    public String time() {
        return send("time\n");
    }
}
```


Java Client Continued

```
private String send(String text) {  
    try {  
        Socket connection = new Socket(server, port);  
        OutputStream rawOut = connection.getOutputStream();  
        PrintStream out = new PrintStream(new BufferedOutputStream(rawOut));  
        InputStream rawIn = connection.getInputStream();  
        BufferedReader in = new BufferedReader(new  
InputStreamReader(rawIn));  
  
        out.print(text);  
        out.flush();  
        String answer = in.readLine();  
        out.close();  
        in.close();  
        return answer;  
    }  
    catch (IOException e) {  
        return "Error in connecting to server";  
    }  
}  
}
```

Running the Client

```
System.out.println("hi");  
DateClient client = new DateClient("127.0.0.1", 4444);  
System.out.println( client.date());  
System.out.println( client.time());
```

Issue - Avoid Small Packets

```
OutputStream rawOut = connection.getOutputStream();  
PrintStream out = new PrintStream(new BufferedOutputStream(rawOut));
```

Issue - Actually Send the request

```
out.flush();
```

Issue - Client will not work on all platforms

```
String answer = in.readLine();
```

Don't Do this

```
String answer = in.readLine();
```

Issue - Close the connection when done

```
out.close();  
in.close();
```

Issue - Testing

How does one test the client?

Issue - Background material

Java

Streams

Read Chapter 4

Sockets

Read Chapter 10

Java Network Programming, Harold 3rd Ed

Server

Basic Algorithm

```
while (true) {  
    Wait for an incoming request;  
    Perform whatever actions are requested;  
}
```

Basic Server Issues

How to wait for an incoming request?

How to know when there is a request?

What happens when there are multiple requests?

How do clients know how to contact server?

How to parse client request?

How do we know when the server has the entire request?

Java Date Server

```
public class DateServer {
    private static Logger log = Logger.getLogger("dateLogger");

    public static void main (String args[]) throws IOException {
        ProgramProperties flags = new ProgramProperties( args);
        int port = flags.getInt( "port" , 8765);
        new DateServer().run(port);
    }

    public void run(int port) throws IOException {
        ServerSocket input = new ServerSocket( port );
        log.info("Server running on port " + input.getLocalPort());

        while (true) {
            Socket client = input.accept();
            log.info("Request from " + client.getInetAddress());
            processRequest(
                client.getInputStream(),
                client.getOutputStream());
            client.close();
        }
    }
}
```

Java Date Server Continued

```
void processRequest(InputStream in,OutputStream out)
    throws IOException {

    BufferedReader parsedInput =
        new BufferedReader(new InputStreamReader(in));

    boolean autoflushOn = true;
    PrintWriter parsedOutput = new PrintWriter(out,autoflushOn);

    String inputLine = parsedInput.readLine();

    if (inputLine.startsWith("date")) {
        Date now = new Date();
        parsedOutput.println(now.toString());
    }
}
}
```

This server needs work

Starting the Server

```
rohan 16-> java -jar DateServer.jar  
Feb 19, 2004 10:56:59 AM DateServer run  
INFO: Server running on port 8765
```

Issue - Date Format

What format does the server use for time and date?

Clients need to know so can parse them