As for Bell's first telephone lines, they were as simple as clothes-lines. Each short little wire stood by itself, with one instrument at each end. ... There were no switchboards of any account, no cables of any value, no wires that were in any sense adequate, no theory of tests or signals, no exchanges, no telephone system of any sort whatever.

But there had now come a time when more than two persons wanted to be in the same conversational group. ... Here was the new problem, and a most stupendous one -- how to link together three telephones, or three hundred, or three thousand, or three million, so that any two of them could be joined at a moment's notice.

From: The History of the Telephone
Herbert N. Casson
1910
Homework 4 due today (Hw 5 available soon)

Lab 5 -

“Working with Multiple Classes”

Get Lab 4 finished first (except &’s)

Midterm

Evening of March 17th
At 6 p.m. or 7:30 p.m. in TPL 203
75 minute closed book exam
Review session 3/16; Time/room:TBD
Sample midterm will be available online
Today’s Plan

- Java Constructor Parameters & Instance Variables.
- Broadcast Networks -- Ethernet (and WiFi)
Ray Tomlinson, credited with founding email, passes away at 74.

Ray Tomlinson, the man credited with founding email, has passed away at the age of 74, according to a report from the *Sydney Morning Herald*. According to the report, Tomlinson died of a heart attack.

Tomlinson was a part of a team of computer programmers at research and design company Bolt Beranek and Newman (now BBN Technologies) in Cambridge, Mass., and had begun experimenting with internal messaging in 1971. Tomlinson had an idea to send a text message between computers using a new network (and a predecessor to the Internet) called Arpanet, routing it using an “@” symbol.

He recalled the first email sent in an interview with NPR from 2009: “The keyboards were about 10 feet apart...I could wheel my chair from one to the other and type a message on one, and then go to the other, and then see what I had tried to send.”
public class PopConnectionClient extends GUIManager {
    ...
    // Used to enter the name of the SMTP server to user for replies
    private JTextField smtpServer =
        new JTextField( "cs.williams.edu", 20 );
    ...
    // The connection to the server
    private POPConnection toServer;
    // The most recently accessed message
    private MailMessage cureMessage;
    // Create the GUI components and display the login components
    public PopConnectionClient() {
        ...
    }
    // When the button is clicked, interact with the POP server
    public void buttonClicked( JButton which ) {
        if ( which == login ) {
            ...
            String userID = ...;
            String mailServer = ...;
            toServer = new POPConnection( mailServer, userID,
                pass.getText() );
            if ( toServer.isConnected() ) {
                messageCount.setText("of " +
                    toServer.messagesAvailable() );
            }
        } else if ( which == get ) {
            // Retrieve and display the specified email message
            cureMessage =
                toServer.getMessage( messageNum.getText() );
            message.setText( cureMessage.getText() );
        } else if ( which == reply ) {
            // Create a new reply window
            new ReplyWindowVariableServer( smtpServer.getText(),
                cureMessage.getFrom(),
                user.getText(),
                cureMessage.getText() );
        }
    }
}

public class ReplyWindowVariableServer extends GUIManager {
    // The size of the program's window
    private final int WINDOW_WIDTH = 460, WINDOW_HEIGHT = 520;
    ...
    // address of SMTP server to use
    private String smtpServer;
    ...
    // Place fields and text areas on screen to enable user to
    // enter mail together with a send button.
    public ReplyWindowVariableServer( String serverName,
        String toAddr, String fromAddr,
        String original ) {
        // Create window to hold all the components
        this.createWindow( WINDOW_WIDTH, WINDOW_HEIGHT );
        smtpServer = serverName;
        ...
        // Send a message when the button is clicked
        public void buttonClicked() {
            NetConnection toServer =
                new NetConnection( smtpServer, SMTP_PORT );
            log.append( toServer.in.nextLine() + "\n" );
        }
    }
}
As for Bell's first telephone lines, they were as simple as clothes-lines. Each short little wire stood by itself, with one instrument at each end. ... There were no switchboards of any account, no cables of any value, no wires that were in any sense adequate, no theory of tests or signals, no exchanges, no telephone system of any sort whatever.

But there had now come a time when more than two persons wanted to be in the same conversational group. ... Here was the new problem, and a most stupendous one -- how to link together three telephones, or three hundred, or three thousand, or three million, so that any two of them could be joined at a moment's notice.

From: The History of the Telephone
Herbert N. Casson
1910
1. Background

One can characterize distributed computing as a spectrum of activities varying in their degree of decentralization, with one extreme being remote computer networking and the other extreme being multiprocessing. Remote computer networking is the loose interconnection of previously isolated, widely separated, and rather large computing systems. Multiprocessing is the construction of previously monolithic and serial computing systems from increasingly numerous and smaller pieces computing in parallel. Near the middle of this spectrum is local networking, the interconnection of computers to gain the resource sharing of computer networking and the parallelism of multiprocessing.

The separation between computers and the associated bit rate of their communication can be used to divide the distributed computing spectrum into broad activities. The product of separation and bit rate, now about 1 gigabit-meter per second (1 Gbmps), is an indication of the limit of current communication technology and can be expected to increase with time:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Separation</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote networks</td>
<td>&gt; 10 km</td>
<td>&lt; .1 Mbps</td>
</tr>
<tr>
<td>Local networks</td>
<td>10–1 km</td>
<td>.1–10 Mbps</td>
</tr>
<tr>
<td>Multiprocessors</td>
<td>&lt; .1 km</td>
<td>&gt; 10 Mbps</td>
</tr>
</tbody>
</table>

1.1 Remote Computer Networking
Prof. Steve Lewis ’60
Economics

Larry Wright ’58
Computer Center Director

Shane Riorden
College Business Manager

Xerox 530
Computer with 64KB of memory

Bronfman 119 ~ 1974
The Alto Computer, 1973

In 1972, Xerox decided to produce a personal computer for research purposes. This became the Alto computer, the result of a joint effort by Ed McCreight, Chuck Thacker, Butler Lampson, Bob Sproull and Dave Boggs, who were attempting to make a device that was small enough to fit in an office comfortably yet powerful enough to support a reliable, high-quality operating system and graphics display. Its GUI featured windows and icons. A few years later, Steve Jobs and Steve Wozniak borrowed some of these ideas and started Apple Computer.
Broadcasting on a Cable
### Ethernet Frame Format

<table>
<thead>
<tr>
<th>PREAMBLE</th>
<th>TO</th>
<th>FROM</th>
<th>LENGTH/TYPE</th>
<th>DATA</th>
<th>ERROR CHECK</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>48</td>
<td>48</td>
<td>16</td>
<td>between 46 and 1500 bytes</td>
<td>32</td>
</tr>
</tbody>
</table>
Carrier Sense Multiple Access with Collision Detection

a.k.a.

SHARING NICELY
Ethernet-etiquette

- Don’t interrupt
- Be persistent
- Know when to back off
- Know how to back off
Ethernet-etiquette

- Don’t interrupt (Carrier Sense)
- Be persistent
- Know when to back off
- Know how to back off
Ethernet-etiquette

- Don’t interrupt (Carrier Sense)
- Be persistent (1 Persistence)
- Know when to back off
- Know how to back off
Ethernet-etiquette

- Don’t interrupt (Carrier Sense)
- Be persistent (1 Persistence)
- Know when to back off (Collision Detection)
- Know how to back off
Ethernet-etiquette

- Don’t interrupt (Carrier Sense)
- Be persistent (1 Persistence)
- Know when to back off (Collision Detection)
- Know how to back off (Exponential Backoff)
Binary Exponential Backoff

After detecting a collision:

- Pick a random number between 0 and $2^{\text{attempts}} - 1$ slots
- Wait that many “time units” and then try again.
ETHERNET TRANSMISSION ALGORITHM

1. Eager to send a packet
   - Receiver becomes idle
   - Receiver becomes busy
   - Receiver not idle

2. Transmitting packet preamble
   - Preamble complete
   - Receiver not idle
   - Receiver becomes busy

3. Transmitting packet contents
   - Receiver not idle
   - Receiver becomes busy
   - Receiver becomes ready

4. Waiting for backoff slots
   - Backoff complete
   - Receiver not idle

5. Jamming complete

DONE

CARRIER SENSE MULTIPLE ACCESS WITH COLLISION DETECTION
AMAZING ANIMATED DEMONSTRATION!!!
Summary of Ethernet

- Carrier Sense = Wait if network idle
- 1-persistence = If waiting, start when idle
- Collision Detection = Stop and Backoff
- Minimum packet transmission time = $2 \times \text{max propagation time} = \text{waiting slot time}$
- Backoff = Delay random # between 0 and $2^{\text{failures}} - 1$ slots after collision
ETHERNET TRANSMISSION ALGORITHM

Packet becomes ready

Eager to send a packet

Transmitting packet preamble

Transmitting packet contents

Receiver becomes idle

Waiting for backoff slots

Jamming signal

Receiver becomes busy

Receiver becomes idle

Preamble complete

Receiver not idle

Receiver idle

Reset range of delay slots to 1

DONE

Double delay slot range = $2^0$ failures