Objectives

- Describe common Transport layer protocols
- Explain the power of port numbers
- Define common TCP/IP applications such as HTTP, HTTPS, Telnet, e-mail (SMTP, POP3, and IMAP4), and FTP
Overview
Three Parts to this Area

- Transport Layer Protocols
- The Power of Port Numbers
- Common TCP/IP Applications
Transport Layer Protocols
How People Communicate

- Connection-oriented
  - Acknowledgement between two people beginning conversation
  - The conversation
  - Close of conversation
Figure 9.1  A connection-oriented session starting
• Connectionless
  – No opening acknowledge
  – Short message shouted across a room
  – No closing

• Session
  – Any single communication between computers
  – All session must begin and eventually end
Figure 9.2  A connection-oriented session ending
• Transmission Control Protocol (TCP)
  – In charge of connection-oriented communication
  – Most common type of TCP/IP session
Connection-oriented session

- Browser sends an ACK packet
- Server responds with a SYN, ACK packet
- Client sends an ACK, requests Web page
- Server sends Web page and a FIN packet
- Client responds with RST, ACK
• User Datagram Protocol (UDP)
  – Used by very few applications
  – Requires much less overhead than TCP
  – No start, no acknowledgement, no end
DHCP uses UDP

- Client broadcasts discovery packet
- Server responds with DHCP offer (sent directly to MAC address)
- Client sends DHCP request directly to server MAC address
- Server sends DHCP acknowledgement with IP configuration
- Client responds with DHCP lease
Figure 9.3  DHCP steps
• **Trivial File Transfer Protocol (TFTP)**
  - Uses UDP
  - Transfers files between computers
  - Does not have any data protection
  - Never use it over the Internet
Internet Control Message Protocol (ICMP)

- For connectionless communications that never need more than a single packet
- Handles maintenance issues like disconnect (host unreachable)
- Applications use ICMP to send status information to the other end of a session
• **PING**

  - Sends a single ICMP packet
    - Echo request
    - To an IP address
  - All computers (unless blocked by a firewall) respond with echo reply
C:\>ping www.google.com

PINGING www.google.com [74.125.95.147] WITH 32 BYTES OF DATA:
Reply from 74.125.95.147: bytes=32 time=70ms TTL=242
Reply from 74.125.95.147: bytes=32 time=70ms TTL=242
Reply from 74.125.95.147: bytes=32 time=70ms TTL=242
Reply from 74.125.95.147: bytes=32 time=70ms TTL=242

PING STATISTICS FOR 74.125.95.147:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 70ms, Maximum = 70ms, Average = 70ms

C:\>
- Internet Group Management Protocol (IGMP)
  - Used for multicasts
  - Routers use to determine a "group" membership
  - Class D IP addresses with network ID 224/8
More about multicast

- Does not assign IP addresses to hosts
- A multicast is assigned a certain 224/8 address
- Those who wish to receive this multicast must join the IGMP group
- Upstream router will send multicasts
Figure 9.5  IGMP in action
More about multicast

- Does not assign IP addresses to hosts
- A multicast is assigned a certain 224/8 address
- Those who wish to receive this multicast must join the IGMP group
- Upstream router will send multicasts
Transport Layer Protocols
• Port numbers
  – Memorize common port numbers
  – Every TCP/IP app requires a server and a client
  – Defined port number for popular (well-known) TCP/IP applications
• By the (port) numbers
  - 16-bit values (0 to 65,535)
  - Well-known port numbers (0-1023) for specific TCP/IP applications
  - Web servers use port number 80
  - Web client sends HTTP ACT to server (port 80)
  - Server replies using ephemeral port
Figure 9.6 HTTP ACK packet

Destination info: 147.58.201.183
Port: 80
Source info: 4.8.15.16
Figure 9.7  Dealing with the incoming packet
Figure 9.8  A more complete IP packet

Destination info:
147.58.201.183
Port: 80
Source info:
4.8.15.16
Ephemeral Port: 52142
• Ephemeral ports
  – Pseudo-randomly generated by Web client
  – Ephemeral port numbers 1024-5000
  – Dynamic or private port numbers 49152-65535
  – IANA recommends using only 49152-65535
Figure 9.9  Returning the packet
• Registered ports

  - 1024-49151
  - Less-common TCP/IP applications register their ports with IANA
  - Most operating systems avoid registered port numbers and use dynamic/private ports
Summary of port numbers

- 0-1023  well-known ports
- 1024-49151  registered ports
- 49152-65535  dynamic or private ports
• Using ports in a session

– Both computers keep track of status

• Status info held in RAM
• Socket or endpoint is one side’s session information
• Socket pairs or endpoints refer to data each computer stores about the connection
• Session or connection refers to a connection in general
• Endpoint information
  – Source and destination for one session
  – Many simultaneous sessions
  – Use `netstat -n` to see sessions
    • Usually shows many connections
    • TCPView for Windows: dynamic
    • Net Activity Viewer for Linux
Figure 9.10  Two open windows
Figure 9.11  TCPView in action
### Figure 9.12  Net Activity Viewer

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Local Port</th>
<th>State</th>
<th>Remote Address</th>
<th>Remote Port</th>
<th>Remote Host</th>
<th>Pid</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp</td>
<td>901 swat</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>27015</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>3306 mysql</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>139 netbios-ssn</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>10000 webmin</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>80 www</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>22 ssh</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>631 ip</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>445 microsoft-ds</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>58694</td>
<td>ESTABLISHED</td>
<td>91.189.94.9</td>
<td>80 www</td>
<td>avocado.canonical.com</td>
<td>8485</td>
<td>firefox</td>
</tr>
<tr>
<td>tcp</td>
<td>42787</td>
<td>CLOSED</td>
<td>91.189.90.19</td>
<td>80 www</td>
<td>yangmei.canonical.com</td>
<td>8485</td>
<td>firefox</td>
</tr>
<tr>
<td>tcp</td>
<td>38186</td>
<td>CLOSED</td>
<td>91.189.90.19</td>
<td>443 https</td>
<td>yangmei.canonical.com</td>
<td>8485</td>
<td>firefox</td>
</tr>
<tr>
<td>tcp</td>
<td>38191</td>
<td>CLOSED</td>
<td>91.189.90.19</td>
<td>443 https</td>
<td>yangmei.canonical.com</td>
<td>8485</td>
<td>firefox</td>
</tr>
<tr>
<td>tcp</td>
<td>38189</td>
<td>CLOSED</td>
<td>91.189.90.19</td>
<td>443 https</td>
<td>yangmei.canonical.com</td>
<td>8485</td>
<td>firefox</td>
</tr>
<tr>
<td>tcp</td>
<td>38192</td>
<td>CLOSED</td>
<td>91.189.90.19</td>
<td>443 https</td>
<td>yangmei.canonical.com</td>
<td>8485</td>
<td>firefox</td>
</tr>
<tr>
<td>tcp</td>
<td>38188</td>
<td>CLOSED</td>
<td>91.189.90.19</td>
<td>443 https</td>
<td>yangmei.canonical.com</td>
<td>8485</td>
<td>firefox</td>
</tr>
<tr>
<td>tcp</td>
<td>43699</td>
<td>CLOSED</td>
<td>209.85.225.97</td>
<td>443 https</td>
<td>yy-in-f97.google.com</td>
<td>8485</td>
<td>firefox</td>
</tr>
<tr>
<td>tcp</td>
<td>38190</td>
<td>CLOSED</td>
<td>91.189.90.19</td>
<td>443 https</td>
<td>yangmei.canonical.com</td>
<td>8485</td>
<td>firefox</td>
</tr>
<tr>
<td>tcp6</td>
<td>5900</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td>7066</td>
<td>vino-server</td>
</tr>
<tr>
<td>tcp6</td>
<td>22 ssh</td>
<td>LISTEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>udp</td>
<td>27015</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6133</td>
<td>srcds_i486</td>
</tr>
<tr>
<td>udp</td>
<td>137 netbios-ns</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>udp</td>
<td>137 netbios-ns</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>udp</td>
<td>138 netbios-dgm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>udo</td>
<td>138 netbios-dgm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Established: 1/23  Sent: 23 KB +645 B/s  Received: 91 KB +315 B/s
Connection Status

- State changes continually
- Listening port or open port
- ESTABLISHED ports are active, working endpoint pairs
- CLOSE_WAIT indicates that a client is making a graceful closure
- TIME_WAIT indicates a lost connection
• Detecting local program in a connection
  - `Netstat -ano` will show local process ID (PID) for each connection
Figure 9.13 Process Explorer
• Determining Good vs. Bad
  – Memorize a bunch of common ports
  – Learn how to use NETSTAT
  – Learn the ports that normally run on your operating system
  – Research processes you don’t recognize
  – Get rid of bad processes
Common TCP/IP Applications
• Web Servers
  – Store HTML documents
  – XHTML is an updated HTML with XML syntax
### Figure 9.14  My router’s Web page

<table>
<thead>
<tr>
<th>Router Information</th>
<th>Services</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Router Name</strong></td>
<td>DHCP Server</td>
<td>Total Available</td>
</tr>
<tr>
<td><strong>Router Model</strong></td>
<td>WRT-radiauth</td>
<td>13.8 MB / 16.0 MB</td>
</tr>
<tr>
<td><strong>LAN MAC</strong></td>
<td>WRT-rflow</td>
<td>Free</td>
</tr>
<tr>
<td><strong>WAN MAC</strong></td>
<td>MAC-upd</td>
<td>Used</td>
</tr>
<tr>
<td><strong>Wireless MAC</strong></td>
<td>Samba Mount</td>
<td>Buffers</td>
</tr>
<tr>
<td><strong>WAN IP</strong></td>
<td></td>
<td>Cached</td>
</tr>
<tr>
<td><strong>LAN IP</strong></td>
<td></td>
<td>Active</td>
</tr>
<tr>
<td><strong>WAN IP</strong></td>
<td></td>
<td>Inactive</td>
</tr>
</tbody>
</table>

### Wireless Statistics

<table>
<thead>
<tr>
<th>Radio</th>
<th>Mode</th>
<th>Network</th>
<th>SSID</th>
<th>Channel</th>
<th>Xmit</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio is Off</td>
<td>AP</td>
<td>Disabled</td>
<td>TotalTemp</td>
<td>1</td>
<td>28 miW</td>
<td>54 Mbps</td>
</tr>
</tbody>
</table>
Figure 9.15  XHTML source code
Web browsers (client side)
- Request HTML pages from Web servers
- Enter address into browser
- All browsers have a default Web page
- Web sites use text addresses using DNS
• **HTTP**
  - Stands for **Hypertext Transport Protocol**
  - Underlying protocol of the Web
  - Uses port 80 to transmit Web page data
  - `http://` at beginning of Web server address
• **HTTP weakness**
  – Relays commands without reference to any commands the user previously executed
  – Difficult to design complex and interactive Web pages
  – Other technologies enhance HTTP
    • JavaScript/AJAX
    • Server-side scripting
    • Adobe Flash
    • Cookies
• Publishing Web pages
  – Web server will “host” a HTML document
  – You can self-host
    • Install Web server software
    • Acquire a public IP address
    • Time-consuming and challenging
  – Host through your ISP
  – Use a Web hosting service company
  – Free Web hosting (nothing is free)
Web Servers and Web Clients

- Web server serves up Web pages
- Listens on port 80
- Fetches and sends requested HTML pages
- To create a Web server
  - Install Web server software
  - Connect computer to the Web
• Web Server Software
  – **Microsoft Internet Information Services (IIS)**
    • 20-connection limit on non-server versions of Windows
    • Only run IIS on Server versions of Windows
  – **Apache Server** runs on UNIX/Linux/Windows
    • On over 50% of Internet Web servers
    • Free
    • Non-GUI
    • Web administrators use an add-on GUI (Webmin)
Figure 9.16  IIS in action
Figure 9.17  Webmin Apache module
• Web Client Software (browsers)
  – Request and display Web pages
  – Many have multiple functions
  – Most popular
    • MS Internet Explorer (IE)
    • Mozilla Firefox
    • Apple Safari
    • Opera
    • Google Chrome
Secure Sockets Layer and HTTPS

- HTTP not secure
- Requirements for secure Internet apps
  - Authentication
  - Encryption
  - Nonrepudiation
- SSL and HTTPS offer security
Secure Sockets Layer (SSL)
- Netscape-developed protocol
- Encrypts data with a public key
- Sends encrypted data over an SSL connection
- Data decrypted on receiving end with private key
- Supported by Web browsers and servers
- Many Web sites use SSL for confidential data
- Look for HTTPS or small lock in browser
HTTP over SSL

- Uses TCP port 443
- Being replaced by Transport Layer Security (TLS)
Figure 9.18  Secure Web Page
• Telnet

- First networks were dumb terminals connected to more than one mainframe
- Run commands as if sitting at the mainframe
- Still exists as a way to connect remotely
- Uses port 23
- Used to administer servers
- Requires long on with user name and password
Ubuntu 8.04.1
VMubunti login: vmuser
Password:
Last login: Mon Nov 10 11:30:01 CST 2008 from michaels.totalhome on pts/1
Linux VMubunti 2.6.24-19-generic #1 SMP Fri Jul 11 23:41:49 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
vmuser@VMubunti:~$
Telnet (cont.)

- Has no form of encryption
- Rarely used on the Internet
- Replaced by Secure Shell (SSH), which has encryption
- Telnet still used on trusted networks
- Most routers support Telnet (often turned off for security)
• **Telnet (cont.)**
  
  - Most OSs have built-in Telnet clients and servers
  - Most servers allow access using Telnet
  - Third-party clients and servers have more features
Figure 9.21  freeSSHD
- **Telnet (cont.)**

  - Configuring a Telnet client
    - Host name (name or IP address)
    - User login name
    - Password
Figure 9.22  Ubuntu Telnet
Figure 9.23 PuTTY
• Rlogin, RSH, and RCP
  – Old UNIX remote programs
  – Remote access and control of servers
  – No encryption
  – Do not use across the Internet

• Rlogin – interactive, automatic login, TCP port 513
• RSH – non-interactive, sends a single command to server, use in scripts, TCP port 514
• RCP – copy files, use in scripts, shares TCP port 514 with RSH
• SSH and the Death of Telnet
  – Has replaced Telnet
  – Encrypts data
  – Creates a terminal connection to remote host
  – TCP port 22
• Electronic mail (e-mail)
  – Major contributor to Internet revolution
  – Streamlined junk mail industry
  – Provides quick way for people to communicate
  – Sends messages and attachments
  – Normally offered free by ISPs
  – Most e-mail clients have simple text editors
• Electronic mail (e-mail)
  - Messages stored on e-mail server
  - Most e-mail clients notify you when new message arrives or automatically download
  - You manage messages (forward, delete, etc.)
  - Most clients delete downloaded messages
  - E-mail programs use application-level protocols
• Simple Mail Transfer Protocol (SMTP)
  – Used by clients to send e-mail
  – TCP port 25
• **Post Office Protocol version 3 (POP3)**
  - Clients use to retrieve e-mail from SMTP servers
  - TCP port 110
  - Used by most e-mail clients
Internet Message Access Protocol version 4 (IMAP4)

- Alternative to POP#
- Retrieves e-mail from an e-mail server
- TCP port 143
- Supports features not supported by POP3
  - Search messages by keyword
  - Select messages before download
  - Supports folders on IMAP4 servers
• Alternatives to SMTP, POP3, and IMAP4

  – Web-based e-mail
    • Access your e-mail from anywhere
    • Free
    • Handy for throw-away accounts
    • Do not confuse with Web-based e-mail services provided by traditional SMTP/POP/IMAP accounts

  – Proprietary solutions
Figure 9.24  Gmail in action
- E-mail Server software
  - E-mail server market fragmented
  - Sendmail for UNIX/Linux is leader (SMTP only)
    - No GUI interface
    - Third-part interfaces (Webmin)
    - Controls about 20% of e-mail servers
    - Must use a POP3 or IMAP server program to support e-mail clients
  - Eudora’s Qpopper sends mail to POP3 e-mail clients
Figure 9.25  Webmin with the sendmail module
• E-mail Server software (cont.)
  – MS Exchange Server (both SMTP and POP3)
  – Mailboxes are holding areas on server for each user’s messages
  – Server arranges incoming messages
  – Server returns messages with unknown recipient
  – Difficult to manage
Figure 9.26  Microsoft Exchange Server
- E-mail Client Software
  - Enables you to send, receive, and organize
  - Communicates with SMTP server to send
  - Communicates with IMAP or POP3 server to download messages
  - Hundreds of e-mail client programs
    - Microsoft Windows Mail
    - Microsoft Outlook
    - Mozilla Thunderbird
    - Qualcomm Eudora
Figure 9.27  Windows Mail
Figure 9.28  Entering server information in Windows Mail
• Configuring E-mail Client Software
  – Obtain server’s address and your mailbox user name and password
  – Enter POP3 or IMAP4 server’s IP address
  – Enter user name and password
• File Transfer Protocol (FTP)
  – Original Internet file transfer protocol
  – Faster and more reliable than HTTP
  – Includes security and data integrity
  – TCP ports 20 and 21
  – Anonymous or secured sites
  – Some are both
• FTP Servers
  – Store files
  – Accept incoming connections
  – Verify user names and passwords
  – Transfer files
  – Easy to set up an FTP server
  – UNIX/Linux have built-in FTP servers
  – Third-party servers better
• FTP Clients
  – Access FTP servers many ways
    • Web site
    • Command line
    • FTP client applications
  – Most Web browsers support FTP, but lack features
  – Dedicated FTP clients work best
    • FileZilla client
    • Mozilla FireFTP add-on to Firefox
Figure 9.29  FileZilla Server
Passive vs. Active FTP

- Traditional FTP uses active process
  - Clients send FTP request on TCP port 21
  - Server responds on an ephemeral destination port with TCP port 20 as the source port
- Passive FTP server doesn’t use port 20
  - Works with NAT
  - Client must support passive FTP


**Figure 9.30**  FTP in Web browser

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Last Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCS</td>
<td>2 KB</td>
<td>9/23/2008 11:53 PM</td>
</tr>
<tr>
<td>README</td>
<td>8 KB</td>
<td>8/5/2007 12:00 AM</td>
</tr>
<tr>
<td>README_ABOUT_BZ2_FILES</td>
<td>1 KB</td>
<td>3/18/2003 12:00 AM</td>
</tr>
<tr>
<td>dist</td>
<td>7 KB</td>
<td>7/22/2005 12:00 AM</td>
</tr>
<tr>
<td>index.html</td>
<td>3 KB</td>
<td>9/23/2008 11:53 PM</td>
</tr>
<tr>
<td>linux</td>
<td>6 KB</td>
<td>6/25/2008 12:37 AM</td>
</tr>
<tr>
<td>lost+found</td>
<td>10 KB</td>
<td>10/27/1998 12:00 AM</td>
</tr>
<tr>
<td>media</td>
<td>9 KB</td>
<td>9/23/2008 11:35 PM</td>
</tr>
<tr>
<td>scrn</td>
<td>11 KB</td>
<td>6/2/2008 6:27 PM</td>
</tr>
<tr>
<td>site</td>
<td>18 KB</td>
<td>11/5/2003 12:00 AM</td>
</tr>
<tr>
<td>software</td>
<td>9 KB</td>
<td>9/26/2008 11:43 AM</td>
</tr>
<tr>
<td>tools</td>
<td>4 KB</td>
<td>4/30/2008 12:00 AM</td>
</tr>
</tbody>
</table>
Figure 9.31  Author’s FireFTP hard at work
<table>
<thead>
<tr>
<th>Application</th>
<th>TCP/UDP</th>
<th>Port</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>TCP</td>
<td>80</td>
<td>The Web</td>
</tr>
<tr>
<td>HTTPS</td>
<td>TCP</td>
<td>443</td>
<td>The Web, securely</td>
</tr>
<tr>
<td>Telnet</td>
<td>TCP</td>
<td>23</td>
<td>Terminal emulation</td>
</tr>
<tr>
<td>SSH</td>
<td>TCP</td>
<td>22</td>
<td>Secure terminal emulation</td>
</tr>
<tr>
<td>SMTP</td>
<td>TCP</td>
<td>25</td>
<td>Sending e-mail</td>
</tr>
<tr>
<td>POP3</td>
<td>TCP</td>
<td>110</td>
<td>E-mail delivery</td>
</tr>
<tr>
<td>IMAP4</td>
<td>TCP</td>
<td>143</td>
<td>E-mail delivery</td>
</tr>
<tr>
<td>FTP</td>
<td>TCP</td>
<td>20/21</td>
<td>File transfer</td>
</tr>
<tr>
<td>TFTP</td>
<td>UDP</td>
<td>69</td>
<td>File transfer</td>
</tr>
</tbody>
</table>