Applied Networks & Security

TCP/IP Networks – with Critical Analysis

http://condor.depaul.edu/~jkrystof/it263/

John Kristoff
jtk@depaul.edu
Critical analysis disclaimer

Following this disclaimer are slides used in other versions of the course. We *mark up* some slides using *strikethroughs* and *underlined red in comic sans ms 20pt font*. This is not meant to slight other teachers or their material. Much of the material is good and helpful so we use it.

We do this to explore complex issues, refresh dated material, correct inaccuracies and stimulate critical thinking. In some cases we are pedantic where it seems useful, but we are not exhaustive and try to avoid being overly tedious when it is unnecessary.
This Week’s Class Topics

- Layered Protocols
- Ethernet Local Area Network (LAN) basics
- Ethernet data delivery (more on Ethernet next week)
- IP data delivery
- TCP services
- UDP services
- Port numbers and firewalls
Web Service over a LAN
3-Layer Protocol Stack

Browser
  OS
  NIC Card / Driver

HTTP Protocol
  TCP/IP Protocols Stack
  Data transmission (Ethernet) Protocol

Server
  OS
  NIC Card / Driver

LAN Network
Historical Note: OSI 7-layer model

- The original layered protocol model was the 7-layer Open Systems Interconnect model (1977)
  - Theoretical model used to describe 7 separate layers of functionality required for end-to-end data communications
- We’ll just use the 3 or 4-layer Internet Model in this class.
  - The layers are:
    - Application
    - TCP / IP (no IP is its own layer)
    - Datalink and Physical
The 7 OSI Layers

- Layer 7: Application Layer (ex: HTTP)
- Layer 6: Presentation Layer (ex: codeset translation)
- Layer 5: Session Layer (ex: session management)
- Layer 4: Transport Layer (ex: TCP)
- Layer 3: Network Layer (ex: IP)
- Layer 2: Data Link Layer (ex. Ethernet Framing)
- Layer 1: Physical Layer (ex. Ethernet Hardware)
Layering Example

WinXP PC running FireFox

User clicks “http://www.depaul.edu”

FireFox Software (Client App)

TCP Software

IP Software

Ethernet Driver

Ethernet Card

WinXP OS

Create GET Request in HTTP format

Add TCP Header

Add IP Header

Add Ethernet Header & Trailer

Receive and Process GET Request

Remove Ethernet Header & Trailer

Remove IP Header

Remove TCP Header

WinXP OS

Apache Software (Server App)

TCP Software

IP Software

Ethernet Driver

Ethernet Card

LINUX OS

www.depaul.edu

Linux PC running Apache
Why All The Layers?

Why do we need multiple layers? Why not just use IP and forget everything else?

- The Internet is a very complex system: network application developers should not have to deal with all the complexities of networks;
- “Code re-use”: many network applications require similar “services” from the network;
- Implementation of a layer can change without the other layers needing to know about it -- as long as the modified layer provides the same services to the layer above it and uses the same services of the layer below it.
Actual examples of layers

- We will now look at the specific technologies for each layer that we will focus on in class:
  - Ethernet
  - TCP/IP
What is an Ethernet LAN?

- A group of PCs connected into a common **hub** device (which may also be a **switch**) that allows each to send data to all others.

- Each LAN PC has an Ethernet interface called a Network Interface Card (NIC) or adapter.
The Network Interface Card (NIC) and Networking Software (NS) must be compatible with each other and with the computer or device into which they are installed.
Typical NICs

**Figure 2.6** Ethernet Industry Standard Architecture (ISA) Bus (right) and Peripheral Component Interconnect (PCI) Bus (left) NICs.
Typical NICs

Figure 2.7
PCI Ethernet NIC RJ-45 or Mod-8 connector and light-emitting diode 100 Mbps and FDX indicators.
Ethernet was developed in the 1970s as a simple way to provide communications among multiple devices in the same building.

Ethernet works at the hardware level – every device has standardized Ethernet hardware sending and receiving bits.
Ethernet

- Star-wired Bus Topology - not originally, physical bus
- Logically operates as a bus, but physically looks like a star.
- Star design is based on hub. All workstations attach to hub.
- Unshielded twisted pair usually used to connect workstation to hub.
- Hub takes incoming signal and immediately repeat it out all connected links.
- Hubs can be interconnected to extend size of network.
- Nowadays, hubs are rarely used and replaced by switches. More on the difference in a future lecture.
Ethernet

Diagram of a network with workstations, server, printer, and workstation connected to a hub.
Ethernet
Ethernet LANs

- Every Ethernet interface (NIC) has a 6-byte **physical address** (also called **MAC (medium access control) address**) assigned and burned into hardware when it is manufactured.
- MAC address is like a serial number.
- MAC address of every Ethernet device is **(theoretically) guaranteed to be globally unique.**
### Ethernet MAC Address format

**(length = 6 bytes = 48 bits)**

<table>
<thead>
<tr>
<th>Manufact. ID (22 bits)</th>
<th>Serial Number (24 bits)</th>
</tr>
</thead>
</table>

- Globally/Locally (0/1) Administered bit
- Unicast/Multicast (0/1) bit

- Manufacturer IDs are uniquely assigned to Ethernet equipment manufacturers by IEEE (Institute for Electrical and Electronics Engineers).
- Each manufacturer ensures that each Ethernet interface on every device they make has a unique Serial Number.
- Result: every Ethernet interface has unique address.
To send data, a PC simply puts the destination MAC address into the Ethernet header and transmits the data frame.

- PC needs no knowledge of the network or how the frame will be delivered.
- The Ethernet frame gets delivered unchanged to destination.

Originally, Ethernet hubs simply repeats every frame to every device on the LAN.

- Destination PC recognize that the destination MAC is its own, copies frame, all others ignored it (except for broadcast and relevant group addresses).
Ethernet Frame

- Ethernet frame header:
  - **Preamble** field contains fixed bit values for synchronizing sender and receiver clocks.
  - **Destination** and **Source** MAC addresses (6 bytes each).
  - **Ethernet Type** field used to identify the protocol carried inside the frame (IP, ARP, AppleTalk, etc.)

- Ethernet frame trailer
  - **FCS** used for error checking
Example: PC #1 puts MAC address “#2” into Destination Address field of Ethernet header and transmits data frame to Hub

Hub repeats the frame to #2, #3 and #4

#2 copies frame while #3 and #4 ignore it (not their address)
“Layer 2 Ethernet Devices”

- "Layer 2 Ethernet devices" utilize only the Ethernet header in a data frame. (they do not understand TCP or IP or any other protocol). Examples are:
  - Ethernet interface hardware (NICs)
  - Ethernet hubs
  - Layer 2 Ethernet switches
The IP World

- **Internet Protocol** was developed in the 1970s as a way to provide communications among IP devices (called “IP hosts”) located anywhere in the world.
- IP is implemented in the operating system software of each IP Host on the network.
- IP Hosts are grouped into IP Subnets (which are usually Ethernet LANs), which are interconnected by IP Routers.
- IP software is not concerned with how bits are physically transmitted between devices – this is left to a lower level protocol (such as Ethernet).
The Internet Protocol (IP)

- Layer 3 of OSI model
- IP prepares a packet for transmission across the Internet.
- The IP header is encapsulated onto a transport data packet.
- The IP packet is then passed to the next layer where further network information is encapsulated onto it.
- IP allows end-to-end transmission of a packet (layer 3 function)
- Ethernet (or any other layer 2 technology) allows transmission of a packet from one point to another point that it directly connects to.
The Internet Protocol (IP)

- IP Addresses
- For example, the 32-bit binary address
  - 10000000 10011100 00001110 00000111
- translates to
  - 128.156.14.7
IP Addresses

- Each **IP address** is **4 bytes** long
- Dotted decimal notation
  - Each byte (8 bits) is written in decimal separated by dots, like
  - Each of the 4 values is in range 0 - 255.
  - Example: 150.21.39.52
IP Addressing

- **IP Addresses**
- All devices connected to the Internet have a 32-bit IP address associated with it.
- Think of the IP address as a logical address (possibly temporary), while the 48-bit address on every NIC is the physical, or permanent address.
- Computers, networks, and routers use the 32-bit binary address, but a more readable form is the dotted decimal notation.
IP Data Delivery

- IP routing only provides “best effort data delivery”
  - Does not guarantee delivery of data packets
  - Does not guarantee that data packets will be delivered in the same order they were sent.
- Just like the postal services.
IP Addresses

- IP addressing is **hierarchical**.
- Each **IP address** contains 3 parts: *(not that simple)*
  - An **IP Network** part that is used by Internet backbone routers to deliver packets to a particular IP Network. IP Network values are assigned by Internet Assigned Numbers Authority ([www.iana.org](http://www.iana.org)) to guarantee global uniqueness.
    - No, they manage the entire address space, but delegate to RIRs, who in turn allocate to network operators.
  - An **IP Subnet** part that is used by internal routers within an IP Network to deliver packets to a particular Subnet.
  - An **IP Host** part that identifies a particular individual device on the subnet.
### Address Example

<table>
<thead>
<tr>
<th>Network</th>
<th>Subnet</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>130</td>
<td>88</td>
<td>55</td>
</tr>
</tbody>
</table>

- **Network** = 130.88
- **Subnet** = 55
- **Host** = 12
DePaul University was assigned IP Network prefix 140.192 by the IANA back in the 1980s. DePaul controls all IP addresses that start with 140.192.

- DePaul Information Services (IS) assigns Subnet IDs to various departments and groups at the university. For example:
  - IP subnet 140.192.32 – CTI servers
  - IP subnet 140.192.34 – 6th and 7th floor CTI office PCs
  - IP subnet 140.192.35 – 8th and 9th floor CTI office PCs

- Individual devices in each subnet are then each assigned a unique Host ID, either manually or automatically (using Dynamic Host Configuration Protocol (DHCP)).
IP Address Assignment

- How does a PC get assigned an individual IP address? Two methods:
  - **Static Assignment** – IP address is manually entered into PC by administrator.
  - **Automatic Assignment** – IP address is dynamically assigned by a DHCP server each time PC powers up.
DHCP (Dynamic Host Configuration Protocol) servers automatically allocate IP addresses to host machines as they power up.

- When PC powers up, sends broadcast to DHCP server.
- DHCP server dynamically chooses an available IP host address from a pool of IP addresses for that subnet.
- Address has a renewal time associated so it can be recovered and reused even when if PC is suddenly powered off.
- Server can also assign other parameters to PC, such as DNS server address, local router address, etc.
To send IP data, a Source Host PC must determine whether the destination is on the same IP subnet or different one.

- If destination host is on the same IP subnet as source host, then data packet gets delivered directly to destination using Ethernet delivery.
- If destination host is on a different IP subnet from source host, then source host delivers the packet to the nearest IP router using Ethernet delivery.
  - IP router then determines how to deliver packet to the destination host.
IP Delivery Examples

- My PC’s IP address is 140.192.34.162.
- To deliver an IP packet to Sam’s PC down the hall (IP address 140.192.34.77) on the same IP subnet (same Ethernet network):
  - My PC creates an Ethernet frame with Sam’s Ethernet address in Destination field of Ethernet header and sends it.
  - How do I determine Sam’s Ethernet (MAC) address? I ask him broadcast using Address Resolution Protocol (ARP)
- To deliver an IP packet to www.microsoft.com (IP address 207.46.20.60), which is on a different IP subnet:
  - My PC creates an Ethernet frame with a local DePaul router’s Ethernet address in the Destination field of the Ethernet header and sends it.
  - The DePaul router then figures out how to forward the packet towards the destination IP address 207.46.20.60, which is found in the Destination address field of the IP Header.
- How does my PC find out the IP addresses and Ethernet addresses for other devices? We’ll cover more on that later (DNS and ARP)
How do you find your address?

- Address information (both MAC and IP) for a network connection by clicking the following: Start -> Control Panel -> Network Connection -> Local Area Connection -> Support (WinXP) or by running “ipconfig /all” in a DOS window.
How do you find someone else’s address?

- You can generally determine the IP address of any IP host (if you know its name) using the “ping” program in a Command Window - no, just use DNS (dig, host or nslookup command)
IP Header

- IP adds (at least, but almost always in practice) 20 bytes of *IP Header* to every data packet.
- These 20 bytes include all information needed by IP routers to direct this packet to its destination.
# IP Header

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4-bit value</td>
</tr>
<tr>
<td>IHL</td>
<td>1-bit value</td>
</tr>
<tr>
<td>Type of service</td>
<td>8-bit value</td>
</tr>
<tr>
<td>Total length</td>
<td>16-bit value</td>
</tr>
<tr>
<td>Identification</td>
<td>13-bit value</td>
</tr>
<tr>
<td>DF</td>
<td>Bit to indicate fragment</td>
</tr>
<tr>
<td>MF</td>
<td>Bit to indicate fragment</td>
</tr>
<tr>
<td>Fragment offset</td>
<td>13-bit value</td>
</tr>
<tr>
<td>Time to live</td>
<td>8-bit value</td>
</tr>
<tr>
<td>Protocol</td>
<td>8-bit value</td>
</tr>
<tr>
<td>Header checksum</td>
<td>16-bit value</td>
</tr>
<tr>
<td>Source address</td>
<td>32-bit value</td>
</tr>
<tr>
<td>Destination address</td>
<td>32-bit value</td>
</tr>
<tr>
<td>Options</td>
<td>0 or more 32-bit words</td>
</tr>
</tbody>
</table>
IP Header Fields

- **Version**: IP protocol version. We are currently on IP version 4. Soon the Internet may be moved to IP version 6. IPv4 ubiquitous, IPv6 deployment in progress.

- **IP Header Length**: Length of IP header in 32-bit words.

- **Type of Service**: Indicates whether this packet should be low or high priority (not quite).

- **Total Length**: Length of IP packet in bytes.
IP Header Fields

- **Identification / Fragment Offset**: used to identify *fragments* that are formed when an IP packet needs to be broken up into smaller packets.

- **Time to Live**: Max. number of routers this IP packet may pass through. If exceeded, packet will be discarded.

- **Protocol**: The Layer-4 *encapsulated* protocol carried inside this IP packet.
IP Header Fields

- **Header Checksum**: Allows error checking of IP packets.
- **Source Address**: 4-byte IP source address for this packet.
- **Destination Address**: 4-byte IP destination address for this packet.
Transmission Control Protocol (TCP)

- Provides reliable data delivery services
- Connection oriented
  - Requires the establishment of a connection between communicating nodes before the protocol will transmit data
- TCP segment
  - Holds the TCP data fields
  - Becomes encapsulated by the IP datagram
Addressing Example

GET
DEFAULT.HTM
HTTP/1.0

Dest. IP Addr = 140.192.81.6
Dest. Port = 80

Browser → Web Server
Transmission Control Protocol

- TCP provides sequence numbers and Acknowledgement bits to provide Error Control between Clients and Servers
  - NO! To provide flow control and reliable in-order reassembly
- TCP adds (at least) 20 bytes of **TCP Header** to the data packet
Who looks at the TCP header?

- TCP headers are generally ONLY examined and processed by the following devices:
  - The sending device (client or server)
  - The receiving device (client or server)
  - Firewalls (NAT/PNAT, packet shapers, proxies - and all kinds of “middle” boxes today)

- In particular, IP routers and Ethernet devices do not (usually) examine or process TCP headers at all (unless they are multi-purpose devices, such as IP router/firewall combinations).
TCP Header

- **Source port**
- **Destination port**
- **Sequence number**
- **Acknowledgement number**
- **Window size**
- **Checksum**
- **Urgent pointer**
- **Options** (0 or more 32-bit words)
- **Data** (optional)
TCP Header Fields

- **Source / Dest Port**: Source port number and destination port number for this packet.

- **Sequence Number / Acknowledgement Number**: used for error reliability and flow control.

- **TCP Header Length**: Indicates whether any options are used.
TCP Header Fields

- **Window Size**: For flow control – number of bytes other end can send without buffer overflow
- **Checksum**: Used for error checking
- **Urgent pointer**: Can indicate high-priority data within packet.
TCP Header Field Bits

- **URG bit**: Indicates whether there is urgent data in this packet,
- **ACK bit**: Indicates whether this packet is (also) acknowledging another packet, received bytes
- **PSH bit**: Indicates whether this data should be quickly pushed up to application program at receiver.
TCP Header Field Bits

- **RST bit**: Used to reset communications session
- **SYN bit**: Indicates a request to set up a new communications session (synchronize)
- **FIN bit**: Indicates final packet – closes down a communications session.
What does TCP do?

- Provides error-free \textit{reliable} in-order data delivery between Client and Server
- Sets up connections across the Internet between Client and Server
- Reorders data if it arrives out-of-order
- Detects errors and re-transmits data if errors occur
- Congestion Control: Automatically slows down if Internet is too busy
What does TCP do?

- Sets up connections across the Internet between Client and Server
  - TCP uses a **3-way handshake** to set up a connection
    - Client → Server: SYN bit set
    - Server → Client: SYN, ACK bits set
    - Client → Server: ACK bit set
What does TCP do?

- Reorders data if it arrives out-of-order
  - TCP assigns a sequence number to each byte of data sent
  - TCP software delivers data in sequence number order to the receiving application
  - Out-of-order data is buffered until it can be delivered to application program
  - **Window** field is used to tell sender how much more data can be sent before buffers are full.
Error Detection

- Sender calculates a CRC Code checksum (typically 4 bytes) over all the data bytes in the packet.
- The Sender transmits the calculated CRC code at the end of the data packet.
- The Receiver calculates a new CRC code based on received data and compares it with CRC in packet.
  - If these CRC values match, data is correct!!
  - If CRC values do not match, some data error has occurred and packet is discarded.
Error Correction using TCP

- Sender puts sequence number in each packet indicating the byte number of first byte in packet.
- Sender starts timer for each packet sent.
- Receiver sends back acknowledgement packet containing sequence number of the next packet expected.
- If sender’s timer expires before corresponding acknowledgement is received:
  - Re-send packet
  - Re-start timer

This is reliability, not “error correction”
TCP: Transmission Error
The User Datagram Protocol (UDP) is typically used by application programs that do not need Error Control transport layer services.

UDP is a Transport Layer protocol, but just barely.

Applications that use UDP rather than TCP: streaming audio or video transfer, network management applications.

UDP adds 8 bytes of UDP Header.
**UDP Header**

<table>
<thead>
<tr>
<th>Source port</th>
<th>Destination port</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP length</td>
<td>UDP checksum</td>
</tr>
</tbody>
</table>
TCP vs. UDP

- TCP provides end-to-end error checking detection. UDP does not.

- Applications that use TCP/IP
  - HTTP to access web pages
  - SMTP to send e-mail

- Applications that use UDP/IP
  - Streaming audio or video
  - Polling the status of a device
Network Applications

- Both TCP and UDP use a **port number** to identify the particular application that a packet pertains to.

- Port numbers are registered by via the **Internet Assigned Numbers Authority (IANA – www.iana.org)**.
Port Number Ranges

3 defined ranges of port numbers:

- **Well Known Ports** (0-1023)
  - These port numbers are assigned by IANA to correspond to specific globally recognized applications.

- **Registered Ports** (1024-49151)
  - These port numbers are assigned by software vendors to applications. IANA will register any port number in this range upon request. **No it won't, you have to justify it.** Global use of this number for this purpose is recommended, but not required.

- **Dynamic/Private Ports** (49151-65535)
  - These port numbers have no fixed global assignment.

See [http://www.iana.org/assignments/port-numbers](http://www.iana.org/assignments/port-numbers)
Network Services

- A network service is defined by:
  - A port number (TCP or UDP) that is used by the client machine to request connection to the service on the server.
  - A service process running on the server that listens for messages containing that port number, accepts connections, processes requests and sends back responses.
  - A protocol that defines acceptable contents and format for the request and response messages.

- Example: Web service on a Linux server might be provided on TCP port 80 via an Apache web server application process following the HTTP protocol.
TCP example: Web service

- Example: Web service on a Linux server could be provided on TCP port 80 via an Apache web server application process following the HTTP protocol.

- To access this service, a client browser will:
  - Open a connection to port 80 on server (using TCP 3-way handshake)
  - Send an HTTP command (such as “GET <file>”) to the server
  - Wait for reply from server
  - Display results
  - Close TCP connection.
Example: Ping service is typically provided on UDP using ICMP message type 7 via a general OS network service process using the ICMP protocol.

To access this service, a client browser will:

- Send an ICMP Echo Request message using UDP packet to the server port 7 destination host
- Wait for reply from server
- Display results
A key aspect of security on modern networks is choosing what network services should be offered on the network (i.e. what ports are open).

- Some services have known vulnerabilities that allow denial-of-service or server compromising.

- Each server can be configured to accept or deny requests to each port.

- Firewalls/routers can be configured to block all messages to particular port numbers.
What ports are open?

- On Windows, UNIX or Linux, the `netstat` command displays open ports and active connections.

  Example: Windows XP: “netstat –a”
What services are offered?

Under Windows, all services can be displayed under Services on the Computer Management page — may have little to do with networking.

- To open Computer Management, click Start, and then click Control Panel. Click Performance and Maintenance, click Administrative Tools, and then double-click Computer Management.
Win XP Security

Windows XP Firewall

- XP provides a built-in personal firewall service that can block/allow any port on any network connection.
- Security logging can also be turned on – which writes a message to a log file every time a TCP or UDP port connection is attempted or established.
Windows XP Firewall

Windows Firewall blocks most incoming service requests by default, but allows the user to specify exceptions by service name or port number.