18-345: Introduction to Telecommunication Networks Lectures 2: Protocol Stack

Peter Steenkiste

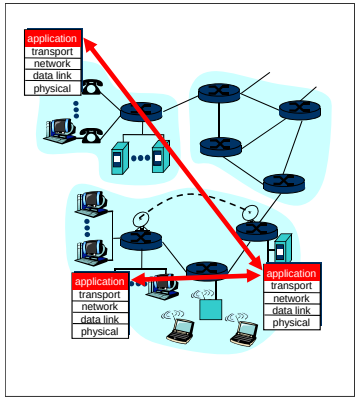
Spring 2015
www.cs.cmu.edu/~prs/nets-ece

Today's Lecture

- Network applications
  - Requirements
  - Latency and bandwidth
- Internet architecture
  - Protocols
  - A layered design
  - Life of a packet
- Network utilities

Applications and Application Protocols

- Application: communicating, distributed processes
  - Running in network hosts in "user space"
  - Exchange messages to implement app
  - e.g., email, file transfer, the Web
- Application protocols
  - One "piece" of an app
  - Define messages exchanged by apps and actions taken
  - User services provided by lower layer protocols



Client-Server Paradigm

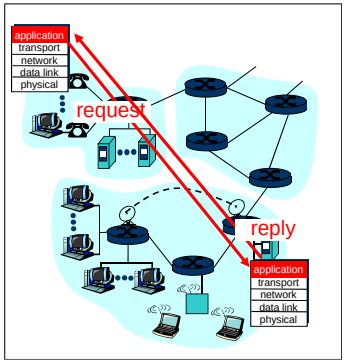
Typical network app has two pieces: *client* and *server*

Client:

- Initiates contact with server ("speaks first")
- Typically requests service from server,
- For Web, client is implemented in browser; for e-mail, in mail reader

Server:

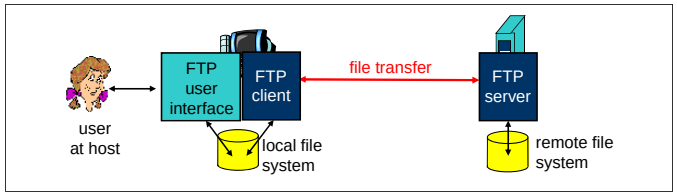
- Provides a service to client
- e.g., Web server sends requested Web page, mail server delivers e-mail



Yesterday's Applications

- FTP: transfer files to a host
  - No distributed file systems!
  - Mostly replaced by "the web" – http
- Telnet: use a computer remotely
  - Similar to ssh today (minus the security)
- Mail: exchange electronic e-mail
  - Similar today (kind of)
  - Initially host-to-host: name@my.computer.edu
- Already very useful!

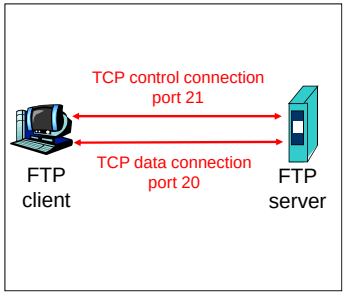
FTP: The File Transfer Protocol



- Transfer file to/from remote host
- Client/server model that allows a host (client) to transfer a file to/from another host (server)
- Application is a minimal wrapper - a command line "user interface"
  - All the heavy lifting is done in the protocol implementation
- Project: will learn about HTTP
  - Same idea but much richer functionality

FTP: Separate Control, Data Connections

- Ftp client contacts ftp server at port 21, specifying TCP as transport protocol
- Two parallel TCP connections opened:
  - Control: exchange commands, responses between client, server.
    "out of band control"
  - Data: file data to/from server
- Ftp server maintains "state": current directory, earlier authentication



Ftp Commands, Responses

Sample Commands:

- sent as ASCII text over control channel
- `USER` *`username`*
- `PASS` *`password`*
- `LIST` return list of files in current directory
- `RETR filename` retrieves (gets) file
- `STOR filename` stores (puts) file onto remote host

Sample Return Codes

- status code and phrase
- `331 Username OK, password required`
- `125 data connection already open; transfer starting`
- `425 Can't open data connection`
- `452 Error writing file`

Today's Applications

- Amazon, Facebook, etc.
  - What matters most?
  - 2009 quote: "*Amazon* found every 100ms of *latency* cost them 1% in sales"
- Video streaming
  - Accounts for very high percentage of bandwidth
  - Interactive versus broadcast versus playback
  - What matters most?
- Skype audio and video conferencing
  - Traditional telephone app
  - What matters most?

Requirements

- Performance: latency and throughput
- Network reliability
  - Network service must always be available
- Security: for users and the network itself
  - Privacy, authentication, deal with various attacks, …
  - Attacks on the network, versus enabled by the network
- Scalability.
  - Scale to large numbers of users, traffic flows, …
- Manageability: monitoring, enforcing policies, billing, …

What Service Does an Application Need?

Data loss

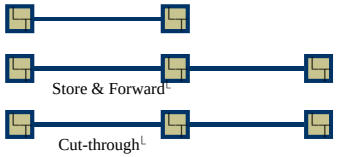
Timing

Bandwidth

Transport Service Requirements of Common Apps

Application	Data loss	Bandwidth	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
web documents	no loss	elastic	no
real-time audio/ video	loss-tolerant	audio: 5Kb-1Mb video:10Kb-5Mb	yes, 100's msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few Kbps	yes, 100's msec
financial apps	no loss	elastic	yes and no

A Closer Look at Packet Delay



Prop + xmit

2*(Prop + xmit)

2*prop + xmit

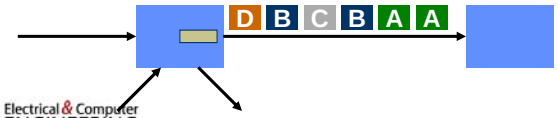
When does cut-through matter?

Next: Routers have finite speed (processing delay)

Routers may buffer packets (queueing delay)

Packet Delay Components

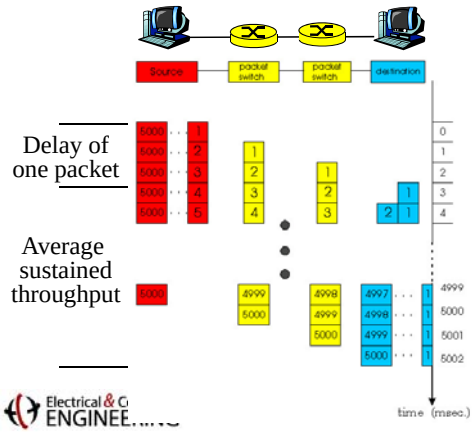
- Sum of a number of different delay components.
- Propagation delay on each link.
  - Proportional to the length of the link
- Transmission delay on each link.
  - Proportional to the packet size and 1/link speed
- Processing delay on each router.
  - Depends on the speed of the router
- Queuing delay on each router.
  - Depends on the traffic load and queue size



A Word about Units

- What do “Kilo” and “Mega” mean?
  - Depends on context
- Storage works in powers of two.
  - 1 Byte = 8 bits
  - 1 KByte = 1024 Bytes
  - 1 MByte = 1024 Kbytes
- Networks work in decimal units.
  - Network hardware sends bits, not Bytes
  - 1 Kbps = 1000 bits per second
  - To avoid confusion, use 1 Kbit/second
- Why? Historical: CS versus ECE.

Application-level Delay



$$\text{Delay}^{*} + \frac{\text{Size}}{\text{Throughput}}$$

Units: seconds + bits/(bits/seconds)

* For minimum sized packet

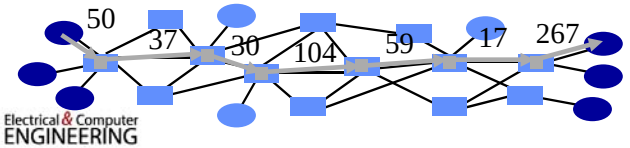
Some Examples

- How long does it take to send a 100 Kbit file?
  - Assume a perfect world
- Is the transfer latency or throughput limited?
- What about a 10 Kbit file?

Latency \ Throughput	100 Kbit/s	1 Mbit/s	100 Mbit/s
500 µsec			
10 msec			
100 msec			

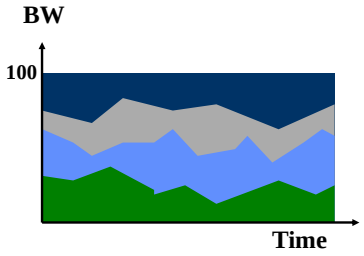
A Closer Look at Throughput

- When streaming packets, the network works like a pipeline.
  - All links forward different packets in parallel
- Throughput is determined by the slowest stage.
  - Called the bottleneck link
- Does not matter why the link is slow!
  - Low link bandwidth
  - Many users sharing the link bandwidth



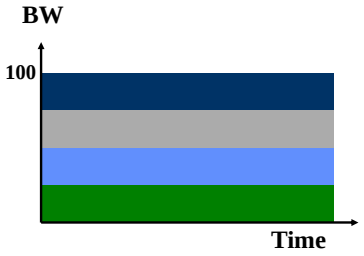
Bandwidth Sharing

- Bandwidth received on the bottleneck link determines end-to-end throughput.
- Router before the bottleneck link decides how much bandwidth each user gets.
  - Users that try to send at a higher rate will see packet loss
- User bandwidth can fluctuate quickly as flows are added or end, or as flows change their transmit rate.

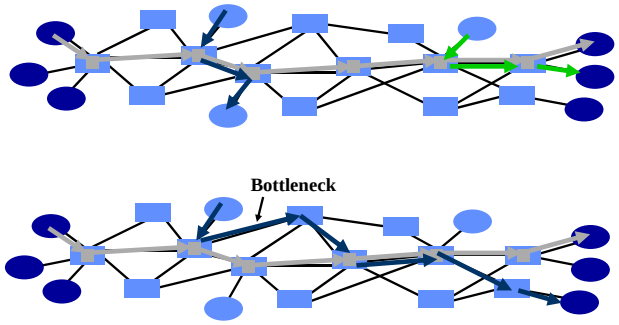


Fair Sharing of Bandwidth

- All else being equal, fair means that users get equal treatment.
  - Sounds fair
- When things are not equal, we need a policy that determines who gets how much bandwidth.
  - Users who pay more get more bandwidth
  - Users with a higher "rank" get more bandwidth
  - Certain classes of applications get priority



But It is Not that Simple



Today's Lecture

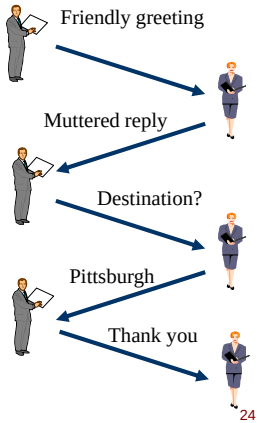
- Network applications
  - Requirements
  - Latency and bandwidth
- Internet architecture
  - Protocols
  - A layered design
  - Life of a packet
- Network utilities

Lots of Protocols (and Acronyms!)

- IP: Internet protocol
- UDP: User datagram protocol
- TCP: Transmission control protocol
- FTP: File transfer protocol
- SMTP: Simple mail transfer protocol
- HTTP: Hypertext transfer protocol
- ARP: Address resolution protocol
- BGP: Border gateway protocol
- ICMP: Internet control message protocol
- DHCP: Dynamic host configuration protocol
- And many more …

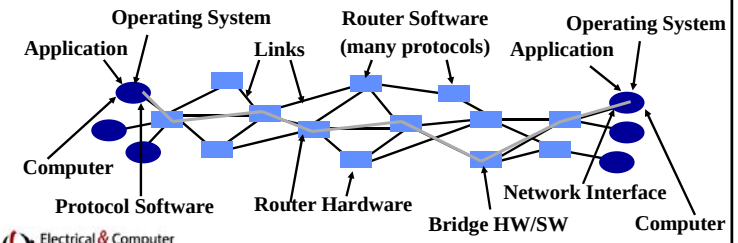
What are Protocols?

- An agreement between parties on how communication should take place
- Module in layered structure
- Protocols define: Interface to peer (syntax & semantics)
  - Actions taken on receipt of a messages
  - Format and order of messages
  - Error handling, termination, ordering of requests, etc.
- Example: Buying airline ticket

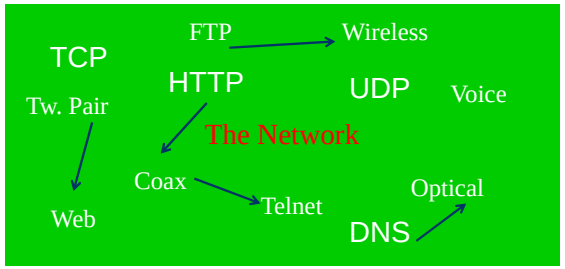


How to Design a Network?

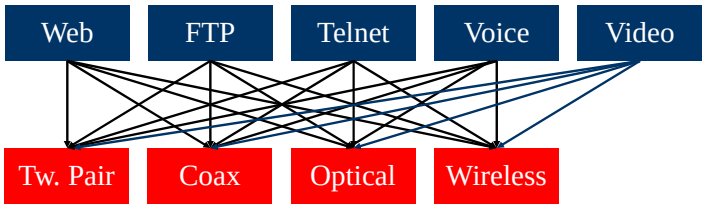
- Has many users
- Offers diverse services
- Mixes very diverse technologies
- Components built by many companies
- Diverse ownership
- Can evolve over time



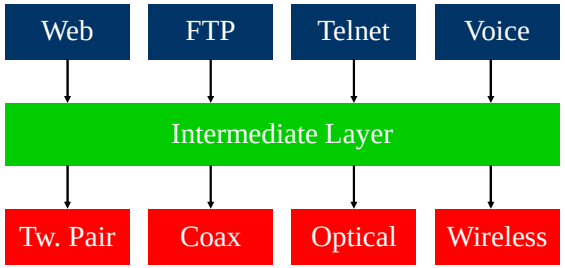
Solution #1



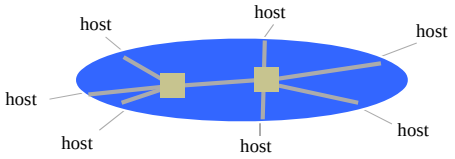
Solution #2?



Solution #3

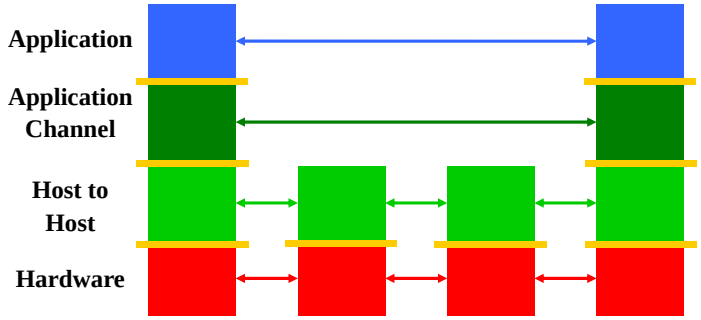


Types of Protocols



- Core network: responsible for transferring data between a sending and receiving host.
- End-to-end protocols: present a network service to applications and users.
  - May add value to the core network protocols
- Driven by differences in constraints: scalability, power, management, speed, etc.

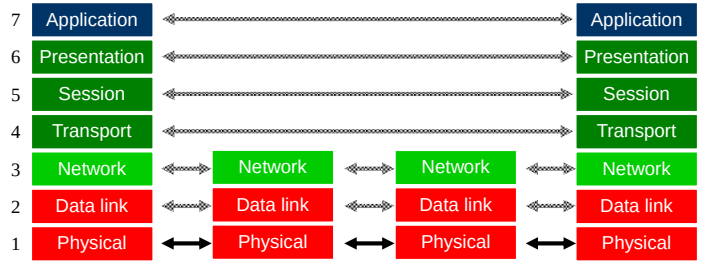
Protocol and Service Levels



Layering: modular approach to network functionality

A Layer Network Model

The Open Systems Interconnection (OSI) Model



Layering Characteristics

- Each layer relies on services from layer below and exports services to layer above
- Interface defines interaction with peer on other hosts – called protocols
- Modules hide implementation - layers can change without disturbing other layers (black box)

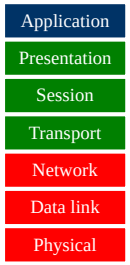
OSI Model: 7 Protocol Layers

- Physical: how to transmit bits
- Data link: how to transmit frames
- Network: how to route packets
- Transport: how to send packets end2end
- Session: how to tie flows together
- Presentation: byte ordering, security
- Application: everything else

- TCP/IP has been amazingly successful, and it is not based on a rigid OSI model. The OSI model has been very successful at shaping thought

Different Sources of Components

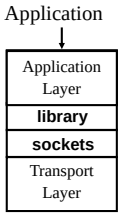
- Application: web server/browser, mail, distributed game,..
- Presentation/session
  - Often part of application
- Transport/network
  - Typically part of the operating system
- Datalink
  - Often written by vendor of the network interface hardware
- Physical
  - Hardware: card and link



Application & Upper Layers

- Application Layer: Provides services that are frequently required by applications: DNS, web acess, file transfer, email…
- Presentation Layer: machine-independent representation of data…
- Session Layer: dialog management, recovery from errors, …

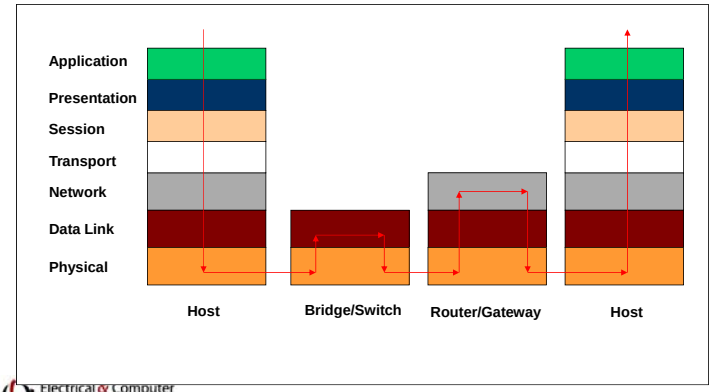
Mostly incorporated into Application Layer



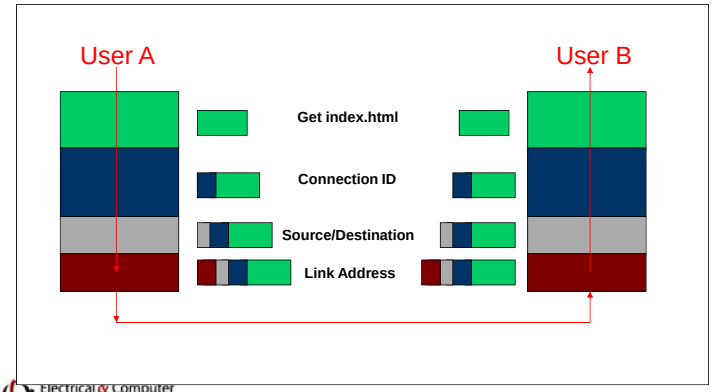
The Internet Engineering Task Force

- Standardization is key to network interoperability
  - The hardware/software of communicating parties are often not built by the same vendor → yet they can communicate because they use the same protocol
- Internet Engineering Task Force
  - Based on working groups that focus on specific issues
- Request for Comments
  - Document that provides information or defines standard
  - Requests feedback from the community
  - Can be "promoted" to standard under certain conditions
    - consensus in the committee
    - interoperating implementations
  - Project 1 will look at the Internet Relay Chat (IRC) RFC

Life of Packet

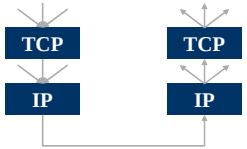


Layer Encapsulation



Multiplexing and Demultiplexing

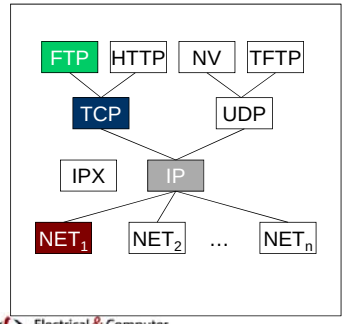
- There may be multiple implementations of each layer.
  - How does the receiver know what version of a layer to use?
- Each header includes a demultiplexing field that is used to identify the next layer.
  - Filled in by the sender
  - Used by the receiver
- Multiplexing occurs at multiple layers. E.g., IP, TCP, …

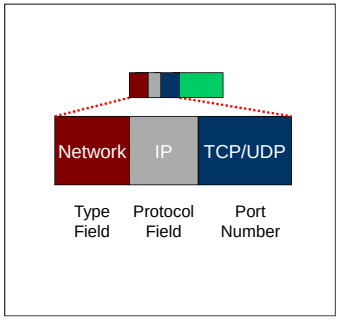


V/HL	TOS	Length
ID		Flags/Offset
TTL	Prot.	H. Checksum
Source IP address		
Destination IP address		
Options..		

Protocol Demultiplexing

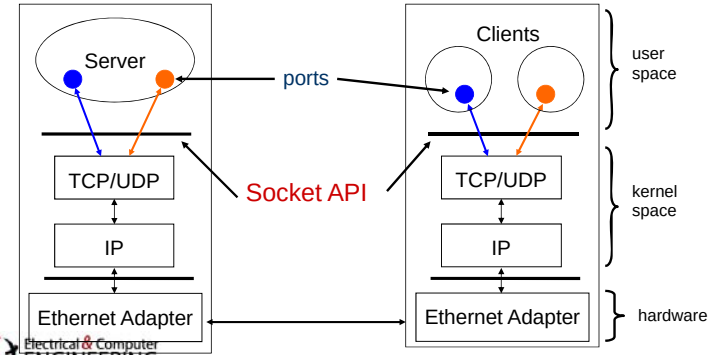
- Multiple choices at each layer



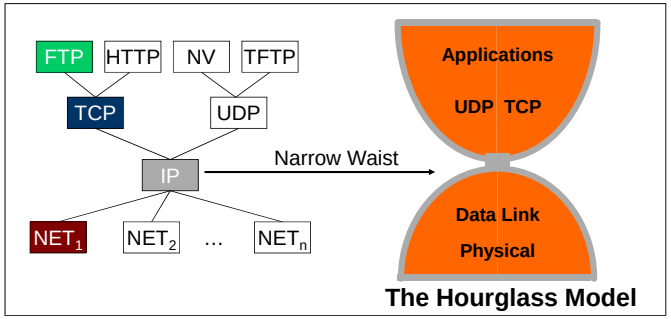


Server and Client

Server and Client exchange messages over the network through a common Socket API



The Internet Protocol Suite



The waist facilitates interoperability
… but evolution is hard

IP based on a Minimalist Approach

- Dumb network
  - IP provide minimal functionalities to support connectivity
    - Addressing, forwarding, routing
- Smart end system
  - Transport layer or application performs more sophisticated functionalities
    - Flow control, error control, congestion control
- Advantages
  - Accommodate heterogeneous technologies (Ethernet, modem, satellite, wireless)
  - Support diverse applications (telnet, ftp, Web, X windows)
  - Decentralized network administration

Today's Lecture

- Network applications
  - Requirements
  - Latency and bandwidth
- Internet architecture
  - A layered design
  - Protocols
  - Life of a packet
- Network utilities

Network tools

- **ping**
- **traceroute**
- **ipconfig**
- **tcpdump**
- **…**

ping

- Application to determine if host is reachable
- Based on Internet Control Message Protocol
  - ICMP informs source host about errors encountered in IP packet processing by routers or by destination host
  - ICMP Echo message requests reply from destination host
- PING sends echo message & sequence #
- Determines reachability & round-trip delay
- Sometimes disabled for security reasons

traceroute

- Find route from local host to a remote host
- Time-to-Live (TTL)
  - IP packets have TTL field that specifies maximum # hops traversed before packet discarded
  - Each router decrements TTL by 1
  - When TTL reaches 0 packet is discarded
- Traceroute
  - Send UDP to remote host with TTL=1
  - First router will reply ICMP Time Exceeded Message
  - Send UDP to remote host with TTL=2, …
  - Each step reveals next router in path to remote host
- **`tracert`** (windows), **`tracepath`** (linux)

ipconfig

- Utility in Microsoft Windows to display TCP/IP information about a host
- Many options
  - Simplest: IP address, subnet mask, default gateway for the host
  - Information about each IP interface of a host
    - DNS hostname, IP addresses of DNS servers, physical address of network card, IP address, …
  - Renew IP address from DHCP server

netstat

- Queries a host about TCP/IP network status
- Status of network drivers & their interface cards
  - #packets in, #packets out, errored packets, …
- State of routing table in host
- TCP/IP active server processes
- TCP active connections

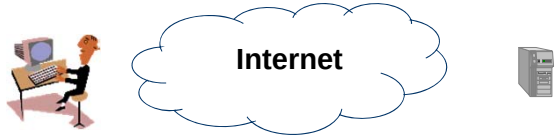
Electrical & Computer ENGINEERING

tcpdump and Network Protocol Analyzers

- `tcpdump` program captures IP packets on a network interface (usually Ethernet NIC)
- Filtering used to select packets of interest
- Packets & higher-layer messages can be displayed and analyzed
- `tcpdump` basis for many network protocol analyzers for troubleshooting networks
- We use the open source Ethereal analyzer to generate examples (or wireshark, etc.)
  - www.ethereal.com

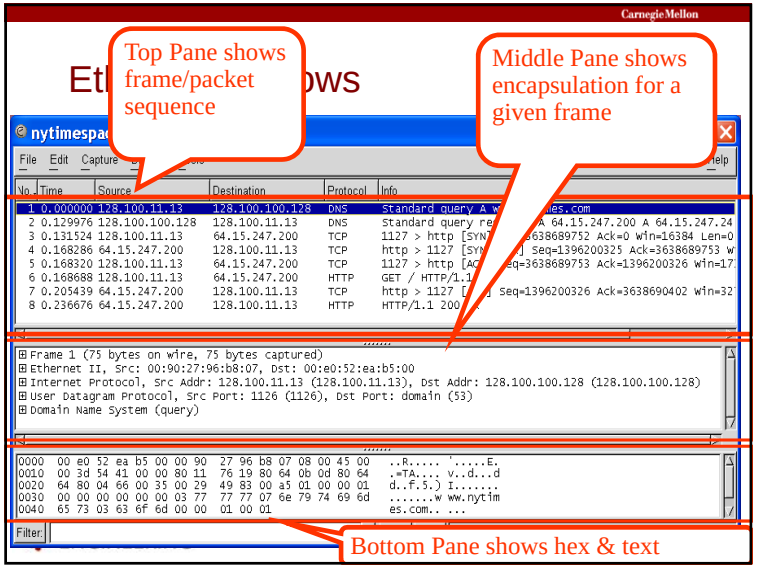
Electrical & Computer ENGINEERING

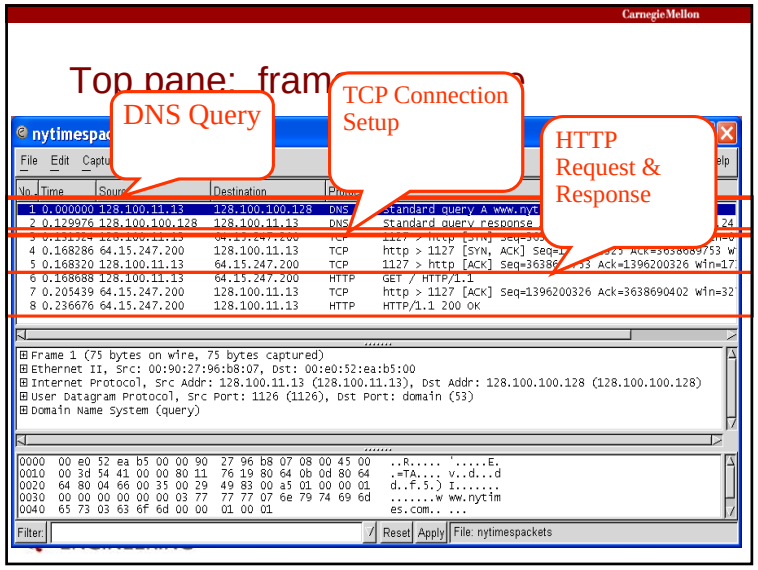
How the layers work together: Network Analyzer Example

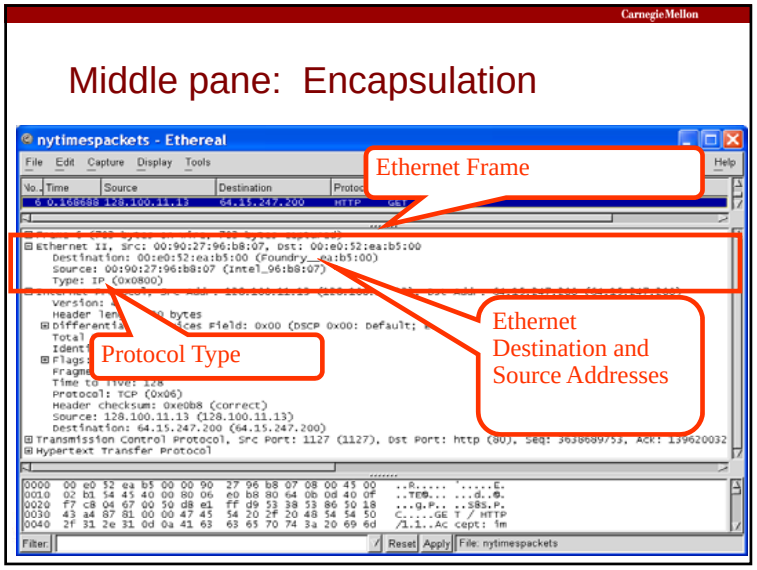


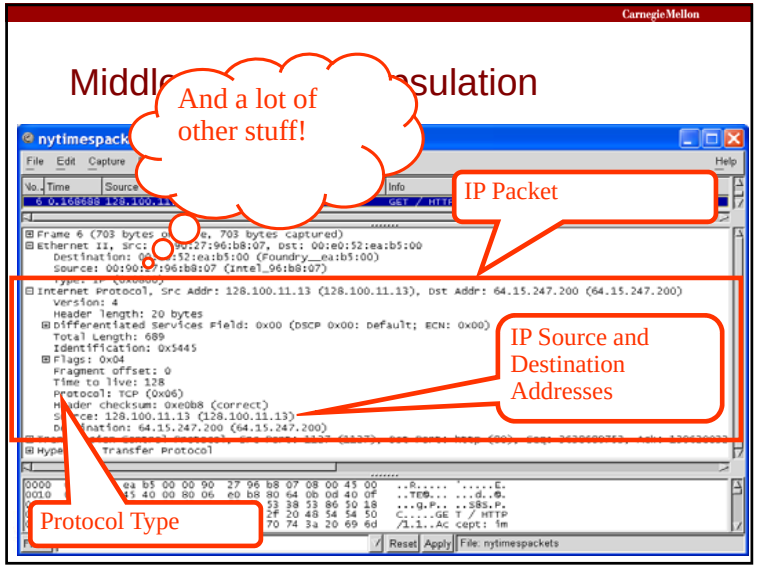
- User clicks on http://www.nytimes.com/
- *Ethereal* network analyzer captures all frames observed by its Ethernet NIC (or Wireshark)
- Sequence of frames and contents of frame can be examined in detail down to individual bytes

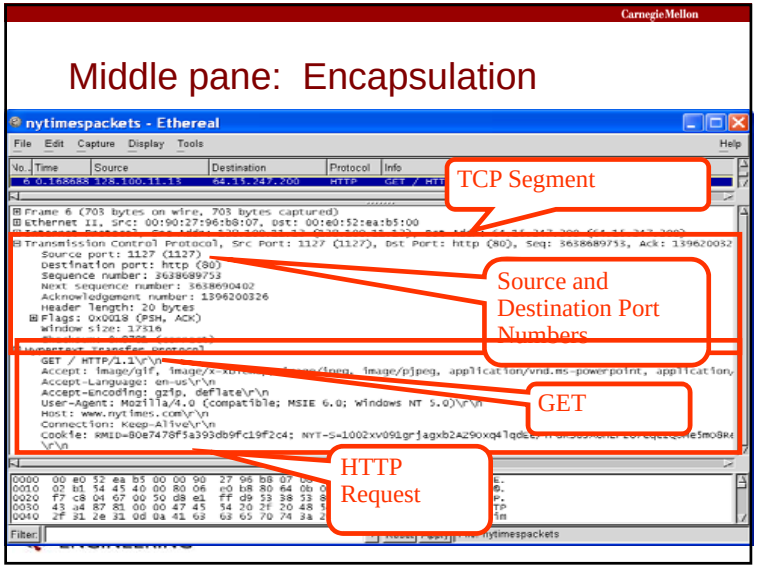
Electrical & Computer ENGINEERING




Top pane: fram
DNS Query
TCP Connection Setup
HTTP Request & Response


Middle pane: Encapsulation
Ethernet Frame
Protocol Type
Ethernet Destination and Source Addresses


And a lot of other stuff!
IP Packet
IP Source and Destination Addresses
Protocol Type


Middle pane: Encapsulation
TCP Segment
Source and Destination Port Numbers
GET
HTTP Request