18-345 – Fall 08

Lecture 2
Layered Network Architectures

Peter Steenkiste

reading: Chapter 2

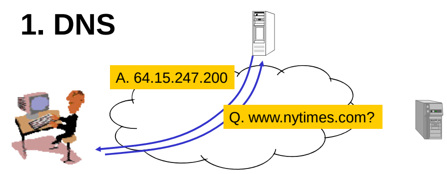
Outline

- An example
- Protocols
- OSI layered network model
- Overview of layers
- Making layers work
- Example revisited

Web Browsing Application

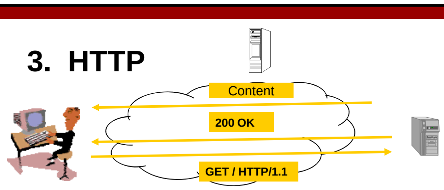
- World Wide Web allows users to access resources (i.e. documents) located in computers connected to the Internet
- Documents are prepared using HyperText Markup Language (HTML)
- A browser application program is used to access the web
- The browser displays HTML documents that include *links* to other documents
- Each link references a *Uniform Resource Locator* (URL) that gives the name of the machine and the location of the given document
- Let's see what happens when a user clicks on a link

1. DNS



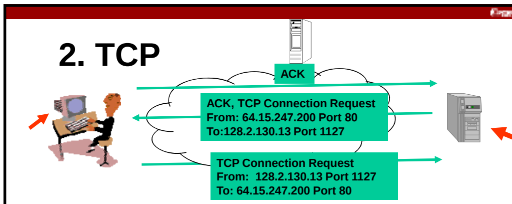
- To translate Internet name to Internet (IP) address
- User clicks on http://www.nytimes.com/
- URL contains Internet name of machine (www.nytimes.com), but not Internet address
- Internet needs Internet address to send information to a machine
- Browser software uses Domain Name System (DNS) protocol to send query to a DNS server for Internet address
- DNS system responds with Internet address

3. HTTP



- To obtain resource (HTML document) from server
- Browser software uses HyperText Transfer Protocol (HTTP) to send request for document
- HTTP client sends its request message to server: "GET …"
- HTTP server sends a status response to client: "200 OK"
- HTTP server sends requested file
- Browser displays document

2. TCP



- But how do client and server communicate?
- HTTP server waits for requests by listening to a well-known port number (80 for HTTP)
- HTTP client sends request messages through an "ephemeral port number," e.g. 1127
- Client and server exchange messages by first opening a Transmission Control Protocol (TCP) connection between them
- TCP is a connection-oriented protocol
- TCP allows reliable transfer messages

Protocols?

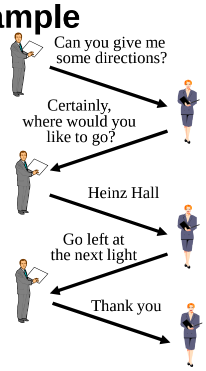
- Web browsing requires the operation of several protocols
  - DNS, HTTP, TCP
  - Many lower level protocols have been left out of the example
- So, what is a protocol?

Protocols

- A *protocol* is a set of rules that governs how two or more communicating entities are to interact
- *Messages* that can be sent and received
- *Actions* that are to be taken when a certain event occurs, e.g. sending or receiving messages, expiring of timers

Protocol Example

- Protocol is not a network-specific term!
- Protocols may have to define many aspects of the communication.
  - Data encoding, language, error recovery, termination conditions, ..
- Network protocols can exist between computer programs or hardware components.



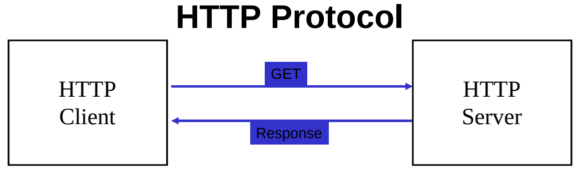
Protocols Are Important

- Protocols are the key to interoperability.
  - The hardware/software of communicating parties are often not built by the same vendor
    - Sun workstation with PC, 3COM with Cisco bridge, ..
  - They can communicate by using the same protocol
- Protocols exist at many levels.
  - Application level protocols, e.g. access to mail, distribution of bboards, web access, ..
  - Protocols at the hardware level allow two boxes to communicate over a link, e.g. the Ethernet protocol
  - Intermediate protocols can "add value" to a lower-level protocol, e.g. provide a reliable communication service over an unreliable link

Example: HTTP and TCP

- HTTP is an application layer protocol
- Retrieves documents on behalf of a browser application program
- HTTP specifies fields in request messages and response messages
  - Request types (GET); Response codes (200 OK)
  - Content type (HTML), options, cookies, …
- HTTP specifies actions to be taken upon receipt of certain messages (e.g. server response to GET)

HTTP Protocol



- HTTP assumes messages can be exchanged directly between HTTP client and HTTP server
  - How? Using an underlying layer (TCP)
- In fact, HTTP client and server are processes running in two different machines across the Internet
- HTTP uses the reliable stream transfer service provided by TCP

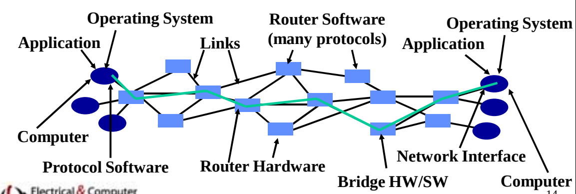
Example: TCP

- TCP is a transport layer protocol
- Provides *reliable byte stream service* between two processes in two computers across the Internet
- Sequence numbers keep track of the bytes that have been transmitted and received (*byte stream*)
- Error detection and retransmission used to recover from transmission errors and losses (*reliable*)
- TCP is *connection-oriented*: the sender and receiver must first establish an association and set initial sequence numbers before data is transferred
- A TCP connection is specified uniquely by

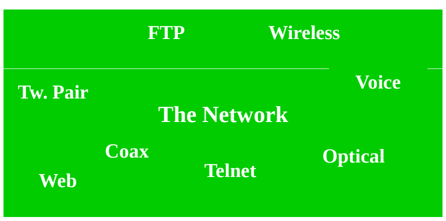
(send port #, send IP address, receive port #, receiver IP address)

How to Design a Network?

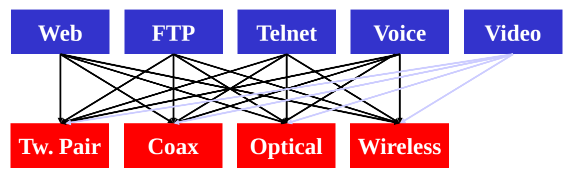
- Has many users
- Offers diverse services
- Mixes very diverse technologies
- Components built by many companies
- Diverse ownership
- Can evolve over time



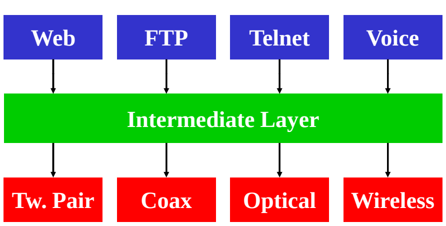
Solution #1



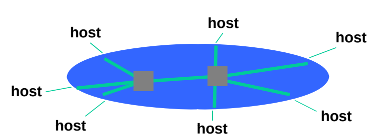
Solution #2?



Solution #3

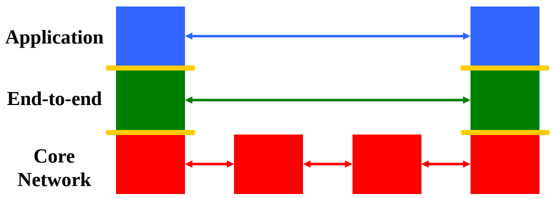


Types of Protocols

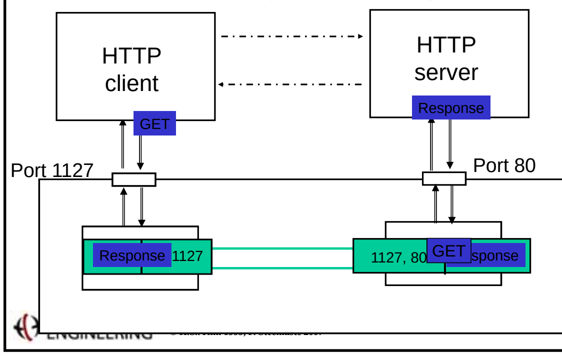


- Core network: responsible for transferring data between a sending and receiving host.
- End-to-end protocols: present a network service to applications and users.
  - May add value to the core network protocols

Protocol and Service Levels

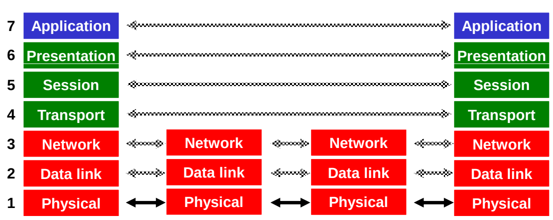


HTTP (higher layer) uses service of TCP (lower layer)



A Layer Network Model

The Open Systems Interconnection (OSI) Model.



OSI Motivation

- Standard way of breaking up a system in a set of components.
  - Traditional modularity argument
  - Components can be implemented and modified in isolation
- Modules are organized as a set of layers.
- Inter-module communication is restricted.
  - Only horizontal and vertical communication
- Clearly this is not the only way of building a network!

Inter-Module Communication

- "Peer" layers on different systems communicate via a protocol.
  - TCP modules communication using the TCP protocol
- Each layer offers a service to the higher layer, using the services of the lower layer.
  - E.g. TCP offers a reliable data transport service to HTTP
- Can have a choice of protocols at each layer.
  - E.g. TCP and UDP offer transport services
  - Must have the same or very similar service interface
  - Many higher level protocols can run over many lower level protocols with "order N" implementation effort

OSI Functions

- (1) Physical: transmission of a bit stream.
- (2) Data link: flow control, framing, error detection.
- (3) Network: switching and routing.
- (4) Transport: reliable end to end delivery.
- (5) Session: managing logical connections.
- (6) Presentation: data transformations.
- (7) Application: specific uses, e.g. mail, file transfer, telnet, network management.

Multiplexing takes place in multiple layers

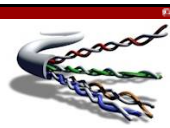
Advantages of Layering?

- Layering simplifies design, implementation, and testing by partitioning overall communications process into parts (modularity)
  - Similar to writing a software program using function calls
- Protocol in each layer can be designed separately from those in other layers
- Protocol makes "calls" for services from layer below
- Layering provides flexibility for modifying and evolving protocols and services without having to change layers below
- Monolithic non-layered architectures are costly, inflexible, and soon obsolete

Open Systems Interconnection History

- By the 1970s every computer vendor had developed its own proprietary layered network architecture
- Problem: computers from different vendors could not be networked together
- Open Systems Interconnection (OSI) was an effort by the International Organization for Standardization (ISO) to enable multivendor computer interconnection
  - Defined a seven-layer abstract reference model for a network architecture - a framework for the development of protocols
- Detailed OSI standards were developed for each layer, but TCP/IP protocols preempted deployment of OSI protocols
  - OSI protocols are not used, but layered framework survived

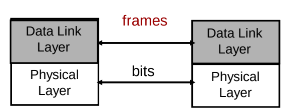
Physical Layer



- Transfers bits across link
- Definition & specification of the physical aspects of a communications link
  - Mechanical: cable, plugs, pins...
  - Electrical/optical: modulation, signal strength, voltage levels, bit times, …
  - functional/procedural: how to activate, maintain, and deactivate physical links…
- Lots of different physical layers exist:
  - Ethernet, DSL, cable modem, telephone modems…
  - Twisted-pair cable, coaxial cable optical fiber, radio, infrared, …

Data Link Layer

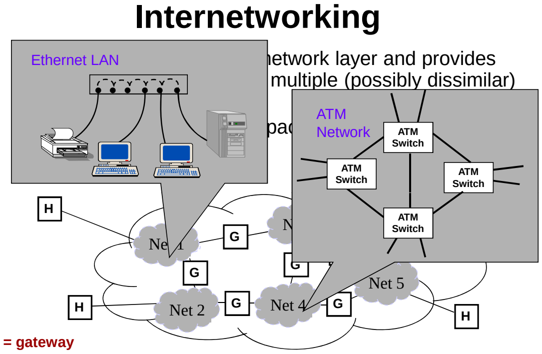
- Transfers *frames* across *direct* connections
- Groups bits into frames
- Detection of bit errors; Retransmission of frames
- Activation, maintenance, & deactivation of data link connections
- Medium access control for local area networks
- Flow control



Network Layer

- Transfers *packets* across multiple links and/or multiple networks
- Addressing (i.e. assignment of addresses to different nodes) must scale to large networks
- Nodes *jointly* execute *routing algorithm* to determine paths across the network
- Forwarding transfers packets across a node
- Congestion control to deal with traffic surges
- Connection setup, maintenance, and teardown when connection-based

Internetworking

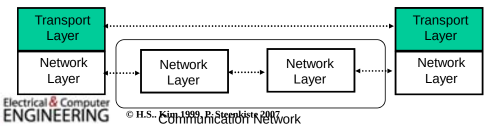


G = gateway

H = host

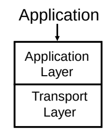
Transport Layer

- Transfers data end-to-end from process in a machine to process in another machine
- Reliable stream transfer (e.g. TCP) or quick-and-simple single-block transfer (e.g. UDP)
- Port numbers enable multiplexing
- Message segmentation and reassembly
- Connection setup, maintenance, and release



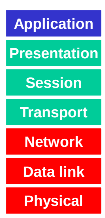
Application & Upper Layers

- 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, …Incorporated into Application Layer

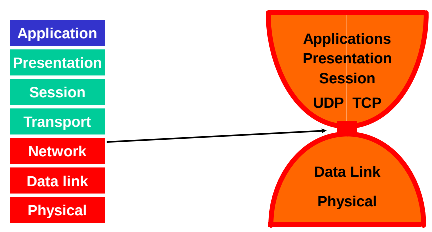


Different Sources of Components

- 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



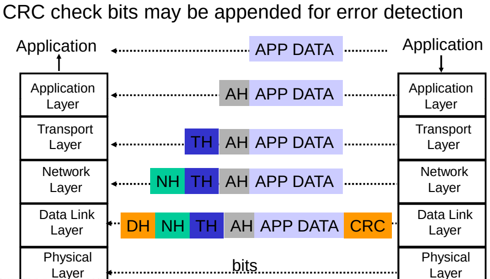
The Internet Protocol Suite



The Hourglass Model

Headers & Trailers

- Each protocol uses a header that carries addresses, sequence numbers, flag bits, length indicators, etc…
- CRC check bits may be appended for error detection

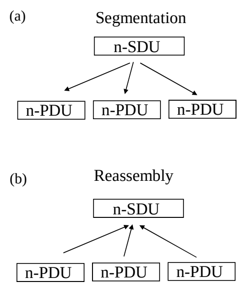


Connection-Oriented and Connectionless Services

- Connection-Oriented
  - Three-phases:
    1. Connection setup between two SAPs to initialize state information
    2. SDU transfer
    3. Connection release
  - E.g. TCP, ATM
- Connectionless
  - Immediate SDU transfer
  - No connection setup
  - E.g. UDP, IP
- Layered services need not be of same type
  - TCP operates over IP
  - IP operates over ATM

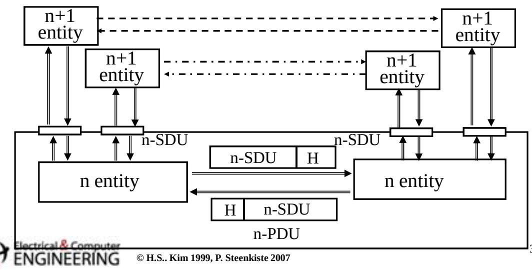
Segmentation & Reassembly

- A layer may impose a limit on the size of a data block that it can transfer due to implementation or other reasons
  - E.g. wireless link is error-prone
- Thus a layer-n SDU may be too large to be handled as a single unit by layer-(n-1)
- Sender side: SDU is segmented into multiple PDUs
- Receiver side: SDU is reassembled from sequence of PDUs

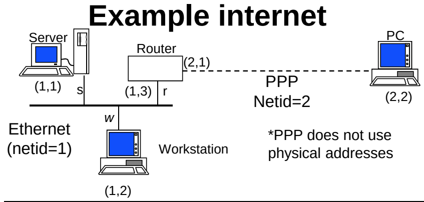


Multiplexing

- Sharing of layer n service by *multiple* layer n+1 users
- Multiplexing tag or ID required in each PDU to determine which user an SDU belongs to
  - E.g. Port number in TCP/IP

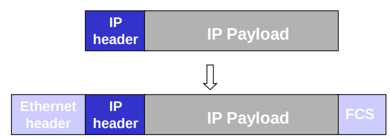


Example internet



	netid	hostid	Physical address
server	1	1	s
workstation	1	2	w
router	1	3	r
router	2	1	-
PC	2	2	-

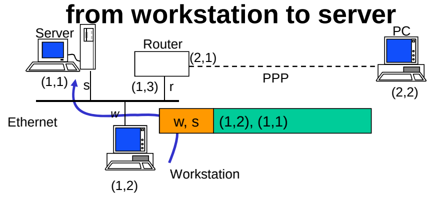
Encapsulation by Internet and Ethernet



FCS = frame check sequence

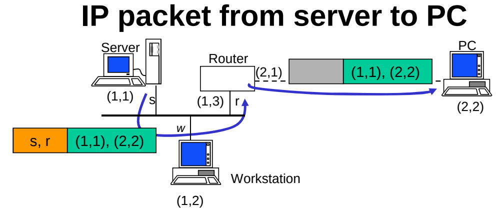
- Ethernet header contains:
  - source and destination physical addresses
  - network protocol type (e.g. IP)

How the layers work together: IP packet from workstation to server



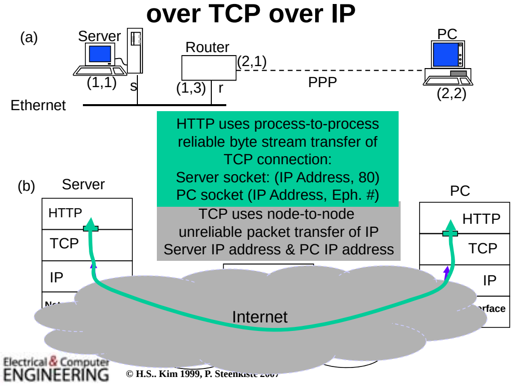
1. IP packet has (1,2) IP address for source and (1,1) IP address for destination
2. IP table at workstation indicates (1,1) connected to same network, so IP packet is encapsulated in Ethernet frame with addresses w and s
3. Ethernet frame is *broadcast* by workstation NIC into the ethernet.
4. This frame is captured by server NIC since ethernet NIC is always listening for frames
5. NIC examines protocol type field and then delivers packet to its IP layer

IP packet from server to PC

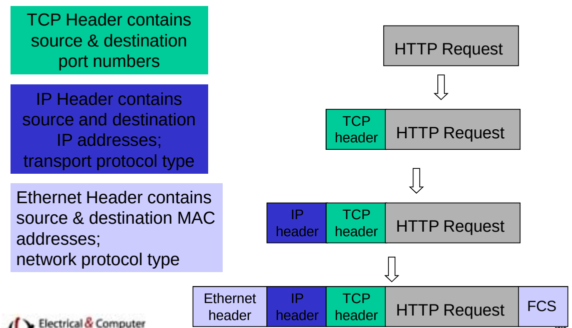


1. IP packet has (1,1) and (2,2) as IP source and destination addresses
2. IP table at server indicates packet should be sent to router, so IP packet is encapsulated in Ethernet frame with addresses s and r
3. Ethernet frame is broadcast by server NIC and captured by router NIC
4. Router NIC examines protocol type field and then delivers packet to its IP layer
5. IP layer of router examines IP packet destination address and determines IP packet should be routed to (2,2)
6. Router's table indicates (2,2) is directly connected via PPP link
7. IP packet is encapsulated in PPP frame and delivered to PC
8. PPP at PC examines protocol type field and delivers packet to PC IP layer

How the layers work together: HTTP over TCP over IP



Encapsulation



Summary: OSI reference model

- Layering simplifies overall design process and provides flexibility to evolve architecture
- Each layer provides service to layer above
- Protocol defines messages and actions within a layer required to provide a service
- OSI Reference Model provides framework for discussion of protocols
- Layers: Application, Transport, Network, Data Link, Physical