SC/CSE 3213 Winter 2014

L6: Protocols, Services, Layers

Sebastian Magierowski
York University

Outline

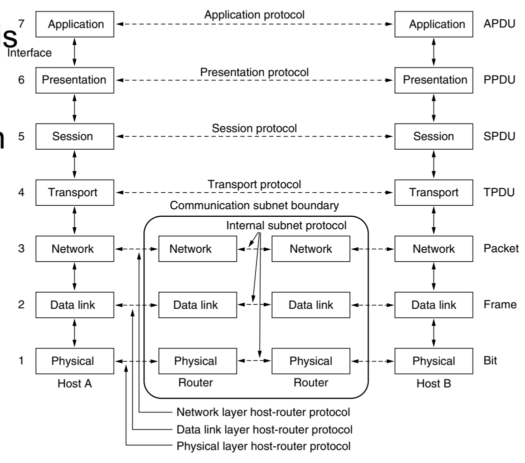
- Network Layering Terminology
  - protocols, services, peers, clients, etc.
- Network Protocol Examples
  - HTTP, TCP, DNS, UDP

Protocols

- For remote entities to establish working communications a set of rules needs to be in place
- Protocols are just these rules
  - HTTP
  - TCP
  - DNS
  - UDP
  - BGP
  - OSPF
  - etc., etc.,…
- In this lecture we look at some of these rules at work

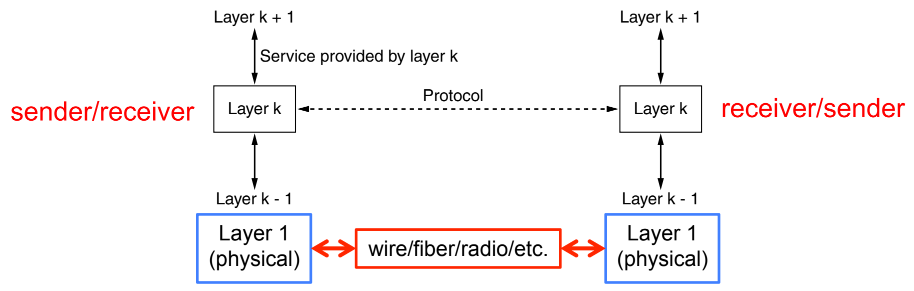
The Layered Partition

- Communication between internetworked machines is *complex*
- Partition this process in an intelligent way
- Layering partitions related communications functions into groups that are manageable
- A preview ⇒



Terminology: Layers, Services, Protocols

- Each layer provides a service to the layer above



- Each layer operates according to a protocol (e.g. HTTP)
- Entities comprising the same layer on different machines are called peers

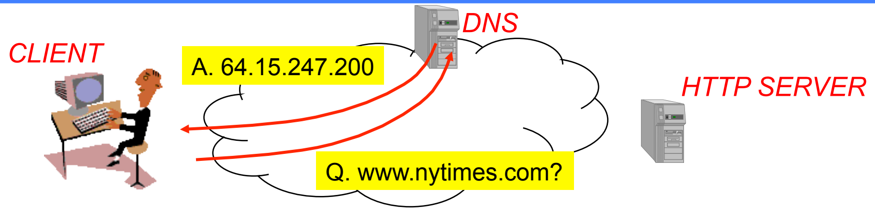
Clients and Servers

- Client-server model of network usage
  - servers (daemon) provide services (e.g. web pages, query answers, video streaming, etc.)
  - clients request services
  - protocols run on both to conduct transfer of information among running programs
- Consider a web browsing example…

Web Browsing Application

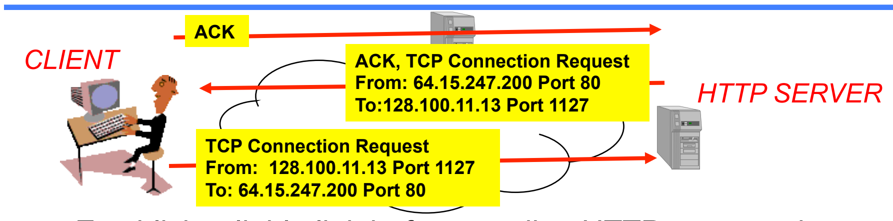
- WWW allows users to access resources (i.e. documents) located in computers connected to the Internet
- Documents are prepared using HyperText Markup Language (HTML)
- A browser application program is used to access the web
- The browser displays HTML documents that include links to other documents
- 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 a link

1. DNS



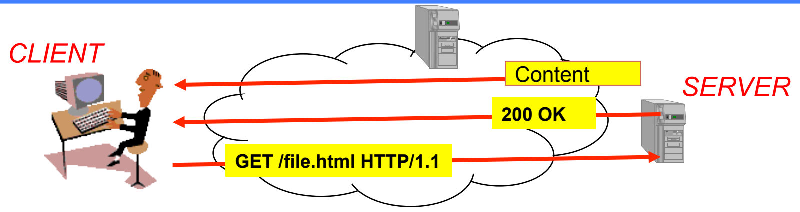
- User clicks on http://www.nytimes.com/
- URL contains Internet name of machine (www.nytimes.com), but no Internet address
- Internet needs Internet address to send information to a machine
- Browser software uses Domain Name System (DNS) protocol to send query for Internet address
- DNS system responds with Internet address

2. TCP


ACK
CLIENT
ACK, TCP Connection Request
From: 64.15.247.200 Port 80
To:128.100.11.13 Port 1127
HTTP SERVER
TCP Connection Request
From: 128.100.11.13 Port 1127
To: 64.15.247.200 Port 80

- Establish reliable link before sending HTTP commands
- HTTP runs on top of TCP to transfer messages reliably
- Browser invokes the TCP entity by…
  - Specify a TCP source ID defined by:
    - Source internet address & port (e.g.: 128.100.11.13 and 1127)
  - Specify a TCP destination ID defined by:
    - Destination internet address & port (e.g.: 64.15.247.200 and 80)
- A 3-step exchange follows establishing a TCP connection

3. HTTP

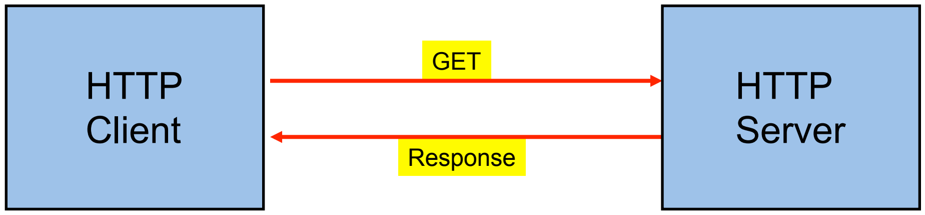


- HTTP client sends its request message: “GET …”
- HTTP server sends a status response: “200 OK”
- HTTP server sends requested file
- Browser displays document
- Clicking a link sets off a chain of events across the Internet!
- Let’s see how protocols & layers come into play…

Example: HTTP

- HTTP is an application layer ASCII protocol
- Retrieves documents for a browser program through a series of request-response messages
- *Requests* specify actions (methods) for server
  - 6 methods, e.g. GET, POST, etc.
- *Responses* return status info & requested information
  - e.g. 404 page not found
- Request/Response headers give detailed info on users and information content
  - Browser/platform information, freshness, MIME type, etc.
  - Cookies!

HTTP Connections



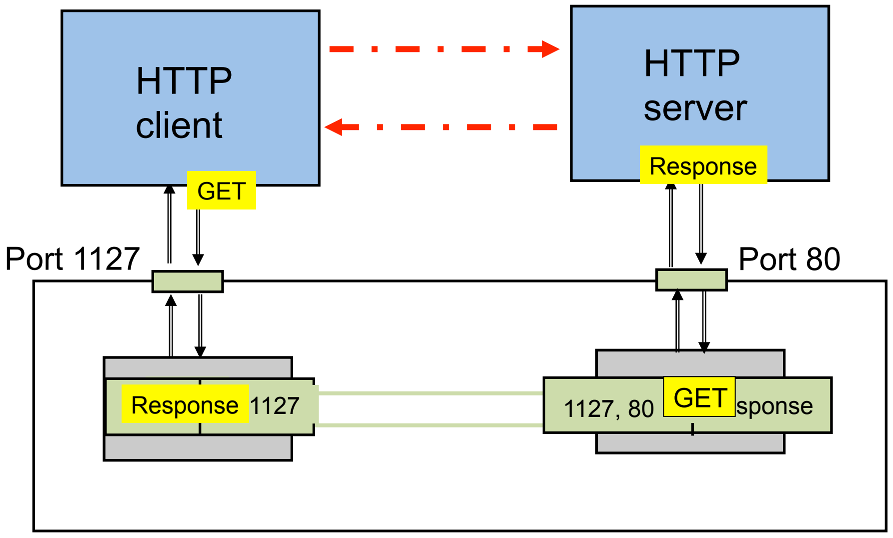
- HTTP assumes messages can be exchanged directly between HTTP client and HTTP server
- In fact, HTTP client and server are processes running in two different machines across the Internet
- HTTP uses the reliable stream transfer service provided by TCP

Example: TCP

- TCP is a transport layer protocol
- Provides *reliable byte stream* *service*
- Sequence numbers keep track of exchanged bytes
- Error detection and retransmission used to recover from transmission errors and losses
- TCP is connection-oriented: sender and receiver must first establish an association and set initial sequence numbers before data is transferred
- Connection ID is specified uniquely by *(socket1, socket2)*

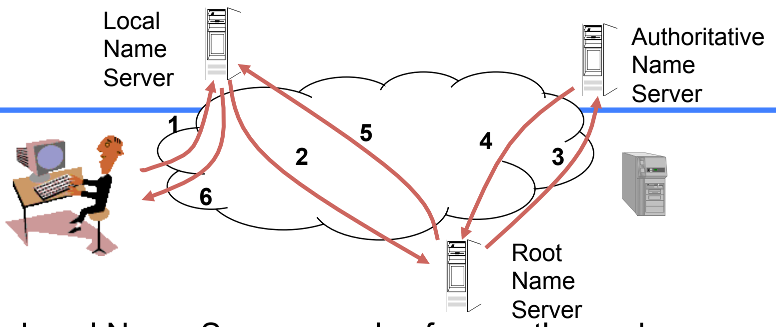
(send port #, send IP address, receive port #, receiver IP address)

HTTP Uses Service of TCP



Example: DNS Protocol

- DNS protocol is an application layer protocol
- DNS is a distributed database that resides in multiple machines in the Internet
- DNS protocol allows queries of different types
  - Name-to-address or Address-to-name (URL⇔ IP)
  - Mail exchange
- DNS usually involves short messages and so uses service provided by UDP
- Well-known port 53



- Local Name Server: resolve frequently-used names
  - University department, ISP
  - Contacts Root Name server if it cannot resolve query
- Root Name Servers: 13 globally
  - Resolves query or refers query to Authoritative Name Server
- Authoritative Name Server: last resort
  - Every machine must register its address with at least two authoritative name servers

Example: UDP

- UDP is a transport layer protocol
- Provides datagram service between two processes in two computers across the Internet
- Datagram is sent without bothering to first establish a connection
- UDP is connectionless
- Quick, simple, but not reliable