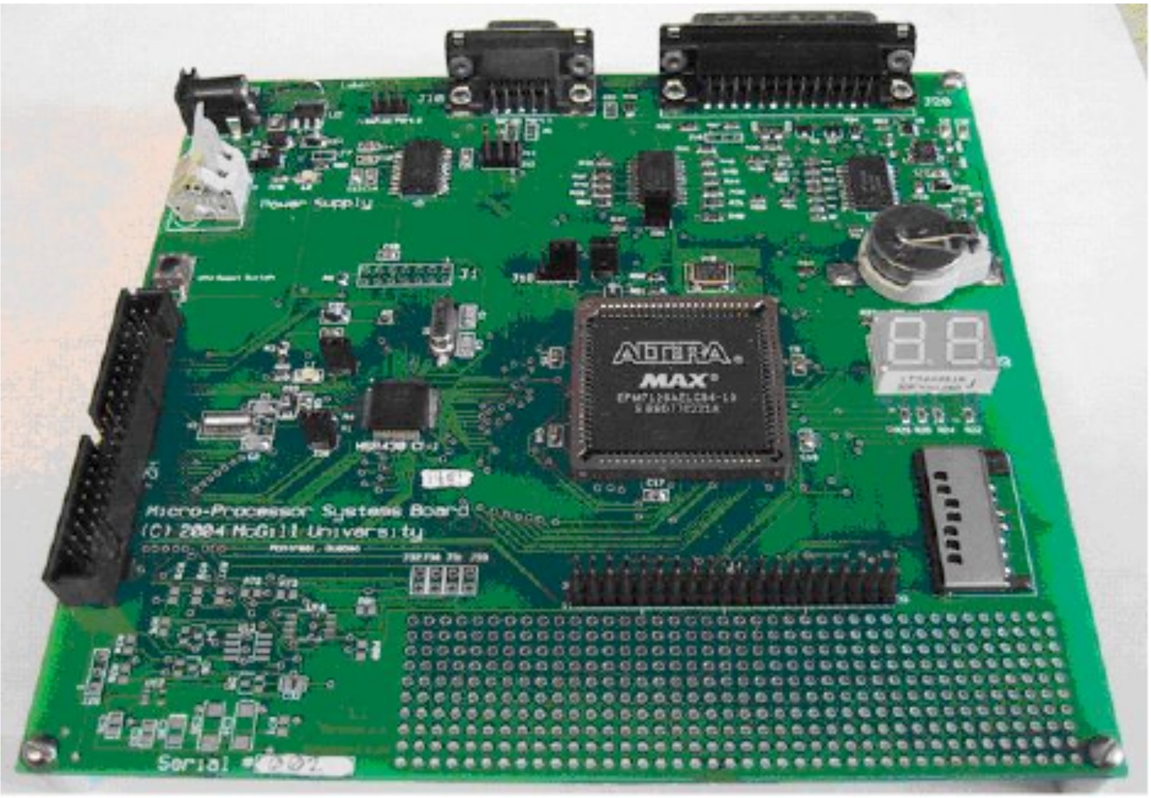
ECSE 444
Embedded Networking

Zeljko Zilic

zeljko.zilic@mcgill.ca



Acknowledgments: to STMicroelectronics for material on processors and the board

Outline

- Computer Networking
- Client-server Computing
- Networking Protocols
- (Embedded) Web Serving
- Project Proposal – 1 page
- Lab 4: Sensors+ OS
- Quiz 3 in week from now (Mar. 29th)

Computer Network

- Def.: Computer network is a group of computers and devices connected to each other
- Computer networks can be classified according to hardware and software technology that is used to interconnect individual devices in the network (optical fiber, Ethernet card, Wireless LAN, etc.)
  - Ethernet uses physical wiring to connect devices such as hubs, switches, bridges and routers
  - Wireless LAN is designed to connect devices without wiring using only radio waves or infrared signals as transmission medium

Embedded Systems Networking

- Significant boost to device flexibility
  - Quickly adds to the range of possibilities
- Change function after deployment
- Add capabilities that are not even possible to imagine at the time of the device design
- Add worldwide access to the device
- Examples: routers configured via web, Internet fridge, control and automation devices

Client-Server Model Description

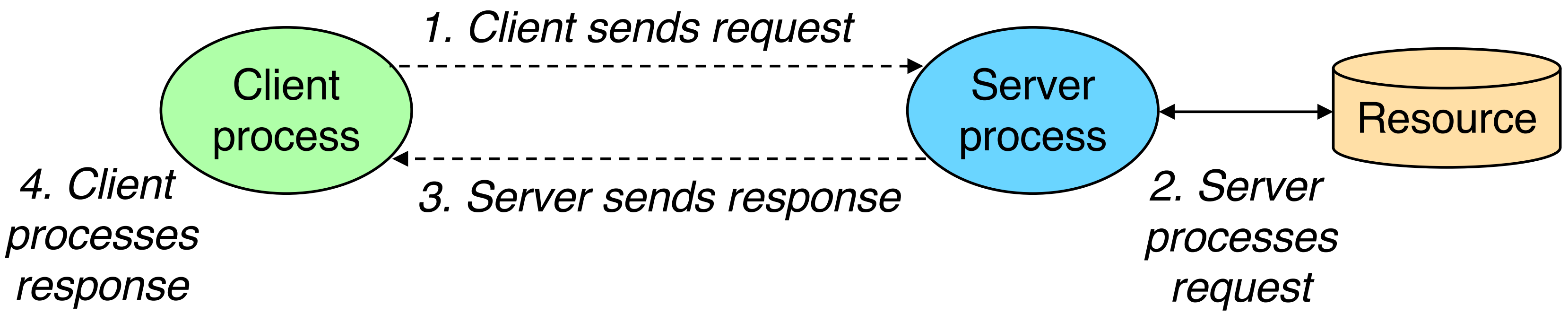
- Client-server model distinguished client system from server system, which communicate over a computer network
  - The client program makes a service request to another, the server program
  - Client-server model found its big application in network computing
    - Internet main application protocols such as HTTP, Telnet, DSN
    - Standard network functions such as e-mail exchange, web access and database access are based on this model
      - Example: web browser is a client program at the user computer that may access information at any web server in the world

Client-Server Architecture

- Each instance of the client software can send data to request to one or more servers
  - Servers can accept these requests, process then and return the requested information to the clients
- Server manages resources, provides services to clients in terms of resources
- The most basic type of client-server architecture (two-tire) employs only two types of hosts: clients and servers
  - Client acts as one tier and application in combination with server acts as another tier
- Clients and servers are processes and not machines or hosts
  - A single host can run many different clients and servers concurrently, and the client and host translations can be on the same or different hosts
  - The client-server model is the same regardless of the mapping to the actual host

Client-Server Transaction Steps

- Basic operation in client-server model is transaction consisting of four steps:
  - Step 1: When client needs server it sends a request to the server
  - Step 2: Upon receiving a request, the server interprets it, and manipulates its resources
  - Step 3: The server sends response to the client and waits for the next request
  - Step 4: The client receives the response and manipulates it



Client-Server Advantages

- Allows on a computing system to be distributed among several independent computers that are known to each other only through a network
  - It is possible to replace, repair, upgrade or relocate a server while its clients remain both unaware and affected by that change
  - All data is stored on the servers, which have much greater security controls than most clients
    - Servers can better control access and resources to guarantee that only those clients with the appropriate permission may access and change data

Client-Server Disadvantages

- Traffic congestion is a huge issue
    - As the number of simultaneous client requests to a given server increases, the server can become severely overloaded
- If a critical server fails then client requests cannot be fulfilled

Network Host

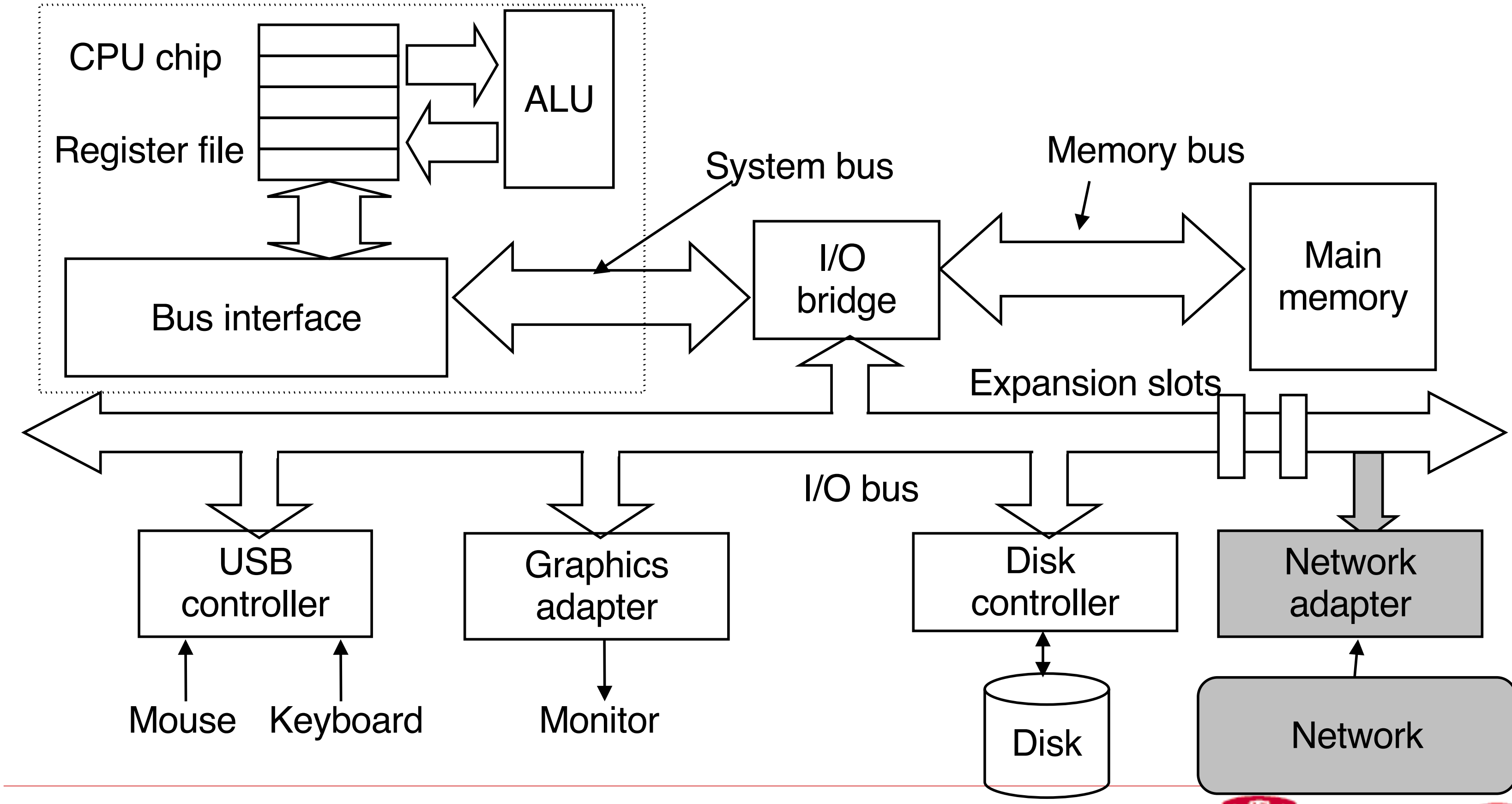
- Network host is a computer connected to the Internet
- It can host information as well as client and server software
- Every Internet host has a unique IP address including a host address part
- Every host is a network node (network device) but every node is not a host
  - Network nodes such as modems and network switches are not assigned host addresses, as they are not hosts
  - Devices such as network printers and hardware routers are assigned IP host addresses, but since they are not general-purpose computers, they are often not considered as hosts

Network Host Structure

- Clients and servers often use different hosts and hence need to communicate using hardware and software of computer network
- Host views a network as a I/O device that serves as a source and sink for data
- Adapter connected to the I/O bus provides the physical interface to the network
  - Data received from the network is copied from the adapter into memory
  - Similarly data can be copied from the memory to the network

Network Host Device

- Usually, looks and feels like a PC



Internet Hosting Service

- Internet hosting service is a service that runs Internet servers, allowing organizations and individuals to serve content to the Internet
  - A common kind of hosting is web hosting
- Types of hosting
  - Full-featured hosting includes
    - Dedicated hosting service (managed hosting service) where the hosting service provider owns and manages the machine, leasing full control to the client
    - Management of the service can include monitoring to ensure the server continues to work effectively, backup services, installation of security patches and various levels of technical support
  - Virtual private server, in which virtualization technology is employed in order to allow multiple logical severs to run on a single physical server
  - Colocation facilities, which provide just the Internet connection, uninterruptible power and climate control, but let the client do his own system administration

Ethernet and Internet

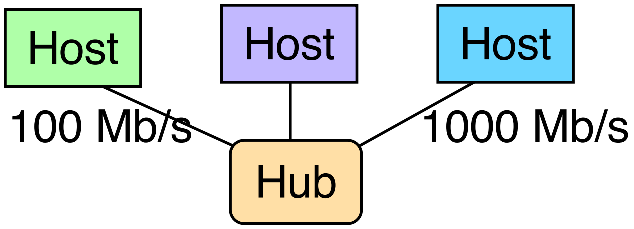
- Ethernet is a family of frame-based computer networking technologies for LANs
  - The name comes from the physical concept of the ether
- Internet is a global network of interconnected computers, enabling users to share information along multiple channels
  - A computer connected to Internet can access information from a vast array of available servers and other computers by moving information from them to the computer’s local memory
  - The same connection allows the computer to send information to servers on the network
  - The movement of information in the Internet is achieved via a system of interconnected computer networks that share data by packet switching using the standardized Internet Protocol Suite (TCP/IP)
    - It is a network of networks that consists of millions of private and public, academic, business and government networks of local and global scope that are linked by copper wires, fiber-optic cables, wireless connections, etc.
  - Communication infrastructure of Internet consists of its hardware and systems of software layers

Network Hubs and Bridges

- Network hub is a device for connecting multiple twisted pair or fiber optic Ethernet devices together and causing them to act as a single network segment
  - Hubs work at the physical layer of OSI model (layer 1)
  - The device is in fact a form of a multiport repeater
    - Repeater hubs also participate in collision detection, forwarding a jam signal to all parts if it detects a collision
  - Hubs can be used to connect segments such that there were up to five Ethernet segments between any two hosts
- A network bridge connects multiple network segments at the data ling layer (layer 2) of the OSI model
  - Only well formed packets are forwarded from one Ethernet segment to another, collisions and packet errors are isolated
  - Bridges are very similar to hubs (repeaters)
    - The difference is that in bridging traffic from one network is managed rather than simply rebroadcasted to adjacent network segments
      - Hubs have a problem with limitations on how many machines could communicate on an Ethernet network
      - Segments joined by hubs had o all operate at the same speed
  - Bridges are generally more complex than hubs due to the fact that they are capable of analyzing incoming data packets on a network to determine if the bridge is able to send the given packet to another segment of that same network

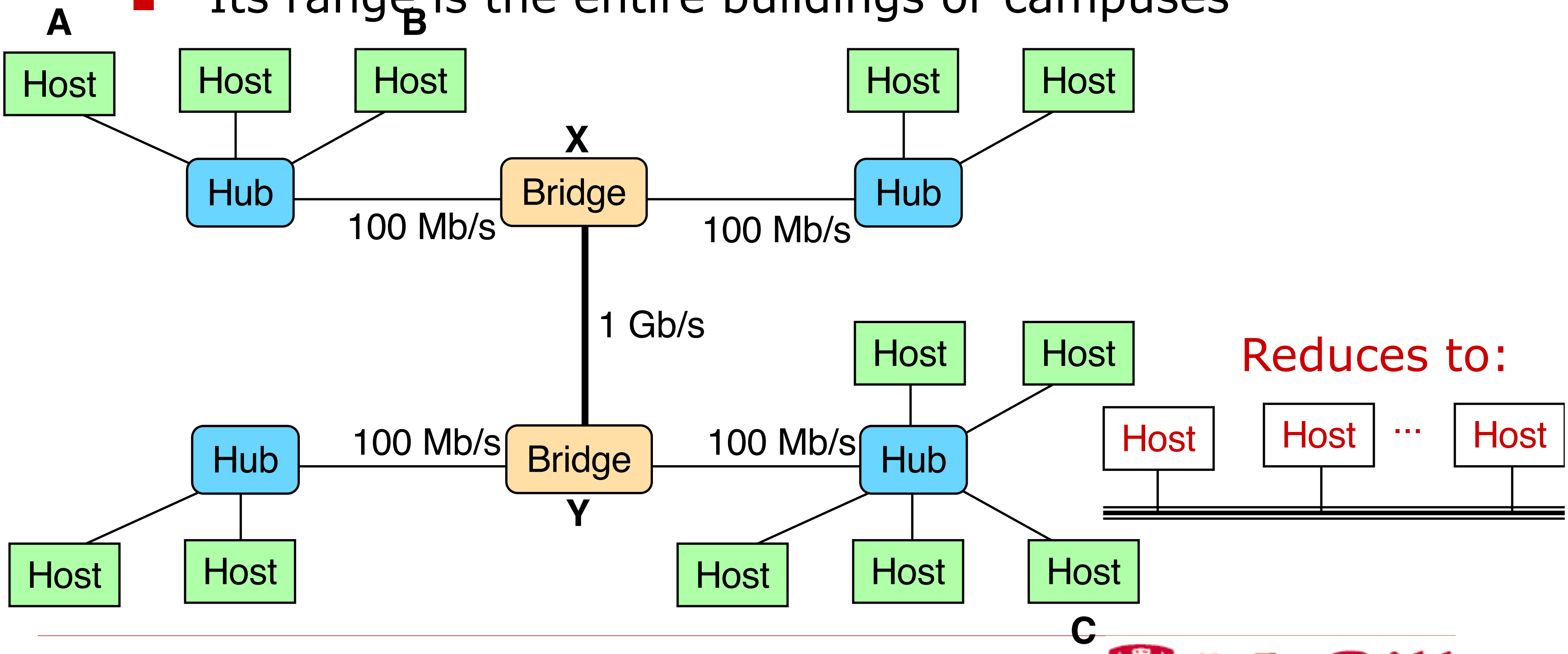
Ethernet Segment

- Each segment consists of twisted pairs of wires and a hub
- Its typical range is a small area such as rooms or floors in the building
- Each wire has the same bandwidth (100 Mb/s or 1 Gb/s)
- One end is attached to an adapter or host and the other is attached to a port or hub



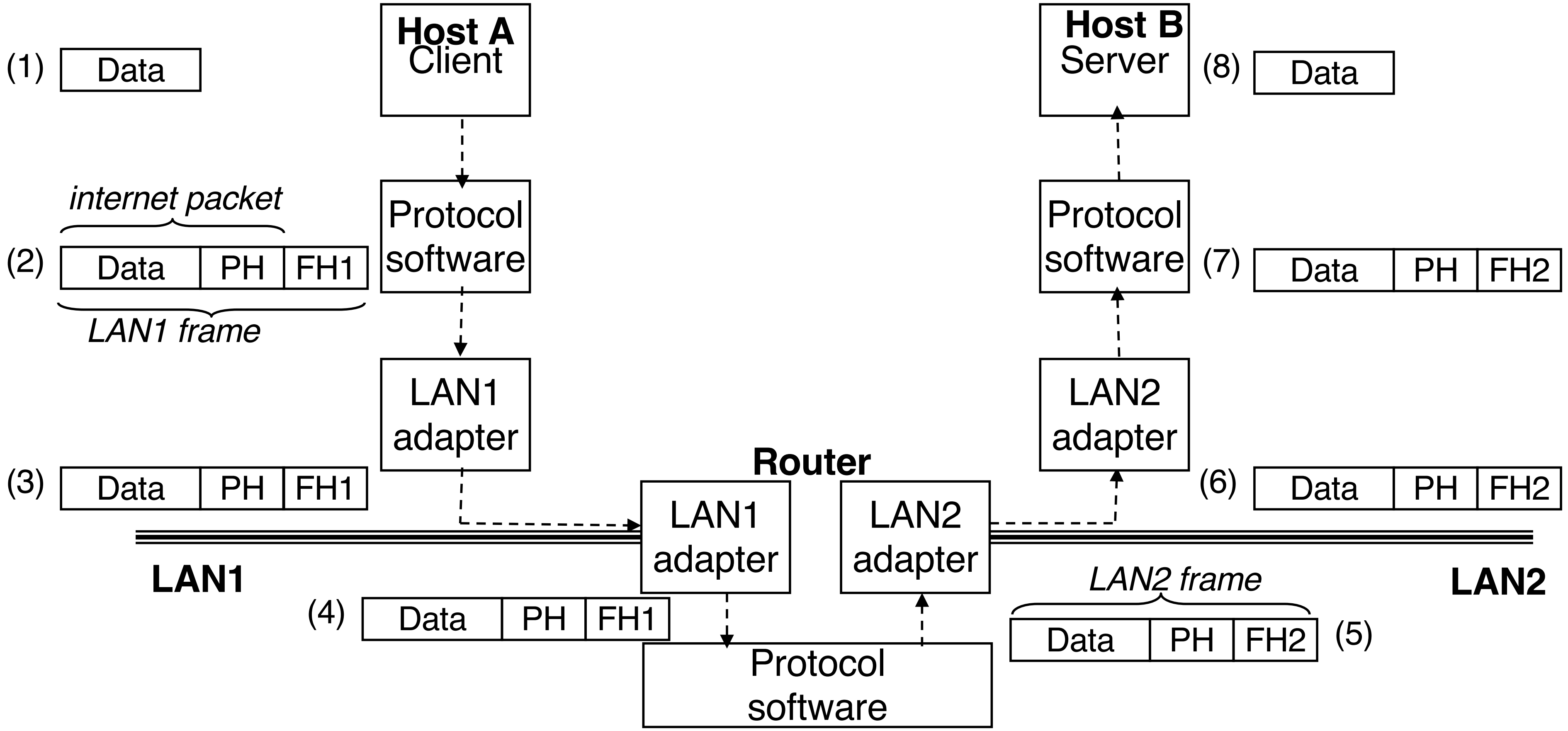
Bridge Ethernet

- Bridge Ethernet consist of multiple Ethernet segments connected into LAN
  - Its range is the entire buildings or campuses



Data Transfers on Internet

- Data encapsulation, naming, protocol role



Data Transfers on Internet, cont.

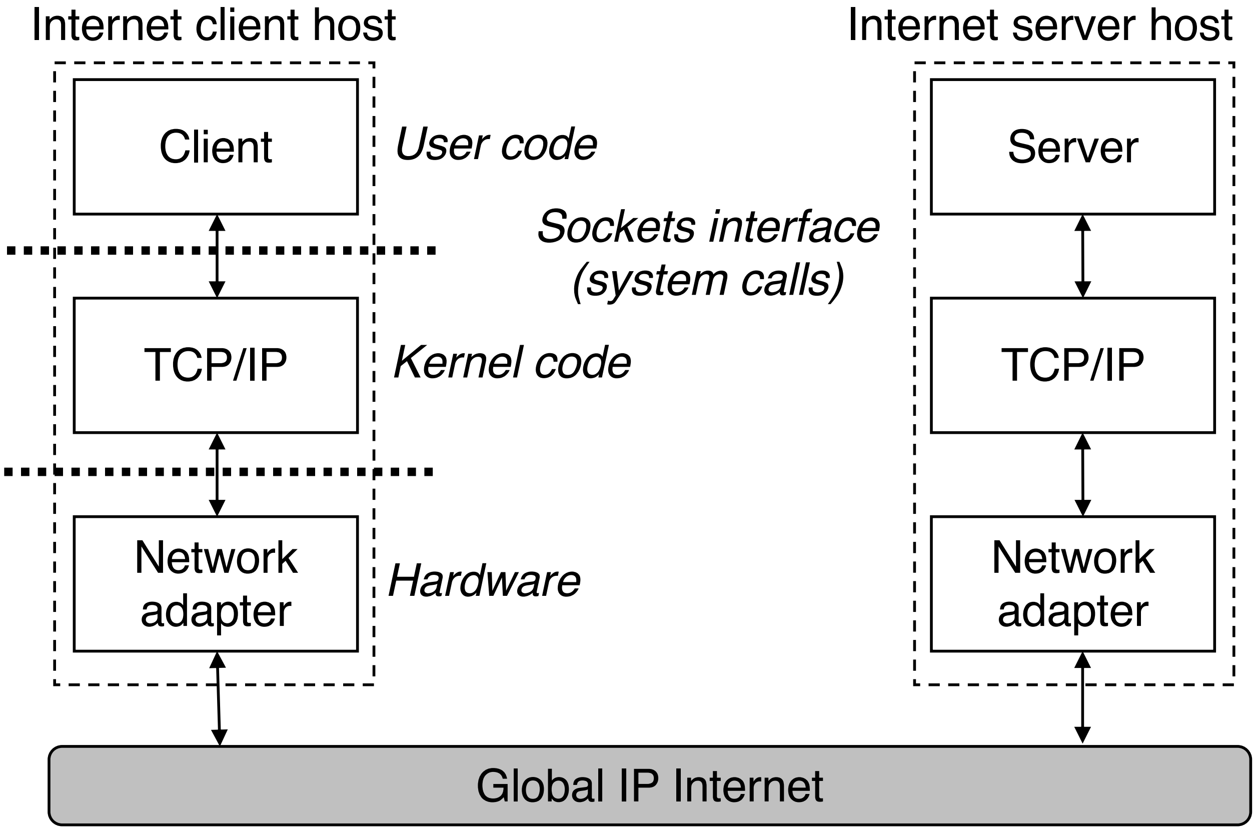
- Consider two LANs connected by a router
  - A client on host A attached to LAN1 sends a sequence of data bytes to the server running on host B attached to LAN2
- Eight steps in data transfer
  - Step 1: Client (host A) initialized a system call that copies data from the client's virtual address space into a kernel buffer
  - Step 2: Protocol software on host A creates a LAN1 frame
    - Internet header and LAN1 frame header are appended to the data
      - Internet header is an address of internet host B
      - LAN1 frame header addresses the router
    - The frame is passed to the adapter
  - Step 3: LAN1 adapter copies the frame to the network
  - Step 4: Upon reaching the router , the router's LAN1 adapter reads it from the wire and passes it to the protocol software
  - Step 5: The router fetches the destination internet address from the internet package header and uses it to determine where to forward the frame (LAN2)
    - The router cancels the old LAN1 frame header and appends the new LAN2 frame header address to host B
  - Step 6: The router's LAN2 adapter copies the frame to the network
  - Step 7: Upon the frame reaching host B, its adapter reads the frams from the wire and passes it to the protocol software
  - Step 8: Protocol software on host B removes the packet header and frame header
    - Remaining frame data is copied into server's virtual address space

IP, TCP and UDP

- Internet Protocol (IP) is a protocol used for communicating data across a packet-switched internetwok using the Internet Protocol Suite (TCP/IP)
  - It is a primary protocol with the task of delivering distinguished protocol packages from the source host to the destination host solely based on their addresses
  - IP can be used over a heterogeneous network, I.e., a networks connecting computers may consist of a combination of Ethernet, ATM, token ring, etc.
    - Each link layer implementation may have its own method of addressing with a corresponding need to resolve IP addresses to data link addresses
- Transmission Control Protocol (TCP) is one of the core protocols of the Internet Protocol Suite. It is so important that the entire suit is generally referred to as TCP/IP
  - IP handles lower-level transmission from computer to computer as a message makes its way across the Internet
- TCP operates at a higher level dealing only with the two end systems
  - For example a Web browser and a Web server
- TCP provides reliable, ordered delivery of a stream of bytes from one program on one computer to another program on another computer
  - Common applications include Web, e-mail and file transfer
- User Datagram Protocol (UDP) belongs to the Internet Protocol Suite (set of network protocols used for the Internet)
  - Using UDP computer applications can send messages (datagrams) to other hosts on an IP network without requiring prior communications to set up special transmission channels or data paths
  - UDP uses simple transmission model without implicit hand-shaking dialogues for guaranteeing reliability, ordering or data integrity
    - UDP provides unreliable services and datagrams may arrive out of order, duplicated of missing without notice
  - UDP is often used in time-sensitive applications as dropping packets is preferable to using delayed packets. Also used by servers that answer small queries from huge number of clients

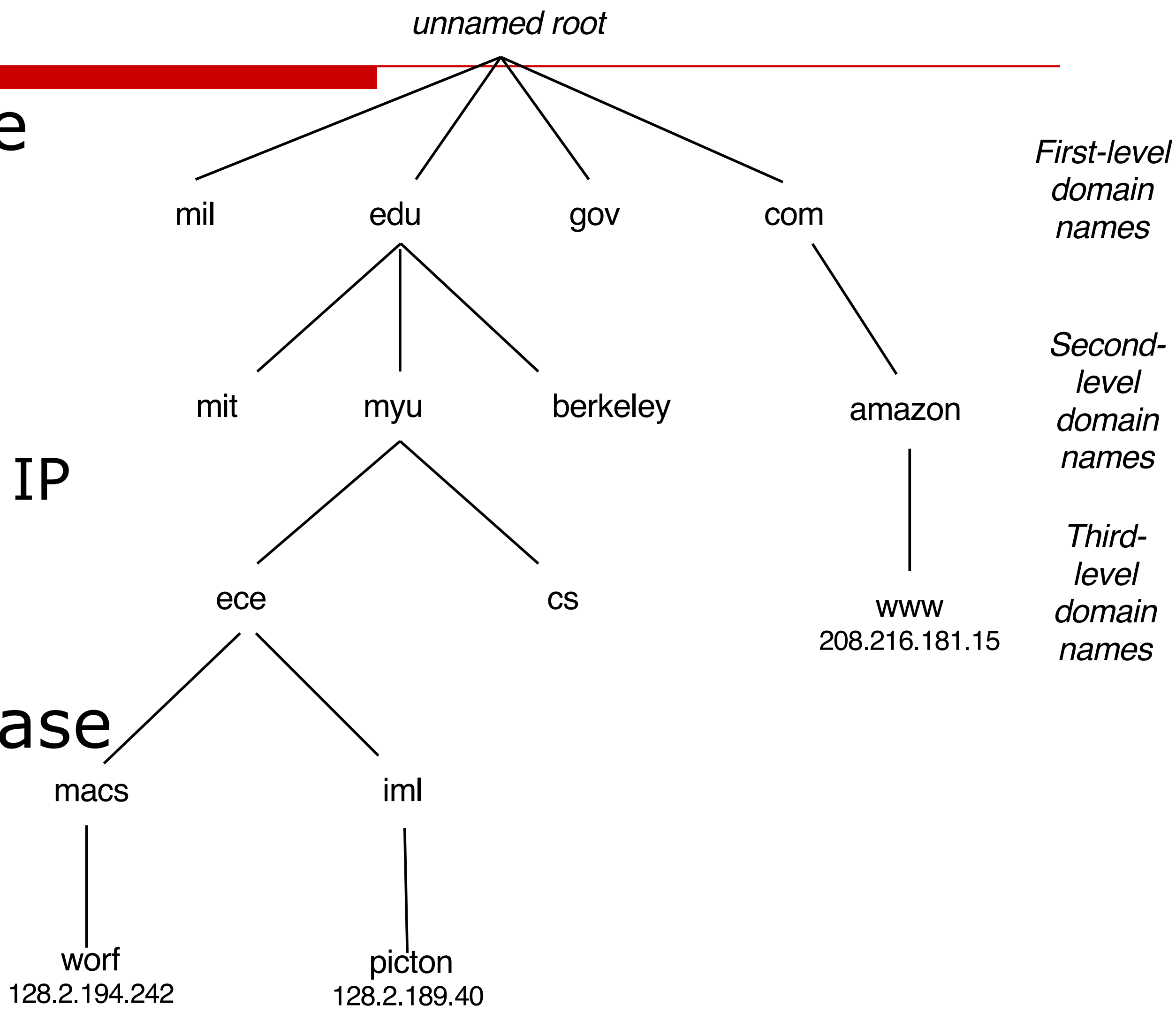
System Organization in Networked Application

- Applications independent of lower-level protocols, HW ..
  - IP: naming, routing
  - TCP: Connection-oriented
  - UDP: connectionless



Addressing for Users: Domains

- Hard to memorize 32-bit IP address
- Domain name server
  - Domain name -> IP address
- World-wide distributed database
  - Network of DNS servers

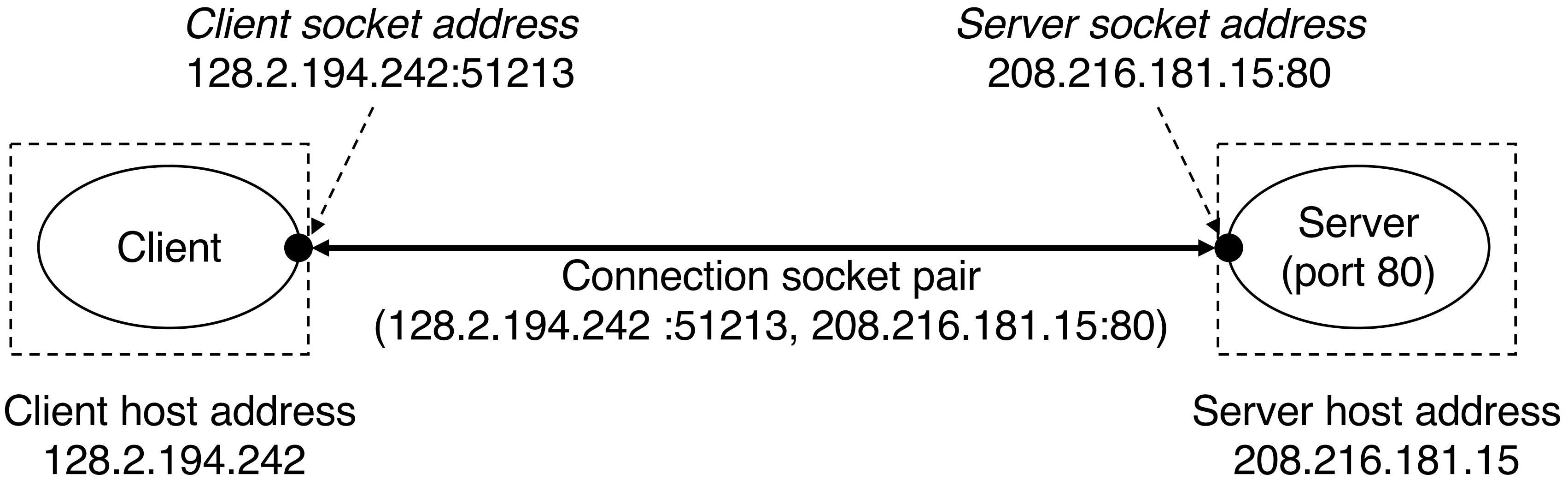


Addressing for Users

- Internet defines a set of human-friendly domain names, which is a sequence of words (letters, numbers and dashes) separated by periods: microp.ece.mcgill.ca
- Set of domain names form a hierarchy
- In the example on the previous slide, the hierarchy is represented as a tree
  - Nodes of the tree represent domain names that are formed by the path back to the root
  - Subtrees are represent subdomains
  - The first level in the hierarchy is an unnamed root node
  - The next level is a collection of first-level domain names used by non-profit organizations
    - First-level domains include: com, edu, gov, org, etc.
  - Second level domain names are such as mcgill.ca

Establishing a Connection

- Reliable, full-duplex connection between two points
- Endpoint: socket
  - 16-bit socket port
- Identified by a pair of IP address:socket port



Internet Connections - Sockets

- Internet clients and servers communicate through sending and receiving streams of bytes over a point-to-point connections
- The connection is full-duplex
  - Data can flow in both direction at the same time
- Socket interface is a set of functions, that together with Unix I/O functions are used to build network applications
  - Developed at Berkeley, and hence often referred to as Berkeley socket
    - Berkeley socket application programming interface comprises a library for developing applications in the C programming language that perform inter-process communication, most commonly across a computer network
    - It allows on the communications between hosts or between processes on one computer using a concept of an Internet socket
    - It can work with many I/O devices and drivers
    - This interface implementation is implicit for TCP/IP and is one of the fundamental technologies underlying the Internet

Internet Sockets

- Internet socket is used in inter-process communications across an IP based network such as Internet
  - Internet socket provides a mechanism for delivering incoming data a packets to the appropriate application process based on a combination of local and remote IP addresses and port numbers
  - Socket is defined as the local end-point of a bidirectional communication flow between local and remote pair of processes connection
- Internet socket is identified by the operating system as a unique combination of the following:
  - Protocol (TCP, UDP or IP)
  - Local IP
  - Local port number
  - Remote IP address
  - Remote port number
- Operation of the socket
  - OS forwards incoming IP data packets to the corresponding application process by extracting the socket address information from the IP, UDP and TCP headers
  - Common and simplified definition occurring in the literature is: “The combination of an IP address and a port number is referred to as a socket”

Socket Implementation

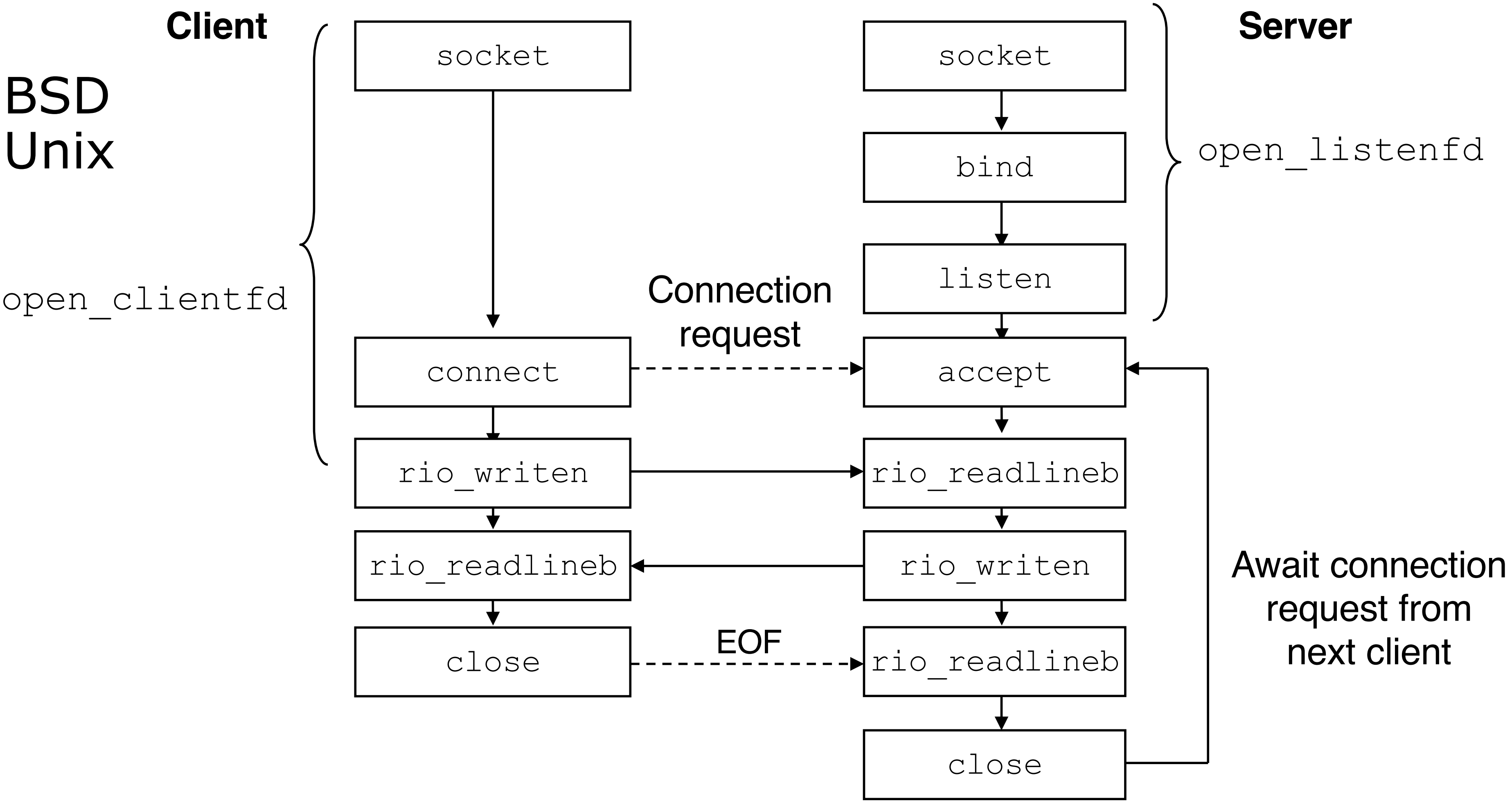
- Most socket implementations are based on the Berkeley sockets
  - Developing of application program using API is called socket programming or network programming
- Examples of functions provided by API library
  - **`socket()`** - creates a new socket of a certain socket type, identified by an integer number, and allocates system resources to it
  - **`bind()`** - is typically used on the server side, and associates a socket with a socket address structure, i.e., a specified local port number and IP address
  - **`listen()`** - is used on the client side, and assigns a free port number to a socket. In case of a TCP socket, it causes an attempt to establish a new TCP connection
  - **`accept()`** - is used on the server side. It accepts a received incoming attempt to create a new TCP connection from the remote client, and creates a new socket associated with the socket address pair of this connection
  - **`send()`** and **`secv()`** or **`write()`** and **`read()`** or **`recvfrom()`** and **`sendto()`** - are used for sending and receiving data to/from a remote socket
  - **`close()`** - causes the system to release resources allocated to a socket, In case of TCP, the connection is terminated

Internet Connections

- **connect(), bind()** and **accept()** functions need a pointer to a protocol-specific sockets address structure
  - **connect()** function tries to establish an Internet connection with the server socket address
- **socket()** and **connect()** functions are wrapped into a helper function referred to as **open_clientfd()**
  - **open_clientfd()** function is used to establish a connection with a server running on host
- **bind(), listen()** and **accept()** functions are used by servers to establish connections with clients
  - **bind()** function tells the kernel to associate the server's socket address with my_addr with the socket descriptor
    - Clients are always active, initializing connection requests, which servers are passive waiting for connection requests from clients
  - **listen()** function converts an active socket into a listening socket that can accept connection requests from clients
  - **accept()** function waits for the connection request from a client, and then fills in the client's socket address

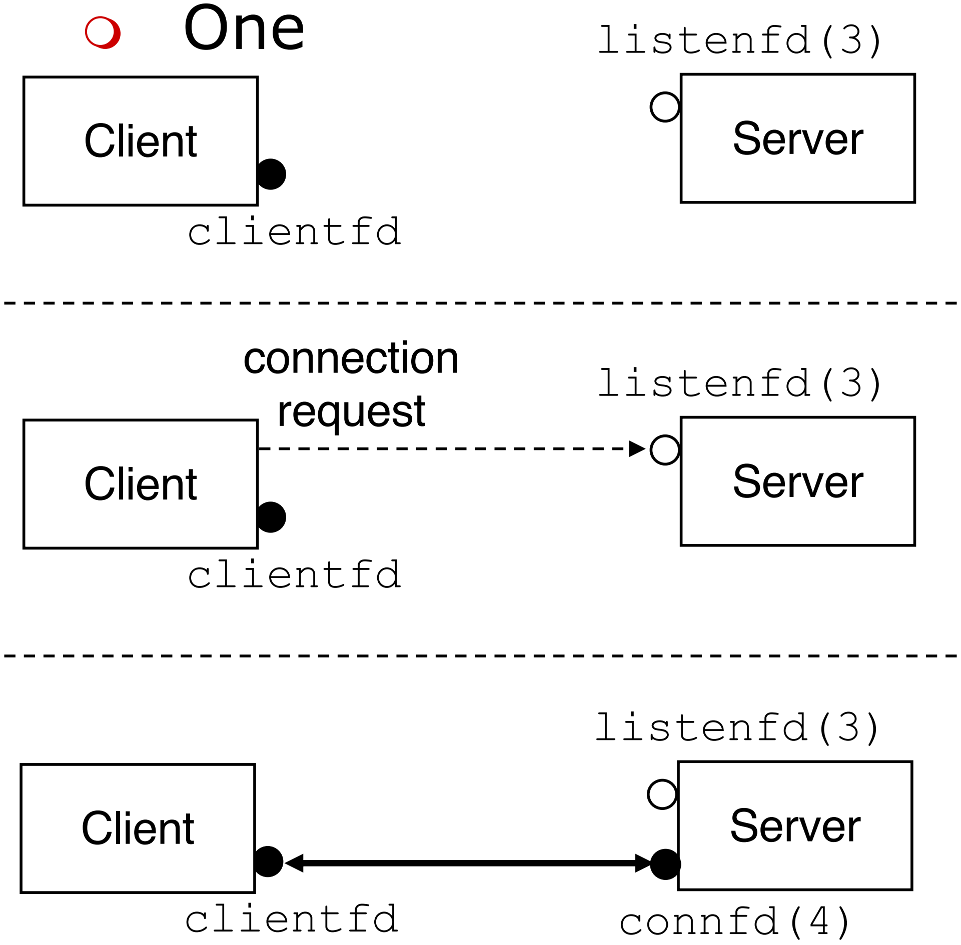
Programming Interface: Sockets

- BSD Unix



Accepting Connection

- One



1. Server blocks in `accept`*, waiting for connection request on listening descriptor* `listenfd`*.*

2. Client makes connection request by calling and blocking in `connect`*.*

3. Server returns `connfd` *from* `accept`*. Client returns from* `connect`*. Connection is now established between* `clientfd` *and* `connfd`*.*

Web Serving

- Web server is a computer program that is responsible for accepting Hypertext Transfer Protocol (HTTP) requests from clients and serving them HTTP responses along with optional data contents which usually are web pages such as HTML documents and linked objects
- Simple: server answers to several commands, most notably GET
  - GET - 90% of Web requests
- Server provides content and closes connection
  - Needs Universal Resource Locator (URL)
- Content: sequence of bytes in MIME type
- Dynamic content: server runs executable file and outputs the results to the client

HTTP

- Web clients and servers interact using text-based request/response standard protocol HTTP (Hypertext Transfer Protocol)
  - The client is the end user while the server is the web site
  - Web client is referred to as web browser
- Operation of HTTP
  - Typically HTTP client initiates a request by establishing a TCP connection to a particular port on a host
  - Any HTTP server listening on that port waits for the client to send a request message
  - Upon receiving the request, the server sends bask a status line, and a message is on its own
  - The server closes the connection
- Web content can be written in HTML (Hypertext Markup Language)

Commonalities of Web Servers

- HTTP: every web server program operates by accepting HTTP requests from the client, and providing an HTTP response to the client
- Logging: usually web servers have the capability of logging some detailed information about client requests and server responses to log file
- Authentication: (optional) authorization request (user name and password) before allowing access to some kind of resources
- Content compression (gzip encoding) to reduce the size of the responses
- Large file support to be able to serve files whose size is greater than 2 GB

Anatomy of a Web Transaction

- HTTP: Text lines transmitted over Internet
- Can open terminal connection to any Web server
- Displays the content of the (readable) directories
  - “/” = homepage, i.e., /index.html

```
mac-[~] 6>telnet www.bbc.com 80
Trying 212.58.240.33...
Connected to www.bbc.com.
Escape character is '^]'.
GET /
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01 Transitional//EN">

<html>
<head>
        <title>BBC - homepage …</title>
…

</html>
Connection closed by foreign host.
```

HTTP Requests

- HTTP request consists of request line
- Request line: <method> <uri> <version>

  e.g., GET / HTTP/1.2
- Methods: GET, POST, OPTIONS, HEAD, PUT, DELETE, TRACE
  - For example GET (constitutes 99% of all requests) instructs the server to generate and return the content identified by URI
    - URI (Uniform Resource Identifier) is a string of characters used to identify or name a resource on the Internet
    - URL (Universal Resource Locator) is a type of URI that specifies where an identified resource is available and the mechanism for retrieving it
    - URI is the suffix of the corresponding URL that includes the file name and optional arguments
    - URI completes URL or the suffix of URL (no “http://”)

HTTP Requests, cont.

- <version> field in the request line stand for the HTTP version of the request
- Request headers <header name>: < header data>

  e.g., host: www.mcgill.ca
  - Provide additional information to the server such as the name of the browser
- Empty text line
  - End of inputs, equivalent to pressing “enter” button

HTTP Responses

- HTTP responses are similar to HTTP requests
  - The format of the response consists of the a response line followed by zeros or more response headers, followed by an empty line terminating the headers, ending with the response body
- Response line: <version> <status code> <status message>

  e.g., HTTP/1.1 200 OK
- <version> field stands for the HTTP version that the response conforms to
- <status code> is a three-digit positive integer that indicates the disposition of the request
- <status message> is the English equivalent of the error code

HTTP Responses, cont.

- Response header gives additional information about the response
- Response headers <header name>: < header data>

  e.g.Server: Apache/2.0.52 (Red Hat)

  X-Powered-By: PHP/5.0.4

  X-Cluster-Name: mccarthy

  Connection: close

  Transfer-Encoding: chunked

  Content-Type: text/html; charset=UTF-8
- Empty text line
- Response body: requested content

  e.g. <html> … </html>

Passing Arguments and Dynamic Content

- HTTP protocol support the passing of arguments to the server
  - Arguments are postpend to the URL separated from the URL by a question mark (?)
    - Most requests are dealing with database search, and the passed arguments are the search parameters
      - More than one argument can be passed. They are then separated by a plus sign (+)
- Content provided by executables, CGI programs

  e.g., GET /cgi-bin/multiply?24?7 HTTP/1.1

  where "multiply" is your program written in C (or anything else, including scripting)
    - Cgi-bin directory is a special location known to the server containing executable programs
      - These programs interface with the web
      - URL can access and pass argument to programs in this directory and these programs can act on the arguments and return the information to the browser

A Simple Web Server - Summary

- Repeats forever
  - Listens to a port (say, 80)
  - Parses input
  - Sends response
- Feasible in smallest embedded systems

Concurrent Computing

- Concurrent computing is a form of computing in which programs are designed as collections of interacting computational processes that may be executed in parallel
- The programming can be executed sequentially on a single processor by interleaving the execution steps of each computations process, or executer in parallel by assigning each computational process to one of a set of processors that may be in close proximity or distributed across network
- The main challenges are ensuring the correct sequencing of the interactions or communications between different computational processes, and coordinating access to resources that are shared between processes

Multithreaded Server

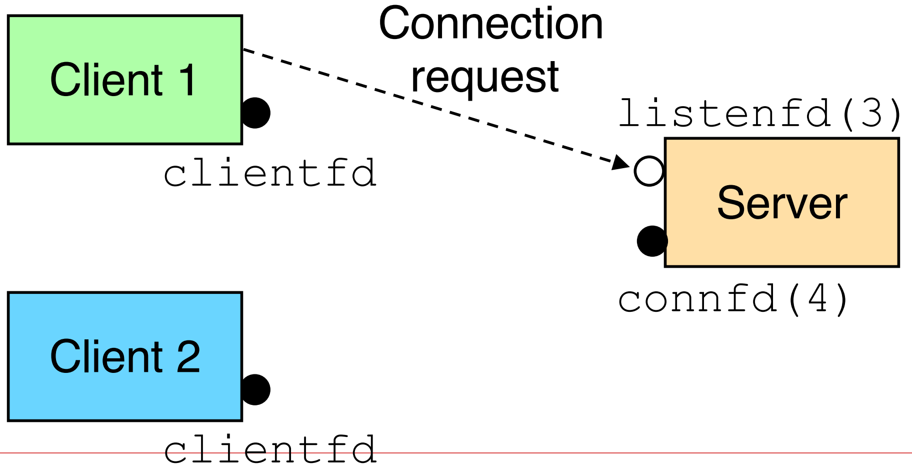
- A necessity for large throughput servers accommodating the concurrent programs
  - A sound way of building a concurrent server is to accept client connection requests in the parent and then create a new child to service each new client

Thread

- Thread of execution is a fork of a computer program into two or more concurrently running tasks
- The implementation of threads and processes results in a thread contained inside a process
- Multiple threads can exist within the same process and share resource sich as memory

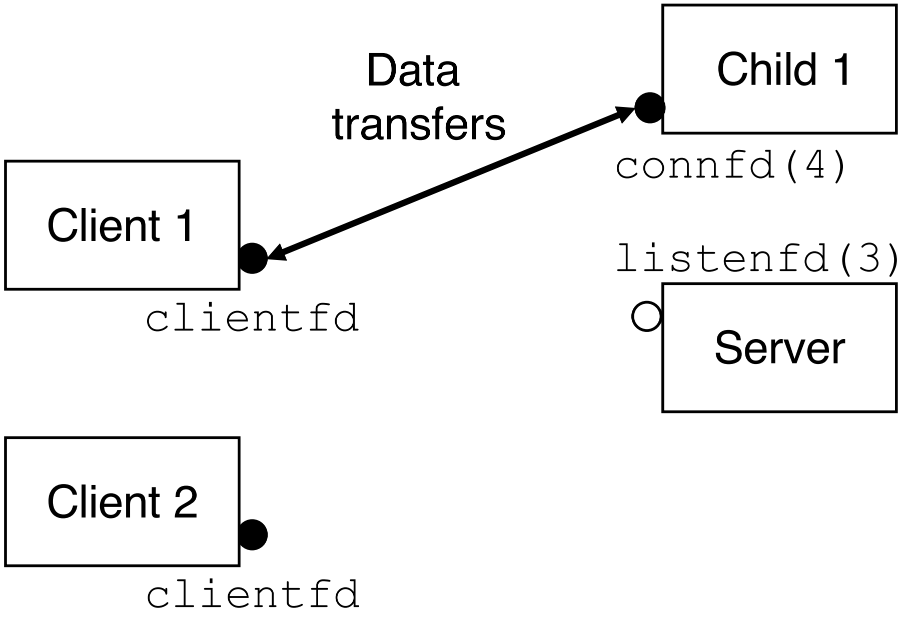
Multithreaded Server - Example

- Assume that there are two clients and a server that is listening for connection requests on a listening descriptor
- Suppose that the server accepts a connection request from client 1
    - After accepting a connection request, the server forks a child, which gets a complete copy of the server's description table



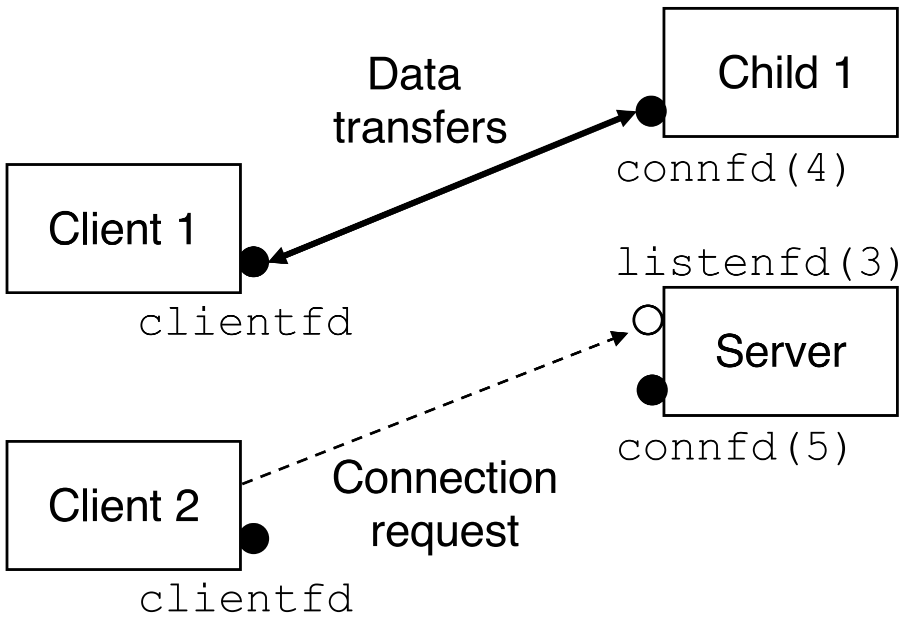
Spawning a Thread

- Response to request



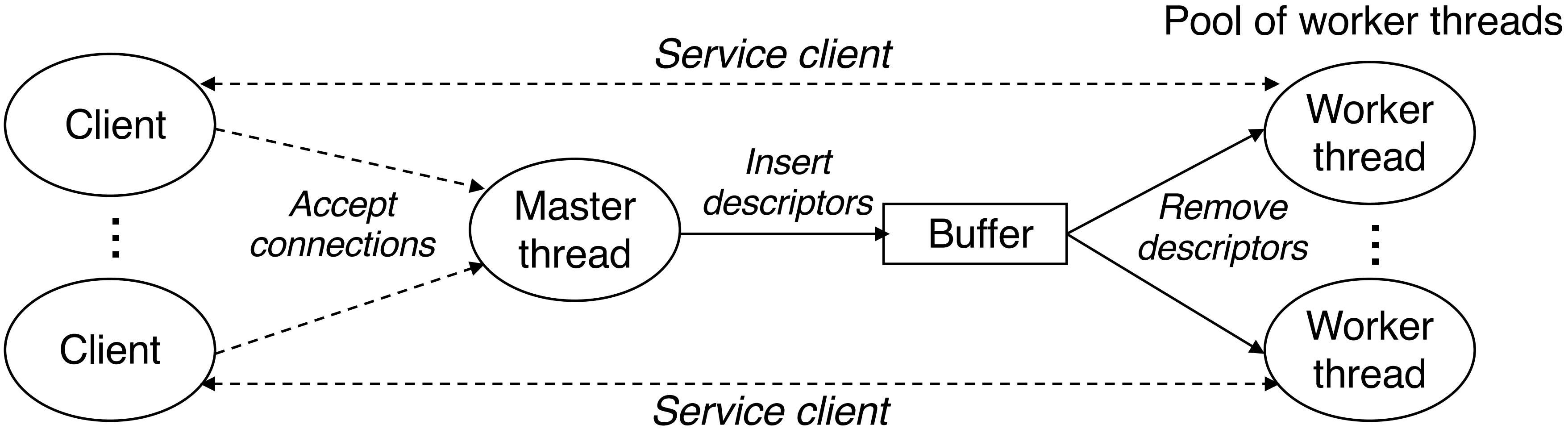
Answering Multiple Requests

- Reason for multithreaded servers



Pre-threading for Web Servers

- Common solution on high-performance servers
- Instantiate threads that idle until a request comes



Going Beyond Web Serving

- Running code remotely
  - Networked program execution
- Issues:
  - Security
  - Standardization
  - Performance

Lab 4: Outline

- Exercise sensor interfacing
- Design sensor processing
  - FreeRTOS
- Concrete task 1: Sensor data reading
- Task 2: OS incorporation
  - Determine fit to OS primities