CSE 3231
Computer Networks

Chapter 1 - Overview of Networking

William Allen, PhD

Spring 2022

What is a Computer Network?

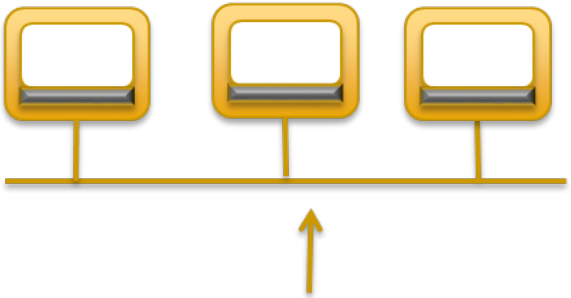
- Consider the case where two or more machines (or users) are locally connected

- Because networks are often drawn as a graph (a mathematical *graph*, not a chart)
  - each “machine” or “device” is called a *node*
  - the connection between them is called a *link*

What is a Computer Network?

- Physical links can be limited to a pair of nodes or can include multiple nodes

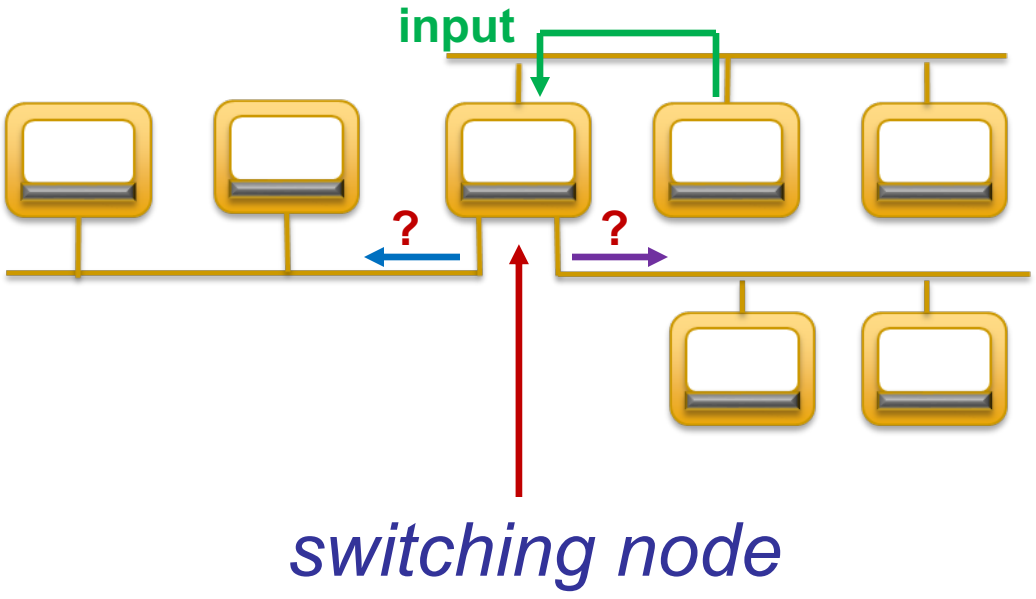
Point to Point Link



Multiple Access Link

Switched Networks

- Nodes connecting multiple links can control which link receives incoming traffic, creating a *switched network*



Network Hardware

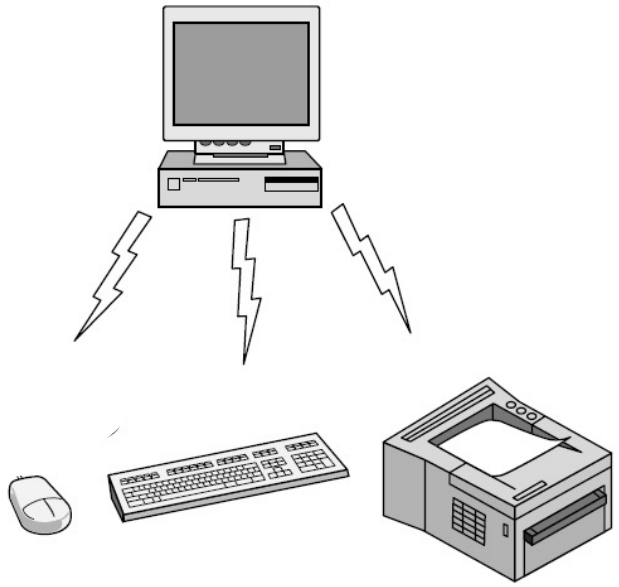
- Networks can be classified by their scale:

Scale	Type
Vicinity	PAN (Personal Area Network)
Building	LAN (Local Area Network)
City	MAN (Metropolitan Area Network)
Country	WAN (Wide Area Network)
Planet	The Internet (network of all networks)

- An "internetwork" is any larger network made up of smaller component networks.
- The "Internet" (with a capital **I**) is the set of *all* connected networks.

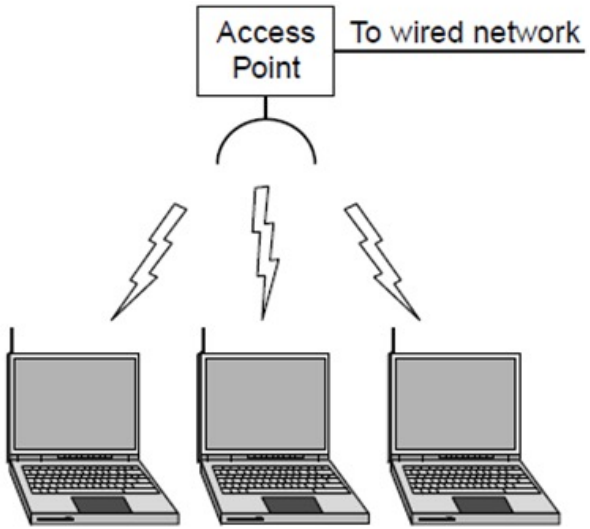
Personal Area Network

- Connect devices in the area around a person
  - Example: a Bluetooth (wireless) PAN

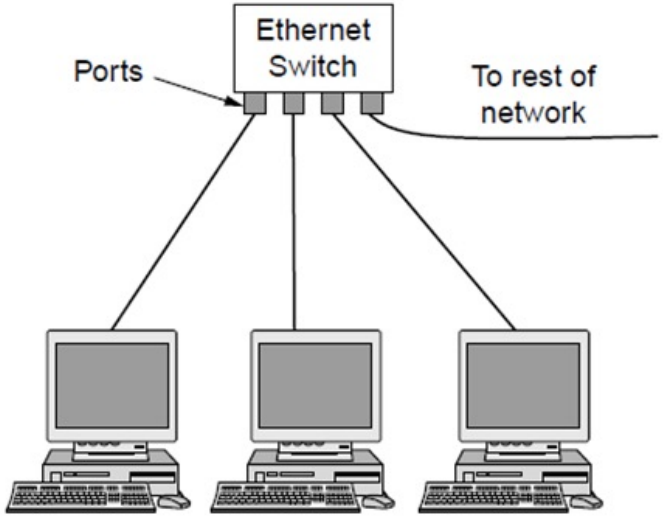


Local Area Networks

- Connect devices in a home or office building
  - May be referred to as an enterprise network



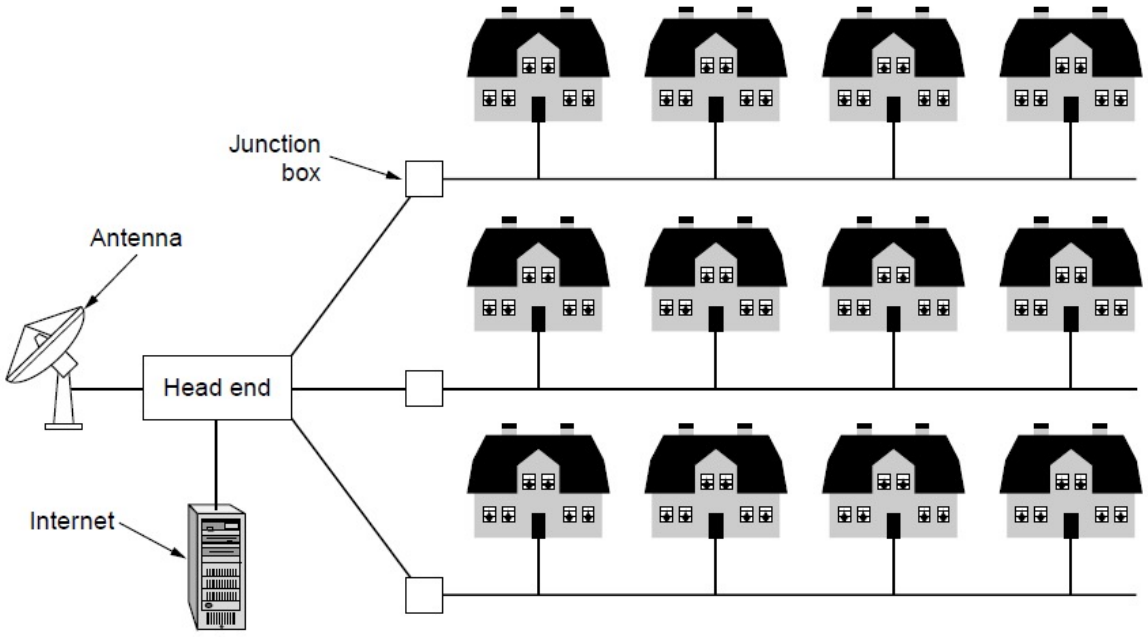
Wireless LAN with 802.11



Wired LAN with switched Ethernet

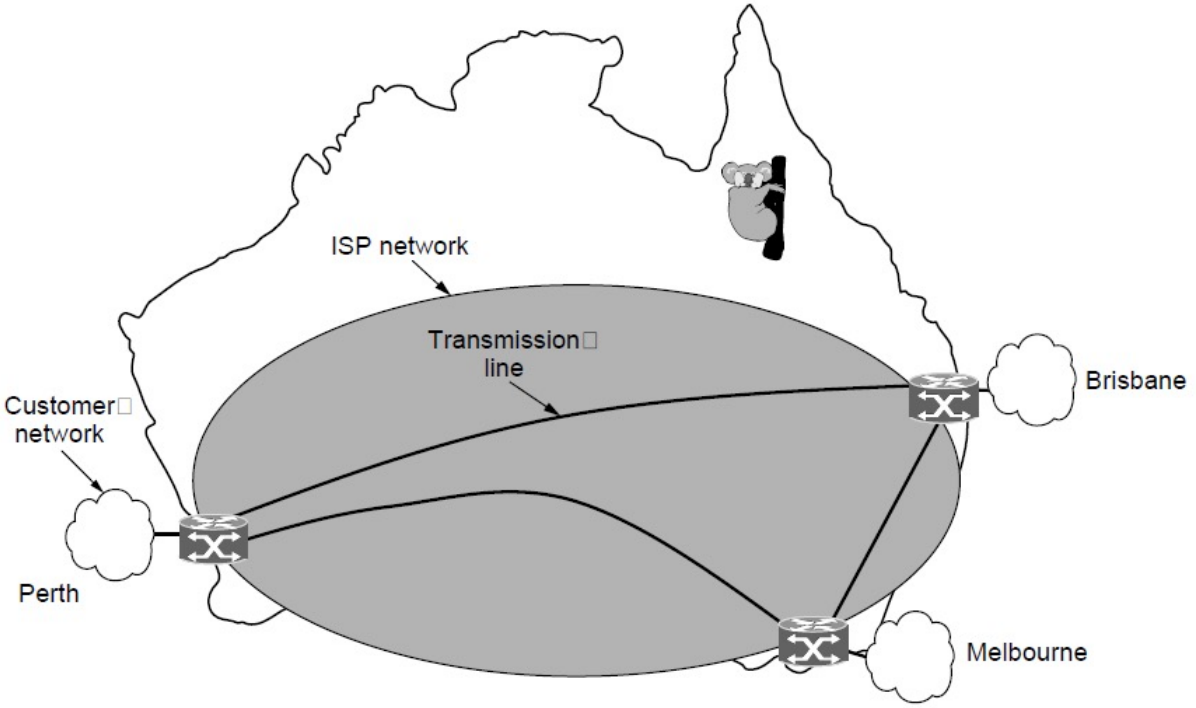
Metropolitan Area Networks

- Connect devices over a metropolitan area
  - Example: MAN based on cable TV



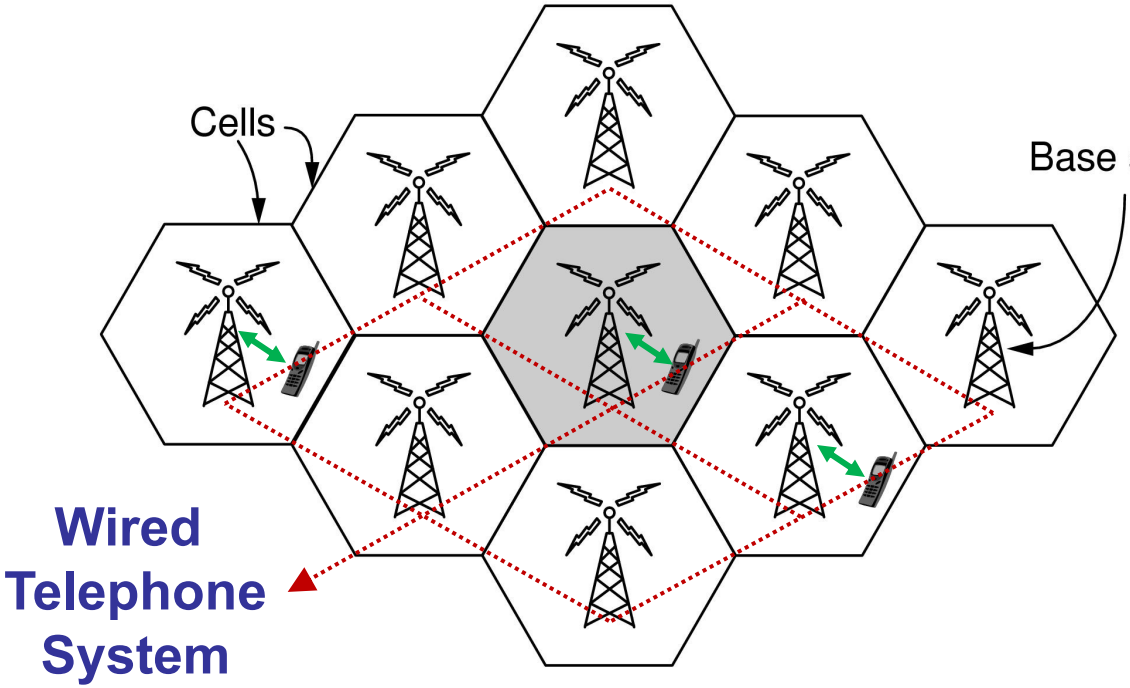
Wide Area Networks

- An ISP (Internet Service Provider) network is also a WAN and it connects to other WANs
  - ISPs charge customers to connect to the Internet



Mobile Phone Networks

- Cellular networks are based on spatial cells; each cell provides wireless service to mobile devices within it and connects to a base station

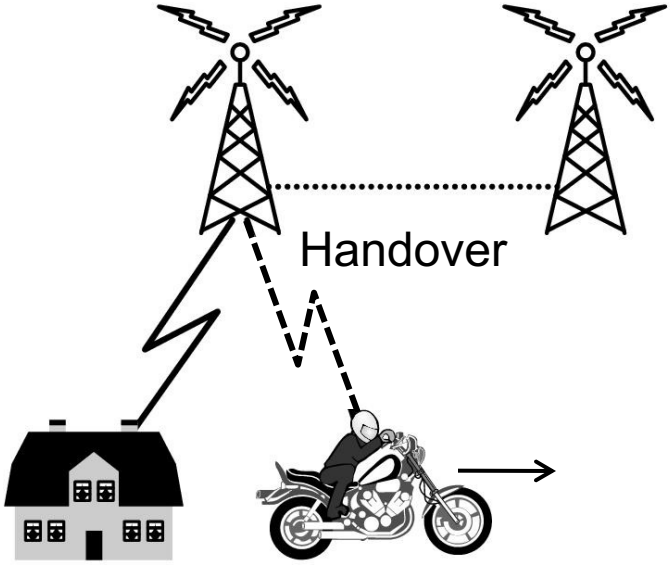


Each base station connects to other base stations in the cellular network and to the wired telephone system

Mobile Phone Networks

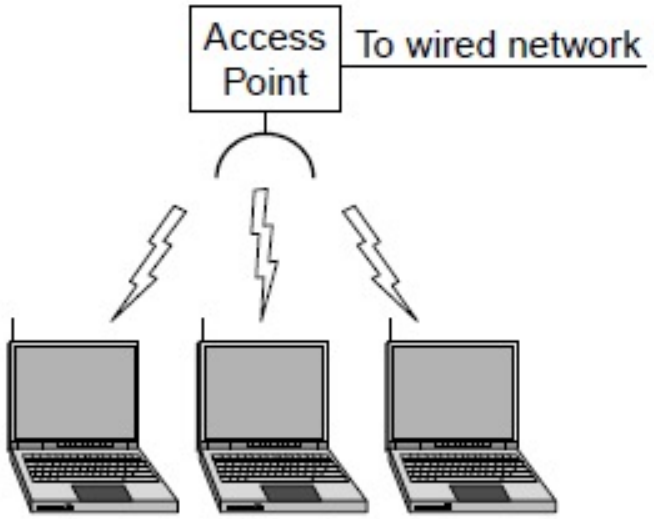
As mobile devices move, base stations hand them over from one cell to the next

- networks track the device's signal and location to predict the next cell they will enter so it can allocate a radio channel and prepare for the *handover*



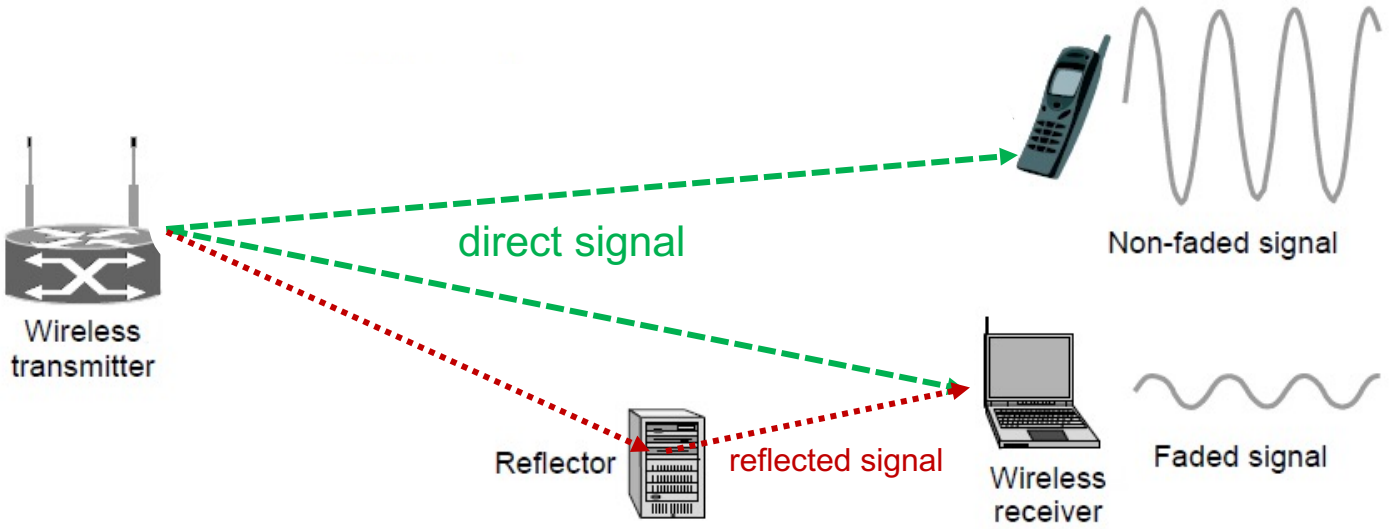
Wireless LANs

- Wireless devices using the IEEE 802.11 protocol communicate via an AP (Access Point) that is wired to the rest of the network
  - It is also possible to create an *ad hoc* (when needed) network by connecting devices directly to each other (no AP needed)



Wireless LANs

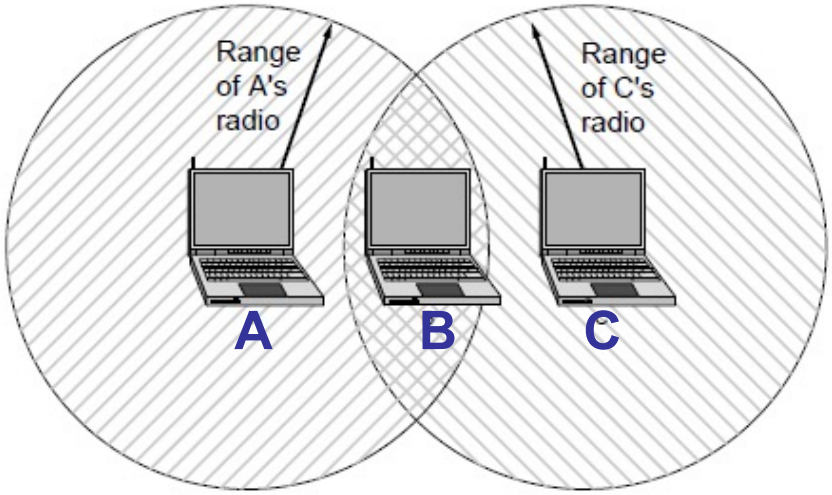
- Signals in the 2.4GHz band vary in strength due to many effects, such as *multipath fading* due to reflections or atmospheric effects
  - if the reflection arrives "*out of phase*", it can interfere with the original signal



Wireless LANs

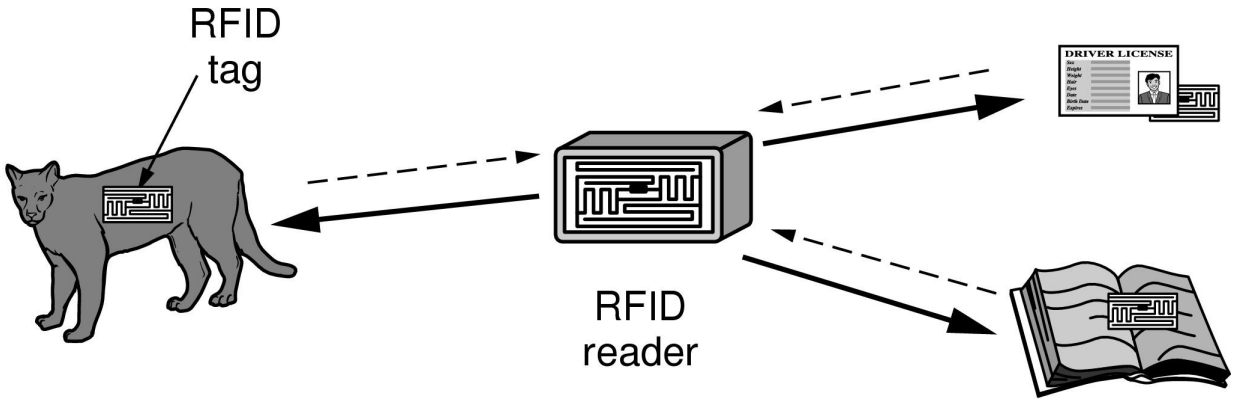
- Radio signals may interfere with each other and radio ranges may not overlap completely
  - as mentioned earlier, Multiple Access Collision Detection and Avoidance techniques can be used, but may not always solve the problem

In this example, **A** and **B** can detect and avoid collisions. But **C** does not "know" when **A** is transmitting. Thus, **C** can interfere with signals sent from **A** to **B** without being able to detect that a collision occurred



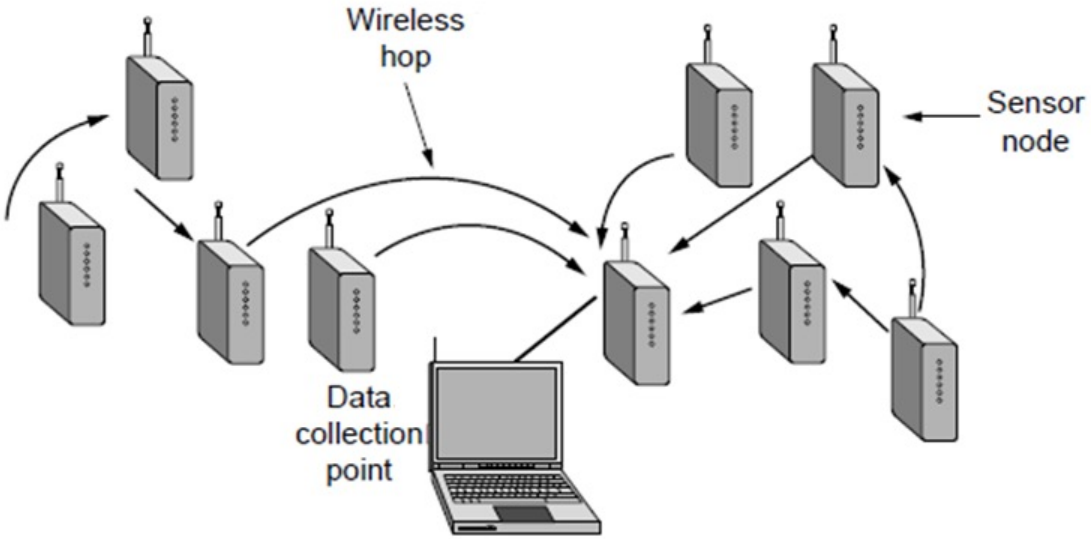
RFID Networks

- RFID (Radio Frequency Identification) networks can be used to track objects
  - RFID readers send signals to detect tags on objects
  - When they receive the signal, the RFID tags on those objects will respond with their ID information

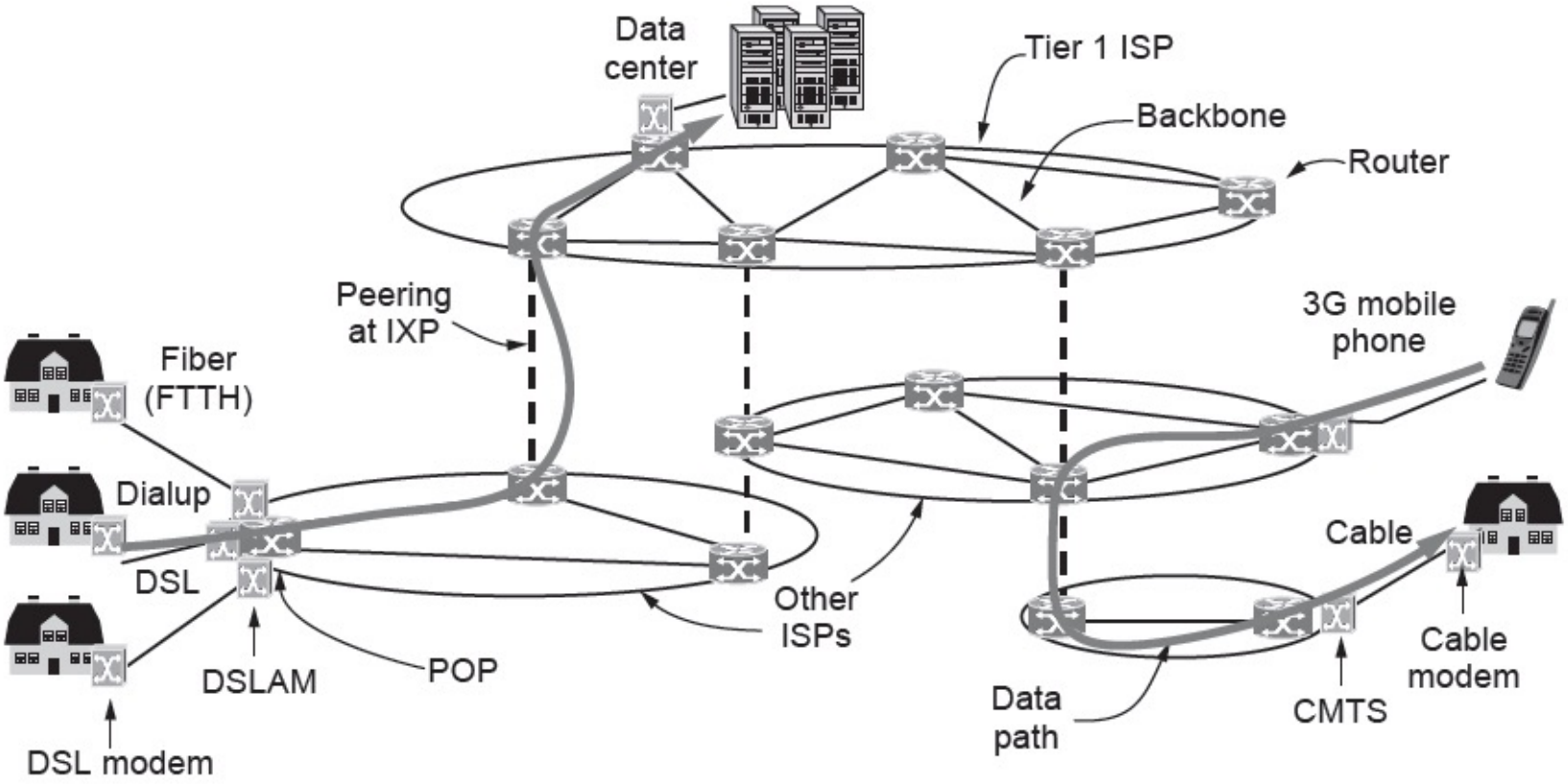


Sensor Networks

- Sensor networks spread many small devices over an area for data collection
  - The devices send the collected data to a data collection point via wireless connections



The Internet



The architecture of the Internet

Common Protocols

- Personal Area Networks
  - Bluetooth - originally IEEE 802.15, now managed by a group of companies that make devices that use Bluetooth
- Local Area Networks
  - wired - currently most often IEEE 802.3 (Ethernet) with updates
  - wireless - most commonly IEEE 802.11 (WiFi) with updates
- Metropolitan Area Networks
  - may be an optical fiber-based extension of Ethernet
- Wide Area Networks
  - also use optical fiber with higher speeds (SONET, ATM, etc.)
- The Internet
  - at lower layers uses WAN protocols, higher layers use TCP/IP

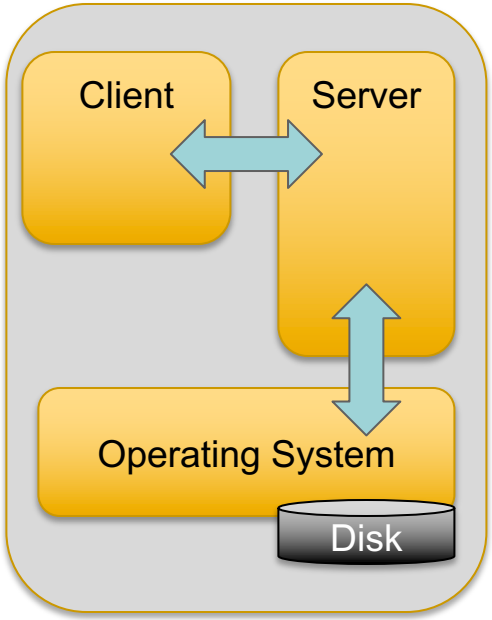
Basic Requirements

- Communications Medium
  - are there multiple users on a *shared medium*?
- Encoding – (how is the data transmitted?)
  - frequency encoders and decoders
  - conversion between signals & recognizable symbols
- Common vocabulary and language
- Coordination and pausing during a conversation
  - *subdividing* information into transmittable sizes
  - *acknowledgement* of reception
  - *taking turns* to transmit without interferring

Simple example of a non-networked application

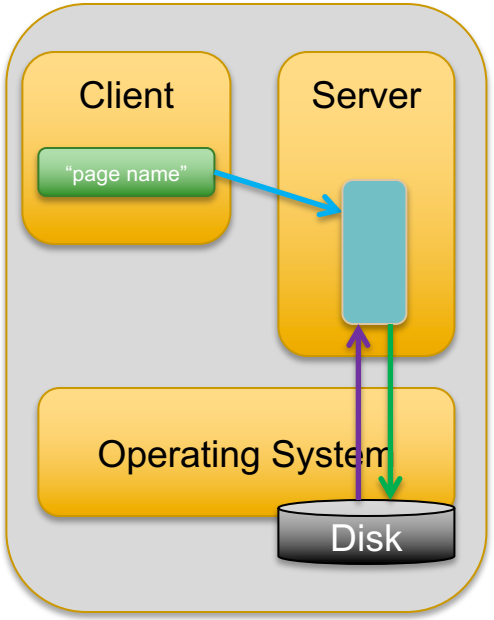
Consider a simple, client/server application where both the client and server are running on the same machine

- There is a physical database in the machine with a set of files
- The server has access to the database containing the files (through the OS)
- The client requests access to the files through the server



Simple example of a non-networked application

- Example: A Simple Client Request
  - The client creates a request object (for example a string, or a query with the name of a file)
  - The request is sent to the server through a **method call**
  - The server receives the request and retrieves the desired page by making **system calls** to the OS



Basic Requirements

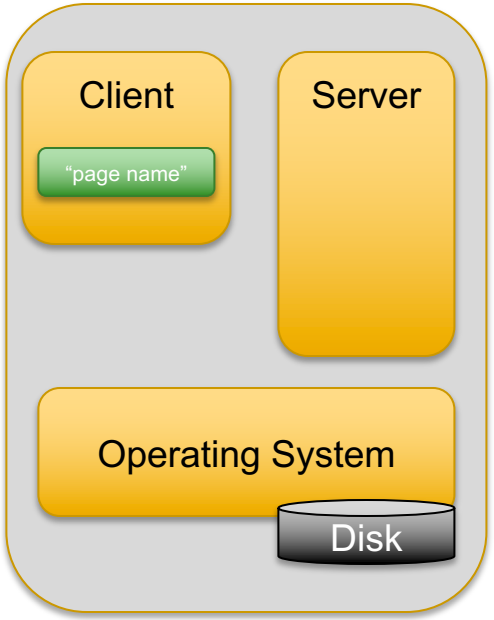
- Communications Medium
  - are there multiple users on a *shared medium*?
- Encoding – (how is the data transmitted?)
  - frequency encoders and decoders
  - conversion between signals & recognizable symbols
- Common vocabulary and language

Simple example of a non-networked application

Consider a simple, client/server application where both the client and server are running on the same machine

Simple example of a non-networked application

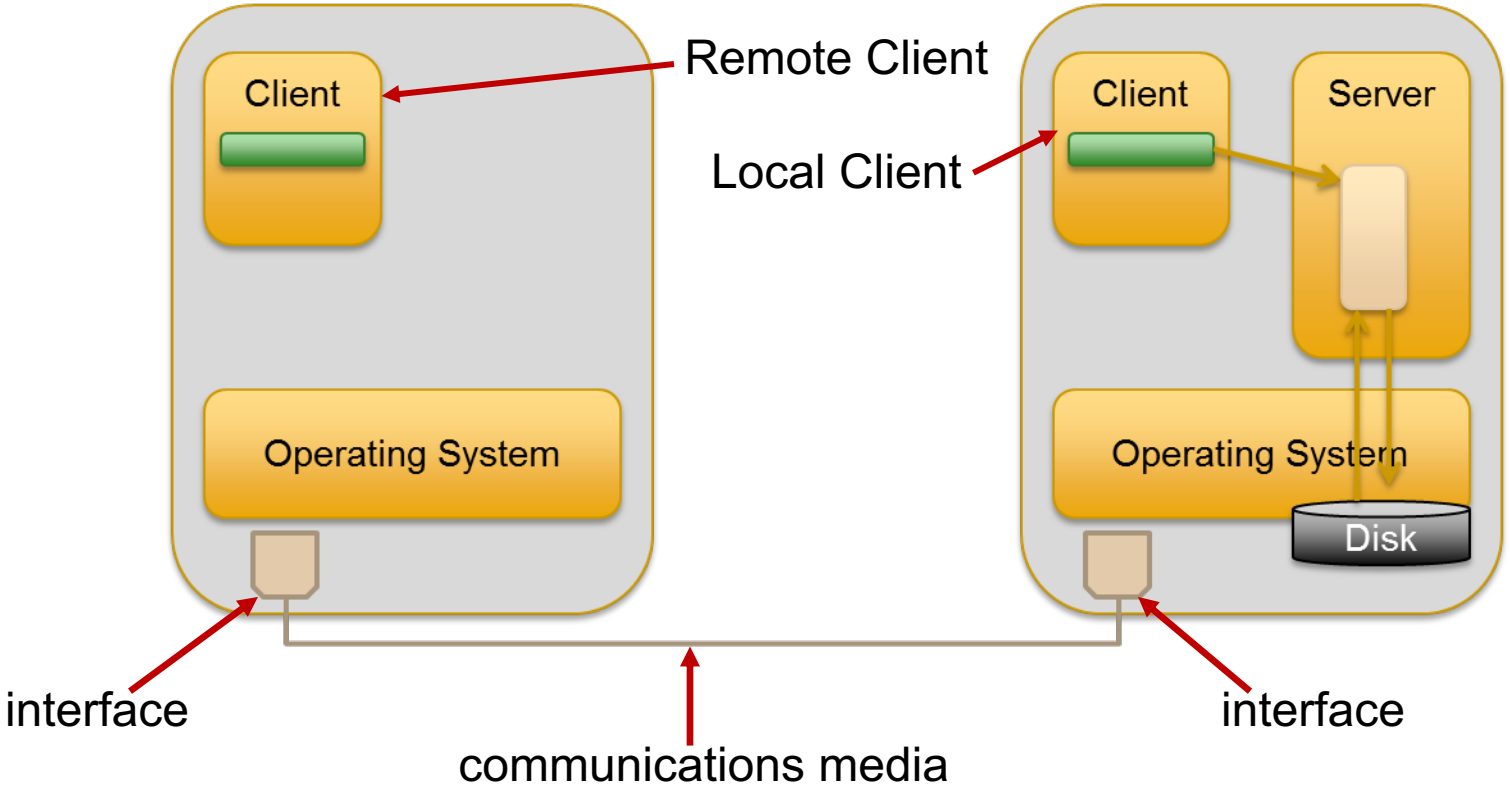
- Example: A Simple Client Request
  - The client creates a request object (for example a string, or a query with the name of a file)



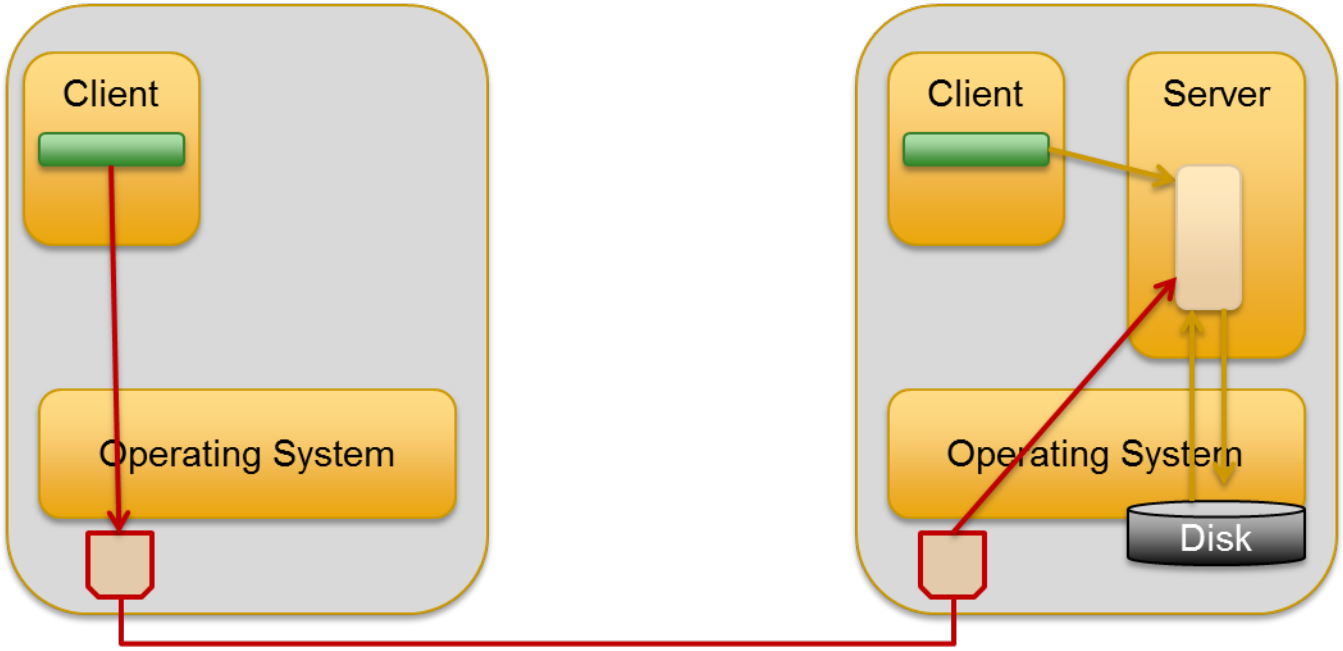
Networked Version of the Client/Server Application

- We must separate the Client and Server so that they can connect over a network:
  1. we need a communications media (wire, radio)
  2. also, each node needs a physical interface to that communications media
  3. we must encode/decode data into blocks of bits that are then convered into a signal to be transmitted
  4. and we have to detect any bit-errors or data loss during transmission

Networked Version of the Client/Server Application

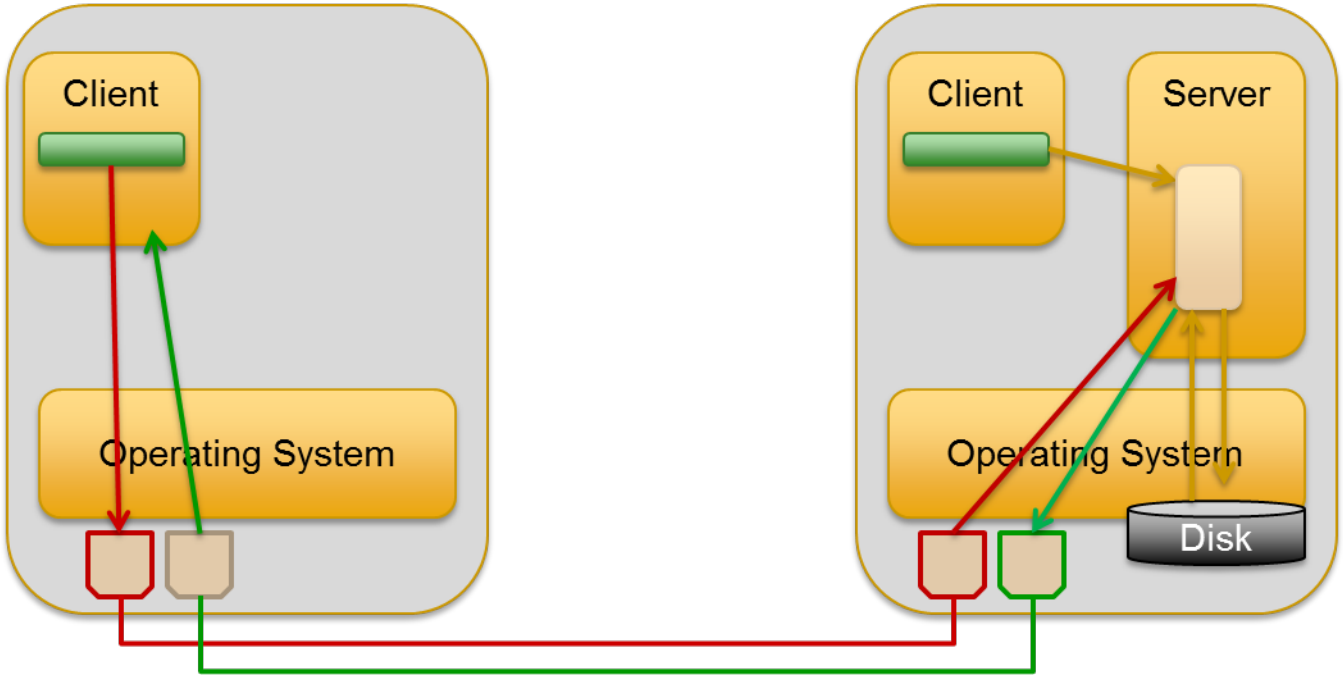


Networked Version of the Client/Server Application



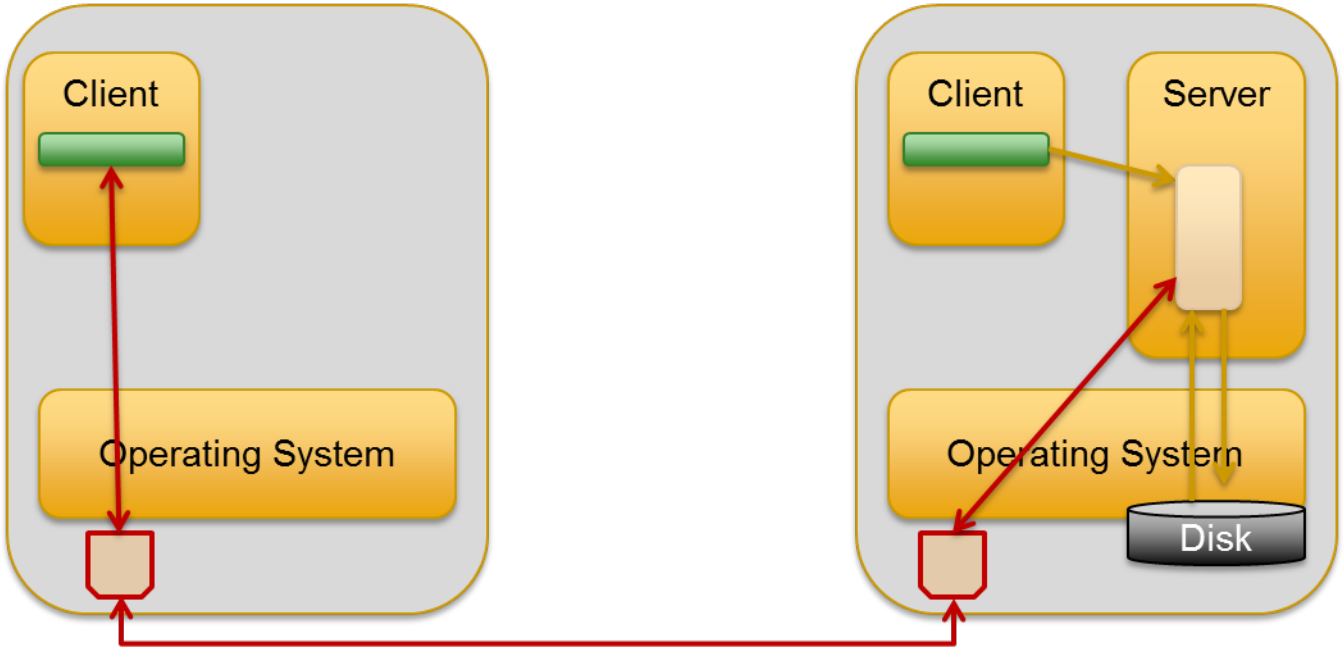
- Communications Media
  - first, consider a dedicated, uni-directional circuit.
- The client can encode and send the request as a stream of bits which are transmitted to the server and decoded

Networked Version of the Client/Server Application



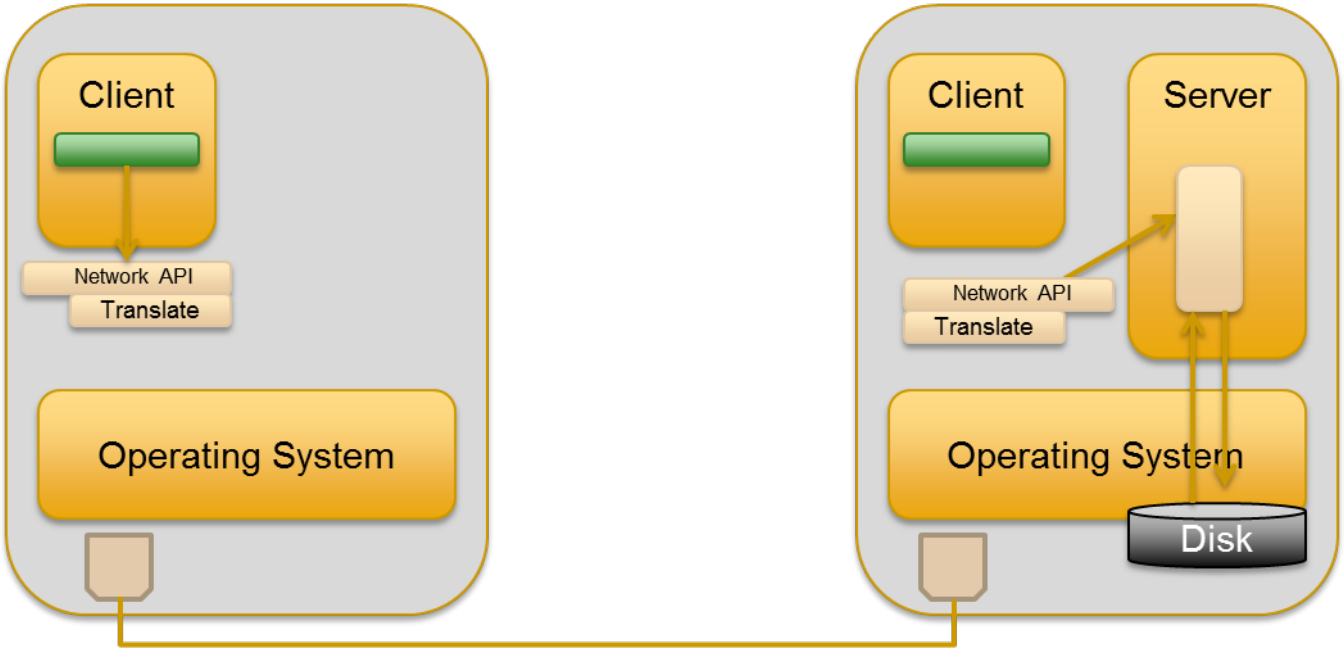
- Communications Media
  - first, consider a dedicated, uni-directional circuit.
- However, in this case, the *response* would require another dedicated circuit with its own interfaces

Networked Version of the Client/Server Application



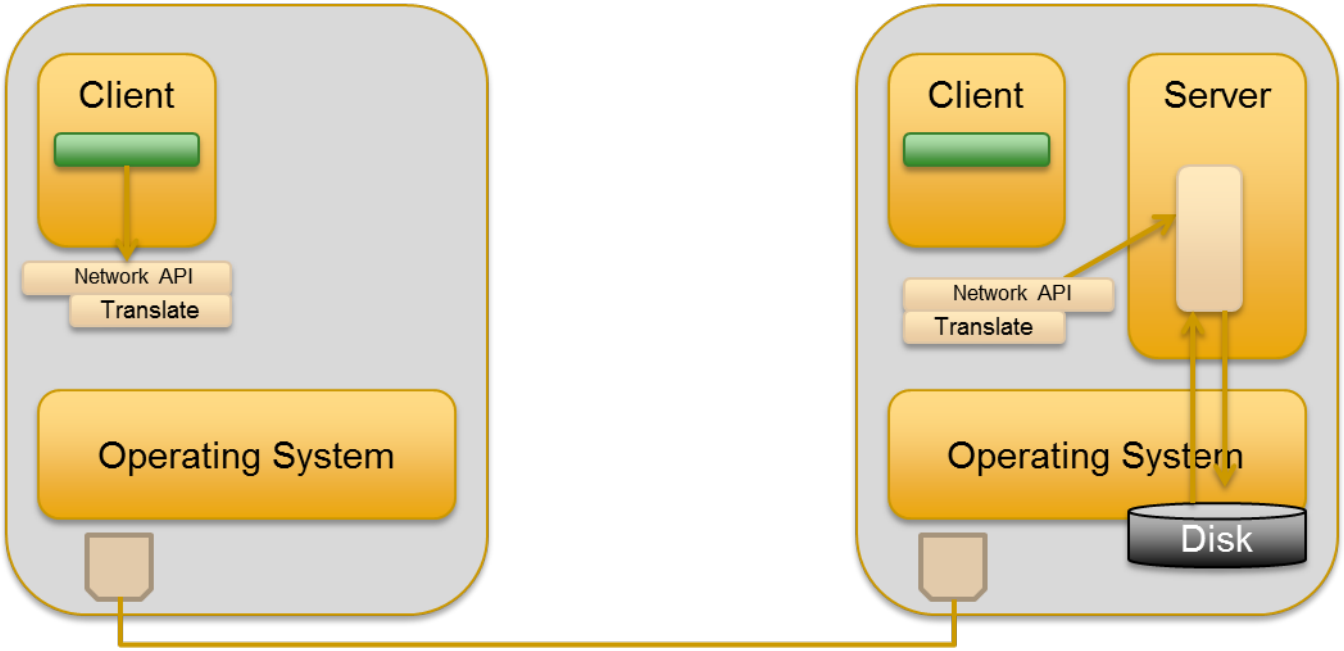
- Communications Media
  - instead, we could used a *shared link* between nodes
- In this case, the client and server will have to coordinate so they can take turns to use the shared link

Networked Version of the Client/Server Application



- Network API for applications
  - Applications and Operating Systems on each machine may have different ways to format the data objects

Networked Version of the Client/Server Application

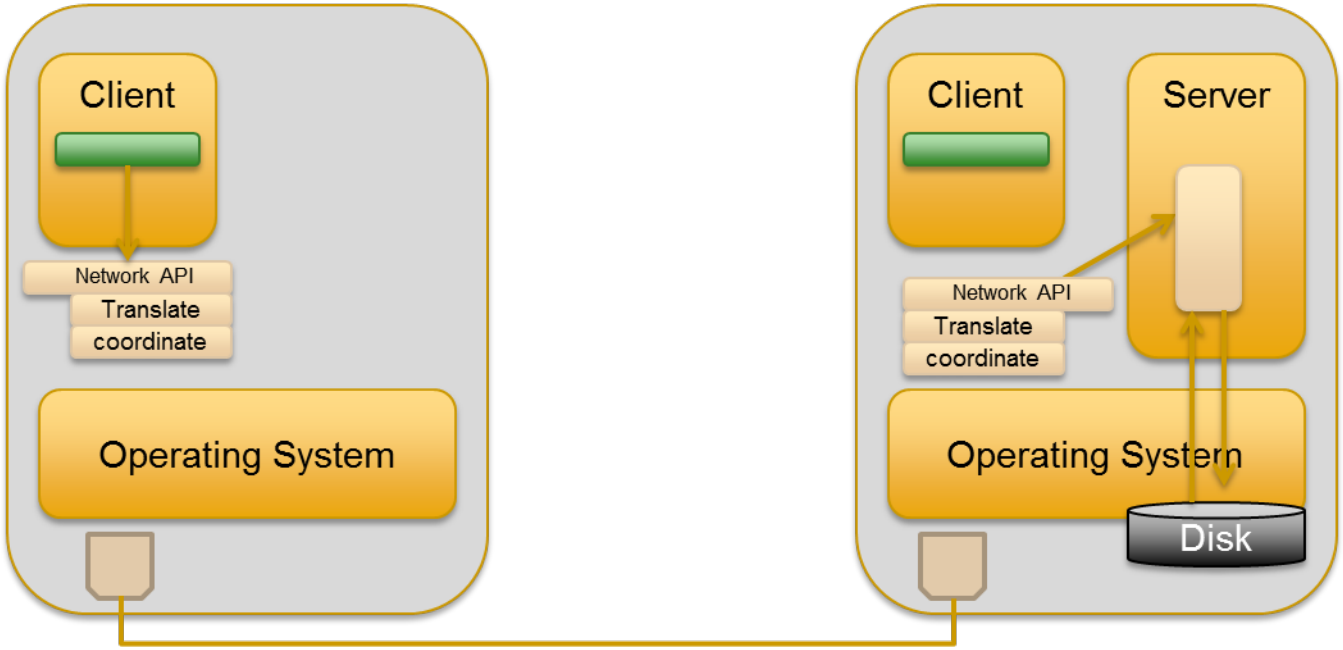


- Network API for applications
  - Applications and Operating Systems on each machine may have different ways to format the data objects
  - There must be a common representation to share information and a way to convert objects to/from that common network representation

Data Representations

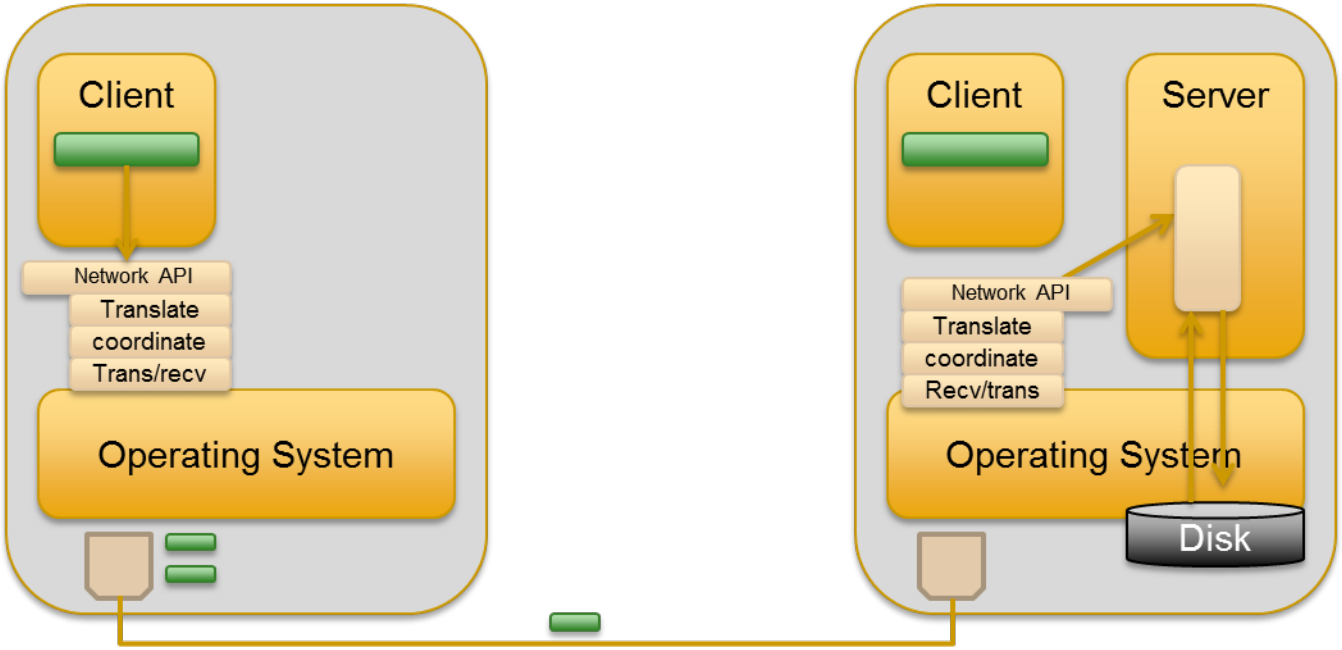
- Alphabetic characters are often stored in ASCII, but they may also be in Unicode (and other character formats exist)
  - if you receive 12 bytes, is that twelve ASCII characters or six 16-bit Unicode characters?
- Binary data can be stored in either Little-Endian or Big-Endian order, how do we know which?
  - which does your computer use? your network?
- There has to be a way of determining which format is being used and a way to convert data from one format to another, when needed

Networked Version of the Client/Server Application



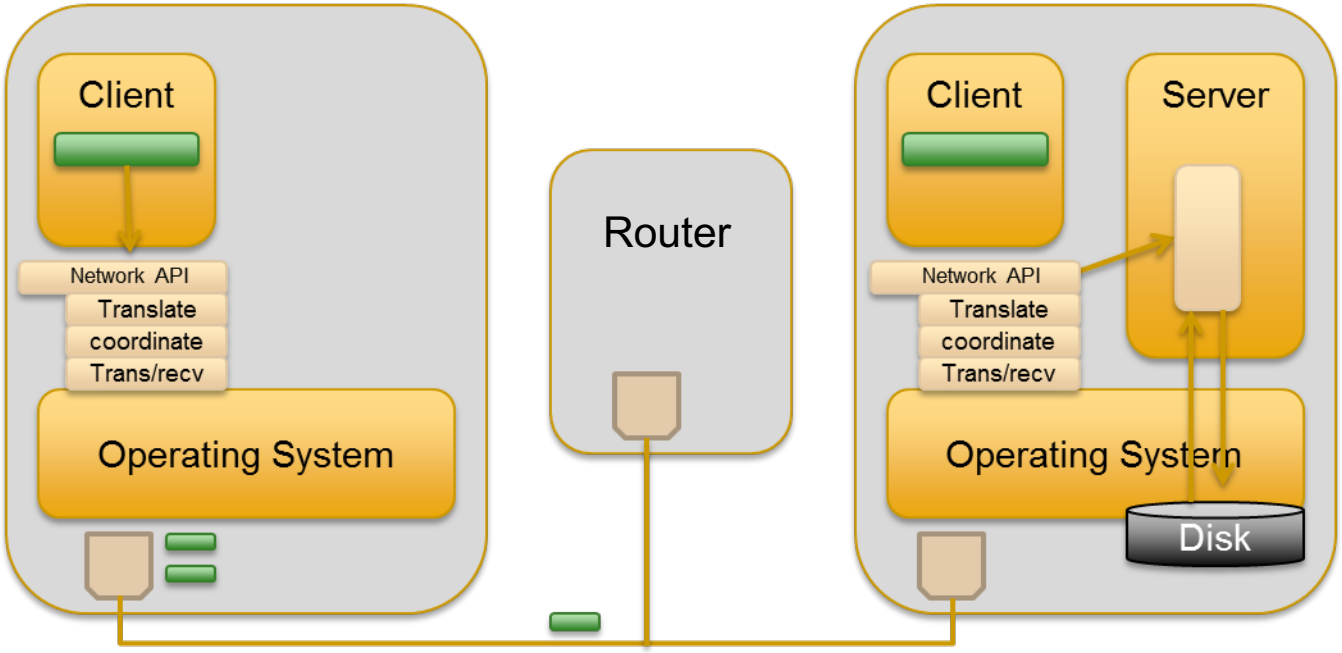
- We have to coordinate the actual transmission between machines
  - Start and end sessions, synchronize, timeout sessions…
  - Software interfaces are needed to send/receive messages
  - Error checking is needed for large-sized data transmissions

Networked Version of the Client/Server Application



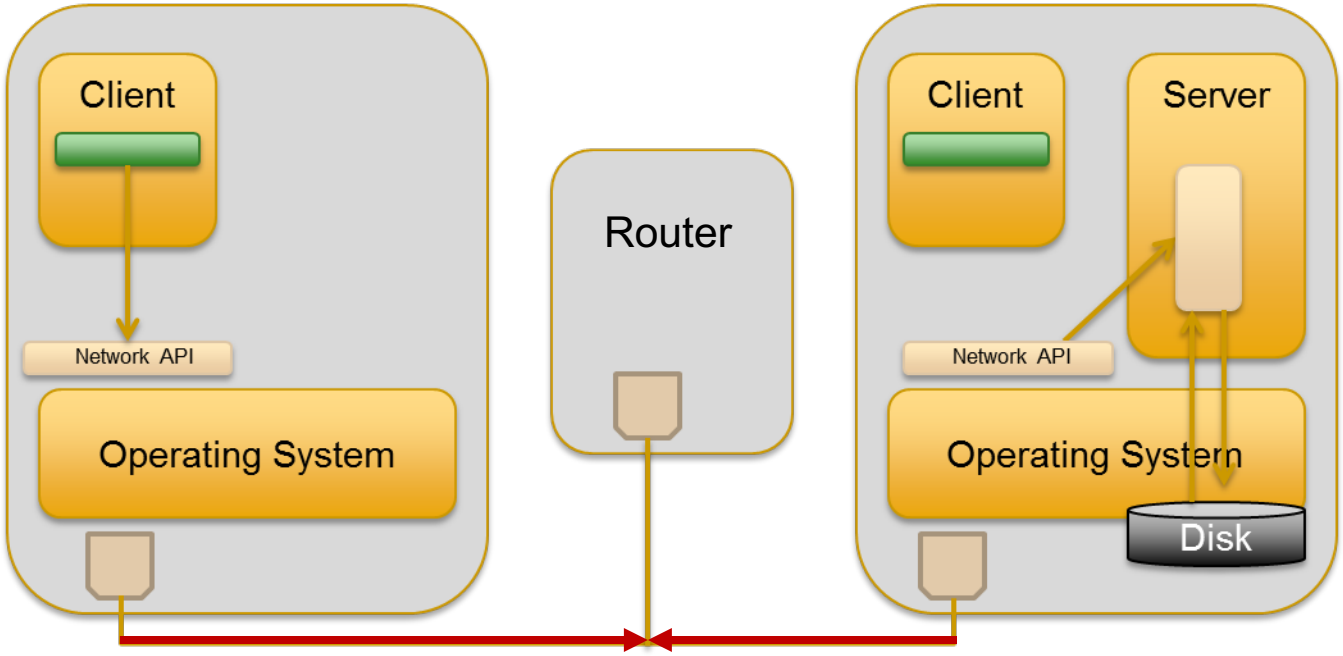
- We have to manage the end-to-end transmission of large data
  - Break a large message into ‘chunks’
  - Create mechanisms to determine how to transmit, receive, and verify those chunks

Networked Version of the Client/Server Application



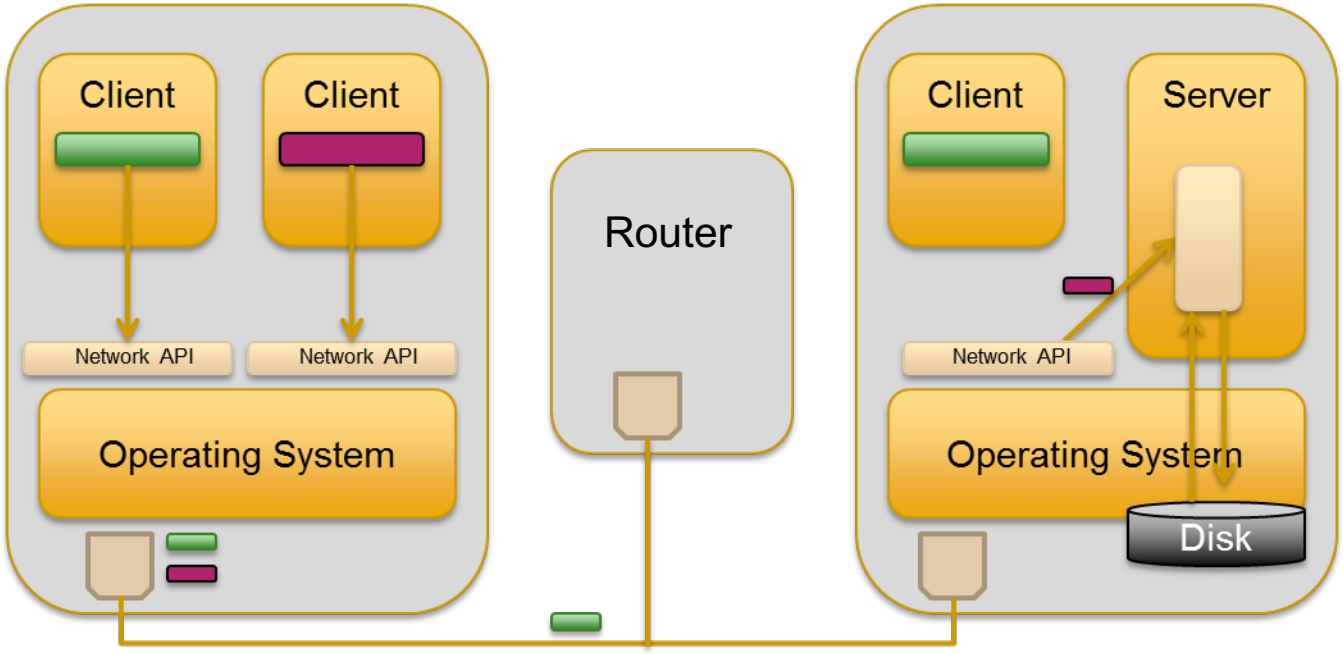
- Addressing of data packets
  - We must support the client's need to find the addresses of servers
  - We must handle issues with conflicts in addressing

Networked Version of the Client/Server Application



- Multiple Access Control
  - Multiple nodes accessing the shared medium at the same time can cause *collisions* which prevent the delivery of data

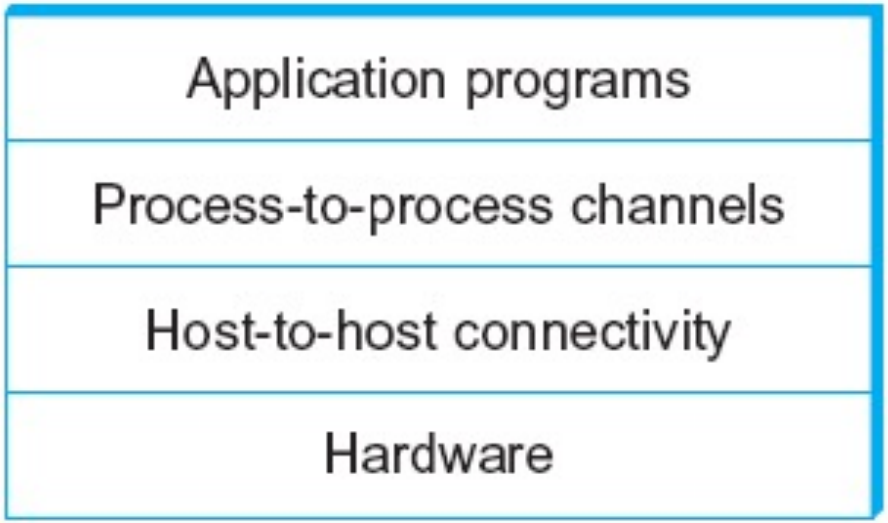
Networked Version of the Client/Server Application



- Channel Multiplexing / De-multiplexing
  - We must handle multiple clients or applications sharing the same link without confusing *which client* is talking to each application

Network Architecture

- Network design is modular; different hardware and software components of the network are handled in different ways and this works best with a layered design
  - Each layer handles a specific part of the network



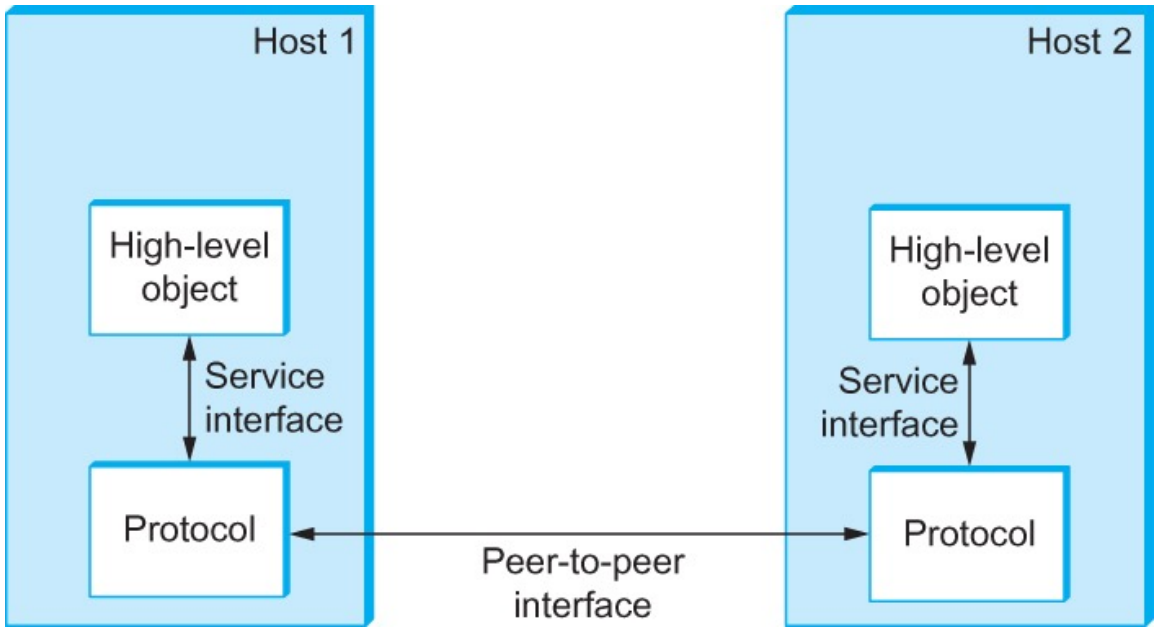
Layered Network Architecture

- Independent of the Network Model:
  - Networks are organized as a series of layers, each higher layer building on the services and capabilities of the layer below.
  - Reduces complexity
  - Supports interoperability
- Layers are designed to handle specific functionalities
  - Link access management
  - Routing
  - End-to-end connectivity

Protocols

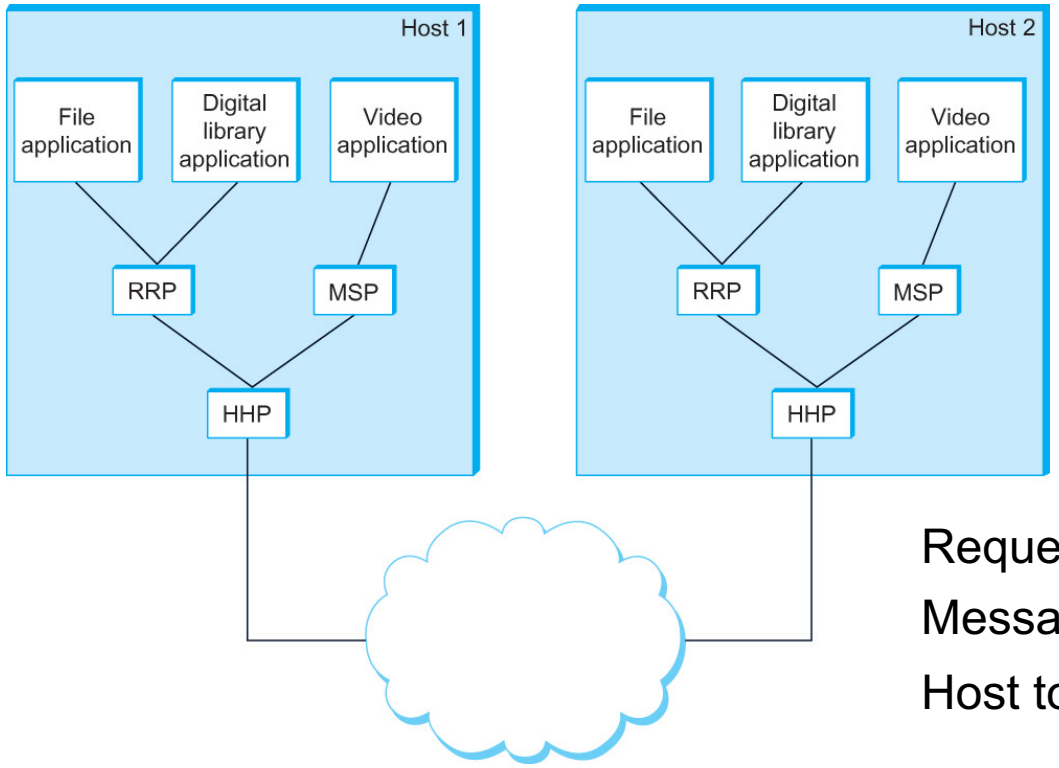
- A **protocol** defines the interfaces *between the layers* in the same system and also the interfaces *with the same layers* of a peer system
  - Forming the building blocks of a network architecture
- Each protocol object has two different interfaces
  - service interface: operations on this protocol
  - peer-to-peer interface: messages exchanged with the same layer on a peer node
- The term "protocol" is overloaded
  - describes the *specification* of peer-to-peer interface
  - also refers to *module* that implements this interface

Interfaces



Service and Peer Interfaces

Protocol Graph



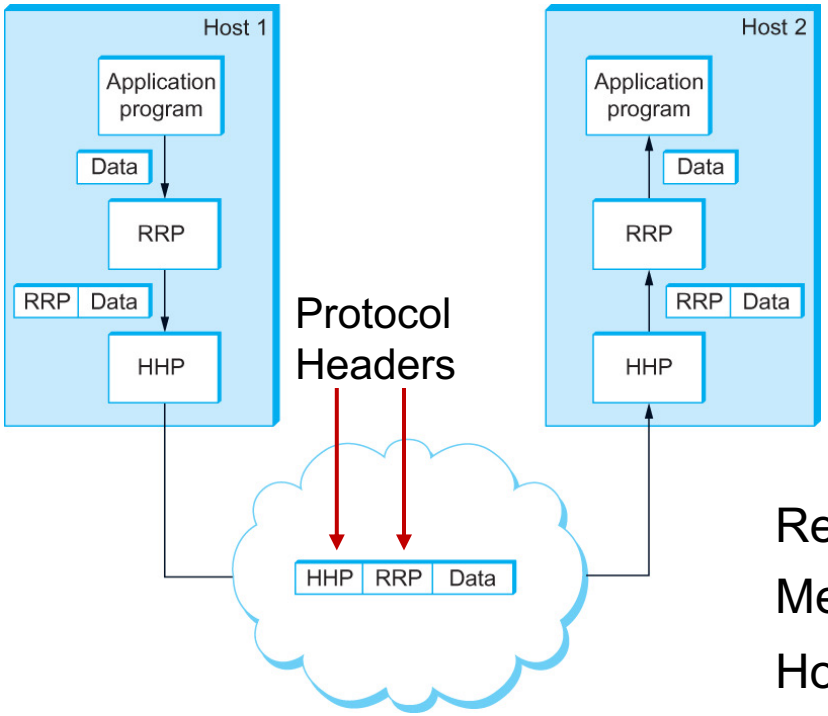
Request/Reply Protocol (RRP)

Message Stream Protocol (MSP)

Host to Host Protocol (HHP)

Example of a protocol graph: nodes are protocols, links show the “depends-on” relation

Encapsulation



Request/Reply Protocol (RRP)

Message Stream Protocol (MSP)

Host to Host Protocol (HHP)

High-level messages are encapsulated inside of low-level messages and protocols *add headers* with control/configuration information

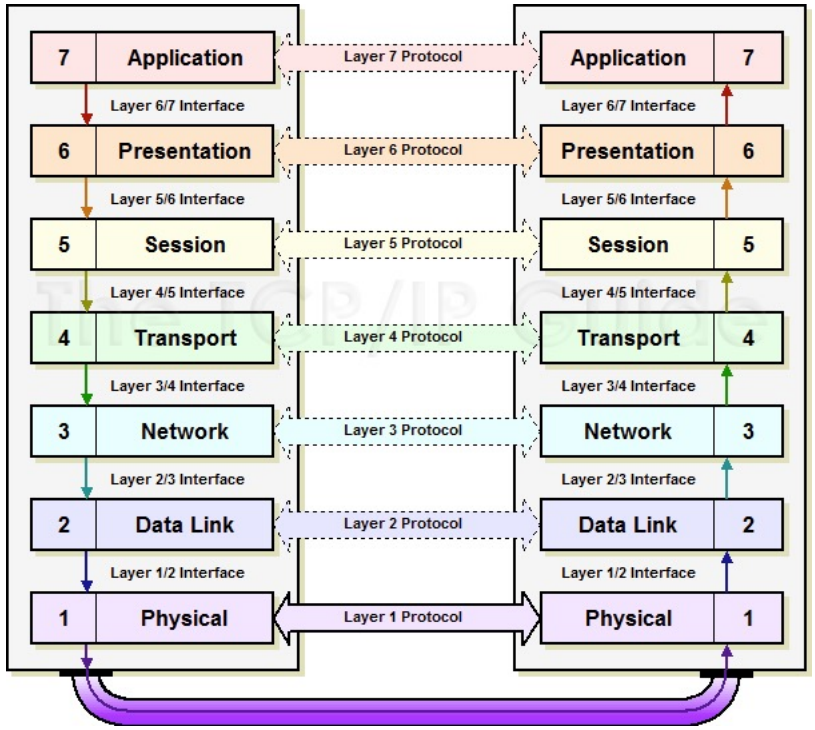
The OSI Model

A Layered Network Architecture:

- Networks are organized as a series of layers
  - Each layer builds on the services and capabilities of the layer below.
    - Reduces complexity and supports interoperability
- Layers are designed to handle specific functionalities
  - Link access control, routing, end-to-end connectivity
- Layers provide services to other layers
- The combination of layers and protocols forms the network architecture.

The OSI Model - Interfaces

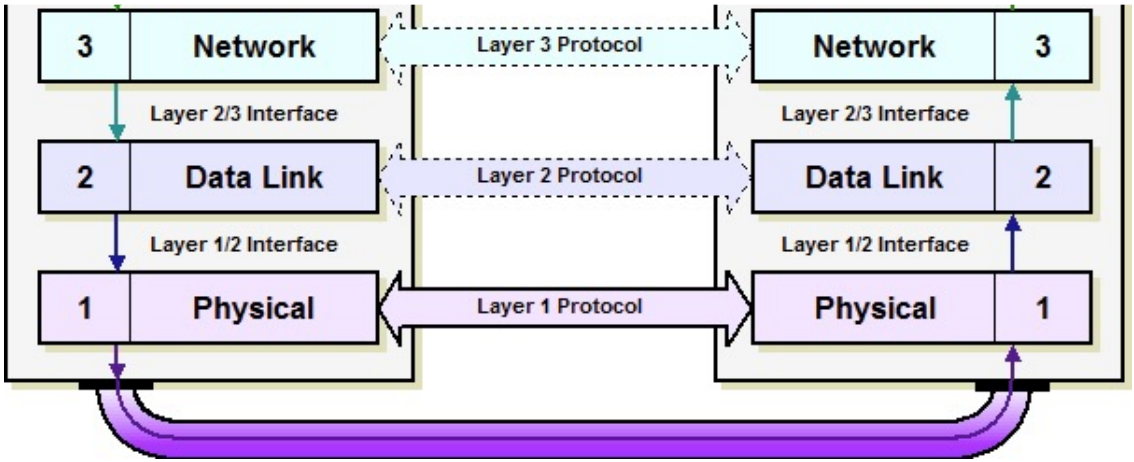
- Each layer interfaces with the layers above and below
- Each layer interfaces with the *peer layer* on the other node
- Layers exchange messages with their peers, using the layers below them for delivery



Description of Layers

- Physical Layer
  - Handles the transmission of raw bits over a communication link – wire, radio, optical, etc.
- Data Link Layer
  - Collects a stream of bits into a *frame*
  - Network adaptor, along with device driver in OS, implements the protocol in this layer
  - Frames are delivered directly from host to host
- Network Layer
  - Handles *routing* between nodes of a packet-switched network
  - Unit of data exchanged by nodes is called a *packet*

The OSI Model



Route packets

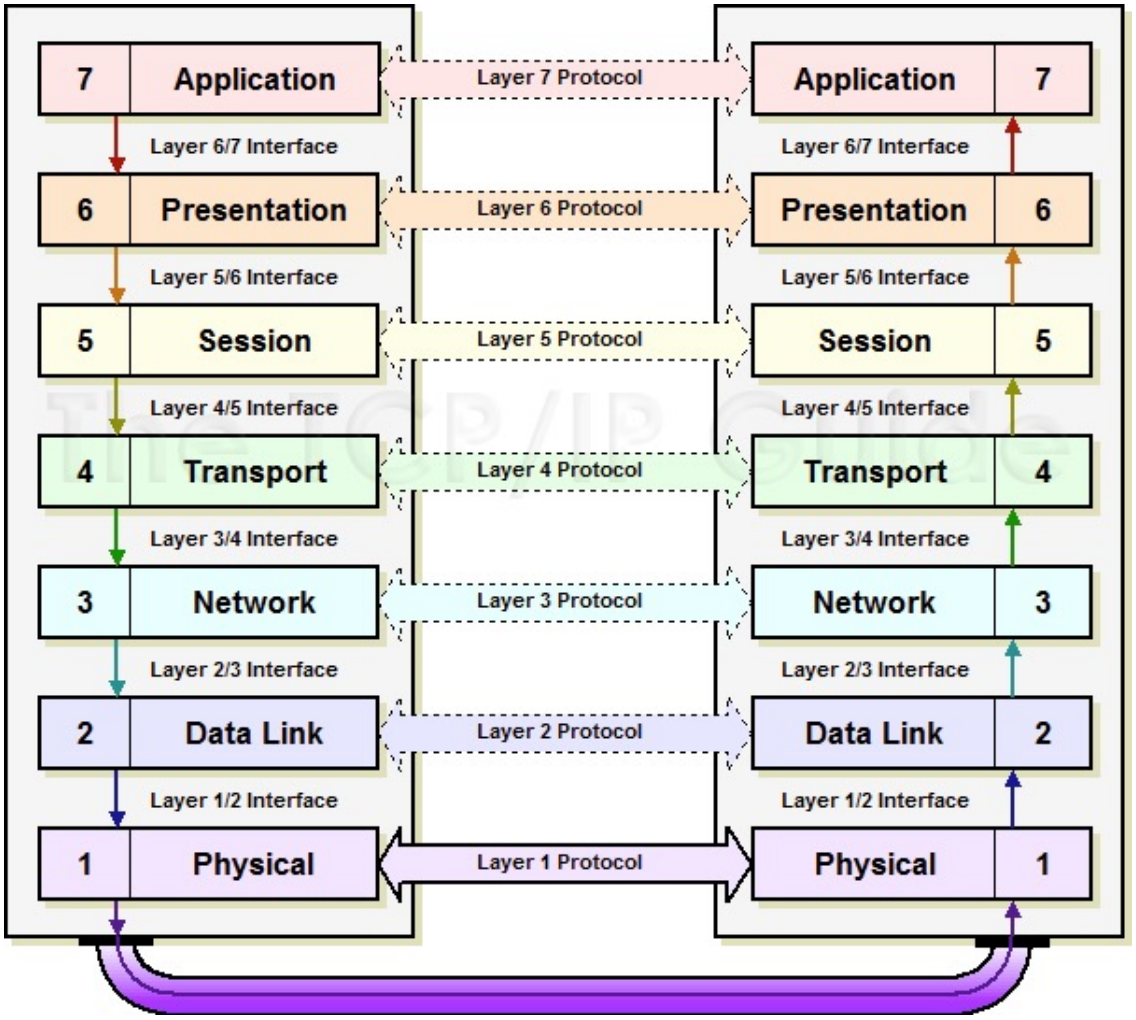
Control access to the medium

Transmit data across a medium

Description of Layers

- Transport Layer
  - Implements a process-to-process *channel*
  - The unit of data being exchanged is called a *message*
- Session Layer
  - Provides a name space used to tie together different transport streams that are part of a single application
- Presentation Layer
  - Concerned about the format of data exchanged between peers
- Application Layer
  - Standardize common type of exchanges

The OSI Model



Create and maintain a session to support an application

Reliable Delivery

Route messages

Control access to the medium

Transmit data across a medium

Connection-Oriented vs. Connectionless

- Service provided by a layer may be:
  - Connection-oriented, must be set up each time messages are sent (e.g., phone call)
  - Connectionless, messages are handled separately (e.g., postal delivery)

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Movie download
	Unreliable connection	Voice over IP
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Text messaging
	Request-reply	Database query

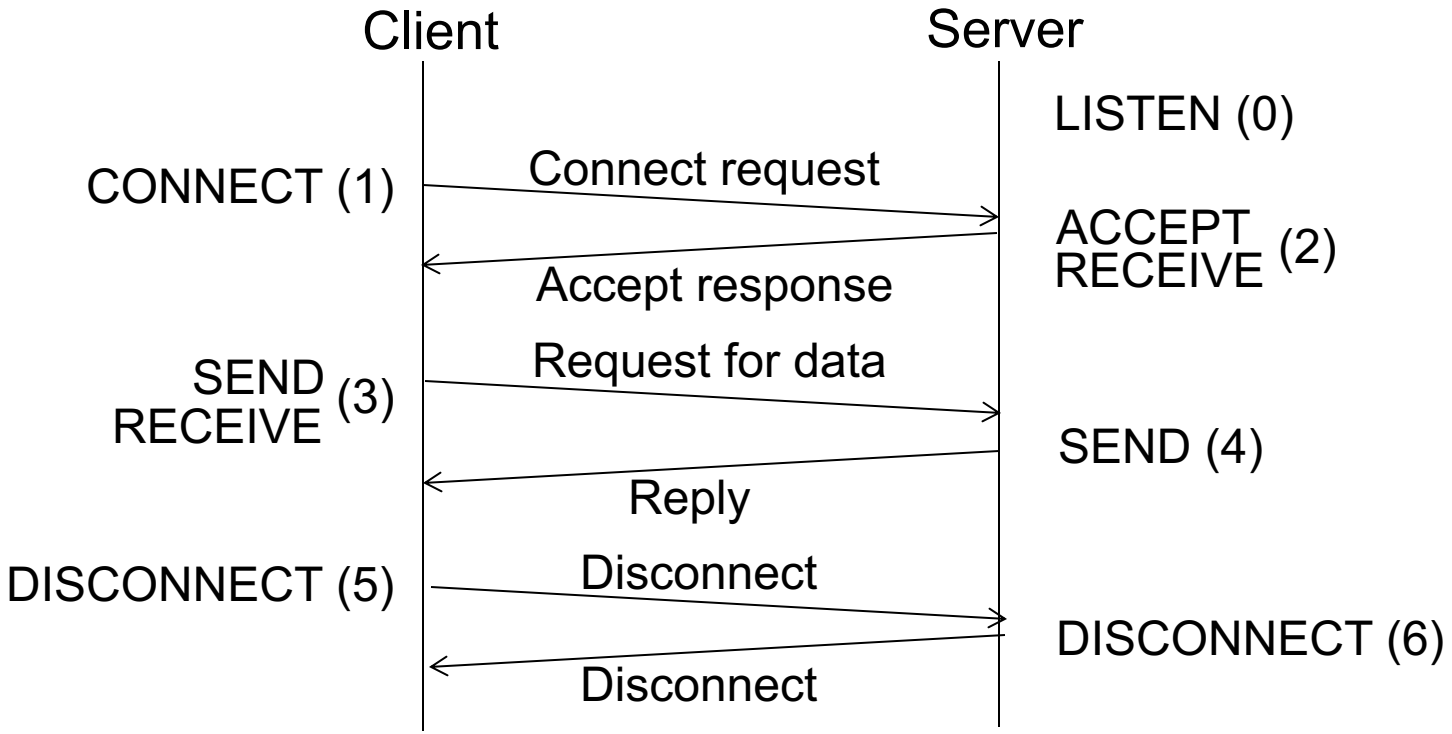
Service Primitives

- Services use a series of primitive commands
- Example of service primitives for a reliable byte stream (connection-oriented) service:

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Service Primitives

- Example of how these primitives may provide a *connection-oriented* client-server interaction

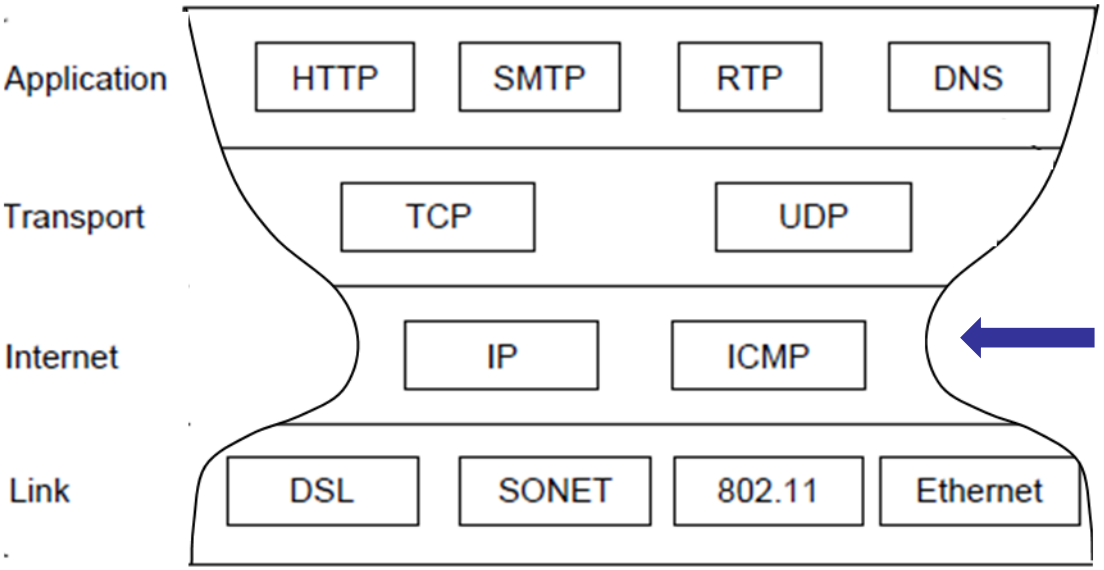


OSI vs. TCP/IP

- The OSI model is useful for ensuring that the design of a network protocol is complete
- However, the Application, Presentation and Session layers can be *merged into the application software*, eliminating the need to include those in the set of network protocols
  - this simplifies the network software and allows the designers of the application to customize those functions for their specific application
- The Internet uses the TCP/IP protocol instead

TCP/IP Reference Model

- The four layers of the TCP/IP reference model provide modularity, flexibility and reliability to support many different network applications



Communications in the Internet generally use the IP protocol as the Network layer

Note that the Physical Layer is not part of this model because neither the Internet Protocol nor the Transport Protocol interact with it directly.

Packets vs. Frames

- The terms packet and frame refer to the combination of data and headers at certain layers of the protocol stack
  - At the Network level, the term *packet* is used
    - this includes the combination of data and headers passed down by the higher layers, plus the header added by the network layer
  - At the Data-link level, the term *frame* is used
    - this includes the entire packet created at higher layers, plus the header & trailer added by this layer

Messages vs. Segments

- An entire file (image, document, etc.) to be transmitted is a *message*
  - However, some protocols can only manage a maximum number of bytes in each packet
    - for example: the IP protocol is limited to 64k bytes
  - The transport layer can divide the message into *segments* that are each within the packet's size limit and transmit them one after another, reassembling the message at the receiving end when all segments have arrived