Lecture 1:

Introduction to computer networks and TCP/IP protocol suite.

The term ‘‘computer network’’ means a collection of autonomous computers interconnected by a single technology. Two computers are said to be interconnected if they are able to exchange information. The connection need not be via a copper wire; fiber optics, microwaves, infrared, and communication satellites can also be used. Networks come in many sizes, shapes and forms. They are usually connected together to make larger networks, with the **Internet** being the most well-known example of a network of networks.

Computer networking continues to grow explosively. Since the 1970s, computer communication has changed from an esoteric research topic to an essential part of everyone's lives. Networking is used in every aspect of business, including advertising, production, shipping, planning, billing, and accounting. Consequently, most corporations have multiple networks. Federal, state, and local government offices rely on networks, as do military organizations. In short, computer networks are everywhere.

Although it functions as a single communications system, the Internet consists of parts that are owned and operated by individuals or organizations. To help clarify ownership and purpose, the networking industry uses the terms *public network* and *private network*.

A *public network* is run as a service that is available to subscribers. Any individual or corporation who pays the subscription fee can use the network. A company that offers communication service is known as a *service provider*. The concept of a service provider is quite broad, and extends beyond *Internet Service Providers* (*ISPs*). In fact, the terminology originated with companies that offered analog voice telephone service.

The term public *means a service is available to the general public; data transferred across a public network is not revealed to outsiders.*

A *private network* is controlled by one particular group. Although it may seem straightforward, the distinction between public and private parts of the Internet can be subtle because control does not always imply ownership. For example, if a company leases a data circuit from a provider and then restricts use of the circuit to company traffic, the circuit becomes part of the company's private network. The point is: A network is said to be private if use of the network is restricted to one group. A private network can include circuits leased from a service provider.

PROTOCOLS AND LAYERING:

Communication always involves at least two entities, one that sends information and another that receives it. In fact, we will see that most packet switching communications systems contain intermediate entities (i.e., devices that forward packets). The important point to note is that for communication to be successful, all entities in a network must agree on how information will be represented and communicated. Communication agreements involve many details. For example, when two entities communicate over a wired network, both sides must agree on the voltages to be used, the exact way that electrical signals are used to represent data, procedures used to initiate and conduct communication, and the format of messages.

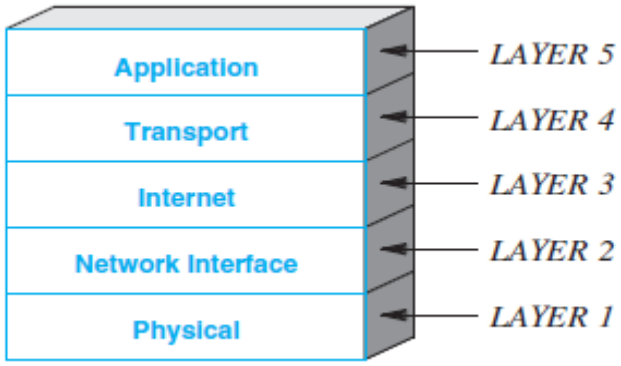
The term *interoperability* refers to the ability of two entities to communicate, and say that if two entities can communicate without any misunderstandings, they *interoperate* correctly. To ensure

that all communicating parties agree on details and follow the same set of rules, an exact set of specifications is written down.
The term *communication protocol*, *network protocol*, or *protocol* to refer to a specification for network communication. A given protocol may specify low-level details, such as the type of radio transmission used in a wireless network, or describe a high-level mechanism such as the messages that two application programs exchange.

Therefore, a communication protocol specifies the details for one aspect of computer communication, including actions to be taken when errors or unexpected situations arise. A given protocol can specify low-level details, such as the voltage and signals to be used, or high-level items, such as the format of messages that application programs exchange.

A set of protocols must be constructed carefully to ensure that the resulting communications system is both complete and efficient. To avoid duplication of effort, each protocol should handle a part of communication not handled by other protocols. How can one guarantee that protocols will work well together? The answer lies in an overall design plan: instead of creating each protocol in isolation, protocols are designed in complete, cooperative sets called suites or families. Each protocol in a suite handles one aspect of communication; together, the protocols in a suite cover all aspects of communication, including hardware failures and other exceptional conditions. Furthermore, the entire suite is designed to allow the protocols to work together efficiently. The fundamental abstraction used to collect protocols into a unified whole is known as a layering model. In essence, a layering model describes how all aspects of a communication problem can be partitioned into pieces that work together. Each piece is known as a layer; the terminology arises because protocols in a suite are organized into a linear sequence. Dividing protocols into layers helps both protocol designers and implementors manage the complexity by allowing them to concentrate on one aspect of communication at a given time.

TCP/IP protocol stack

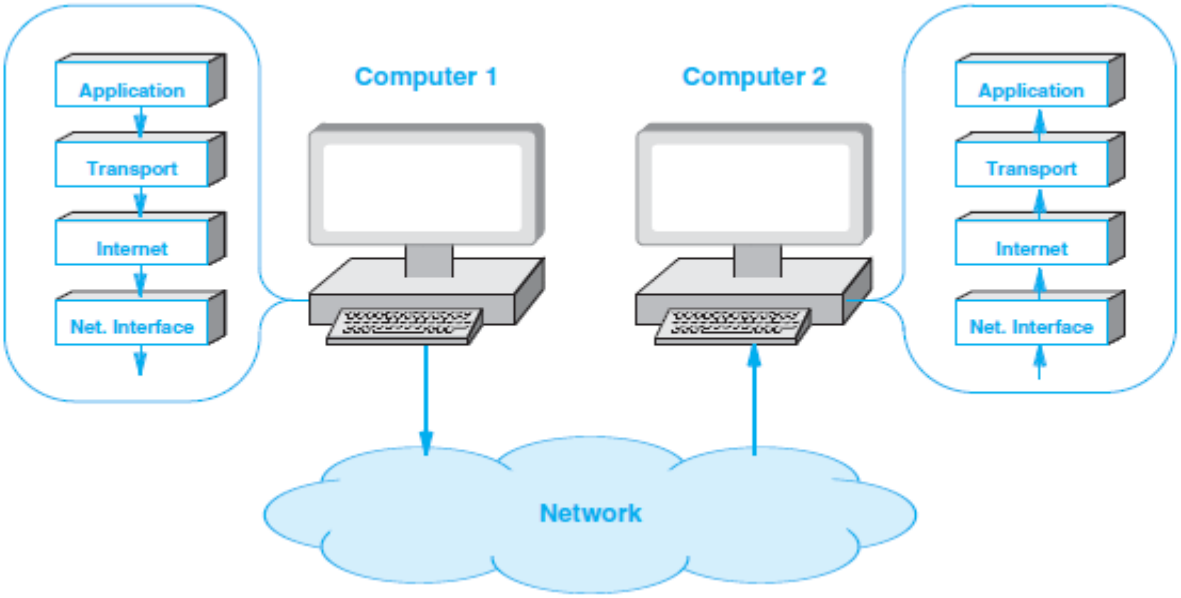


Layer 1: Physical
Protocols in the *Physical* layer specify details about the underlying transmission medium and the associated hardware. All specifications related to electrical properties, radio frequencies, and signals belong in layer 1.

Layer 2: Network Interface or MAC
Protocols in the *MAC* layer specify details about communication over a single network and the interface between the network hardware and layer 3, which is usually implemented in software. Specifications about network addresses and the maximum packet size that a network can support, protocols used to access the underlying medium, and hardware addressing belong in layer 2.
Layer 3: Internet
Protocols in the *Internet* layer form the fundamental basis for the Internet. Layer 3 protocols specify communication between two computers across the Internet (i.e., across multiple interconnected networks). The Internet addressing structure, the format of Internet packets, the method for dividing a large Internet packet into smaller packets for transmission, and mechanisms for reporting errors belong in layer 3.
Layer 4: Transport
Protocols in the *Transport* layer provide for communication from an application program on one computer to an application program on another. Specifications that control the maximum rate a receiver can accept data, mechanisms to avoid network congestion, and techniques to ensure that all data is received in the correct order belong in layer 4.
Layer 5: Application
Protocols in the top layer of the TCP/IP stack specify how a pair of applications interact when they communicate. Layer 5 protocols specify details about the format and meaning of messages that applications can exchange as well as procedures to be followed during communication. In essence, when a programmer builds an application that communicates across a network, the programmer devises a layer 5 protocol. Specifications for email exchange, file transfer, web browsing, voice telephone service, smart phone apps, and video teleconferencing belong in layer 5.

HOW DATA IS PASSED BETWEEN LAYERS

Protocol implementations follow the layering model by passing the output from a protocol in one layer to the input of a protocol in the next layer. Furthermore, to achieve efficiency, rather than copy an entire packet, a pair of protocols in adjacent layers pass a pointer to the packet. Thus, data passes between layers efficiently.



The figure above illustrates layered protocols on the two computers.

When an application sends data, the data is placed in a packet, and the outgoing packet passes down through each layer of protocols. Once it has passed through all layers of protocols on the sending computer, the packet leaves the computer and is transmitted across the underlying physical network†. When it reaches the receiving computer, the packet passes up through the layers of protocols. If the application on the receiving computer sends a response, the process is reversed. That is, a response passes down through the layers on its way out, and up through the layers on the computer that receives the response.