GOALS & APPLICATIONS OF COMPUTER NETWORKS

Computer Networks, which are an interconnected collection of autonomous computers, have replaced the old model of a single central computer serving all the needs of an organization or a community of users. In the context of Computer Networks (or simple, Networks) being interconnected implies ability to exchange information by identification of the sender and receiver. The users of a Network are aware of the existence of multiple computers, and may move data and files between computers using a network interface (not processor bus). Theses characteristics of Networks distinguish them from Multiprocessor or Distributed systems, where the existence of multiple computers or processors is transparent (i.e. not visible) to the user and the Operating System and Software hosted by a master computer delegates the user’s tasks to slave computers. Thus a computer network ties together co-operating computers in a peer-to-peer relationship (not master-slave) so as to harness their collective capability. In this context the WWW (including browser software, search-engine and their HTML content) is an example of a distributed system which uses the services of an underlying collection of networks i.e. the Internet. The WWW by itself is not a Computer Network.

SPECFIFIC GOALS AND APPLICATIONS OF NETWORKS

1. Resource Sharing – Remote access (online and offline) to data repositories e.g. Web Servers, Data Mines.
2. Increased Reliability/Redundancy – Access to multiple data sources with hot-swap capability.
3. Cost-Effectiveness – Smaller computers have higher Performance/Price ratio than large mainframe computers. Hence a network of PCs in a client-server configuration represents a more economical solution.
4. Effective and Economic Communication – Networks provide a powerful medium, for communication of Text (Web, e-mail), Voice (Internet Telephony) and Video (Video Conferencing) across large distances and between remotely collaborating individuals.
5. Facilitate Electronic Commerce – Business-to-Business and Business-to-Consumer transactions using the Internet.

CLASSIFICATION OF NETWORKS

In terms of size or area covered, the combination of transmission technologies used and the topology, Networks can be broadly classified into LANs, MANs and WANs.

LANs are privately owned networks that are limited to a building or a small campus not exceeding a few kilometers. LANs often use broadcast technologies and require upper bounds on worst-case transmission time which must be known in advance. LAN topology may be of linear cable (or bus) type e.g. an IEEE 802.3 (aka Ethernet) network, or, it may be a ring-shaped network e.g. the IEEE 802.5 (aka IBM Token Ring). LAN speeds range from 10 to 100 Mbps when using older technologies such as traditional Ethernet. New Gigabit Ethernet techniques support rates up to 10 Gbps. IBM Token Ring operates at up to 16 Mbps. Recently optical-fiber based FDDI (ring-shaped) LANs have also become popular.

A MAN provides services to a large number of users located close together e.g. in a city. MANs have evolved from Cable-TV networks serving a metropolitan area, to providers of multimedia services, including Internet/Phone services, via cable or broadband wireless (IEEE 802.16) networks.

A WAN spans a large geographical area and connects host computers scattered across cities, countries or even continents. A WAN provides interconnectivity between hosts by using a subnet, consisting of routers and transmission lines. The host computers are generally connected to LANs which interface with the WAN Subnet by connecting to one of its routers.(see fig 1-9,1-10 -\*-). Most WAN subnets are packet-switched networks which fragment the original message into packets, transmit the packets separately across the subnet to their final destination, and, reassemble the packets at the receiving host.

OSI MODEL (developed by ISO)

The OSI Model is a framework for a computer communications architecture and for developing protocol standards. In order to reduce design complexity most networks are organized as a stack of layers, each layer offering well-defined services to the higher layer and shielding it from the implementation details of the lower layer. Each layer uses a set of rules and conventions or a protocol to communicate with its counterpart on a remote host. The list of protocols used by a system, one protocol per layer, is called a protocol stack.

Between each pair of adjacent layers is an interface which defines a set of primitive operations and services provided by the lower layer for the use of the upper layer. See fig 1-13 -\*-.

The OSI Model identifies 7 layers and the functions associated with each layer so that new protocols may be developed to perform the functions of each layer. Note that in this layered model no data is directly transferred from layer n on one machine to layer n on another. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached, at which point the raw data/control information is transmitted through the physical medium.

LAYER DESIGN ISSUES

The following broad set of capabilities is present across all layers in order to ensure proper functioning of the protocol stack:

1. An addressing mechanism capable of identifying multiple senders and receivers.
2. Support for at least 2 logical channels per connection (urgent/normal data).
3. Error Control and Detection.
4. Flow Control – agreement on transmission rates.
5. Multiplexing/Demultiplexing of communicating processes.

SERVICES, INTERFACES, PROTOCOLS

The OSI Model is built around 3 fundamental concepts, which need to be explicitly distinguished:

1. A Service is a set of primitives or operations that a layer provides to the one above it. It defines what the layer does, but says nothing about how these operations are implemented or how the upper layers access it.
2. An Interface specifies the manner in which an upper layer may use the service provided by this one; what parameters are to be given and what results to expect. No implementations details are discussed.
3. A Protocol is a set of rules governing the format and meaning of the packets, or messages, that are exchanged by the peer entities within a layer. Since entities use protocols to implement their service definitions, the protocols may be changed or upgraded freely, as long as the service visible to the users remains unchanged.

OSI Reference Model (Open Systems Interconnection)

The OSI Model was proposed by ISO (International Standards Organization) as an initial step towards international standardization of the protocols used by computer networks. Although its protocols are not used in practice, the Model serves as a reference. The seven layers of the Model were identified using universally accepted design principles (see pg 38 1…5 -\*-)

Physical Layer: It deals with the transmission of raw bits over a communication channel; issues related to electrical and optical characteristics of the transmission medium; timing interfaces required for transmission of an unstructured bit-stream, as well as physical (pin-out) connections used at network end-points.

Datalink Layer: This layer transforms a raw data transmission channel into a reliable one that is relatively free of undetected errors. This is done by organizing data bits into frames whose integrity can be checked, and then transmitting the frames sequentially using necessary synchronization, error-correction and flow-control. This layer also deals with issues related to controlling access to a shared broadcast channel.

Network Layer: This layer provides a simplified mechanism to the upper layers for routing packets from source to destination within a WAN subnet. Packet-level addressing and routing mechanisms, as well as packet-level congestion control are built into this layer. Heterogeneous networks (different packet-sizes, addressing schemes) are interconnected at this level.

Transport Layer: Serves as the first end-to-end layer and establishes communication between source and destination processes (running often on remote machines). Data accepted from upper layers is fragmented into packets, handled over to Network layer for transmission. The type of end-to-end communication may be:

1. A point-to-point error-free channel which delivers bytes in the order sent,
2. Capability to send isolated packets with no guarantee of ordered delivery, or, any delivery at all.
3. Capability to broadcast a packet or message to multiple destinations.

Session Layer: Provides a framework which enables remotely co-operating applications to establish, manage and terminate sessions between them with the help of various session-related services such as token management and synchronization.

Presentation Layer: This layer enables development of network-applications which are independent of the differences in data representation (32 bit versus 64 bit, ASCII versus EBCDIC, Little-endian versus Big-endian) used by the host computers. Universal data representations formats e.g. XDR are used at this level to represent data structures in an abstract way. The convention, for the sequence in which the bytes of a multi-byte word are transmitted over a link, is also established at this layer.

Application Layer: This layer includes a variety of protocols which enable exchange of files or data between hosts. Examples include File Transfer Protocol (FTP), TELNET (Virtual terminal for remotely logging into a host), HTTP (Hyper Text Transfer Protocol, used for retrieving HTML pages from a Web Server).

The TCP/IP Reference Model

This model evolved from the ARPANET, a research network used for defense-related communication. The main goal of ARPANET was to connect heterogeneous networks in a seamless manner, such that the loss of some intermediate nodes, would not halt host-to-host communication. This led to the concept of a packet-switched WAN subnet, an architecture later referred to as the TCP/IP Reference model. This model has 4 layers (beginning from bottom layer)see fig 1-21 -\*-:

Host-to-Network Layer: This is the lowest layer. The model simply specifies that the host must connect to the network using some protocol (which may vary with the host and network) so that it can send IP packets to the network. In essence, this layer of the model captures the equivalent of the Physical and Datalink layers in the OSI model, without actually distinguishing or mentioning them.

Internet Layer: This layer is similar to the OSI network layer. Its main function is to permit hosts to inject packets into any network and have them travel independently to their destination on the target networks. The layer defines the format of these packets as IP packet format and an associated protocol as the Internet Protocol (IP). Like the OSI network layer, this layer deals with packet routing and congestion control. Only connection-less mode of communication is supported.

Transport Layer: This layer performs the same function as in the OSI Model, i.e. to enable peer entities on the source and destination hosts to establish and maintain communication. This layer supports two types of end-to-end connections:

1. TCP (Transmission Control Protocol):

A reliable, connection-oriented protocol that allows a byte-stream originating on one machine to be delivered without error on another machine running the TCP/IP protocol stack. TCP fragments and reassembles the source byte-stream in a manner invisible to the sender or receiver.

1. UDP (User Datagram Protocol):

An unreliable connection-less protocol suitable for network applications that do not require the sending of a perfectly sequenced stream of bytes, or, can handle delayed delivery of data. UDP is ideal for sending data packets of application defined format and size, wherein the receiving application maintains sequencing details at the receiving end, and sending application retransmits undelivered packets. Request-Reply type of queries in Client-Server systems are a good example.

Application Layer: The TCP/IP Reference Model includes the Application Layer directly on top of the Transport Layer and includes high level protocols such as TELNET (Remote Login), FTP (File Transfer), SMTP (E-mail), DNS, HTTP. Note that the TCP/IP Model has been implemented on top of several Host-to-Network interfaces and technologies, other than Ethernet (see fig 1-22 -\*-).

For most practical discussion of computer networks, a hybrid model, which combines specific layers from both reference models, is used.

APPLICATION

TRANSPORT

NETWORK (IP)

DATALINK

PHYSICAL

EXAMPLES OF EARLIER AND EXISTING NETWORKS

Early networks include ARPANET (Advanced Research Projects Agency Network) launched by US DOD; NSFNET (of the National Science Foundation). The merger of these two networks evolved into the current Internet after TCP/IP was adopted as the official protocol in 1983. Other well known networks include X.25 (Connection-oriented network offering data services) which was launched by telephone companies in 1970s. SNA from ICM is another example.

INTERNETWORKING CONCEPT

A collection of interconnected networks is called an internetwork or an internet. In its simplest form, an internet is a collection of Ethernet LANs connected together by a WAN, which uses the Internet Protocol (IP) to route packets. (see fig 1.9 for this simple case -\*-).

In its most complex form, an internet is a collection of dissimilar networks, which may use different protocols at the various layers, but are glued together by IP.

To be on the Internet, a host must run the TCP/IP protocol stack, have a permanent IP address and be able to send IP packets to all other machines on the Internet. In case of hosts that do not have a permanent IP address but are dynamically assigned one (e.g. your home PC dialing up an ISP using a modem), the host is on the Internet as long as it is connected to the ISP’s router.

The areas in which networks may be dissimilar can be several. E.g. different modulation techniques or frame formats may be used in the physical and datalink layers. However, the differences that impact internetworking the most are those that occur in the Network layer such as differing packet sizes, addressing schemes, error and flow control techniques.

INTERNETWORKING BUILDING BLOCKS

The fundamental building blocks used by the internet include (but are not limited to) the following:

Repeaters and Hubs: Operating at the Physical layer, these are low-level analog devices that simply amplify or regenerate weak signals thereby moving bits from one LAN segment to another identical network. For example, Ethernet (IEEE 802.3) requires repeaters every 500 meters.

Bridges and Switches: These devices operate at the datalink layer and mainly forward frames to another network (LAN) which may use a different technology. Minor protocol translation such as Ethernet to FDDI or 802.5 or 802.11 may be done in this process.

Routers: These devices provide interconnection at the network layer, between possibly different networks. If the network layers being interconnected are dissimilar, a specialized router called Multiprotocol Router is used to translate between the packet formats. Most often, Routers forward IP-format packets.

Transport Gateways: They are used to interface between two Transport layer connections by connecting byte streams. E.g. a Transport Gateway could allow packets to flow between a TCP network and a SNA network by gluing a TCP connection to a SNA one. Concatenated VCs (Virtual Circuits) use Transport Gateways.

Application Gateways: They translate the semantics of application-specific messages at the Application layer. For example, an Application Gateway may convert e-mail messages from an Internet-Style format to an X-400 (Directory-Style) format by modifying the header fields. Similarly it may take specific action relative to FTP, TELNET and DNS application semantics. For a good example illustrating the differences between switching at the datalink layer versus routing at the network layer, see fig 5-44 and associated text. -\*-

CONNECTION-ORIENTED (CO) & CONNECTION-LESS (CL) APPROACHES TO INTERNETWORKING

The CO approach uses a concatenation of virtual –circuits which may be embedded within different WAN subnets that support VCs. Such WAN subnets could infact use different WAN technologies e.g. Frame Relay, ATM or X.25, each of which provides a CO mode. In the CO approach, a concatenation of VCs is formed by successive establishment of connections, beginning with the host on which the request originates, and, traversing through a series of Routers (or Gateways, or, MP Routers) until the destination host is reached. Once the connection is established, data-packets (or, frames, or, cells, or, bytes) flow along the VC to their destination in the order transmitted, but may undergo format translation as they traverse dissimilar network technologies
(very rare). The requirement for CO approach is that all intervening subnets must handle VCs and provide reliable service. The advantage is that this approach requires smaller headers and can guarantee Quality of Service (QOS), especially when used for Networking of Multimedia Applications (voice, video). Note that a VC may enable a byte-stream oriented flow of data or, a packet-sequence type of data flow where the notion of a “frame” is retained across transmitted bytes. (See fig 5-45 -\*-).

The CL approach is based on the datagram model, wherein the data packets (aka datagrams) originating at the host are forwarded to their destination, possibly via different sets of intervening routers. The exact sequence of routers over which each individual packet hops to its destination is determined by each router, one hop at a time, based on the total traffic at that time. The availability of multiple, parallel routers may enable higher data rate and provide redundancy if a specific route is unavailable. But no guarantee of arrival or, order of arrival is provided. Like the CO approach, the packets may traverse WAN subnets that use dissimilar network layers (e.g. SNA and IP). But this is rare and generally problematic even with the use of MP Routers because of differences in packet formats and addressing schemes. The disadvantage of the CL approach is that it requires long packet headers. (See fig 5-46 -\*-).