Chapter 1

1.1 Architectural Principles

- TCP/IP protocol suite allows embedded devices of all sizes to communicate with each other
- Internet architecture
  - develop an effective technique for multiplexed utilization of existing interconnected networks

1.1.1 Packets, Connections, and Datagrams

- Telephone call - call was implemented by establishing a connection from on party to another for the duration of the call - a circuit
    - call duration and identification of the connection endpoints
- Packets - chunks of digital information comprising some number of bytes
- Multiplexing - packets can be pulled apart and mixed together from different senders
- Packet switching - chunks can be moved from one switch to another
- When packets are received at a switch processed in FIFO manner
- Virtual circuits - exhibit many of the behaviors of circuits but do not depend on physical circuit switches can be implemented atop connection-oriented packets
- Datagram - is a special type of packet in which all the identifying information of the source and final destination resides inside the packet itself
- Message boundaries or record markers - when an application sends more than one chunk of information into the network, the fact that more than one chunk was written may or may not be preserved by the communication protocol
  - if the application requires boundaries and the protocol fails to preserve them the application must provide its own

1.1.1 The End-to-End Argument and Fate Sharing

- "The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication itself is not possible.
- Efforts to implement what the application is "likely" to need are doomed to incompleteness
  - In short, this principle argues that important functions (error control, encryption, delivery and acknowledgment) should usually not be implemented at low levels
  - Low levels may provide capabilities that make the job of endpoints somewhat easier
- supports a design with a "dumb" network and "smart" systems connected to the network
  - Seen in TCP/IP where many functions are implemented in the end hosts where the applications reside
- Fate sharing - suggests placing all the necessary state to maintain an active communication association at the same location with the communication end points.
  - the only type of failure that destroys communication is one that also destroys one or more of the endpoints, which obviously destroys the overall communication
  - One of the design philosophies that allows virtual connections to remain active even if connectivity within the network has failed for a period of time

1.1.3 Error Control and Flow Control

- error control - can be implemented in the systems constituting the network infrastructure, or in the systems that attach to the network, some combination,
- if a small number of bit errors are of concern, a number of mathematical codes can be used to detect and repair the bit errors when data is received or while it is in transit

  - when more severe damage occurs in a packet network, entire packets are usually resent or retransmitted
  - best-effort delivery - an alternative to expensive within network in order delivery
    - that network does not expend much effort to ensure that data is delivered without errors or gaps
    - if successful, a fast sender can produce information at a rate that exceeds the receiver’s ability to consume it
- **1.2 Design and Implementation**
  - make a distinction between the protocol architecture and the implementation architecture
  - design philosophy for networking protocols involving multiple layers of implementation (and design). This approach is now called layering and is the usual approach to implementing protocol suites
- **1.2.1 Layering**
  - Each layer is responsible for a different facet of the communications
  - Open Systems Interconnection (OSI) -
    - most frequently mentioned concept of protocol layering
      - suggests that seven layers may be desirable for modularity of a protocol architecture implementation
  - Layers (bottom up)
    - 1 - Physical - defines methods for moving digital information across communication medium such as a phone line or fiber-optic cable
    - 2 - Link or data-link -
      - includes those protocols and methods for establishing connectivity to a neighbor sharing the medium
        - some link-layer networks connect only two neighbors
        - when more than one neighbor can access the same shared network, the network is said to be a multi-access network
          - examples - wi-fi and ethernet
    - 3 - Network or internetwork layer -
      - for packet networks such as TCP/IP it provides an interoperable packet format that can use different types of link-layer networks for connectivity
      - The layer also includes an addressing scheme for hosts and routing algorithms that choose where packets go when sent from one machine to another
    - 4 - Transport
      - provides a flow of data between sessions and can be quite complex
    - 5 - Session
      - sessions represent ongoing interactions between applications
      - session-layer protocols may provide capabilities such as connection initiation and restart, plus checkpointing (saving work that has been accomplished so far)
    - 6 - Presentation
      - responsible for format conversions and standard encodings for information
      - internet protocols do not include a formal session or presentation protocol layer, so these functions are implemented by applications if needed
    - 7 - Application
      - applications usually implement their own application-layer protocols, these are the ones most visible to users
  - TCP/IP
    - normally considered to consist of five layers

- ### 1.2.2 Multiplexing, Demultiplexing, and Encapsulation in Layered Implementations
  - one of major benefits of layer architecture is its natural ability to perform protocol multiplexing
  - protocol multiplexing - allows multiple different protocols to coexist on the same infrastructure
    - allows multiple instantiations of the same protocol object (e.g., connections) to be used simultaneously without being confused
  - Multiplexing can occur at different layer, and at each layer a different sort of identifier is used for determine which protocol or stream of information belongs together
  - Protocol identifier field - at the link layer a value in each packet to indicate which protocol is being carried in the link-layer frame
  - Protocol data unit (PDU) - an object at one layer is carried by a lower layer, it is said to be encapsulated by the next layer down
    - Each layer has it's own concept of a message object (PDU)
    - Each layer treats the data from above as opaque, interpretable information
  - Most commonly a layer prepends the PDU with its own header
    - the header is used for multiplexing data when sending, and for the receiver to perform demultiplexing based on a demultiplexing (demux) identifier.
  - TCP/IP networks such identifiers are commonly hardware addresses, IP addresses, and port numbers
  - Pure Layering
    - not all networked devices need to implement all the layers
    - a device needs to implement only a few layers if it is expected to perform certain types of processing
  - Small internet (idealized)
    - two end systems
      - hosts on the left and right
    - intermediate systems
      - a switch and router
      - each device implements a different subset of the layer stack
    - The host on the left implements three different link-layer protocols with corresponding physical layers and three different transport layer protocols that run on a single network-layer protocol
    - End hosts implement all the layers
      - switches implement up to layer 2
      - routers implement up to layer 3
    - Routers are capable of interconnecting different types of link-layer networks and must implement the link-layer protocols for each of the network types they interconnect
  - Today switches and routers implement more than protocols they are absolutely required to implement for forwarding data
    - Number of reasons including management. In such cases devices as routers and switches must sometimes act as hosts and support services such as remote login
      - to do this usually must implement transport and application protocols
  - Layers above the network layer use end-to-end protocols
  - Network layer - provides a hop-by-hop protocol and is used on the two end systems and every intermediate system

  - The switch or bridge is not ordinarily considered an intermediate system because it is not addressed using the internet-working protocols addressing format, and it operates in a fashion that is largely transparent to the network layer protocol
    - from the point of view of routers and end systems, the switch or bridge is essentially invisible
  - Multihomed - any system with multiple interfaces
    - a router by definition, has two or more network interfaces
    - a host can also be multihomed, but unless it specifically forwards packets from one interface to another, it is not called a router
  - Most TCP/IP implementations allow a multihomed host to act as a router also, if properly configured to do so
    - use the term host or router given the context
    - host - FTP or the web is used
    - router - forwarding packets from one network to another
  - One of the goals of an internet is to hide all of the details of the physical layout and lower-layer protocol heterogeneity from the applications
- **1.3 The Architecture and Protocols of the TCP/IP Suite**
  - TCP/IP established term for protocols on the internet
- **1.3.1 The ARPANET Reference Model**
  - Its layering formed the basis for the Internet protocol layering
  - layers bottom up
    - 2.5 Link (Adjunct) - several protocols operate here, but one of the oldest and most important is called the Address Resolution Protocol (ARP).
      - ARP - is a specialized protocol used with IPv4 and only with multi-access link-layer protocols (such as Ethernet and Wi-Fi) to convert between the addresses used by the IP layer and the addresses used by the link layer
    - 3 - Network -
      - IP - the main network-layer protocol for the TCP/IP suite
        - PDU that IP sends link-layer protocols is called an IP datagram
          - may be as large as 64KB
          - use packet to mean an IP datagram when the usage context is clear
        - fragmentation - fitting large packets into link layer PDUs that may be smaller is handled by a function called fragmentation
          - portions of a larger datagram are sent in multiple smaller datagrams called fragments and put back together (called reassembly) when reaching the destination
        - IP addresses - contains the address of the layer 3 sender and recipient
          - IPv4 is 32 bits long
          - IPv6 is 128 bits long
        - Forwarding - the process of making this determination and sending the datagram to its next hop. Both routers and hosts perform forwarding, although routers tend to do it more. Three types of forwarding
          - 1 - unicast
            - destined for a single host
          - 2 - broadcast
            - destined for all hosts on a given network
          - 3 - multicast
            - destined for a set of hosts that belong to a multicast group
    - 3.5 - Network (Adjunct) -

  - The Internet Control Message Protocol (ICMP) is an adjunct to IP
    - Two versions ICMPPv4 and ICMPv6 used with IPv4 and IPv6
    - It is used by the IP layer to exchange error messages and other vital information with the IP layer in another host or router
    - It is possible for applications to use it
    - ICMP messages are encapsulated within IP datagrams in the same way transport layer PDUs are
    - ping and traceroute use ICMP
  - The Internet Group Management Protocol (IGMP) is another protocol adjunct to IPv4
    - It is used with multicast addressing and delivery to manage which hosts are members of a multicast group
- 4 - Transport -
  - Transmission Control Protocol (TCP), deals with problems such as packet loss, duplication, and reordering that are not repaired by the IP layer
    - operates in a connection-oriented (VC) fashion and does not preserve message boundaries
    - provides a reliable flow of data between two hosts
    - concerned with
      - things such as dividing the data passed to it from the application appropriately sized chunks for the network layer below
      - acknowledging received packets
      - setting timeouts to make certain the other end acknowledges packets that are sent
    - TCP PDU is TCP segment
  - User Datagram Protocol (UDP) -
    - provides little more than the features provided by IP.
    - UDP allows applications to send datagrams that preserve message boundaries but imposes no rate control or error control
    - provides a much simpler service to the application layer
    - allows datagrams to be sent from one host to another, but there is no guarantee that the datagrams reach the other end
    - provides
      - set of port numbers for multiplexing and demultiplexing data,
      - data integrity checksum
  - Datagram Congestion Control Protocol (DCCP)
    - midway between TCP and UDP
    - connection-oriented exchange of unreliable datagrams but with congestion control
    - congestion-control
      - comprises a number of techniques whereby a sender is limited to a sending rate in order to avoid overwhelming the network
  - Stream Control Transmission Protocol (SCTP)
    - provides reliable delivery like TCP but does not require the sequencing of data to be strictly maintained
    - It also allows for multiple streams to logically be carried on the same connection and provides a message abstraction, which differs from TCP
    - was designed for carrying signaling messages on IP networks that resemble those used in the telephone network
- 7 - Application -
  - handles the details of the particular application

        - is concerned with the details of the application and not with the movement of data across the network.
    - The lower three layers know nothing about the application but handle all the communication details

- **1.3.2 Multiplexing, Demultiplexing, and Encapsulation in TCP/IP**
    - each layer there is an identifier that allows a receiving system to determine which protocol or data stream belongs together
    - usually there is addressing information at each layer, this information is used to ensure that a PDU has been delivered to the right place
    - Figure description - TCP/IP stack uses a combination of addressing information and protocol demultiplexing identifiers to determine if a datagram has been received correctly and, if so, what entity should process it. Several layers also check numeric values to ensure that the contents have not been damaged in transit
        - Arriving ethernet frame contains 48-bit destination address and 16-bit field called the Ethernet type
        - assuming that the destination address matches one of the receiving system's addresses, the frame is received and checked for errors
        - the ethernet type is used to select the network-layer protocol that should process it
        - assuming that the received frame contains an IP datagram, the Ethernet header and trailer information is removed, and the remaining bytes are given to IP for processing
            - IP checks a number of things including the destination IP address in the datagram
            - If destination matches one of its own and the datagram contains no errors in its header, 8 bit IPv4 Protocol field is checked to determine which protocol to invoke
            - Tunneling - violates the original concepts of layering and encapsulation
        - once the network layer (IPv4 or IPv6) determines that the incoming datagram is valid and the correct transport protocol has been determined, the reusing datagram is passed to the transport layer for processing
        - at the transport layer protocols use port numbers for demultiplexing to the appropriate receiving application

- **1.3.3 Port Numbers**
    - 16-bit nonnegative integers
    - These numbers are abstract and do not refer to anything physical
    - Each IP address has 65,536 port numbers for each transport protocol that uses port numbers, and they are used for determining the correct receiving application
    - For client/server applications a server "binds" to a port number, and subsequently one or more clients establish connections to the port number using a particular transport protocol on a particular machine
    - Internet Assigned Numbers Authority (IANA)
        - well-known port numbers (0-1023)
        - registered port numbers (1024-49151)
        - dynamic/private port numbers (49152-65535)
    - The range of well-known ports is used for identifying many well-known services such as the Secure Shell Protocol (SSH, port 22), FTP (ports 20 and 21) …
    - Protocols with multiple ports often have different port numbers depending on whether Transport Layer Security (TLS) is being used with the base application-layer protocol
    - The registered port numbers are available to clients or servers with special privileges, but IANA keeps a reserved registry for particular uses, so these port numbers should generally be avoided when developing new applications unless an IANA allocation has been procured

- The private/dynamic port numbers are essentially unregulated. Ae we will see, in some circumstance the value of the port number matters little because the port number being used is transient
- In some circumstances (e.g., on clients) the value of the port number matters little because the port number being used is transient. Such port numbers are also called ephemeral port numbers
  - They are considered temporary because a client typically needs one only as long as the user running the client needs service, and the client does not need to be found by the server in order to establish a connection
- Servers
  - conversely, generally require names and port numbers that do not change often in order to be found by clients

- 1.3.4 Names, Addresses, and the DNS

- TCP/IP each link-layer interface on each computer (including routers) has at least one IP address
  - enough to identify a host
- DNS (TCP/IP) - a distributed database that provides the mapping between host names and IP addresses (and vice versa)
  - is an application layer protocol and thus depends on the other protocols in order to operate
- Applications that manipulate names can call a standard API function to look up IP address corresponding to a given host's name. Similarly, a function is provided to do the reverse lookup—given an IP address
  - web browsers support this capability
  - Uniform Resource Locators (URLs)

- 1.4 Internets, Intranets, and Extranets

- Internet - has developed as the aggregate network resulting from the interconnection constituent networks over time
  - lowercase internet means multiple networks connected together, using a common protocol suite
  - uppercase internet refers to the collection of hosts around the world, that can communicate with each other using TCP/IP. The Internet is an internet, but the reverse is not true
- A router - special purpose device for connecting networks. They provide connections to many different types of physical networks
- Intranet - term used to describe a private internetwork, usually run by a business or other enterprise
  - users may connect to the intranet using a virtual private network (VPN)
  - VPNs usually use the tunneling method
- Extranets - consist of computers attached outside the serving enterprise's firewall

- 1.5 Designing Applications

- Networked applications are typically structured according to a small number of design patterns.
  - The most common of these are client/server and peer-to-peer

- 1.5.1 Client/Server

- Most network applications are designed so that one side is the client and the other side is the server
- Server provides some type of service to clients

- iterative and concurrent
- iterative server iterates through the following steps:
  - 1. wait for a client request to arrive
  - 2. process the client request
  - 3. Send the response back to the client that sent the request
  - 4. Go back to step 2
  - problem with iterative server occurs when step 2 takes a long time. During this time period no other clients are serviced
- concurrent server
  - 1. Wait for a client request to arrive
  - 2. Start a new server instance to handle this client's request. This may involve creating a new process, task, or a thread, depending on what the underlying operating system supports. This new server handles one client's entire request. When the requested task is complete, the new server terminates. Meanwhile, the original server instance continues to step 3.
  - 3. Go back to step 1
  - as a general rule most servers are concurrent

- **1.5.2 Peer-to-Peer**
  - each application acts both as client and as a server, sometimes as both at once, and is capable of forwarding requests
    - called peer-to-peer applications
  - A concurrent p2p application may receive an incoming request, determine if it is able to respond to the request, and if not forward the request on to some other peer.
    - Thus the set of p2p applications together form a network among applications, also called an overlay network
    - overlays are now commonplace and can be extremely powerful
  - discovery problem - that is how does one peer find which other peer can provide the data or service it wants in a network where peers may come and go
    - this is usually handled by a bootstrapping procedure whereby each client is initially configured with the addresses and port numbers of some peers that are likely to be operating. Once connected the new participant learns of other active peers and, depending on the protocol, what services or files they provide
- **1.5.3 Application Programming Interfaces (APIs)**
  - Applications whether p2p or client/server, need to express their desired network operations (e.g., make a connection, write or read data). This is usually supported by the host operating system using networking application programming interface.
    - The most popular API is called sockets
- **1.6 Standardization Process**
  - Internet Engineering Task Force
- **1.7 Implementations and Software Distributions**
  - Each popular operating system has its own implementation
  - TCP/IP - in Linux, Windows, sometimes FreeBSD and MacOS
  - In most cases, the particular implementation matters little
- **1.8 Attacks Involving the Internet Architecture**
  - Few attacks target the Internet architecture as a whole
  - Spoofing - The Internet architecture delivers IP datagrams based on destination IP addresses. As a result, malicious users are able to insert whatever IP address they choose into the source IP address field of each IP datagram they send

  - The resulting datagrams are delivered to their destinations, but it is difficult to perform attribution.
  - That is, it may be difficult or impossible to determine the origin of a datagram received from the Internet
- Spoofing can be combined with a variety of other attacks seen periodically on the Internet
- Denial-of-service (DoS) - usually involve using so much of some important resource that legitimate users are denied service
  - For example, sending so many IP datagrams to a server that is spends all of its time just processing the incoming packets and performing no other useful work is a type of DoS attack
  - Other attacks may include clogging the network with so much traffic that no other packets can be sent. This is often accomplished by using many sending computers, forming a distributed DoS (DDoS) attack
- Unauthorized access attacks involve accessing information or resources in an unauthorized fashion
  - Owning - exploiting protocol implementation bugs to take control of a system and turning it into a zombie or bot
  - black hats - programmers who intentionally develop malware and exploit systems for (illegal) profit or other malicious purposes
  - white hats - do the same sorts of technical things but notify vulnerable parties instead of exploiting them

- **1.9 Summary**
  - concepts of network architecture and design in general
  - Internet architecture was designed to interconnect different existing networks and provide for a wide range of services and protocols operating simultaneously
  - Packet switching using datagrams was chosen for its robustness and efficiency
    - security and predictable delivery of data were secondary concerns
  - TCP/IP protocol suite
    - Three main layers are network layer, transport layer, and application layer
    - network layer (IP) provides an unreliable datagram service and must be implemented by all systems addressable on the Internet
    - transport layers(TCP and UDP) provide an end-to-end service to applications running on end hosts.
      - TCP - provides in-ordered reliable stream delivery with flow control and congestion control
      - UDP - provides essentially no capabilities beyond IP except port numbers for demultiplexing and an error detection mechanism
        - supports multicast delivery
    - address and demultiplexing identifiers are used by each layer to avoid confusing protocols or different associations/connections of the same protocol
  - Port numbers do not resent anything physical; they are merely used as a way for applications that want to communicate to rendezvous
  - DNS - a distributed database application running on the Internet
  - internet - is a collection of networks
    - The common building block for an internet is a router that connects the networks at the IP layer
  - Internet - is an internet that spans the globe and interconnects nearly two billion users
  - Intranets - private internets and usually connected to the internet using special devices that attempt to prevent unauthorized access

- Extranets - usually consist of a subset of an institutions intranet that is designed to be accessed by partners or affiliates in a limited way
- Networked applications are usually designed using a client/server or peer-to-peer design pattern.
  - applications invoke APIs to perform networking tasks
- Sockets - the most common API for TCP/IP