Protocol Architecture, TCP/IP, and Internet-Based Applications

To destroy communication completely, there must be no rules in common between transmitter and receiver—neither of alphabet nor of syntax.

—On Human Communication,

Colin Cherry

The Need For Protocol Architecture

To transfer data several tasks must be performed:

1.) the source must activate communications path or inform network of destination

2.) the source must make sure that destination is prepared to receive data

3.) the file transfer application on source must confirm file management program at destination is prepared to accept and store file

4.) a format translation function may need to be performed if the formats on systems are different

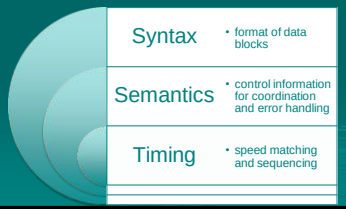
Functions of Protocol Architecture

- breaks logic into subtask modules which are implemented separately
- modules are arranged in a vertical stack
  - each layer in the stack performs a subset of functions
  - relies on next lower layer for primitive functions
  - changes in one layer should not require changes in other layers

Key Features of a Protocol

A protocol is a set of rules or conventions that allow peer layers to communicate.

The key features of a protocol are:



A Simple Protocol

agents involved:
- applications
- computers
- networks

examples of applications include file transfer and electronic mail

these execute on computers that support multiple simultaneous applications

Communication Layers

- communication tasks are organized into three relatively independent layers:
  - Network access layer
    - concerned with the exchange of data
  - Transport layer
    - provides reliable data transfer
  - Application layer
    - Contains logic to support applications

Network Access Layer

- covers the exchange of data between an end system and the network that it is attached to
- concerned with issues like :
  - destination address provision
  - invoking specific services like priority
  - access to & routing data across a network for two end systems attached to the same network

Transport Layer

concerned with providing reliable delivery of data

essentially independent of the nature of the applications

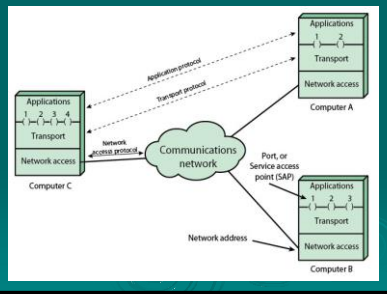
common layer shared by all applications

Application Layer

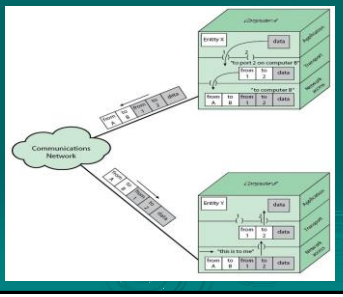
contains the logic needed to support user applications

separate module is needed for each type of application

Protocol Architecture and Networks



Protocols in a Simplified Architecture



Addressing

Two levels of addressing are needed:

each computer on the network has a unique network address

each application has an address that is unique with that computer (SAPs)

Protocol Data Unit (PDU)

- the combination of data and control information is a protocol data unit (PDU)
- typically control information is contained in a PDU header
  - control information is used by the peer transport protocol at computer B
- headers may include:
  - source port, destination port, sequence number, and error-detection code

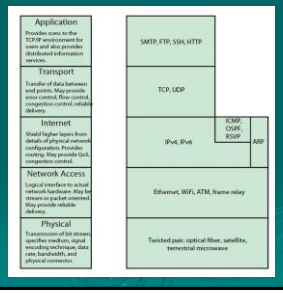
Network Access Protocol

- after receiving segment from transport layer, the network access protocol must request transmission over the network
  - the network access protocol creates a network access PDU (packet) with control information
- header includes:
  - source computer address
  - destination computer address
  - facilities requests

TCP/IP Protocol Architecture

- Result of protocol research and development conducted on ARPANET
- Referred to as TCP/IP protocol suite
- TCP/IP comprises a large collection of protocols that are Internet standards

TCP/IP Layers and Example Protocols



Physical Layer

- covers the physical interface between computer and network
- concerned with issues like:
  - characteristics of transmission medium
  - nature of the signals
  - data rates

Network Access Layer

- covers the exchange of data between an end system and the network that it is attached to
- concerned with issues like :
  - destination address provision
  - invoking specific services like priority
  - access to & routing data across a network for two end systems attached to the same network

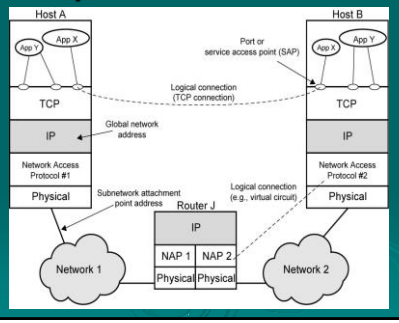
Internet Layer

- implements procedures needed to allow data to travel across multiple interconnected networks
- uses the Internet Protocol (IP) to provide routing function
- implemented in end systems and routers

Host-to-Host (Transport) Layer

- concerned with providing reliable delivery of data
- common layer shared by all applications
- most commonly used protocol is the Transmission Control Protocol (TCP)

Operation of TCP/IP

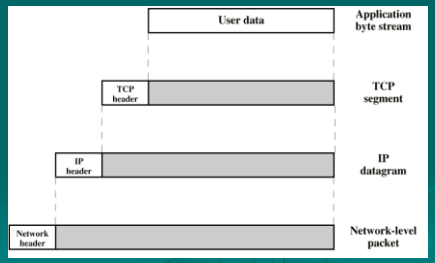


TCP/IP Address Requirements

Two levels of addressing are needed:

- each host on a subnetwork must have a unique global internet address
- each process with a host must have an address (known as a port) that is unique within the host

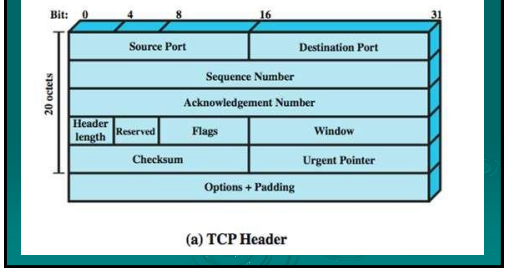
Operation of TCP/IP



Transmission Control Protocol (TCP)

- TCP is the transport layer protocol for most applications
- TCP provides a reliable connection for transfer of data between applications
- A TCP segment is the basic protocol unit
- TCP tracks segments between entities for duration of each connection

TCP Header

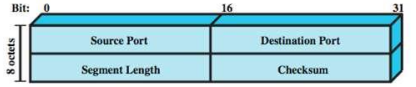


(a) TCP Header

User Datagram Protocol (UDP)

- alternative to TCP
- does not guarantee delivery, preservation of sequence, or protection against duplication
- adds port addressing capability to IP
- used with Simple Network Management Protocol (SNMP)

UDP Header

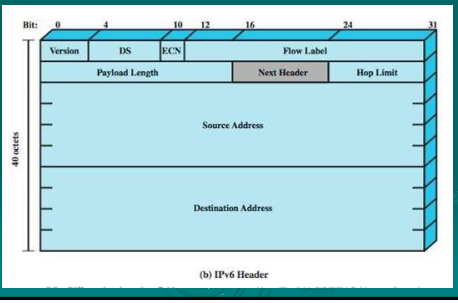


(b) UDP Header

IPv6

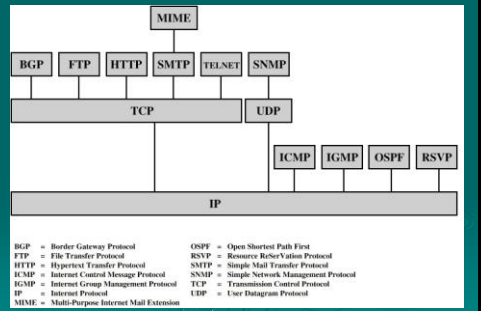
- Provides enhancements over existing IP
- Designed to accommodate higher speeds and the mix of graphic and video data
- Driving force was the need for more addresses due to growth of the Internet
- IPv6 includes 128-bit source and destination address fields

IPv6 Header

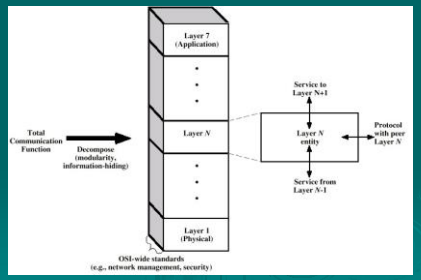


(b) IPv6 Header

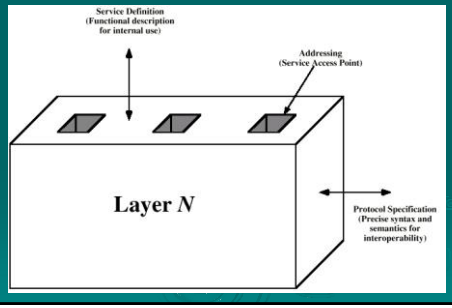
TCP/IP Protocols



Standardized Protocol Architectures



Layer Specific Standards



OSI Standardization

- framework for standardization was motivator
- lower layers are concerned with greater levels of details
- each layer provides services to the next higher layer
- three key elements:

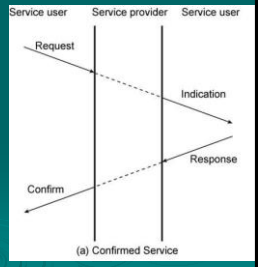
Protocol specification → Service definition → Addressing

Primitive Types

REQUEST	A primitive issued by a service user to invoke some service and to pass the parameters needed to specify fully the requested service
INDICATION	A primitive issued by a service provider either to: indicate that a procedure has been invoked by the peer service user on the connection and to provide the associated parameters, or notify the service user of a provider-initiated action
RESPONSE	A primitive issued by a service user to acknowledge or complete some procedure previously invoked by an indication to that user
CONFIRM	A primitive issued by a service provider to acknowledge or complete some procedure previously invoked by a request by the service user

Service Primitives and Parameters

- define services between adjacent layers using:
  - **primitives** to specify function performed
  - **parameters** to pass data and control information



Internet Applications

Applications that operate on top of TCP include: