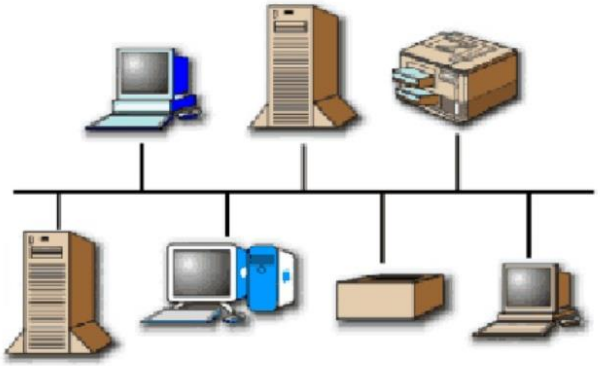
Basics Of Networking

What is a Computer Network?

A network is a collection of computers, printers, routers, switches, and other devices that are able to communicate with each other over some transmission media.

There are two basic types of networks currently in existence:

- A Local Area Network (LAN)
- A Wide Area Network (WAN)



Local Area Network (LAN)

A Local Area Network (LAN) is a group of computers and(LAN) network communication devices within a limited geographic area, such as an office building. *No third party involvement here.*

They are characterized by the following:

- High data transfer speeds
- Generally less expensive technologies
- Limited geographic area

Wide Area Network (WAN)

A Wide Area Network (WAN) interconnects LANs. It is not restricted to a particular geographic area and may be interconnected around the world. *Third party network is involved.*

They are characterized by the following:

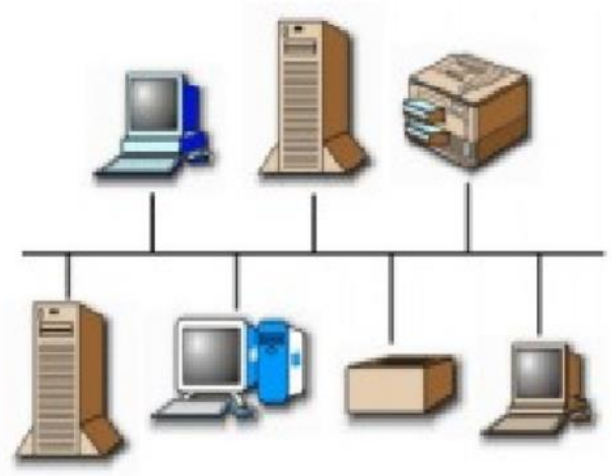
- Multiple interconnected LANs
- Generally more expensive technology
- More sophisticated to implement than LANs
- Exist in an unlimited geographic area
- Less error resistance due to transmission travel distances

Common LAN Topologies

Bus Architecture

In a bus topology:

- a single cable connects each workstation in a linear, daisy-chained fashion.
- signals are broadcasted to all stations, but stations only act on the frames addressed to them.

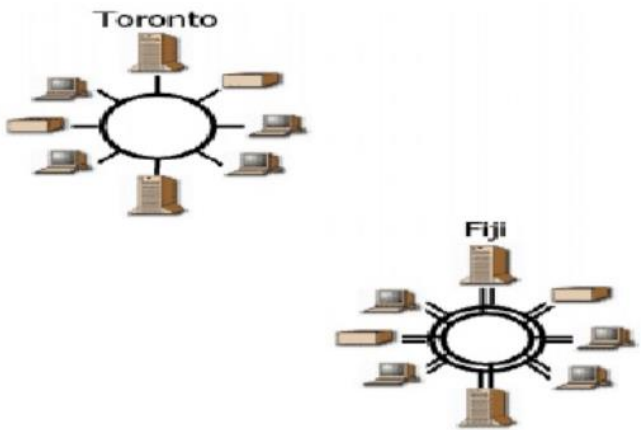


Common LAN Topologies

Ring Architecture

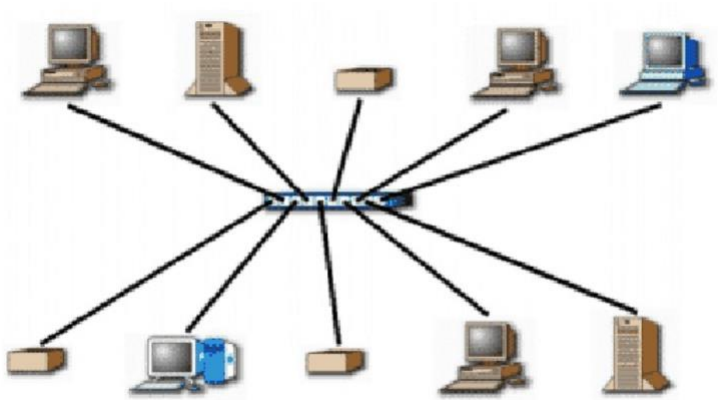
In a ring topology:

- Unidirectional links connect the transmit side of one device to the receive side of another device.
- Devices transmit frames to the next device (downstream member) in the ring.



Star Topology

In a star topology, each station is connected to a central hub or concentrator that functions as a multi-port repeater. Each station broadcasts to all of the devices connected to the hub. Physical LAN topologies are usually characterized as either bus or ring.



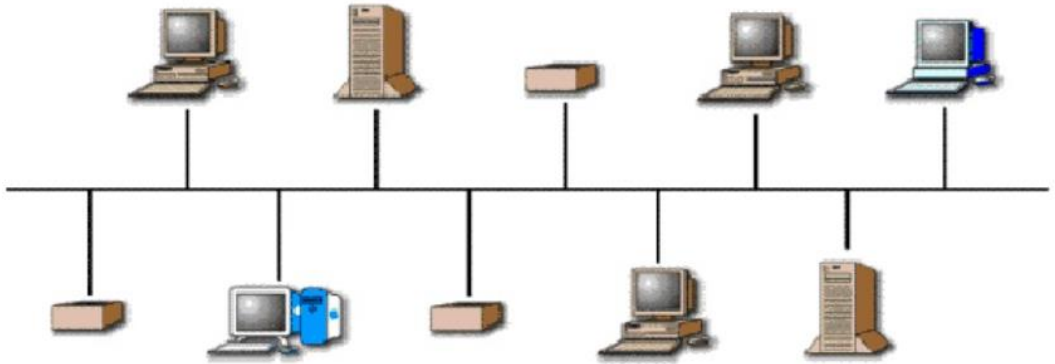
LAN Transmission Methods

LAN transmission methods fall into 3 main categories:

- Unicast transmission
- Multicast transmission
- Broadcast transmission

Unicast Transmission

In unicast transmissions, a single data packet is sent from a source to a single destination on the network.



Unicast Process

- The source addresses the packet with the destination address.
- The packet is sent into the network.
- The network delivers the packet to the destination.

Multicast Transmission

In multicast transmissions, a single data packet is copied and sent to specific destinations on the network

Multicast Process

- The source addresses the packet using a multicast address.
- The packet is sent into the network.
- The network copies the packet.
- A copy is delivered to each destination that is included in the multicast address.

Broadcast Transmission

In multicast transmissions, a single data packet is copied and sent to specific destinations on the network

Broadcast Process

- The source addresses the packet with the broadcast address.
- The packet is sent into the network.
- The network copies the packet.
- The packet copies are delivered to all destinations on the network.

LAN Infrastructure Devices

There are numerous devices associated with data information flow across a LAN. When adjoined, they create the infrastructure of a functional LAN.

These devices include:

- Repeaters
- Bridges
- Hubs
- Switches
- Routers

Protocols

- Cooperative action is necessary
  - computer networking is not only to exchange bytes
  - huge system with several utilities and functions. For examples
    - error detection
    - Encryption
    - Routing
    - etc.
- For proper communication, entities in different systems must speak the same language
  - there must be mutually acceptable conventions and rules about the content, timing and underlying mechanisms
- Those conventions and associated rules are referred as “PROTOCOLS”

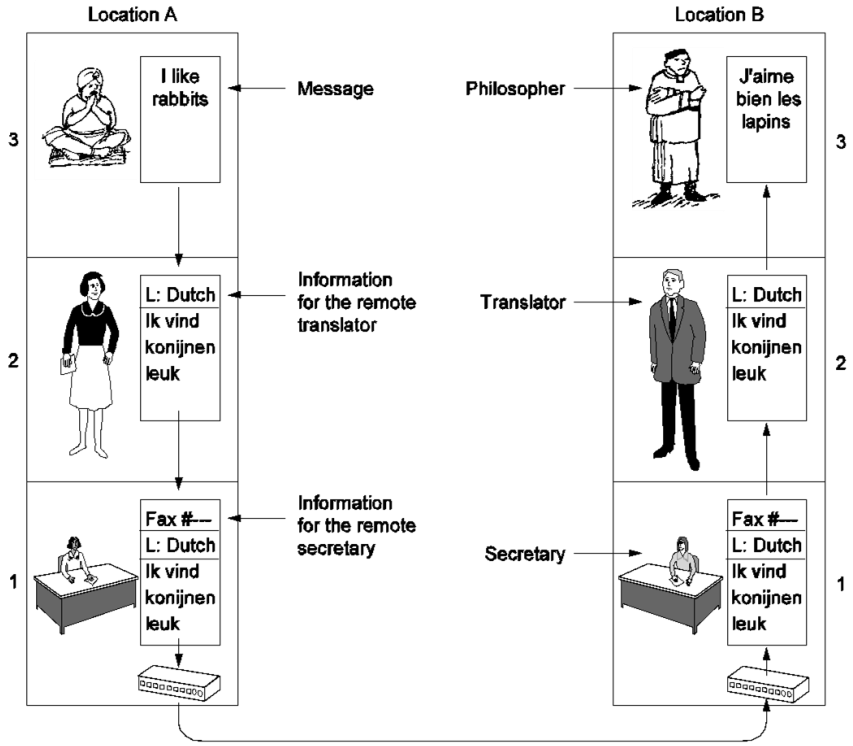
Protocol Architecture

- Task of data transfer is broken up into some modules
  - Why?
  - How do these modules interact?
- For example, file transfer could use three modules
  - File transfer application
  - Communication service module
  - Network access module

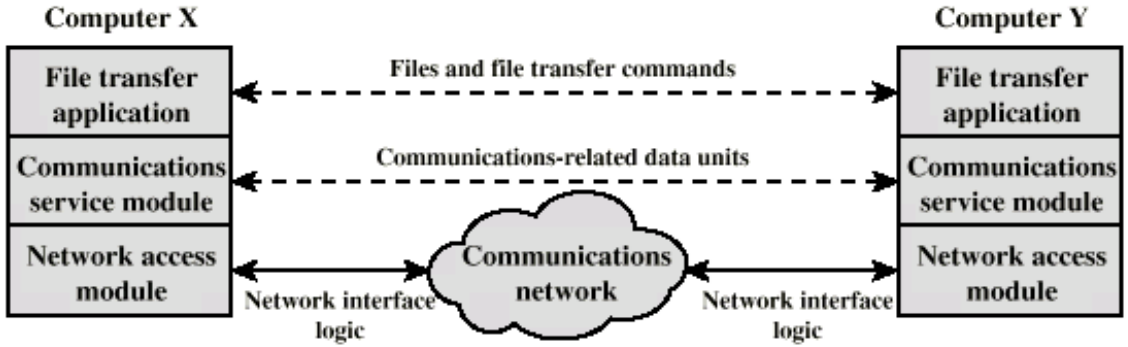
A Real World Example to Protocol Architecture philosopher-translator-secretary architecture

Issues:

- peer-to-peer protocols are independent of each other
  - for example, secretaries may change the comm. medium to email
  - or the translators may agree on using another common language
- Each layer adds a header



Simplified File Transfer Architecture



File Transfer Application Layer: Application specific commands, passwords and the actual file(s) – high level data

Communications Service Module: reliable transfer of those data – error detection, ordered delivery of data packets, etc.

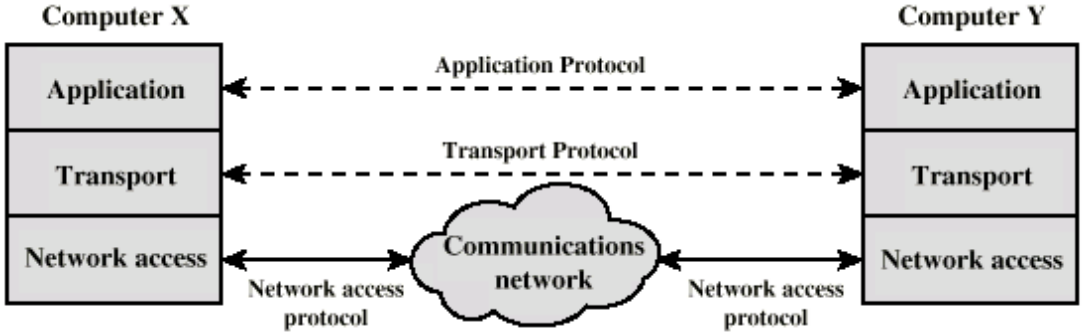
Network Module: actual transfer of data and dealing with the network – if the network changes, only this module is affected, not the whole system

General protocol architecture principles that we have seen so far

- Layered structure
  - Protocol stack
- Each layer provides services to upper layer; expect services from lower one
  - Layer interfaces should be well-defined
- Peer entities communicate using their own protocol
  - peer-to-peer protocols
  - independent of protocols at other layers
  - if one protocol changes, other protocols should not get affected

A General Three Layer Model

- Generalize the previous example for a generic application
  - we can have different applications (e-mail, file transfer, …)



- Network Access Layer
- Transport Layer
- Application Layer

Network Access Layer

- Exchange of data between the computer and the network
- Sending computer provides address of destination
  - so that network can route
- Different switching and networking techniques
  - Circuit switching
  - Packet switching
  - LANs
  - etc.
- This layer may need specific drivers and interface equipment depending on type of network used.
- But upper layers do not see these details
  - independence property

Transport Layer

- Reliable data exchange
  - to make sure that all the data packets arrived in the same order in which they are sent out
  - Packets not received or received in error are retransmitted
- Independent of network being used
- Independent of application

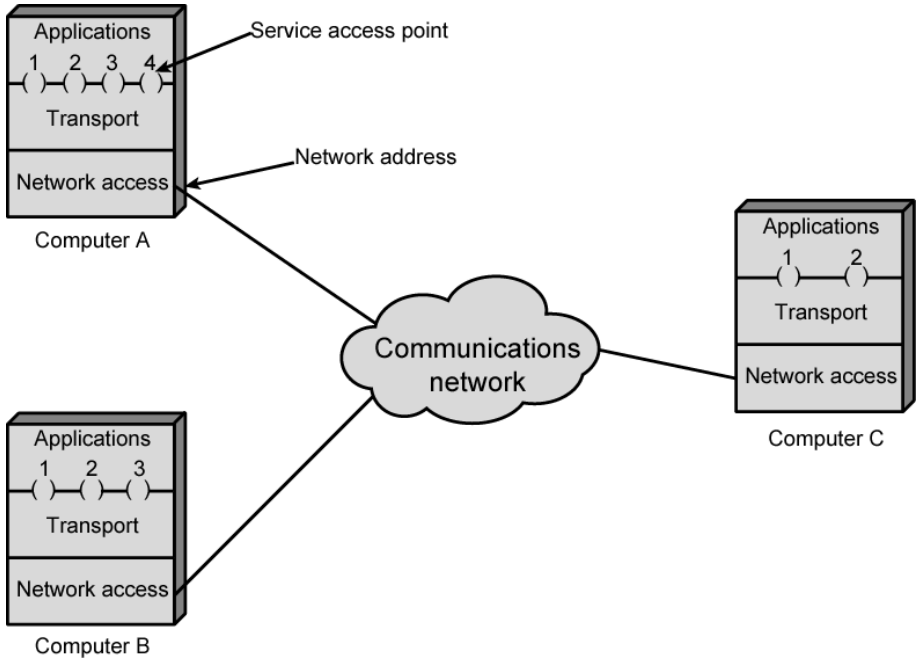
Application Layer

- Support for different user applications
- e.g. e-mail, file transfer

Addressing Requirements

- Two levels of addressing required
- Each computer needs unique network address
- Each application on a (multi-tasking) computer needs a unique address within the computer
    - The service access point or SAP
    - The port number in TCP/IP protocol stack

Protocol Architectures and Networks



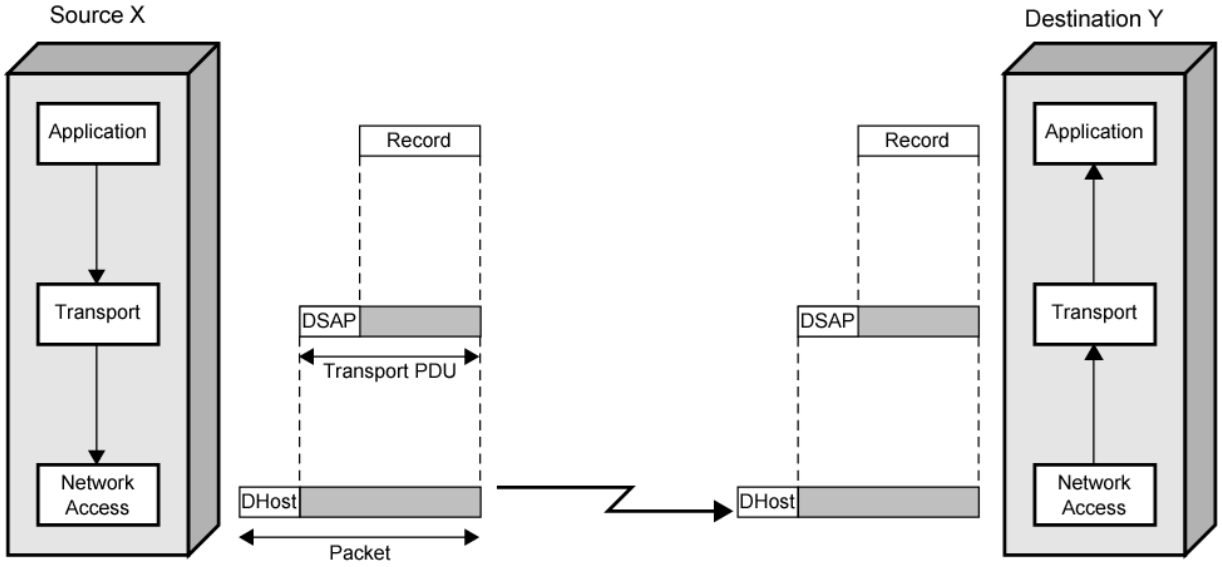
Protocol Data Units (PDU)

- User data is passed from layer to layer
- Control information is added/removed to/from user data at each layer
  - Header (and sometimes trailer)
  - each layer has a different header/trailer
- Data + header + trailer = PDU (Protocol Data Unit)
  - This is basically what we call packet
  - each layer has a different PDU

Network PDU

- Adds network header
  - network address for destination computer
  - optional facilities from network (e.g. priority level)

Operation of a Protocol Architecture



DSAP = destination service access point
DHost = destination host

Standard Protocol Architectures

- Common set of conventions
- Nonstandard vs. standard protocols
  - Nonstandard: K sources and L receivers lead to K*L different protocols
  - If common protocol used, we design only once
- Products from different vendors interoperate
  - Customers do not stick to a specific vendor
  - If a common standard is not implemented in a product, then that product's market is limited; customers like standard products

Standard Protocol Architectures

- Two approaches (standard)
  - OSI Reference model
    - never used widely
    - but well known
  - TCP/IP protocol suite
    - Most widely used
- Another approach (proprietary)
  - IBM’s Systems Network Architecture (SNA)

OSI Reference Model

- Open Systems Interconnection (OSI)
- Reference model
  - provides a general framework for standardization
  - defines a set of layers and services provided by each layer
  - one or more protocols can be developed for each layer
- Developed by the International Organization for Standardization (ISO)
  - also published by ITU-T (International Telecommunications Union)

OSI Reference Model

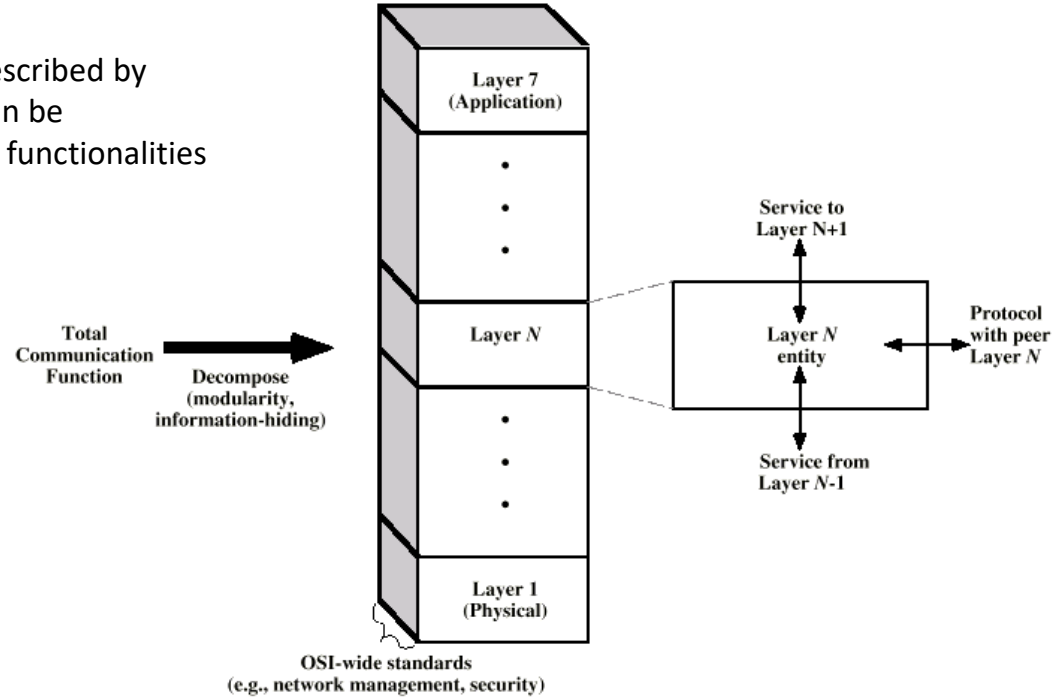
- A layered model
  - Seven layers – seven has been presented as the optimal number of layer
- Delivered too late (published in 1984)!
  - by that time TCP/IP started to become the de facto standard
- Although no OSI-based protocol survived, the model is still valid (in the textbooks)
  - For Data Link Layer (that we will see later) OSI protocols are still valid

OSI - The Layer Model

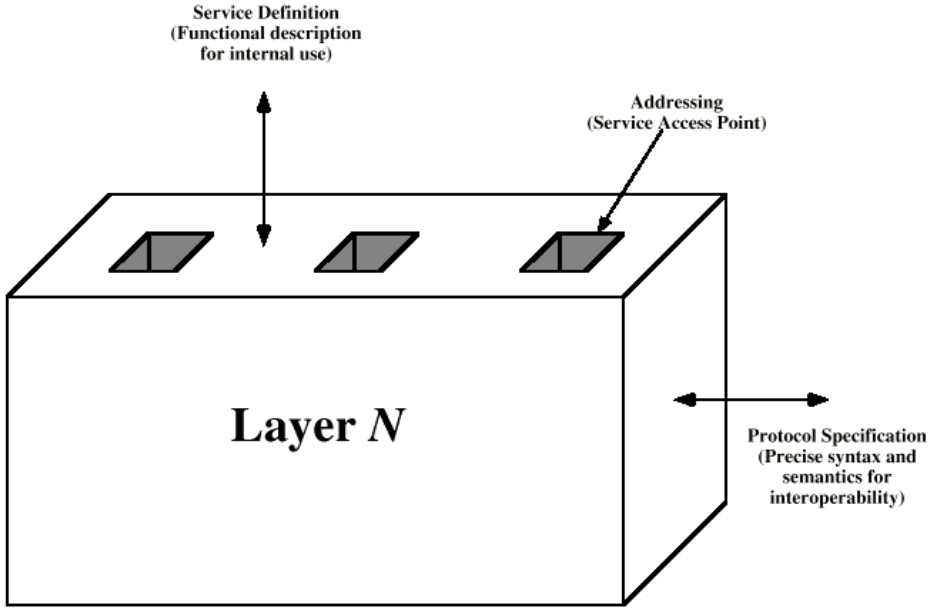
- Each layer performs a subset of the required communication functions
- Each layer relies on the next lower layer to perform more primitive functions
- Each layer provides services to the next higher layer
- Changes in one layer should not require changes in other layers

OSI as Framework for Standardization

layer functionalities are described by ISO; different standards can be developed based on these functionalities



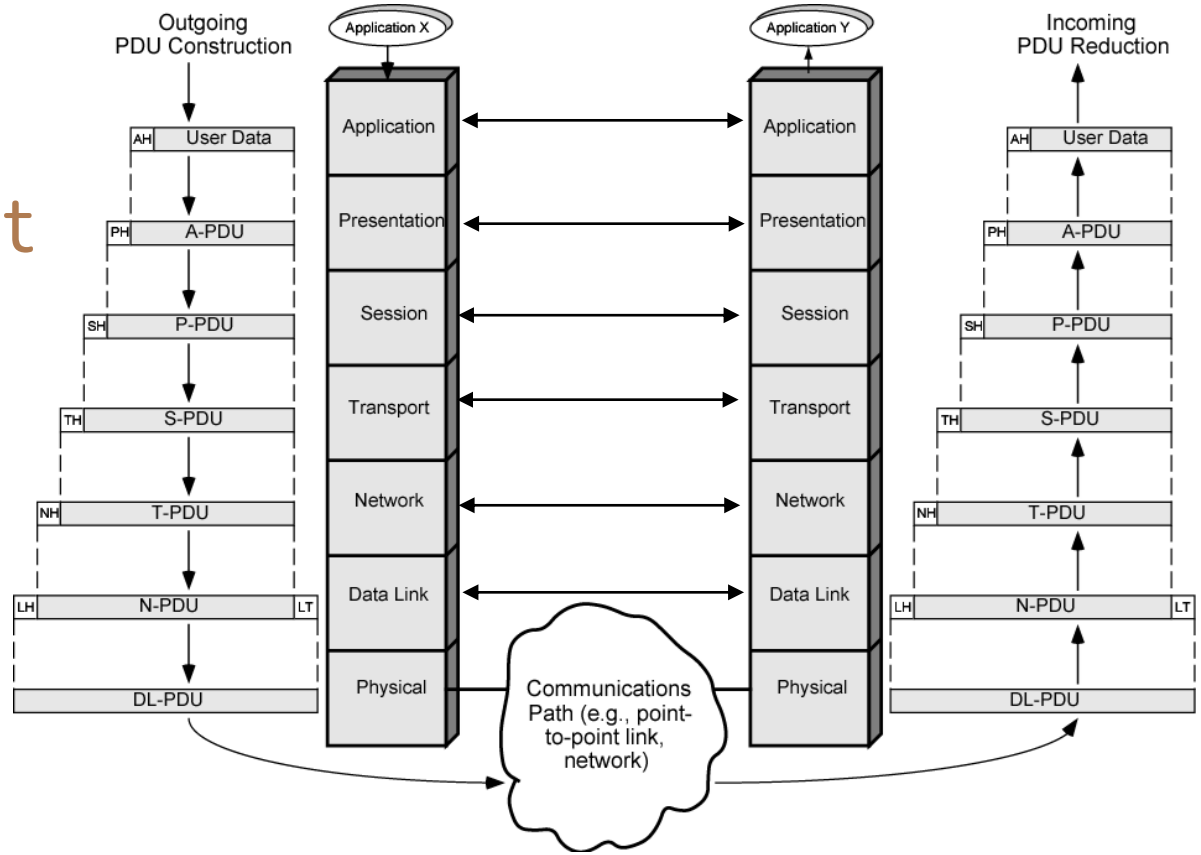
Layer Specific Standards



Elements of Standardization

- Protocol specification
  - Operates between the same layer on two systems
    - May involve different platforms
  - Protocol specification must be precise
    - Format of data units
    - Semantics of all fields
- Service definition
  - Functional description of what is provided to the next upper layer
- Addressing
  - Referenced by SAPs

The OSI Environment



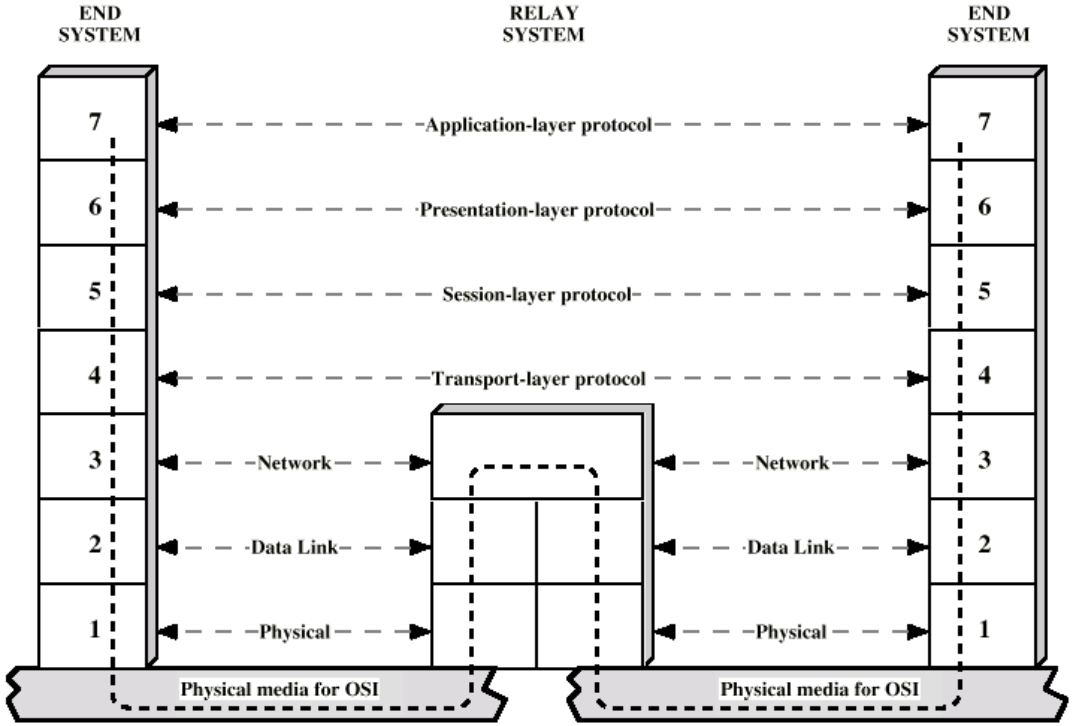
OSI Layers

- Physical
  - Physical interface between devices
  - Characteristics
    - Mechanical - interface specs
    - Electrical - voltage levels for bits, transmission rate, coding, etc.
- Data Link
  - Basic services: error detection and control, flow control at the link level (point to point)
    - Higher layers may assume error free transmission
  - Later a sublayer is added to Data Link Layer
    - MAC (Medium Access Control) sublayer
    - to deal with broadcast networks

OSI Layers

- Network
  - Transfer of information through communication network
    - network related issues
  - Network nodes (relays/routers) should perform switching and routing functions
  - QoS (Quality of Service) and congestion control are also addressed in this layer
  - Several other internetworking issues
    - e.g. differences in addressing, max. data length, etc.
  - Higher layers do not need to know about underlying networking technology
  - Not needed on direct links

Use of a Relay/Router



OSI Layers

- Transport
  - End to end exchange of data
  - In sequence, no losses, no duplicates
  - If needed, upper layer data are split into smaller units
- Session
  - Control of dialogues
    - whose turn to talk?
    - Dialogue discipline (full-duplex, half-duplex)
  - Checkpointing and recovery

OSI Layers

- Presentation
  - Data formats
  - Data compression
  - Encryption
- Application
  - Support for various applications

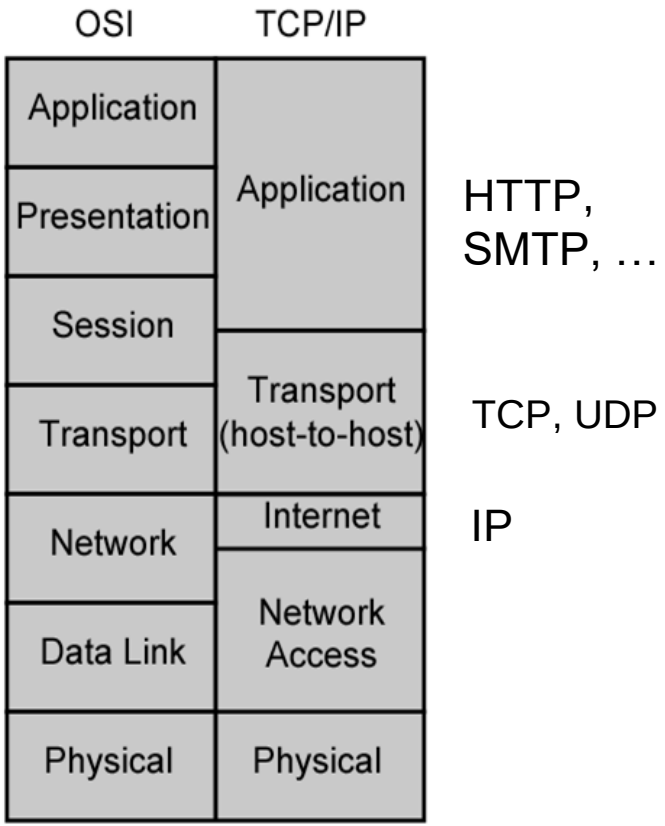
TCP/IP Protocol Suite

- Most widely used interoperable network protocol architecture
- Specified and extensively used before OSI
  - OSI was slow to take place in the market
- Funded by the US Defense Advanced Research Project Agency (DARPA) for its packet switched network (ARPANET)
  - DoD (Department of Defense) automatically created an enormous market for TCP/IP
- Used by the Internet and WWW

TCP/IP Protocol Suite

- TCP/IP does not have an official layer structure
- But protocols imply one
  - Application layer
  - Transport (host to host / end to end) layer
  - Internet layer
  - Network access layer
  - Physical layer
- Actually TCP/IP reference model has been built on its protocols
  - That is why that reference model is only for TCP/IP protocol suite
  - and this is why it is not so important to assign roles to each layer in TCP/IP; understanding TCP, IP and the application protocols would be enough

OSI vs. TCP/IP


OSI
TCP/IP
Application
Presentation
Session
Transport
Network
Data Link
Physical
Application
Transport (host-to-host)
Internet
Network Access
Physical
HTTP, SMTP, …
TCP, UDP
IP

Network Access and Physical Layers

- TCP/IP reference model does not discuss these layers too much
  - the node should connect to the network with a protocol such that it can send IP packets
  - this protocol is not defined by TCP/IP
  - mostly in hardware
  - a well known example is Ethernet

Internet Layer

- Connectionless, point to point internetworking protocol (uses the datagram approach)
  - takes care of routing across multiple networks
  - each packet travels in the network independently of each other
    - they may not arrive (if there is a problem in the network)
    - they may arrive out of order
  - a design decision enforced by DoD to make the system more flexible and responsive to loss of some subnet devices
- Implemented in end systems and routers as the Internet Protocol (IP)

Transport Layer

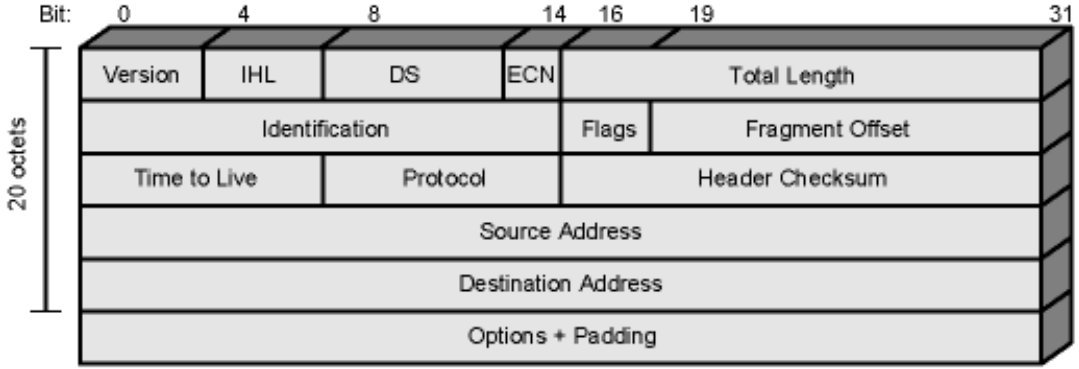
- End-to-end data transfer
- Transmission Control Protocol (TCP)
  - connection oriented
  - reliable delivery of data
  - ordering of delivery
- User Datagram Protocol (UDP)
  - connectionless service
  - delivery is not guaranteed
- Can you give example applications that use TCP and UDP?

Application Layer

- Support for user applications
- A separate module for each different application
    - e.g. HTTP, SMTP, telnet

IP (Internet Protocol)

- The core of the TCP/IP protocol suite
- Two versions co-exist
  - v4 – the widely used IP protocol
  - v6 – has been standardized in 1996, but still not widely deployed
- IP (v4) header minimum 20 octets (160 bits)



IPv6

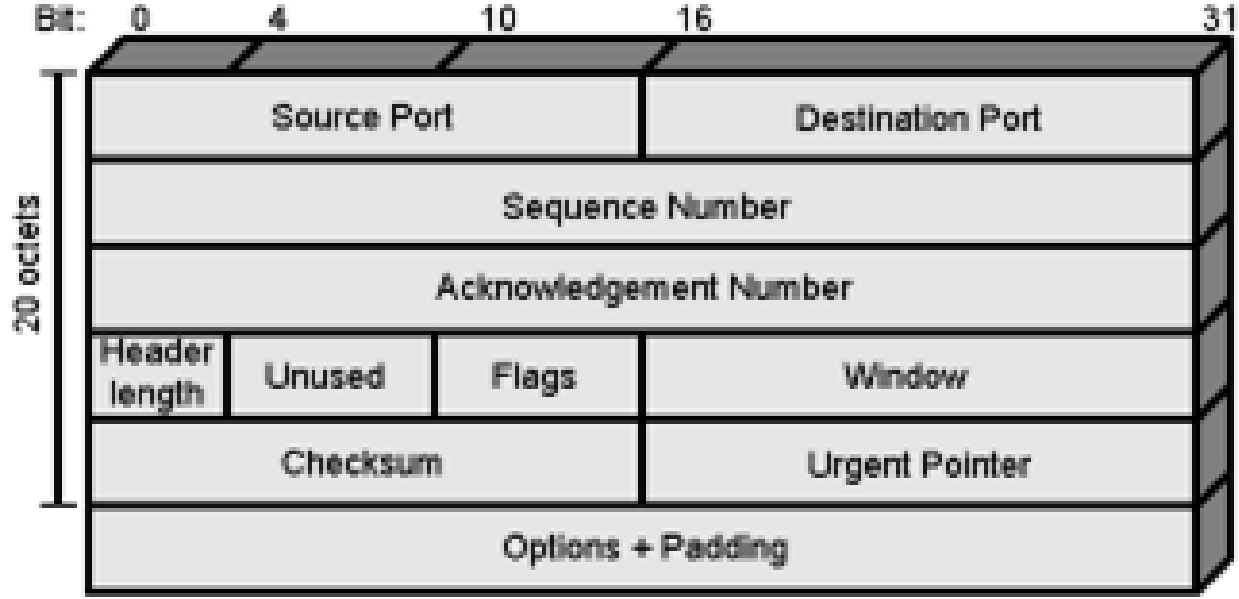
- IPv6
  - Enhancements over IPv4 for modern high speed networks
  - Support for multimedia data streams
- But the driving force behind v6 was to increase address space
  - 128-bit as compared to 32-bit of v4
- Not backward compatible
  - all equipment and software must change

TCP

- Transmission Control Protocol
  - end to end protocol
  - Reliable connection = provides flow and error control
- In TCP terms, a connection is a

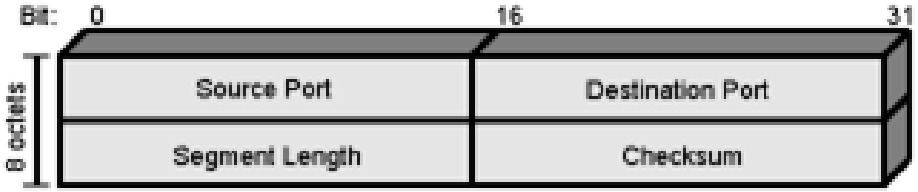
  temporary association between entities in different systems
- TCP PDU
  - Called “TCP segment”
  - Includes source and destination port
    - Identify respective users (applications)
    - pair of ports (together with the IP addresses) uniquely identify a connection; such an identification is necessary in order TCP to track segments between entities.

TCP Header


Bit:
0
4
10
16
31
20 octets
Source Port
Destination Port
Sequence Number
Acknowledgement Number
Header length
Unused
Flags
Window
Checksum
Urgent Pointer
Options + Padding

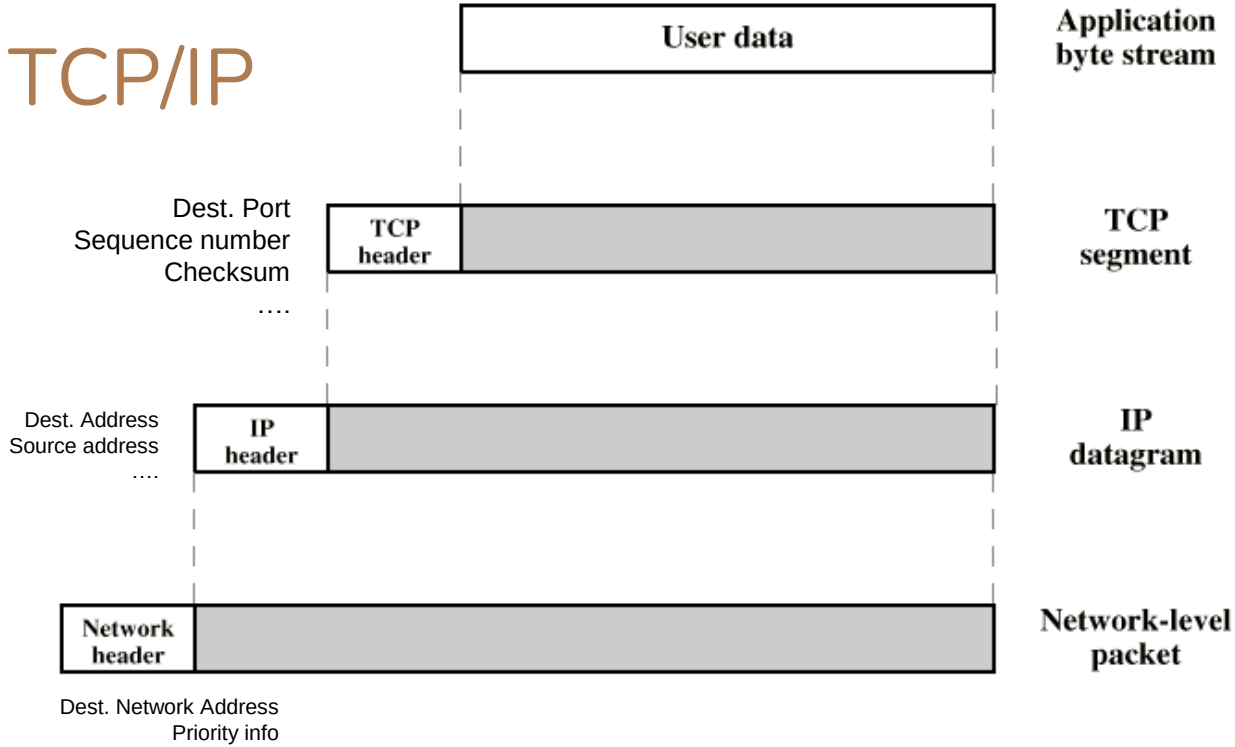
UDP

- User Datagram Protocol
- Alternative to TCP
  - end-to-end protocol
- Not guaranteed delivery
- No preservation of sequence
- No protection against duplication
- Minimum overhead

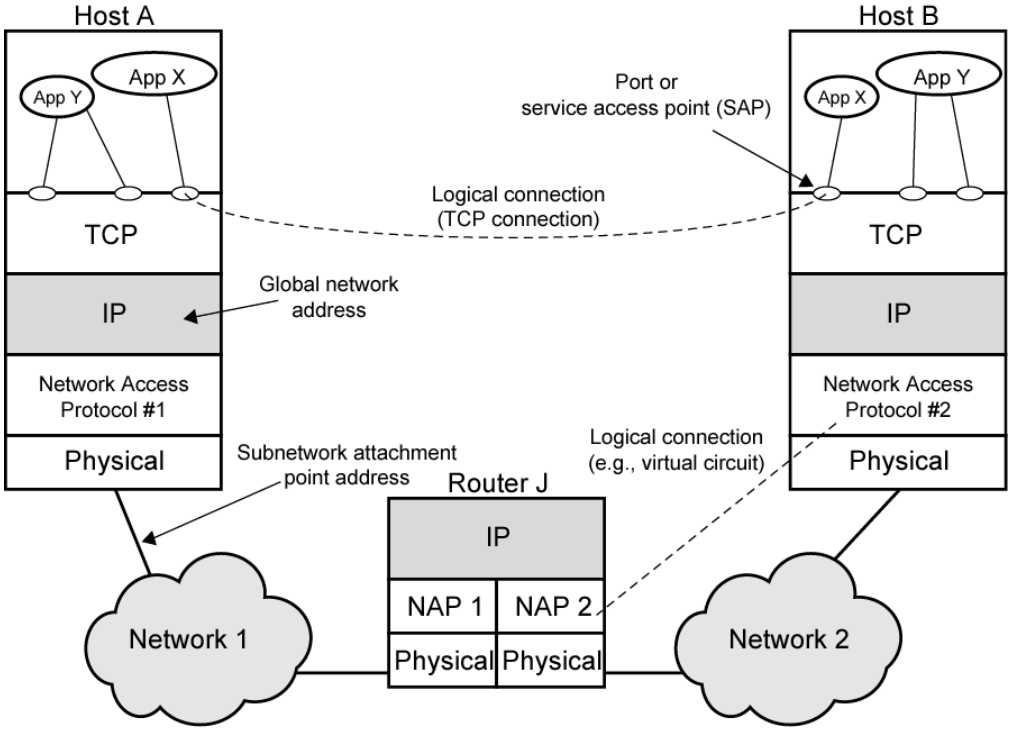


(b) UDP Header

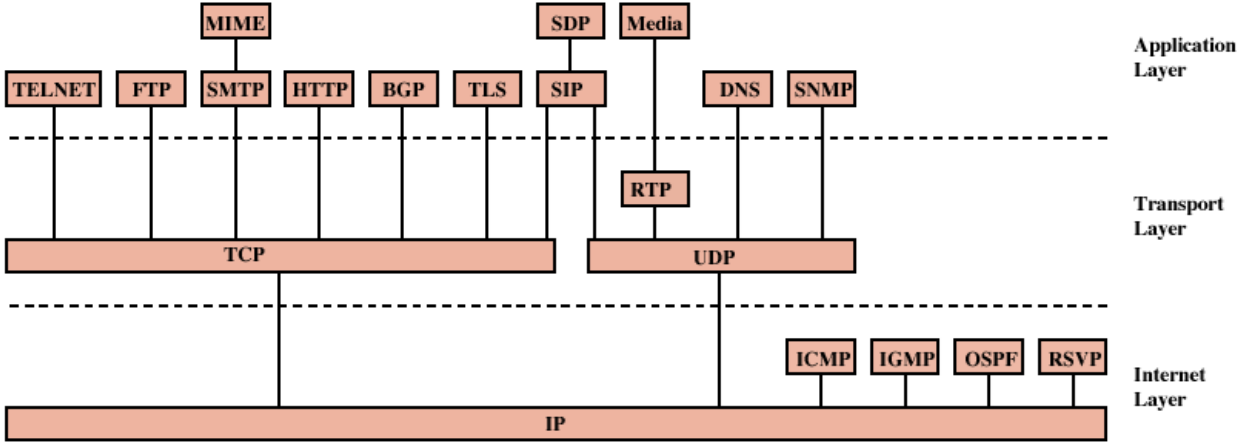
PDUs in TCP/IP



Operation of TCP and IP



Some Protocols in TCP/IP Suite



BGP	=	Border Gateway Protocol	RSVP	=	Resource ReSerVation Protocol
DNS	=	Domain Name System	RTP	=	Real-Time Transport Protocol
FTP	=	File Transfer Protocol	SDP	=	Session Description Protocol
HTTP	=	Hypertext Transfer Protocol	SIP	=	Session Initiation Protocol
ICMP	=	Internet Control Message Protocol	SMTP	=	Simple Mail Transfer Protocol
IGMP	=	Internet Group Management Protocol	SNMP	=	Simple Network Management Protocol
IP	=	Internet Protocol	TCP	=	Transmission Control Protocol
MIME	=	Multi-Purpose Internet Mail Extension	TLS	=	Transport Layer Security
OSPF	=	Open Shortest Path First	UDP	=	User Datagram Protocol

Internetworking

- Interconnected set of networks
  - May be seemed as a large network
- Each constituent network is a subnetwork
- Entire configuration referred to as an internet
  - not the Internet
    - conceptually the same, but by “internet” we do not mean a specific network
    - the Internet is the most important example of an internet

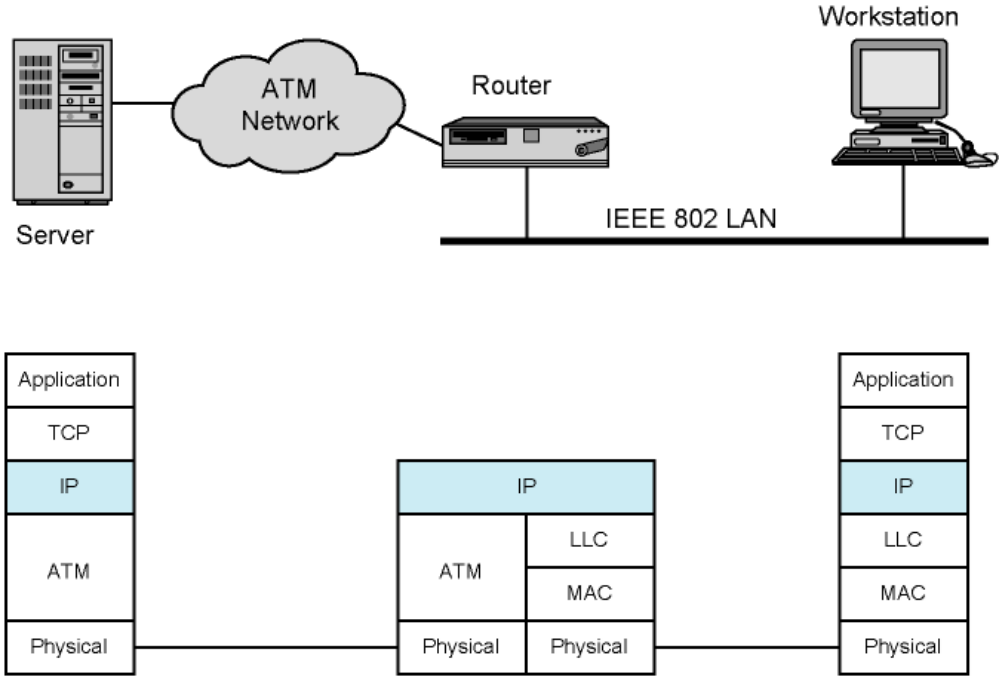
Internetworking Devices

- Each subnetwork supports communication among the devices attached to that subnetwork
  - End systems (ESs)
- Subnetworks connected by intermediate systems (ISs)
  - In practice, ISs are routers that are used to relay and route packets between different subnetworks
  - If subnetworks use different Network Access Protocols, router should support all of the protocols
  - In OSI terminology, a router works at layer 3 (network layer)

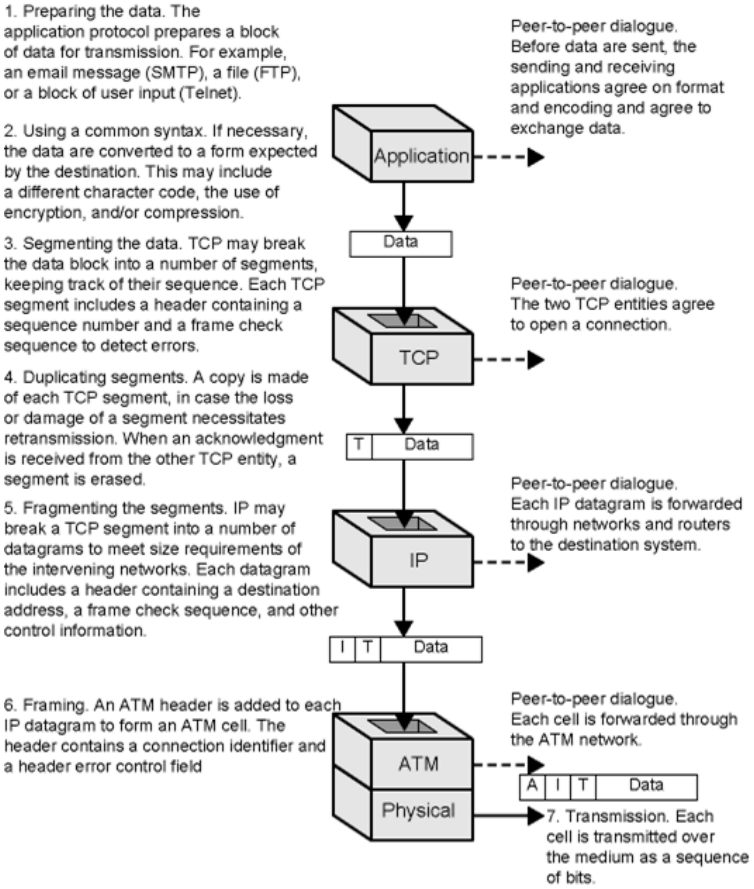
Routers

- Interconnect dissimilar subnetworks without any modifications on architecture of subnetworks
- Must accommodate differences among networks, such as
  - Addressing schemes
    - network addresses may need to be translated
  - Maximum packet sizes
    - if two subnetworks have different limits for max. packet sizes, then router may need fragment/reassemble the packets
- We have seen that subnetworks may have different network access and physical layers, but they have to speak the same (inter)network protocol implemented in all end systems and routers
  - The most important internetwork protocol is the IP protocol

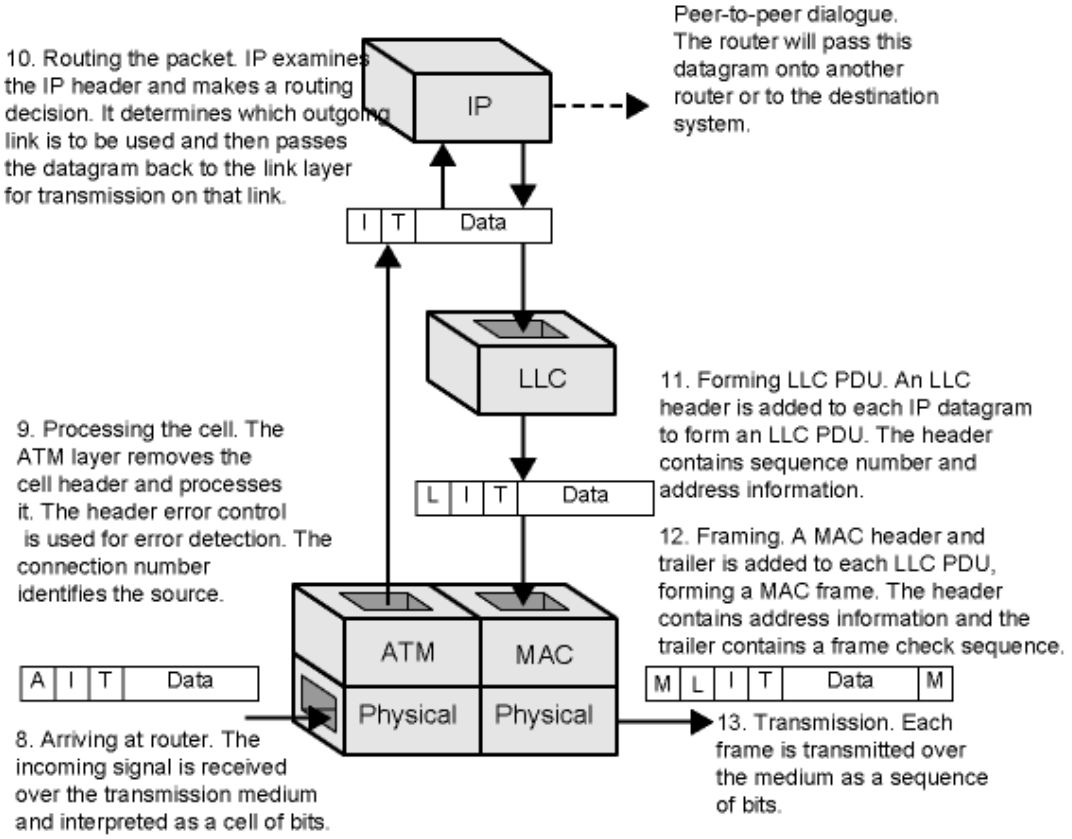
Configuration for TCP/IP **Example**



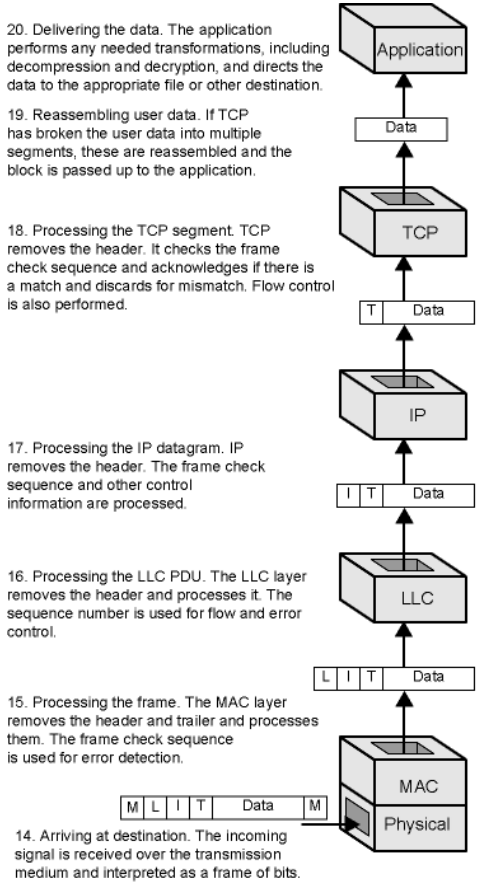
Action of Sender



Action of Router



Action of Receiver



Standards

- Required to allow for interoperability among equipments
- Advantages
  - Ensures a large market for equipment and software
  - Allows products from different vendors to communicate
- Disadvantage
  - Freeze technology (???)

Standards Organizations in Networking

- Internet Society
- ISO (International Organization for Standardization)
  - more formal
  - NGO, but most members are from governments
- ITU-T (formerly CCITT)
  - International Telecommunications Union
  - UN agency
  - governmental

Internet Society (ISOC)

- Internet development and standardization
- 3 suborganizations
  - IAB (Internet Architecture Board)
    - overall Internet architecture
  - IETF (Internet Engineering Task Force)
    - protocol engineering and development
  - IESG (Internet Engineering Steering Group)
    - monitors IETF standardization efforts

IETF Organization

- Grouped in areas
  - e.g. applications, security, routing, etc.
  - each area has an Area Director, who is also member of IESG
- Each area has several working groups
  - working groups actually contribute to standards/protocols, etc.
- Voluntary participation in IETF working groups
- For detail see
  - www.ietf.org or
  - RFC 3160 - The Tao of IETF - A Novice's Guide to the Internet Engineering Task Force

Internet Drafts and RFCs

- Internet Draft
  - Draft and temporary documents
  - expires in 6 months, if IESG does not approve it as an RFC
  - can be resubmitted
  - published online
  - comments are welcome
- RFC (Request for Comments)
  - final version
  - can obsolete previous RFCs about the same topic
  - actually an RFC can be of any type of document
    - not necessarily a standard
    - Best Current Practice, Experimental, Informational RFCs
    - April 1st RFCs (http://en.wikipedia.org/wiki/April_1_RFC)
      - My favorite is IP over Avian Carriers (RFC 1149)

Network Working Group
Request for Comments: 1149

D. Waitzman
BBN STC
1 April 1990

A Standard for the Transmission of IP Datagrams on Avian Carriers

Status of this Memo

This memo describes an experimental method for the encapsulation of IP datagrams in avian carriers. This specification is primarily useful in Metropolitan Area Networks. This is an experimental, not recommended standard. Distribution of this memo is unlimited.

Overview and Rational

Avian carriers can provide high delay, low throughput, and low altitude service. The connection topology is limited to a single point-to-point path for each carrier, used with standard carriers, but many carriers can be used without significant interference with each other, outside of early spring. This is because of the 3D ether space available to the carriers, in contrast to the 1D ether used by IEEE802.3. The carriers have an intrinsic collision avoidance system, which increases availability. Unlike some network technologies, such as packet radio, communication is not limited to line-of-sight distance. Connection oriented service is available in some cities, usually based upon a central hub topology.

Frame Format

The IP datagram is printed, on a small scroll of paper, in hexadecimal, with each octet separated by whitestuff and blackstuff. The scroll of paper is wrapped around one leg of the avian carrier. A band of duct tape is used to secure the datagram's edges. The bandwidth is limited to the leg length. The MTU is variable, and

Internet Standards Track

- Steps involve increasing amount of scrutiny and testing
- Step 1: Internet Draft
- Step 2: Proposed standard
  - Internet Draft approved as an RFC by IESG
  - must remain at least six months to advance
- Step 3: Draft standard
  - at least two independent and interoperable implementations
  - must remain at least 4 months
- Step 4: Internet standard
  - Significant operational experience
    - key difference between ISOC and other standardization organizations
  - Consensus needed

Internet Assigned Numbers Authority (IANA)

- An ISOC entity responsible for all “unique numbers” on the Internet
  - including IP addresses
- Almost all protocols work with numeric parameters
  - e.g. port numbers, error codes, status codes, message types, options, etc.
  - the meanings of all numeric codes are mostly specified in RFCs, but number assignment is formalized by IANA

Firewalld

Firewalld is a firewall management tool for Linux operating systems. It provides firewall features by acting as a front-end for the Linux kernel's netfilter framework via the nftables user space utility (before v0.6.0 iptables backend), acting as an alternative to the nft command line program. The name firewalld adheres to the Unix convention of naming system daemons by appending the letter "d"

firewalld supports both IPv4 and IPv6 networks and can administer separate firewall zones with varying degrees of trust as defined in zone profiles. Administrators can configure Network Manager to automatically switch zone profiles based on known Wi-Fi (wireless) and Ethernet (wired) networks, but firewalld cannot do this on its own.

Services and applications can use the D-Bus interface to query and configure the firewall. firewalld supports timed rules, meaning the number of connections (or "hits") to a service can be limited globally. There is no support for hit-counting and subsequent connection rejection per source IP; a common technique deployed to limit the impact of brute-force hacking and distributed denial-of-service attacks.

firewalld's command syntax is similar to but more verbose than other iptables front-ends like Ubuntu's Uncomplicated Firewall (ufw). The command-line interface allows managing firewall rulesets for protocol, ports, source and destination; or predefined services by name.

Install Firewalld

The default firewall system for Ubuntu is ufw but you can install and use Firewalld if you prefer. Install Firewalld on Ubuntu 18.04 / Ubuntu 16.04 by running the commands:

```
sudo apt-get install firewalld
```

By default, the service should be started, if not running, start and enable it to start on boot:

```
sudo systemctl enable firewalld
sudo systemctl start firewalld
```

Install Firewalld

Confirm that the service is running:

```
$ sudo firewall-cmd --state
running
```

If you have ufw enabled, disable it to make firewalld your default firewall

```
sudo ufw disable
```

Use Firewalld

Now that the package has been installed and firewalld service started, let' look at few usage examples See below examples for the basic usage of firewalld.

1. List all firewall rules configured

`ssh` and `dhcpv6-client` services are enabled by default when you start firewalld service.

```
# firewall-cmd --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Use Firewalld

2. Get a list of all services that can be enabled using a name

```
sudo firewall-cmd --get-services
```

3. Enable `http` service

```
sudo firewall-cmd --add-service=http --permanent
```

The `--permanent` option means persist rules against server reboots.

Use Firewalld

4. Enable both http and https on a single line

```
sudo firewall-cmd --permanent --add-service={http,https} --permanent
```

5. Enable TCP port 7070

```
sudo firewall-cmd --add-port=7070/tcp --permanent
```

Use Firewalld

6. Enable UDP port 514

```
sudo firewall-cmd --add-port=514/udp --permanent
```

7. Create a new zone

```
sudo firewall-cmd --new-zone=myzone --permanent
```

8. Enable service on a specific zone

```
sudo firewall-cmd --zone=myzone --add-port=4567/tcp --permanent
```

9. Set default zone

```
sudo firewall-cmd --set-default-zone=public --permanent
```

Use Firewalld

10. Add an interface to a zone

```
sudo firewall-cmd --get-zone-of-interface=eth0 --permanent
sudo firewall-cmd --zone=<zone> --add-interface=eth0 --permanent
```

11. Allow access to a port from specific subnet/IP

```
$ sudo firewall-cmd --add-rich-rule 'rule family="ipv4" service name="ssh" \
source address="192.168.0.12/32" accept' --permanent
$ sudo firewall-cmd --add-rich-rule 'rule family="ipv4" service name="ssh" \
source address="10.1.1.0/24" accept' --permanent
```

Use Firewalld

12. List rich rules

```
sudo firewall-cmd --list-rich-rules
```

13. Configure Port forwarding

```
# Enable masquerading
$ sudo firewall-cmd --add-masquerade --permanent

# Port forward to a different port within same server ( 22 > 2022)
$ sudo firewall-cmd --add-forward-port=port=22:proto=tcp:toport=2022 --permanent

# Port forward to same port on a different server (local:22 > 192.168.2.10:22)
$ sudo firewall-cmd --add-forward-port=port=22:proto=tcp:toaddr=192.168.2.10 --permanent

# Port forward to different port on a different server (local:7071 > 10.50.142.37:9071)
$ sudo firewall-cmd --add-forward-port=port=7071:proto=tcp:toport=9071:toaddr=10.50.142.37 --permanent
```

Use Firewalld

14. Removing port/service

Replace `--add` with `--remove`

Q&A