

# Computer Networks: Overview and I/O Considerations

Overview of the Internet

Packets and Connectionless Networks

Three protocols: IP, UDP, and TCP

The NIC (Network Interface Card) as a DMA Device

Interaction of IP with the Operating System and Application Programs

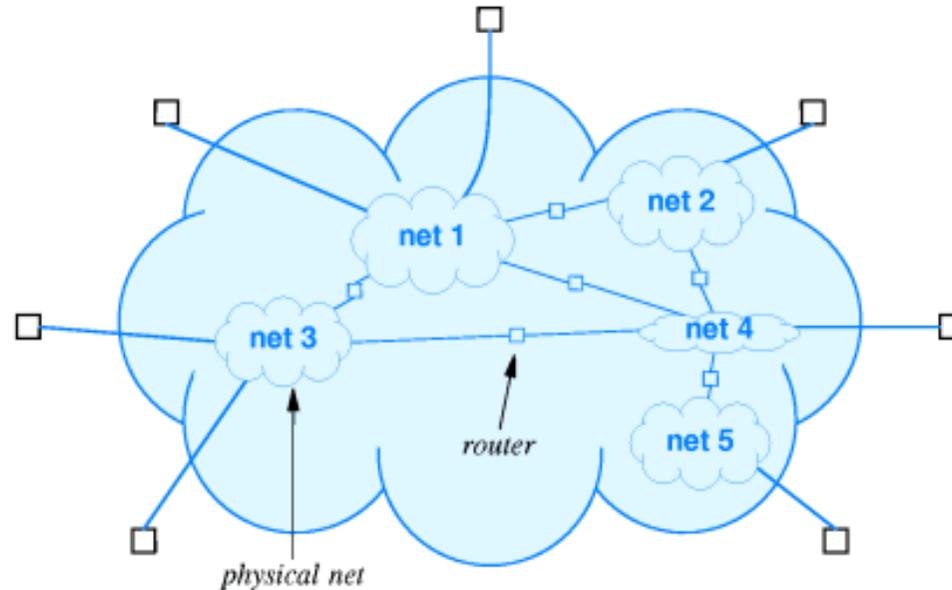
## Sources:

1. The Essentials of Computer Organization and Architecture  
Linda Null & Julia Lobur, Jones & Bartlett, 2006. ISBN 0 – 7637 – 3769 – 0.
2. Computer Networks and Internets with Internet Applications  
Douglas E. Comer, Pearson/Prentice–Hall, 2004. ISBN 0 – 13 – 143351 – 2.
3. Internetworking with TCP/IP: Volume II (Design, Implementation, & Internals)  
Douglas E. Comer and David L. Stevens, Prentice–Hall, 1999.  
ISBN 0 – 13 – 973843 – 6.

# An Overview of the Internet

The global Internet is best seen as a mechanism that allows computers to communicate.

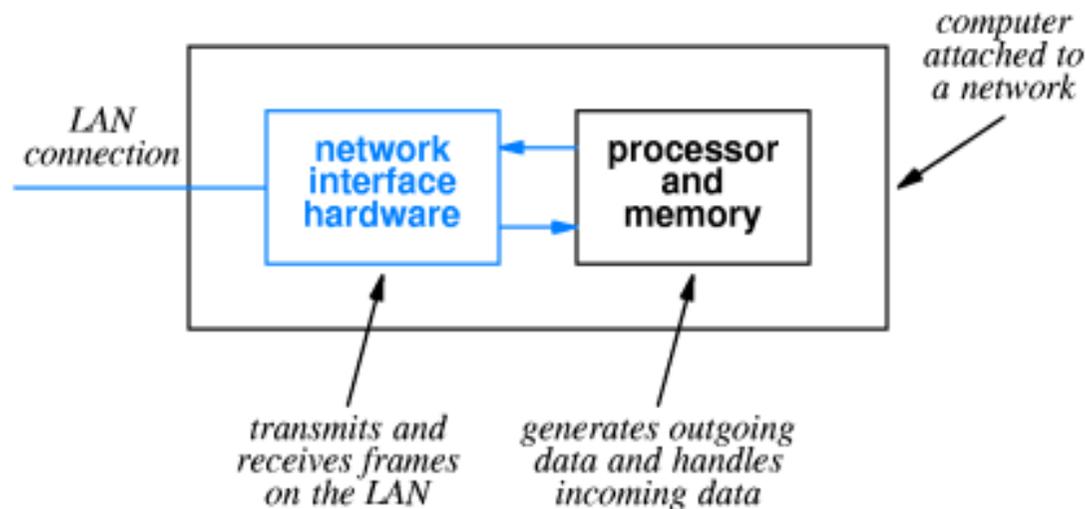
The Internet is a collection of interconnected networks, each with its own protocol. It provides the illusion of a single network, but has considerable internal structure.



# What Is Connected to the Internet?

We consider our computers to be connected to the Internet.

Technically, it is the NIC (Network Interface Card) that is connected.



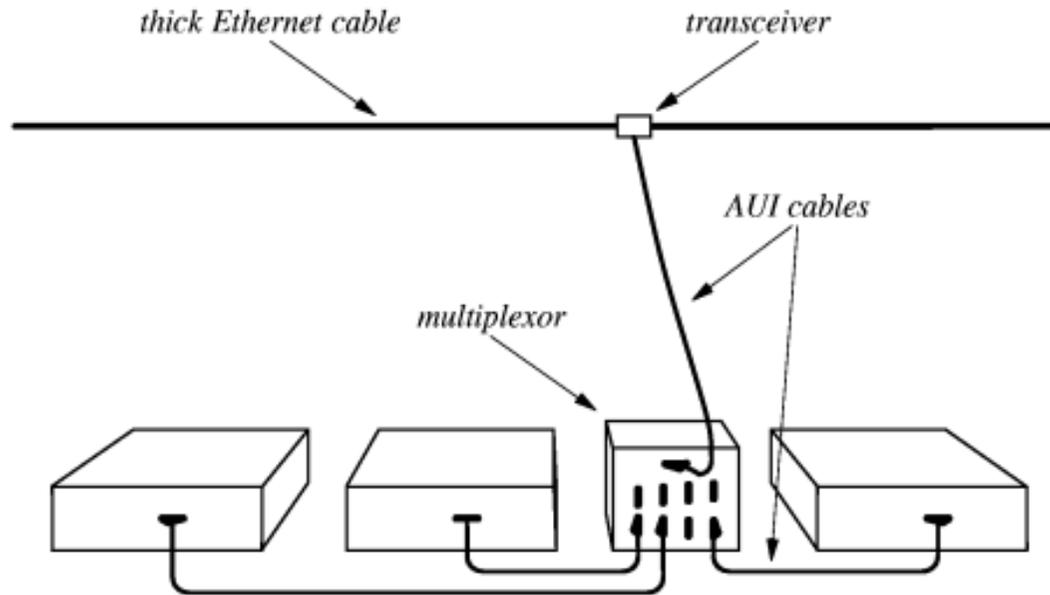
The Network Interface Card is an Input / Output device attached to the computer.

It communicates with the computer using Direct Memory Access (DMA).

The physical network address, called MAC address (for Media Access Control) address, is a 48-bit address that identifies the NIC, not the computer.

# Attaching a Computer to a Network

This is a typical attachment that uses the original (Thicknet) wiring.



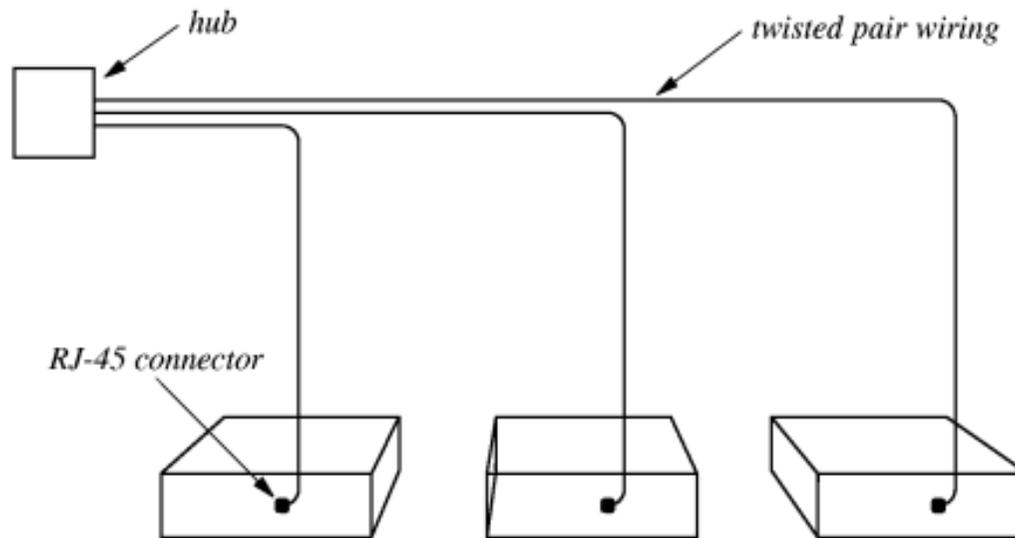
A number of devices in a single room would be connected through a multiplexor to the cable (called an “Ethernet cable”, after its protocol) through a transceiver (called an “AUI” for “Attachment Unit Interface”)

The AUI was typically placed above a false ceiling, making it hard to locate and repair a malfunctioning unit.

## Attaching a Computer to a Network (Part 2)

Current technology for attaching computers to a network uses a technology called **twisted pair wiring**. Names for this wiring include “10BaseT” and “100BaseT”.

The twisted pair cable attaches to the NIC through a **RJ-45 connector**.

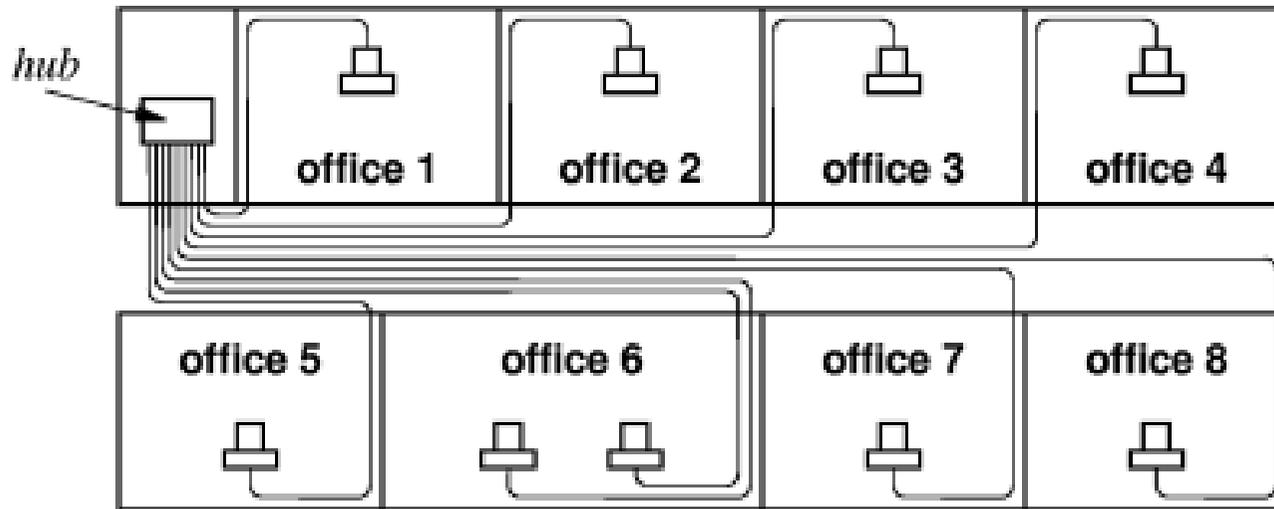


The **hub** connects these computers to the larger network.

The mathematicians in the class will note that this is a star topology and not the expected ring topology, supposedly used for networks. It is treated as if it were a ring.

## Typical Group Setup for Twisted-Pair Networks

Here is a typical group setup in which a number of computers are connected to the Internet through a hub.



The hub is often contained in a “**network closet**”, which is a small locked room.

This facilitates maintaining and securing the network assets.

The only part of the network above the ceiling is a collection of twisted-pair wires, which are usually quite reliable. In any case they are easy and cheap to replace.

# Connection–Oriented vs. Datagram Networks

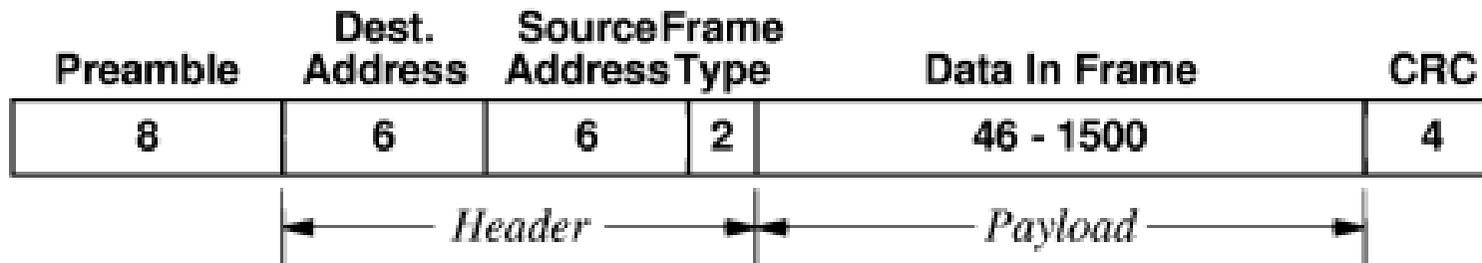
The Internet is a bit unusual in that it is not a connection–oriented network.

The U.S. telephone network is the best example of a connection–oriented network.

One dials a phone number, establishes a connection, and then keeps that connection open for the duration of the conversation. This might include a lot of time “on hold”.

Most networks break the messages and other data being passed into a number of **packets**, also called “**datagrams**”. Each packet can be routed independently from the source to the destination, leading to a great flexibility in the network.

At the physical level, a packet is embedded in an Ethernet **frame** as the frame payload. It has a 48–bit MAC address for the source node and destination node, a frame type, and a CRC for error detection. For IP version 4, the frame type is 0x0800.



# The IP (Internet Protocol)

The current version is IP, version 4. The next version will be called IP, Version 6. It was originally called “IP – the Next Generation”. Guess why?

IP is responsible for communication between computers. It does not connect applications on those computers, as do the next few protocols.

An IP datagram will be placed within an Ethernet frame for transmission over the net.



Remember that the frame header has a flag set to 0x0800 for encapsulated IP.

The IP header contains a protocol flag that indicates the datagram type.

Typical values are	1	ICMP	(Internet Control Message Protocol)
	6	TCP	(Transmission Control Protocol)
	17	UDP	(User Datagram Protocol)

# End-to-End Protocols

IP connects computers, but cannot determine the true end points of a connection, which must be two applications.

The two major end-to-end protocols are TCP and UDP.

The main difference between the two has to do with the reliability of the communication.

TCP is characterized by:

1. **Reliable transport.** TCP guarantees that the data sent across the connection will be delivered exactly as sent, with no data missing or out-of-order.

This implies retransmission of lost packets and rearrangement of those that are delivered out of order.

2. **Connection Oriented** Although it uses datagrams, TCP establishes a connection between two applications that persists for the duration of the session.

UDP is characterized as a “**best effort**” protocol, with no guarantees of packet delivery.

Consider transmission of music over the Internet. In this case, efficiency is important and occasional dropped packets are only a minor nuisance.

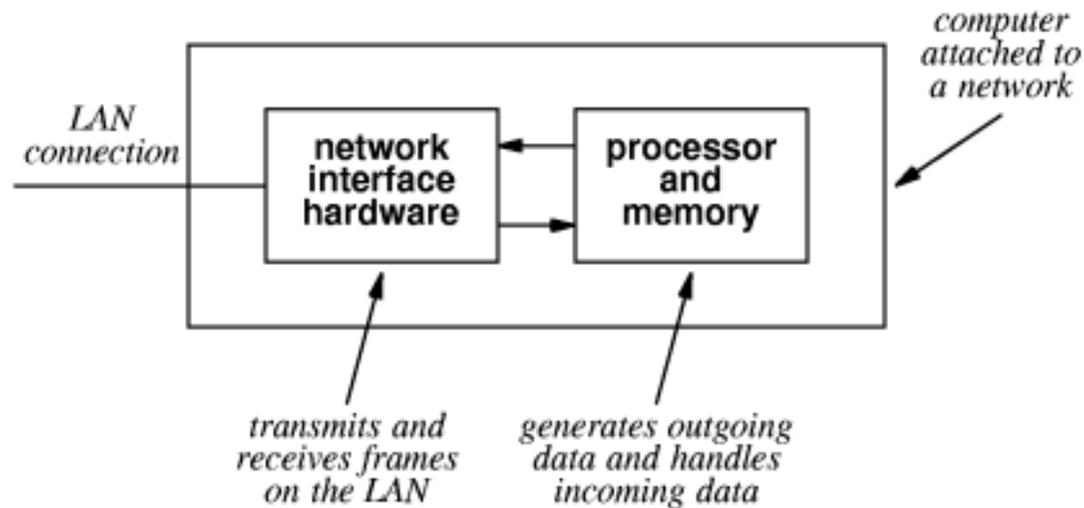
## More on the NIC

As noted above, the NIC is the unit that is actually attached to the network.

For the attached computer, it acts as a DMA Input / Output device.

Each NIC has a unique 48-bit MAC (Media Access Control) physical address.

These are designed to be globally unique, and are administered by the IEEE.



In **standard mode**, a NIC scans frames being sent on the network but stores only those frames with its MAC address as a destination.

In promiscuous mode, it will process any frame on the network. This is used by devices, such as **network sniffers**, used to diagnose networks or spy on them.

## The NIC Takes a Message

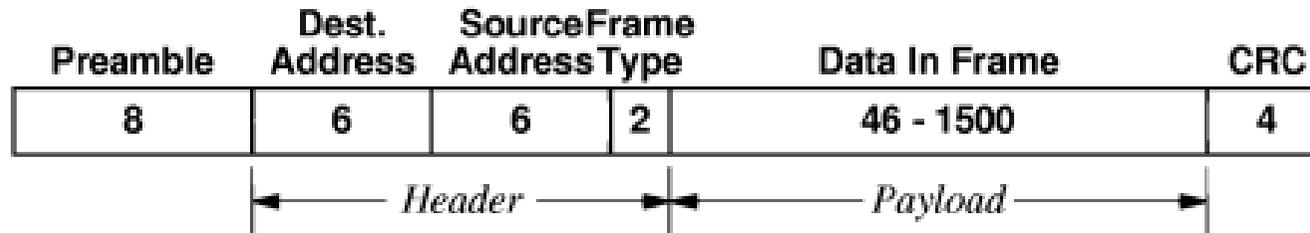
When the NIC has copied an entire frame from the network, the I/O sequence follows the standard DMA process.

1. The NIC asserts an interrupt to the CPU.
2. The CPU sends an ACK to the NIC.
3. The NIC places its **vector** on the I/O data lines.
4. The interrupt handler uses the vector to locate and start the interrupt handler appropriate for the NIC.
5. The interrupt handler sends the NIC a byte count (usually the Ethernet frame size) and a starting physical address in memory. It then commands the NIC to start data transfer and assert an interrupt when the input has been finished.
6. At the end of DMA, possibly due to an error, the NIC again interrupts the CPU. This interrupt is processed much as above.
7. The operating system then examines the frame to determine the type of service to be associated with the frame.

## The O/S Takes a Message

In response to the “DMA done” interrupt issued by the NIC, the operating system schedules the appropriate utility program to examine the frame.

This utility extracts the frame type, and determines that it is 0x0800.



The frame type indicates that the payload is an IP version 4 datagram.

The O/S interrupt handler cannot call IP directly. It places the payload into a dedicated message queue and then uses a message passing primitive to signal IP.

Note that the term “IP” is used in two contexts

1. The protocol dictating how the computers will communicate, and
2. The actual software that implements that protocol.

## IP Takes a Message

When the IP program processes the datagram, it extracts the service type from the IP header. The IP header has quite a few fields, shown in this figure.

0	4	8	16	19	24	31
<b>VERS</b>	<b>H. LEN</b>	<b>SERVICE TYPE</b>	<b>TOTAL LENGTH</b>			
<b>IDENTIFICATION</b>			<b>FLAGS</b>	<b>FRAGMENT OFFSET</b>		
<b>TIME TO LIVE</b>		<b>TYPE</b>	<b>HEADER CHECKSUM</b>			
<b>SOURCE IP ADDRESS</b>						
<b>DESTINATION IP ADDRESS</b>						
<b>IP OPTIONS (MAY BE OMITTED)</b>					<b>PADDING</b>	
<b>BEGINNING OF DATA</b> ⋮						

The version number will be either 4 or 6 for the next little while.

The protocol type is stored in the field called “TYPE”. It has one of several values:

- 1 ICMP
- 6 TCP
- 17 UDP

# Protocol Ports

Each of UDP and TCP is an end-to-end protocol; that is, it connects two application programs (presumably of the same type).

The standard way of identifying an application is known as a protocol port.

For example, the application to handle HTTP will associate itself with protocol port 80. Thus, a datagram will come to the computer with the message “I am a HTTP packet, please route me to whatever application processes HTTP”.

All protocol ports are defined to be 16-bit unsigned integers; 0 to 65,535 inclusive.

The first 1,024 ports (numbered 0 through 1,023) are defined as **well-known protocol ports**, defined by a standard called RFC 793.

Some of the well-known protocol port numbers are as follows.

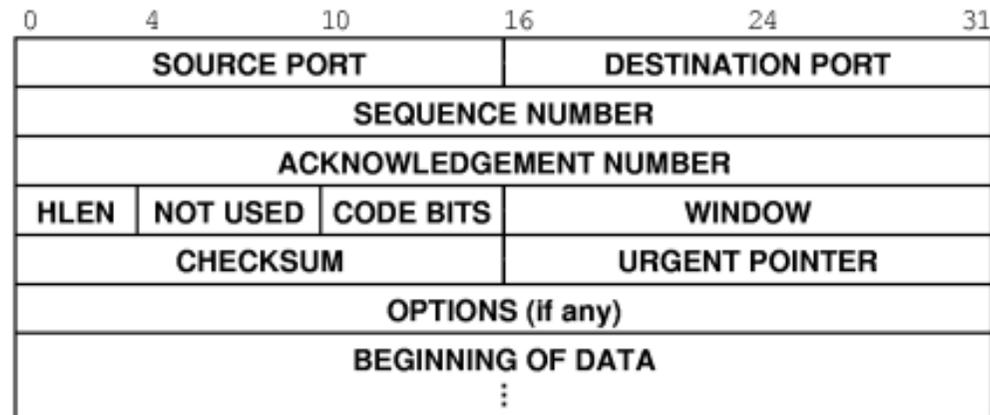
20	FTP	File Transport Protocol Data
21	FTP	File Transport Protocol Control
23	Telnet	
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name System
80	HTTP	Hypertext Transfer Protocol

## TCP Takes A Message

Suppose that IP determines that the datagram contains a TCP packet.

TCP calls the packet a **segment**.

Here is the format of a TCP header.



TCP processes this header, extracting the destination port and determining the application to receive the data contained in the segment.

TCP then places the segment in the appropriate queue and uses an operating system primitive called a “semaphore” to signal the application that it has data.