Unit 01.01.01

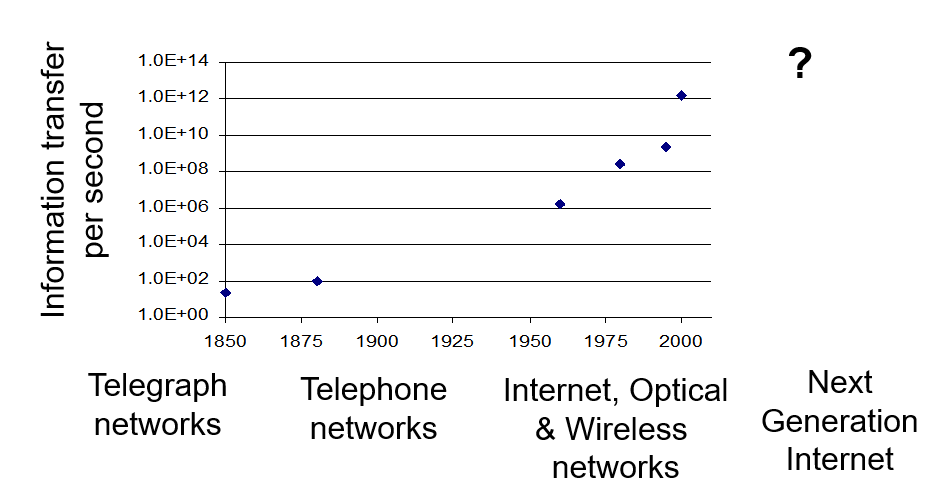
**Evolution of Communication Networks**

**What is a Communication Network?**

The equipment (hardware & software) and facilities that provide the basic communication service.

+ Facilities: Copper wires, optical fiber …

+ Equipment: Routers, servers, switches, …

**Network Architecture Evolution**

**Telegraph Networks**

Telegraph: a message is transmitted across a network using signals

+ Drums, beacons, mirrors, smoke, flags, semaphores…

+ Electricity, light

**Digital Communications**

Morse code converts text message in sequence of dots & dashes

Use transmission system designed to convey dots and dashes

**Electric Telegraph Networks**

Electric telegraph networks exploded

+ Message switching & Store-and-Forward operation

+ Key elements: Framing, Multiplexing, Addressing, Routing, Forwarding

Switches

Message

Destination

Source

Message

Message

Message

**Elements of Telegraph Networks**

**Digital transmission**

+ Text messages converted into symbols

+ Transmission system designed to convey symbols

**Multiplexing**

+ Framing needed to recover text characters

**Message Switching**

+ Messages contain source & destination addresses

+ Store-and-Forward: messages forwarded hop-by-hop across network

+ Routing according to destination address

**Bell’s Telephone**

Alexander G. Bell (1876) working on harmonic telegraph to multiplexing discovered voice signals can be transmitted directly

+ Microphone converts voice pressure variation into analogous electrical signal

+ Loudspeaker converts electrical signal back into sound

Basic telephone service involves two-way, real-time transmission of voice signals across a network

+ Signaling required to establish a call

Signaling + voice signal transfer

**The N2 Problem**

Initially, p2p direct communications - for N users to be fully connected directly

+ Requires too much space for cables

+ Inefficient & costly since connections not always on

**Circuit Switching is Connection-oriented**

Patchcord panel switch invented in 1878

Operators connect users on demand

+Establish circuit to allow electrical current to flow from inlet to outlet

Only N connections required to central office



**1**

**2**

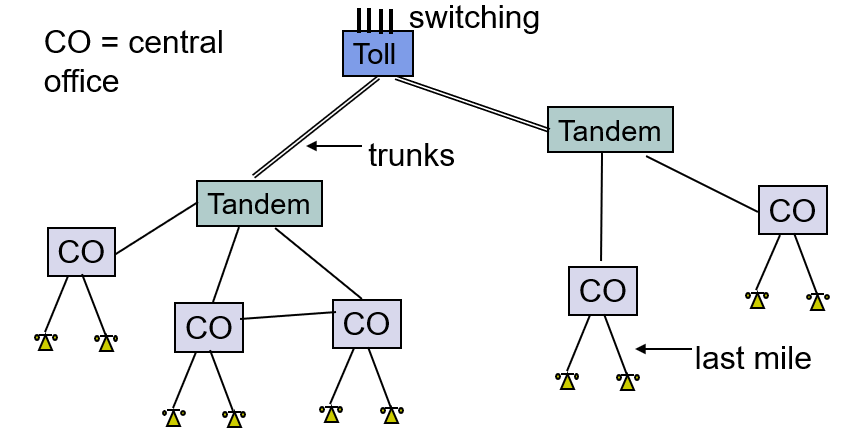
**3**

***N – 1***

***N***

**Hierarchical Tele-Network Structure**

End-to-end connection requires collaborative switching



**Elements of Telephone Networks**

**Digital transmission & switching**

+ Digital voice; Time Division Multiplexing

**Circuit switching – Connection oriented**

+ User signals for call setup and tear-down

+ Route selected during connection setup

+ End-to-end connection across network

+ Signaling coordinates connection setup

**Hierarchical Network Structure**

+ Decimal numbering system

+ Hierarchical structure; simplified routing; scalability

Unit 01.01.01

**Computer Network Evolution**

**1960s: Terminals access shared host computer**

+ SAGE; SABRE airline reservation system

+ Tree-topology terminal-oriented networks

**1970s: Computers connect directly to each other**

+ARPANET packet switching network

+ TCP/IP Internet protocols

+ Ethernet local area network

**1980s - 2000s: New applications and Internet growth**

+ Commercialization of Internet

+ E-mail, file transfer, web, P2P, . . .

+ Internet traffic surpasses voice traffic

**Terminal-Oriented Networks**

Early computer systems very expensive; Time-sharing methods allowed multiple terminals to share local computer

Remote access via telephone modems



Host computer

Terminal

Terminal

**. . .**

Terminal

Modem

Modem

Telephone

Network

**Medium Access Control**

Dedicated communication lines were expensive

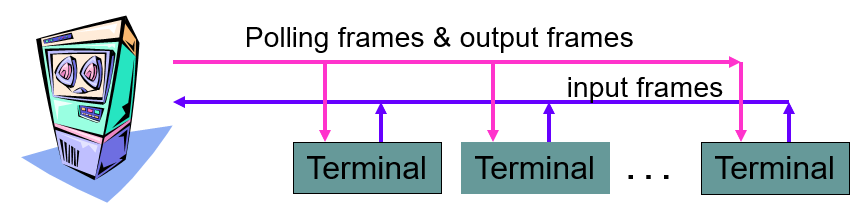
Terminals generated messages sporadically

Frames carried messages to/from attached terminals

Address in frame header identified terminal

Medium Access Controls for sharing a line in arbitrated manner

Example: Polling protocol on a multi-drop line



**Protocols**

A protocol is a set of precise & unambiguous rules that governs

+ how two or more communicating entities in a layer are to interact

+ Messages that can be sent and received

+ Actions that are to be taken when a certain event occurs

***The purpose of a protocol is to provide a service to the layer above***

**HTTP**

+ HTTP is an application layer protocol

+ Retrieves documents on behalf of a browser application program

+ HTTP specifies fields in request messages and response messages

Request types; Response codes

Content type, options, cookies, …

+ HTTP specifies actions to be taken upon receipt of certain messages

+ HTTP server waits for requests by listening to a well-known port number (80 for HTTP)

**DNS Protocol**

+ DNS protocol allows queries of different types

+ DNS protocol is an application layer protocol

+ DNS is a distributed database that resides in multiple machines in the Internet

+ DNS protocol allows queries of different types

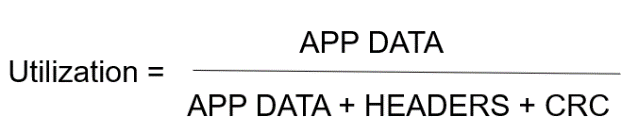
+ DNS usually involves short messages and so uses service provided by UDP

+ Well-known port 53

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Unit 01.02.02

**OSI Unified View of Protocols and Services**

**Bandwidth Utilization**

**Encapsulation in TCP/IP**

TCP Header contains source & destination **port numbers**

IP Header contains source and destination **IP addresses**; transport protocol type

Ethernet Header contains source & destination **MAC addresses**; network protocol type

**Connectionless & Connection-Oriented Services**

**Connection-Oriented**

Three-phases:

+ Connection setup between two SAPs to initialize state information

+ SDU transfer

+ Connection release

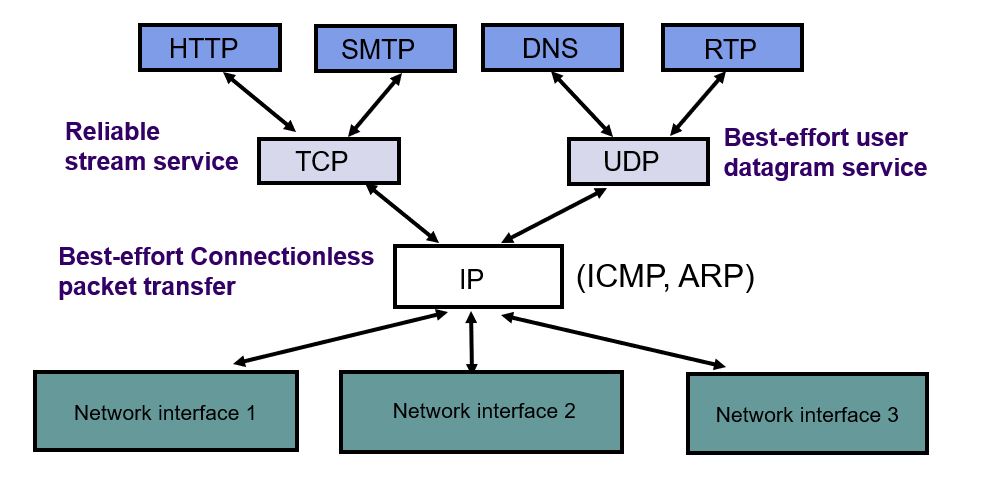
**Connectionless**

Immediate SDU transfer

No connection setup

E.g. UDP, IP

E.g: TCP, ATM



**Physical Addresses**

IP address needs to be resolved to physical address at each IP network interface, by **address resolution protocol (ARP)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Unit 01.03.01

**Berkeley Socket API - I**

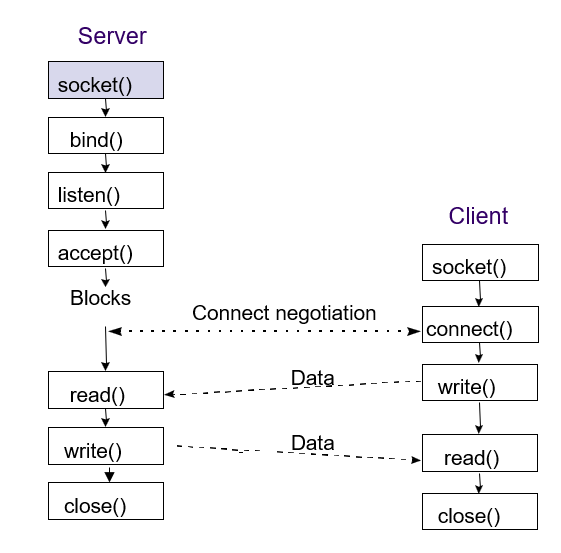
**Transport Protocols**

**process-to-process** communications

User Datagram Protocol (UDP) enables best-effort connectionless transfer of individual **block of information**

Transmission Control Protocol (TCP) enables connection-oriented reliable transfer of a **stream of bytes**

Two services though Sockets: connection-oriented and connection-less

**Socket Calls for Connection-Oriented Mode**

**Server** does Passive Open

+ socket call creates socket to listen for connection requests

+ Server specifies type: TCP (stream)

+ socket call returns: non-negative integer descriptor;

or -1 if unsuccessful

bind assigns local address & port # to socket with specified descriptor

Can wildcard IP address for multiple net interfaces

bind call returns: 0 (success); or -1 (failure)

listen indicates to TCP readiness to receive connection requests for socket with given descriptor

listen call returns: 0 (success); or -1 (failure)

Server calls accept to accept incoming requests, accept blocks if queue is empty

---------------------------------

**Client** does Active Open

+ socket call creates socket to connect to server

+ Client specifies type: TCP (stream)

+ socket call returns: non-negative integer descriptor; or -1 if unsuccessful

connect establishes a connection on the local socket with the specified descriptor to the specified remote address and port #

connect returns 0 if successful; -1 if unsuccessful

accept wakes with incoming connection request

accept fills client address & port # into address structure

accept call returns: descriptor of new connection socket (success); or -1 (failure)

Client & server use new socket for data transfer

Original socket continues to listen for new requests

**Data Transfer**

1) Client or server call write to transmit data into a connected socket

write call returns: # bytes transferred (success); or -1 (failure); blocks until all data transferred

2) Client or server call read to receive data from a connected socket

read specifies: socket descriptor; pointer to a buffer; amount of data

read call returns: # bytes read (success); or -1 (failure); blocks if no data arrives

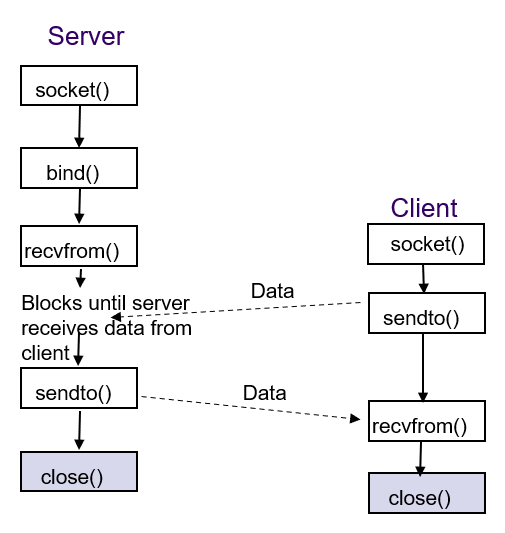
**Connection Termination**

Client or server call close when socket is no longer needed

close specifies the socket descriptor

close call returns: 0 (success); or -1 (failure)

**Socket Calls for Connection-Less Mode**

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Unit 01.03.03

**Digital Communication Fundamentals**

The delay of communication between two nodes has two components, the propagation delay and the transmission delay

**Delay – Propagation Delay**

*tprop = d/v*

*t*prop time for signal to propagate across medium

*d* distance between two nodes in meters

*v* speed of light in the transmission medium (3x108 m/s in vacuum)

**Delay - Transmission Delay**

*ttrans = L/R*

*L* number of bits in message

*R* bandwidth of digital transmission system in **bps**

**Overall Delay = *tprop + ttrans = d/v + L/R***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Unit 01.04

**Error Control**

**Parity check**

Many transmission channels introduce bit errors at random, independently of each other, and with **probability p**

For a n-bit frame,

***P* [1-bit error] = *p* (1 – *p*) n-1**

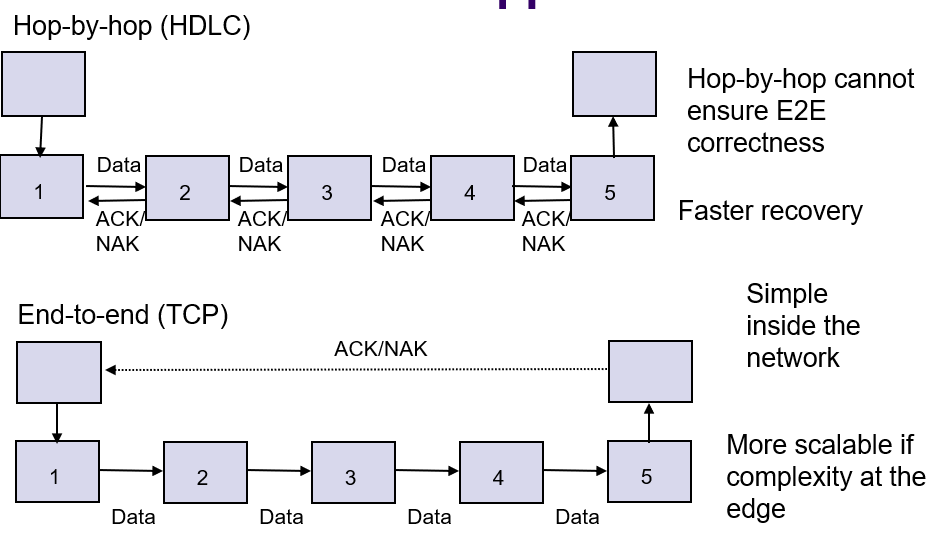
P [10000000] = p(1 – p)7 = (1 – p)8p/(1-p)

P [11000000] = p2(1 – p)6 = (1 – p)8[p/(1-p)]2

***P* [ j-bit error] = *p* j (1 – *p*) n-j**

**Review: slide 42, 43, 44 – C1 M4**

Unit 02.01

**Peer-to-Peer Protocols and Services**

**Automatic Repeat Request (ARQ)**

**Purpose**: to ensure a sequence of information packets is delivered in order and without errors or duplications despite transmission errors & losses

**Sliding window:** a set of Seq.# corresponding to frames permitted to send or receive.

**Three ARQ protocols**

+ Stop-and-Wait ARQ

+ Go-Back N ARQ

+ Selective Repeat ARQ

**Basic elements of ARQ:**

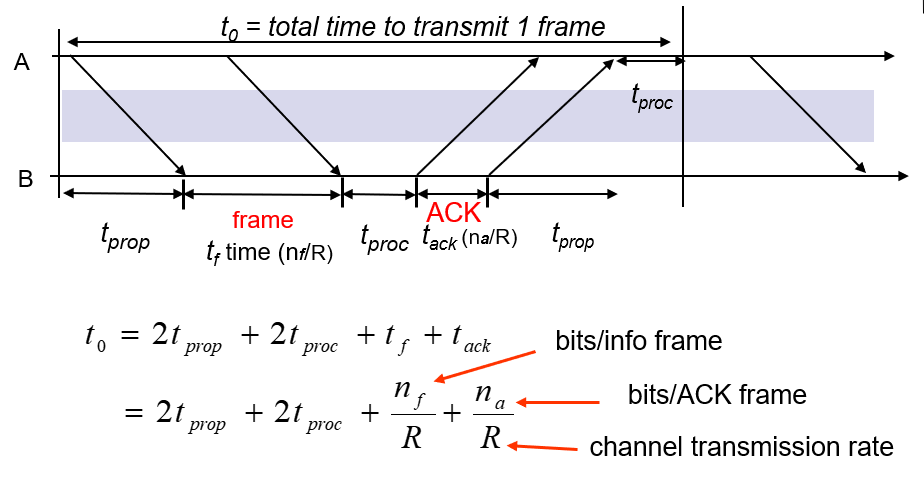
+ Error-detecting code with high error coverage

+ ACKs (positive acknowledgments)

+ NAKs (negative acknowledgments)

+ Timeout mechanism

**Stop-and-Wait Delay model**

****

**Delay-Bandwidth Product**

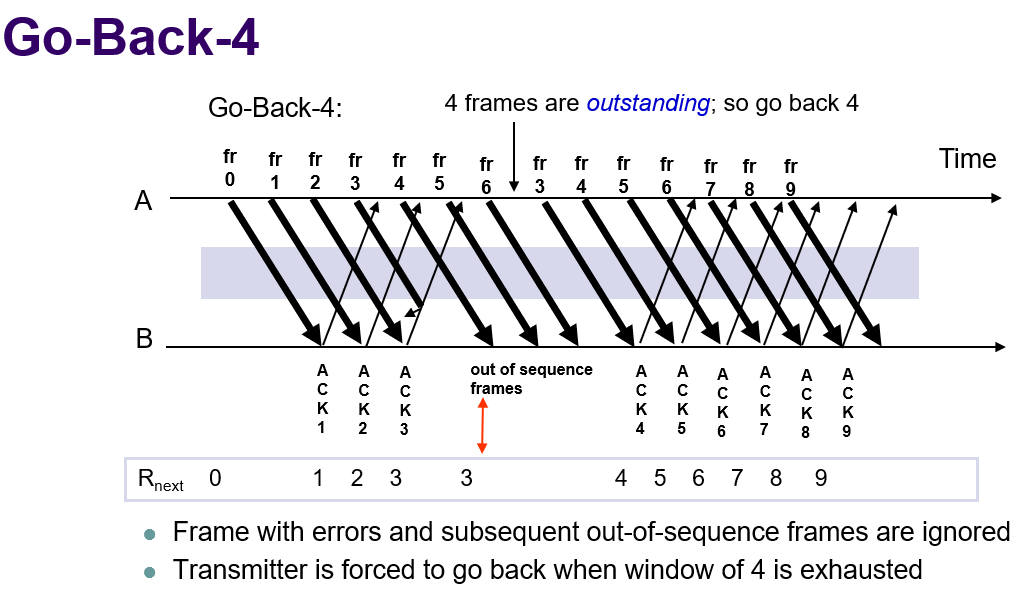
2( tprop + tproc ) \* R, or RTT \* R

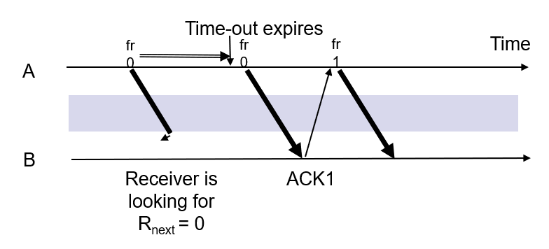
**Go-Back-N ARQ**

+ Improve Stop-and-Wait by not waiting!

+ Keep channel busy by continuing to send frames

+ A procedure where the transmission of a new frame is begun before the completion time of the previous frame transmission is said to be **pipelining**

****

****

**Go-Back-N with Timeout**

When timeout expires, resend all outstanding frames

**Given m-bit seq. numbers, what is the maximum number of frames that can be outstanding in “go back N”?**

MAX\_SEQ = 2^m – 1 while there are 2^m sequence numbers.

*Maximum Allowable Window Size is Ws = 2m-1*

**Piggybacking and Bidirectional Links**

Piggybacking: receiver inserts ACK in the next departing frame

**Selective Repeat ARQ**

Selective Repeat retransmits *only an individual frame*

+ Timeout causes individual corresponding frame to be resent

+ NAK causes retransmission of oldest un-acked frame

**End-to-End Packet Network Topology** (Cấu trúc liên kết mạng gói End-to-End)

In Packet networks,

**+ Individual packet streams are highly bursty**

Statistical multiplexing is used to concentrate streams

**+ User demand can undergo dramatic change**

Peer-to-peer applications stimulated huge growth in traffic volumes

**+ Internet structure highly decentralized**

Paths traversed by packets can go through many networks controlled by different organizations

No single entity responsible for end-to-end service

**Access Multiplexing**

Packet traffic from users multiplexed at access to network into aggregated streams

DSL traffic multiplexed at DSL Access Mux

Cable modem traffic multiplexed at Cable Modem Termination System

**Oversubscription is a commonly used technique in multiplexing.**

**Circuit switching vs. Message switching**

**TCP vs. ARQ protocol**

**TCP vs. HDLC**

**Single Parity Check**

Info Bits: b1, b2, b3, …, *bk*

Check Bit: bk+1= b1+ b2+ b3+ …+ b*k* modulo 2

Codeword: (b1, b2, b3, …, b*k*,, b*k*+!)

All codewords have even # of 1s

**Redundancy**: Single parity check code adds 1 redundant bit per m information bits: overhead = 1/(m + 1)

**Coverage**: all error patterns with odd # of errors can be detected

+ An error patten is a binary (m + 1)-tuple with 1s where errors occur and 0’s elsewhere

+ Of 2k+1 binary (m + 1)-tuples, ½ are odd, so 50% of error patterns can be detected

Code vector (e1, e2, …, en) where ei = 1 if an error occurs in the ith transmitted bit and ei = 0 otherwise

**Error Probability**

In [computer networking](https://en.wikipedia.org/wiki/Computer_networking), **network traffic control** is the process of managing, controlling or reducing the network traffic, particularly Internet [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(computing)), e.g. by the [network scheduler](https://en.wikipedia.org/wiki/Network_scheduler). It is used by network administrators, to reduce [congestion](https://en.wikipedia.org/wiki/Network_congestion), [latency](https://en.wikipedia.org/wiki/Lag) and [packet loss](https://en.wikipedia.org/wiki/Packet_loss).

This is part of [bandwidth management](https://en.wikipedia.org/wiki/Bandwidth_management). In order to use these tools effectively, it is necessary to [measure the network traffic](https://en.wikipedia.org/wiki/Network_traffic_measurement) to determine the causes of network congestion and attack those problems specifically.

**Traffic shaping** is the retiming (delaying) of [packets](https://en.wikipedia.org/wiki/Network_packet) (or [frames](https://en.wikipedia.org/wiki/Frame_(networking))) until they meet specified bandwidth and or [burstiness](https://en.wikipedia.org/wiki/Burstiness) limits. Since such delays involve queues that are nearly always finite and, once full, excess traffic is nearly always dropped (discarded), traffic shaping nearly always implies traffic policing as well.

**Traffic policing** is the dropping (discarding) or reduction in priority (demoting) of packets (or frames) that exceed some **specified bandwidth** and or **burstiness limit.**

**DHCP**

DHCP builds on BOOTP to allow servers to deliver configuration information to a host

+ Used extensively to assign temporary IP addresses to hosts

+ Allows ISP to maximize usage of their limited IP addresses

+ Time thresholds to enforce lease time

**Services provided by IP layer are addressing, fragmenting, packet timeouts**

**Given a generator polynomial g(x) = x^3 + x + 1. Consider the information sequence 1001. By CRC method, what is the resulted codeword for transmission?**

As the degree of generator function is 3, the information sequence is appended three bits of 0s. The resulting bit frame 1001000 is divided by 1011 in binary arithmetic, and the remainder is 110 (check bits). Codeword is 10001 appended by the check bits, which is 1001110.

**CRC is a very effective and popular error detection technique. The error detection capabilities of CRC depend on the chosen generator polynomial.**

* CRC has capacity to detect all single-bit errors.
* CRC has capacity to detect all double-bit errors (three 1’s).
* CRC has capacity to detect any odd number of errors (X + 1).
* CRC has capacity to detect all burst errors of less than the degree of the polynomial.
* CRC has the capacity to detect most of the larger burst errors with a high probability.
* The generator polynomial g(x) can detect any burst of error for which the length of the burst is less than or equal to

**Rule-01:**

* It should not be divisible by x.
* This condition guarantees that all the burst errors of length equal to the length of polynomial are detected.

**Rule-02:**

* It should be divisible by x+1.
* This condition guarantees that all the burst errors affecting an odd number of bits are detected.

**If the CRC generator is chosen according to the above rules, then-**

* CRC can detect all single-bit errors
* CRC can detect all double-bit errors provided the divisor contains at least three logic 1’s.
* CRC can detect any odd number of errors provided the divisor is a factor of x+1.
* CRC can detect all burst error of length less than the degree of the polynomial.
* CRC can detect most of the larger burst errors with a high probability.

**In go-back-N, the maximum number of frames that can be outstanding is 2^m – 1, where m is the number of bits in sequence number.**

**In Selective repeat, the sliding window size at the sender plus the sliding window size at the receiver must be equal to 2^m, where m is the number of bits in sequence number. In this case, the sender has window size 4, which means the receiver has window size 8 – 4 = 4.**

**For a 1-km cable, the one-way propagation time is 5** μ**sec, therefore, round-trip delay 2**τ = **10** μ**sec. To make CSMA/CD work, it must be impossible to transmit an entire frame in this interval. At 1 Gbps, all frames shorter than 10,000 bits can be completely transmitted in under 10** μ**sec, so the minimum frame is 10,000 bits or 1250 bytes.**