THE IEEE LOCAL AREA NETWORK STANDARDS

A 1986 ACM National Lecture

Alfred C. Weaver, Ph.D.
University of Virginia

Computer Science Report No. TR-86-03
February 28, 1986

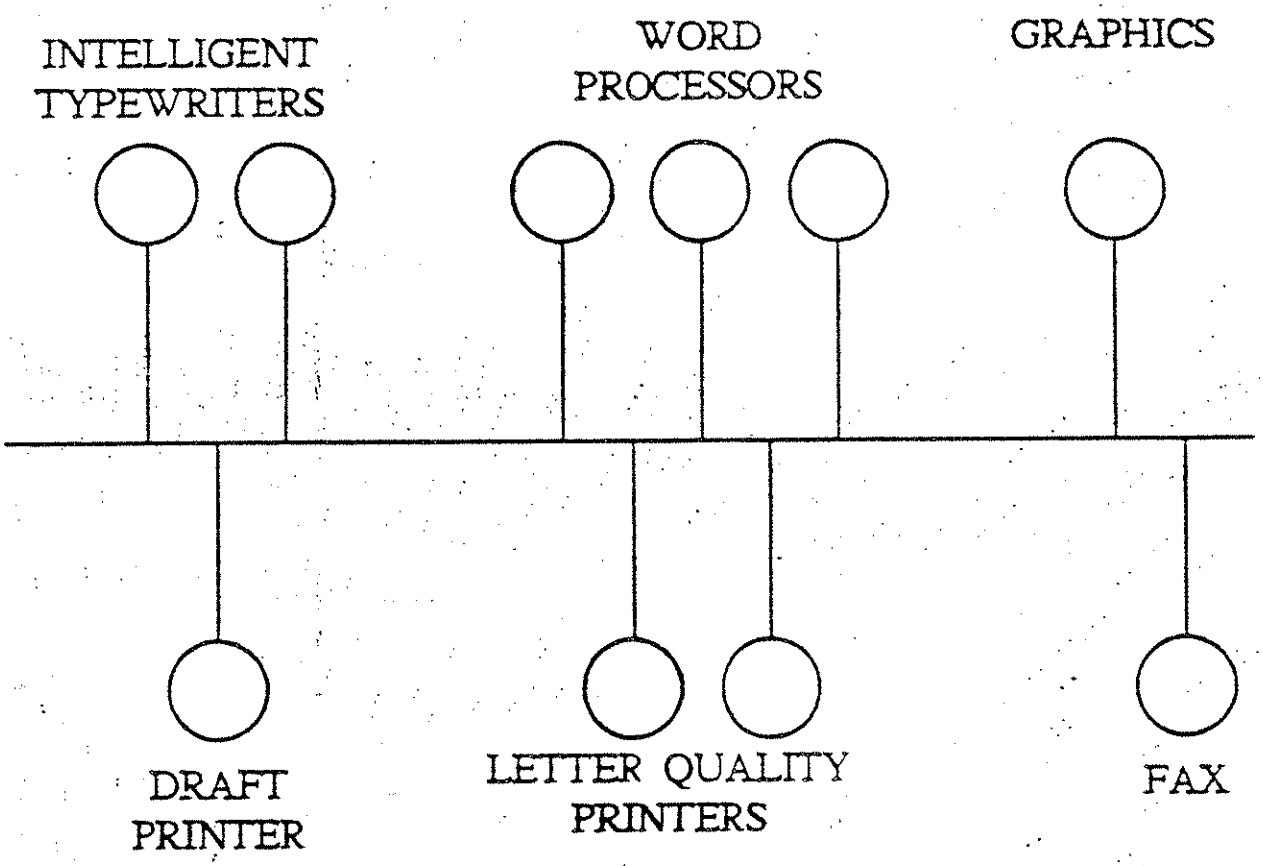
THE IEEE LOCAL AREA NETWORK STANDARDS

Alfred C. Weaver

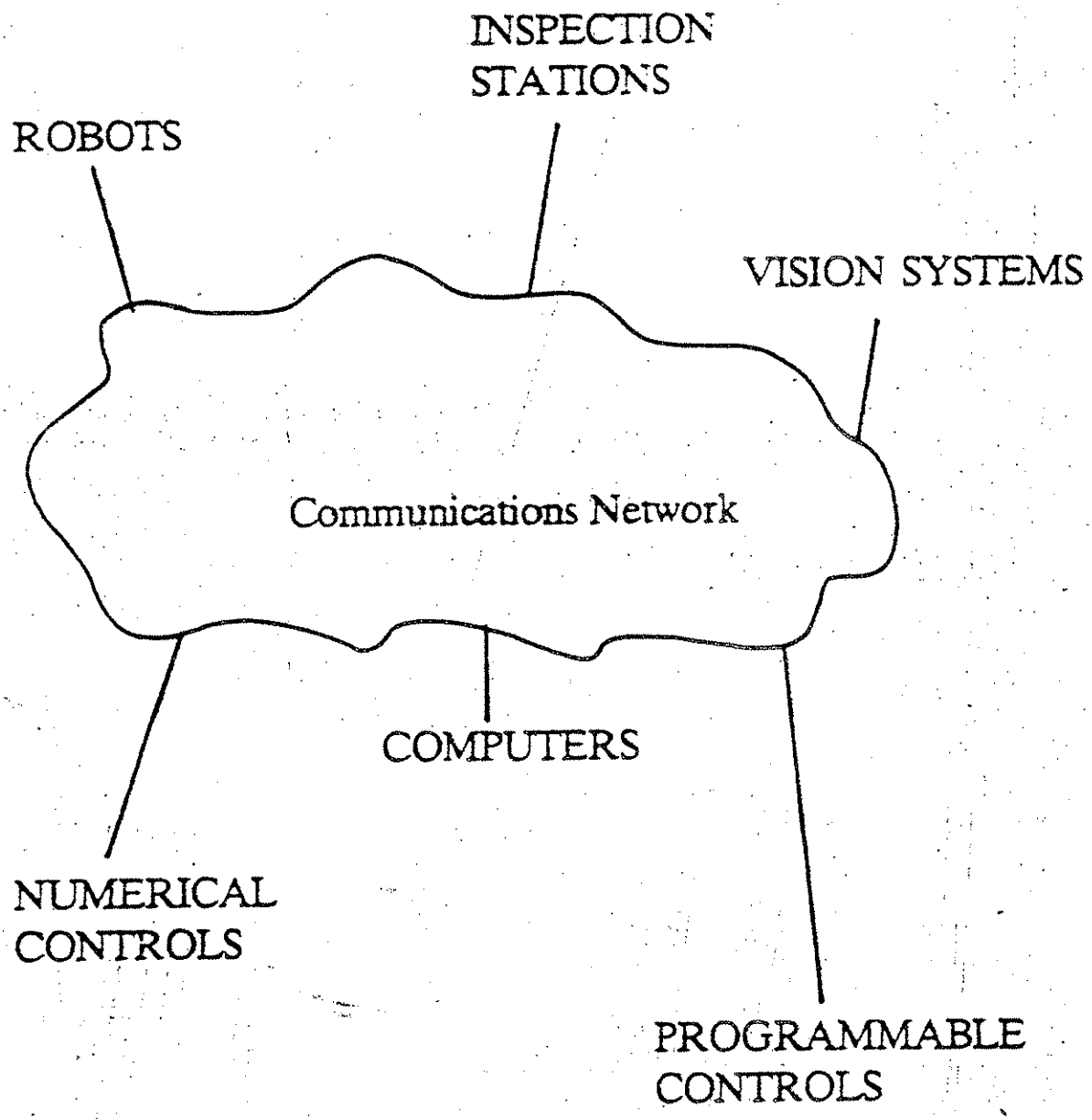
University of Virginia

(slide title)

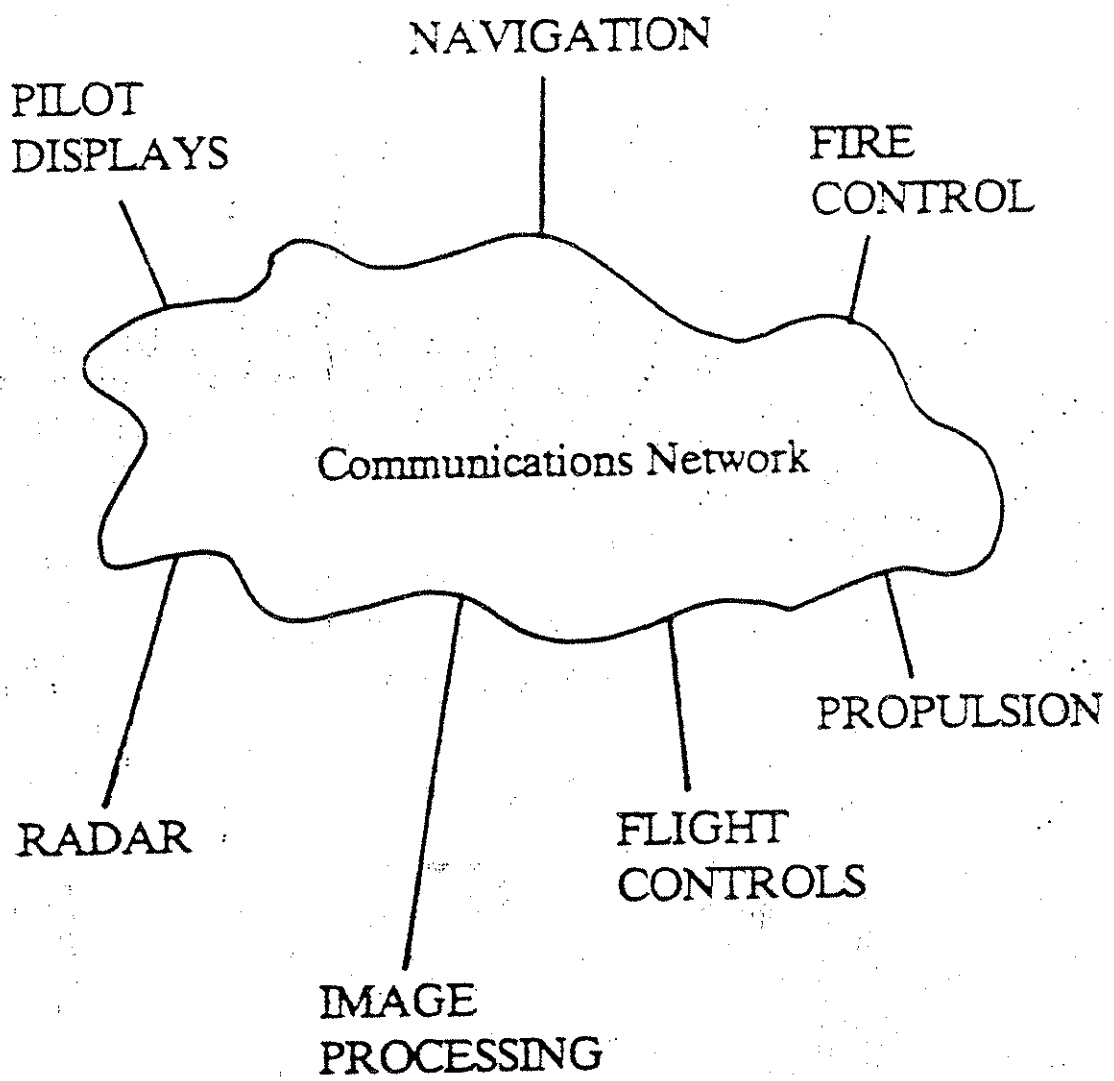
OFFICE AUTOMATION



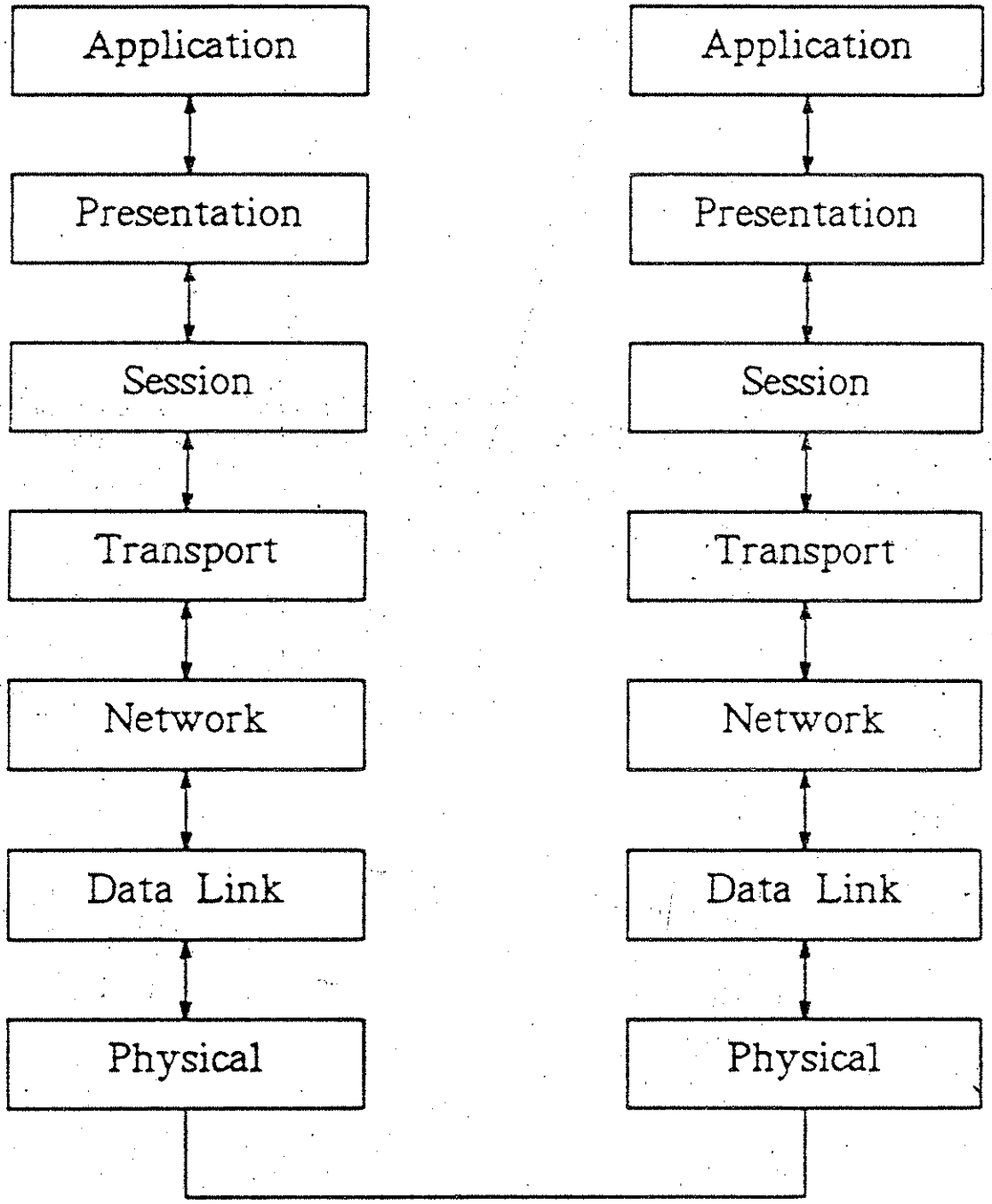
FACTORY AUTOMATION



AIRCRAFT CONTROL



ISO OSI MODEL



IEEE COMMITTEE 802

Charged to develop a local area network standard

Committee membership predominately industrial

Recognized that application areas had different needs

- office automation
- factory automation
- real-time control systems

Impossible to agree on a single standard

- application areas are truly diverse
- mixed corporate membership inhibited agreement

802 SUBCOMMITTEES

802.1 -- Relationship to ISO OSI model

802.2 -- Logical Link Control of Data Link Layer

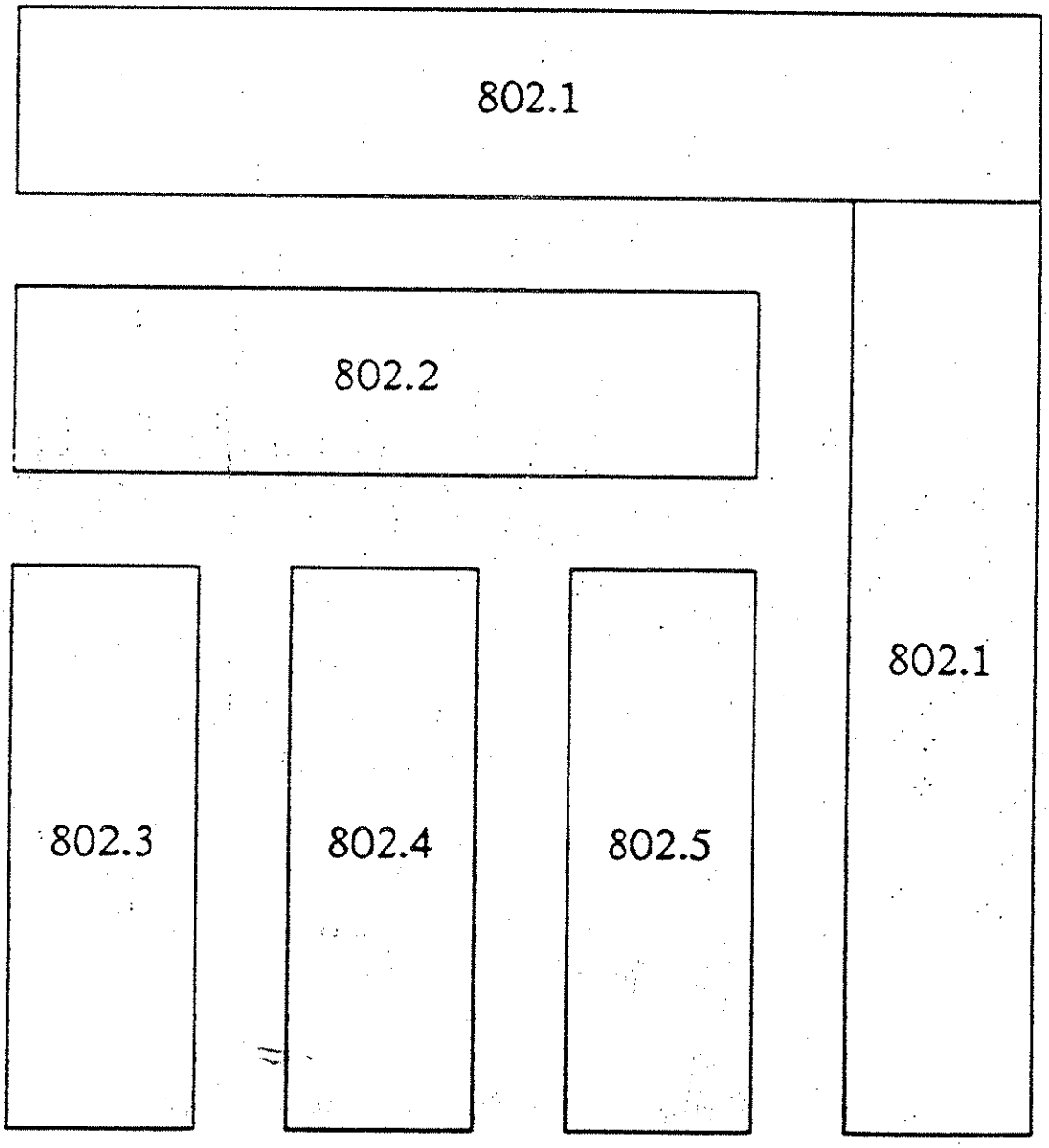
802.3 -- Contention Bus

802.4 -- Token Bus

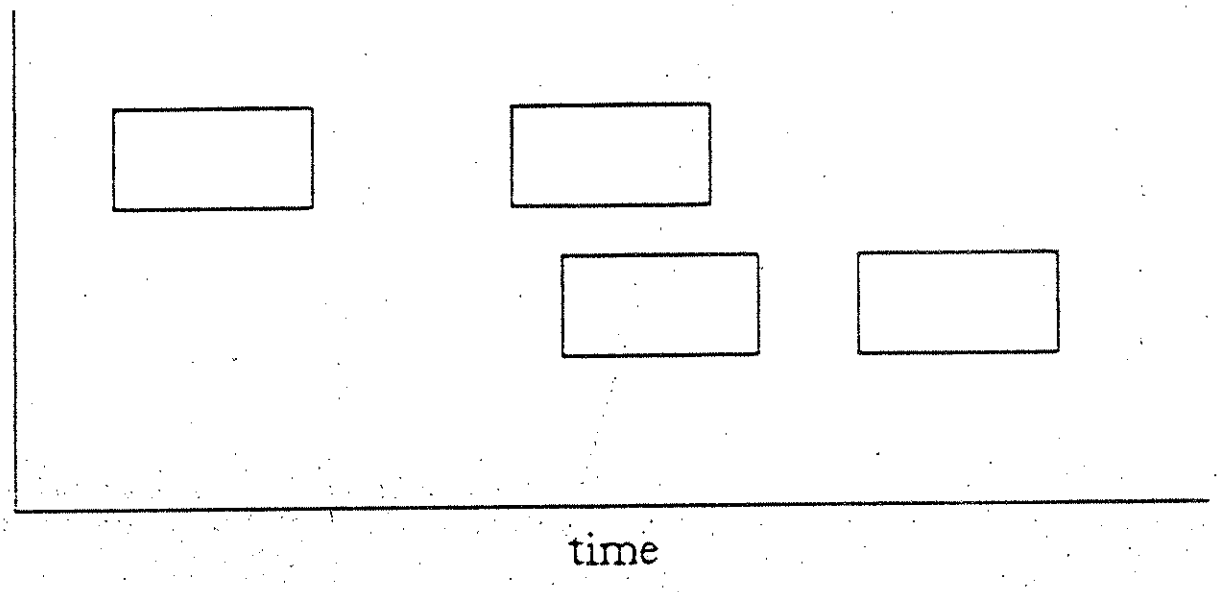
802.5 -- Token Ring

802.6 -- Metropolitan Networks

802.x RELATIONSHIPS



ETHERNET



Form of CSMA/CD

Uses 50 ohm baseband coax cable

Maximum bus capacity of 10 Mbps

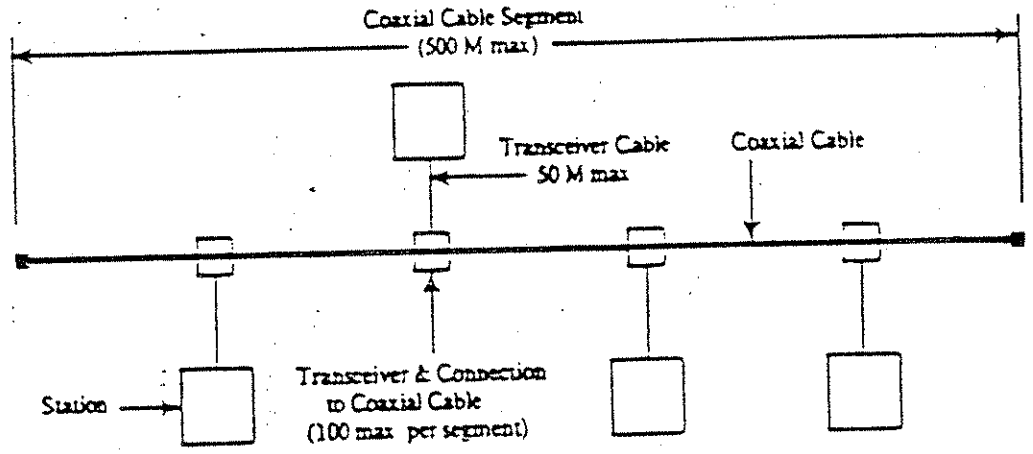
Maximum segment distance of 500 meters

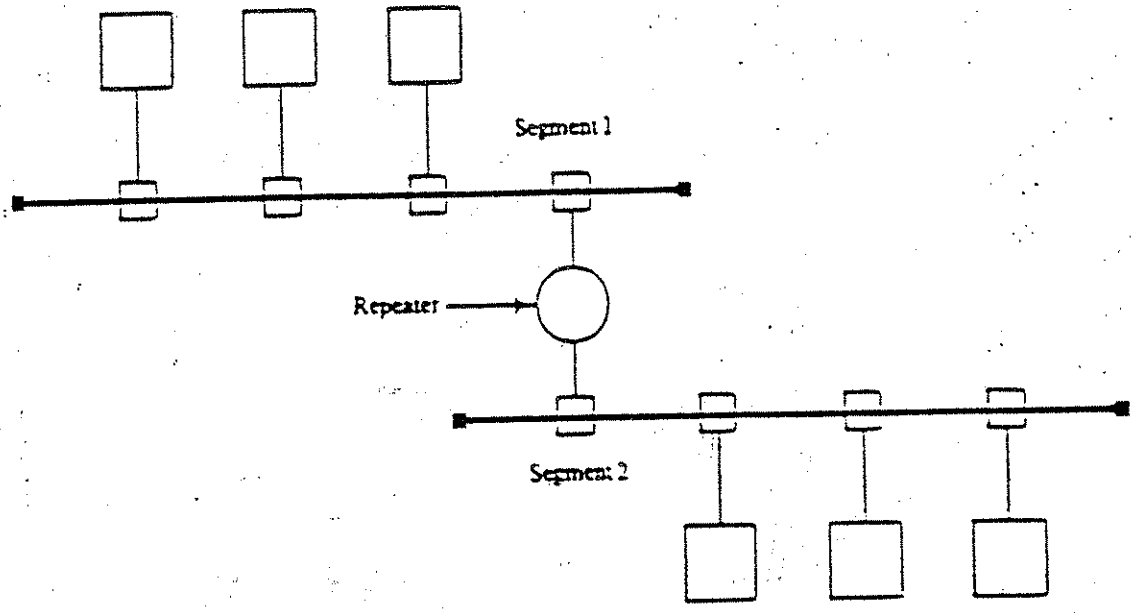
Maximum end-to-end separation of 2,500 meters

Maximum of 1,000 stations

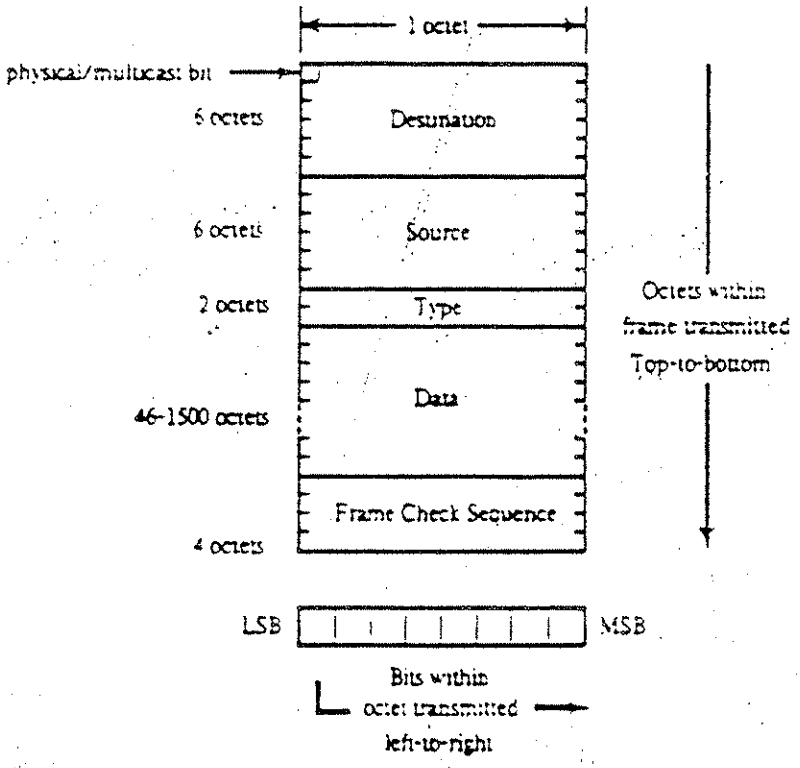
Uses coax taps

ETHERNET SEGMENTS

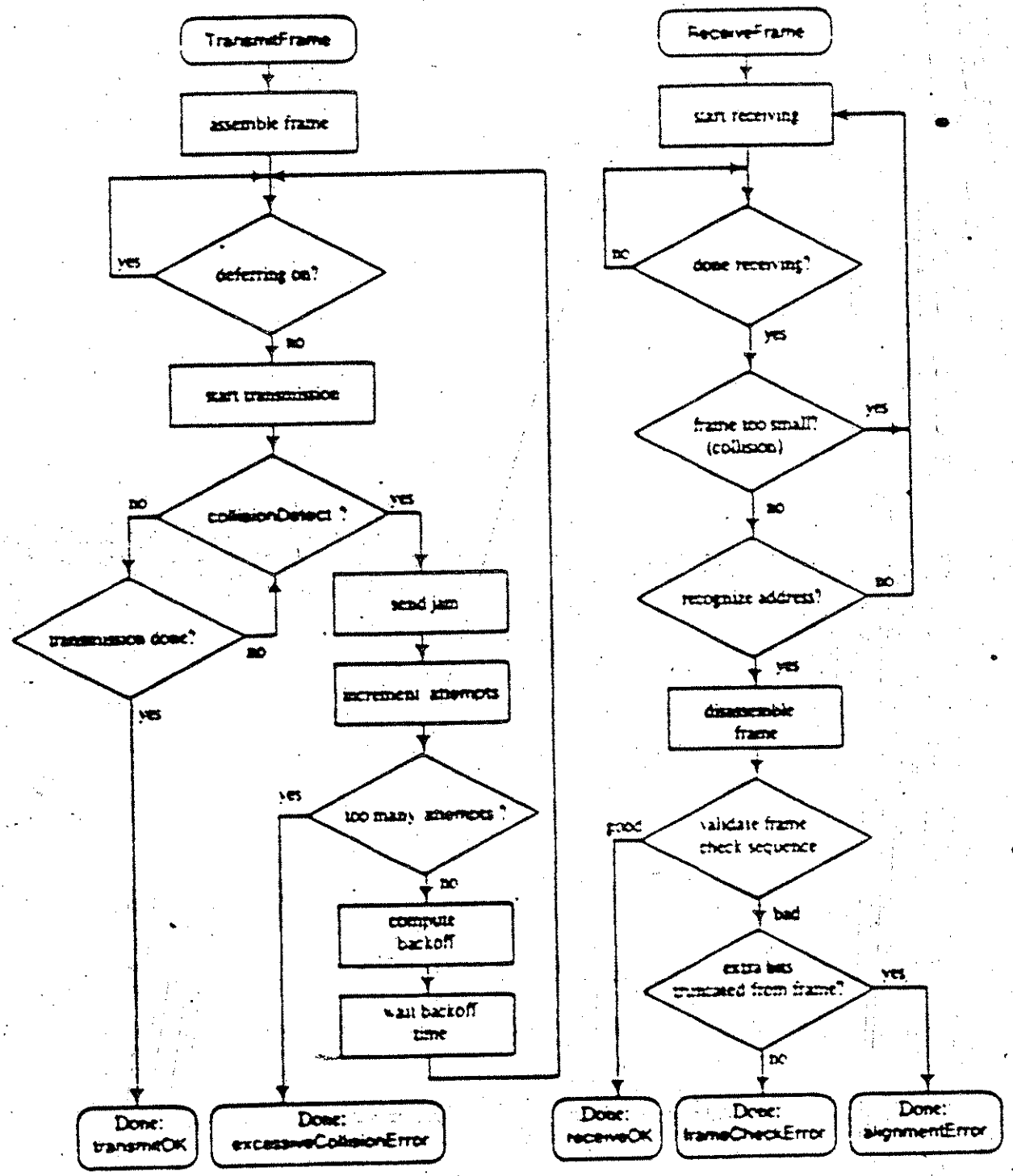




ETHERNET PACKET FORMAT



DATA LINK LAYER



BACKOFF

Ethernet: CSMA/CD, 1-persistent, binary exponential backoff

At the i^{th} collision:

- j := min (i, 10)
- k := random (0 .. 2^j)
- wait k slot times = $512 \cdot k$ bit times
- sense channel
- transmit when idle

WORST CASE DELAY

Retry mechanism: transmit, collide, jam, backoff

Worst case delay

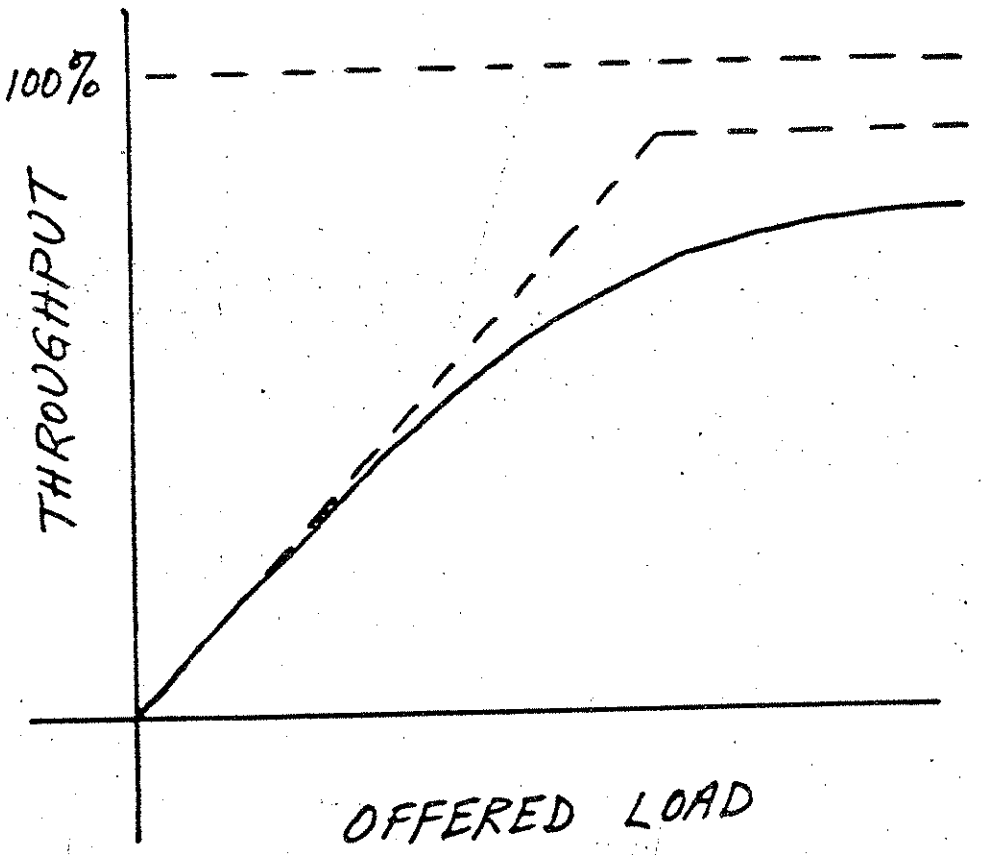
$$= \sum_{i=1}^{10} 2^i \cdot 512 + 5 \cdot 2^{10} \cdot 512 + 15 \cdot 48$$

$$= 512(2^{11} - 2) + 5 \cdot 1024 \cdot 512 + 720$$

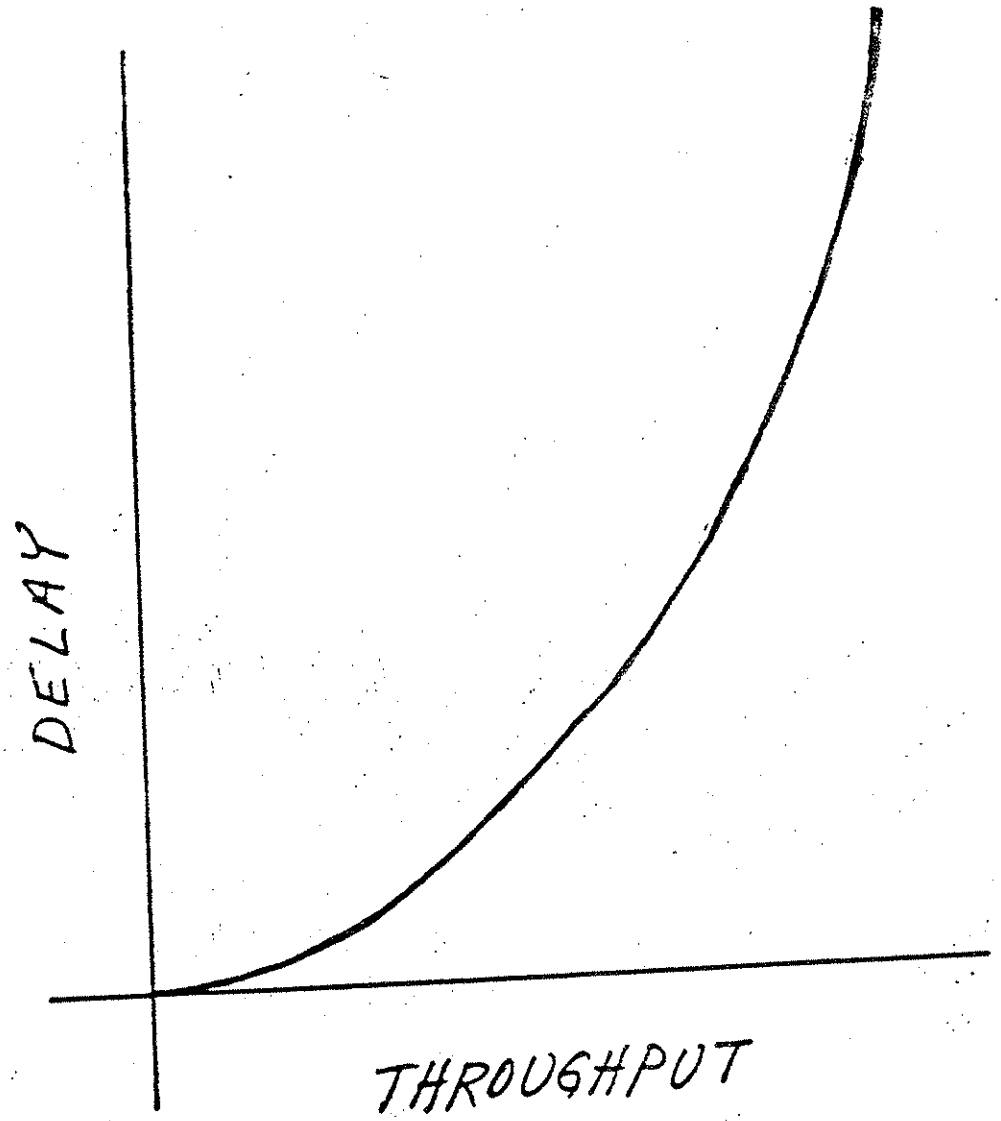
$$= 3{,}669{,}712 \ \textit{bittimes}$$

$$\approx 0.37 \ \textit{seconds}$$

ETHERNET THROUGHPUT



ETHERNET DELAY



ETHERNET SUMMARY

Designed for

- office automation environment
- bursty, asynchronous loads
- low average offered loads

Throughput

- is linear with offered load at low loads
- is stable and becomes asymptotic
- has maximum determined by propagation time and packet size

Delay

- is highly variable
- has large (0.37 sec) worst-case delay

IEEE 802.4 TOKEN BUS

802.4 SUMMARY

A *token* controls access to the physical medium

Token holder is momentarily the network master

Implements four priorities, or *access_classes*

Station must pass the token to a known successor within a bounded time

Orderly progression of the token from station to station forms a logical ring on a physical bus

Station's interface is a Medium Access Controller (MAC)

MAC implements the protocol, including

- token recognition, passing, and regeneration after loss
- message encapsulation and framing
- service of the four priorities
- error control and recovery

TOKEN PASSING

At startup, each station is assigned a unique logical address

During startup, stations add themselves to the logical ring one at a time, thereby learning their successor

A station may transmit only while it holds the token

When all data has been transmitted or certain timers expire, token must be passed to successor

A station will periodically query the network to determine whether additional stations wish to join the ring

Special cases to recover from loss of token, failure of successor, etc.

TOKEN

Standard allows a network-wide choice of either 16 or 48 bit addresses

Token is an explicit message of at least 96 or 160 bits depending upon address size (token frame can carry data)

Token consists of:

- preamble (1 or more octets)
- start delimiter (1 octet)
- control information (1 octet)
- destination address (2 or 6 octets)
- source address (2 or 6 octets)
- optional data (0 or more octets)
- frame check sequence (4 octets)
- end delimiter (1 octet)

MESSAGE FRAME

Message frame contains

- preamble (1 or more octets)
- start delimiter (1 octet)
- control information (1 octet)
- destination address (2 or 6 octets)
- source address (2 or 6 octets)
- data (0 or more octets up to maximum frame size of 8191 octets)
- frame check sequence (4 octets)
- end delimiter (1 octet)

ACCESS CLASSES

Four priorities or *access_classes*

- Synchronous
- Urgent Asynchronous
- Normal Asynchronous
- Time Available

Only *Synchronous* is guaranteed a level of service

Other classes receive "best effort"

An 802.4 station must implement either the *Synchronous* class alone or else all four classes simultaneously

Note that priority applies to a message, not a station

NETWORK PARAMETERS

Let the *Synchronous*, *Urgent Asynchronous*, *Normal Asynchronous*, and *Time Available access_classes* be abbreviated by S, UA, NA, and TA, respectively

High Priority Token Hold Time (HPTHT) -- the maximum amount of time a station may serve its *Synchronous* class

Urgent Asynchronous Target Rotation Time (TRT_{UA}) -- goal token rotation time for class UA

Normal Asynchronous Target Rotation Time (TRT_{NA}) -- goal token rotation time for class NA

Time Available Target Rotation Time (TRT_{TA}) -- goal token rotation time for class TA

STATION TIMERS

token_hold_timer (tht) — when it expires, station may complete sending the packet in progress, but must then sequence to the next lower priority *access_class* or, if serving class TA, must pass the token to the station's successor

$token\_rotation\_timer_{UA}(trt_{UA})$

$token\_rotation\_timer_{NA}(trt_{NA})$

$token\_rotation\_timer_{TA}(trt_{TA})$

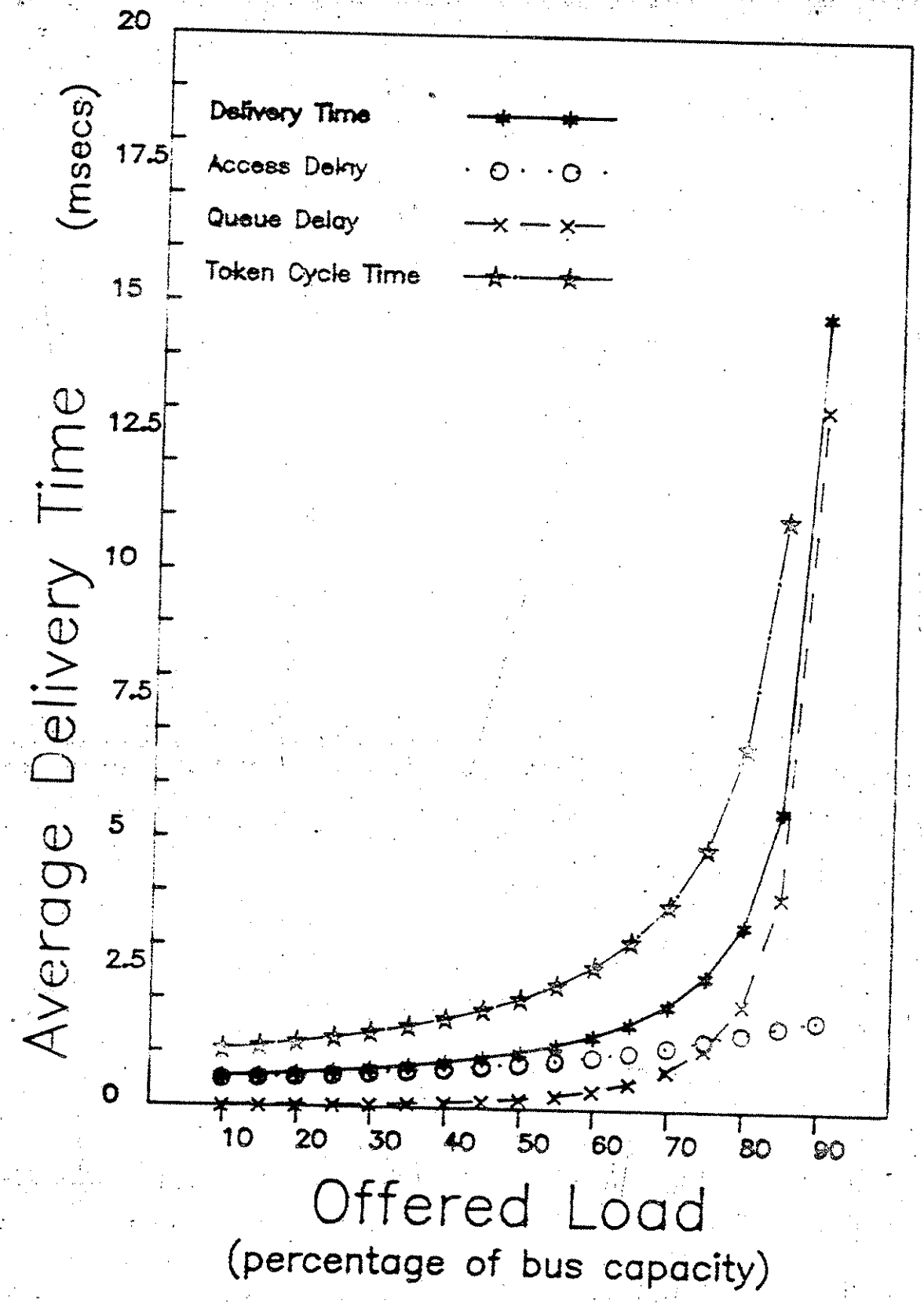
- used to monitor the token cycle time at *access_classes* UA, NA, and TA

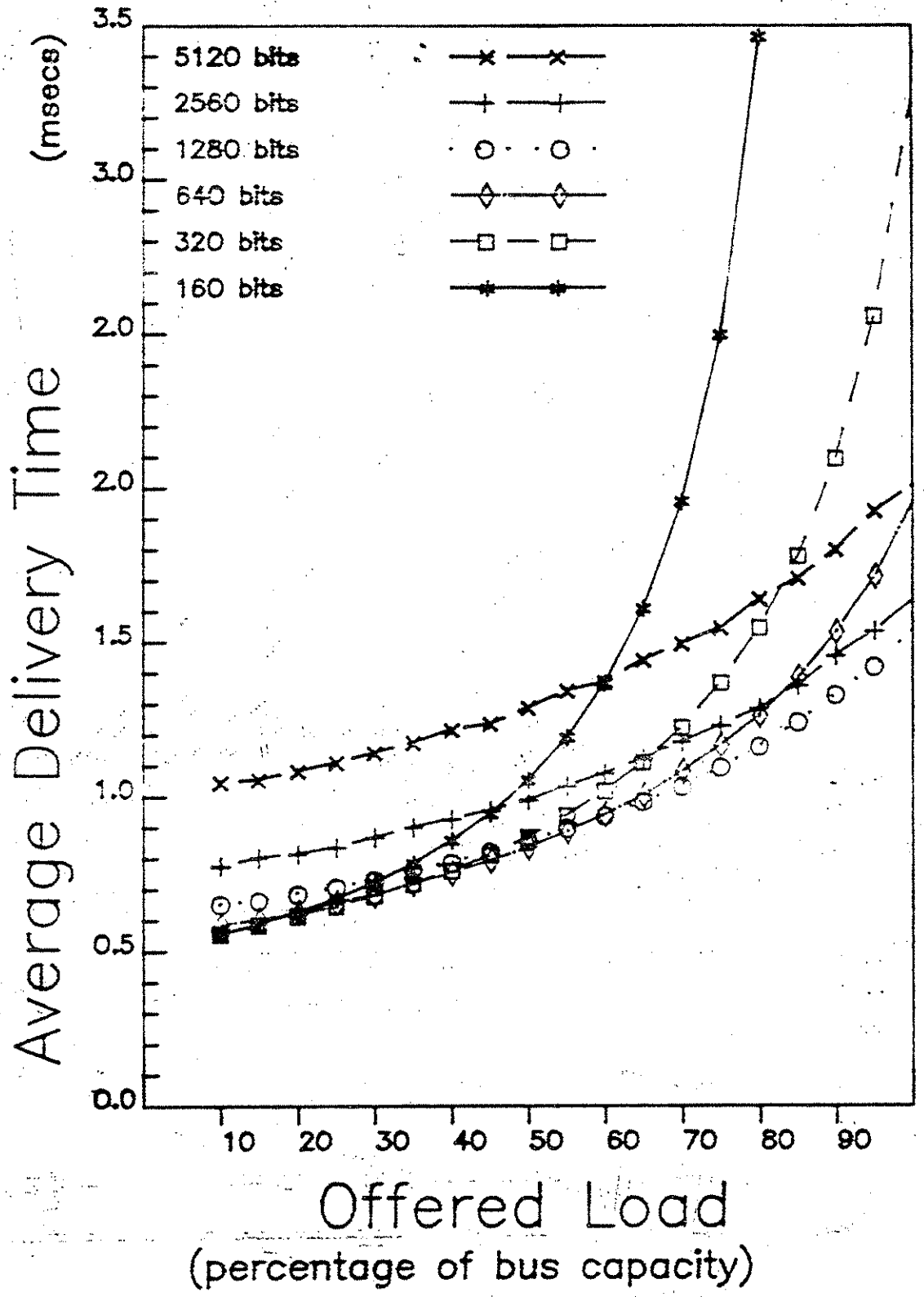
$token\_rotation\_timer_{RM}(trt_{RM})$

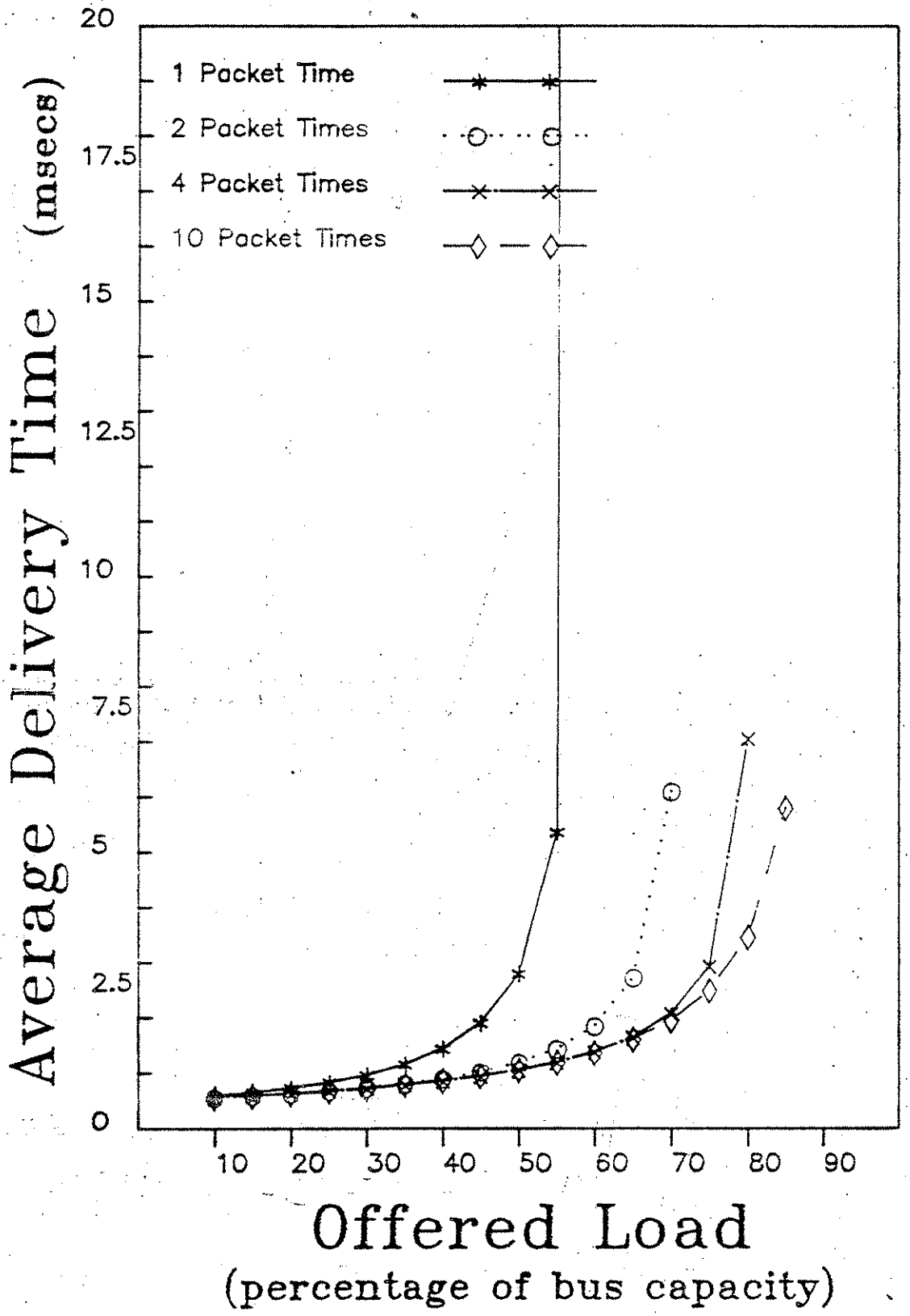
- ring maintenance timer

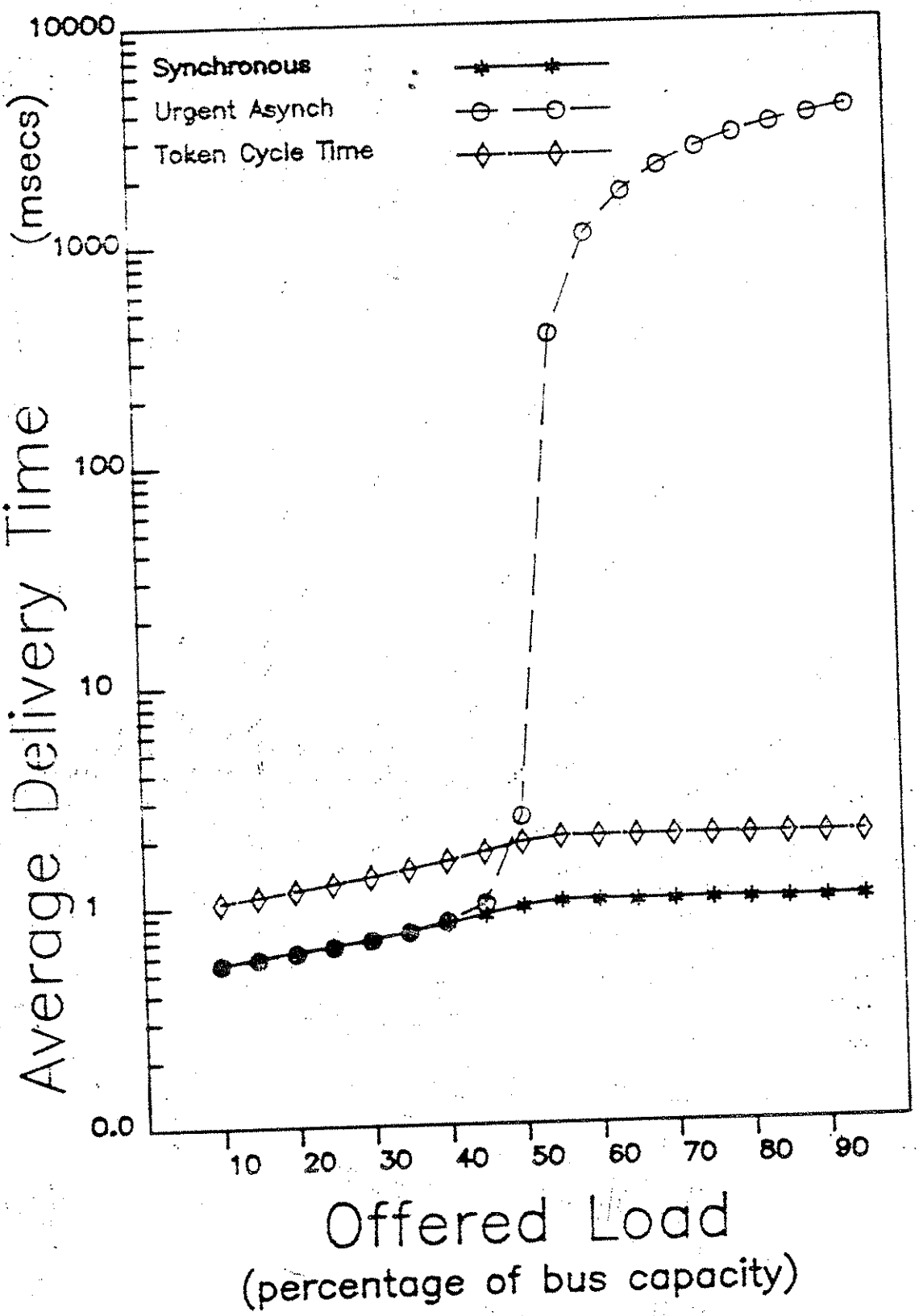
ESTABLISH BASE CONFIGURATION

- 64 stations, always members
- Synchronous traffic only
- Infinite *High_Priority_Token_Hold_Time*
- Constant length 160 bit messages (256 bit frames)
- Max_Inter_Solicit_Count = 255
- Bus capacity = 10,000,000 bits per second
- Error free
- 16 bit addresses
- 96 bit tokens

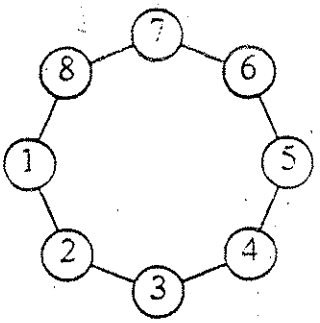

Delivery Time
Access Delay
Queue Delay
Token Cycle Time
Average Delivery Time (msecs)
20
17.5
15
12.5
10
7.5
5
2.5
0
10
20
30
40
50
60
70
80
90
Offered Load
(percentage of bus capacity)


5120 bits
2560 bits
1280 bits
640 bits
320 bits
160 bits
Average Delivery Time
(msecs)
3.5
3.0
2.0
2.0
1.5
1.0
0.5
0.0
10
20
30
40
50
60
70
80
90
Offered Load
(percentage of bus capacity)


1 Packet Time
2 Packet Times
4 Packet Times
10 Packet Times
Average Delivery Time (msecs)
20
17.5
15
12.5
10
7.5
5
2.5
0
10
20
30
40
50
60
70
80
90
Offered Load
(percentage of bus capacity)


10000
1000
100
10
1
0.0
(msecs)
Average Delivery Time
Synchronous
Urgent Asynch
Token Cycle Time
10
20
30
40
50
60
70
80
90
Offered Load
(percentage of bus capacity)

TOKEN RING



A node's physical neighbor is its logical neighbor

All data paths are point-to-point

Mixed media are allowed

Network access is controlled by a token

Message acknowledgements are automatic

Supports eight priorities

Efficiency is high

TOKEN RING

Topology is a closed physical loop

Bits are transmitted serially and unidirectionally

Frames circulate one full circumference before being purged by the transmitter

Receiving stations, if any, copy bits and then repeat them

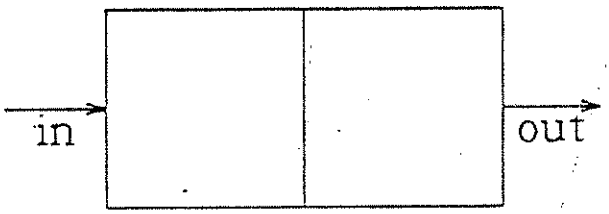
Stations may be in *active* or *bypass* mode

To start transmission, station modifies a *token* to become a Start_of_Frame sequence

LATENCY

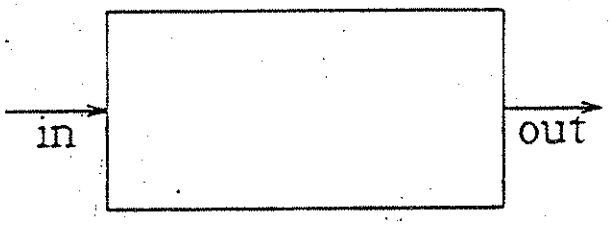
Station latency

- Every station has a "window" between its receiver and transmitter, typically 1-4 bits



Latency buffer in active monitor

- A buffer of 24-30 bits which ensures that the token (24 bits) completely fits on the ring

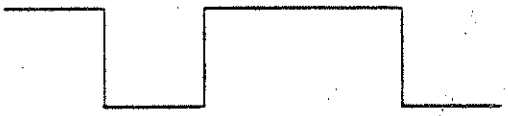


Propagation latency

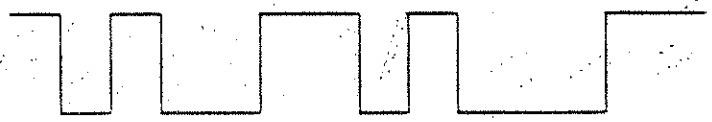
- Signal propagation time from station to station

SIGNAL ENCODING

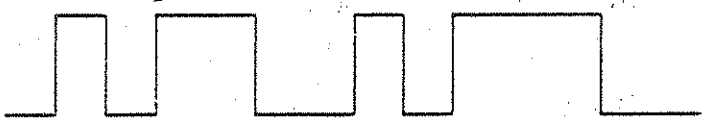
1 0 1 1 0 J K



NRZ Binary Code

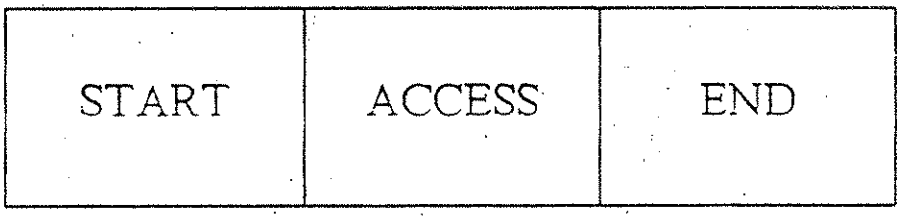


or



Differential Manchester Code

TOKEN



START and END are unique delimiters

START delimiter is "J K 0 J K 0 0 0"

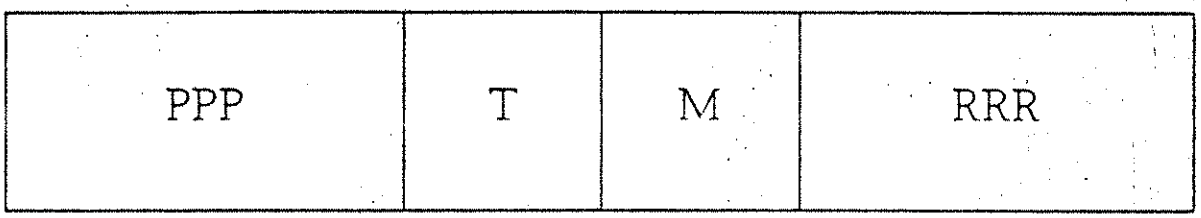
END delimiter is "J K 1 J K 1 I E"

"I" indicates *intermediate*, i.e., not last frame of a sequence

"E" indicates error; if station detects error it sets E=1, else repeats E as received

ACCESS is "P P P T M R R R"

ACCESS CONTROL



Priorities range from 0..7, with 7 highest

PPP -- priority of current frame (token or message)

T -- token bit

- T=0 if token is being repeated (i.e. free token)
- T=1 changes token into Start_of_Frame sequence

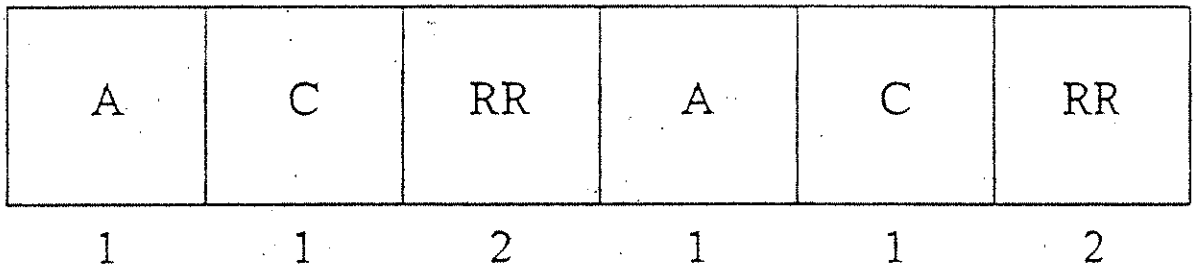
M -- monitor bit (for error detection)

RRR -- reservation priority

FRAME FORMAT

SFS		FCS					EFS	
SD	AC	FC	DA	SA	INFO	CRC	ED	FS

FRAME STATUS



"A" and "C" are transmitted as "0"

If address is recognized, "A" is set

If frame is copied, "C" is set

Frame returns to transmitter

Transmitter now recognizes:

"A"	"C"	
0	0	station non-existent or not active
1	0	station busy or frame in error
1	1	acknowledgement

TIMERS

THT -- *Token_Holding_Timer* (one per station)

- reset to an initial value when token arrives
- limits the amount of time a station may transmit after having accessed a token
- default = 10 ms

TRR -- *Return_to_Repeat_Timer* (one per station)

- limits the time a station may wait to receive the header of a message it transmitted
- if TRR expires, station return to repeat mode
- default = 2.5 ms

TVX -- *Valid_Transmission_Timer* (one in active monitor)

- verifies that station which has accessed the token uses it
- default = THT + TRR = 12.5 ms

TIMERS

TNT -- *No_Token_Timer* (one per station)

- if it expires, station generates a new token
- default = N * (THT + TRR)

TAM -- *Active_Monitor_Timer* (one in active monitor)

- when TAM expires, the active monitor sends an *active_monitor_present* frame
- default = 3 sec

TSM -- *Standby_Monitor_Timer* (one per station)

- if station fails to hear *active_monitor_present* frame or a token within the timeout period, it signals an error
- default = 7 sec

PRIORITY

The value of the priority bits (PPP) in the most recently received access control (AC) field is stored in priority register Pr.

The value of the reservation bits (RRR) in the most recently received access control field (AC) is stored in the reservation register Rr.

If a station raises the token priority, then it stacks the old token priority (old PPP) in stack Sr and stacks the new (higher) token priority (new PPP) in Sx.

PRIORITY

When a station receives a reservation field of higher priority than the token (Sr > Pr) it:

- stacks current token priority in Sr (Sr ← PPP)
- raises current token priority to reservation value (PPP ← Sr)
- stacks new token priority (Sx ← PPP)

PRIORITY

If token returns to a station which has raised its priority, and if token priority received equals token priority sent (PPP = highest Sx), and if reservation field is less than or equal to the priority field (RRR < PPP), then:

- token priority reduced to highest value on stack (PPP ← highest Sr)
- pops Sr and Sx stacks
- reservation bits (RRR) are unchanged

If reservation field exceeds token priority (RRR > PPP), then:

- replaces top Sx value with reservation (Sx ← RRR)
- raises token priority (PPP ← RRR)
- sets reservation field to zero (RRR ← 000)

ACTIVE MONITOR

Only one active monitor, although any station can assume this duty

Periodically send *active_monitor_present* frames

Can purge the ring of bad messages, or force a transmitting station to cease transmission

Can detect a token continuously circulating at high priority and repair it

STANDBY MONITOR

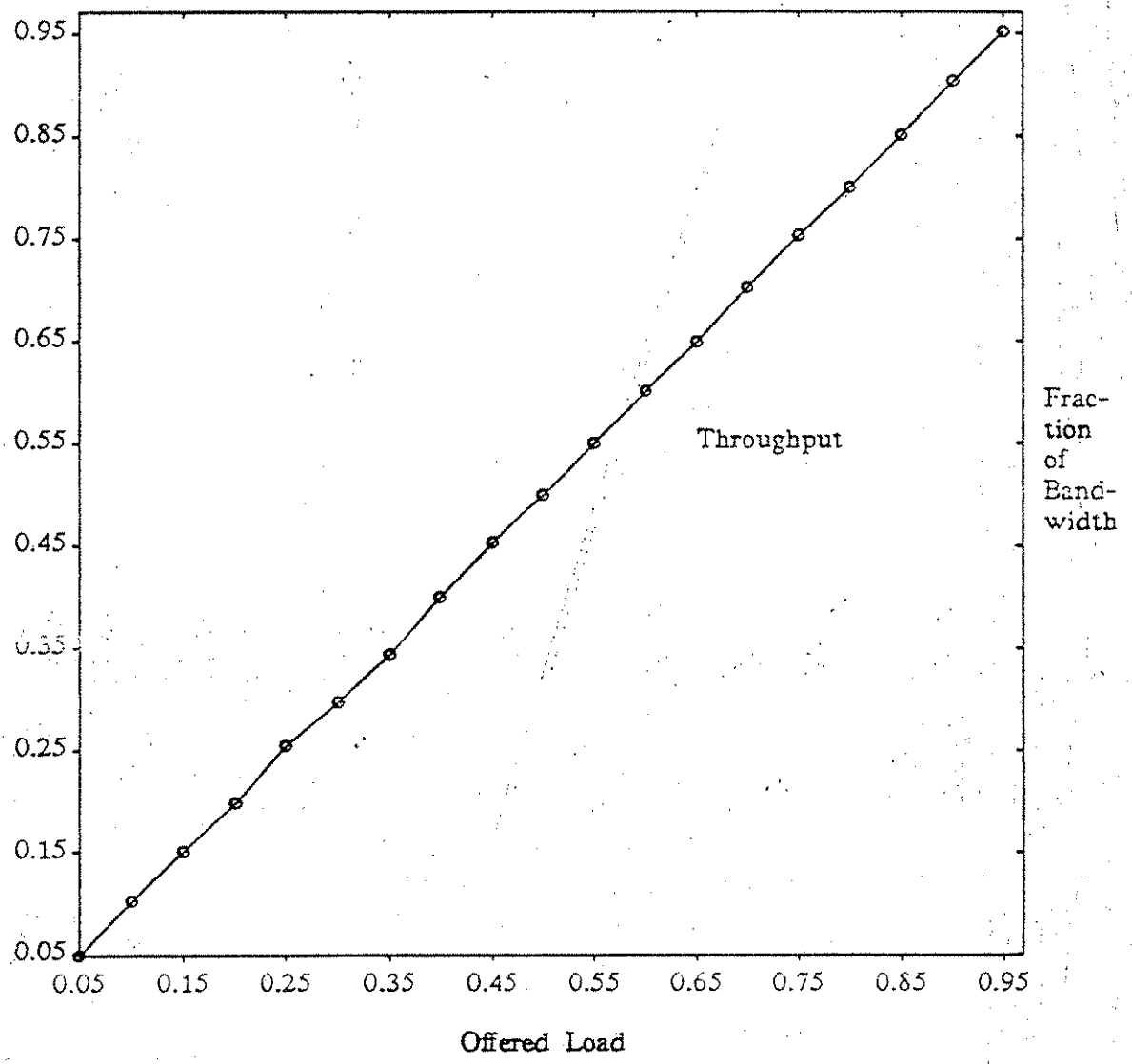
Can sense failure of active monitor (no *active_monitor_present* frames) and replace it

Another station must then become the standby monitor

Everything done with timers

Details contained in section 2.8 of attached paper

Token Ring Throughput

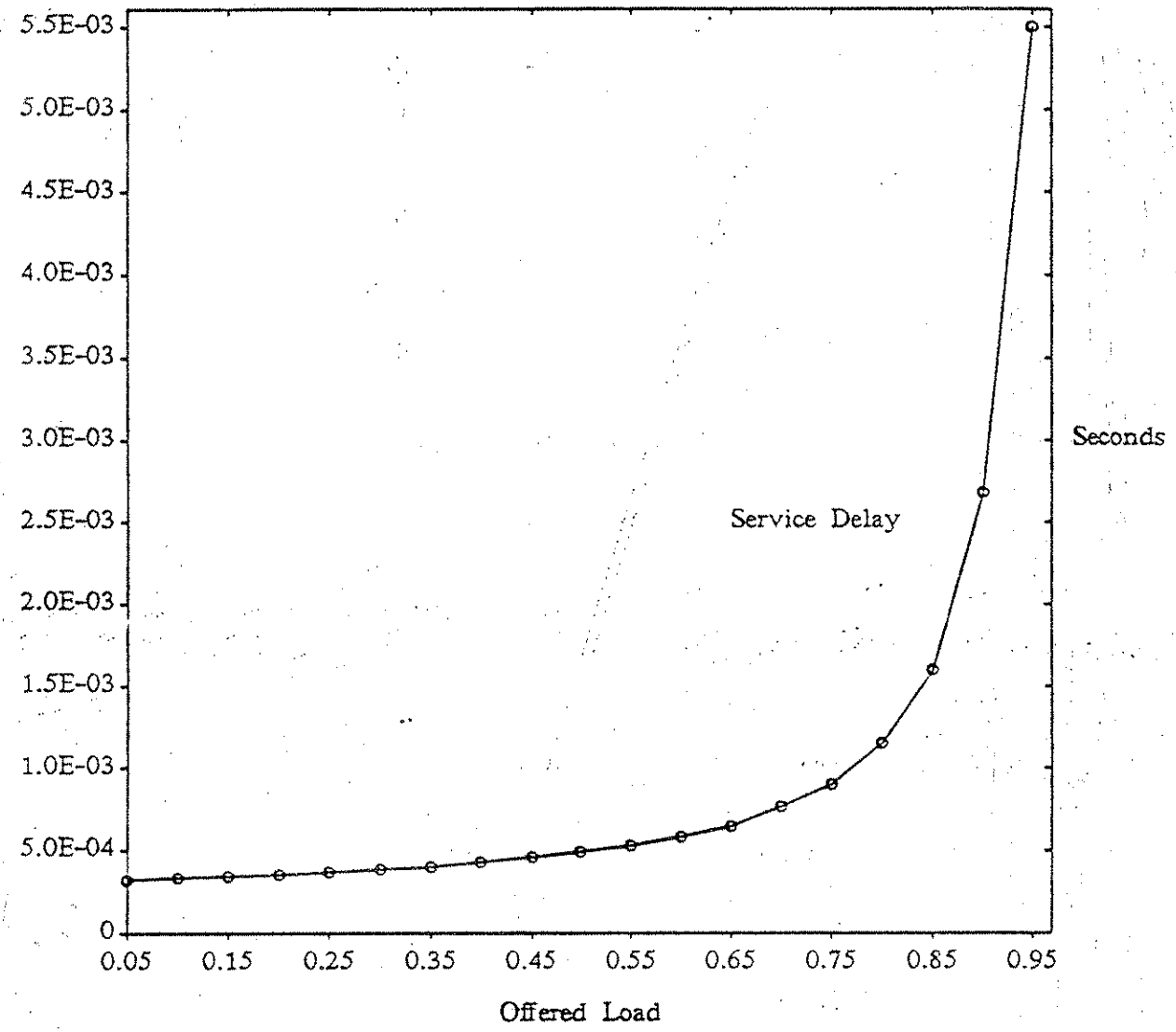


Configuration:

40 Stations
256 Bit Messages including framing
1 Bit Time Station Latency
Exponential Arrivals
TokenHoldTimer set to 10ms
1Mbps Medium

Figure IEEE1.6

Token Ring
Service Delay

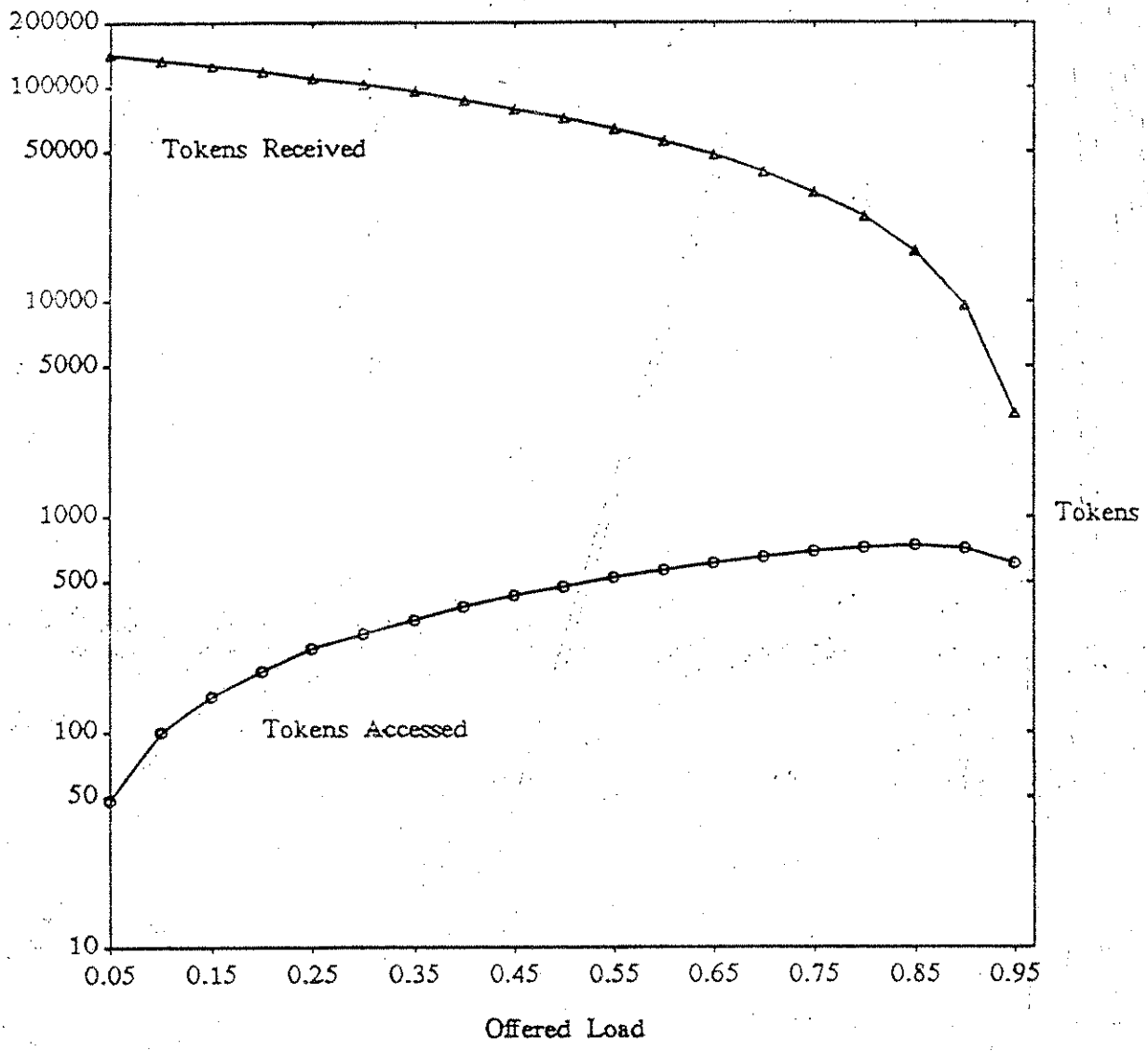


Configuration:

40 Stations
256 Bit Messages including framing
1 Bit Time Station Latency
Exponential Arrivals
TokenHoldTimer set to 10ms
1Mbps Medium

Figure IEEE1.3

Token Ring
Token Traffic



Configuration:

40 Stations
256 Bit Messages including framing
1 Bit Time Station Latency
Exponential Arrivals
TokenHoldTimer set to 10ms
1Mbps Medium

Figure IEEE1.5

GENERAL MOTORS MAP

Purpose:

- Define a MAP message standard which supports application-to-application communication
- Identify application functions to be supported by the message standard
- Recommend protocol(s) that meet GM's functional requirements

Goals:

- The driving force behind the MAP effort is the need for compatibility of communications to integrate the many factory floor devices
- It is the intention of MAP to promote a multi-vendor environment

MAP ARCHITECTURE

Layer 1 -- IEEE 802.4 Broadband

Layer 2 -- IEEE 802.2 Class 1

Layer 3 -- Null

Layer 4 -- ISO Transport Class 4

Layer 5 -- ISO Session Kernel

Layer 6 -- Null

Layer 7 -- ISO CASE Kernel

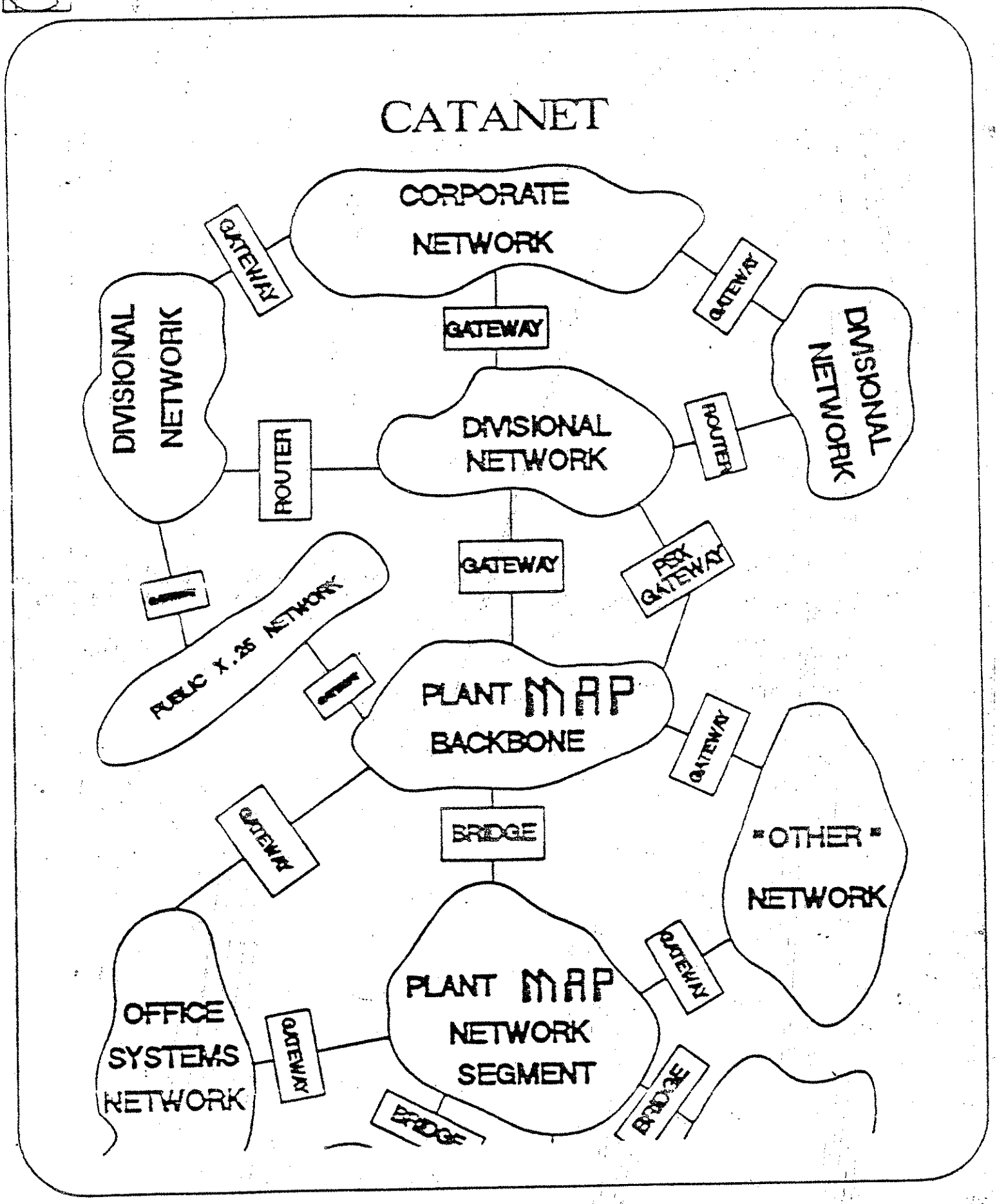
PROGRAMMABLE DEVICES

MAP must support programmable devices through all 7 layers

- programmable controllers
- robots
- CNC machines
- weld controllers

The minimum set of network functions include

- program upload and download
- storage and retrieval of data
- status reporting
- remote diagnostics



(slide catanet)