



REDES

VOLUMEN I

POR: CARLOS ARDILA ARENAS

FACULTAD DE INGENIERIA
DEPARTAMENTO DE INGENIERIA
DE SISTEMAS Y COMPUTACION

ENERO DE 1988

UNIVERSIDAD DE LOS ANDES

Cra. 1a. Este No. 18A-10 Tels. 2849911 - 2439518 A.A. 4976 Bogotá, Colombia

BIBLIOGRAFIA

1. TANENBAUM-Computer Networks (2ª ed.)
2. MARTIN-Telecommunications and the Computer (3ª ed.)
3. STALLINGS-Local Networks an Introduction
4. ARCHER- The Practical Guide to LANs
5. CHOU-Computer communications Vols. I y II

INTRODUCCION

QUE ES UNA RED DE COMP?

QUE ES UN SISTEMA DISTRIBUIDO?

VENTAJAS DE LAS REDES

APLICACIONES

CONSECUENCIAS

QUIEN ES QUIEN EN TC?

DIFERENCIA ENTRE VOZ Y DATOS

ELEMENTOS COMPONENTES DE UNA RED

TOPOLOGIAS

TIPOS DE ENLACE

ARQUITECTURA DE REDES

EJEMPLOS DE REDES

QUE ES UNA RED DE COMP?

Computadores autónomos interconectados

Pantallas remotas colgadas a CPU no son red

Similar a red telefónica

QUE ES UN SISTEMA DISTRIBUIDO?

Sist. cuyos elem. - se distribuyen - los tareas q' el sist realiza ; aplic. q' corre / una red.

Unico para toda la red

Funciones dispersas en varios computadores

VENTAJAS DE LAS REDES.

- Evita aislamiento de centros administrativos
- Distribuye el riesgo de las fallas
- Procesamiento al lado de la fuente
- Mejor relación beneficio/costo
- ^{Realiza el} Crecimiento modular

APLICACIONES

- Acceso a programas remotos
- Acceso a bases de datos remotas
- Correo Electrónico - Teleconferencia

Acceso a bases de datos remotas

Reservas de aviones, teatros, trenes
Transferencia Electrónica de Fondos
Periódico automático especializado
Información meteorológica

→ SITA

CONSECUENCIAS

Se reducirá la necesidad de transporte

Ahorro de energía

Trabajo en casa

Incremento de educación a distancia

Ventas por correo electrónico

Ciudades más dispersas

Revolución informática

**QUIEN ES.
QUIEN EN TC?**

En gral son
monopolio
del estado

Correo,
Telégrafo,
Teléfono,

Se conocen
como PTTs

Post,
Telegraph &
Telephone

Hay + de 1600

Llamadas
Common
Carriers

En EEUU son
privadas

AT&T es la +
grande

Maneja ^{el} 80% de
los teléfonos

Cubre ^{la} mitad
de EEUU

El resto la
otra mitad

op.

FCC

③

Naciones
unidas

Coordinan por
medio de ITU

International
Telecommu-
nication Union

Tiene tres
órganos

2 para radio y
uno para
telefonía y

El último es la
CCITT

③

25

Red de COMM de PAQUETES $\rightarrow$. redes de computadores

↳ pg. de inf. q' se recibe y analiza (x con las redes telefónicas (de com. de circ))

CONM DE PAQUETES: Canal dedicado toda la llamada
CONM DE CIRCUITOS: ^{asigna el canal} Sólo ~~se usa~~ ^a durante la trans-
ferencia real de información

Ejemplo: Reserva de avión

COMPUTADOR

2

△

Nombre Pasajero

54 caracteres

40 caracteres

94 caracteres

V de transmisión 120cps^{af}_{6s}^{eq}

$$t_{\text{efectivo uso canal}} (94/120) = 0.78 \text{ seg}$$

t total 90 seg

RELACION: 115

ELEMENTOS COMPONENTES DE UNA RED

COMPUTADORES DE USUARIO.....HOST's

procesos, aplicaciones

SUBRED.....SUBNET

Elementos de Conmutación.....IMP ^{IMP}

de paquetes

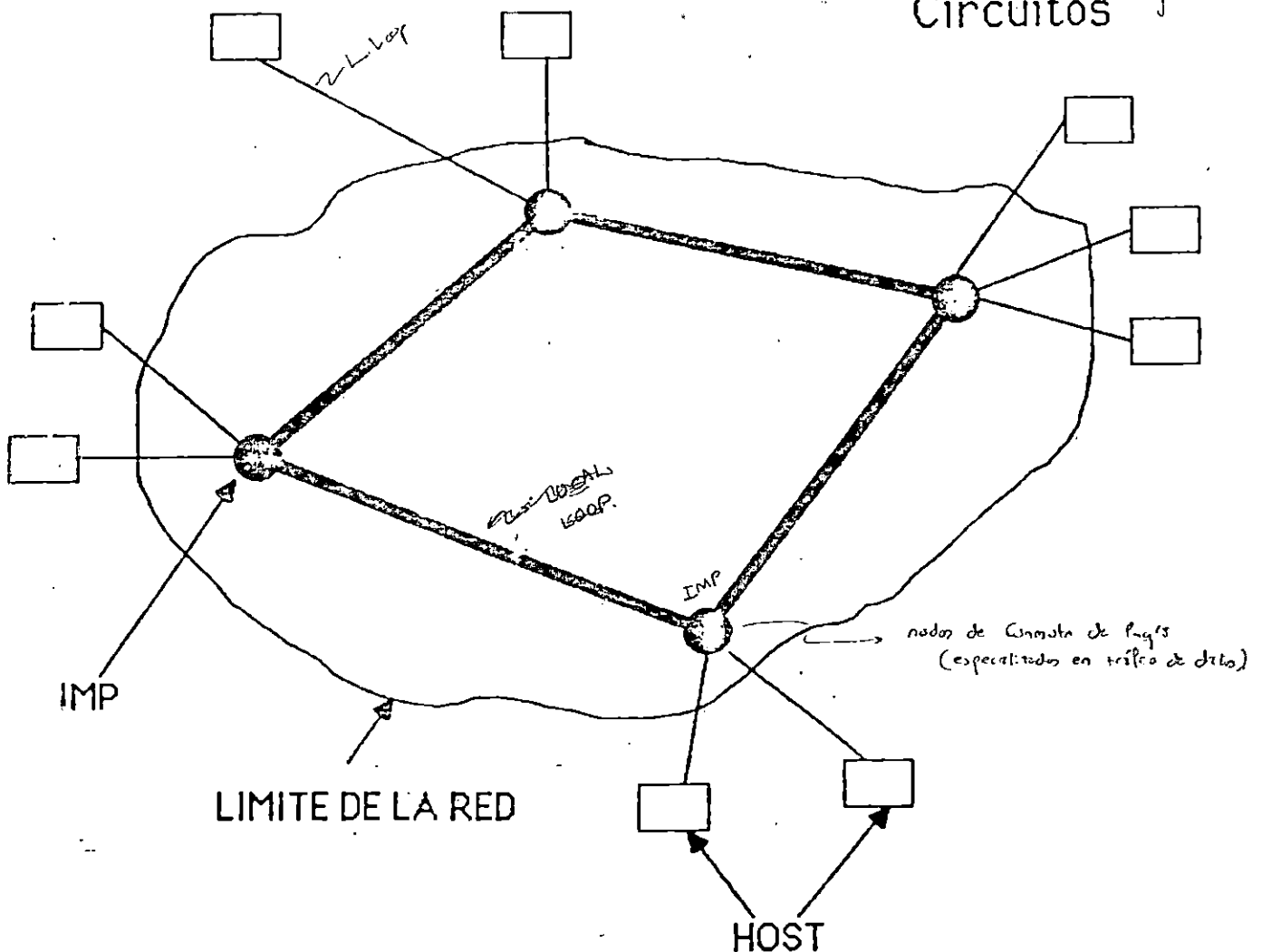
^{IMP}
_P

?

Elementos de transmisión.....Canales o
de inf.

Líneas o
Circuitos

Términos
"equivivalentes"



Distancia
entre
Procesadores

Procesadores
situados en
el mismo

0.1 m	Tarjeta	Maquinas Data Flow
1 m	Sistema	Multiprocesadores
10 m	Cuarto	Redes locales
100 m	Edificio	
1 km	Campus	
10 km	Ciudad	Redes geográfica- mente dispersas
100 km	Pais	
1000 km	Continente	Interconexión de Redes geográfica- mente dispersas
10000 km	Planeta	

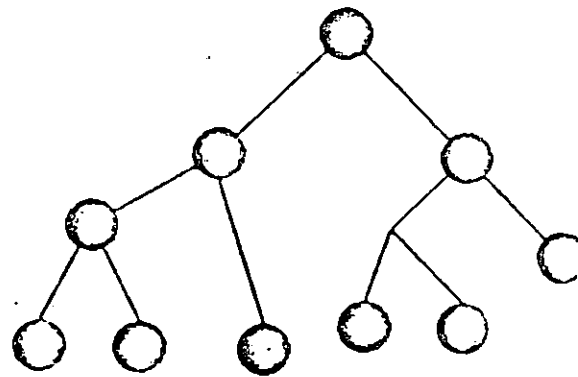
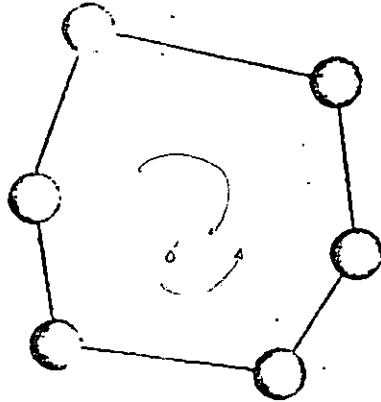
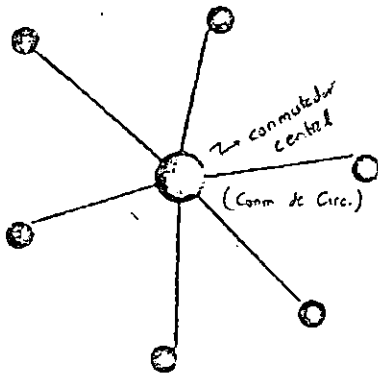
CLASIFICACION DE PROCESADORES INTERCONECTADOS POR TAMAÑO

DISTANCIA

Gate Wates → conectores de redes para hacerlas compatibles.

TOPOLOGIAS

- Digitales
- Telefonía + Datos } Voz
 } Datos.



1) ESTRELLA

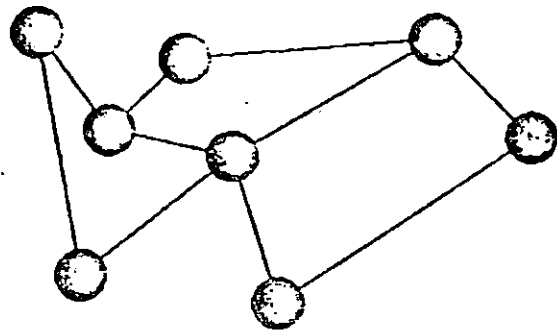
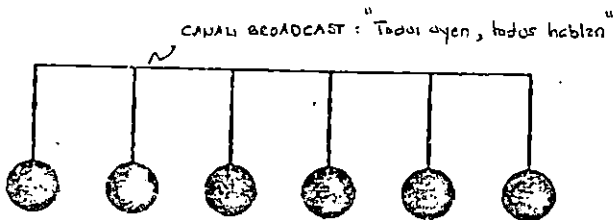
- redes LOCALES de CONMUT. de CIRCUITOS
- Ej: redes telefónicas (conmutadores)

2) ANILLO

- Flujo UNIDIR. de inf.
- Ej: TOKEN RING de IBM®

4) ARBOL

- Extensión de la red en BUS.



3) BUS

5) IRREGULAR

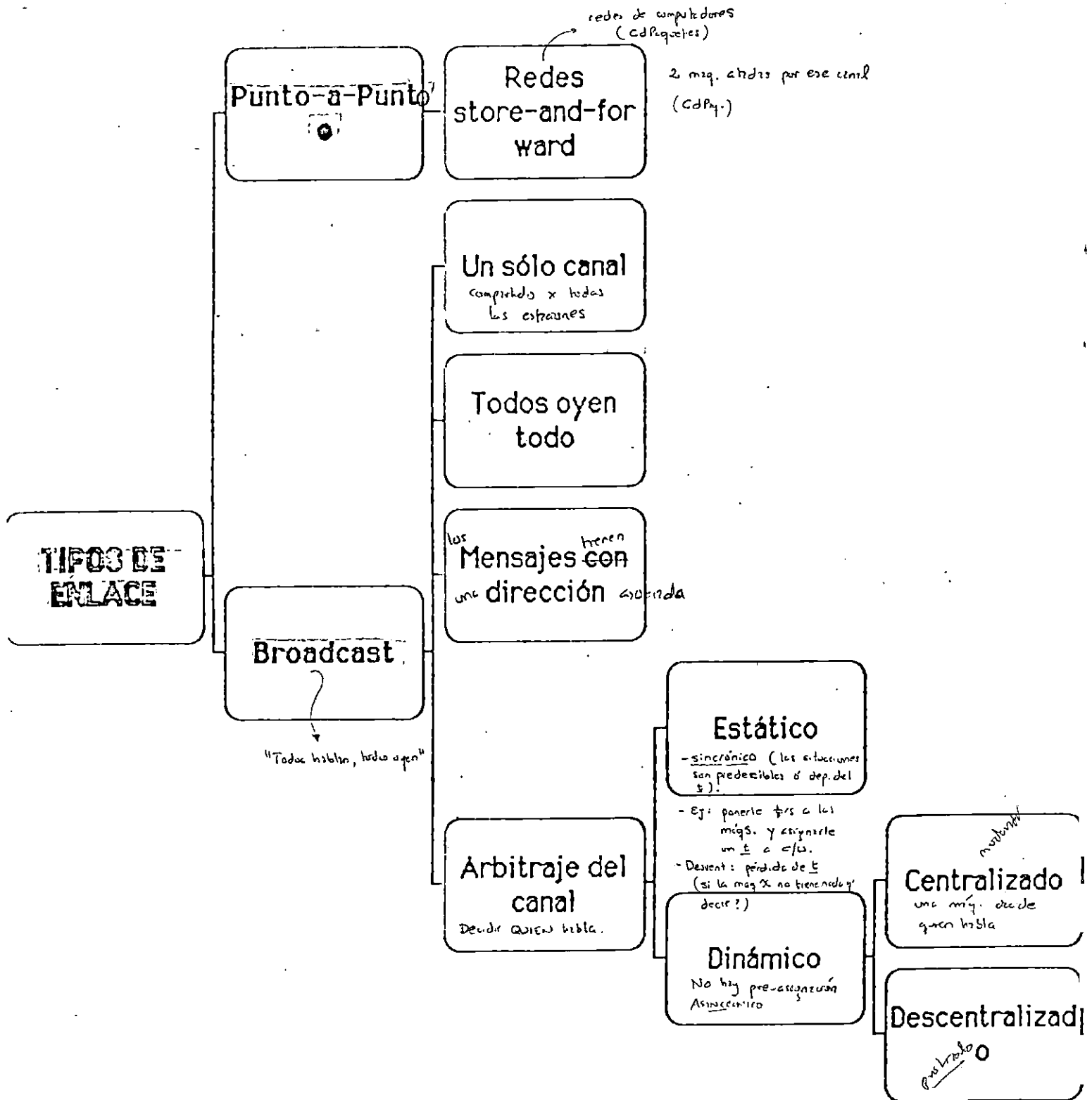
- Red geográfica y dispersa
- no tiene una forma en particular

Topologías Simétricas $\xrightarrow{1) - 4)}$ Redes Locales

Topologías Irregulares $\xrightarrow{5)}$ Redes Geográficamente dispersas

Red Telefónica → la inf no cambia ; es transparente

Paquetes → se recibe, guarda, retransmite y se transmite
se puede lavar la inf. (Criptografía)



ARQUITECTURA DE REDES

ANALOGIA

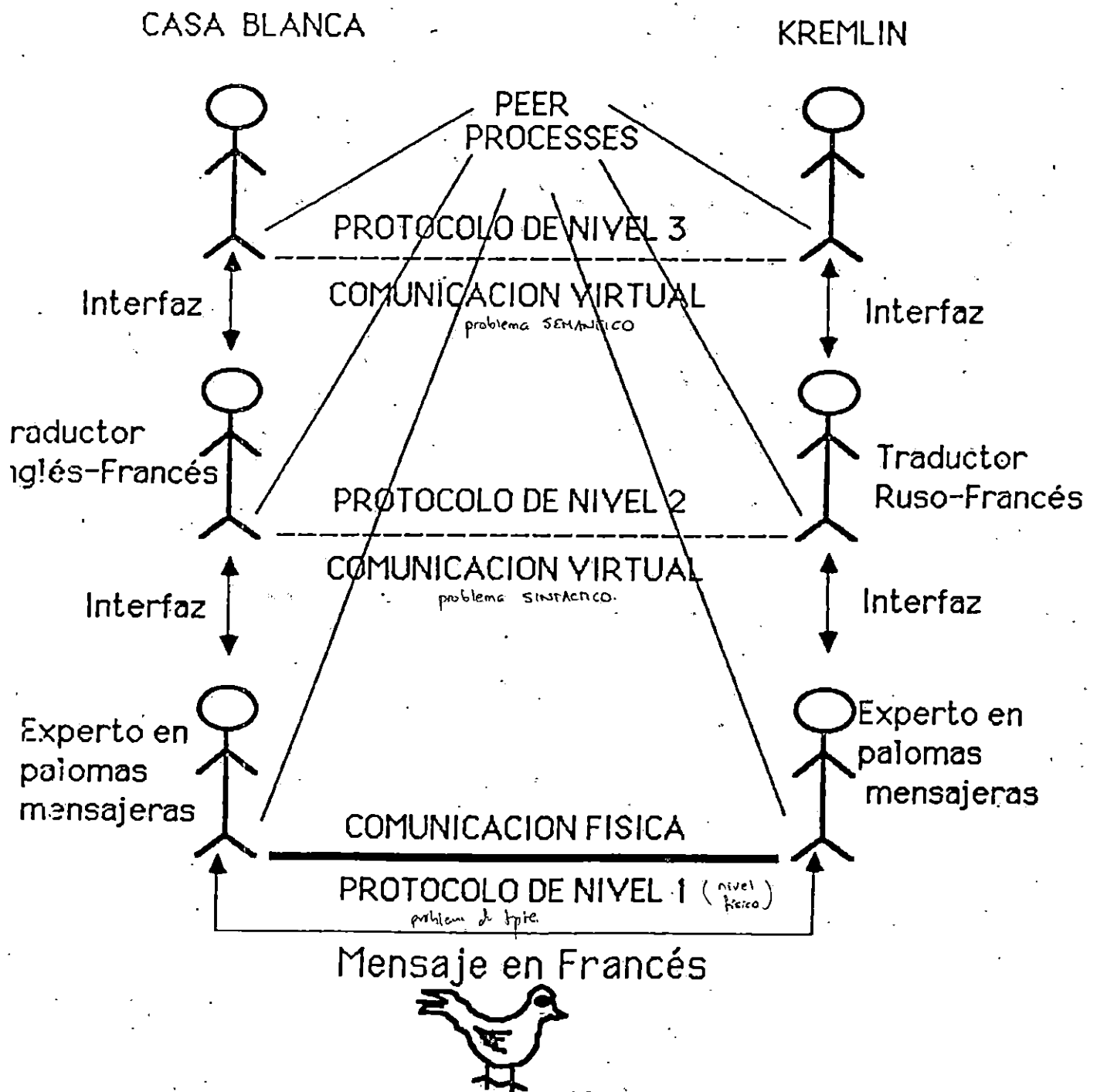
DEFINICIONES

PROBLEMAS DE DISEÑO DE REDES

MODELO OSI DE LA ISO

ANALOGIA

INTERCAMBIO DE MENSAJES ENTRE LOS PRESIDENTES DE EEUU Y URSS



DEFINICIONES

Nivel → estado sobre el cual 2 (máx) procesos
pueden comunicarse
alems. } están lógicamente de ser.

Procesos colegas → procesos
(Peer Processes) q' están
al = nivel lógico.

5^{er} nivel

Comunicación física

→ RS232
protocolo de nivel
físico

4^{to} nivel

Comunicación virtual

Sint
Sem.

Interfaz → serie de reglas p' comunicarse entre
niveles *'s pero en la misma máquina.
(1)

Protocolo → reglas q' rigen la com. entre
procesos colegas. (= nivel)
(-)

PROBLEMAS DE DISEÑO DE REDES

Cómo transmitir información por un cable?

Qué hacer con ruidos e interferencias? comparar errores en la com.

Cómo decir a qué computador va? @'s

Cómo decir a qué proceso?

• Por qué camino transportar el mensaje?

Si un computador es + rápido que el otro?

Cómo asegurar la privacidad de la información? criptografía

Fallas en ^{el} medio de transmisión de archivos? check-points

Qué pasa si ^{los} mensajes llegan en desorden?

Simplex? Half Dúplex? Full Dúplex? CS-TS?

↳ Com. Receptor - Transmisor ↳ Com. Bidir NO-Simult (Radio Tel.) ↳ Com. Bidir SIMULT (Tel.)

Qué hacer con mensajes muy largos?

Figur tam. del paquete *

Comuto. de Paquetes → se trabaja SIEMPRE en MEM, NUNCA en DISCO (por el t. de respta.)

* Paquete = bloque de bytes (128B, 8K, etc) + inf. de CTRL.

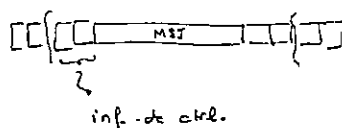


- # MODELO OSI DE LA ISO → 7 niveles

ISO = International Standards Organization

OSI = Open Systems Interconnection

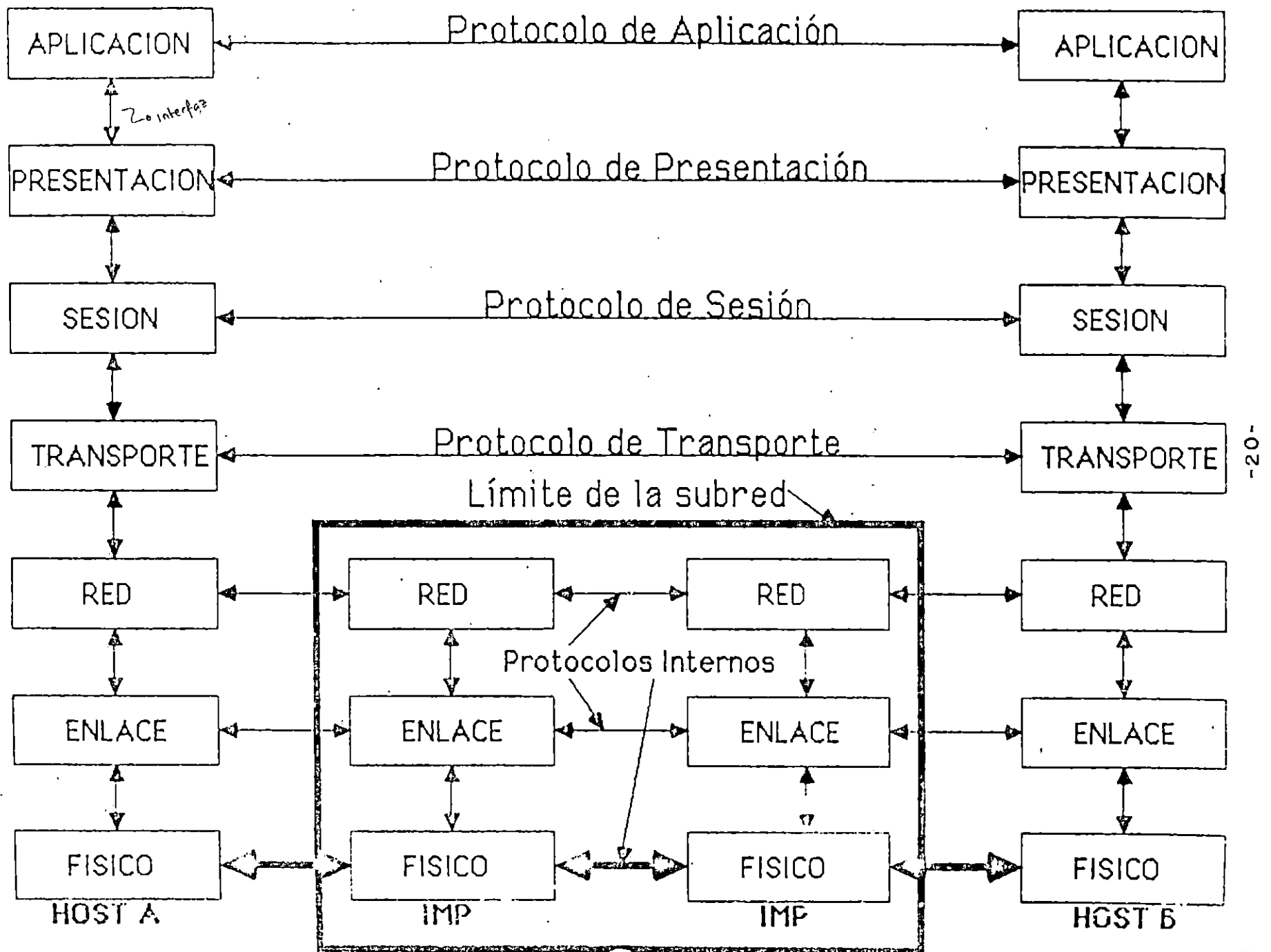
Funciones de los niveles: ^{siete (7)} q' lo componen:



OSI $\rightarrow$ ppius. q' aphró k OSI pora llegar a sus 7 niveles

- 1) c/nuel de la creacion por c/ $\neq$ nivel de abstr. nes.
 - 2) c/ nuel de la creacion un fn. def.
 - 3) la fn. de c/nuel de la creacion en ch. la def. de protocolos stricto
 - 4) los limites de la red deben escoparse y minimizar el flujo de info. a traves de las interfaces.
 - 5) $\neq$ niveles - la qda q' hayen por $\neq 5$.
- 19-
- por q' la creacion no requiere de ser aplicada

MODELO OSI DE LA ISO





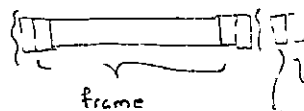
→ cómo se puede hablar,
sentido de la com.
VIAS,

→ características electromec.
de la comunic.

2) Nivel de Enlace.

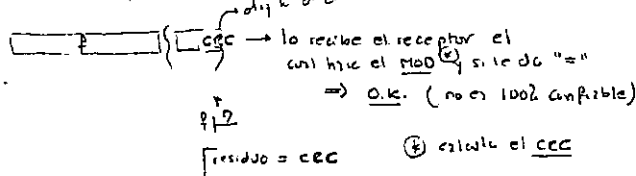
- Virtualiza línea → asegura q' la comunic. es CONFIABLE fna. entre pares de mós
- Arma marcos (frames) → confirma q' la inf. es correcta
- Acks y retransmisiones
- Headers, trailers, transparencia
- Duplicados por pérdida de acks
- Control de flujo → para saber si la inf. se está recibiendo BIEN.
- evitar q' una mg. 'chague' a otra.

ojo!
No de exiemo-
a-exiemo



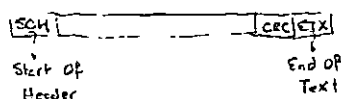
inf. de enlace
- red

transmisor

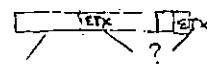


• ACK = Asci (44) → la inf. llegó BIEN!

• Headers & Trailers :
→ Ej CRC del N de ENLACE

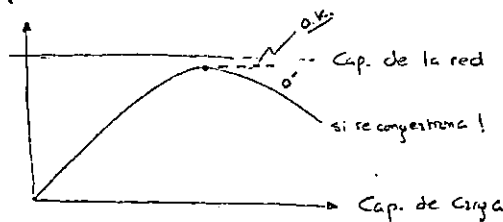


• Problema de TRANSPARENCIA :



• Periodo de TIME-OUT → ¿entre EDG y lo espera
de un ACK por pte. del Tr.
Si no se recibe el ACK ? ⇒ #r los frames.

Traffic (bytes) & funciones encordadas



3) Nivel de Red

último nivel q' corre en la red. (súber) mejs. q' en el resto de la

Enruta

- por qué camino llevar la inf.
- no le interesa qué inf. es, lo q' transmite.
- se encarga de la construcción de pg. (*)

↳ las máqs. dentro de la red deben saber decidir por q' camino llevar la inf. en el

Interfaz con usuario (forma del frame, etc.)

↳ cómo entregar la inf. al cliente.

Orden de paquetes

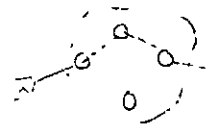
Divide mensajes en paquetes

Controla congestión → demoras pqr. en la red

Cobra

• def. la

Decide si/no



2 extre. -

Reams. de telecom.

- topología → dist
- congestión

→ para poder manejarlos en RAM

- (*) 2 formas de conmutar paquetes →
- Redes de Circ. Virtuales → con. establecida lógica (o física) → or. a conexión
 - Redes de Datagramas → NO ordenada a conexión
- NO puede ser desordenada de pqr.

4) Nivel de Transporte

- protocolo q' regula la inf. q' hay entre hosts.
com.

q'! /
{ d' q' solo como
en los hosts.

ignora el hecho de q' haya un red entre máqs.
Comunicación entre hosts
Interfaz con red

3 Control de flujo host-host

Ej → minframe → puede estar hablando con varios usuarios a la vez (ajós)
Acc's → ples. del cdFlujo

Maneja conexiones entre
puertos

Multiplexa → una misma línea está sendo usada
por varias comunicaciones.

→ conexiones de tpte en circ. lógras.

Acc's { a nivel de entre
" tpte.

5) Nivel de SESIÓN → (Pel. Clive)

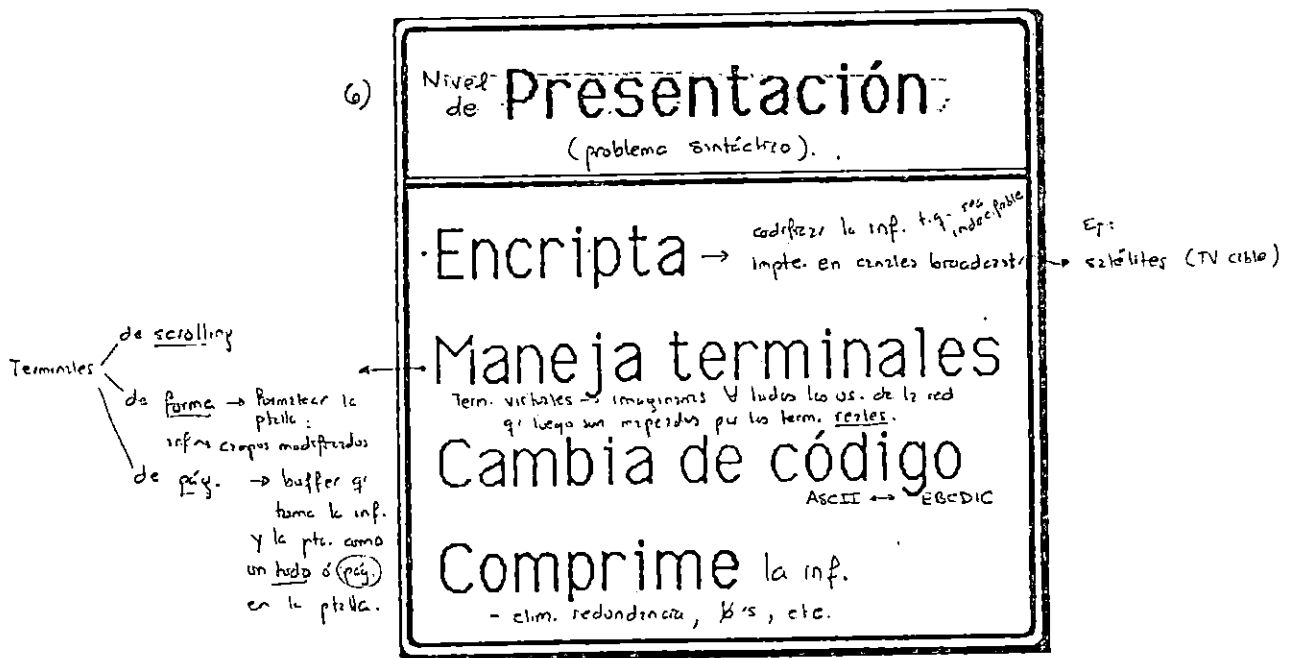
- Ctrla la interacción red-usario
Flujo ordenado

- Derechas de acceso

- Checkpoints

- Atomicidad de transacciones en BDD

↳ giras bancos entre ciudades → Débito al padre
↳ Crédito al hijo ó ninguno de los dos.



4-6-6-5-4-4-4

7) ~~Nivel de~~ Aplicación

- Bases de datos distribuidas
- Sistemas operativos distribuidos

· Correo Electrónico , · Teleconferencia

CORRESPONDENCIA OSI Y REDES EJEMPLO

penetration con todo tipo de mg.
open systems interconnection
OSI (7)

advanced research project agency
ARPANET (?)

physical networks
SNA (7)

digital equipment operation
DECNET (5)

7	Aplicación	Usuario	End User	Aplicación
6	Presentación	Telnet, FTP	Servic. NAU	
5	Sesión	No existe	Data Flow Control	Sesión
			Control de Transmisión	
4	Transporte	Host-Host	Path Control	Servicios de Red
		Source-Destination IMP		
3	Red			Transporte
2	Enlace	IMP-IMP	Data Link Control	Data Link Control
1	Físico	Físico	Físico	Físico

•/. (Parcial # 1)

EJEMPLOS DE REDES

ARPANET : Departamento de defensa EEUU

SNA: System Networks Architecture (IBM)

DECNET: (Digital Equipment Corporation)

X.25: Redes públicas (CCITT) → regulación pública
trabajo de
dicho pública

SITA : Red. de reservas en las aerolíneas
cupos

ITU { CCIR
CCITT

N.º ondas

ARPANET

Advanced Research Project Agency

En 1960 se concibió la idea

En 1969, 4 nodos experimentales

Hoy tiene ^{cientos} 4 de 100 IMPs en 25 ciudades

Inicialmente usada para mostrar conmut. de paq.

En 1975 dejó de ser experimental

Primera red importante q' existía

Usada expresamente para asuntos de estado

Atraviesa EEUU, Noruega, Hawaii e Inglaterra

Usa ^{procesadores} Honeywell 516 y 316, y Pluribus ^{multiprocesador}

TIP: Terminal Interface Message Processor

Honeywell-316 reciben hasta 63 terminales (TIPs)

IMPs conectados por líneas de 50Kbps

Satélite de 50 y 9.6 Kbps para nodos interoceánicos

Pluribus maneja 10s de hosts y 100s de terminales

Longitud máxima del mensaje = 8095 bits → 8 paquetes

Hosts conectados: Serial, async, 100Kbps

TIPs reciben terminales asincrónicas

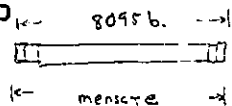
Investigación actual en ^{protocolos q' integran} ^{interconexión con otras redes} ^{protocolo TCP/IP} ^{tr. de voz paquete a paquete}

NCC (Network Control Center) → Modo de gestión de esta red.

Demora promedio medida: 0.5 seg → ^{dura el viaje a través de la red.} ^(t. de resp. de la transacción)

Troughput en transm de archivos: 8-20 Kbps

1 pag = 1k



- C/ host tiene su especificidad
- GTE telerat salió de ARPA
- DATAPAC (Cinedó)
- TRANSPAC (Foca.)

Investigación actual en

Protocolos gráficos

Interconexión con otras redes

Transmisión de voz paquetizada

TCP/IP
protocolo p' comunic
con otras redes

NCC (Network Control Center)

"Nodo de Gestión" de ARPANET

Distribuye software

Diagnóstico remoto de software

Diagnóstico remoto de hardware

Recolección de estadísticas de tráfico

SA: System Networks Architecture
(IBM)

Redes privadas para usuarios IBM

Complicada pero completa

Versiones

SNA llama "nodos" a todas las máquinas

NAU: Network Addressable Unit

1974: Redes centralizadas

1976, 2o release

LU: Logical Unit (@ proceso)
dirig.

PU: Physical Unit (@ máquina)
nodo

SSCP: System services control point

Host -
Terminales conectados

Múltiples hosts c/u con su árbol

Comunicación entre hosts.

Host - Front-end-processor (p. dedicado p' comunic.) -> cliente de term.

SSCP -> su q' maneja & direcciona los elem. de/red de hosts.
A cada se dir. el "dominio"
Ej: para enrutamiento

peer to peer

mantenimiento

Todas las máquinas son Nodos

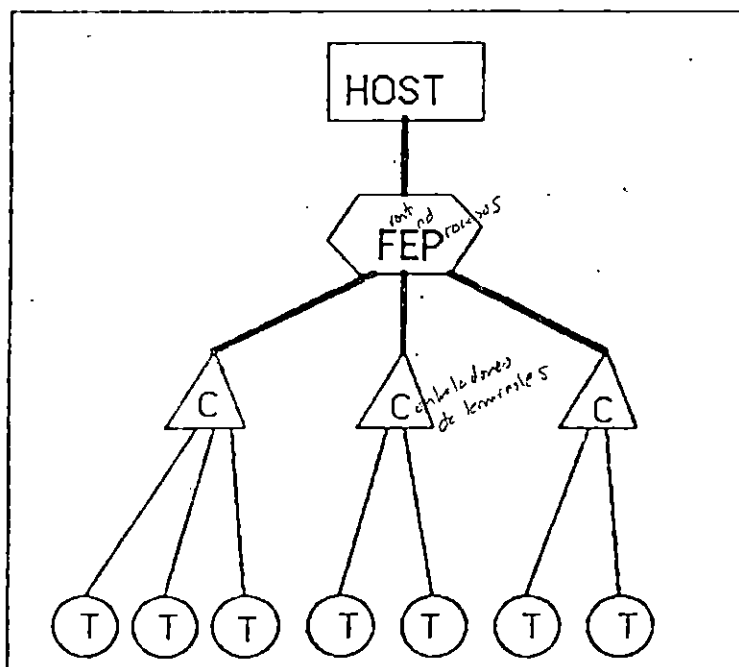
Tipo 1: Terminales

Tipo 2: Controladores

Tipo 3: No hay

Tipo 4: Front End Processors

Tipo 5: Hosts



SSCP
DOMINIO

DECNET

Pdte. de la DIGITAL Enc. con
req. propia (DNA)

Digital Equipment Corporation

DNA (Digital Network Architecture)

modelo
OSI

Para redes privadas de usuarios DEC

Existen muchas DECNETs aisladas

No distingue entre Host e IMP.

Varias versiones

Más simple que SNA^{la system networks de IBM} ~ redes / VAX's.

En Host ~ usuario de pr. de aplic.

En IMP's ~ Conn. de Paquetes

No distingue entre Host e IMP

Función de un nodo puede cambiar dinámicamente

Correr programas de usuario

Conmutar paquetes

Hacer las dos cosas

Varias versiones

Inicial: Comunicación entre máquinas adyacentes

Nueva: Small nodes (versión inicial) y full nodes

Tr { Sinc → + eficiente → nec. circuitos de mayor calidad (\$) { - como frames y los tr. como una unidad
Asinc → { - maneja la tr por car. indep. ⇒ se incrementan los PAD's
 como antes req no tenian req, al tipo ppto. de X25 y q' si son de memoria de trabajo/.

Similar a las telefónicas

Abarca los primeros 3 niveles del modelo OSI

Físico: X.21 y X.21bis
 protocolo digital
 RS-232C → perm. com. serial y Pull-duplex
 maneja líneas telefónicas

Enlace: LAPB
 protocolo de enlace
 SDLC (protocolo de la COM)
 ISO: HDLC
 CCITT: LAPB

Red: X.25
 interfaz
 Nivel de comunicación

Packet Assembler/Disassembler

3 protocolos para el manejo del PAD:
 X3: Funcionamiento Interno

X.28: Terminal-PAD
 reglas de comunicac.

X.29: Pad-Host remoto

X.25: Redes públicas (CCITT)

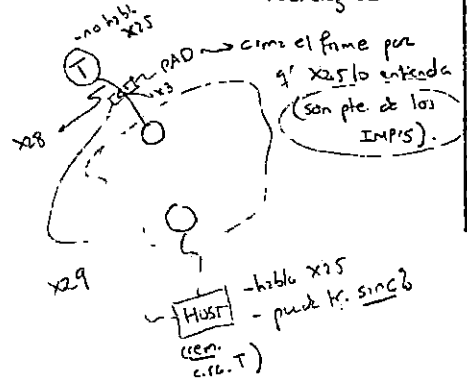
Es una INTERFAZ porq' no define cómo es por dentro del. como entregarle los datos a la red (col.)

orig. de los N.U. para acceso de datos y telefonía

Manejo de terminales PAD

padro. ESTANDAR

La problema: su lentitud → freno al desarrollo tecnológico.



SITA → Soc. Intern. de Com. Acústicas.

- Aprox 250 acústicos

- 160 países

- SMP's conectados a 9600 y 14400 bps

- Tráfico:

orig. prior. A → de reser. (80 circ.)

E de resp = 3 seg.

B → de msjes (250-400 circ.)

- 200 nodos de connc.

11K term. de reser.

12K impresora.

- 35 -

NIVEL FISICO

- Conceptos de AB y PB
- Sistema telefónico
- Conmutación de circuitos
- Comparación voz vs datos
- Señalización analógica
- Señalización digital
- Comparación de los 2 tipos de señaliza.
- Módems y modulación
- Configuración de canales
- Direccionalidad de canales
- Capacidad Máxima de un canal
- Imperfecciones
- Multiplexación
- Medios de transmisión (101)
- PCM ^{ultra} _{modulación} → telefonia digital.
- Representación de Inf. Digital
- Transmisión asincrónica y sincrónica
- Muestreo en transmisión serial
- Dispositivo de transmisión serial
- RS232-C
- Manejo de terminales
- Conmut de circ, mensajes y paquetes

+ FRECUENCIA →

- Hertz: Medida de frecuencia = 1 ciclo/seg
- Rango del sonido audible: 30-20.000 Hz

+ Ancho de Banda (AB) → *medida relativa*

- Ancho del rango de ^{10⁵} frecuencias que pasan por un canal
- Canal Telef: 3.100 Hz (300-3400Hz)
 $AB = (3400 - 300) \text{ Hz}$

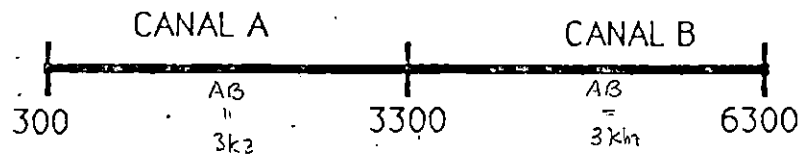
+ Paso de banda

+ Rango ^{de Proc.} situado en espectro electro magnético

+ Canales A y B

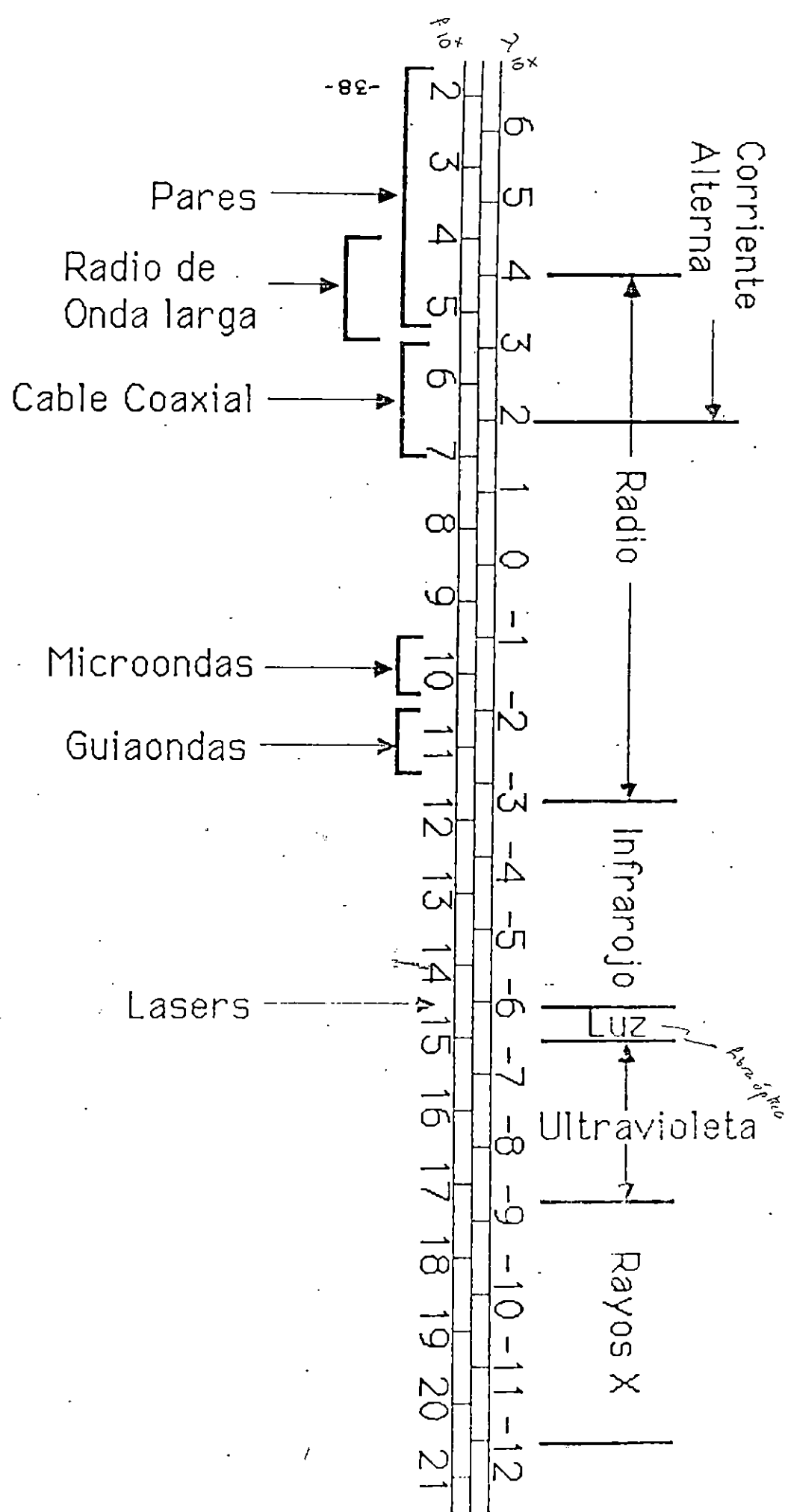
Tienen el mismo AB (3KHz)

Tienen distinto paso de banda

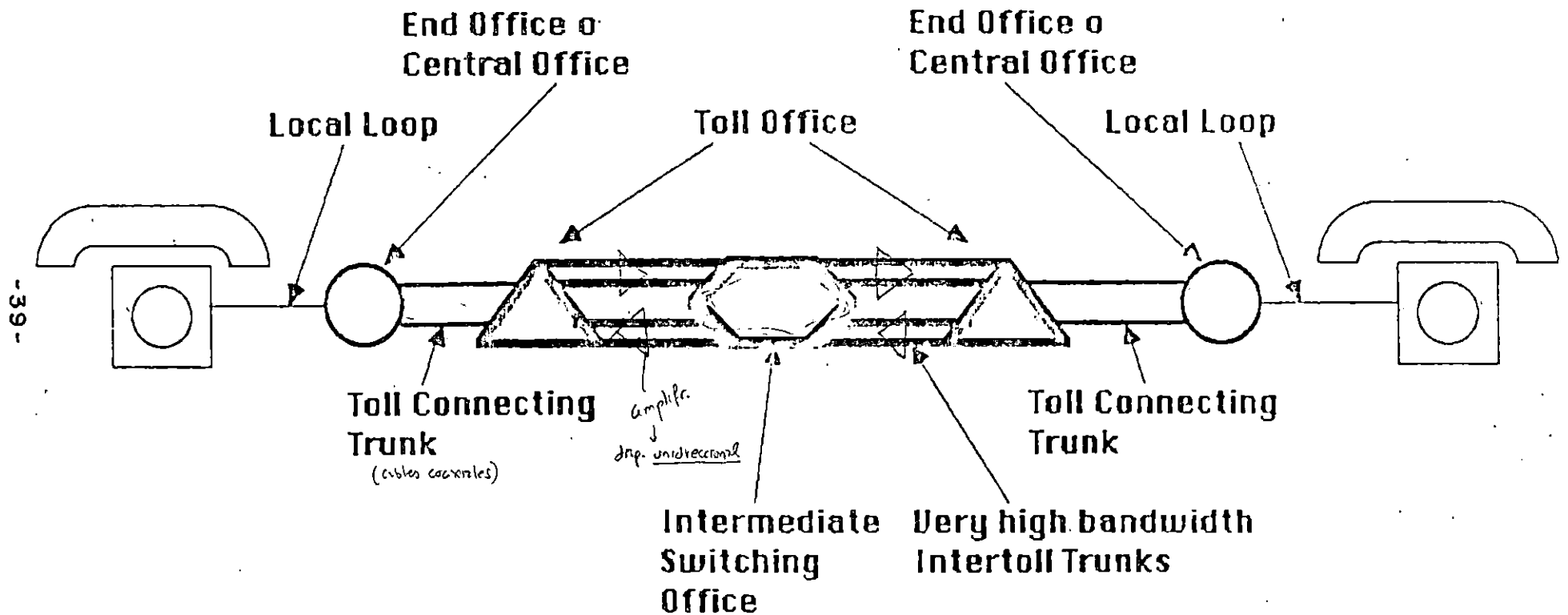


- Espectro Electromagnético

ESPECTRO ELECTROMAGNETICO

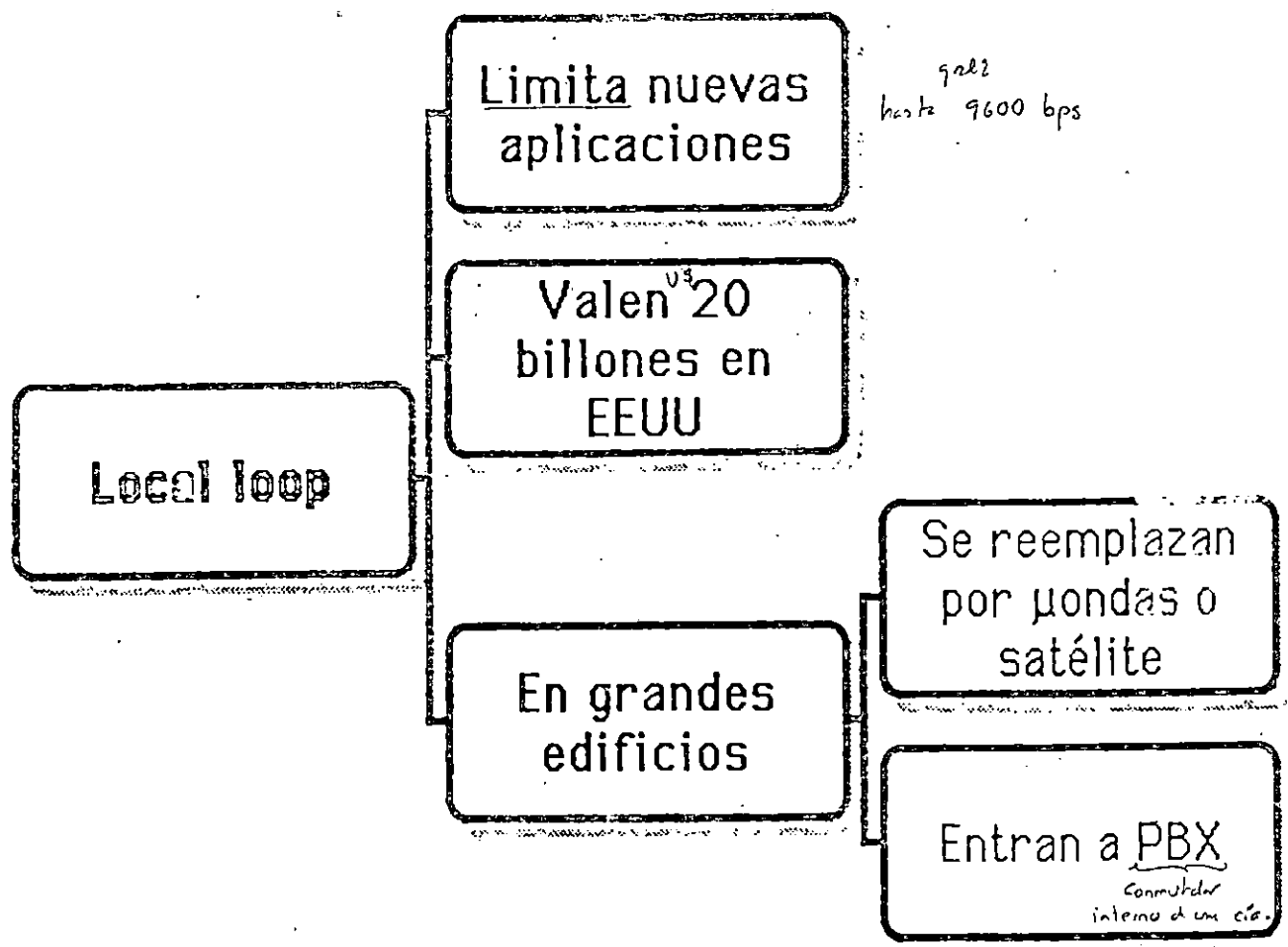


SISTEMA TELEFONICO

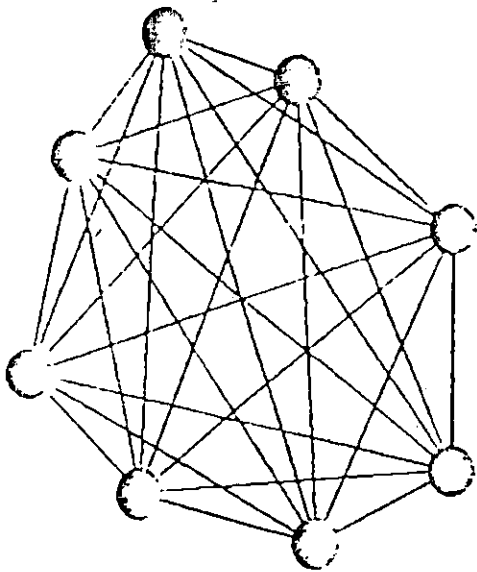


SISTEMA TELEFONICO

- Usado para datos por ubicuidad
q' est' en todos ples
- + Local loop
- Conmutación
- + Inversión inmensa
- + Voz .vs. Datos

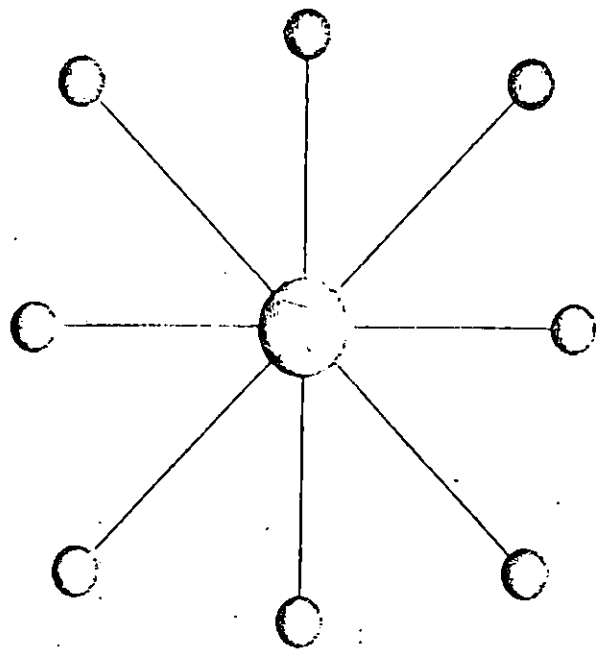


Conmutación *de circ.*

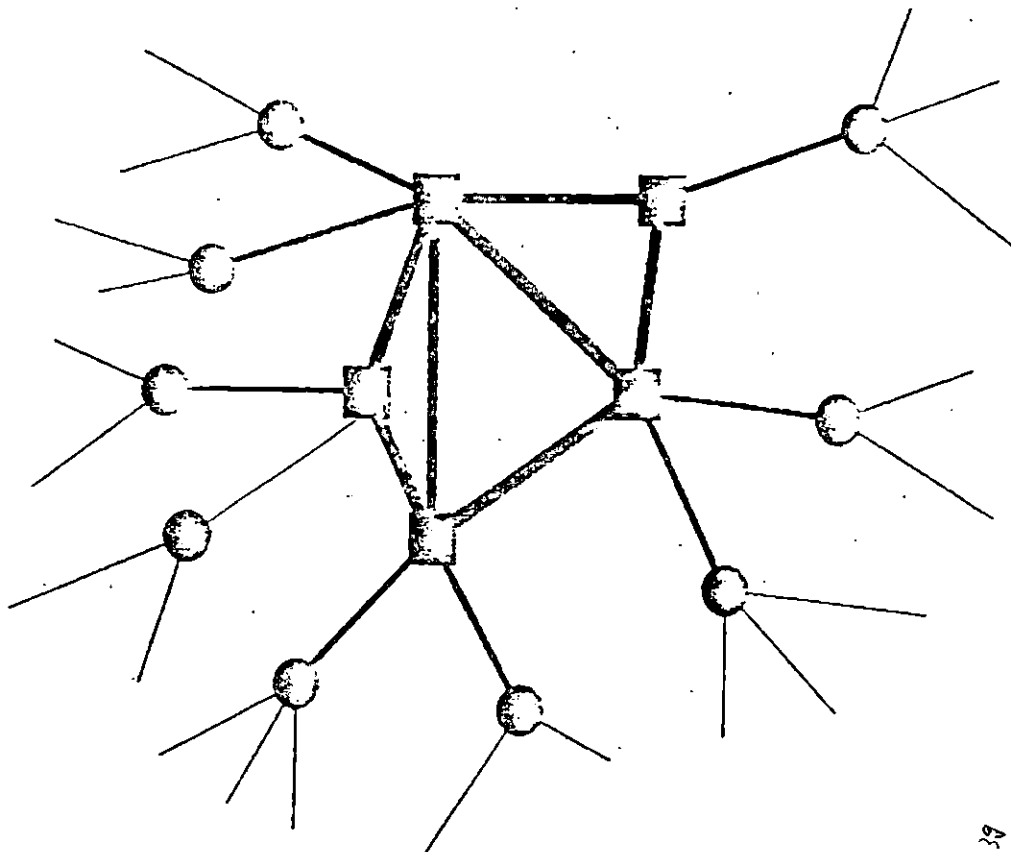


SIN CONMUTACION

todos conectados pto-a-ptu.



CONMUTACION CENTRALIZADA



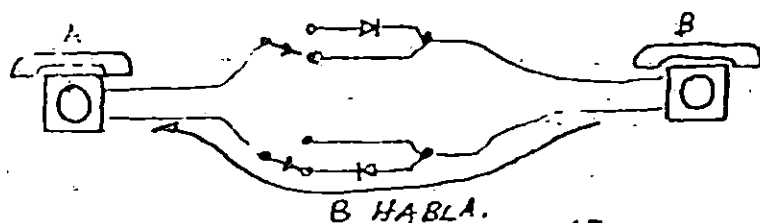
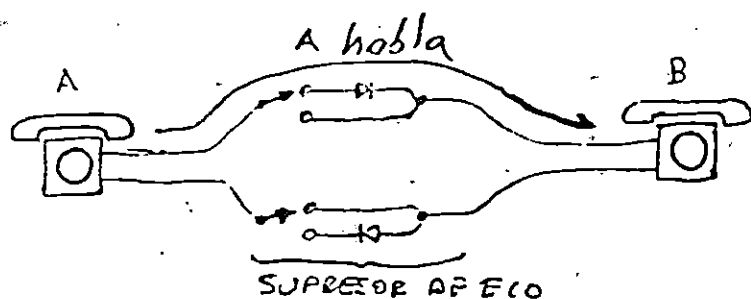
CONMUTACION JERARQUICA

redes telefónicas

EL PROBLEMA DE LOS SUPRESORES DE ECO.

12-4

- Unión Local Loop - Troncal $\rightarrow$ ECO.
- ECO $\rightarrow$ Dañoso para personas.: Supresores de eco.
- Instalados en líneas con longitud > 2.000 km.
- Tiempo de regreso ≈ 45 ms^{eqs.}
- Convierte línea FULL-DUPLEX en HALF-DUPLEX
- Tiempo alto en cambiar dirección ≈ 100 ms.
- Solución: se deshabilitan automáticamente si oyen un tono de 2100 Hz puro.
- Esta señal debe ser emitido por modem FDX durante 400 ms
- Aunque la transmisión HDX genera problemas en polling^(paseo) por tiempo de conmutación de dirección.



Supresores de eco

- Unión local loop-troncal $\Rightarrow$ eco
- Dañoso para personas
- + Supresores de eco

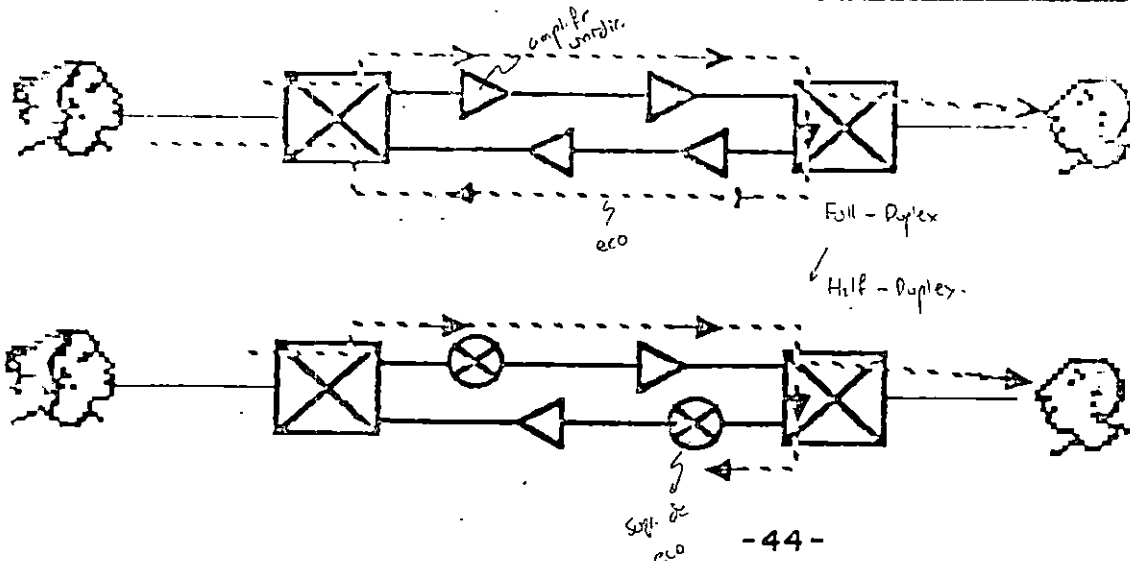
reflejo de la inf. cuando
ella pasa por el cable,

debido a
impedancia,
etc.

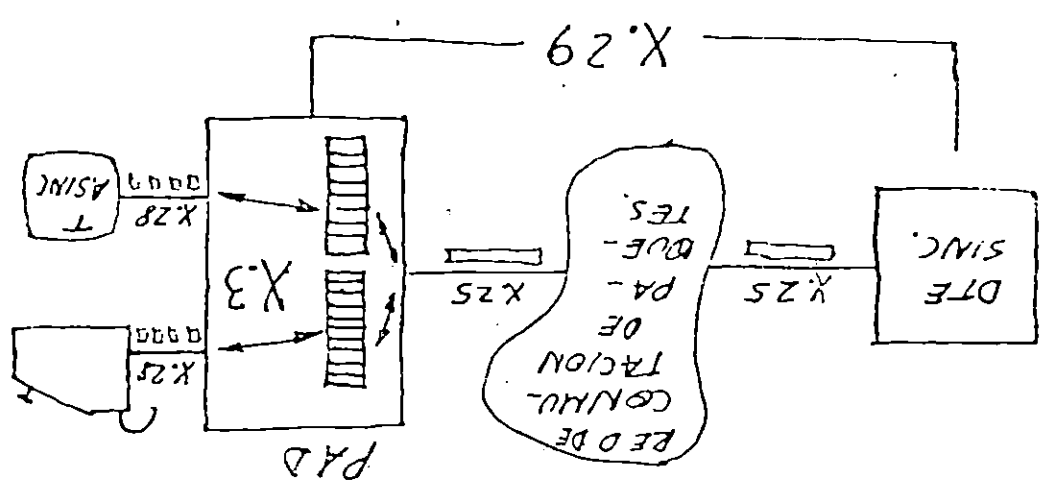
Supresores de eco

- Para líneas de long $> 2.000\text{km}$,
- t de regreso $\approx 45\text{ms}$ $\Rightarrow$ incómodo $\Rightarrow$ "Sd Eco."
- Línea full-duplex $\Rightarrow$ half-duplex
- t de conmutación de $\approx 100\text{ms}$
- Se deshabilitan con tono de 2.100 Hz
- Módem la emite 400 ms
- + Problemas para polling
 - t de inversión de $\approx$ muy largo,
en términos de tráfico de datos

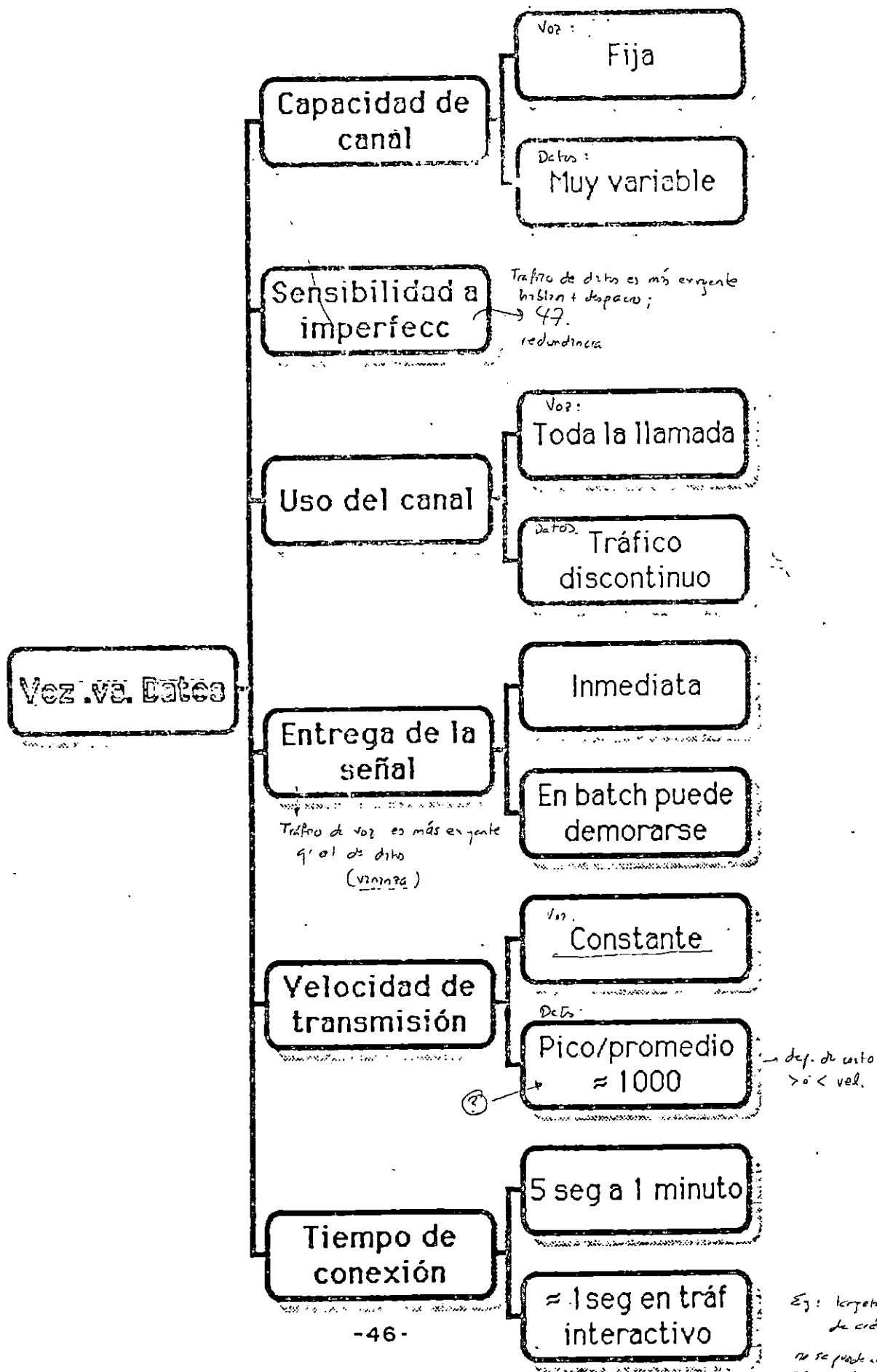
Polling.
asignar t a cada par
q'
pueda
hacer



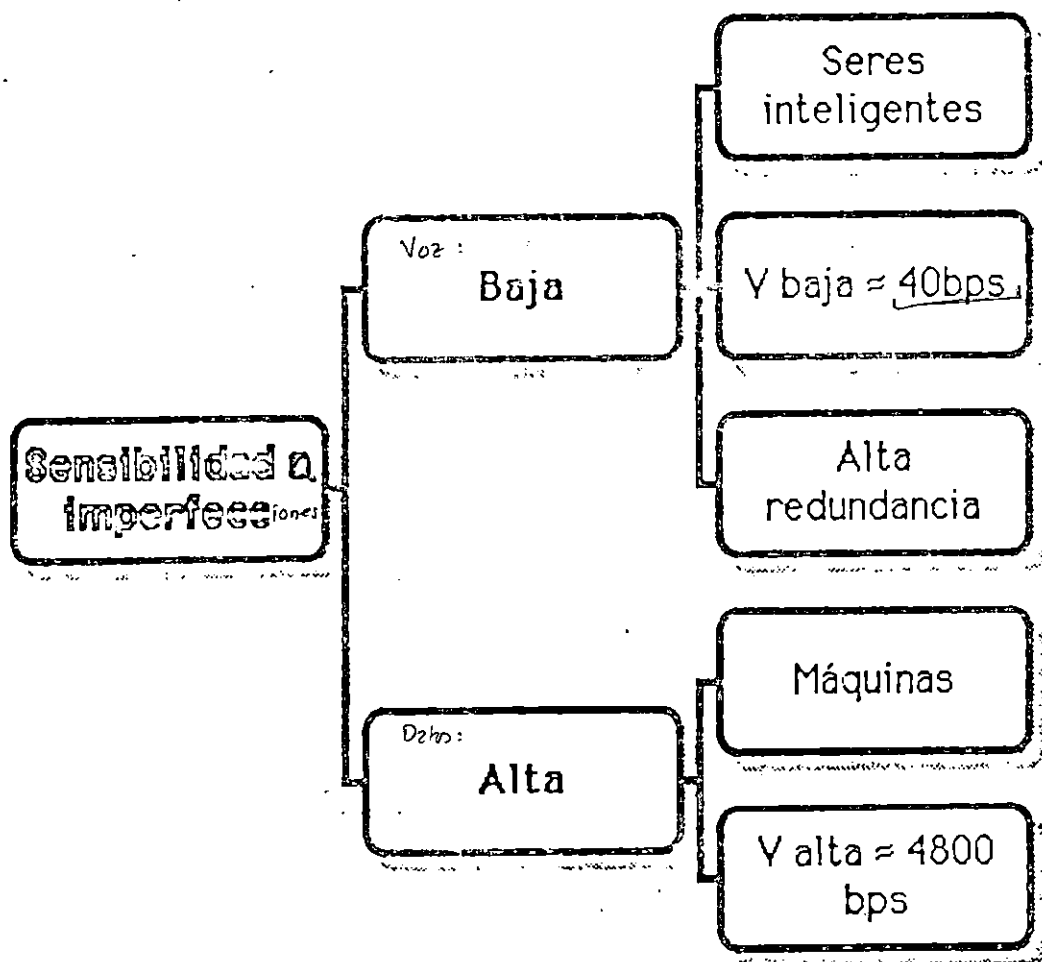
- X'25 trabajo con paquetes, 760 caracteres
- Terminales asincrónicas envían caracteres
- Que hacer para conectar T-Asincrónicas a X25?
- * Usar un PAD entre lo terminal y lo red.
- Como caracteres transmitidos asincrónicamente
- Les quito su propia información de control.
- Los agrupo en paquetes por medio de buffers.
- Coloca información de control X.25
- Lo envío a la red.



- X.28: Protocolo entre DTE Asincrónico y PAD
- X.29: Maneja intercambio de paquetes de datos y define significado de paquetes de control que permiten instruir al PAD sobre manejo de terminales asincrónicas
- X.3: Define parámetros del PAD
- Mensajes intercambiados PAD-Terminal
- Métodos de recuperación
- Parámetros de manejo de terminal



Ej: tarjetas
de crédito
no se puede recibir
q' por el cliente
se eq. me crean (t)

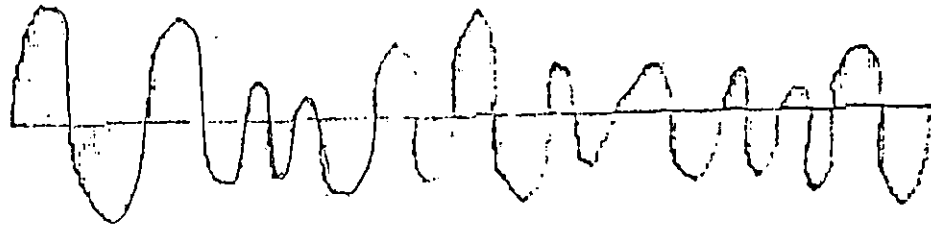


CANALES

- + Señalización $\leq^A_D$
- + Módem $\leq^{\text{Pro}}_{\text{estándar CCITT}} \begin{matrix} \text{Modem} \\ \text{Modem} \end{matrix} \begin{matrix} \times \text{Pro} \\ \times \text{Pro} \\ \times \text{Pro} \end{matrix}$
- + Configuración $\leq^{\text{P-a-P}}_{\text{Multi-P}}$
- + Direccionalidad $\leq^S_{\text{HD}} \text{FD}$
- + Capacidad Máxima
- + Imperfecciones

+ Señalización

+ ^{señal} Análoga $\rightarrow$ Continua



- Amplitud y/o frecuencia varían sobre rango continuo
- Ej. Sonido, Luz, voz

- Red telefónica es análoga, por manejar la voz $\rightarrow$ hay distorsión por fase o retardo
 sen $\rightarrow$ popinazo/de la línea
 (espasmos p' $\uparrow$ la inductancia)

+ ^{señal} Digital $\rightarrow$ discreta



- Cadena de pulsos on/off (1/0)

- Tipo de señal no depende del medio $\rightarrow$ uno puede señalizar analógico o digital, indep. del medio de transm.

+ Señal Digital .vs. Análoga

+ Independiente del tipo de información

+ Información análoga puede digitalizarse $\rightarrow$ discreto: tomar instantes y medir amplitud

- Ej. Voz, Música, facsímil

Inf. digitalizada $\rightarrow$ señal de analógico
 $\rightarrow$ modem

- V (bps) proporcional a AB

Inf. analógica $\rightarrow$ digital
 Ej. disco laser

Video 10.1.100 6 MHz de AB

Red telefonica 4000 hz

Cuando se entrega un canal de TV

⇒ 5500. 1500 Mmidos telef. (6 MHz / 4000 hz)

Factores en contra

Mayoría de información analoga

Necesidad de conversiones

Digital
Requiere AB mayor

Repetidores; no amplificadores

Facilidad de mezclar información: ISDN

Factores a favor

Circuitos de codificación baratos hoy

Disponemos de gran AB

Menos sensible, sólo 2 valores posibles

Puede compensar
⇒ AB requerido

Señal Digital
vs. Analoga

req. AB
(cable + inf. + pte, amplif.)
100
(d, e, w, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, etc)

señal analoga
señal digital
debido a la
señal es 1 a 0
⇒ mejor
fpt. todo
digital

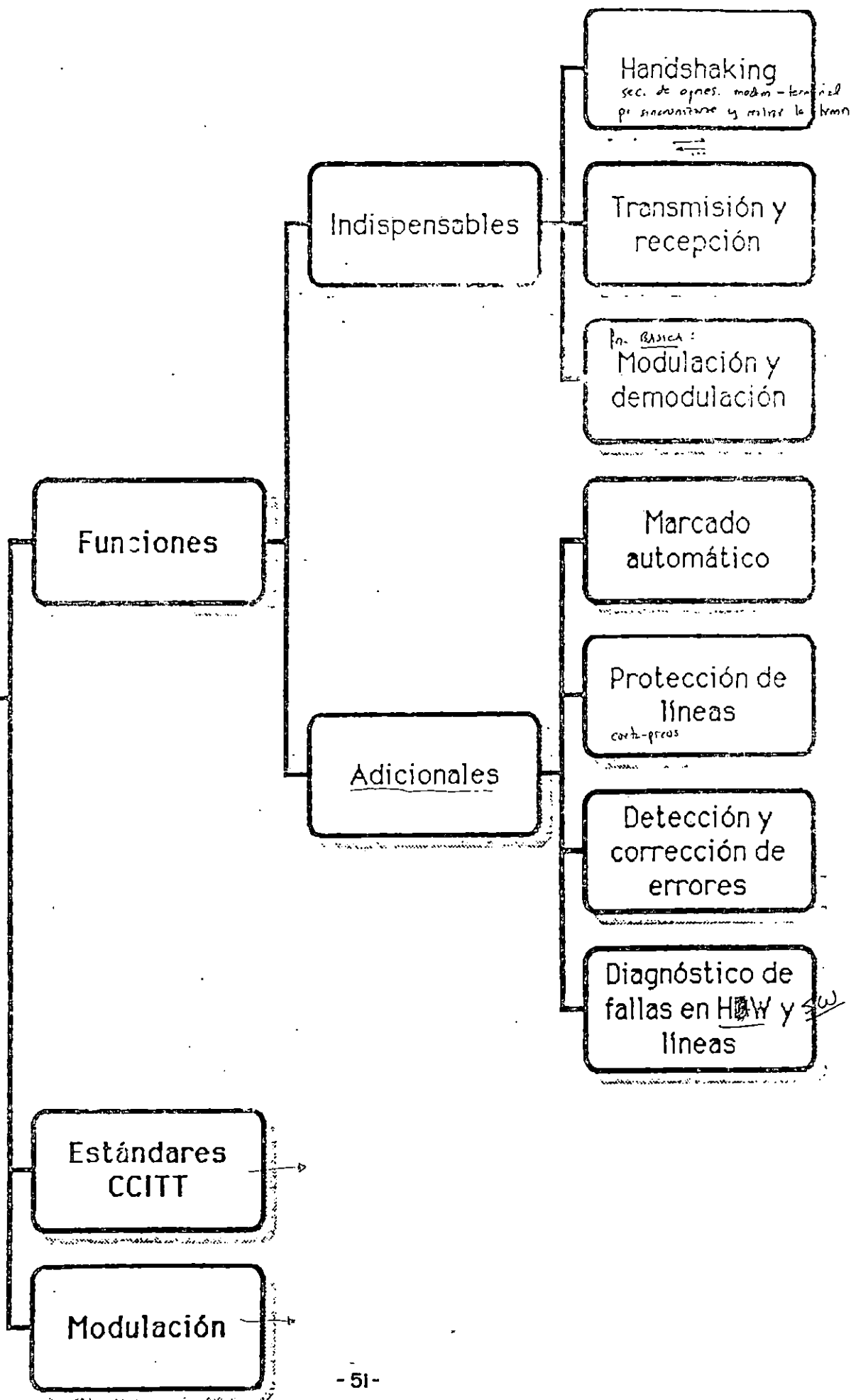
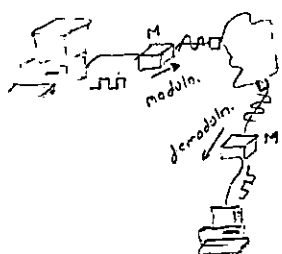
ed. hz
RDS
id. mezclar
haz h. imp
en 1 solo canal

2
Pox
1

satélites
líneas ópticas
c. coaxiales

Modem: Modulador/ Demodulador

aparelo q' se utiliza entre
otro. computador y la red
telefónica.



Estándares CCITT

estándar CCITT
 Y24: Interfaz
 modem terminal

RS232C

Y25:

Deshabilitación
 supresores de eco

cómo se hace
 la Llamada
 automática

estándar
 Y21: 200 baudios

Y23: 600/1200
 baudios

Para Línea
 conmutada

1200 / 2400
 Y26b: 2400/2400
 bps

llamada x l. conmut.
 x 2. alquilada

Y15: Acoplador
 acústico

especie de
 "modem"
 p/ tr. inf.
 a rel. muy
 lenta

Modulación
 de código: "5" = 5 pulsos en
 del. Pres.
 no ll. presente.

colgar → poner un tono a 2600 hz
 → el mod. de código de
 modulación a 2600 hz puede
 colgar la llamada.

Y26: 2400bps

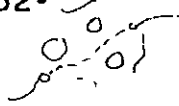
Línea alquilada
 dedicada, arrendada
 de todo a todo

Y27: 4800bps

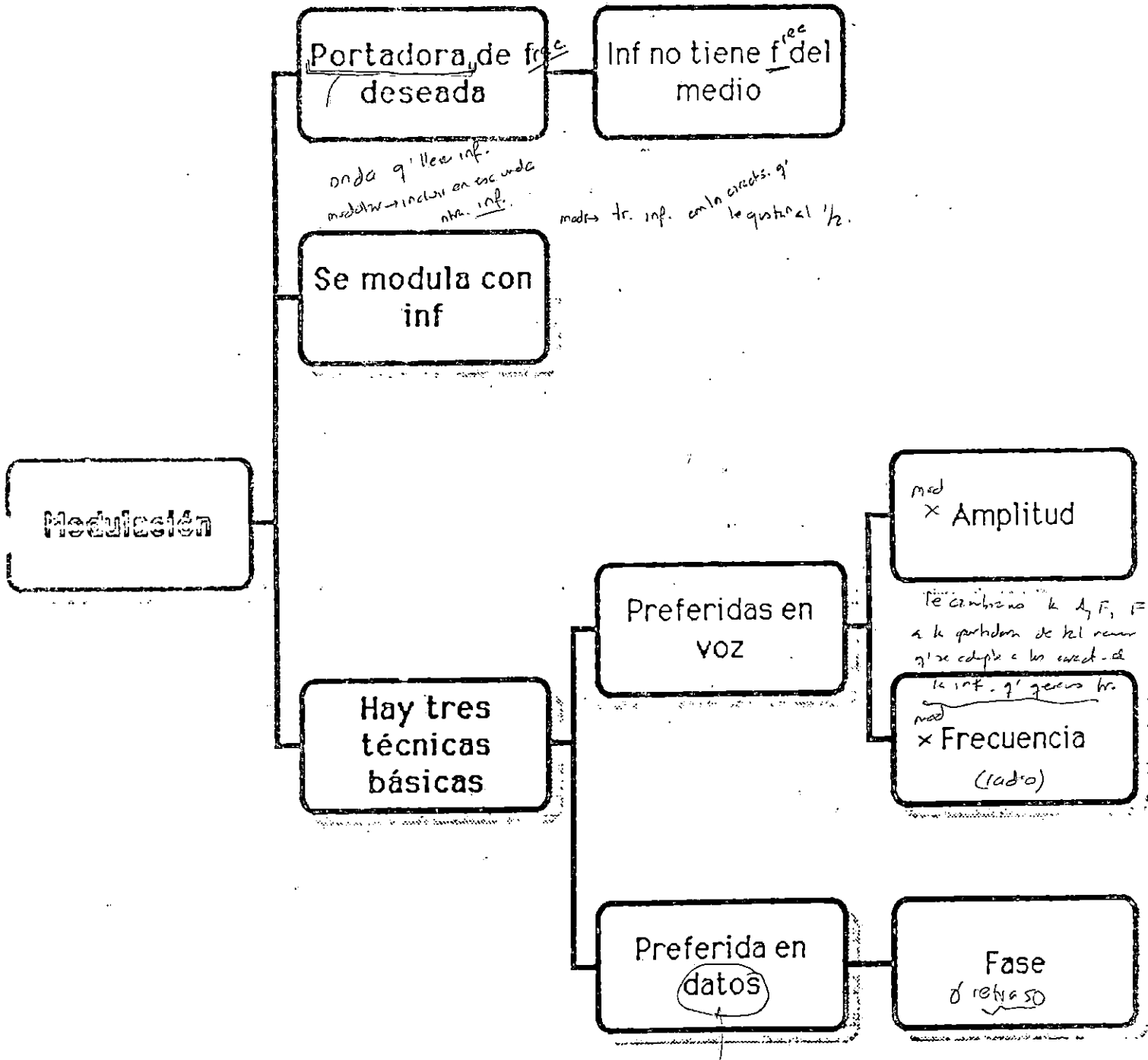
y q' no pasa por
 los centros de
 conmut. y por tanto
 no está expuesto a ruido
 e interf. cruzados
 -52-

Y36: 48000bps
 usando grupo
 p₃ = (60² - 108)

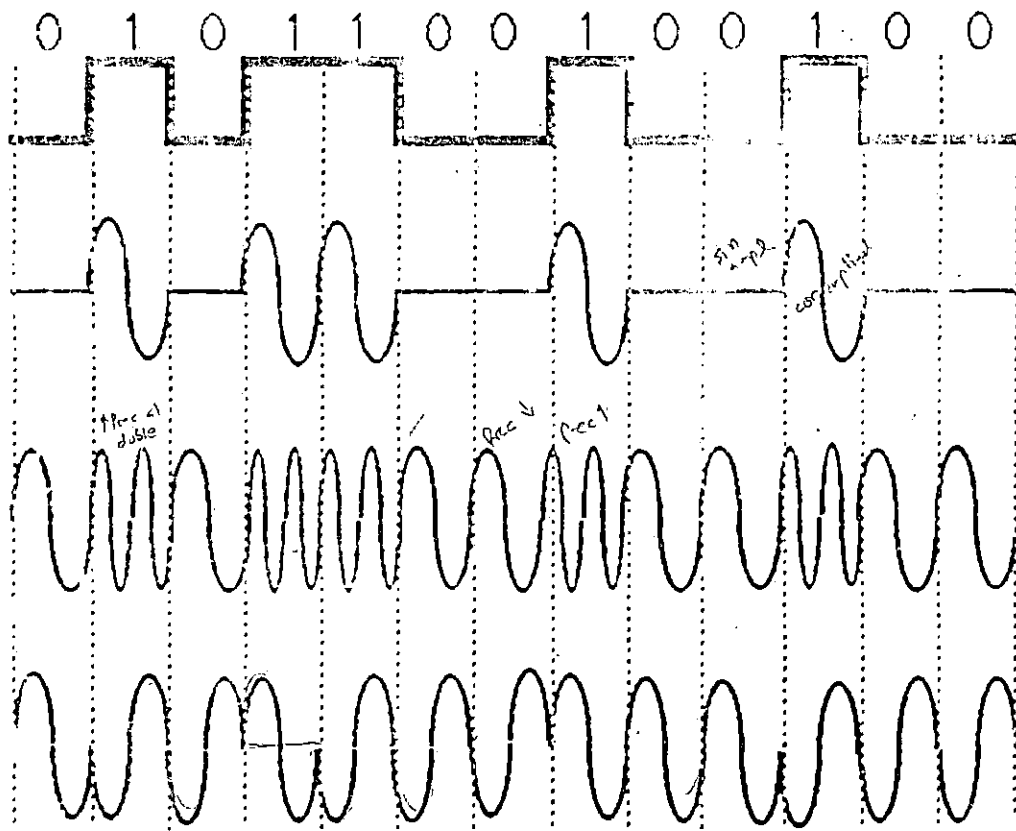
12 líneas al 4.



$\frac{1}{2}$ de f_c $\downarrow$
 $f_{MO} \rightarrow$ para a elección de f_{c}

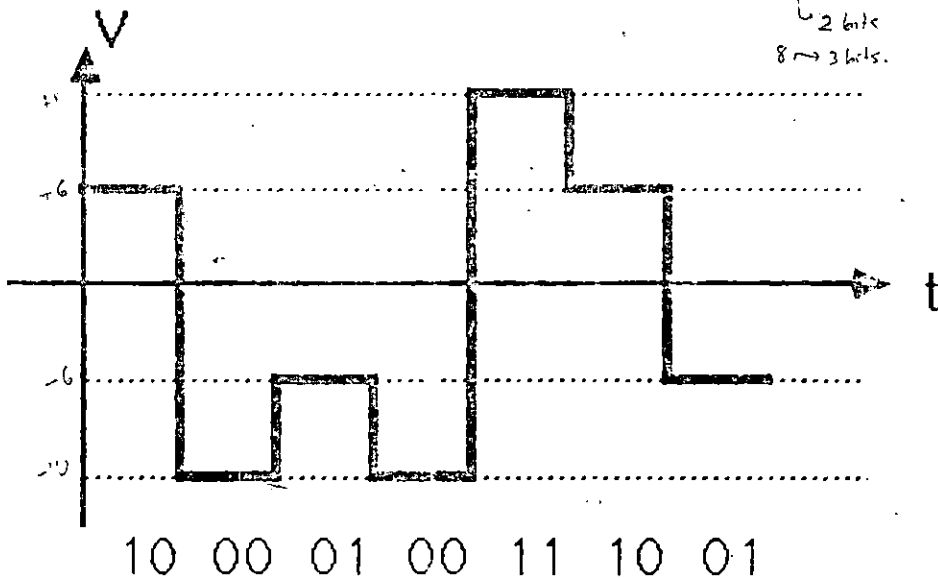


MODULACION



0 → no Δ
1 → alto si

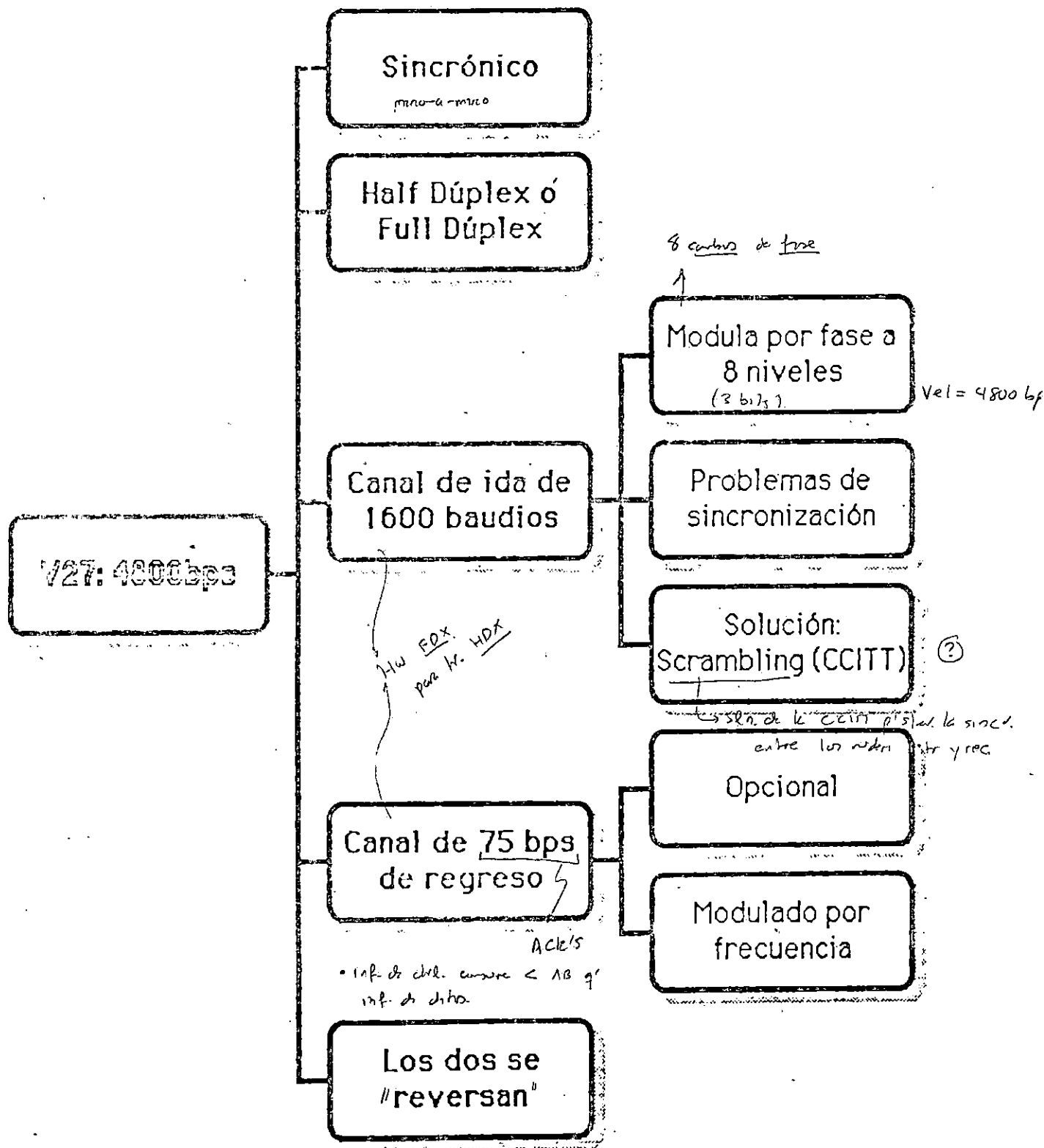
MODULACION A VARIOS NIVELES (4 en este ej.)



baudrate ~ tasa de símbolos/seg y de el tr. y q' modula el rec
vel ~ baudrate * $\log_2(\# \text{ niveles})$.

Diferencia entre baudio y bits/seg

Σ: modula a 4 niveles



Modula por fase a 8 niveles

- 001..... 0°

- 000..... 45°

- 010..... 90°

- 011..... 135°

- 111..... 180°

- 110..... 225°

- 100..... 270°

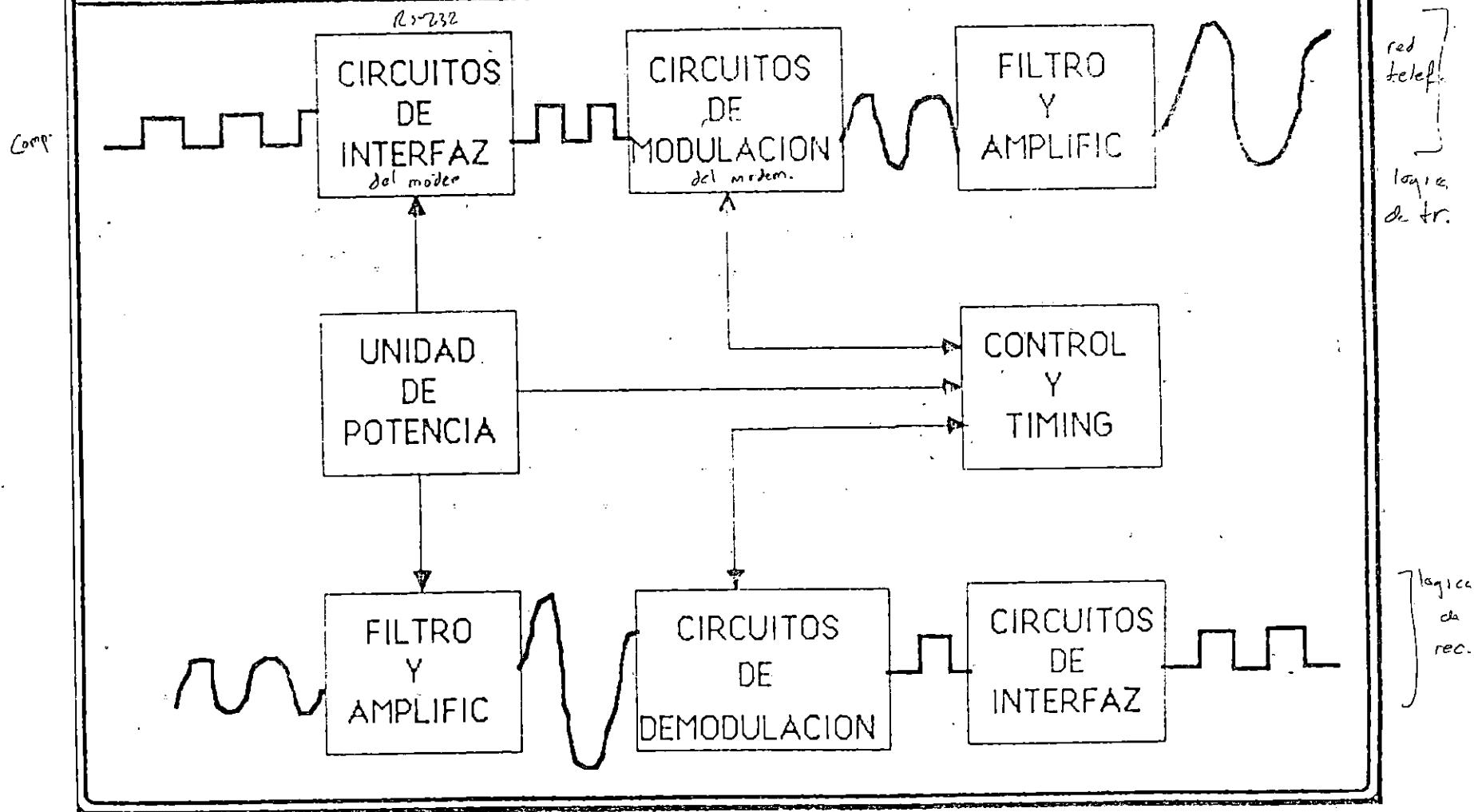
- 101..... 315°

- Tren de tribits 001 ~~no~~ cambia fase

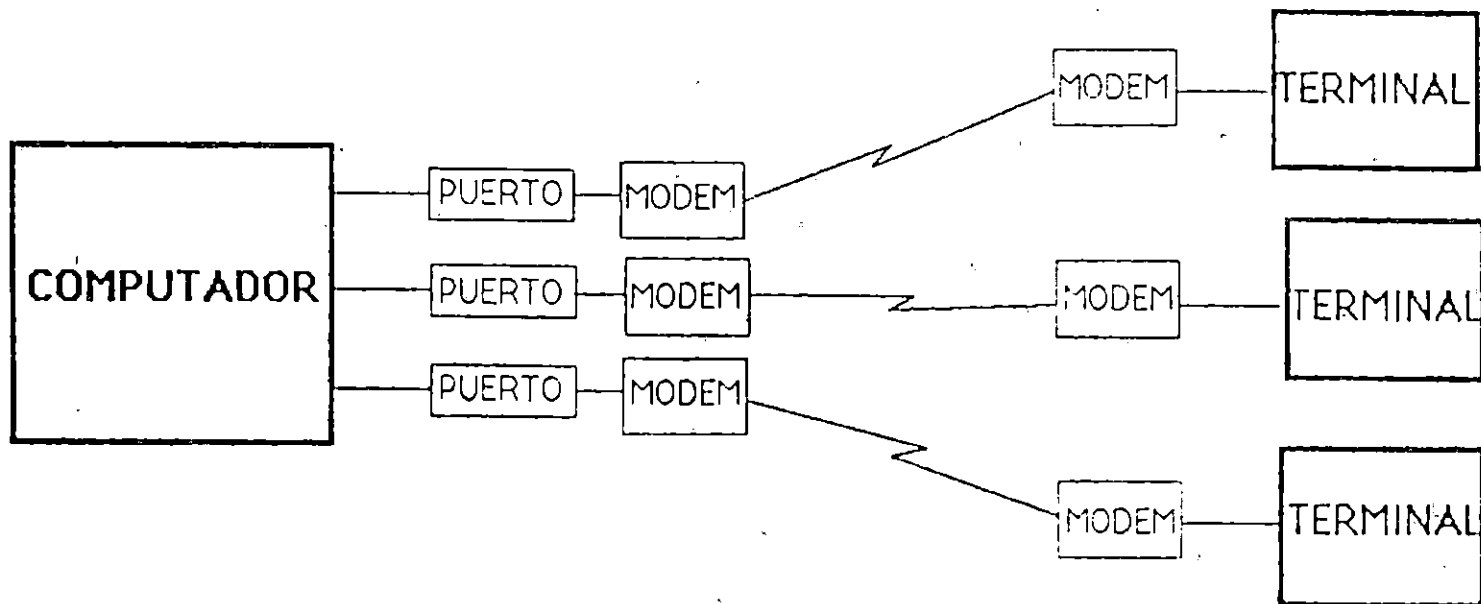
000 $\rightarrow 0^\circ$ millo dea!

Ej: muchos cambios de fase
el receptor no detecta
dado con una nr. señal ó ciclo
(problemas de sincronización).

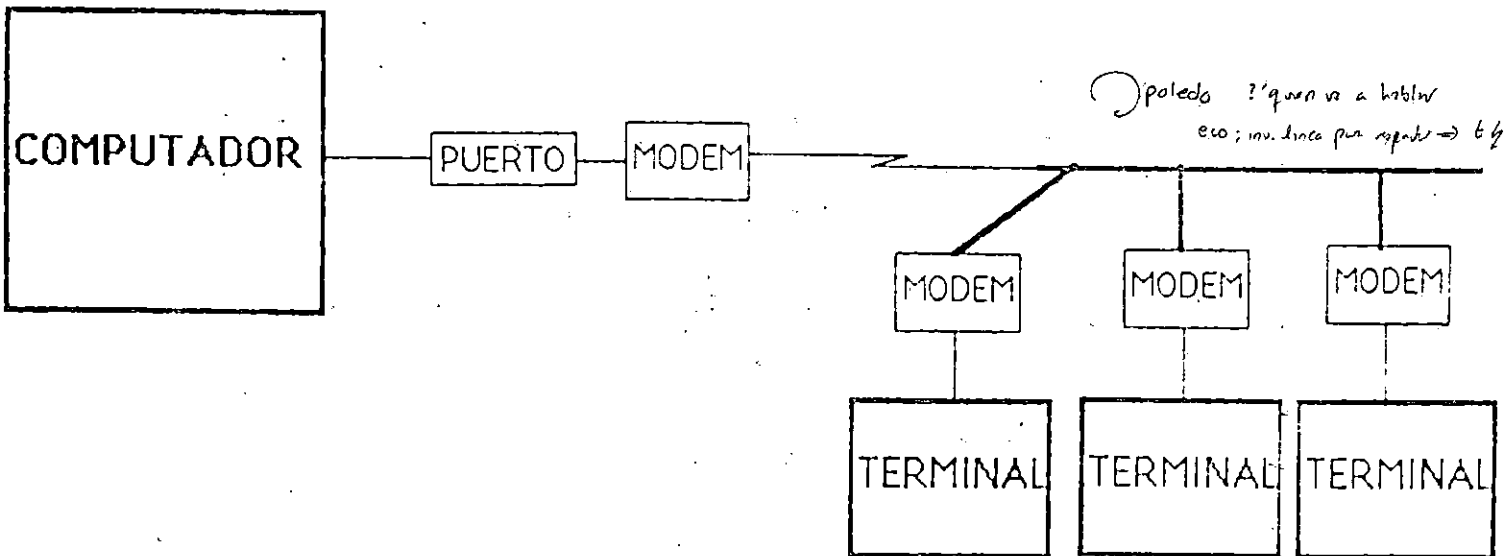
Módem: Modulador/Demodulador



Punto-a-punto

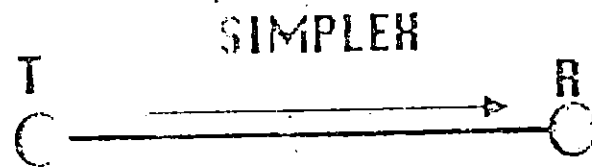


Multipunto

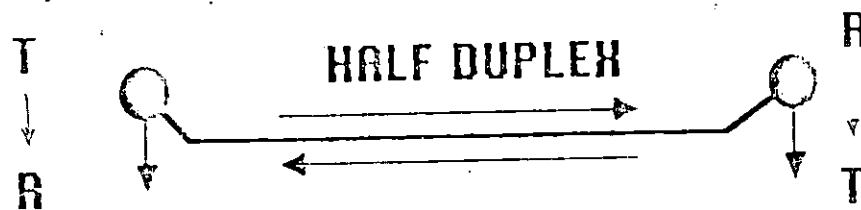


+ Birecionalidad

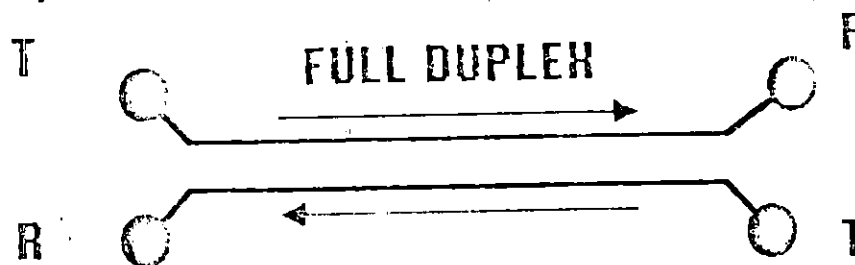
+ Simplex



+ Half Duplex



+ Full Duplex



Imperfección
es
en los cables

Distorsión

producido por los cables.

líneas de la línea.

De atenuación

Ruido

prod. por los fen. de
Capacitancia e
inductancia

De fase o
retraso

de π ang
→ unos frenos V_{origen} + rapidez q. otros
Reflexión → pues espaldas de carga (bobina) por
= la vel. de propagación.

Pérdida de
potencia

de el V_{orig} & el canal

$1\text{db} = 10\text{Log}(P_{out}/P_{in})$

P_{out}/P_{in}

Medida
relativa

Se mide en
decibels

medida "relativa"

Ejemplo :

rel pot/onda = 30db.

Usados para
relacionar S/N

1000 V. + potencia
señal q. el ruido.

$30\text{db} = 10\text{Log}(?)$

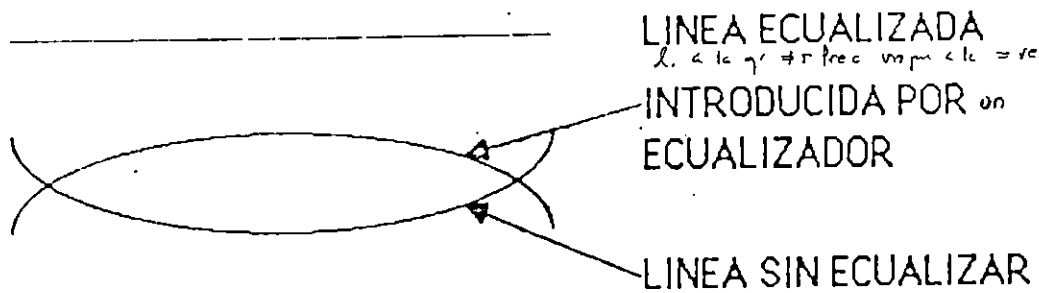
(10^3)

Wojing

→ 64

De fase o retraso

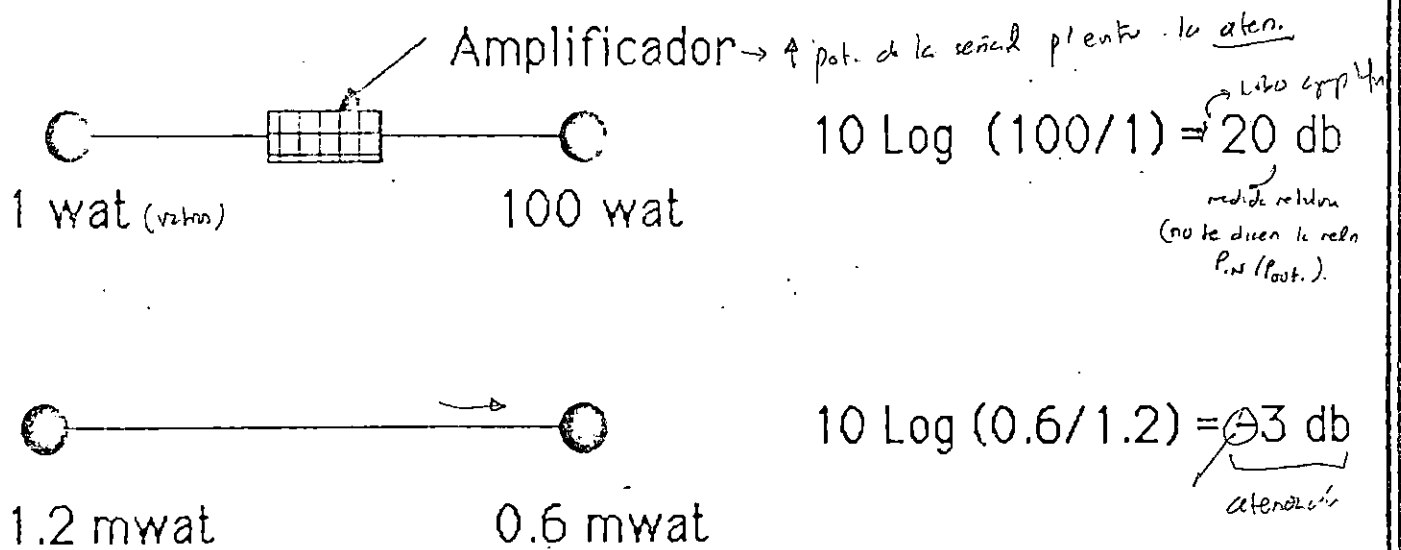
Velocidad de propag.
DISTORSION (Demora)
 en la entrega
 de la señal.



FRECUENCIA

expresión
 de carga.
 /
 problema de
 la L. Lupa

Ejemplo de medición ^{da} de atenuación



pero en un canal hay + de un bit por
c/sense $\rightarrow$ la velocidad en un
canal

Ruido

de Impulso
(problema de los comas.)

Impredecible
ni cuando va a pasar,
ni la prec. con q' va a pasar

Corto e intenso
durada de $1/100$ s.

Fuente ppal de
errores en DP

Puede durar $\approx$
0.01 seg
 $1/100$ seg.

En datos.
Suficiente para
dañar 50 bits a
 $V=4800$ bps

Se oye como
click en el
teléfono

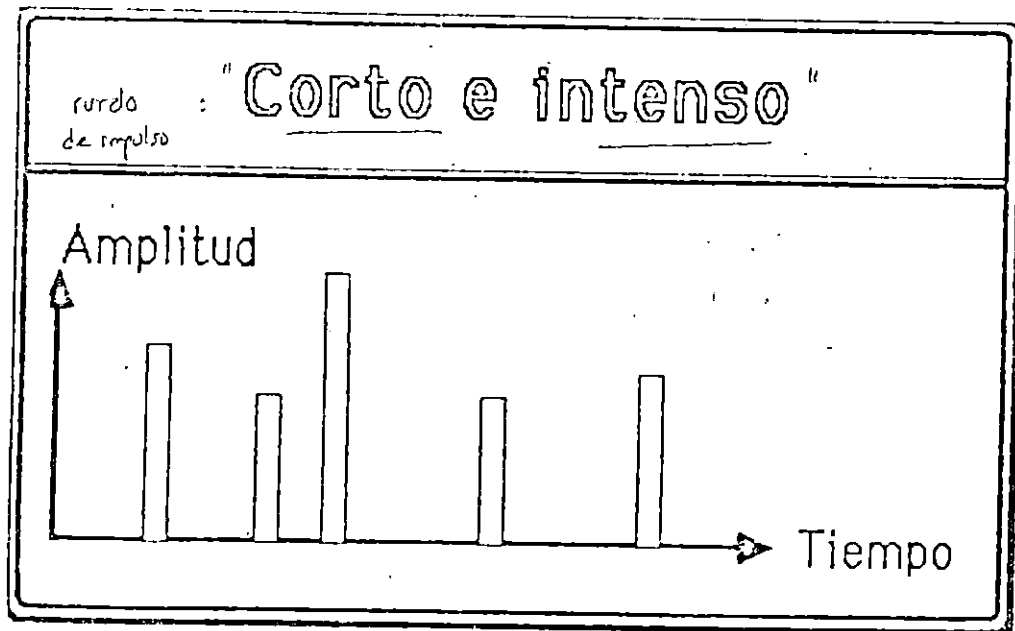
No hace daño a
voz

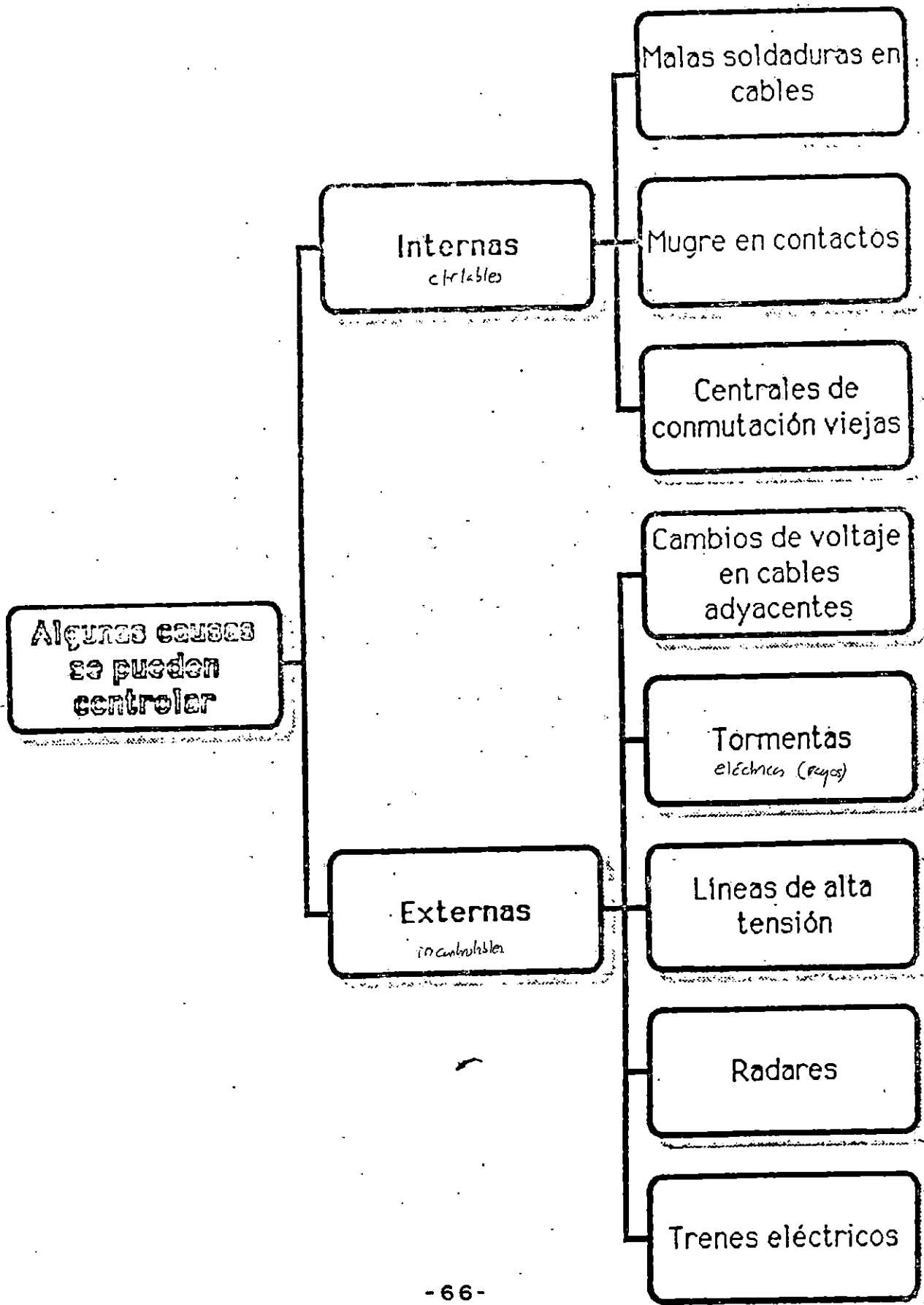
Algunas causas
se pueden
controlar

Internas
controlables

Externas
incontrolables

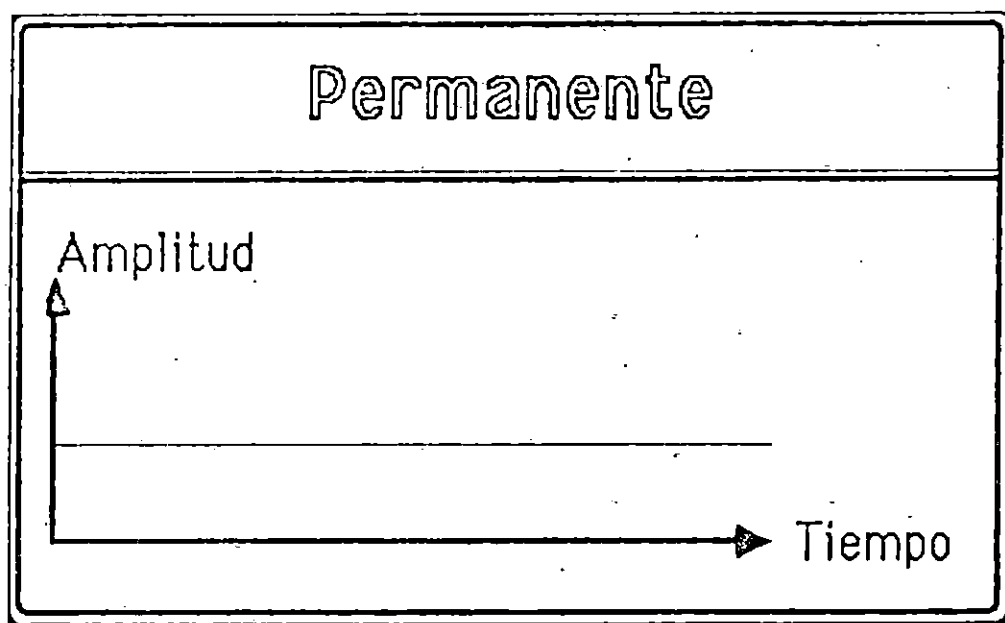
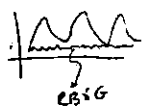
Blanco o
Gaussiano





Blanco o Gaussiano

- Causado por inducción característico de los
señales electromagnéticas
- No puede evitarse
- Permanente
- Zumbido permanente en teléfonos
- Puntos en TV
- Se controla manteniendo $S/N > 30 \text{ db}$



Capacidad Máxima

- + Nyquist-(1924) ^{considero} Canales "perfectos" → (may teórica)
- + Shannon-1948 ^{considero lo} Canales con ruido →

Nyquist-1924
Canales perfectos

$$\text{BAUDIOS}_{\text{máx}} = 2 \times AB$$

banda disponible
en una línea es
 $2 \times AB$
(la vel de 1 # niveles
en q' modulamos)

Ejemplo

Si $AB = 3\text{KHz}$

$$\text{BAUDIOS}_{\text{máx}} = 6.000$$

Con modulación a

2 niveles $\rightarrow 6000$
bps

4 niveles $\rightarrow 12000$
bps

8 niveles $\rightarrow 24000$
bps

$(\log_2 8) \cdot 6000$

Shannon-1948 Canales con ruido

$$C_{\text{máx}} \text{ en (bps)} = AB \log_2(1 + S/N)$$

Independiente del
#niveles

Canal telefónico

$$AB = 3000 \text{ Hz}$$

$$S/N = 30 \text{ db}$$

$$\text{rel } \frac{S}{N} = 1000$$

$$C_{\text{máx}} = 3000 \log_2(1 + 1000)$$

$$\log_2(1 + 1000) \approx 10$$

Ejemplo:

$$C_{\text{máx}} = 30.000 \text{ bps}$$

$$9600 \text{ bps} = B \cdot 4$$

$$\rightarrow B = \frac{9600}{4} = 2400 \text{ bps}$$

Se logran
modulando a 16
niveles

es excelente en
canal de voz

Se logran
modulando a 16
niveles

$$C_{\text{máx}} = 4 \cdot 2400 = 9600 \text{ bps}$$

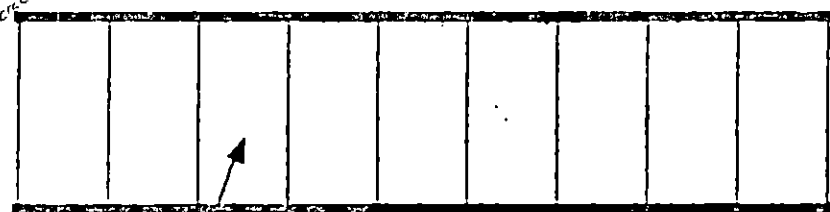
se da en el
según el caso

+ MULTIPLEXACIÓN → combinación de recursos (vías de transmisión).

- AB depende del medio de transmisión
- Si AB disponible >> AB requerido --> multiplexar
- Por tiempo (TDM) → señales digitales

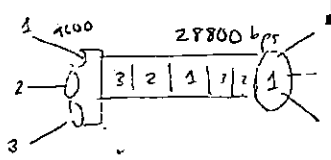
TDM

moduln → proceso de modular
una portadora para llevar
inf. a través de ella



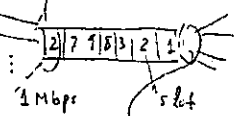
SLOT

Ej: este canal → capacidad de tr. a 10 Mbps



se encarga de producir en el t
decide de q' estuán le
llega la inf.
c/slot en 1s.

multiplexar:
buffer q' recibe inf. por t < c/micro

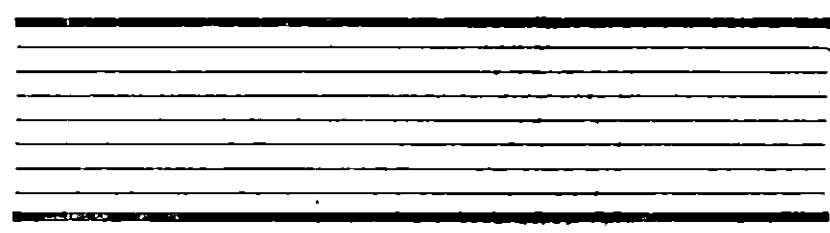


demultiplexar (HCl o SW)

+ Por frecuencia (FDM) → usada en señales analógicas

FDM

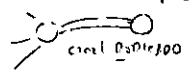
Desventaja →
desperdicio de t, si alguna
estación no tiene nada q'
transmitir



Subcanal

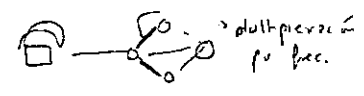
g01 → construye de p'q → se usa la
f'ult x t

Código → result. por frec

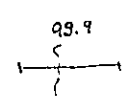


La red telef. convencional se está
conv. en digital ⇒ f'ult x t
(d.f. 1.000)

Ejemplo: Radio AM y FM



Cancel estereo

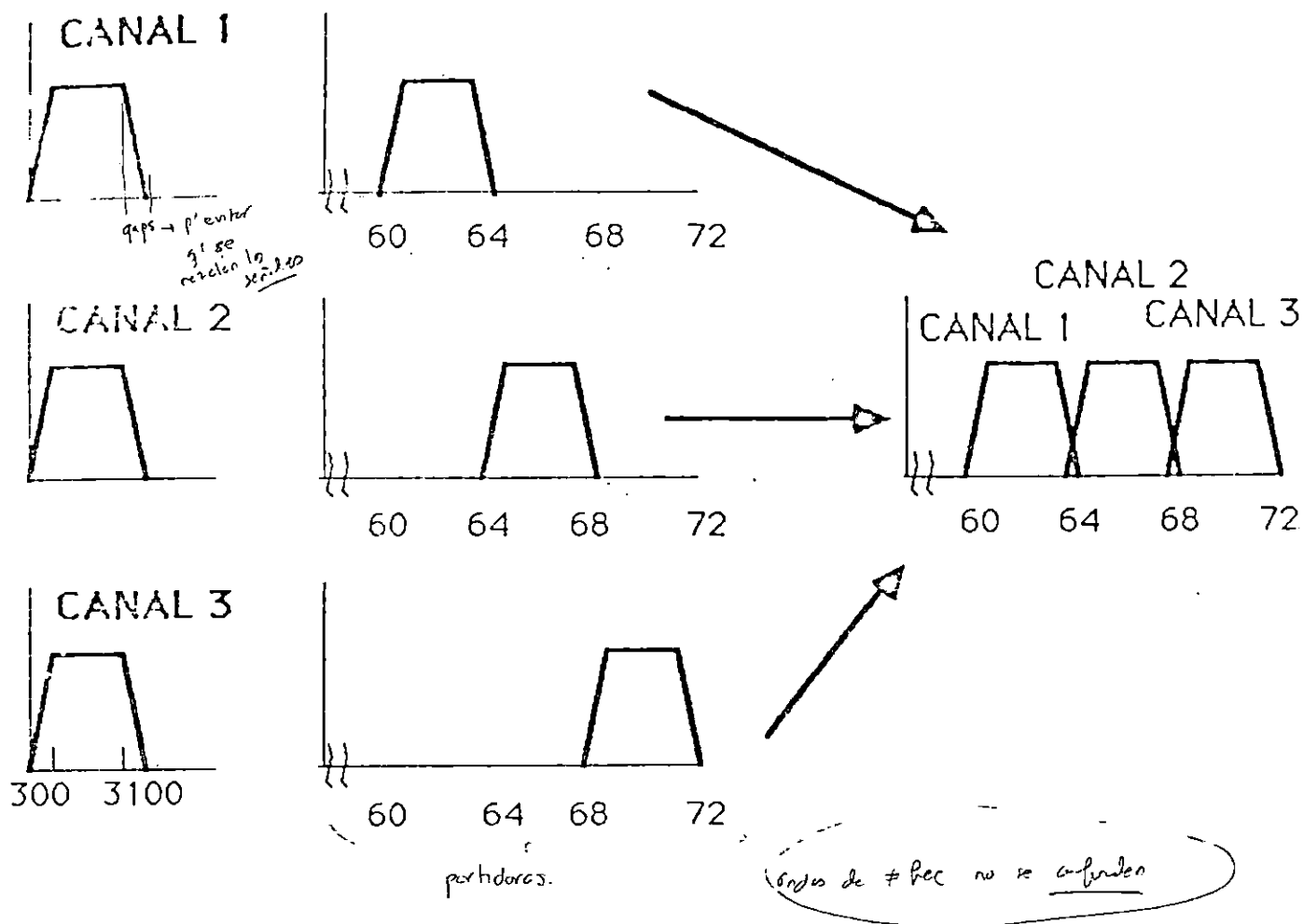


$$f_0 = 99.6 \text{ MHz} \pm 100.1 \text{ MHz}$$

emisor → modula la frecuencia original (99.9)
p' tr. su señal (portadora)

AB = 0.15 MHz

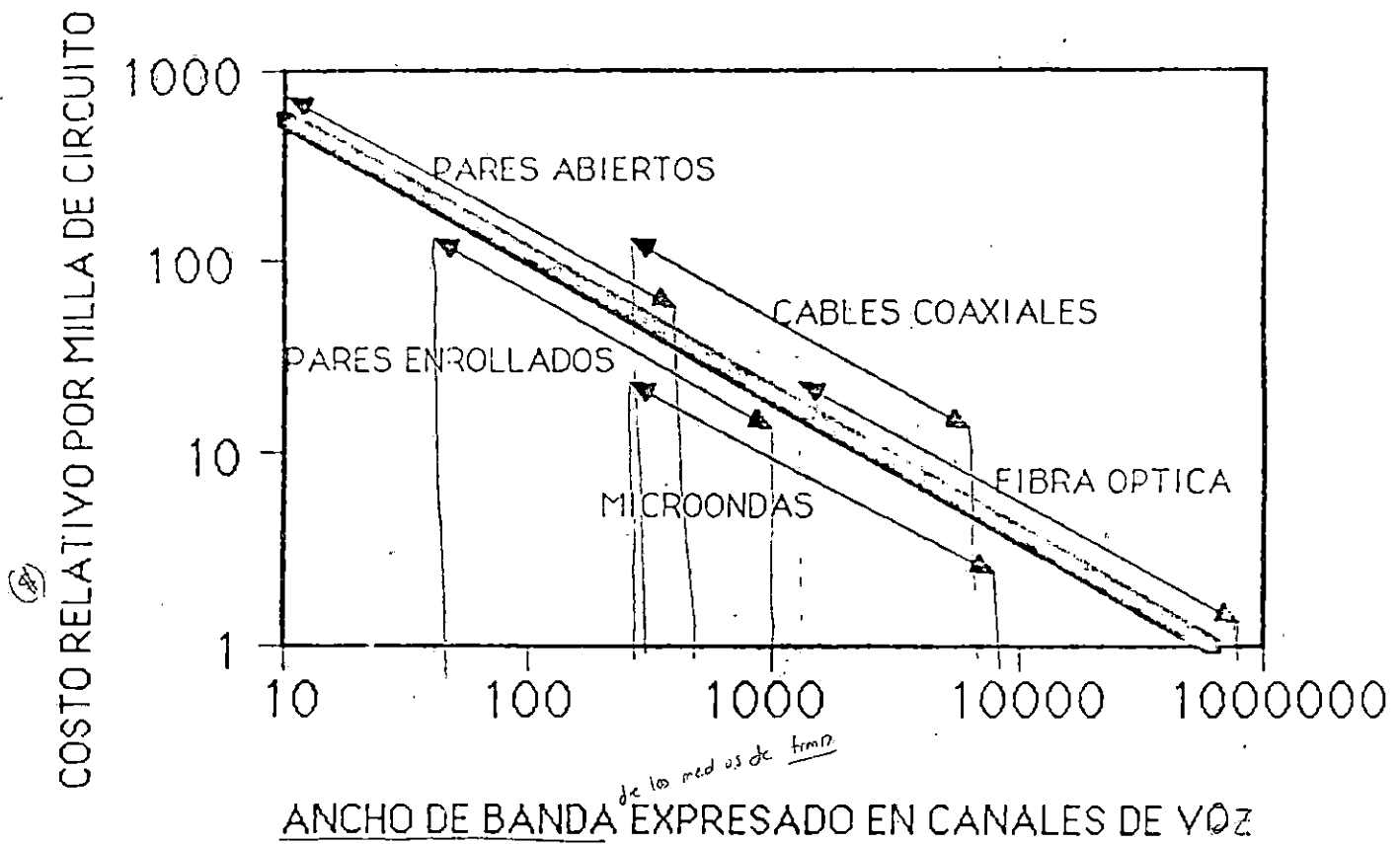
Señales de radio, TV
multiplexadas en el t're
cuando intermiten, estuán
demultiplexadas x señal



MULTIPLEXACION POR FRECUENCIA

(en la l. telefoniz.)

MEDIOS DE TRANSMISION



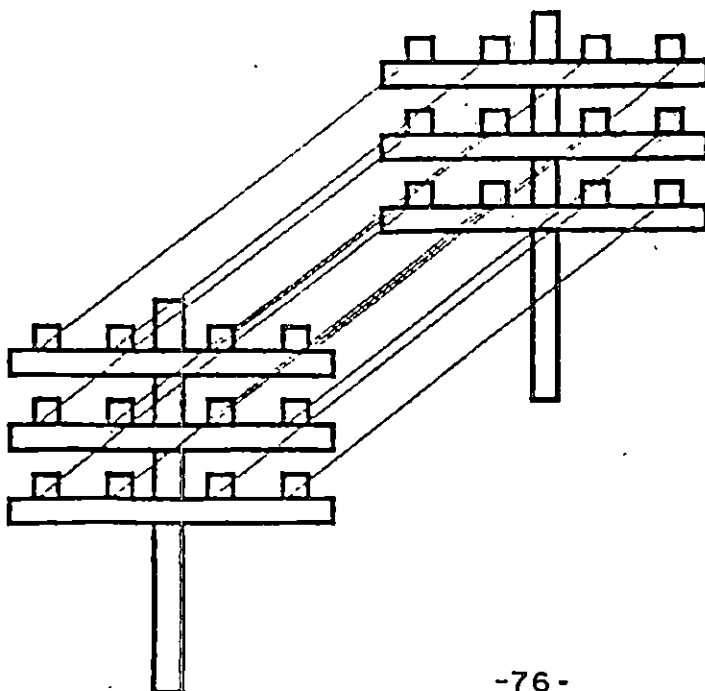
Costo de comunicacion por satelite no dep de la dist.

MEDIOS DE TRANSMISION

- 1) Pares abiertos
- 2) Pares enrollados
- 3) Cable coaxial
- 4) Microondas
- 5) Satélites → Costo de com ~~no~~ dist
- 6) Cables submarinos
- 7) Radio de alta frecuencia
- 8) Circuitos de dispersión troposférica
- 9) Radio de corta distancia
- 10) Guías de Onda
- 11) Fibra óptica

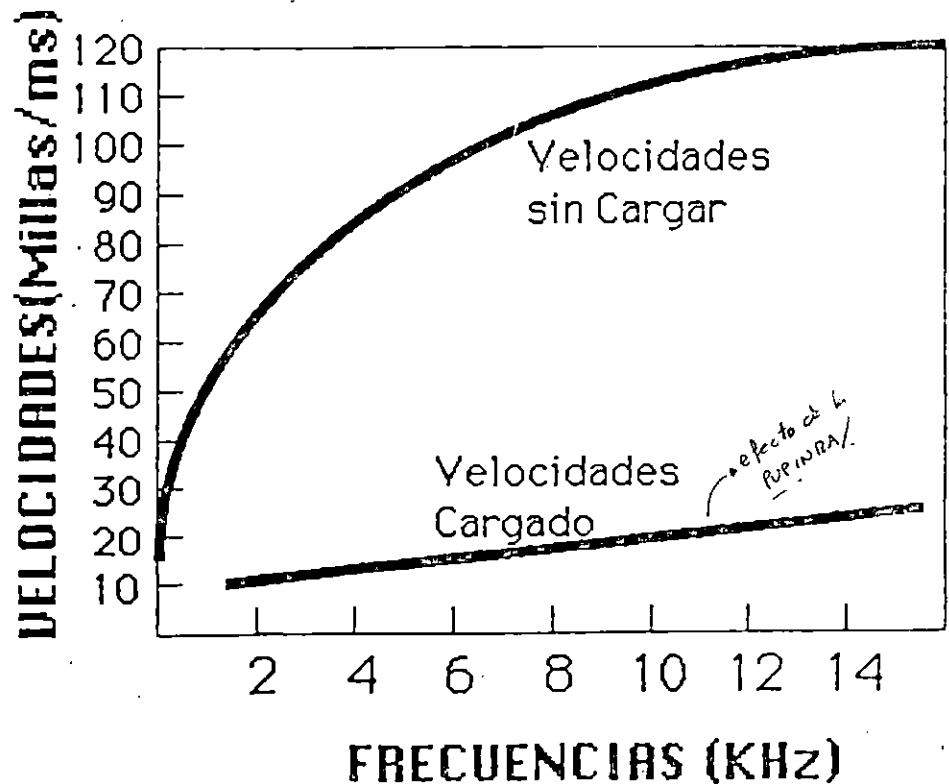
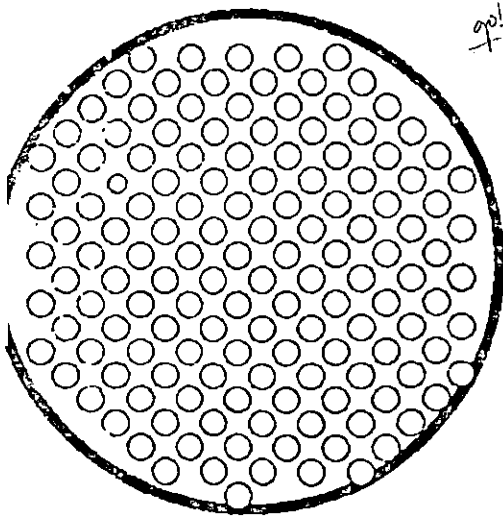
1) Pares abiertos

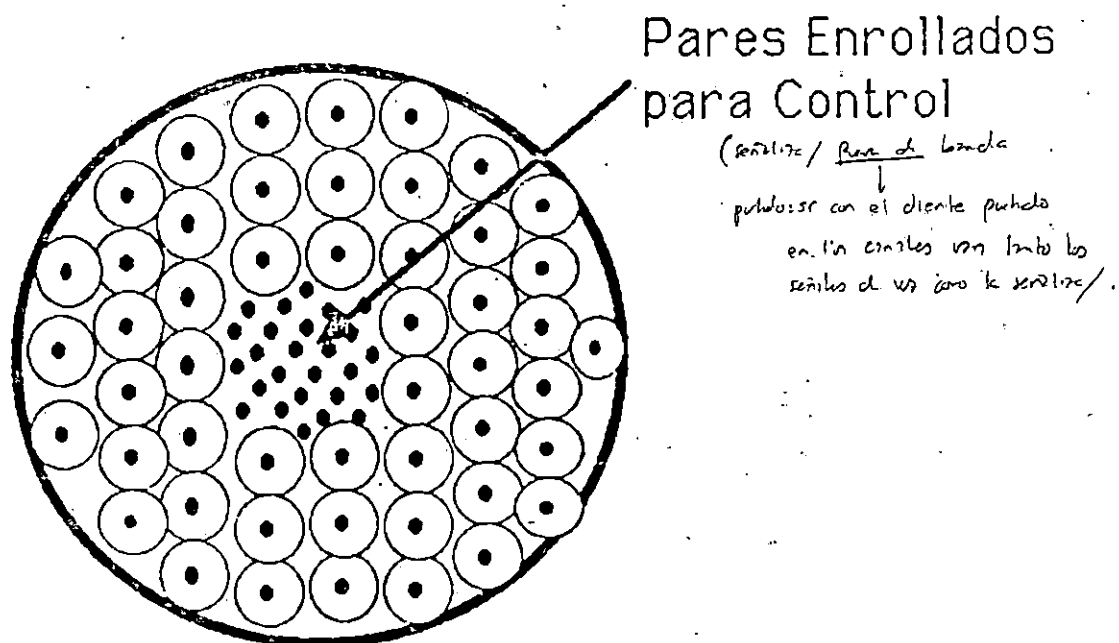
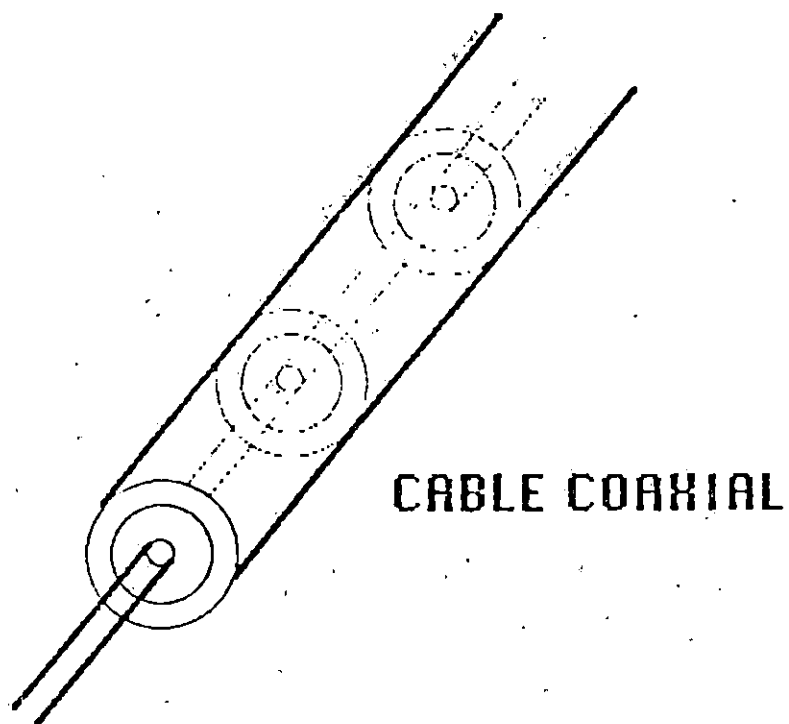
- Cobre o acero recubierto de cobre (costo, ductilidad, efecto de piel → a frecu. altas, la inf. viaja por la superf. del cable. ⇒ probl. de resist ⇒ atenua/.
- Si $f_{rec} > 1000$ KHz corriente viaja por superficie
- Diámetro aproximado = 3mm
- Espaciados entre 20 y 30 cm
- Susceptibles a Crosstalk (cuando uno está hablando x teléfono uno a veces escucha otras conversaciones. c/puñeta)
- Capacidad baja. Típico: 24 canales de voz multiplexado $\Rightarrow AB = 24 \div 4 = 96$ KHz
- Se usó con éxito en distancias largas
- New York - Denver sin amplificadores
- Afectados por el clima
- Poco usuales en áreas urbanas



2) Pares enrollados

- Reemplazaron a ^{los} abiertos
- Los conductores se aíslan están reb. x plásto
- Cientos se agrupan en un sólo cable
- Se entrelazan a distancias variables ^{para evitar el crosstalk}
- Diámetro entre 0.5 y 1.5 mm según dist
- Requieren ^{de} amplific^{adores} c/4Km aprox?
- Hoy se logran $f > 1\text{MHz}$ ^{requiere 2} $0 < f < 1\text{MHz}$
AB "estando con" PB
- Más sensibles a crosstalk
- Se cargan para evitar distorsión de retraso
(propagación)





**TRONCAL TELEFONICA A
 BASE DE CABLE COAXIAL**

3) Cable coaxial

- Consta de:
- Cilindro hueco de cobre u otro conductor
- Este cilindro rodea un cable
- Espacio lleno de aire o aislante
- No hay Crosstalk → *por efecto de piel, si la velocidad $c/v > \text{precis}$, la señal hunde a [dirección] el centro (bien aislado).*
- LLeva hasta 10.000 canales de voz
- Se logran AB hasta de 40Mhz
- Distorsión de retraso mínima
- Velocidad de propagación similar a la luz
- Requieren amplificadores $\approx C/5\text{Km}$
- Buenos para distancias cortas (*redes locales*)

Costo

300k $\frac{\text{Km}}{\text{s}}$

4) Microondas

- Usadas en distancias largas
 - Gran AB $\approx 10,000$ MHz 10GHz
 - Usadas en TV ^Δ
 - Transmisión en línea visual
 - Torres repetidoras $\approx C/50\text{Kms}$
 - Diámetro antena $\approx 3\text{m}$
 - Alternan TV, teléfonos y datos
 - 1 canal de TV ≈ 1200 ^{canal} de teléfonos
 - Afectadas por ^{la} lluvia (1/2 la atenuación).
clima, humedad, ---
- Necesitan q' los enlaces estén estables en línea dta.

plg crítica del
disño

h - s

se aliam en los centros

5) Satélites

- a) Generalidades :
 - No Geoestacionarios
 - Geoestacionarios
- a) Componentes fundamentales
- b) Costos
- c) Repartición del espacio
- x) Problema de la demora
- s) "Eclipses"

b) hqo

El espacio p' colocar satélites es limitado

El problema es muy grande

Satélites formen $\subseteq$ de la sol $\Rightarrow$ si hay eclipses no funcionan.

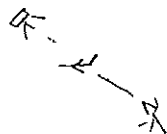
dia la

Tierra: 1 h, 43^{das} días { Prim. otoño

Luna: < frecuentes

Sol detrás: 10 min, 5 d. consecutivos
2 veces/año

Aviones



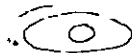
Generalidades

satélite $\Leftrightarrow$

- ✓ Estación de ondas en el cielo
- Sin problema de obstáculos por altura
- $f \approx \mu$ ondas $\Rightarrow$ estaciones lejos de ciudades
- Nuevos satélites evitan problema con $f_s > s$
- f de transm $\neq f$ recep
- Transmisión broadcast
 - muchas estaciones terrenas
leen las señales del satélite
 $\hookrightarrow$ seguridad $\Rightarrow$ criptografía
- Técnicas de seguridad sofisticadas

$P_s \neq P_b$
 por no recibir
 la señal

1^{er} 5^a) No Geoestacionarios

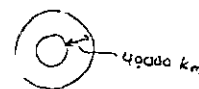


→ en ch. le
deben la wellh
la tierra

- Orbita elíptica con $v_{orbita} \neq v_{rot}$ de la tierra
- Nació TV trasatlántica (entre continentes)
- Visibles sólo parte de la órbita
- Programas ^{de TV} muy cortos
- Antenas móviles

5 b)

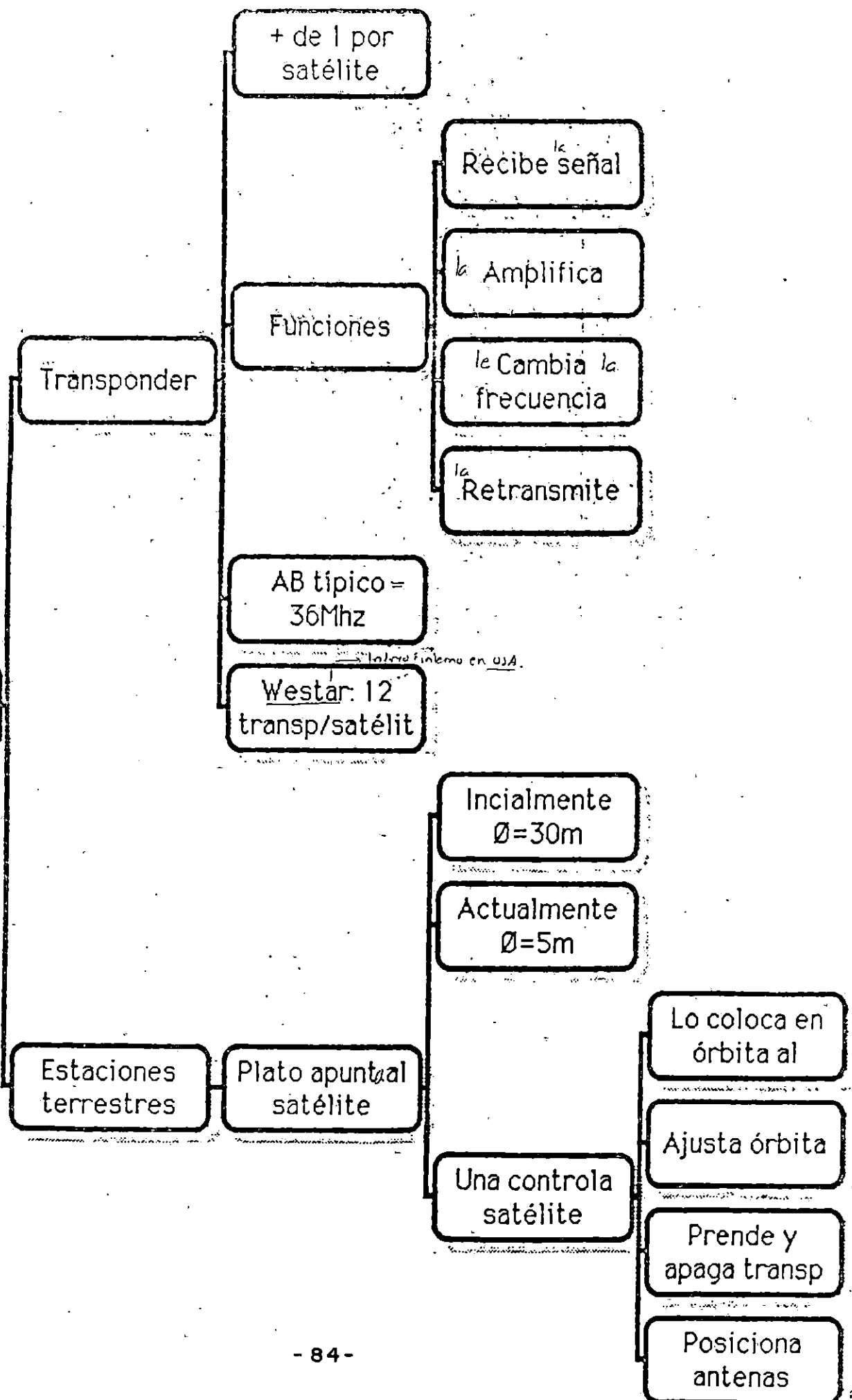
Geoestacionarios →

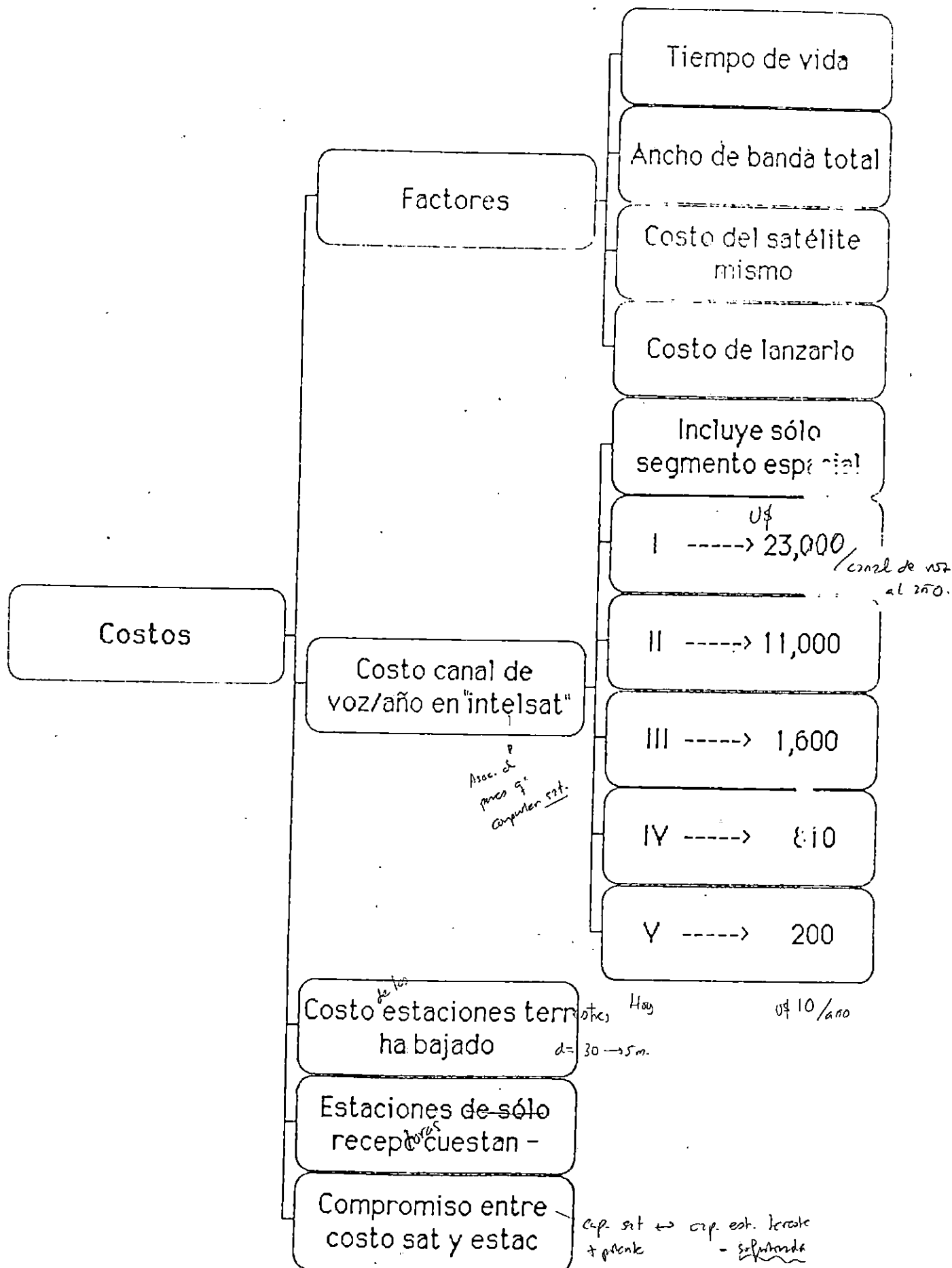


Lo p' lugar q' el satélite rote a la misma prop. q' la Tierra

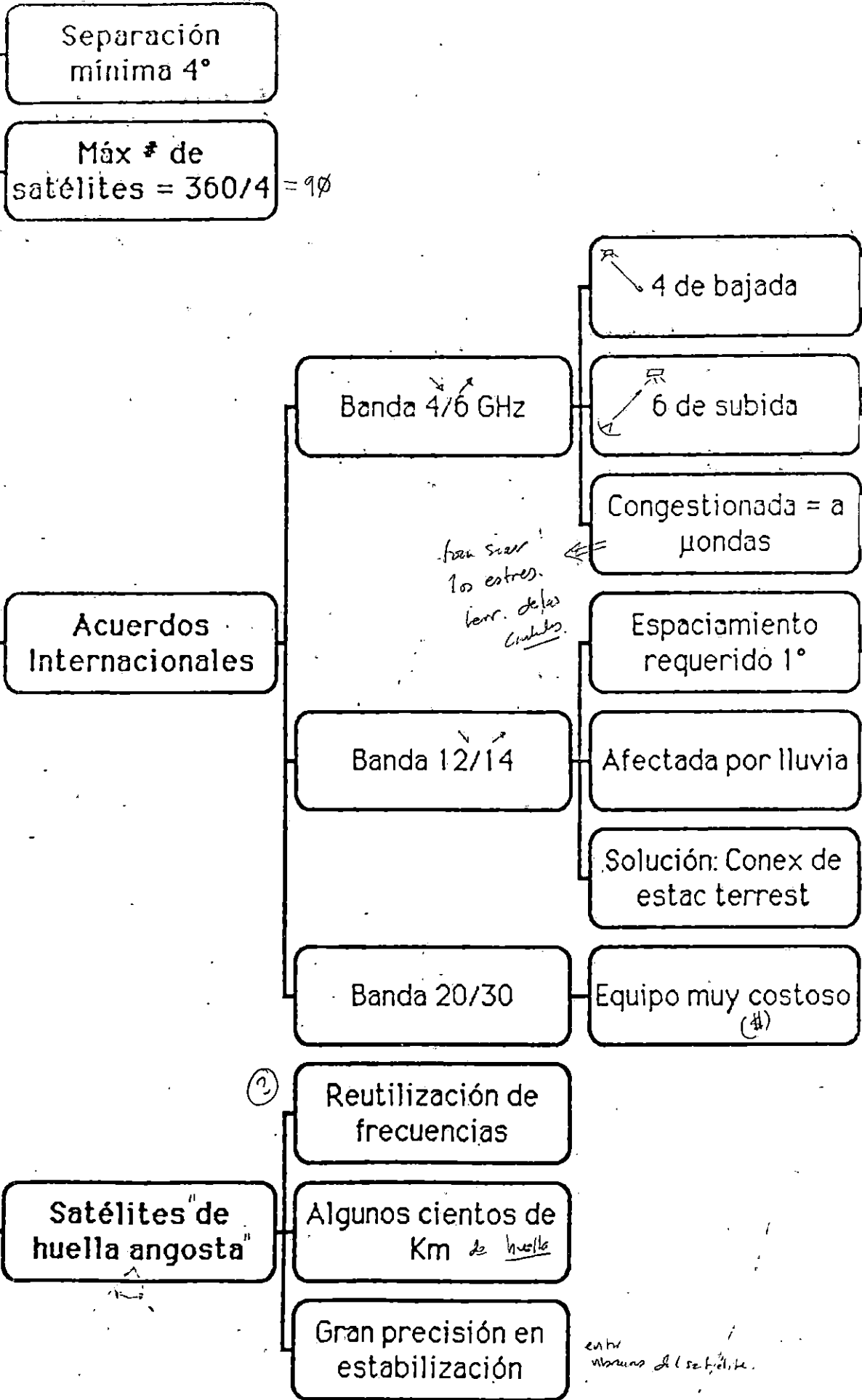
- A 36,000 km de altura (son Pijos)
- Deben estar sobre el ecuador (col → p'mitigado)
- Colocación en órbita ^{es} delicada, c
- 3 ^{preciden} cubren ^{se} toda la tierra ^{bién} y ^{convergen} con ^{hacia} el mdo.
- Ajustan órbita c/5 semanas con chorros de gas
- Usan baterías solares ^{para p' tener} (p'oble. dan ^{los eclipses})
- Manejan miles de canales de voz, de TV.

Componentes fundamentales

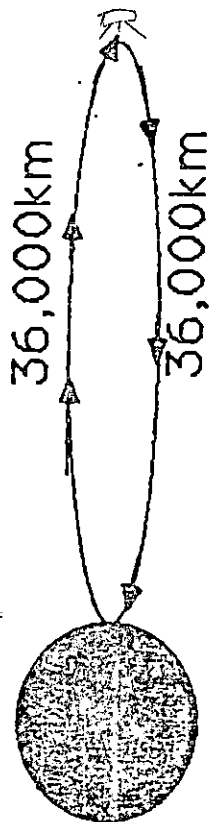




Repartición del espacio



• El Problema de la demora ($t_{propaga}$ es muy pequeña)



$$T_t = t_{propaga} + t_{comuln}$$

$$t_p = \frac{36,000 + 36,000 \text{ km}}{300,000 \text{ km/s}} = \underline{240 \text{ ms}}$$

$$t_c = \underline{30 \text{ ms}} \quad t_{\text{margen de latencia}}$$

$$T_t = 240 + 30 = \underline{270 \text{ ms}} \quad \cdot \frac{1 \text{ s}}{1000 \text{ ms}} = \frac{1}{4} \text{ seg}$$

Problema de la demora

- Exagerado por operadores terrestres
- Menos grave que TASI ^{interleaving} cuando halla → le da al cable
no → se le quitan
- Problema
- Solución: Satélite de ida-cable de venida
- Independiente de la distancia
- Problemas en polling (muy costoso
en un satélite).
- Nivel de enlace con pipelining

cable $\rightarrow$ < ms $\Rightarrow$ podría ser tr. menor
por el.

Exerc! $\rightarrow$ Problema

¿' es mejor: un cable o un satélite?

En g' menor, desde el pto de vista de la
long. del mensaje, es + ef. tr. por cable

A partir de qué longitud de mensaje (L) su
transmisión trasatlántica es más eficiente por
satélite? Suponga velocidades de trans de
9600bps para el cable y de 5Mbps para el
satélite. Longitud del cable = 8,000km. Velocidad
de prop en cable 150,000km/seg

$T = t_t + t_p$ *respon.* $t_t = t_p + t_{\text{trans.}}$

*hacer q' no hay entre los bits
q' le van a fluir de los bits mayor
de est. a est.*

Punto de corte $\rightarrow T(\text{satélite}) = T(\text{cable})$

$$t = d/v \Rightarrow$$

$$t(\text{satélite}) = L/5,000,000 + \overbrace{0.270}^{270 \text{ ms}}$$

$$t(\text{cable}) = L/9,600 + 8,000/150,000$$

$$L = 2084 \text{ bits} = 260 \text{ bytes}$$

A partir de esta L del pto de
tr. por satélite.

6) Cables submarinos

- Unico medio trasatlántico antes de satélites
- Primero tendido en 1858 para telégrafo
- En 1956 primero para voz
- Son cables coaxiales especiales
- Su capacidad ha aumentado
- ^{de 48 canales} 48 canales ^{multiplexado} de voz en 1956
- 140 en 1963
- 845 en 1969
- 4,000 en 1976
- Desde 1969 repetidores con transistores
- Se usa TASI ^{time sharing} para optimizar ^{se optimiza} su utilización

→ cuando se habla en se repite el canal } disminuye
→ se gana

7) Radio de alta frecuencia

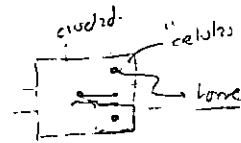
- Usada en comunicación trasatlántica ^{entre} < 1956 _{de los cables submarinos}
- Usada aún para telegrafía de barcos
- También para países sin cables subm ni sat
- No se usa en datos (poco usada)
com. de voz
- Se refleja en ionosfera (50 km ↑).
Inestable (clima, etc) → genera distorsiones
- Cambios en ionosfera generan distorsión

4) Circ^{uitos} de dispersión troposférica

- Tropósfera: + baja que ionósfera y + estable
- Tropósfera $\approx 10\text{km}$; Ionósfera $\approx 50\text{km}$
- Usada para comunicaciones a $d \approx 1,000\text{ km}$
- Existen en ^{Cuba} Alaska y entre USA y Nassau ^(Bahamas)
- Se usan para transmitir + allá del horizonte.
- # de canales depende de distancia ^{se multiplica}
- Para $d > 150\text{ km}$ es típico 72 ^{canals}
- Afectadas por condiciones atmosféricas ^{mas}
- Costo alto por canal
- Antenas con $\varnothing = 18 - 36\text{ m}$

1) Radio de corta distancia

- Es posible usar radios estilo taxi o talkie para DP ^{walkie-}
- Ventaja: Puede ser móvil (cmo)
- Problemas de ruido (interf)
- Investigar sobre CB ^{por} Radio celular



com. celular - (torre celular)
se encargan de recibir
a la red local

4') Guías de Onda

- usado p' llevar la señal de la antena de ondas al canal.

- su rango es 10

- Tubo metálico que transporta ondas de alta f
- Las hay rectangulares y circulares
- Se usan en d cortas < 1km
- Circulares son más poderosas
- No son flexibles ----> difíciles de instalar
- Capacidad hasta de 300,000 conversaciones

(PMIS)

11) Fibra óptica
 llevan/goran ondas electromagnéticas de altísima freq. (ondas de luz).

Generalidades

- ^{es} noticia visible de crecer en multipunto
- cuando no se reg. vel y des, se usa el d.c.

donde conv. en ser a el. a lum y lum. a el. a el.

A
 mientras > AB,
 más info. le puede
 no refer.

= Por q' el pero inorden
Guíasondas, de luz
y no em

se here un q'n n PdB
 => se puede hacer > AdB

Como otros cables:
Punto-a-punto

P.a.r.

≠ satelites no broadcast

Costos de producción masiva < coaxial

"la f.o." está costando con el c.c."

AB inmenso

Trabaja con portadoras en zona infrarroja

Long de onda ≈ 1 μm

Se han logrado AB > 1GHz

donde el lee la info
 por ser del tipo =>
mucha en 2 se la tubo

Por qué tanto escándalo?

Desventajas
temporales

Tipos de fibras ópticas

En la práctica

clave: λ onda, atenuación
 (excepto en ondas con λ = 800-1300 nm.)

Por qué tanto escándalo?

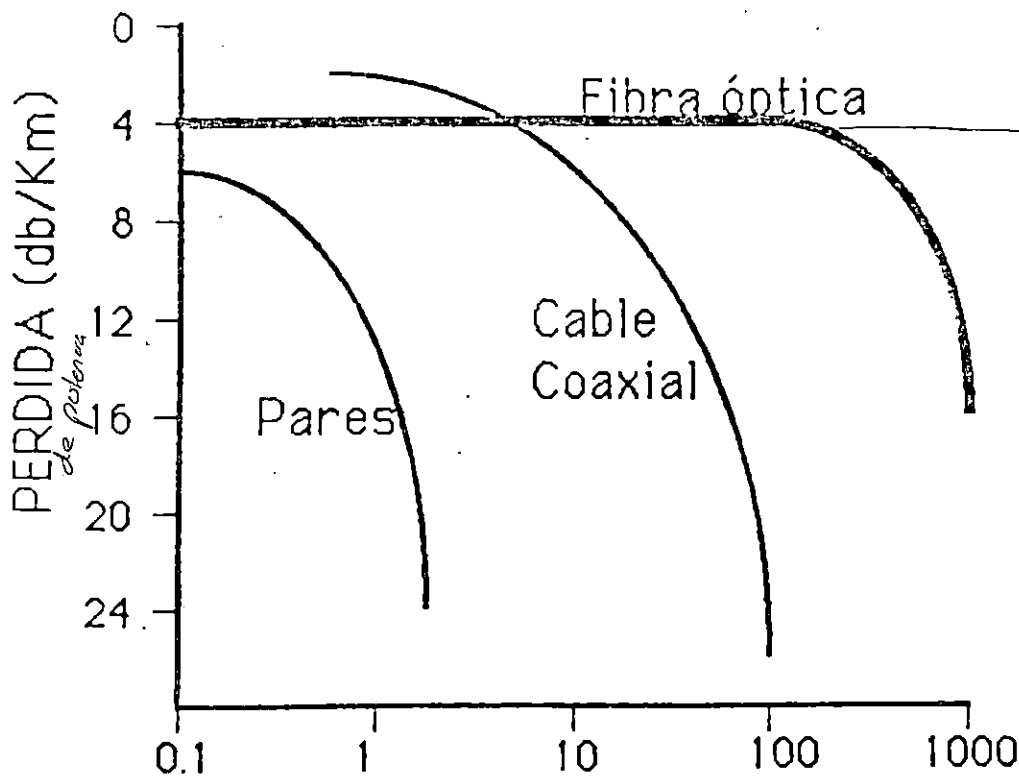
minifitcom F-0 $\Rightarrow$ $\downarrow$ cost

- Combinación ^{entre} costo, AB, facilidad de instalación
- Otras ventajas ^{tech.} (gde y potencializ. inmensa)
- Tamaño ^{delgado}
- Peso ^{liviano}
- ^{alta} Inmunidad al ruido (trabaja con luz y no con ondas EM's perturbables) $\rightarrow$ inestables en una red cable en una fibra
- Baja atenuación - no es afectado por el ruido de motores, etc
- Seguridad (dado q' maneja sen. lum. y no elect $\rightarrow$ no hay cortos circuitos, etc)
- Distorsión de retraso mínima

] conduct

Baja atenuación

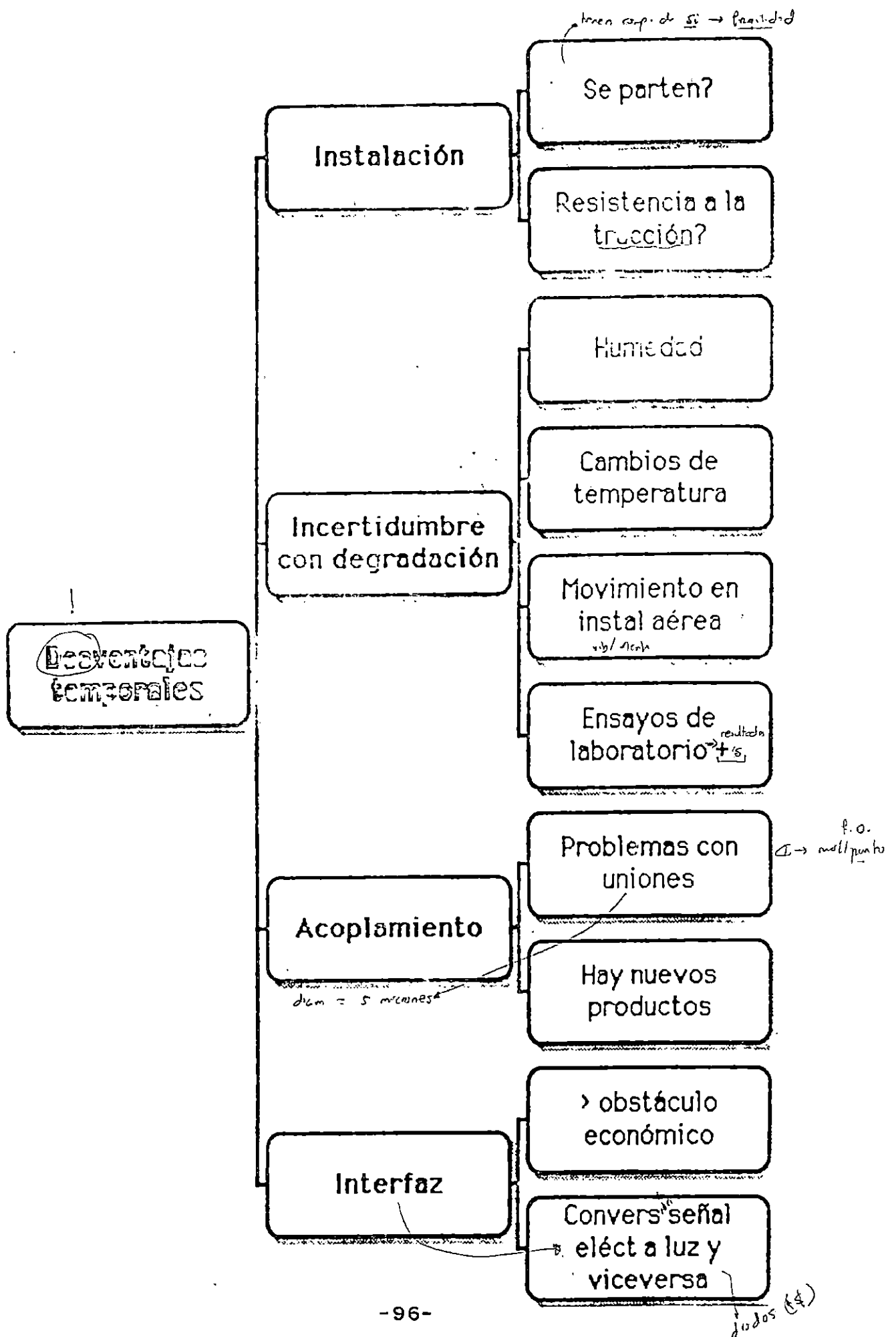
una fibra uspa + rápida q' otra



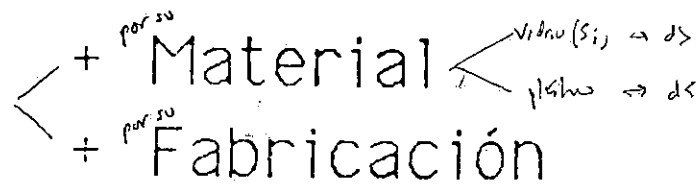
hey!

PASO 6
ANCHO DE BANDA EN MHz

que coincide pues se llama a partir de 6.



Tipos de fibras ópticas



Material

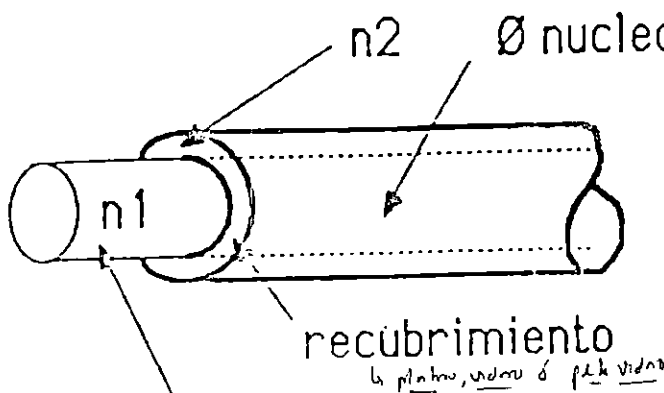
- Vidrio para d medias y largas ^{est. + 15.}
- Plástico para d cortas ^{plás.} (redes locales)

mayor atenuación (5)
más barato
menor calidad

Fabricación

- Step-Index $\rightarrow$ multi-modo
- Single mode $\rightarrow$ mono-modo
- + Graded Index - índice graduado
- + Special Profiles ^{fibras W}

Step-Index *multimodo*



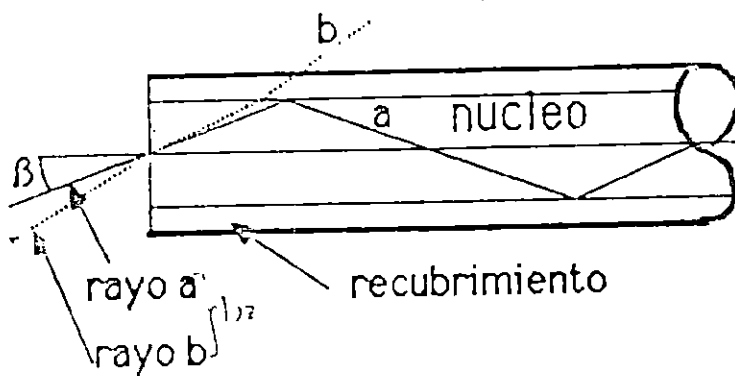
FIBRA STEP INDEX

$n1, n2$: Índices de Refracción

$n1 > n2$ *en* aprox 1%

núcleo : Vidrio a base de silicio puro

Si $n1 > n2$ ----> Guíaondas (inyección al núcleo, y se refleja en la periferia).



Si $\beta > \beta_{\text{Brewster}}$ rayo pasa a recubrimiento

Ley de Snell

$$AN = \sqrt{n1^2 - n2^2} = \text{sen } \beta$$

$AN = 0.2$ excelente

$\beta = 12^\circ$

CONCLUSION: HAGAMOS $n1/n2$ GRANDE PARA ACOPLAR MAS ENERGIA

FALSO! *eso no es la meta.*

Genera Dispersión

Modal

Fibras Multimodo

Mejor cuidemos la Atenuación

$$P_f = P_i * 10^{-ca \cdot l / 10}$$

ca = Coef de atenuación

(l es la longitud que se entra a la fibra y P es la potencia)

2) F. Single mode monomodo

Si en la ecuación:

$$N = \frac{2\pi}{\lambda} * a * NA \quad \text{donde}$$

λ = Long de Onda

NA = Apertura Numérica

a = Radio del núcleo

N = 1 tenemos una fibra monomodo  AB >

1 en f. multimodo

Energía acoplada  Area del núcleo

NA $\approx$ 0.1 para pequeña diferencia entre n1 y n2

Para que N pueda ser 1, a debe ser pequeño $\approx$ 5 μ m

 Se hace difícil acoplar energía 

Ancho de banda potencial: 100 GHz

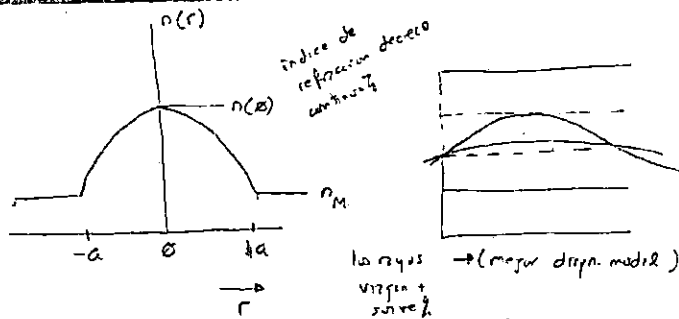
problema  más la 107.

Graded Index

índice de refracción

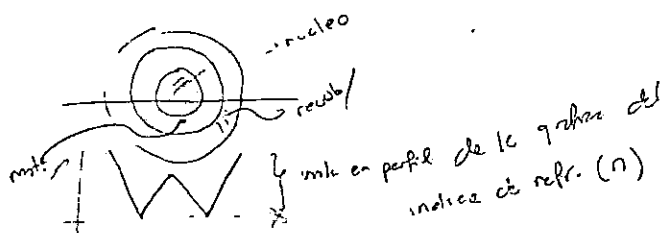
gradado

- n decrece en función del radio
- Luz viaja en forma ondulada; no en Zig-Zag
- Dispersión modal = 2 órdenes de magnitud
- $AB >$ $< en$
- La mayoría de las F. O de larga distancia son GI



Special Profiles

- LLamadas en Japón fibras W
- W se refiere a perfil de n
- Núcleo y recubrimiento separados por material $n <$
- Se han mostrado ejemplares a 1 GHz



En la práctica

- Fácil: 10Mbps en 10Km sin repet
- Algunas veces 32 Mbps en 53 Km sin repet
- + Bajo costo: Plastic Clad Silica (PCS)
- + Las más baratas

Bajo costo: Plastic Clad Silica (PCS)

- Distancias < 1Km
- V máx 10Mbps
- Atenuación: 10db/km
- Dificultades con conectores

Las más baratas

- Son + grandes
- Tipo step index
- Atenuación: 100-500db/Km
- NA grande: + energía compensa atenuac
- Totalmente plásticas
- Útiles para aprox 300m
- AB bajo
- Gran flexibilidad y resistencia

TELEFONIA DIGITAL

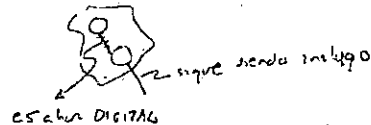
señal analógica → digital → res. repet. en circuitos
analógica → " amplific.

señal digital → exact. por la indep. de la Δ en la calidad de la señal

PCM (PULSE CODE MODULATION)

Sist. de comunicación. Técnicas de digitaliz. de señales analógicas.

- + Generalidades
- + Procesos Básicos
- + Calidad de la señal reconstruida
- + Bell T1 → 1^{er} est. "convencional" PCM.
(PCM 24)
- + Estándar de la CCITT diferente
- + Otro estándar (empres).
- + Utilidad en datos ^{de q. la señal telef. digitalizada.}



Usado por circuitos que pueden albergar una l. de ens. lineales

(9)
(0)

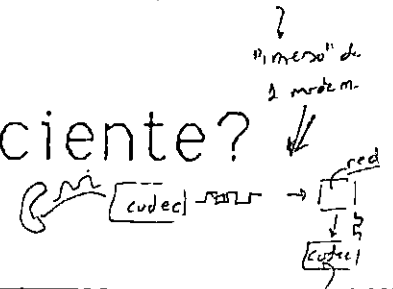
Generalidades

PCM → Sist. de

- Digitalización de voz *envío de bits por la l. telefónica*
- Desarrollada en 1937
- Comercialmente instalada en 1962 (semicond)
- *señal digital* Consume más AB *hecho gr. ruido + males*
- CCITT no logró estándar a tiempo
- Se generaron versiones incompatibles
- Cajas negras entre países

Procesos Básicos del swt rec

- ^{se $\leq 10^3$} CODEC: Codificador-Decodificador
- + PCM lineal
- + Puede hacerse más eficiente?
- + Otras Técnicas ^{de digitalización} de swt
 - discreción x amplitud



PCM lineal

→ para la "recup." total de la señal q se envía

- Digit. completa: #muestras = $2f_{\text{máx}}$ } Teorema del muestreo (Nyquist)
- Canal voz 4KHz ----> 8.000 muest/seg
- 1 muestra cada $125\mu\text{seg}$ (code)
- 1 muestra ^{se codifica con} ----> 8 bits
- $8.000 \text{ mst/seg} * 8 \text{ bits/mst} = 64.000 \text{ bps}$ ^{vel. en el canal digital de 1 canal de voz}

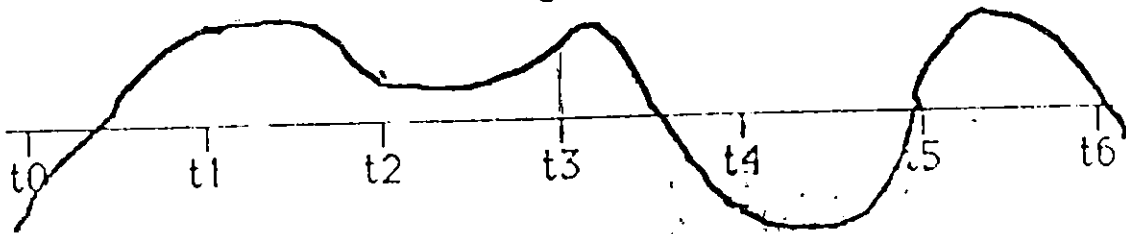
Puede hacerse más eficiente?

- Más muestras a menos niveles?
- Menos muestras a más niveles?
- Otros algoritmos para - $\text{muest} * \text{niv}$?
- Resp. depende del tipo de onda ^{voz / video etc.}

a digitalización

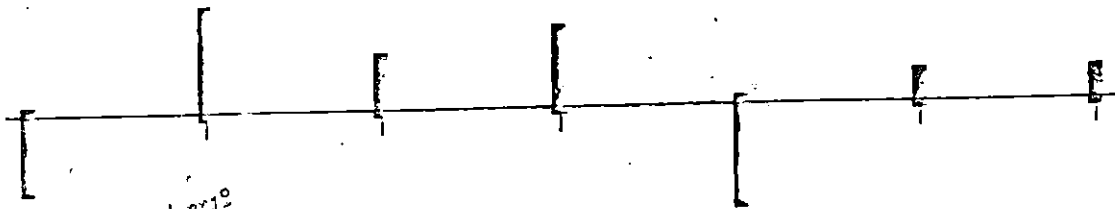
Procesos Básicos

Señal Análoga Original



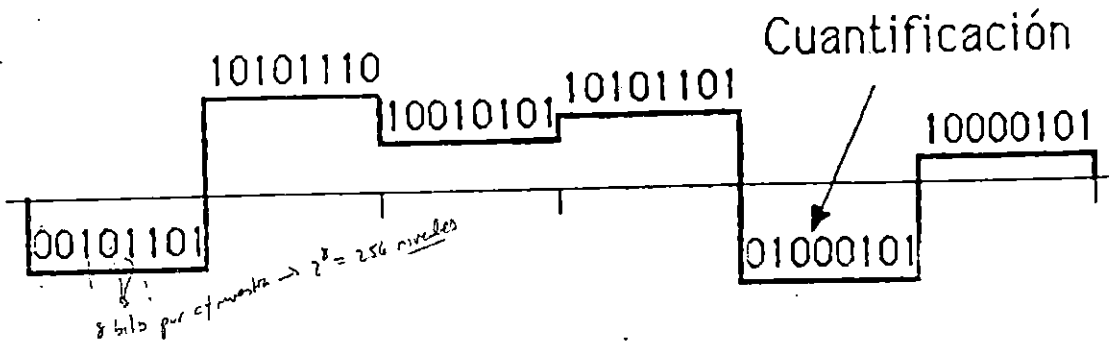
Se muestrea en instantes discretos; la dist. entre ser peg

① PAM *also called modulation*



MIC → PCM
2 bits
4 bits
8 bits

Representación de PAM



② digital

Señal Digital PCM

3) *env*
(bit a bit)



PROCESOS DE DIGITALIZACION PCM

1) PAM

2) digital
3) envío de 1 bit a la vez
-106-
3

Otras Técnicas pl optimizar PCM

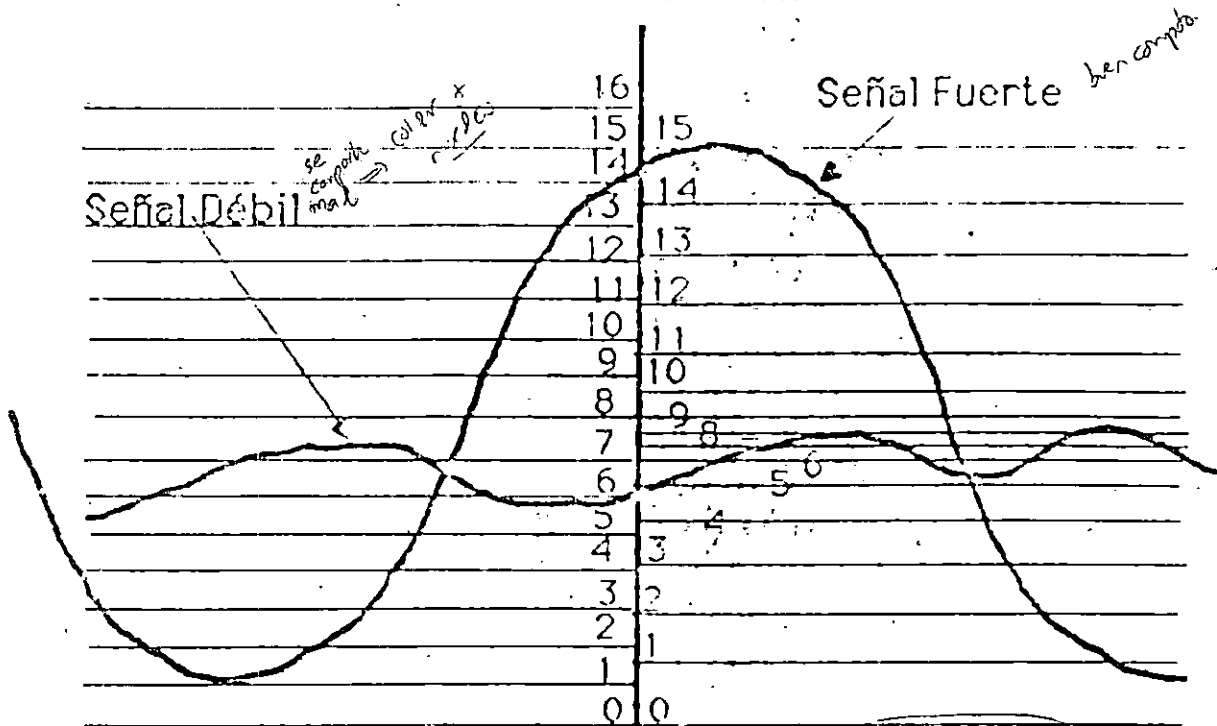
- * V de codificación Variable
- ⊕ ✓ Codificación sin ruido
- ⊕ ✓ Companding
- * Adaptable
- ⊕ ✓ PCM Diferencial
- ⊕ ✓ Delta Modulation
- ⊕ ✓ Codificación Predictiva

× V de codificación Variable

- Toma ventaja de gaps y redundancia
- Detección de silencio
- Detección de actividad (TASI)
- Transm sólo cuando haya cambios import
- Speech Predictive Encoding Communication (SPEC)
- # diferente de bits por muestra

COMPANDING

Niveles de
Cuantificación



Sin Compandor

coloca más niveles
p' darle mejor resolución y
amplificar más

mejora la calidad
de la digitalización
de la voz.

Con Compandor

4 el # niveles en la zona
en la q' se usa la señal
débil p' lograr una gran
+ línea de la señal débil

Codificación sin ruido

- Puede reducir V^e requerida en la línea
- No usa N bits para cada nivel de cuantific
- # de bits por muestra ^{es} Variable (no todos tienen 8 bits)
- Se arman grupos de bits
- 10, 000, 011, 110, 111, 0101, 00100, 00101
- Un grupo no debe ser = a la 1a parte de otro (ninguna es subcadena ppal de otra)
- La cadena 11001101011000100
- Puede leerse solamente 110 011 0101 10 00100
- Buffering necesario para salida kte
- Problemas de overflow y underflow de buffers
- Niveles que + ocurren se codifican con grupos ...

si
fueran
p' comprimir
inf

si hay un
q' se repite
muchas
→ se lo da al
grupo +
pequeño

si hay
un q' se
repite
muchas
→ se lo da al
grupo +
pequeño

8

Adaptable

- Niveles de cuantificación no son fijos
- Puede variarlos según comport de señal
- También puede cambiar de cuantificador
- Hay varios con diferentes tamaños de interv
- Se pueden usar técnicas silábicas
- Cambio de interv considerado sólo después de t
- Para voz $t \approx 5-10\text{ms}$
- El objetivo es reducir error sin aumentar AB
- Cuantifasí se llaman APCM (Adaptative PCM)
- + Pueden usar esquemas hacia adelante o atrás

X

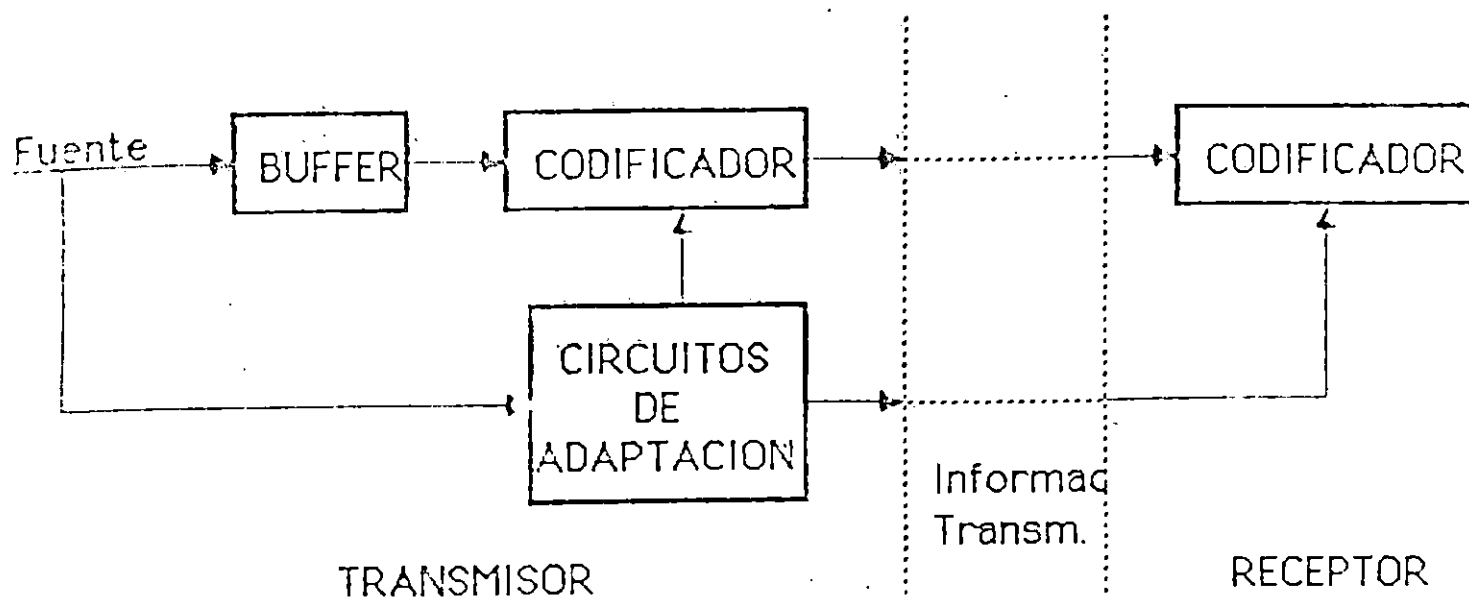
Adelante (Forward)

- Se basa en muestras de la señal fuente
- Muestras en buffers
- Se analizan para determinar interv
- Se debe mandar inform de interv a recept
- Ejemplo: NIC (Nearly Instantaneous Companding)
- Aumenta requerimiento de AB
- Atractivo para conversión de tipos de PCM

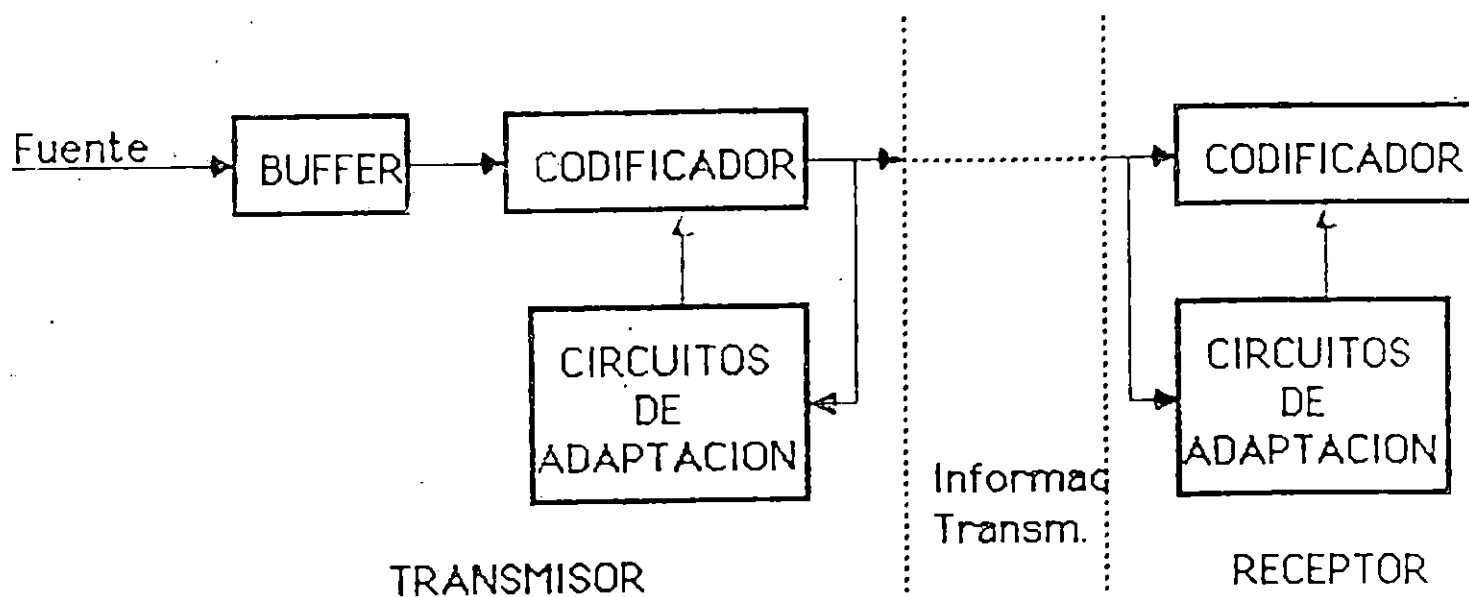
+ Atras (Backward)

- Basado en muestras históricas
- Receptor hace el mismo análisis
- Más simple, pero genera + errores

PCM ADAPTABLE



ESTIMACION HACIA ADELANTE



ESTIMACION HACIA ATRAS

PCM Diferencial

- Cambios no fuertes entre muestras consecutivas

De una muestra a otra los cambios son pqs (pues las series de voz son continuas)

① Redundancia $\rightarrow$ si la muestra n es igual a la $n-1$, solo se transmite un bit (0 o 1) no hay res. de la info codificada en la info anterior

- Supone que no cambia más de ± 16 niveles

$\rightarrow$ 5 bits por muestra

$range = 32 \text{ niveles}$

Delta Modulation

$\rightarrow$ supone $\Delta = \pm 1$ nivel en la señal

- Parecida a la anterior
- Más atrevida
- Usa un sólo bit $\rightarrow$ sube / baja un pto!

Codificación Predictiva

guarda 5/10 muestras anteriores y

- Extrapola con valores anteriores
- Codifica diferencia = medido - predicho

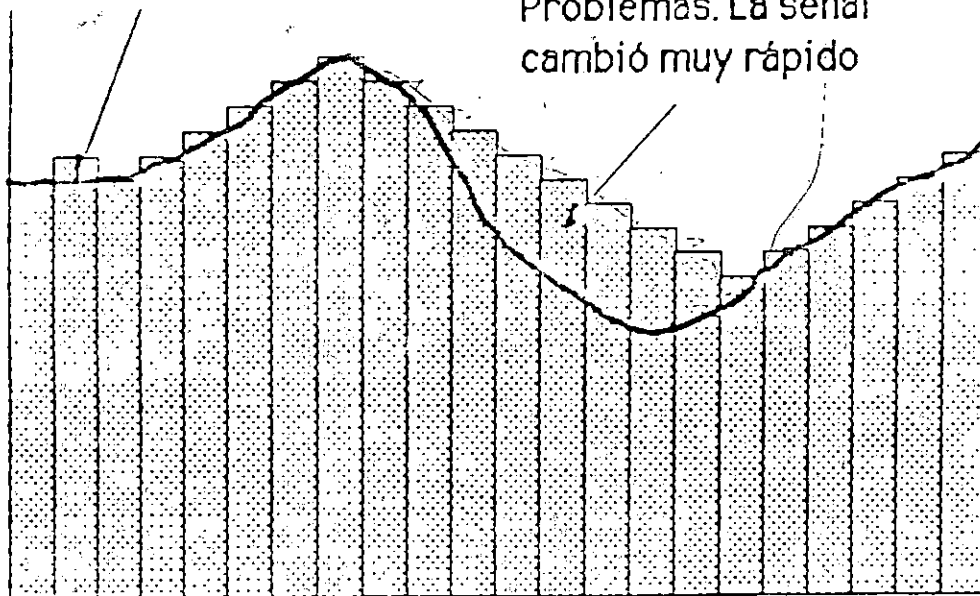
hay que dar a que cambio por línea

Delta Modulation

→ no se necesita de receptor
si hubo o no Δ en
la señal!

Muestras consecutivas
siempre difieren en ± 1

Problemas. La señal
cambió muy rápido



1 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1

Cadena
de bits
enviados

Intervalo de
muestreo

TIEMPO

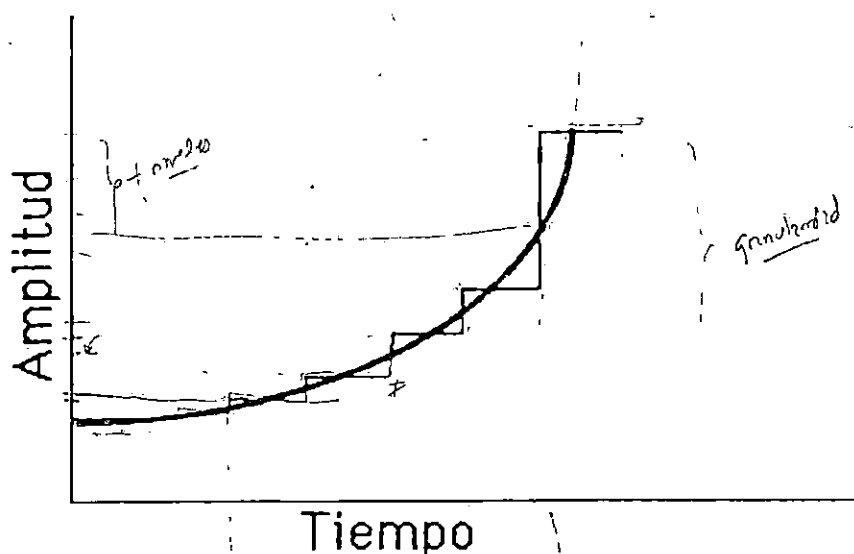
Calidad de la señal reconstruida

o recuperada.

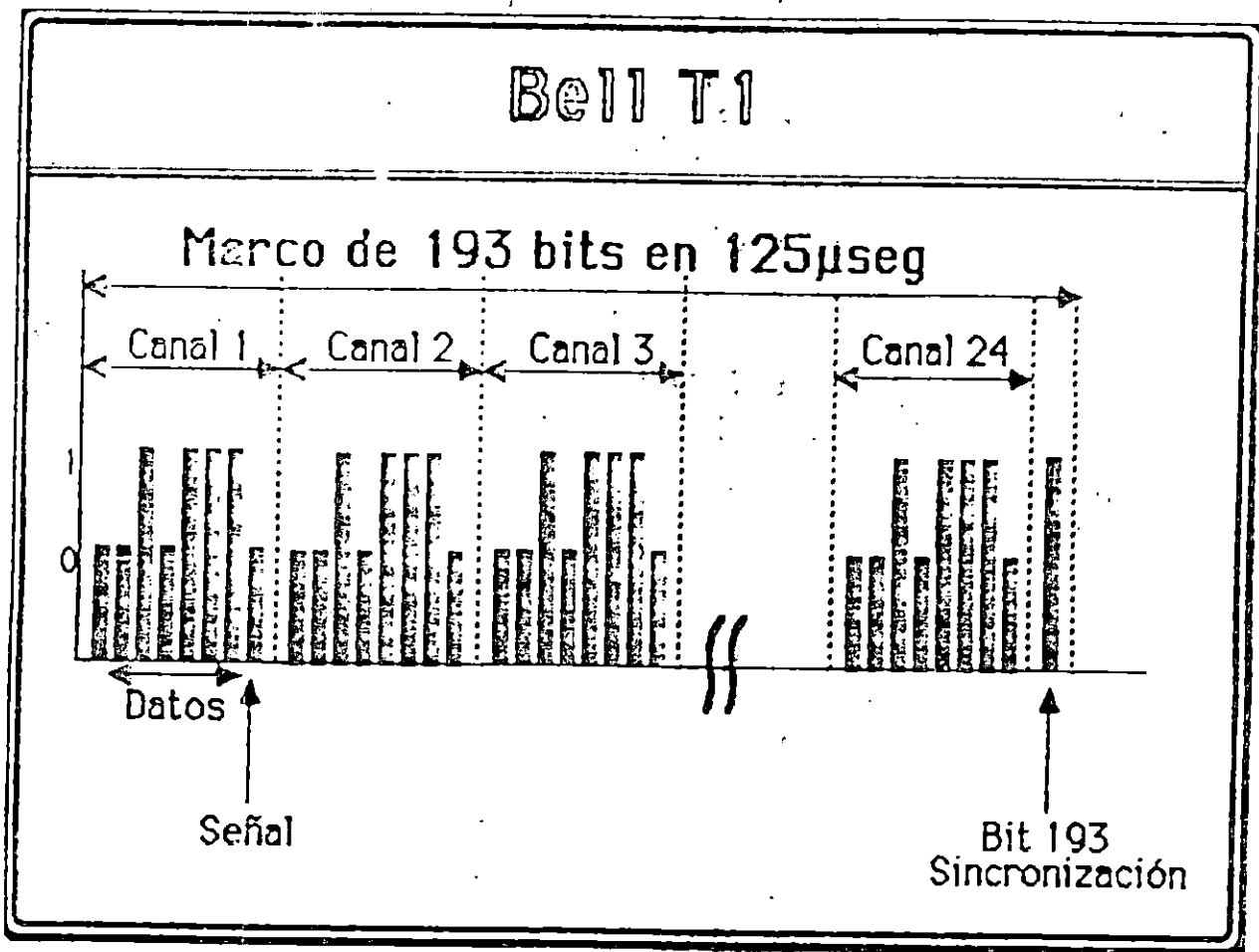
- A pesar de Nyquist discretización ----> errores
- Estos errores son llamados ruido cuántico *en la discretiz. del proceso de digitaliz.*
- Determinado por V de muestreo y # de niveles *de la s.c.n.*
- Ruido cuántico tiene tres componentes
 - ① Granular: Relacionado con # de niveles *aprox. a cada nivel de la discret.*
 - ② Desuso (Idle Noise): Aproximación a 0 con ruido
 - Sobrecarga: Nivel del rango dinámico excedido
 - Rango dinámico = Rango de ampl esperadas

③ presencia de ruido
conlleva a aprox. a cada
nivel q' está fuera del rango!
(grito inesperado)

RUIDO CUANTICO

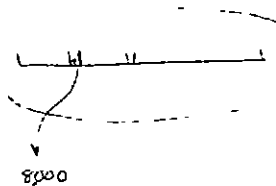


Bell T.1



de los 1.5' se usan 192 k p' sincron.

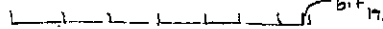
T1:



8000/seg ó 1 c/125 μ seg.

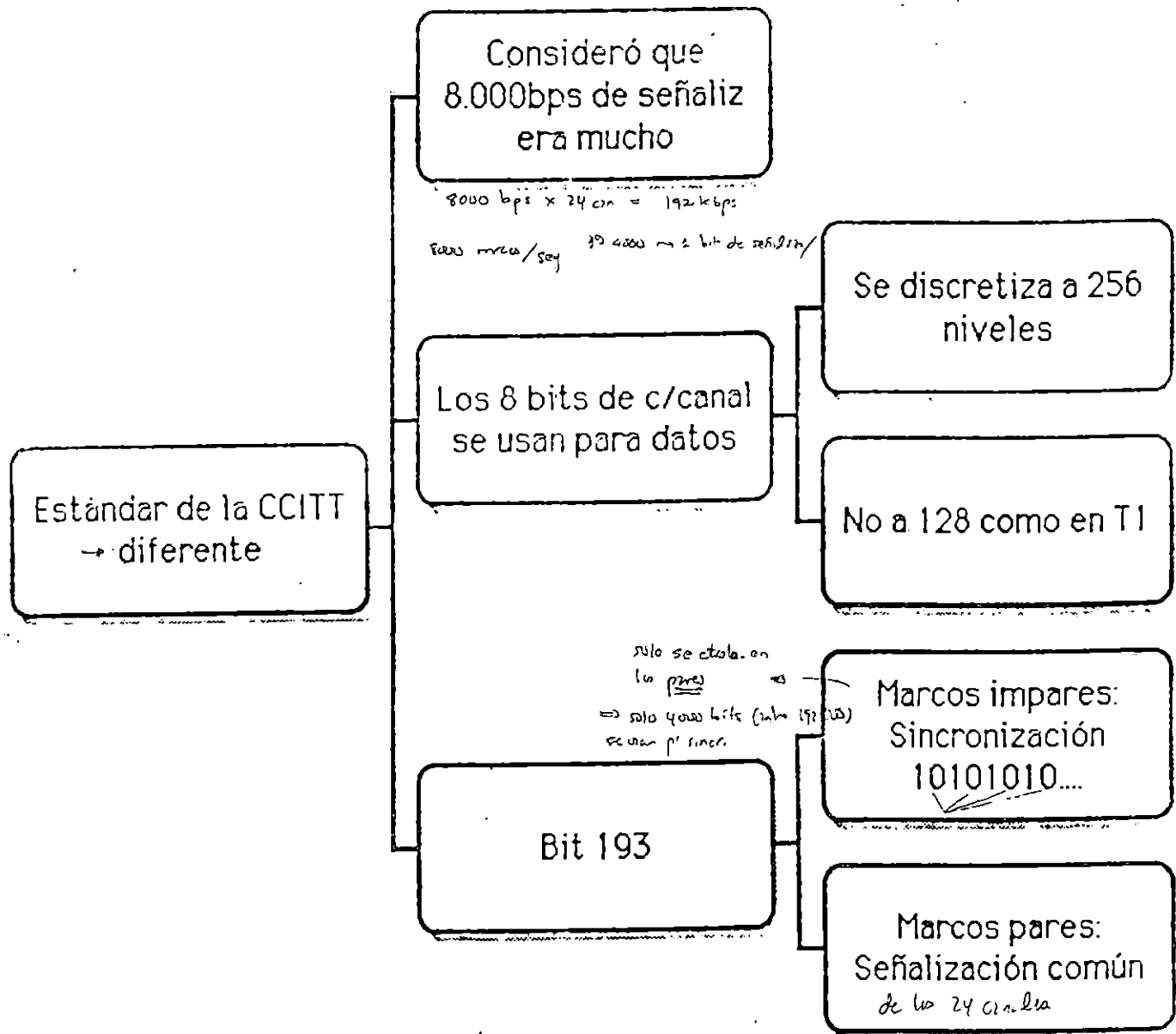
de los 1.5' solo se usan 4000 bits

CCITT:



bit 193 $\rightarrow$ prod. 8000 v/seg.

0, 1, 0, 1, $\rightarrow$



Otro estándar : PCM/30

- Usado en mayoría de países $\neq$ USA y Japón y Corea
- 32 canales de 8 bits c/u
- 30 de datos
- 2 de señalización (cód. de 12 bits)
- V requerida: 2'048.000 bps
- Se denomina PCM30

Anti-CCITT
(1-67. PRIMA)

Se necesita 8000 V/sec (Nyquist)

$32 \times 8 \times 8000$

Utilidad en datos

- Probl ^{es el} en local loop ^(propiedad)
- $\rightarrow V_{el} = 1.544$ y 2.048 Mbps ^{cargado con 12 bits}
- Podrían lograrse ^{los} vel.
- Descargándolos L.L.
- Repetidores c/2Km
- Se alquilan canales PCM ^{ja}

no puede hr. c. de $\rightarrow$

24 líneas
30

C. de 64 kbps
USA 56000 bps

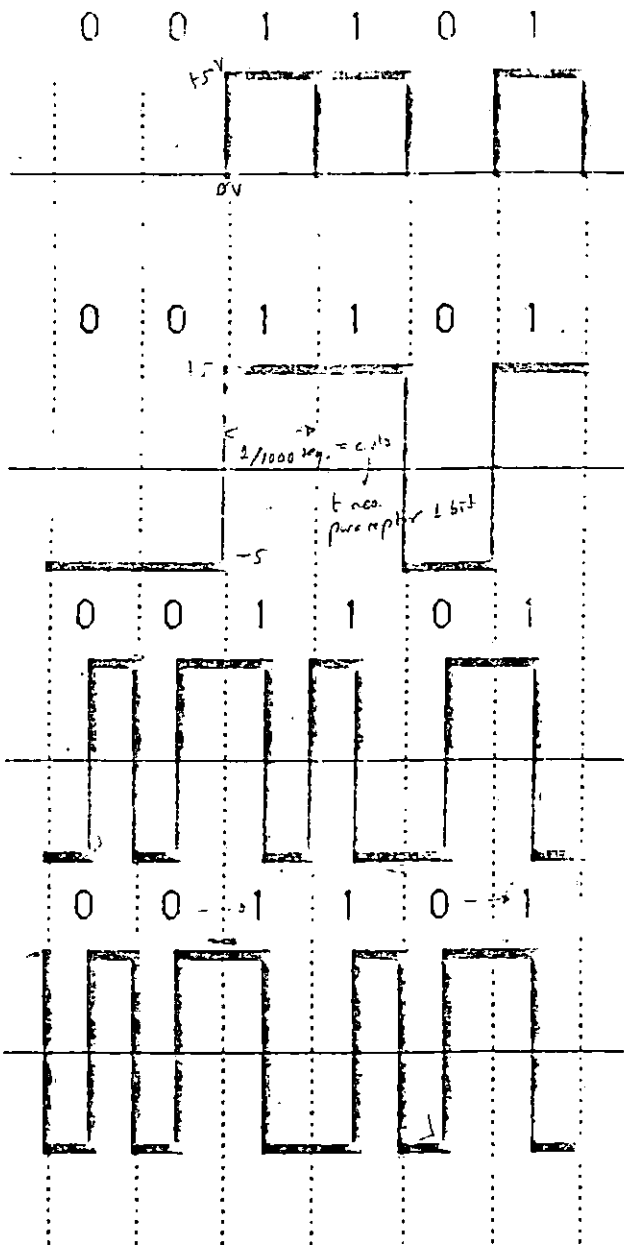
$2864 + 16 \rightarrow 144$ par/ bñtñ en el llugo

ISBN $\rightarrow$ serial/ $2B + D$ (3 caracteres)
 de banda
 cruzada
 3ª con serial de ISBN
 bien al probl. de los llugos.

Grule Home 2881188 Extra 318

REPRESENTACION DE INFORMACION DIGITAL

Como repñe un 0 y un 1 en una línea.



1) RETURN-TO-ZERO

Difícil reconocer principio y fin de cada bit

1 = Voltaje +

0 = Ausencia de voltaje

problem $\rightarrow$ meñe qñ 1's segñda $\Rightarrow$ 1's meñe meñe (dñfice)
 Debe 3r algo pñ reconer el cam. de c/bit.

2) NON-RETURN-TO-ZERO

Mismo Problema

1 = Voltaje +

0 = Voltaje -

RS-232C
 $-3V \rightarrow -25 \rightarrow 1$
 $+3V \rightarrow +25 \rightarrow 0$
 que $\rightarrow$ pin de reloj

3) MANCHESTER

Muy usado en LANs (redes locales).

gñ Incluye reloj en los datos ($\neq$ de los datos)

1 = Cambio $\uparrow$

0 = Cambio $\downarrow$

$\Rightarrow$ Hay pñ lo meñe 2 cambios en
 c/ciclo
 $\Rightarrow$ se sabe donde es la 1/2 de c/bit.

4) MANCHESTER DIFERENCIAL

0 = Cambio al principio del bit

1 = Ausencia de cambio al principio

incluye reloj también

Todo lo qñ se
 mñe visto es a
 nivel de funciñ
 y no de extr-
 s-extr. (ISBN)

The diagram illustrates three serial data frames. Each frame consists of a Start bit, a data field, and Stop bits. The first frame's data field contains the character 'A'. Handwritten annotations include:

- Start bit:** Points to the first bit of each frame.
- Stop bits:** Points to the final bits of each frame, with a note $\frac{1}{2}, \frac{1}{2} \text{ or } 2$.
- bit de parada:** A note pointing to the stop bits.
- 3 V por ms** and $\frac{2}{3} \text{ V de } \frac{1}{2} \text{ ms seg.}$: Notes indicating voltage levels and timing.
- 1 p. $\rightarrow$ 10 / ms.**: A note indicating a rate of 10 bits per millisecond.
- 5, 6, 7 u 8 bits por carácter:** A note indicating the data field length for the first frame.
- die cilo de se tr. bit de parada:** A note indicating the transmission cycle for the stop bit.

TRANSMISSION ASYNCRONICA

The diagram illustrates the structure of a packet. It is represented as a horizontal bar divided into three main sections. The leftmost section is labeled 'Encabezado' (Header) and contains fields for 'G' (likely source address) and 'C' (likely destination address). The middle section is labeled 'INFORMACION' (Information) and contains the payload data. The rightmost section is labeled 'Cola' (Trailer) and contains a field for 'E' (likely error control or sequence number). To the right of the packet structure, there is a box labeled 'Correspond' with a question mark and a box labeled 'SYN'.

TRANSMISION SINCRONICA

(ex: 101111)
 Bit stuffing
 0 1 1 1 1 1
 5 seconds → bit stuffing
 receiver here to know
 the byte language
 hay asociación a nivel de frame → no es pedet. el es en el frame y otro
 protocolo sinc $\left\{ \begin{array}{l} \text{or a cor} \rightarrow \text{un frame con encendido de cor} \\ \text{or a bits } \rightarrow \text{pre-charge de bits} \end{array} \right.$ $\rightarrow$ $\text{Eg: SBC (10M)} \rightarrow$ "cor de 10M"
 "bit stuffing"
 de $\text{EBC} \rightarrow \text{cor}$
 -120-
 1 cor de cor/frame de frame.

TIPUS DE TR

TRANSMISION ASINCRONICA

Sincr → + eficiente

asincr

11 par. br. 7 bits

- Receptor debe conocer principio y duración de bits
- La más simple *pero es la + lenta*
- Cada caracter se transmite independientemente
- Tiene un bit de comienzo = 0
- Tiene 1, 1.5 ó 2 bits de parada = 1
- En ausencia de inform. se envían bits de parada
- Inicio de caracter reconocido por transición ^{bit} 1-0
- No requiere gran precisión por resinc en *caracter*
- Barata pero ineficiente: 30% de sobrecarga *↑ impo de car*
- Transmisor emite caracteres a ratas no uniformes

(sincr en bit)

TRANSMISION SINCRONICA

- Abloca bits o caracteres
- Llegada de c/bit dentro de un bloque es predecible
- Relojes de transmisor y receptor sincronizados
- Línea de reloj entre ellos
- Codificación Manchester
- Recept debe reconocer comienzo y final de bloques
- Se le coloca a c/bloque Cabeza y Cola
- Un caracter para protocolos orientados a caracter
- Ejemplo: Bysinc de IBM (SYNC)
- Un patrón de bits para protocolos orientados a bits
- Ejemplo: SDLC 01111110 en cabeza y cola
- Qué pasa si cabeza o cola ocurren en datos? tres procesos
- Character stuffing
- Bit stuffing

→ Sincronía

TRANSMISION SERIAL

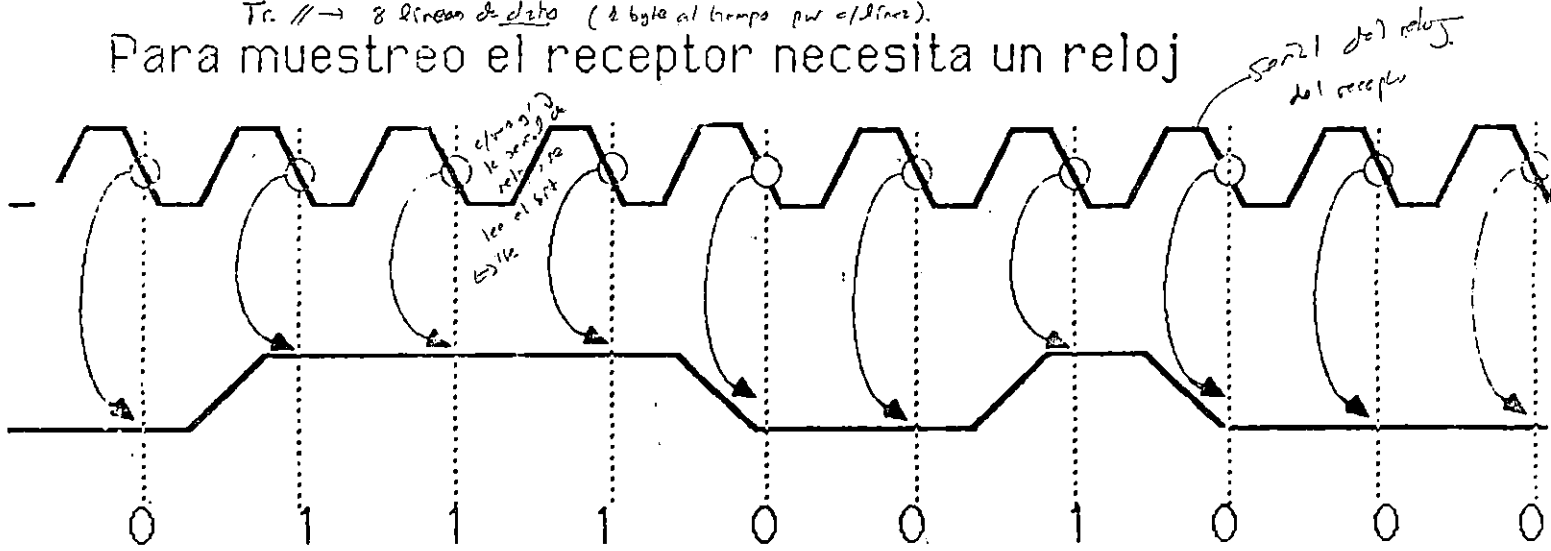
- Líneas telefónicas son seriales
- Salida del procesador en paralelo
- Se necesita conversión
- La hace chip de la interfaz serial

pasador del env
p → S (serie)
Y S → p (ent)
muestreo ← interfaz serial → comp

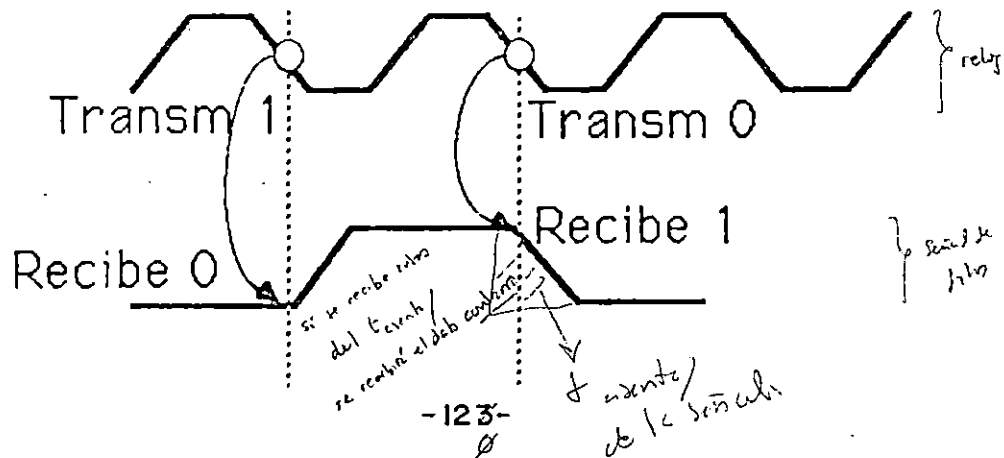
• RS-232 → por un solo cable mandan los bits uno a la vez.
pin 2 → tr. (1 cable).

Tr. // → 8 líneas de dato (1 byte al tiempo por 8 líneas).

Para muestreo el receptor necesita un reloj

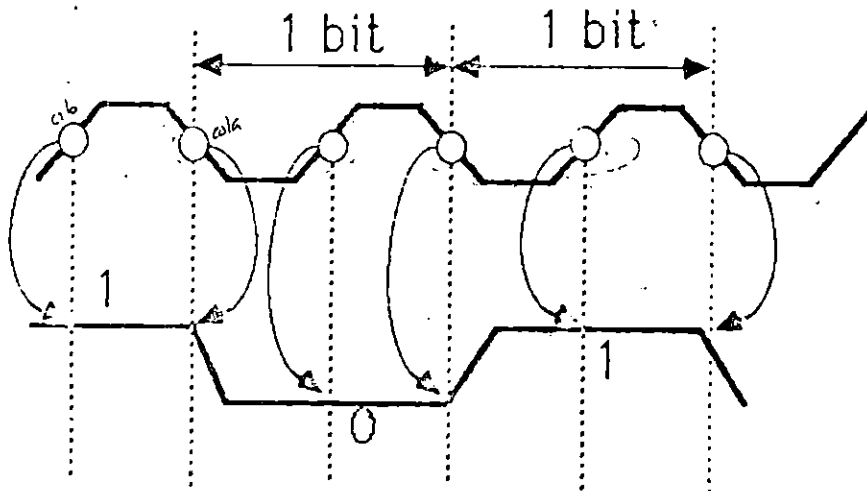


Relojes de transmisión y recepción no deben ser iguales

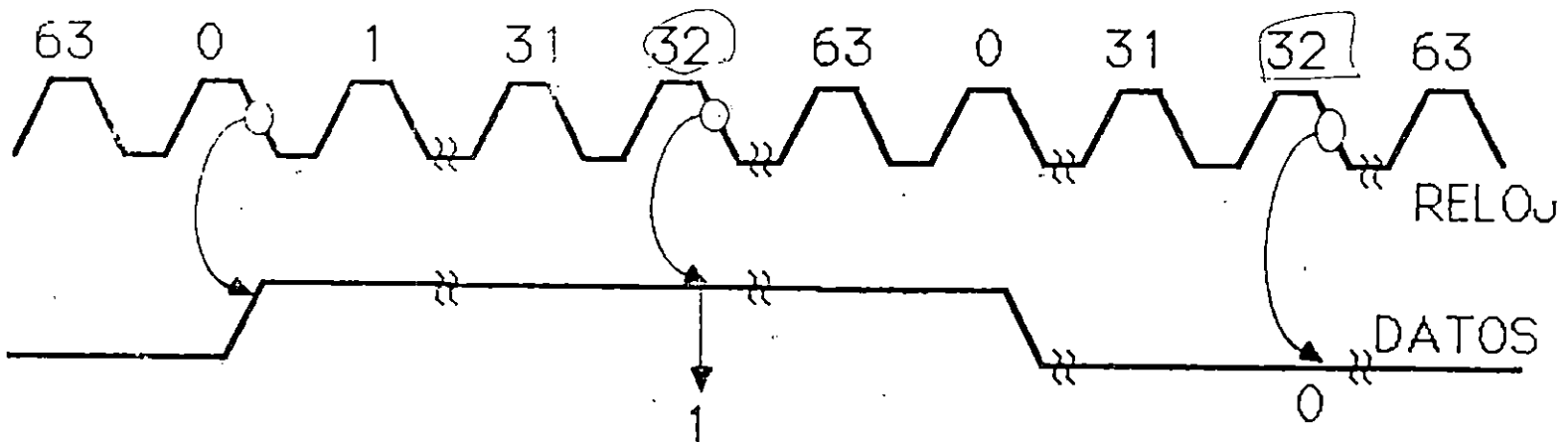


Para usar la misma señal de reloj se puede:

- Transmitir con trailing edge
- Recibir con leading edge



Reloj puede tener frecuencia mayor que la señal:



Se lee con mejor precisión

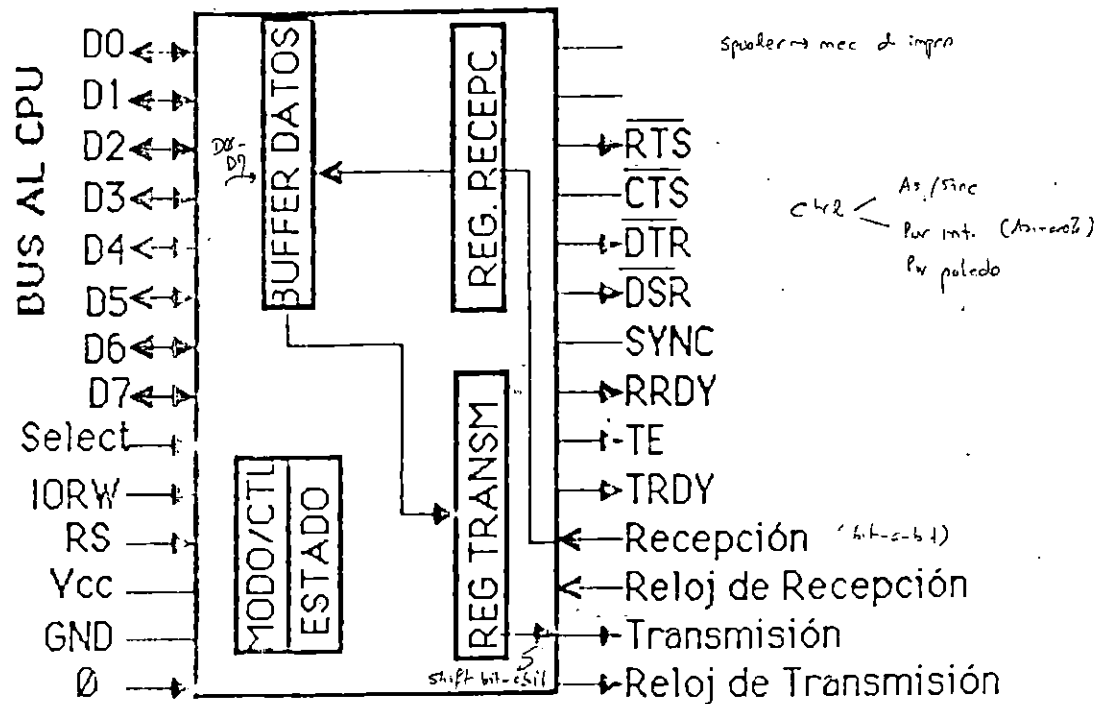
MANEJO DE ERRORES

- Errores vienen en tandas no repartidos
- Ventajas
 - Desventajas
 - Suponga transm en bloques de 1000 bits
 - Tasa de error = 0.001/bit no es uniforme repartida
 $\frac{1}{1000} \text{ de } 1/1000 \text{ bits}$ errores vienen en tandas o bloques
"50, 100 bits al 6"
 - Si errores son independientes ent
 - La mayoría de los bloques tienen error
 - Si vienen en tandas de 100 ent
 - Sólo uno o dos de cada 100 se dañan
 - Desventajas
 - Más difícil de detectar los errores en bloques
- Detección
 - Bits de paridad
 - CRCs ó checksums

$$\text{ruido de impulso} = \text{dura } \frac{1}{100} \text{ seg.}$$

$$\begin{aligned} &\text{si suponemos qz en 1 env de tr. se dañan } \frac{1}{1000} \text{ bit} \\ &\Rightarrow \text{se transmite paridad total en el } \frac{1}{1000} \text{ bit} \\ &\text{cierto} \quad (\approx 1000) \end{aligned}$$

DISPOSITIVOS TÍPICOS



Chip de interfaz bus a RS-232C en conv. el //fmo. del procesador a la 2.ª serial.

TE: Transmit Data Register Empty

TRDY: Transmit Ready

Buffer de transmisión Vacío

Genera Interrupción

RRDY: Receive Ready

Lógica de recepción pasó byte en buffer de datos

Genera Interrupción

SYNC: Verdadero si se detecta carácter SYNC

Puede Reemplazar al carácter SYNC

SEÑALES DE CONTROL DEL MODEM

DSR: Data Set Ready. Modem listo

DTR: Data Terminal Ready.

Dispositivo avisa al modem que puede recibir

RTS: Request to send. Se usa cuando DTR y DSR ya están bajos

Transmisor quiere empezar a transmitir

CTS: Modem la pone en bajo cuando DSR, DTR y RTS están bajos

para indicarle al transmisor que puede empezar

DISPOSITIVOS TIPICOS

- Algoritmo de transmisión
- + Registro de Control
- + Registro de Estado
- + Direccionamiento

Registro de Control

- Write Only
- Modo de transmisión Sinc o Asinc
- # de bits/caracter
- Paridad (Par, Impar, Mark, 0, Ninguna)
- Bits de Parada (1, 1.5, 2)
- Caracteres SYNC (1 ó 2)

Registro de Estado

- Read Only
- Error de paridad
- Error de framing. Start o stop bits malos
- Error de Overrun: CPU no sacó caracter a...
(reception)
- Underrun: CPU no escribió caracter a tiempo
(transmission)

Algoritmo de transmisión

INICIE

modem → term.

Baje DTR

REPITA

Pruebe DSR

HQ bajo

Prepare Character

Baje RTS

REPITA

Pruebe CTS

HQ bajo

Transmita

PARE

"hand-
shake"
term-modem

term (comp.)

modem

Perly 7

Direccionamiento

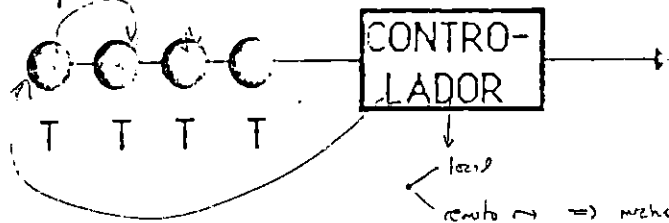
- Buffer de datos
- Registro de Control
- Registro de Estado
- No se direccionan registros de transm y recepc
- Registros de control y estado usan la misma @
- Pin RS selecciona uno de los dos registros
- Se direccionan por puertos o memory mapping

EIA RS232-C

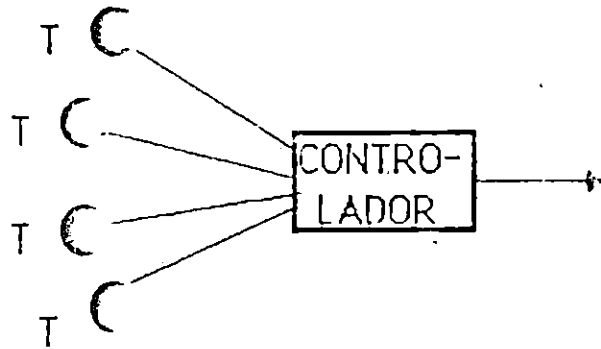
- EIA = Electronic Industries Association
- + Limitaciones
 - $V = 20\text{Kbps}$ 19200 bps
 - Distancia = 15mts.
- + Voltajes
 - + Alto $> 3.0\text{V}$ $< 25\text{V}$
 - Datos: Spacing, 0
 - Control: ON
 - + Bajo $< -3.0\text{V}$ entre -3 y -25
 - Datos: Marking, 1
 - Control: OFF
 - Circuitos Resisten hasta $\pm 25\text{V}$
- + Señales adicionales a las vistas
 - Indicador de timbre (p' control)
 - Detector de la calidad de la señal
 - Detección de señal en la línea
 - Conector no hace parte del estándar

MANEJO DE TERMINALES

Multipunto



Punto-a-Punto



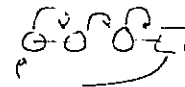
$\Rightarrow$ mayor de eco
 $\Rightarrow$ t de inv. muy largo
 la d. se conv. en half duplex

Polling

- Usado típicamente en multipunto

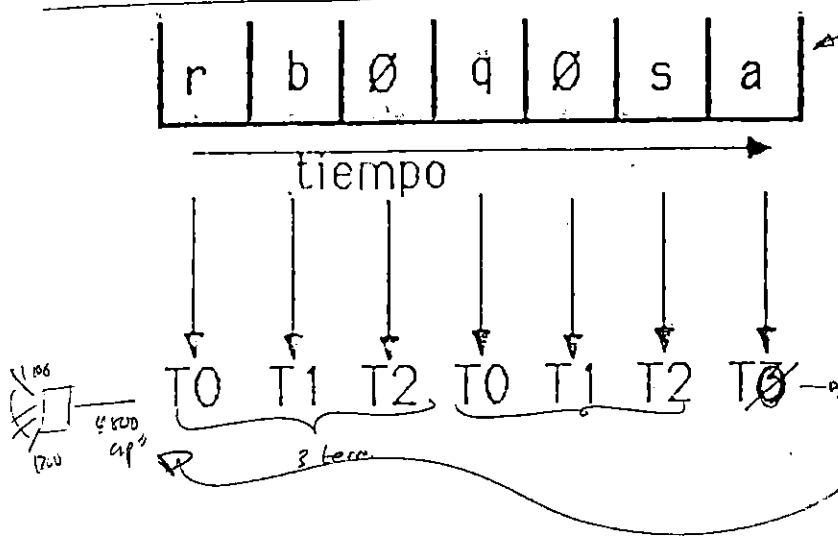
(Roll Call Polling $\rightarrow$? rto a el t. si quiere responder
 $\frac{1}{2} t$

(Hub Polling $\rightarrow$ solo inv. la linea 1 vez/ciclo
 donde la señal de pñeado es la últ.
 y es a la vez de la línea



~~En punto-a-punto asigna prioridades~~

Multiplexación vs. Concentración



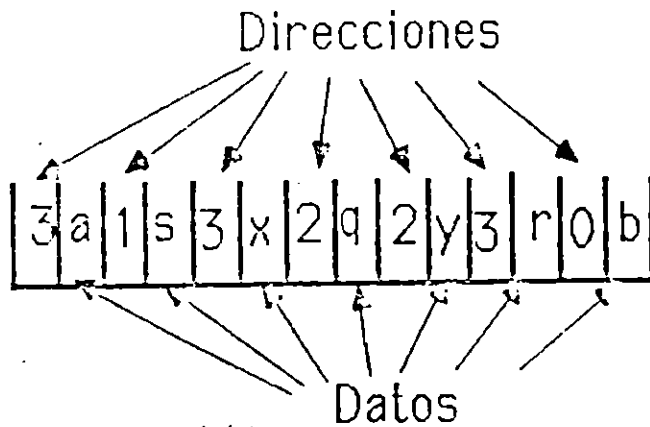
Synchronous Time Division Multiplexer (STDM)

Multi-sincr.

eficaz. pero los slots asignados de p. del b.

Capacidad de línea de salida = suma de las de entrada

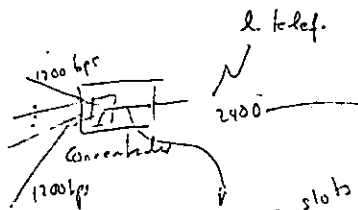
→ desperdicio de slots en la l. cuando hay algún q. no q. de uso real



Statistical Multiplexing

Línea de salida puede tener menos capacidad que la suma de las de entrada

Requiere buffer



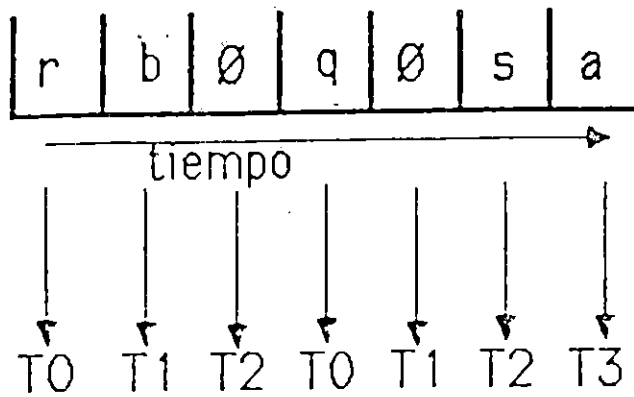
varios slots "vacíos" pero eficaz al no desperdiciar de q. terminal es + inf. de chil !
 sup. estadísticas y los terminales.
 solo cuando algo < la mitad en prop.

Factor de C = reln. entre la cap. de entrada y la cap. de salida de la línea.

$$\frac{C_{sal}}{C_{ent.}}$$

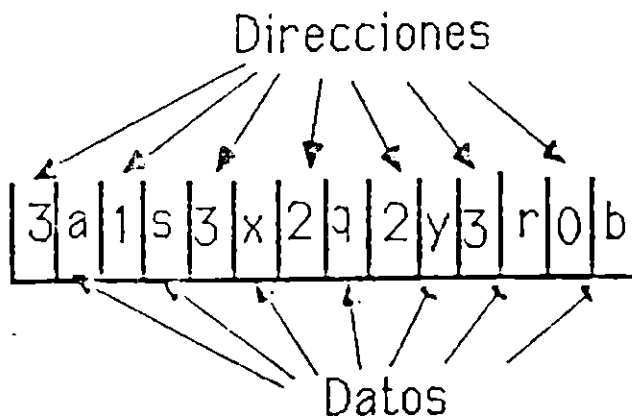
reln. NO-rem

Multiplexación .vs. Concentración



Synchronous Time
Division Multiplexing
(STDM)

Capacidad de línea:
de salida = suma de
las de entrada



Statistical Multiplexing

Línea de salida pu
tener menos capa
que la suma de las
entrada

Requiere buffer

Front End Processors

o IOP (I/O processors)

- Liberan al CPU de carga de TC
- Chequean errores
- Manejan terminales asincrónicas lentas
- Con disco puede ser store-and-forward
- Conversión de código
- Conversión de velocidades
- Conversión de formato
- Terminales incompatibles
- Compresión de texto
- Multiplexación estadística

→ chrl de terminales

desarrolla el pr.
control del manejo
de pr. de
{ rem (inpr.)
{ disco
{ etc.

CONMUTACION

- + Voz .vs. Datos
- + Conmutación de circuitos
- + Conmutación ^{de} mensajes
- + Conmutación de Paquetes

⇒ lo q' justifica la q' d'n
(red) de computadores.

Voz .vs. Datos

Gaps

"espacios"

Voz:

No hay largos

CiR enjudo cont-2.

Dato:

Son la regla

muy normales

Voz:

Poco, y continuo

Y cto → se nes. stre pre.
Y cto → nes (9kbs) por computador de.

Ancho de Banda

Dato:

so nes. AB

Altos e
intermitentes.

dado el vel. de inf.

No a veces nes la hme

y cuando la nes → queremos q' se
rápido.

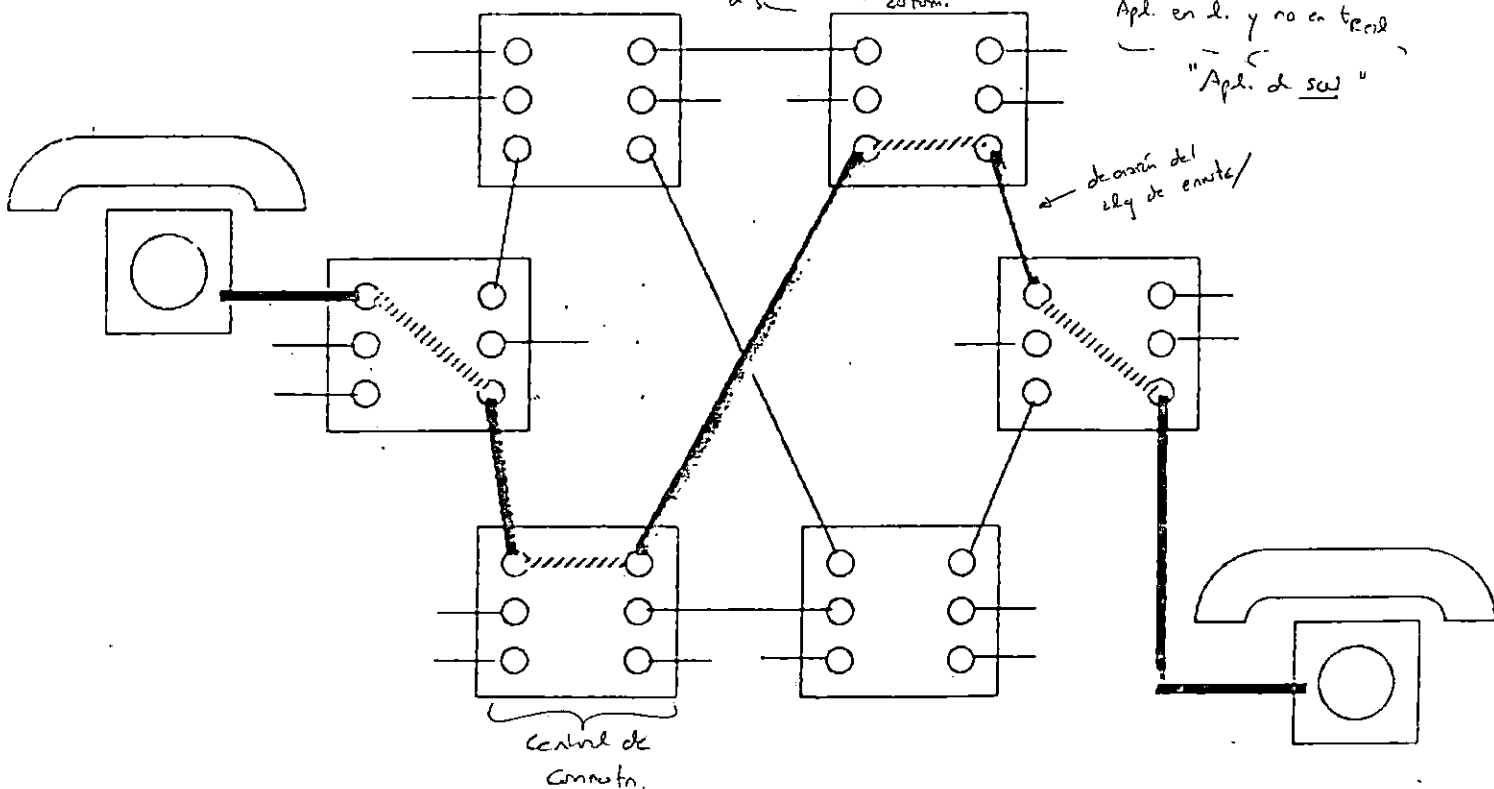
Conmutación de circuitos

(Voz, Tel.)

- Conexión permanece ^{todo} la llamada
- Circuito debe establecerse antes de transmitir
- t desde fin de marcar hasta timbre $\approx 10\text{seg}$
- Inadmisibles para ^{de procesamiento} DP en tiempo real. t es el tiempo de la com.
- Establecido el circuito:
- No hay peligro de congestión
- Unica demora es propagación $\leftarrow$ es más de todo el circ.
- $V \approx 150.000 \text{ km/seg}$ ($\frac{1}{2} V_{\text{vac}}$)
 pequeños $\rightarrow$ + demora por q. el control se hace en el paquete
- Reserva recursos anticipadamente
- Desperdicia canal no usado

Aplic. bit $\rightarrow$ com. de inf. en los q. luego se procesa
" Intercambio $\rightarrow$ más - usado, no requiere 2 en línea $\leftarrow$ en l. (sw apl.).
alg. no en l. (circuitos) $\leftarrow$ el transm.

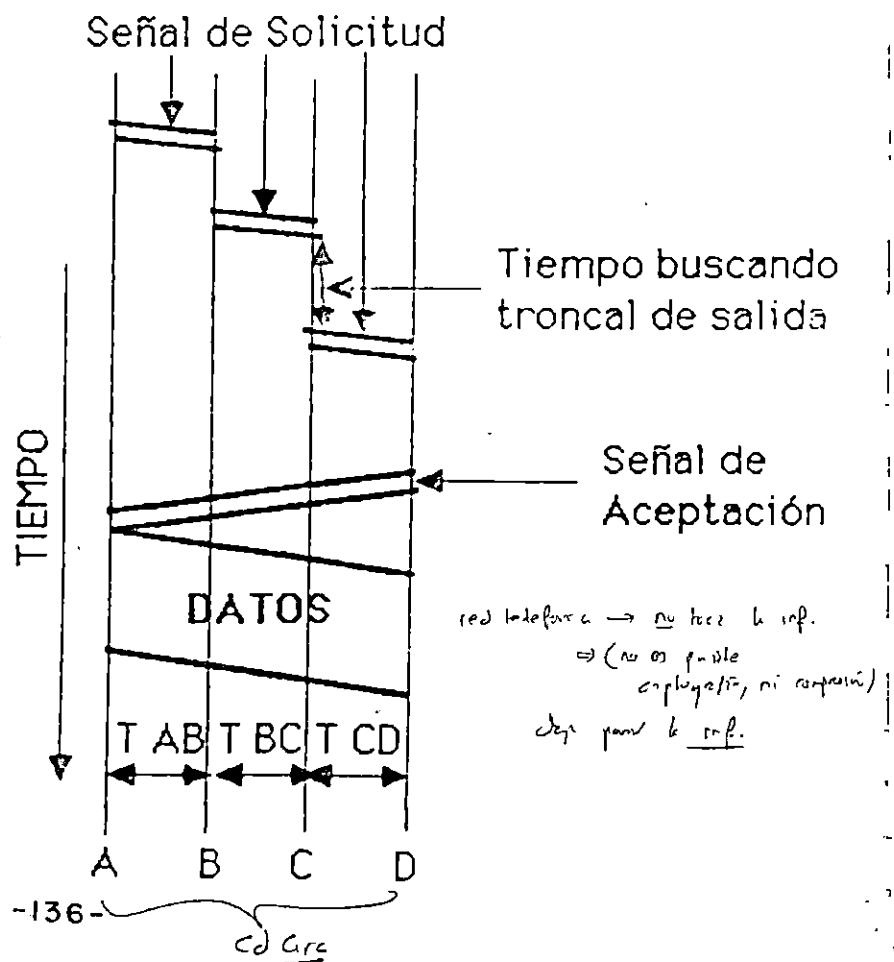
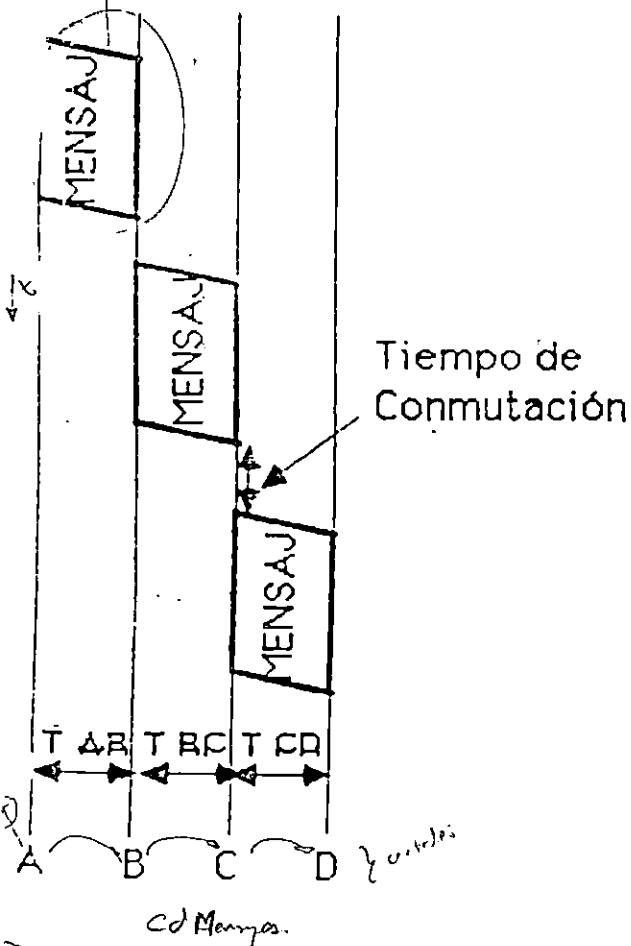
Si se está en l. $\rightarrow$ t. Recel.
Pensar, etc.
Apl. en l. y no en t. real
"Apl. de sw"



Conmutación ^{de} mensajes

- Transmisor envía un bloque de datos
- Se almacena en primera central (IMP)
- Más tarde se envía a la siguiente ^{de Cd Prog.}
- Cada bloque
- Se recibe totalmente
- Se inspecciona en busca de errores
- Se retransmite
- Redes así se llaman STORE-AND-FORWARD

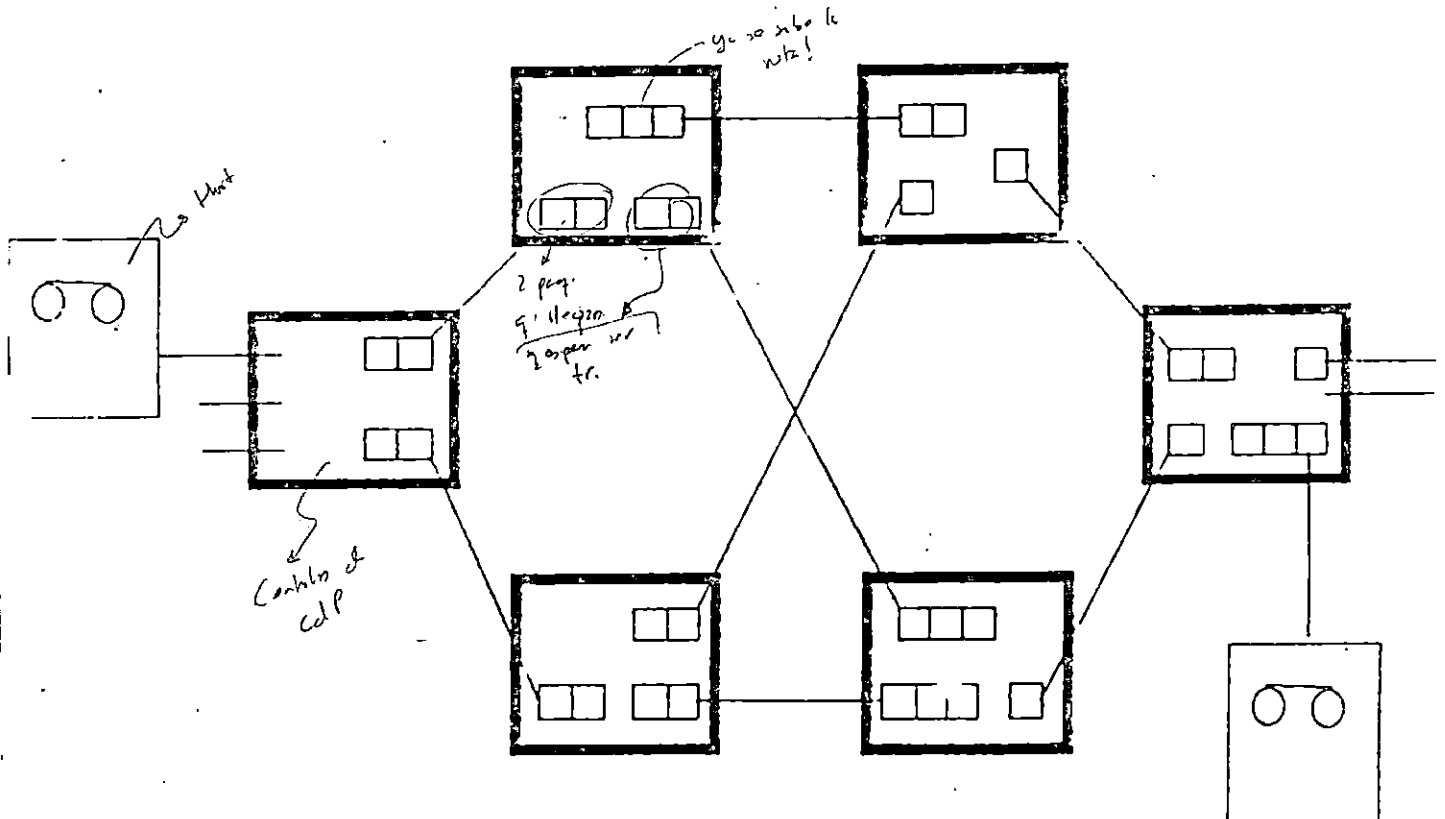
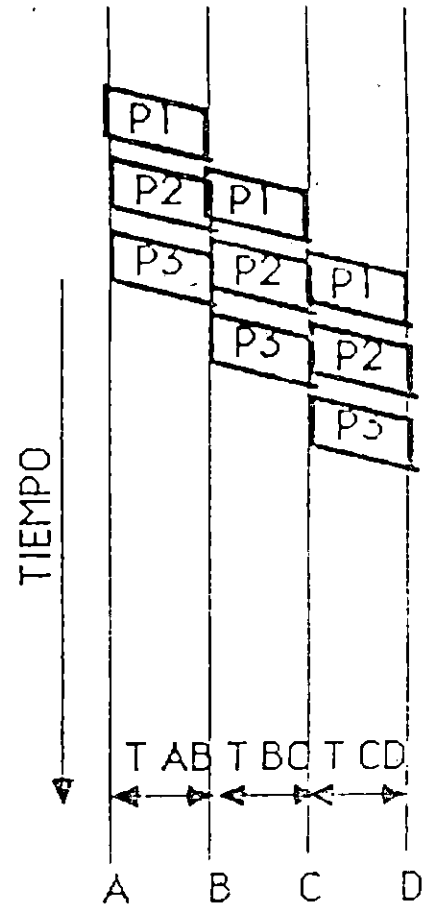
de Cd Prog.



Commutación de Paquetes

En MEMORIA

- En ^{cd} mensajes:
 - No hay límite de tamaño de bloque
 - Mensajes se deben guardar en disco
 - Un bloque puede ocupar línea mucho t
 - Malo para tráfico interactivo
- En ^{cd} paquetes:
 - Bloques pequeños
 - Se guardan en memoria
 - Un paquete ocupa la línea poco t
 - Bueno para tráfico Interactivo
 - Se aprovecha mejor la línea



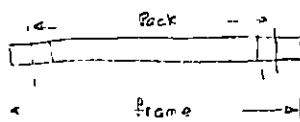
CAPITULO 3 - NIVEL DE ENLACE

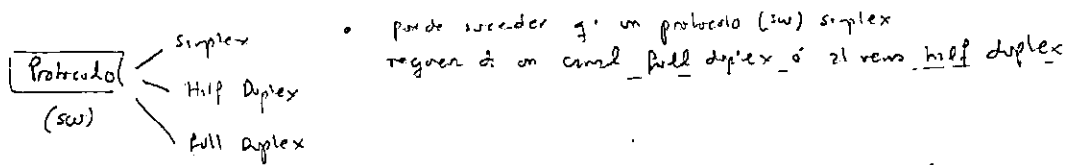
- Algoritmos para lograr comunic fiable (vs. ruido en una línea)
- Comunicación de máquinas adyacentes
- + Factores a considerar
 - Canales Imperfectos
 - Velocidad de transmisión limitada
 - t de propagación > 0
 - V de proceso de Hosts e IMPs limitada

entre pares de una red
no q. hacer confab.
en la red " "

Algoritmo de este nivel $\rightarrow$ protocolo vs. ruido (canales imperfectos)

Enlace $\rightarrow$ "frames"
red $\rightarrow$ "paquetes"





Protocolo #1

+ Condiciones ("ideales")

- Transmisor y receptor siempre listos o
- Tiempo de procesamiento = 0 disponibles a tr/rec.
- Buffers "infinitos" ↳ esperando en el fr.
- Canal "Perfecto"

+ Dos grandes procedimientos

- Transmisor - Corre en IMP transmisor
- Receptor - Corre en IMP Receptor

+ Estos llaman a los procedimientos:

+ Fromhost (var m: mensaje)

- Trae información de host a IMP

+ Tohost (m: message)

- Pasa mensaje de IMP a Host

+ Wait (var event: Evttype); → evita q' se piense en la concurrencia!

- Loop infinito esperando que algo ocurra
- En Evttype devuelve código de evento Evento Host → "No hay un frame"
- Evitamos pensar en otros procesos (concurr., etc.)

+ Getf (var r: frame); ↳ Para recibir del nivel físico

- Recibe un frame y lo deja en r

+ Sendf (s: frame)

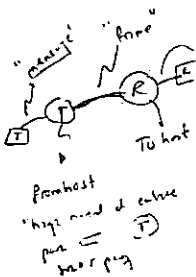
- Transmite el contenido de s ↳ Para enviar al nivel físico

Simplex:



Red:

- 2 hosts T y R
- 2 IMPs T y R



Protocolo #1

```

Program ridículo;
  const LastBit = ?;
  Eternidad = false;
  type bit = 0..1;
  mensaje = packed array
    [0..Lastbit] of bit;
  frame = Packed Record
    info: mensaje
  end;
  Evtype = (FrameArrival);
  
```

ponerle un límite al codinf del frame.
 10000 → el frame es de 10000 bits
 código de inf.

info: mensaje
 inf-ctrl: — (sup. q' no es necesaria!)

agregue el
 frame con el
 código de ctrl

Protocolo #1 (cont'd)

```
Procedure Transmisor; come en (1)
  var   s:frame;
        mbuffer:mensaje;
  begin
    repeat
      Fromhost (mbuffer);
      s.info:=mbuffer;
      sendf(s)
    until eternidad;
  end;
```

```
Procedure receptor; come en (2)
  var r:frame;
      Event: Evtype;
  begin
    repeat
      wait(event);
      getf(r);
      Tohost(r.info);
    until eternidad
  end;
```

Condiciones

del protocolo # 2

- Transmisión de datos en una dirección (simplex)
- Canal perfecto
- + t de procesamiento en receptor > 0
- + Se debe evitar "inundarlo"
- + Si $(T_p = T(\text{getf}) + T(\text{Tohost})) \text{ent}$
- $\Rightarrow \frac{1}{T_p} \text{ frame/} \frac{1}{T_p}$ debe ser entre enviar + el un frame por el T_p !
- Control de flujo una técnica de
- + Trans espera para c/frame ack de recept.
- Protocolos stop-and-wait es a q tr d frame y debe esperar una confirmación del resp. para enviar el sig frame. el protocolo es
- Aunque simplex necesita canal half-duplex al menos.
- Unico evento posible: Frame arrival (Ack)

transmisor va a recibir en Tr y Rec

Protocolo #2

```

Procedure      transmisor;
var
    s:frame;
    buffer:mensaje;
    event:Evtype;

begin
    repeat
        Fromhost(buffer);
        s.info:=buffer;
        sendf(s)
        "esperar la confirmación" [wait(event)]
    until eternidad
end;
```

Protocolo #2 (cont'd)

```
Procedure Receptor;  
  var    r,s: frame;  
         event: Evttype;  
begin  
  repeat  
    [wait(event);  
    getf(r);  
    Tohost(r.info);  
    Sendf(s)]  
  until eternidad;  
end;
```

Consideraciones

el protocolo #3

- Simplex
- Canal con ruido
caracter de canal y info. redundante q' va con el frame
- Se usa CRC o checksum
- Si CRC cuadra con error cualquier protocolo falla
- Transm hace timeout para evitar "abrazo mortal"
- Control de duplicados por pérdida de acks
- + Solución: Colocar #s de secuencia a los frames
 - Deben ocupar poco espacio
- + Confusión posible:
 - (- Un frame dado y su sucesor dado el protocolo stop & wait y q' solo se envía un frame.
 - (no) en error (- Un bit es suficiente 1-0 → 1-0 → 1-0 → ...
- + Timeout debe ser suficientemente grande
 - Timeout prematuro puede hacer fallar...
- + Nuevos procedimientos
- + StartTimer(k:SequenceNr); NonSec el cual le genera timeout el time-out → comienza el periodo de timeout

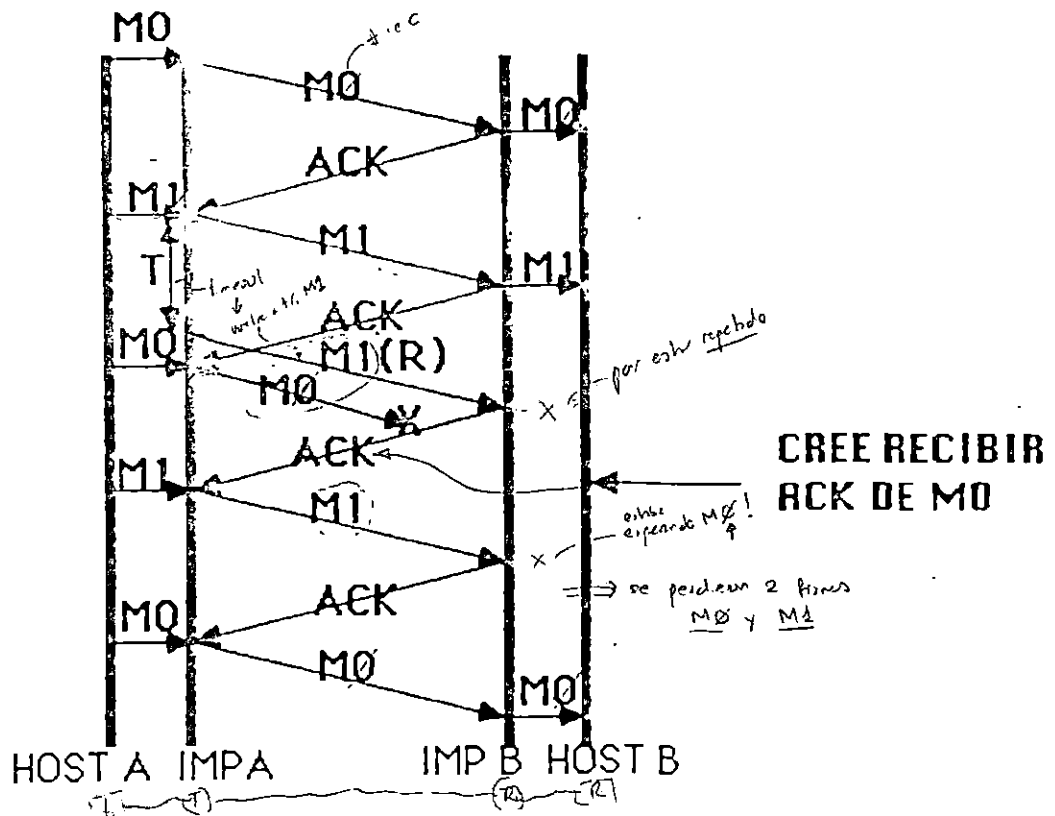
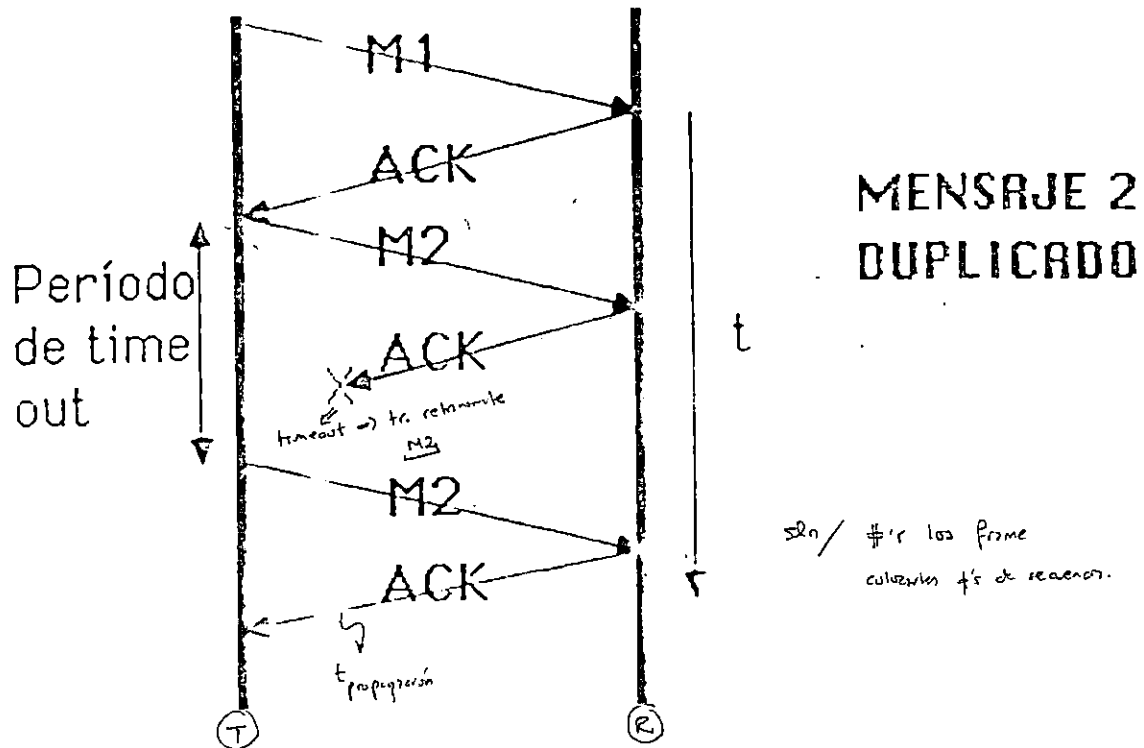
CRC → 16×2 generando $\times H_{16}$ (chips) int.
por ello es q' no se en el SW

Y solo 3 en la línea
un error se refleja
si usa una int.

Consideraciones (cont'd)

- Arranca ^{el} timer para timeouts
- + Inc(var k:SequenceNr);
 - Incrementa k circularmente ^{anterior}
- + ~~Nuevos tipos de datos~~ elem. del ambiente
- + MaxSeq
 - ~~Este~~ con #máx de secuencia
- + SequenceNr
 - Tipo de las variables de número de seq
- + Seq
 - Campo adicional en registro frame
 - LLeva el número de secuencia
- + Evtype = (Frame Arrivel, CksumErr, Timeout)
 - Nuevos elementos en el conjunto
 - Cksumerr: Se genera en receptor ^{cuando recibe un CRC erróneo.}
 - Timeout: Se genera en el transmisor ^{para retr. lo que mandó.}

pro
 push
 <---
 pop
 2 veces
 cuando a la vez
 por q' retr.
 opera q' el
 el bit termina
 el 80st para
 retr



Protocolo #3

PROGRAM

PAR

```

const  Lastbit = ?;
        Eternidad = false;
        Maxseq = 1;  { mx # sec. }
type   bit = 0..1;
        sequenceNr = 0..Maxseq;
        mensaje = packed array [0..Lastbit] of bit;
        frame = packed record
            seq: sequenceNr;
            info: mensaje;
        end;
        Evtype = (FrameArrival, CksumErr, Timeout);

```

Procedure Transmisor

```

var NextFrameToSend: SequenceNr;
    s: frame;
    buffer: messagenJ;
    event: Evtype;

begin

```

Protocolo #3 (cont'd)

```

NextFrameToSend := 0;
FromHost(buffer);
repeat
  s.info := buffer;
  s.seq := NextFrameToSend;
  sendf(s);
  StartTimer(s.seq)
  wait(event);
  if(event = FrameArrival)then
    begin
      FromHost(buffer);
      inc(NextFrameToSend);
    end
until eternidad
end;

```

Procedure Receptor;

var FrameExpected: SequenceNr;

 r,s: frame;

 event = evtype

Protocolo #3 (cont'd)

```

event Etype;
begin { Receiver }
  FrameExpected := 0;
  repeat
    wait(event);
    if (event = Frame Arrival) then
      begin
        getf(r);
        if (r.seq = FrameExpected) then
          begin
            ToHost(r.info);
            inc(FrameExpected)
          end;
        sendf(s);
      end
    until eternidad
end; { Receiver }

```

Handwritten annotations:

- A handwritten "FA" with an arrow pointing to the "Frame Arrival" condition in the if statement.
- A handwritten "ck sum for" with an arrow pointing to the "wait(event)" statement.

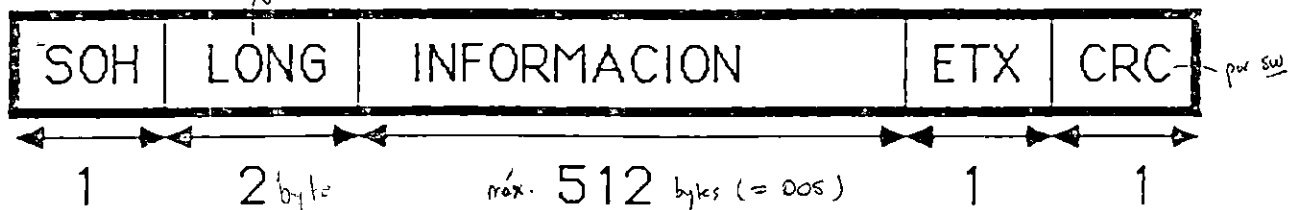
Ejemplo Práctico

- + Generalidades
- + Manejo de transparencia
- + Sincronización:
- + Cuadro estructurado

Generalidades

- Transferencia de archivos entre 2 micros.
- Escrito en "C"
- ^{protocolo} Stop and wait (solo hay un frame en tránsito).
- Transmisión asincrónica Gr-a-Civ
la inf. de char.
está en el carácter
- Formato del frame
- CRC = XOR de Long & Información crc = xor (c.c., long + info)
- Protocolo Half Duplex tr. de 2ch
(no simultánea)

FORMATO DEL FRAME



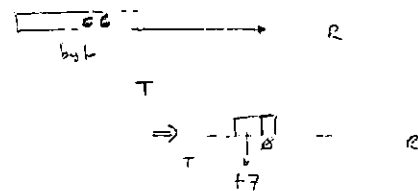
Manejo de transparencia

+ Caracteres de control usados

- NULL ASCII 0 Character stuffing (transpare
- SOH ASCII 1 Sincr y principio frame
- LISTO ASCII 2 Sincronización lobo p' rec. } R→T
- ETX ASCII 3 Fin de Frame
- EOT ASCII 4 Fin de transmisión
- NACK ASCII 5 Negative Ack ⇒ retr. de un frame
- ACK ASCII 6 Confirmación

+ Transmisor

- Al detectar ASCII ≤ 6
- Inserta un NULL
- Le suma 7 al caracter



+ Receptor

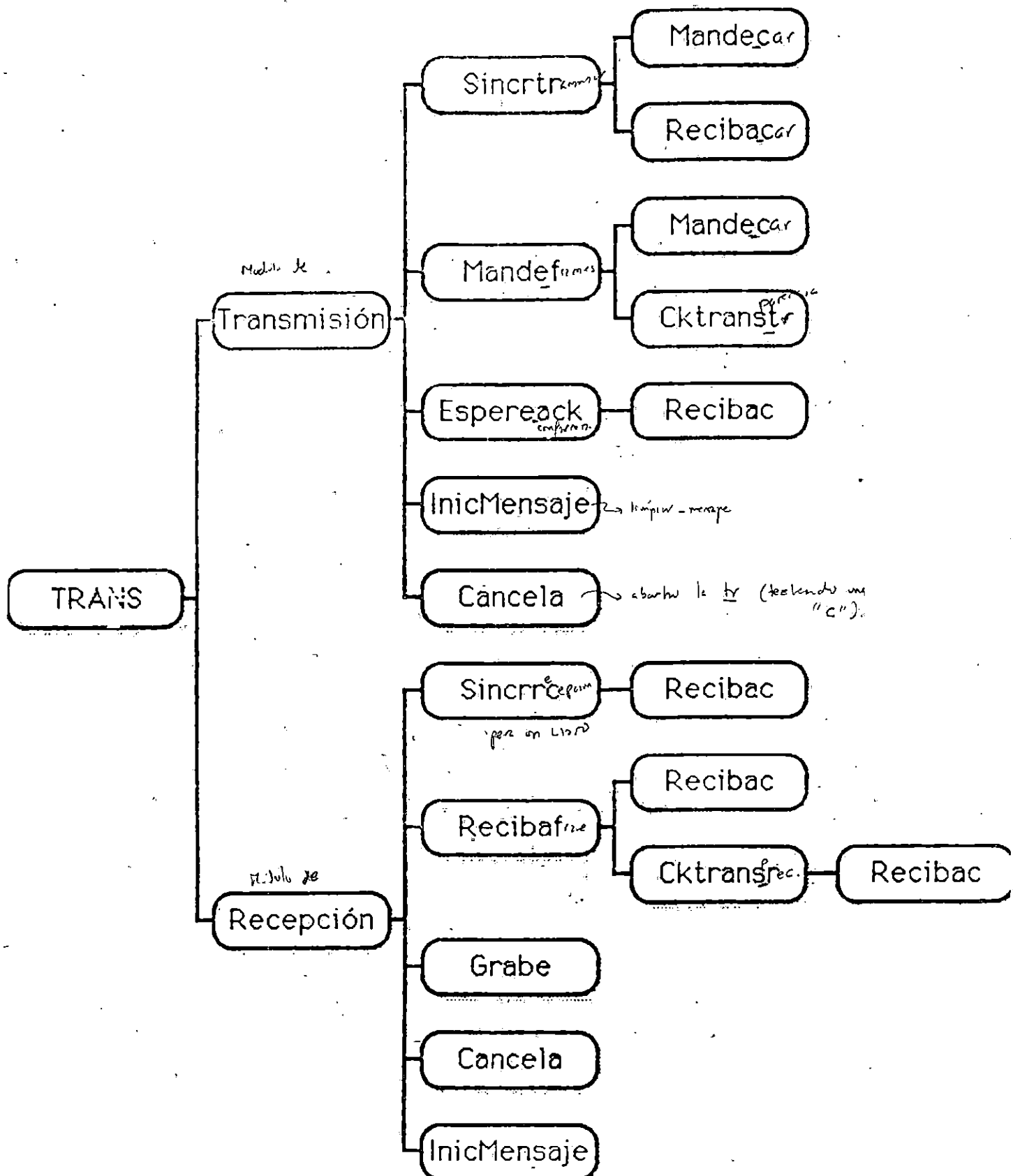
- Al detectar NULL ⇒ lo ignora
- ~~Lo ignora~~
- Resta 7 al caracter que le sigue
-

Sincronización:

- Transmisor Envía SOH
- Espera LISTO (ASCII 2)
- Timeout después de 2.000 intentos

a. timeout

CUADRO ESTRUCTURADO



Código

- Definiciones y datos
- Programa Principal
- Transmisión
- Recepción
- Sincrtr
- Mandef
- EspereAck
- Sincrrc
- Recibaf
- Grabe
- Cktranst
- Cktransr
- Mandec
- Recibac
- InicMensaje
- Cancela

Definiciones y datos

```
#include "stdio.h"
```

```
#include "fcntl.h"
```

```
#define NULL 0
```

```
#define SOH 1
```

```
#define LISTO 2
```

```
#define ETX 3
```

```
#define EOT 4
```

```
#define NACK 5
```

```
#define ACK 6
```

```
int    puerto, archivo; nodo
```

```
char    arch[15];
```

(nombre del archivo)

```
struct framedef
```

```
{
```

```
    char cabeza;
```

```
    int longitud;
```

```
    char mensaje[512];
```

```
    } frame;
```

*ETX y EOT
son "byte"*

*por lo mismo
el fin es unico
y no finco*

Programa Principal

```
main()
{
    char opción;
    modo = 2 | O_RAW;
    puerto = open("com1:", modo);
    printf("%s", "\nEscriba el nombre del
        archivo que desea manejar\n");
    scanf("%s", arch);
    printf("%s", "\n Desea transmitir o
        Recibir (T/R)\n");
    do
    {
        opción = getch();
    }
    while (opción != 'T' && opción != 't' &&
        opción != 'R' && opción != 'r');
    if (opción == 'T' || opción == 't')
        transmisión();
    else
        recepción();
}
```

Handwritten notes:

- Next to `opción`: *e/t*
- Next to `modo`: *depende del compilador*
- Below *depende del compilador*: *modo* with a bracket pointing to `O_RAW` and `O_TEXT` (labeled *translado* and *no-translado* respectively).

Transmisión

```

transmisión()
{
    int siga, j, bueno, fin, numf;
    numf = 0;
    if ((archivo = open(arch, modo) != -1))
    {
        inicmensaje();
        siga = 1;
        fin = 0;
        while (siga && !fin)
        {
            numf++;
            j = read(archivo, frame.mensaje, 512);
            frame.cabeza = SOH;
            frame.longitud = j;
            if (j < 512)
            {
                fin = 1;
            }
        }
    }
}
    
```

Handwritten notes:
 - *g' detecha puros en la tr (abarto)* (pointing to 'g' in 'g' in the code)
 - *entra de la confirm* (pointing to 'confirm' in the code)
 - *indica g' se cerró el arch (long < 512)* (pointing to 'long < 512' in the code)
 - *contador de frame (pinchito)* (pointing to 'frame' in the code)
 - *contador de carácter* (pointing to 'numf' in the code)
 - *Por # carácter q' real se le gana* (pointing to 'numf' in the code)
 - *limpio el buffer con zeros* (pointing to 'inicmensaje()' in the code)
 - *una vez / frame* (pointing to 'while' in the code)
 - *se cerró el archivo* (pointing to 'if (j < 512)' in the code)

Transmisión (cont'd)

```
do      2 ver / retransmision del pme.  
{  
    printf("%s %5d", "\n Transmitiendo  
        frame:", numf);  
    siga = sincronizar();  
    if (siga != 0  
    {  
        mandeframe(fin);  
        bueno = esperarack();  
        siga = cancelar();  
    }  
} while (!bueno && siga);  
inicmensaje();  
close(archivo);  
else  
    printf("%s", "El archivo no existe\n");  
close(puerto)  
}
```

Handwritten notes:

- et EOF / ETX
- si Pa = 4.
- retraso max c/c o Non-Int. p/llo
- si retraso ACK
- cancelar

Recepción

recepción()

{

int siga,j,bueno,fin,cr;

*← código de estado por recibir frase
retorno*

if((archivo=creat(arch,modo) != -1)

{

inicmensaje();

1 siga = 1;

fin = 0;

while(siga && !fin)

{

siga = ^{no}sigrec();

if (siga)

{

2 cr = recibaf();

3

↳ si recibí el frase

}

}

Recepción (cont'd)

```

switch (cr)
{
    case 0: mandec(ack); "bueno" ← T.
        fin=grabe();
        break;
    case 1: mandec(NACK); /* CRC malo */
        break;
    case 2: mandec(ACK); /* llegó EOT */
        % fin=grabe();
        fin=1;
        break;
    case 3: fin=1; /* TimeOut esp frame */
        break;
} /* switch */
3
{
    siga = cancela();
} /* if */
4
{
    inicmensaje();
} /* while */
{
    close (archivo);
} /* if */
else
    printf("%s", "Error al leer el archivo\0");
close(puerto);
}
    
```

Sincrtr

```
int sincrtr();
{
    char car;
    int i;
        ↳ "loop out" (contador de vezes/intento)

    i=0;
    do
    {
        i++;
        mandec(SOH);
        recibac(&car);
    }
    while (car != LISTO && i < 2000);
    if (car == LISTO)
        return(1);
    else
    {
        printf("%s", "\nNo pude sincronizar\0");
        return(0);
    }
}
```

Mandef

```
mandef(pfin)
```

```
{ int pfin;
```

```
char *p, crc, car;
```

```
int i, tope, t;
```

ⓑ para saber si dato 3 no
necesario de transpencia

```
p = &frame;
```

```
mandec(*p);
```

```
p++;
```

```
crc = 0;
```

+ crc. a envia

```
tope = frame.longitud + 2;
```

```
for (i=0; i<tope; i++; p++)
```

```
{
```

```
car = *p;
```

```
crc = crc xor car;
```

```
t = cktranst(car);
```

```
if (t)
```

```
car += 7;
```

```
mandec(car);
```

```
}
```

```
if (!pfin)
```

```
mandec(ETX)
```

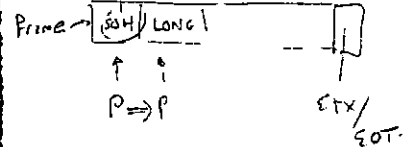
```
else
```

```
mandec(EOT);
```

```
mandec(crc);
```

```
}
```

XOR
(long, comp. de inf)



EspereAck

```
int espereAck()
{
    int i;
    char car;

    i = 0;
    do
    {
        i++;
        recibacal(&car);
    }
    while (car != ACK && car != NACK
           && i < 2000);
    if (car == ACK)
        return(1); OK p' bueno
    else
        { no NACK o timeout +/
            if (i == 2000)
                printf("%s", "\nTimeout esperando ACK");
            return(0);
        }
}
```

Sincrrc

```
int sincrrc()
{
    char car;
    int i;

    i = 0;
    do
    {
        por ref!
        recibac(&car);
        i++;
    }
    while (car != SOH && i < 2000);
    if (car == SOH)
    {
        mandec(LISTO);
        return(1);
    }
    else
    {
        printf("%s", "\nImposible Sincronizarse\0");
        return(0);
    }
}
```

Recibaf

```

int recibaf()frame
{
    char car, *p, car1, crc;
    int i, pérdida
    ↪ B p' cuando se detecta pérdida de caracteres.

    p=&frame;
    p++;
    i=0;
    do
    {
        recibac(&car);
        i++;
    }
    while (car == SOH && i<20);
    if (car == SOH)
    {
        printf("%s", "\nTimeout esperando frame\0");
        return(3);
    }
}

```

Llegó el ETX y no se detecta q' la long es frame.long.

Purgar la línea de SOH's.

Recibaf^{rene} (cont 1)

else

{

crc = 0;

*p = car;

cktransr(p);

crc = crc $\oplus$ (*p);

p++;

recibac(p);

cktransr(p);

crc = crc $\wedge$ (*p);

p++;

pérdida^{de-crs.} = 0;

for (i = 0; i < frame.longitud &&
!pérdida; i++, p++)

{

recibac(p);

if ((*p) == ETX $\vee$ (*p) == EOT)

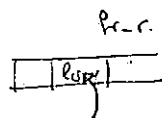
pérdida = 1;

cktransr(p);

crc = crc $\oplus$ (*p);

}

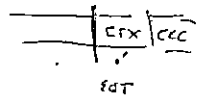
XOR



crc

2^o byte de longitud

frame.
long



ETX

Recibaf (cont2)

```
if (!pérdida)
{
    recibac(&car1)
    if(car1 != ETX && car1 != EOT)
    {
        printf("%s", "\nRetransmisión por
                ETX no recibido\n");
        return(1);
    }
    recibac(&car);
    if (car != crc)
    {
        printf("%s", "\n Retransmisión por
                crc malo\n");
        return (1);
    }
}
```

Recibaf (cont3)

```
    else
        if (car1 == ETX)
            return(0);
        else
            return(2); /*Fin de Transmisión*/
    }
    else
    {
        printf("%s", "\nRetransmisión por pérdida
            de caracteres\0");
        return(1);
    }
}
```

Grabe

```
int grabe()  
{  
    int i;  
    i = write(archivo, frame.mensaje,  
              frame.longitud);  
    if (i < frame.longitud)  
    {  
        print ("%s", "\nError de grabación.\n0");  
        return(1)  
    }  
    else  
        return(0);  
}
```

Cktranst_r

```
int cktranst(pcar)  
  
char pcar;  
{  
    if (pcar < 7)  
    {  
        mandec(NULL);  
        return(1);  
    }  
    else  
        return(0);  
}
```

Cktransr

```
cktransr(pcar)
char *pcar;
{
    if ((*pcar) == NULL)
    {
        recibac(pcar);
        (*pcar) -= 7;
    }
}
```

Mandec

```
mandec(pcar)
char *pcar;
{
    char car;
    int i;

    car = pcar;
    write(puerto,&car,1);
}
```

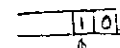
Recibac

```
recibac(pcar)
char *pcar;
{
```

```
    char car;
    int t;
```

```
    do
    {
        t = inp(0x0032);
        while ((t & 2) == 0);
        car = inp(0x0030);
        *pcar = car;
    }
```

→ leer dh 2 el LSR & 1 chip
p' ver si hay o no car


↑
si hay car

InicMensaje

```
inicmensaje()
```

```
{
```

```
    int i;
```

```
    for (i=0; i < 512; i++)
```

```
        frame.mensaje[i] = '\0';
```

```
}
```

Cancela

```
int cancela()
{
    int i;
    char letra;

    if ((i=kbhit()es word))
    {
        letra = getch();
        if (letra == 'c' || letra == 'C')
            return 0;
        else
            return (1);
    }
    else
        return(1);
}
```

Filosofía

Protocolo #4 → SW
de 1 bit
(s & w)

Generalidades

Consideraciones

Simulación de
múltiples Timers

Comportamiento
de ventanas

Protocolo #5

Consideraciones

Comportamiento
de ventanas

Protocolo #6

Goback N

$\Sigma_T \rightarrow \text{SDLC}$

Pipelining

Selective repeat

$\Sigma_T \rightarrow \text{HDLC}$

Problemas DUPLEX

técnica
Sliding Window

p' protocolo
SDLC,
HDLC
&
X25

Pro/ de las redes públicas

p' canales
satelitales

+ de un frame
en tránsito x la
línea
(window timers ≠ 's)
↓
emulador con
↓ solo reloj

por def.
variable en la
de fin
de trama

lo contrario a
Stop & Wait

sin retrans. recepción

3
2
1
0
X
Y
Z
W
V
U
T
S
R
Q
P
O
N
M
L
K
J
I
H
G
F
E
D
C
B
A

net. unid.

-172-
Si se da a
⇒ retrans. de la
de tras
(solo retrans. de)
Goback N

retrans. / límite
C / slot de la v.e. ⇒ no de 2
buffer
(cálculo)

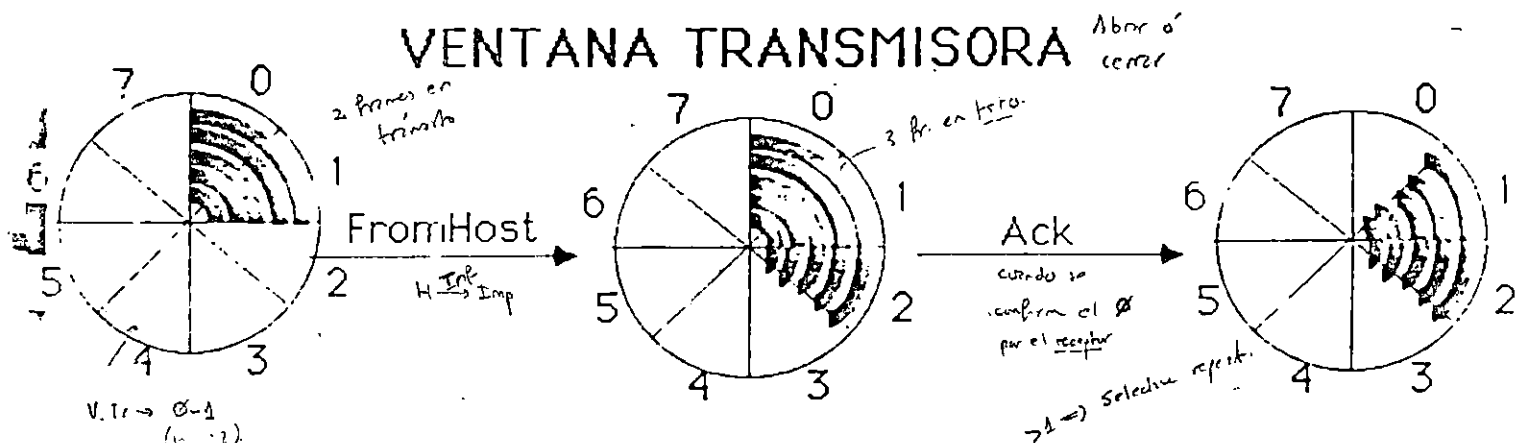
Sliding Window - Filosofía

VENTAS FRAMES → controladas BISN uno-a-uno.

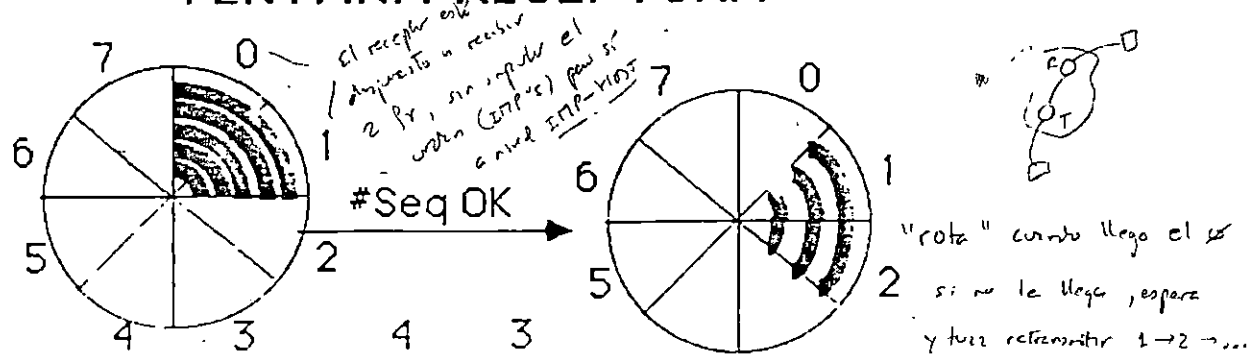
- permite com
 - Dúplex (21 menos HD), desde el pto de vista del SC } (no "chocan" el receptor; no agotar su buffer p' q' los datos).
 - Usan piggybacking los frames llevan "a caballo" las confirmaciones.
- Demora de ack por datos limitada 
- Rango de #seq manejado circularmente 
- + Ventana del Transmisor
 - #s de seq de frames enviados no confirmados ⇒ (nos un buffer si es necesario, que son retornables)
- + Ventana del Receptor
 - #s de seq aceptables en un momento dado
- Las ventanas pueden ser de tamaños ≠s
- + Ventana del transmisor
 - Se incrementa al recibir mensaje del host
 - Se disminuye al recibir confirmación
- + Si n = tamaño máx de vent del transmisor

confirm. de 7
problema
timeout se
complete por q'
B no tiene más
q' mandar
seq/
timeout p' B
D está q' en
A se cumple, y
le hace más
la conf. sola
la manda

VENTANA TRANSMISORA



VENTANA RECEPTORA



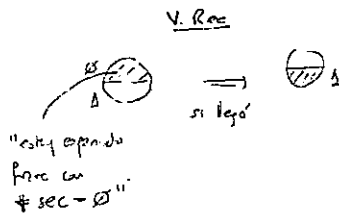
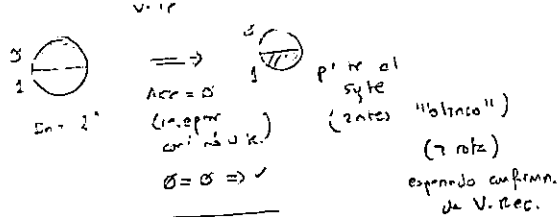
10 Ventanas sin el tamaño 75

skip & wait → solo 1 buffer/ciclo
"cuello" el tr. por q'
no "guste" al rec.

es un chid de plugo
por q' espera a que le v. A.
el IMP lo hace q' dem. a su host
→ como el pared. double
porq' no hace buffer

Sliding Window -... (cont'd)

- Se necesitan n buffers en transmisor
- Por qué? Para permitir retransmisiones
- + Si ventana de transmisión llena ent
 - IMP transmisor deshabilita su host
- + Ventana del receptor
 - Recibe frame
- + Si su #seq coincide con límite inferior
 - Pasa mensaje al host
 - Envía ack
 - Rota ventana
- + Si tamaño > 1 ent
 - IMP acepta frames en desorden
 - Los pasa al host en orden

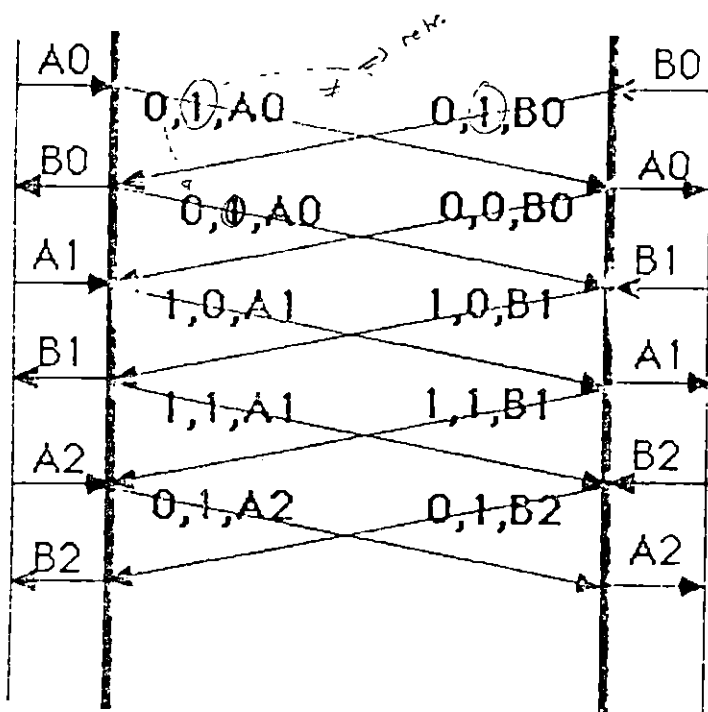
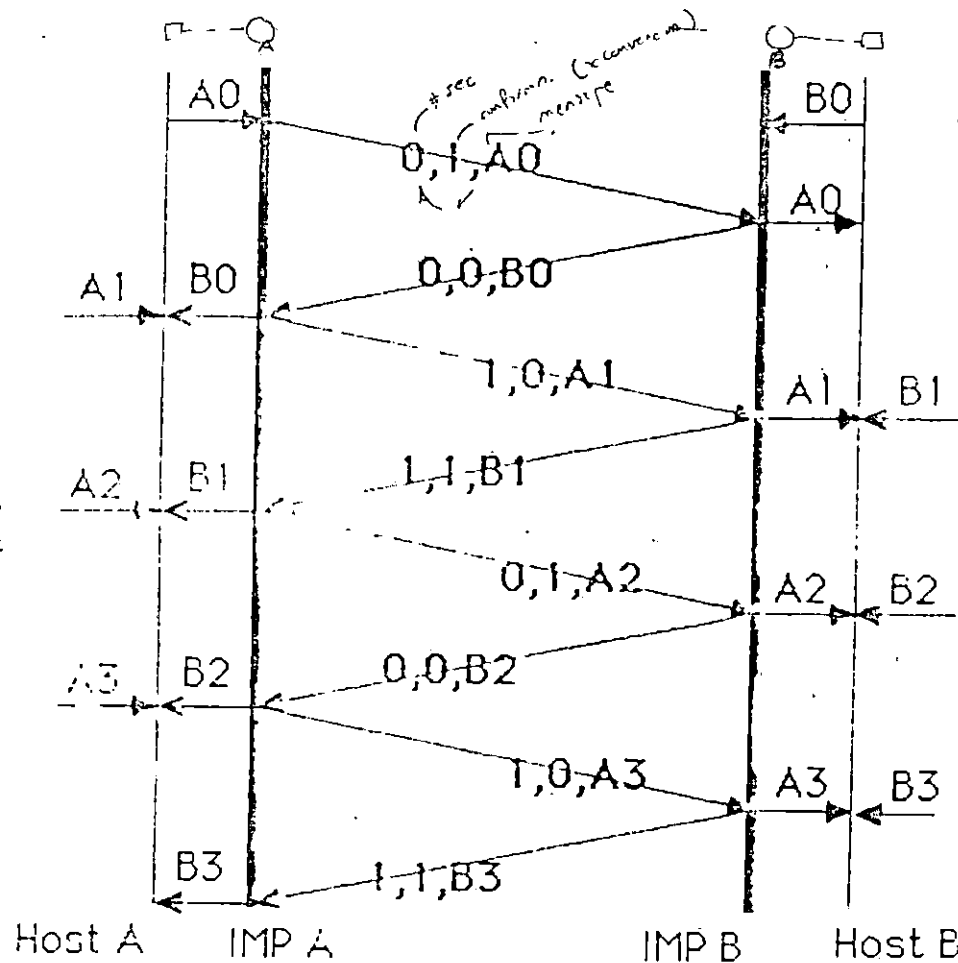


Protocolo #4 - SW de 1 bit

- Stop-and-Wait $\rightarrow$ porq' el max t.m. de la U.Tr es 1
 $\Rightarrow$ (1 solo frame en tránsito)
- Dúplex $\times$ Pipelining (1 de 3 frame en tránsito)
- Uno de los dos manda primero
- Campo de ack en el frame
Contiene #seq del último recibido OK
- Siempre que llega un frame se envía otro
- Si Ack recibido = posición vent transm ent
cambie vent y traiga nuevo mens de host
Sino
retransmita.
- Un sólo procedimiento a ambos lados (antes 2)
Hay una diferencia
En uno sólo
sendf y starttimer fuera del loop
 $\hookrightarrow$ p' darle prioridad a uno de los 2 lados.

packet
number Ack
on a link
to receive

Ack = # seq del ult
frame recibido OK.



Host A IMP A IMP B Host B

Código Protocolo #4

```

Program SW1b;
  const Lastbit = ?
        Eternidad = false;
        Maxseq = 1;
  type  bit 0..1;
        sequenceNr = 0..Maxseq;
        mensaje = packed array [0.. Lastbit]
                        of bit;
        frame = packed record
            seq : sequenceNr; de 1 pr recibido
            ack : sequenceNr; de pr q' estara confirmado
            info: mensaje;
        end;
        Evttype = (FrameArrival, de int al receptor CksumErr,
                    le int a la de recepo. Timeout); (perzaine CAC)

```

```

Procedure idayvuelta;
  var  NextFrameToSend: SequenceNr;
        FrameExpected: SequenceNr;
        recebo P, S: frame;
        buffer: mensaje;
        event: Evttype;

```

```

begin
  NextFrameToSend := 0;
  FrameExpected := 0;
  FromHost (buffer);
  s.info := buffer;
  s.seq = NextFrameToSend;
  s.Ack := 1 - FrameExpected;
  [sendf (s)
  startTimer (s.seq);
  repeat if request expires!
    wait (event);
    if (event = FrameArrival) then
      begin
        getf (r);
        if (r.seq = FrameExpected) then
          begin
            ToHost (r.info);
            inc (FrameExpected);
            - end; "role" V. Pec.
          if (r.ack = NextFrameToSend) then
            begin
              FromHost (buffer);
              inc (NextFrameToSend);
            end;
          end;
        s.info := buffer;
        s.seq := NextFrameToSend;
        s.ack := 1 - FrameExpected;
        [sendf (s);
        startTimer (s.seq)] if checker el. Time-out.
      until Eternidad
    End;
  End;

```

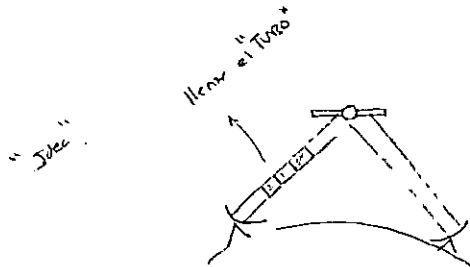
solo en $\rightarrow$
 $\frac{A}{p' q' A \text{ enve}}$
 $\underline{\underline{2^\circ}}$

if eval - checksum error
v timeout
→ rel. del = fine

11 sec
(uns br.)

confirm.
(USN receipt)

of character of Time-out.



protocolo

"Pipelining: Generalidades"

(X25, Dgch2, etc.).

- Optimiza uso AB si T_{prop} /Tamaño frame es grande
- Ejemplo: Canal de satélite

$V_{trans} = 50Kbps$ (probable!)

Tamaño del frame = 1000 bits (≈ 128 bytes).

$T_{prop} = 250$ msg

Protocolo stop-and-wait

se envía 1 rda pr. y hay que esperar la confirmación
pr. envía otro.

t(msg)	EVENTO
0	Transm inicia transm de 1er frame
20 $\neq$ 20 $\frac{1000}{50K}$	Frame completamente transmitido
$\frac{1250}{270}$	Frame llega al receptor
$\frac{1250}{520}$	Ack llega al transmisor

Conclusión: canal desperdic. 500 de los 520msg (2)

eficiencia: $4\% \frac{2}{520}$ $\frac{520 - 100}{520} = ?$

"pipelining" permitir q' el tr. pueda enviar varios fr. antes de esperar confirmación del 1º q' envió.

Solución: Hacer ventana de transmisor > 1

Envía varios frames antes de recibir ack

Cuál es la eficiencia del ejemplo si Vent transm = 20?

Protocolos q' usan Pipelining:
 con GBN
 con S.R.

Consideraciones de Go Back n

x25

- IMP receptor sólo acepta frames en orden
- Ventana transmisora > 1
- Ventana receptora = $\frac{1}{T_{\text{tm}}}$ (no se puede desorden!)
- Transmisor tiene varios frames en espera de ACK
 c/u le envía a x veces (!)
 ⇒ Necesita controlar timeout de c/u por aparte
 Qué hacer? Simular timers por software
- Si (transmisor recibe ACK de un #seq) ent

hay ack implícito de todos los anteriores

- Piggybacking en dos sentidos

¡¡¡! → • Campo ACK tiene #seq de último frame recibido OK

- Tamaño máx de vent transm debe ser Maxseq

Análisis: Máxseq = 7, ventana transm = 8

Transmisor envía frames 0-7

si lido → Llega ACK de 7 piggybacked del receptor.
 lo llega OK

Transmisor envía otros 8 frames (0-7)

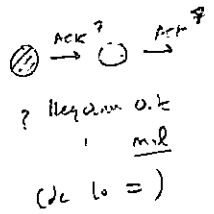
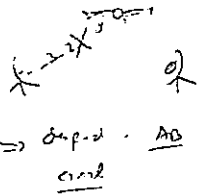
Llega ACK de 7 piggybacked

Este ACK es del 1er grupo o del segundo?

Si los últimos 8 se perdieron todos?

⇒ lím. V.Tr. de b. ac. < Máx seq.

problema de GBN

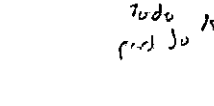


¿llegaron a.k.
 mil
 (de lo =)

(por el ult
 # seq. a.k. fue 7!)

⇒ la vent. de
 br. máx

la podemos llevar



tudo ack 6
 (por lo) → se puede
 dar lugar?

Simulación de múltiples Timers

- Un sólo reloj en HW puede controlar varios timers
- Se usa lista encadenada

Cada nodo de la lista contiene.

Ticks adicionales al nodo anterior

del frame controlado por el nodo

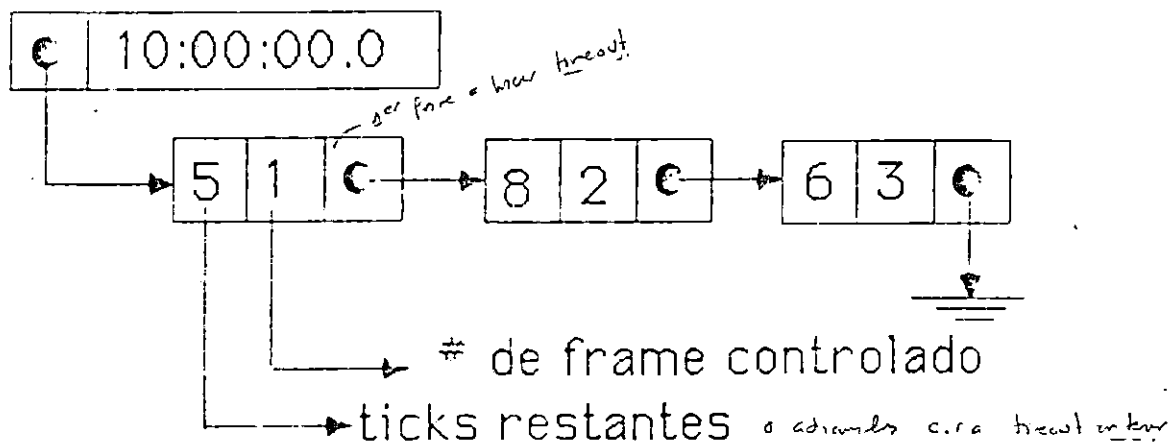
Apuntador al siguiente nodo

Cada tick de reloj HW actualiza solo la cabeza de la lista

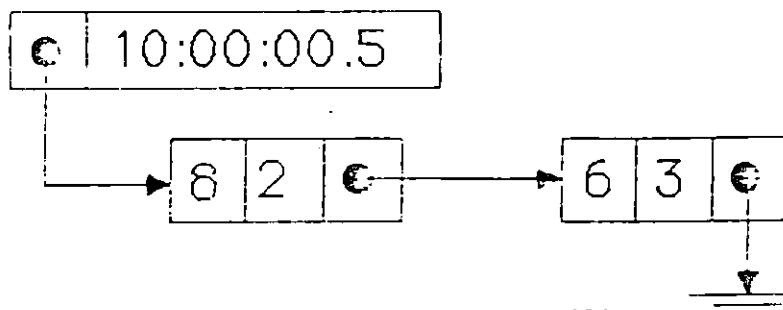
Cuando hay timeout se saca nodo de la cabeza

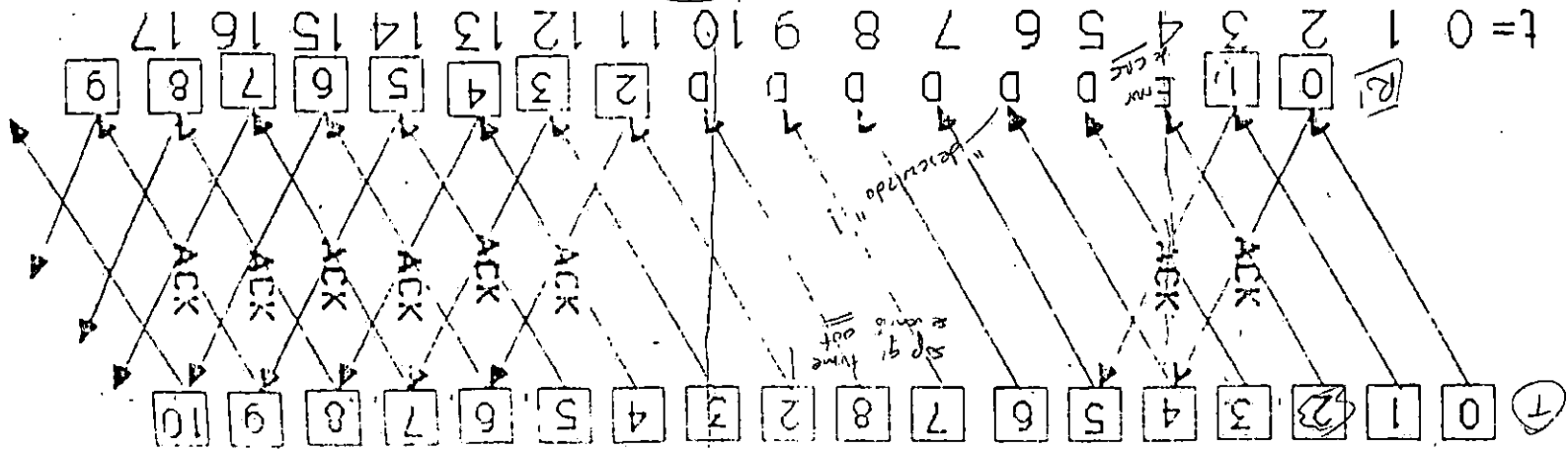
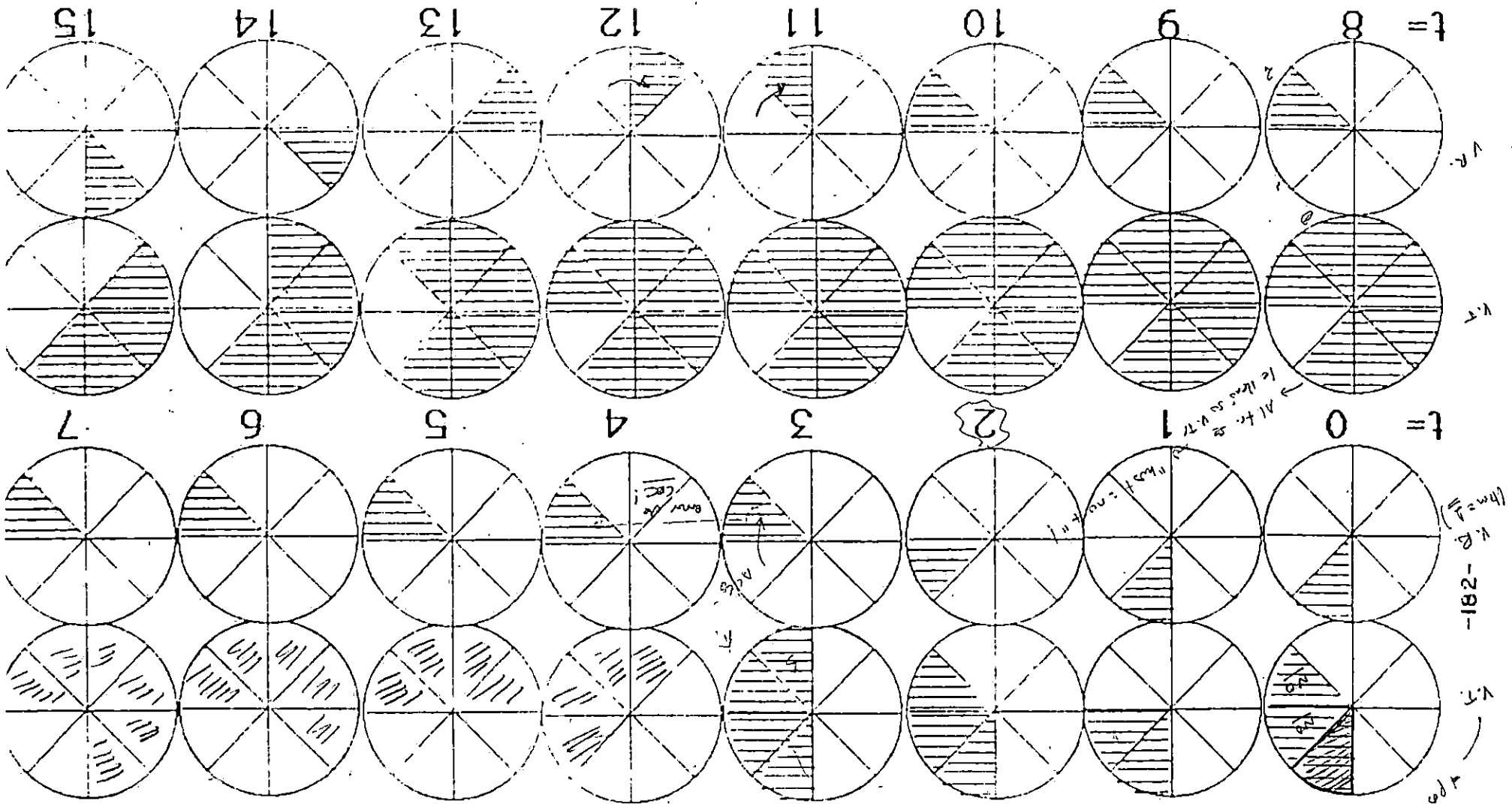
Un tick = 100mseg ($\frac{1}{10 \text{ seg}}$).

Hora actual:	+ sec	10:00:00.0	↑
Timeout frame 1		10:00:00.5	↓ $\Delta=5$
Timeout frame 2		10:00:01.3	↓ $\Delta=8$
Timeout frame 3		10:00:01.9	↓ $\Delta=6$



DESPUES DE MEDIO SEGUNDO





CSA

PROGRAM PIPELINEGBN;

(const
Lost Bit = ?
Eternidad = false
MaxSeg = ?

type

$\text{seq} = \text{d} \cdot \text{Max seq}$
 $\text{micro} = \text{packed array}$
 $\text{from} = \text{packed array}$

$\text{אף} = \text{מבטאים}$
 $\text{גז} = \text{מבטאים}$
 $\text{גז} = \text{מבטאים}$

[illegible]

procedure idoyv:elto.

For Airt Force 100:500000

from 5:30 to 6:00 PM

ՀԱՅԿԱԳԹ
[Կրճառագրության համաժողով] : Երևան

[Handwritten notes:]

VIII 007, 065 : 0913; 0914
VIII 008, 065 : 0915; 0916

event: EVTYPE

function between (a, b) : sequence : (a, b) : boolean.
 { returns true if $a < b$ and false otherwise ;

begin (7=70) (7>9) 7

between = true

2510f = 433.7739

5. $\rho u \partial u / \partial x$

Handwritten: "Handwritten: (Frame Nr.: Sequence Nr.):"

$$f(x) = \frac{1}{2} \log \left(\frac{1+x}{1-x} \right)$$
$$\begin{aligned} \text{Hypotenuse} + \text{Perpendicular} &= 70 \\ \text{Hypotenuse} - \text{Perpendicular} &= 6 \end{aligned}$$

... (Mars), ... (1977)

54101 + 10451 (100000)

1908

Consideraciones de Selective Repeat

- ^{mr} Receptor acepta frames en desorden
- Los pasa al host en orden
- Ventana receptora > 1

Cómo sabe qué frames de ventana ya llegaron?

Con arreglo de indicadores (bits)

- Cuando llega frame con $\#seq = \text{límite inf vent recep}$

Lo pasa al host

Pasa los que llegaron antes conservando orden

Protocolo #6 (S.R.).

- Solucionamos demora excesiva para piggyback?

Nuevo timer que limita la espera (StartAckTimer) ← el DIS_R del Host R

Si se vence enviamos ack sin datos al DR_R

- Se usan NAKs

- Si NAK se pierde

No hay problema pues el Transmisor hace Timeout (solo se diferencia en retr. rapidez).

- Si desviación del t de llegada del ACK pequeña entonces Timer ajustado y NAKs no útiles

Si

Sino t de llegada de confirm. y de.

Timer Amplio y NAKs muy útiles

- Frame que hace timeout no necesariamente es el #seq + bajo

Pudo ser retransm después de frames nuevos

nes. saber que frame hizo t.out.

- Simplificación:

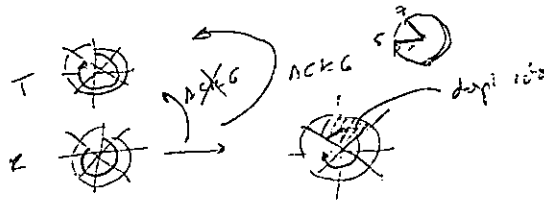
Después de un evento timeout

Mágicamente **oldestframe** contiene #seq

algun problema

enta!)

Si se vence el T.O.
se cancela el P.R.
de la h_k ⇒
pued. recibir en
después!



S.R. → aceptar pr. en orden por los pines al host en orden

Consideraciones (Cont)

- Problema: Ventana receptora "debe ser" $\leq \text{Maxseq} + 1$

Análisis.

Tenemos #seq de 3 bits (0-7)

"Supongamos vent de transm y recepc = Maxseq "

sup Transmisor envía frames 0-6 y

Ventana receptora está en 0-6 ;

y Los 7 frames llegan OK al receptor

⇒ Receptor mueve ventana a 7, 0, 1, 2, 3, 4, 5 ; sup. Ack's se perdón

⇒ Los 7 buffers se marcan vacíos

sup TODAS LAS CONFIRMACIONES SE PIERDEN

Transmisor hace timeout y retransmite 0

Recep chequea si está en vent y si buffer vacío

Las dos cosas OK

⇒ Lo acepta

Receptor repite ack de 6 (Ultimo recibido OK)

Transmisor avanza ventana 7, 0, 1, 2, 3, 4, 5

Receptor al recibir frame 7

Lo pasa al host junto con 0

Conclusión: FRAME 0 DUPLICADO

La razón:

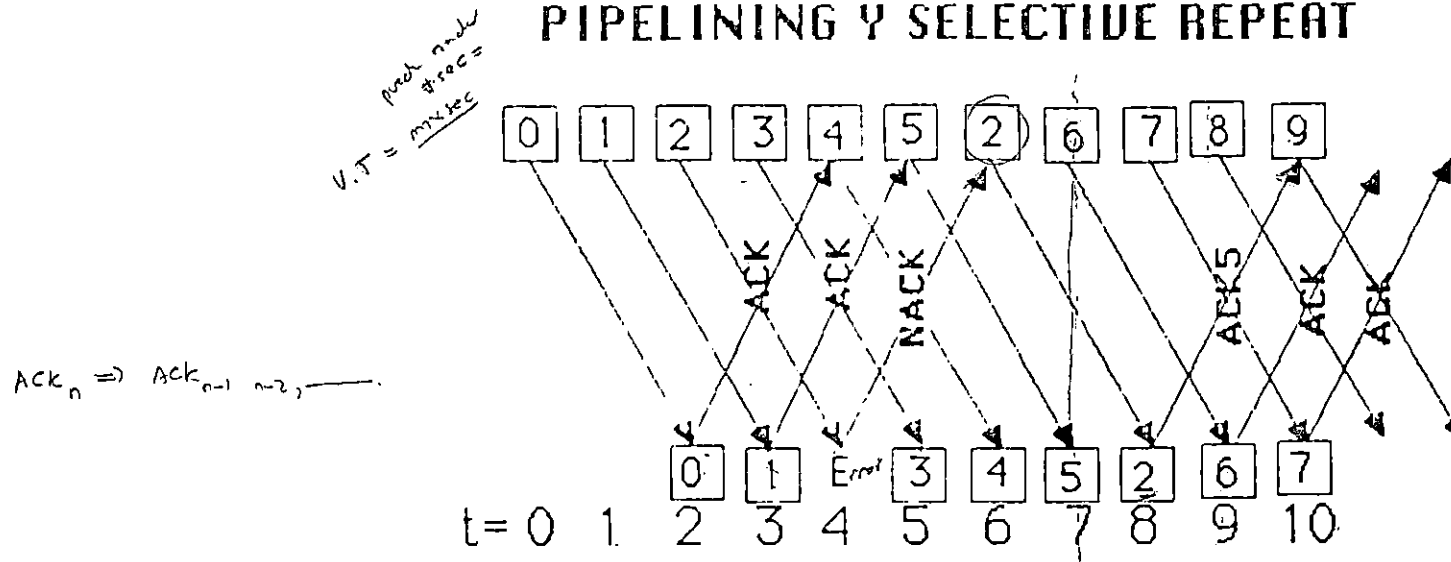
Traslapo de ventana receptora entre grupos

- # buffers necesario en receptor = Tamaño de vent
No al rango de #s de seq

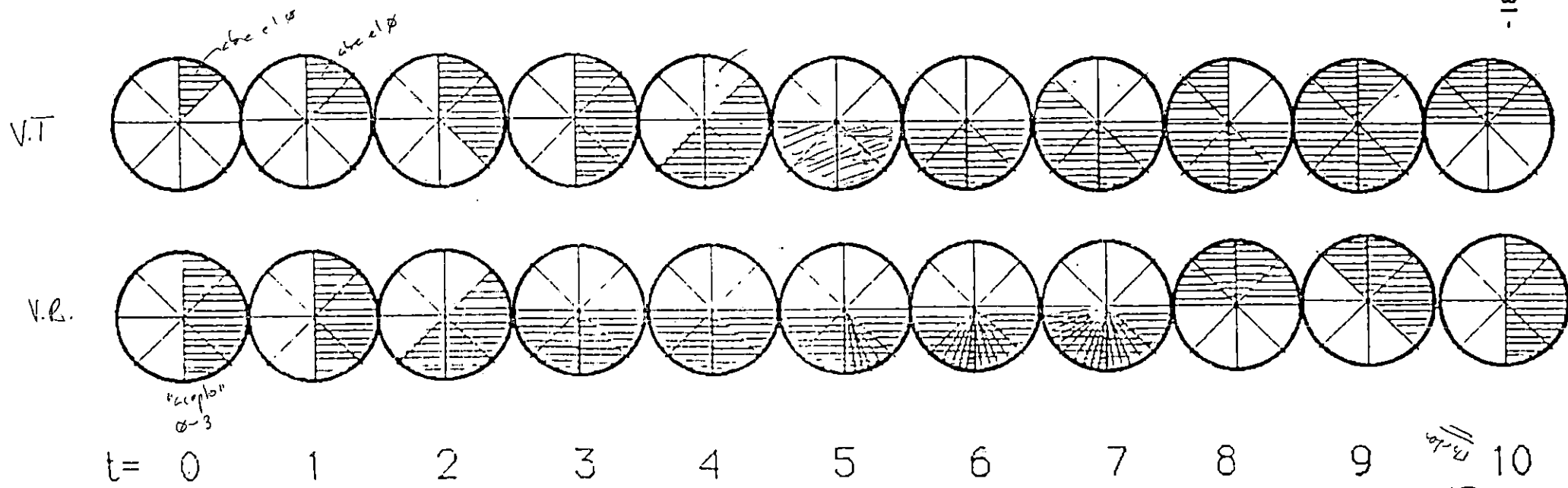
sup
→ porq' en el momento de recibir V.T. no hay #seq's antes de la rec. y el siguiente (host)

seq !

COMPORTAMIENTO DE VENTANAS EN SLIDING WINDOW CON PIPELINING Y SELECTIVE REPEAT



en el 5 recibir envia 5
 y recibe el 1
 envia #5 y recibe el
 slot 5 y
 en V.R.
 recibe 3, no es el 2
 inf.
 pero no recibe 2



Receptor
 #3
 1

LOGRAM PIPELINESK;

```

const LastBit = ?;
Eternidad = false;
MaxSeq = ?;
NrBufs = MaxSeq + 1 div 2; { # buffers = tamaño ventana
                             tamaño vent. transm = vent. recept
                             { # seq último buffer }
MaxBuf = NrBufs - 1

```

```

type bit = 0..1;
sequenceNr = 0..MaxSeq
mensaje = packed array [0..LastBit] of bit;
frameKind = (data, ack, nak); { posibles tipos de frame }
frame = packed record
    kind: frameKind;
    seq: sequenceNr;
    ack: sequenceNr;
    info: mensaje;
end;
bufnr = 0..MaxBuf;
EVTYPES = (FrameArrival, CNUMERR, TIMEOUT, HostReady, HostId);

```

procedure duplex;

```

var
    ArrExpected: sequenceNr; { Limite Inf. Vent. Transm. }
    NextFrameToSnd: sequenceNr; { Limite Vent. Transm. }
    FrameExpected: sequenceNr; { Limite Inf. Vent. Recpt. }
    TCFor: sequenceNr; { Limite sup. Vent. Recpt. + 1 }
    OldestFrame: sequenceNr; { Indica que frame hizo }
    i: bufnr; { Indica al pool de buffers }
    r, s: frame;
    errort: array [0..LastBit] of mensaje; { buffers del transmisor }
    errord: array [0..LastBit] of mensaje; { buffers de recepción }
    indr: array [0..LastBit] of boolean; { indicador de buffers de }
    { recepción ocupados }
    h: sequenceNr; { # de buffer de transm. de }
    { indica si no fue enviado }
    hien: boolean;
    event: EVTYPES;

```

```

function between (a, b, c: sequenceNr): boolean;
{ retorna cierto si  $a \leq b < c$  circularmente }

```

```

begin
    if (a <= b and (b < c)) or { Lim Inf < Lim Sup }
    ((c < a) and (0 <= b)) or { Lim Inf > Lim Sup - Sector Inf }
    ((b < c) and (c < a)) then { Lim Inf > Lim Sup - Sector Der }
        between := true; { Está en la ventana }
    else
        between := false; { No está en la ventana }
end;

```

[illegible]
$$= 1.0034 \pm 5.2\%$$

[Faint handwritten notes at the bottom of the page]

Incidivie indicodorus dohoffii

170000 200000 250000 300000 350000 400000 450000 500000 550000 600000 650000 700000 750000 800000 850000 900000 950000 1000000

1 - 128 7021 - 005 w, 17
2 - 12133 - 44211 - 445 w, 17

1. $3 = \omega(10.7 - 7.43) \cdot 7.05 \cdot 10^{-7}$
1. $10.7 - 7.43 = 3.27$

1. Procedural - 7-11-2011

only no extra demand for

correspondence!
Date of letter given 5/15/55

5! se esta mandando un timbre a la
Arquidiócesis del buffer!

Indep. ar. 505 ar. 505

1. (Cont.)

forma elementare del buffer (corresponding)

25/8/77

{ 2017 08 19 } { 2017 08 19 }

10/10/10

1. *Journal of the American Medical Association*, 1997; 277: 1001-1005.

1000
1001
1002

10/10/10

$$V_{\text{eff}} = 1.46 \times 10^4 \text{ V}$$

1. $\frac{1}{2} \pi \sqrt{\frac{1}{g}}$
 2. $\frac{1}{2} \pi \sqrt{\frac{1}{g}}$

[illegible]

01639
E 0011 H. 1.

1950年10月1日

(5) 9 1/2 mod

4347 7700 = 7.71

$1000 \div 100 = 10$
 $1000 \div 100 = 10$

4. $1000000 = 10^6$ (1000000) = 1000000

$$\therefore \text{True MI} = 5.55$$
$$\sin f_c = 60^\circ / 100^\circ = 0.6$$

$u_{34} f \quad o f o p = 7 f \quad f!$
 $(x_1) = 100117.5$

416399
1015105401

Reduce Scintiform. (f.k.)

Frame Arrival: { Llegó un frame sin resumen }

begin
getf(r);
if r.kind = dato then
Decibalo }

begin
if r.kind > frame Expected) and
if (r.seq < frame Expected) and

Alcivar then { sospecho que el frame es corrupto
if between (frame Expected, r.seq, TooFar) and
(arrived r.seq mod NrButs] = false) then
{ si frame recibido está en ventana receptora
y no es un duplicado ent }

begin { se puede recibir en orden
y no es un duplicado ent }

begin { se puede recibir en orden
y no es un duplicado ent }

begin { se puede recibir en orden
y no es un duplicado ent }

begin { se puede recibir en orden
y no es un duplicado ent }

begin { se puede recibir en orden
y no es un duplicado ent }

begin { se puede recibir en orden
y no es un duplicado ent }

begin { se puede recibir en orden
y no es un duplicado ent }

begin { se puede recibir en orden
y no es un duplicado ent }

begin { se puede recibir en orden
y no es un duplicado ent }

Assume:

If NoAck then

sendFrame (NAK, ϕ)

if lego frame molo

Timeout:

SendFrame (data, OldestFrame); if transmission of frame

give him time out
old frame expires
for event τ
if host did not receive frame
or receive frame y : key & identifier
ack x.f.f.

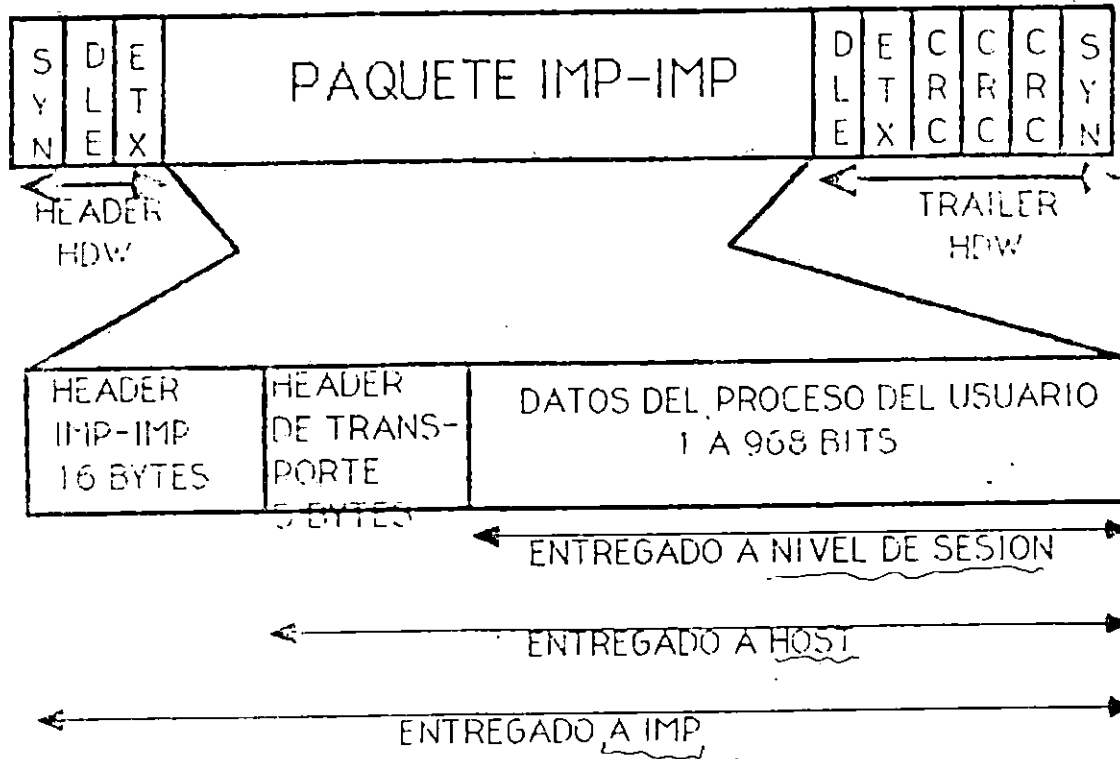
end if findel case

if $x.f.f. = A.F.f.$ then: 1.5! Version transm. 7/11/2003

if $x.f.f. = A.F.f.$ then: 1.5! Version transm. 7/11/2003
if $x.f.f. = A.F.f.$ then: 1.5! Version transm. 7/11/2003
if $x.f.f. = A.F.f.$ then: 1.5! Version transm. 7/11/2003

end if findel protocol

Arpanet - Formato del frame



Protocolo

- Orientado a caracteres
- Entre 2 paquetes HDW transmite syns
- Manejo de transparencia
 - Confusión si DLE ETX ocurre en datos
 - Si sucede, HDW inserta DLE adicional
- Si DLE ETX se daña,
 - Dos frames parecerán uno al recept
 - ⇒ CRC debe detectar el error
- No usa NAKs
- Período de timeout = 125 ms
- Usa sistema equivalente a Sliding Window de 3 bits (leer)

no pinta en los 3 bits

SDLC Y H.25

LAPB

→ protocolo SW.

IBM

CCITT

Parte de datos:

01111110	Dirección	Control	Datos	Cksum	01111110
----------	-----------	---------	-------	-------	----------

8

8

20

16

Header y trailer
del frame

lo impone el nivel de red
no el de enlace

Dirección: Útil en líneas multipunto

Control: #seq, acks y otros

Cheksum: Generado por polinomio estándar CCITT

6 CRC

Historia

- + 1970, IBM, SDLC^{inc. link control}
 - Usado en SNA
 - SDLC = Synchronous Data Link Control
- + ANSI lo transformo en ADDCP
 - ADDCP = Advanced Data Comm Cntrl Procedure
- + ISO lo transformó en HDLC
 - HDLC = High Data Link Control
- + CCITT convirtió HDLC en LAP
 - LAP = Link AccessProcedure
 - Modifico LAP a LAPB

Características técnicas

- Todos orientados a bits
- Transparencia con bit stuffing
- Flags Principio y final 01111110
- + Sliding window de tres bits
 - Ventana transm máx = 7
- HDLC y ADCCP son Selective repeat
- SDLC y LAPB son Goback n
- En ausencia de info se transmiten flags
- + Tres tipos de frames *además de los de datos.*
 - Información
 - Supervisión y control
 - No numerados

→ V.R. max $\frac{7+1}{2} = 4$

Receive Ready → confirm. sin datos.
 Reject → "NACK"
 Receive Not Ready → para el receptor hasta q' pueda recibir. (algunos problemas al probarlo: chaga, etc.).

≠: siempre se recibe en el ACK, al pñt #sac recibido estos vuelven al byte y están esperando

NACK por S.Repeat.

problem arose al probarlo: chaga, etc.).

Formatos del Frame de Control

bits

0	SEQ	P/F	NEXT
1	0 Type	P/F	NEXT
1	1 Type	P/F	Modif

Frames de Información

RR/R/RNR/SR.

Frames de Supervisión y chel

Frames no numerados

poll/finsh. poll → "¿quiere tr?"
 finsh → "cubí la tr." } podido.

x Características técnicas

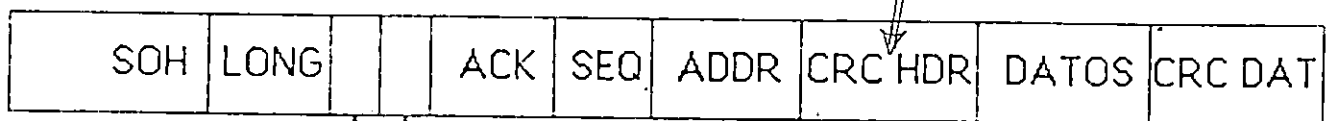
- Todos orientados a bits
- Transparencia con bit stuffing
- Flags Principio y final 01111110
- + Sliding window de tres bits
 - Ventana transm máx = 7
 - HDLC y ADCCP son Selective repeat
 - SDLC y LAPB son Goback n
- En ausencia de info se transmiten flags
- + Tres tipos de frames
 - Información
 - Supervisión y control
 - No numerados

No numerados

- + Desconectar
 - IMP anuncia caída (Mantenimiento)
- + Reconectar
 - IMP anuncia regreso
- + Cambio #seq
 - De 3 bits a 7 bits
- + Rechaza frame
 - CRC ok pero semántica imposible
- Acks de frames no numeados

Decnet

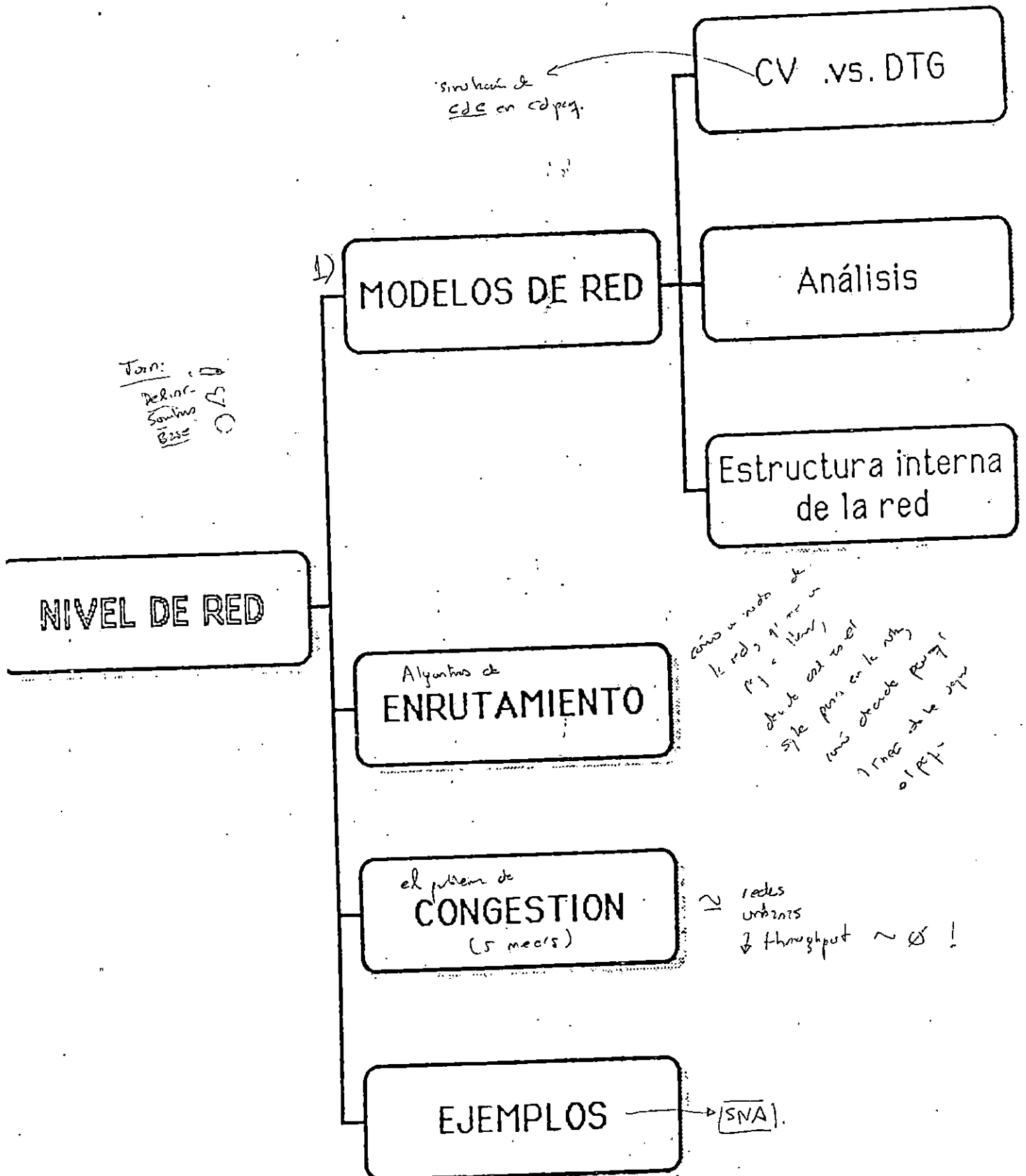
- + LLamado DDCMP
 - Digital Data Comm Message Protocol
- + Formato del frame
 - No termina con caracter de ctrl
 - Se incluye longitud en header
 - Header tiene su propio CRC
 - Sliding window con #seq de 8 bits
 - Long máx del pkt "16.383 bytes" (16k)!
- + Frames de control
 - ACK, NACK y REP *→ cuando se vence el T_{ro} , envia un paquete REP p/ saber si recibí uno y envia el byte correcto.*
 - REP evita retransmisiones de frames largos



↑ ↑
 SYNC SELECT (Δ sentido):

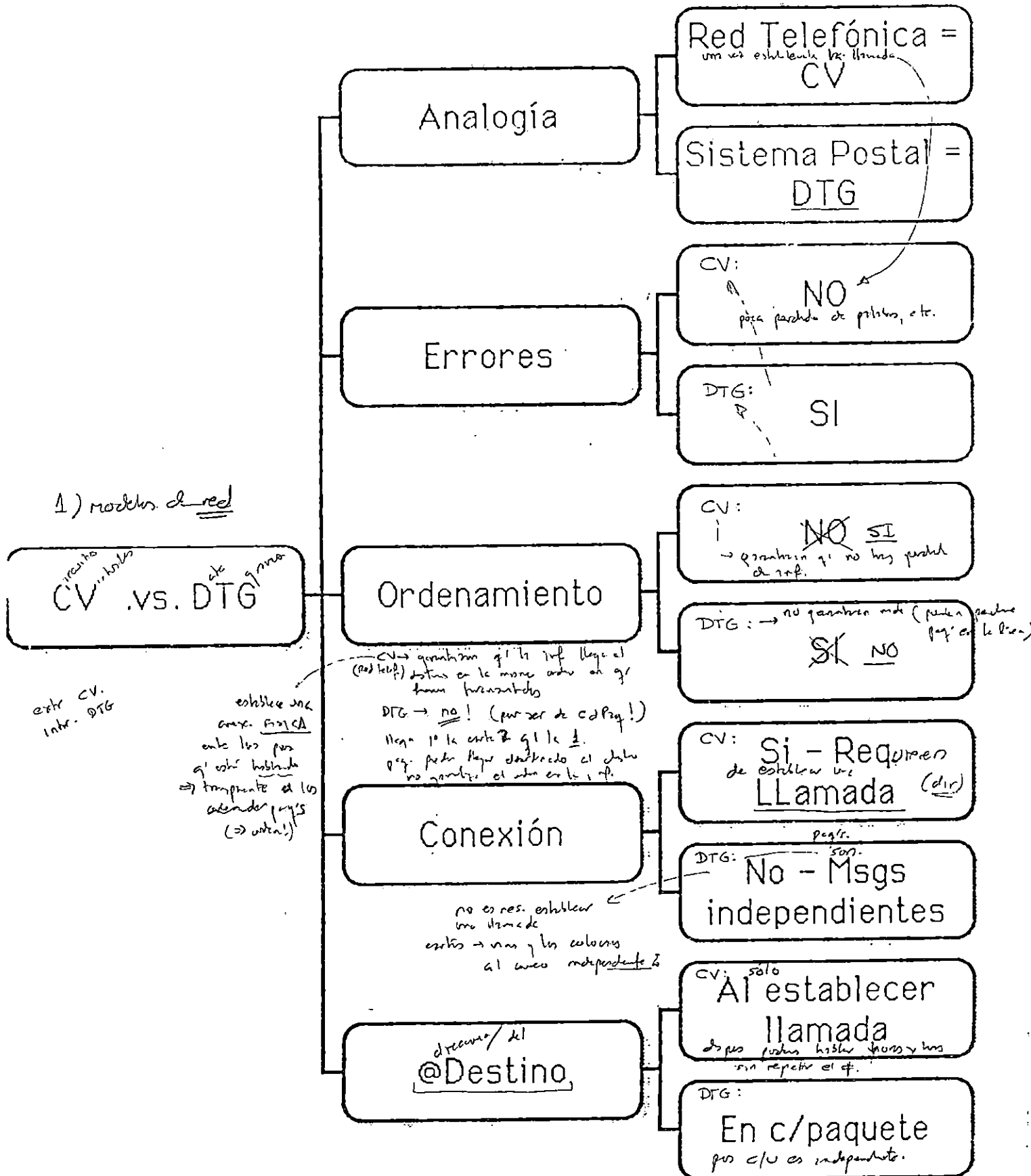
SELECT: Controla Cambio de sentido en HD y multipunto

ADDR: Dirección en multipunto



redes en CV → simulan la com. de env. en redes de Collg. ← OTO
 ↓ % de uso de la línea c. a. b. de consumo de la línea
 → computar las lms de com. entre vnos usuarios.
 US → env. a paquete

1) modelos de red



Throughput $\rightarrow$ t p/p/ un de t q' llega a la red
 ongo $\rightarrow$ " " " " " " " "

Análisis

+ 1^o DTG ⁵⁰⁰ más primitivos

$\Rightarrow$ nec. de un

- Nivel de transporte + complejo

para debe clasificar en el N de Red No tiene, dado q' el mismo es "falta" los puntos de orden, etc.

+ Ordenamiento: puede ser ventaja o no

- Ej: Backup físico ^{en CV} ^{desventaja se tener orden en paquetes} ^{copy "remoto"}

+ Manejo de errores puede ser ventaja o no

- Ej. Voz digitalizada ^{pequeñizada} ^{desventaja!}

- En tráfico "interred" CV pierde sus ventajas

+ Tráfico de tiempo real ^{CV} ^{DTG} ^{USA} ^{Ch.} ^{Inglaterra}

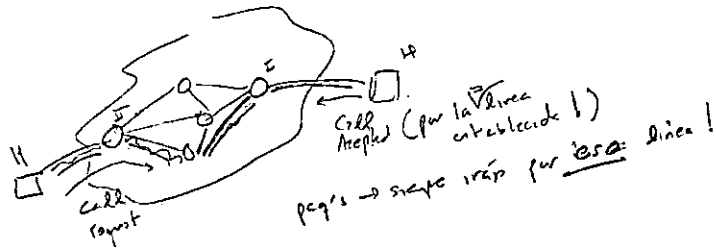
- Costo de establecer llamada >> Datos ^{Costo de envió}

CV $\rightarrow$ quiere orden prop. 7 q' 7 hay errores

Ej. como autoshop

Se pierde el W hecho por los eds de CV en la granja/ de orden y del de un p' q' en una red interred (LPRC) no se tiene en cuenta.

X25 $\rightarrow$ 4 p' de ctrl:
 2 p' establecer la línea
 (Call Request y R
 Call Accepted)
 y 2 p' "colgar"
 (Clear Request y R.
 y Clear Confirmation)



Estructura interna de la red

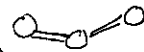
pl. repór. un CV. (Ed Dts)

CV: Una vez establecida la llamada, a nes.

⊖ Pckts de una llamada usan la misma ruta

+ C/IMP mantiene una tabla (EdD)

pl. repór. (lógica) que cuando está establecida



perm. de repór
la com.
(cambio de servicio)

Y ^{Dicha} ^{hacia} Una entrada por C/CV que lo involucre

Un IMP puede tener muchos CVs

+ C/entrada de la tabla tiene 2 secciones:

①+ Sección de llegada

De q' ^{IMP} Máquina de la que viene

- Número de CV

②+ Sección de salida

CV no res. dir. y remite (destinatario)
sin f. CV.

- Máquina siguiente en ese CV

- Número de CV

- Todo paquete debe llevar # de CV

Una vez establecida la CV, no hay res. de dir. en C/IMP, la @ de pckts de la fuente y la del destino; basta con enviar el # CV.

+ Para establecer un CV,

- el Host busca ^{su} ^{IMP asociado} # CV no usado en esa máquina

Estructura interna de... (cont'd)

+ Cómo evitar que 2 Hosts usen ^{el MISMO} # de CV?

- No se evita _{El # CV es atributo a lo largo de la ruta}

- Se evitan #s repet entre parejas de máq ^{IMP's}

- ^{el} # de CV no se conserva en toda la ruta
_{pero es posible entre q'z hosts concurrentes ocupen el # de CV}

+ En cada paso una máquina busca el

- # + bajo no usado con su compañera

+ Cuando un pckt llega a un IMP, este

- Busca en la tabla #CV del próx paso

- Cambia #CV al pckt

+ Procesos deben cerrar CV <sub>un p'q. para en c/step
p' ser reabriendo.</sub>

^{q'!} - Se cobra algo por t de conexión, p' entrar q' se cobra
_{la mem de la step}

- En DTG no se necesitan estas tablas

_{modo} - DTG debe contener @ en c/pckt

elección
compartido
AB-Mem:

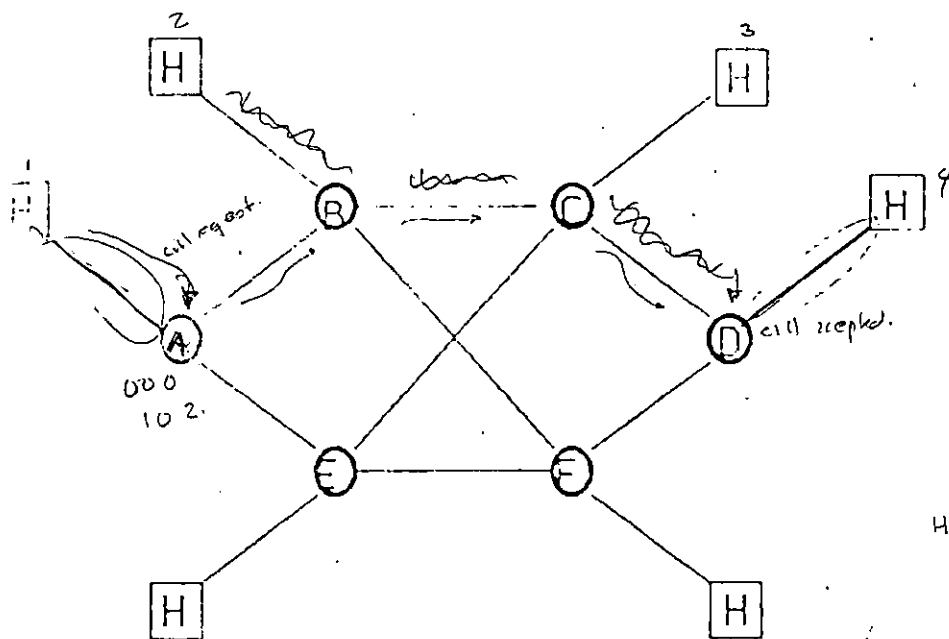
CV → gestión + rem. ^{los steps misma} _{pus real los libro}; pero req < AB. (solo p' el # CV!)

DTG → " - " _{pus no leen cons libro; pero req. > inf. de chrd ⇒ > AB}

nes. @ Punto y @ destino en c/paquete!

gde.

Como Pn CV :



① 000 ④

② 01 ⑤

INICIADO
EN A

INICIADO
EN B

ABCD

BCD

AEFD no se arm. con D

BAE

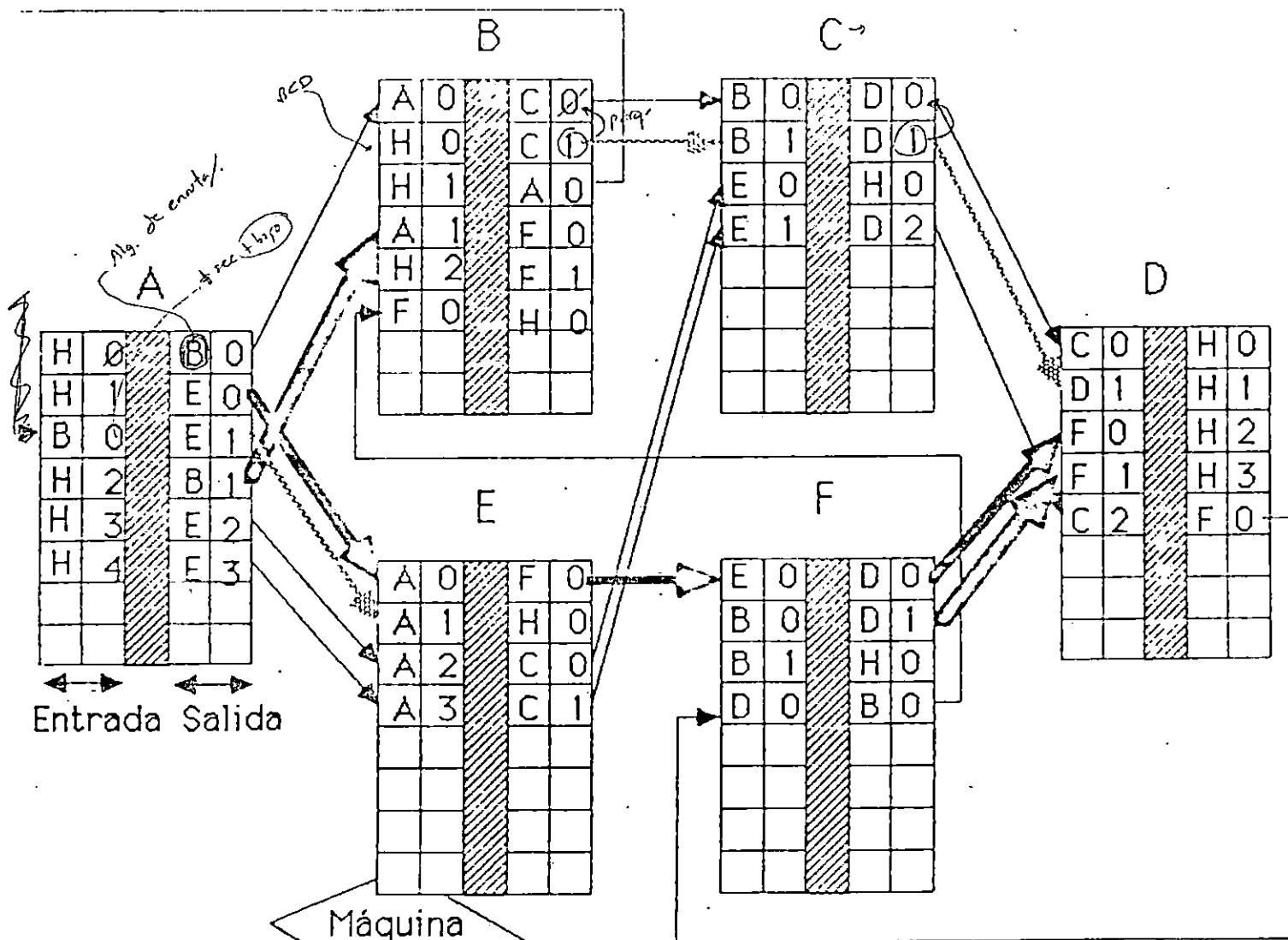
ABFD

BF

AEC

AECDFB

Hut → "esary" el CV



Entrada Salida

Máquina
de Origen

Circuito
Virtual de Entrada

Estructura interna de... (cont'd)

- + Implicaciones elección CV o DTG
 - Compromiso memoria y AB
- + CV son ^{< flexibles} más vulnerables ^{q los DTG}
- + Al caerse IMP
 - Todos los CV que lo cruzan abortan
_{→ no son "perfectos"!}
- + Con DTG solamente sufren
 - ^{los} Pckts en tránsito por ese IMP ^{en ese momento}
 - ^y Confirmados
 - ^{pero no se retransmiten} No retransmitidos
- + DTG permiten mejor balanceo de la red ^{c/c. check}
 - Enrutamiento dinámico ^{se adapta al estado de la red (congestión, fallo, mantenimiento de rts)}
- Interfaz a usuario independ de estruct interna

INTERFASE PRESENTADA

al usuario.

		CV	DTG
ESTRUCTURA INTERNA	CV	SNA	No Usual
	DTG	ARPANET _{← x.25}	DECNET

-204-

termina

interf DTG
interf CV

→ Interf

2)

ENRUTAMIENTO

- + Objetivo General
- + Objetivos específicos
- + Clasificación

Objetivo General

de un sly de enrut en un xrp → al recibir un pkg,

En un IMP

*pe de decidir por q
línea sacarlo.*

Al recibir un paquete

Decidir por qué línea sacarlo

Objetivo General

+ En Datagramas

El enrut/
- es independiente por ^{c/}pckt y ^{por c/}salto
En c/xrp → solo se decide el sigle paso.

+ En Circuitos virtuales

El enrut/ es
- De sesión → *tipo y toda la llamada/correspondencia.*

Objetivos específicos

de los alg's de enrutamiento

- Adaptarse a cambios de topología
(saber cual es el p, cual es el q)
- Adaptarse a cambios de tráfico
Tener info del estado de los routers, del tráfico en la red, etc.
- Buscar equilibrio de la red

con ejemplos de
distribución
de routers.

- Minimizar t de demora de un pkt
+ Maximizar throughput de la red

Contradictorios?

Buscar balance entre los 2

> demora

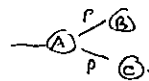
llegar a la sf
red en
2 ASM.
red. por media
10 ASM
red. por media

Clasificación

- + No adaptables al
Algo'so tráfico, topología.
- + Adaptables

No adaptables

- + Inundación
- + Estáticos



Lo manda por todos ptes por donde puede
p' ver quien llega 1º).

Inundación : 1º 2º. de envíos 2º de fuera bñh

Todo paquete que entra a un IMP se retransmite por todas las líneas excepto por la que llegó
(sin importar como se llega al destino)

Inundación

Siempre habrá la retransmisión.

Poner un pag
o/ver se van
generando más y más
paquetes copias de ese
paq.

Se nes. un
+ Control de inundación

+ Contador de saltos = lo p'gnto q' se neces.

- Idealmente contador = Dist al destino

- Usual poner el peor caso = ∞ de la red

- ∞ = Geodésica más larga

- Geodésica = Camino + corto entre 2 nodos

+ Números de secuencia

- IMP fuente coloca #seq

- C/IMP lleva #seq > visto de C/IMP fuente

+ Inundación selectiva

- Pckts no se envían por todas las líneas

- Sólo aquellas que van aprox en @ OK

+ Utilidad del algoritmo : Unican en la tr. de m. enge

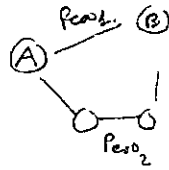
- Aplicaciones militares (Arpanet)

- BD distribuidas con gran redundancia

+ Comparación con otras técnicas

- Inundación ideal en t. de entrega

es. al. de entrega
1º más p' unos steps (Brednet)



Estáticos

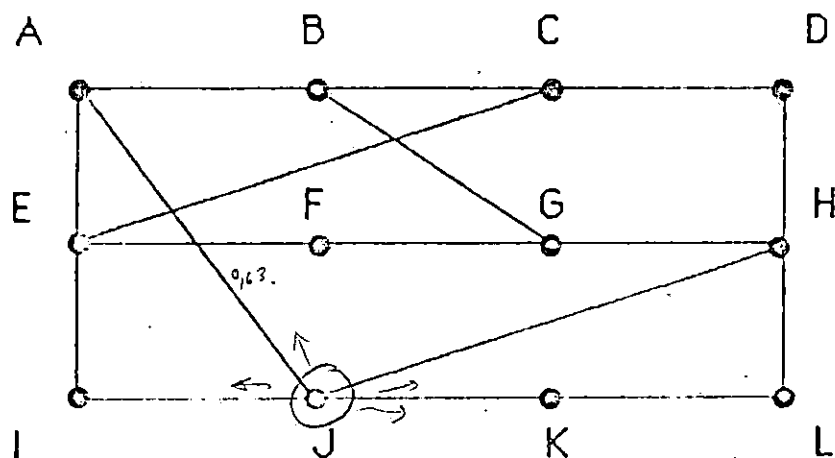
(SNA).

- Muy usuales y comunes por ser usadas en SNA (OSM)
- + Operación
 - + Usa una tabla
 - Creada manualmente por los operadores de la red
 - Una entrada por c/destino posible
 - C/fila ^{entrada} tiene mejor opción, 2a, 3a, etc
 - Cada opción tiene un peso, ^{proporcional a la línea que sea la más}
 - + Para escoger la ruta, ^{algoritmo}:
 - Genera #aleatorio $\infty < \text{peso} \text{ (entre 0 y 1)}$
 $\sum \text{peso} = 1$
 - Entra a la tabla
 - Busca rango
 - Escoge línea correspondiente

La que opina fue el $> \text{peso}$.

def → balanceo / equilibrar la red.

ENRUTAMIENTO ESTÁTICO



DES- MEJOR 2a.MEJOR 3a.MEJOR
TINO OPCION OPCION OPCION

de enlaces p. NO
cargados la red.
(62/64 de la red)

Ring 1

enlaces. > 0.84
⇒ pw H
0.63 y 0.84
⇒ pw I

			0.64	0.84	0.95	
A	A	0.63	I	0.21 (3 p. 160)	H	0.16
B	A	0.46	H	0.31	I	0.23
C	A	0.34	I	0.33	H	0.33
D	H	0.50	A	0.25	I	0.25
E	A	0.40	I	0.40	H	0.20
F	A	0.34	H	0.33	I	0.33
G	H	0.46	A	0.31	K	0.23
H	H	0.63	K	0.21	A	0.16
I	I	0.65	A	0.22	H	0.13
J	-	-	-	-	-	-
K	K	0.67	H	0.22	A	0.11
L	K	0.42	H	0.42	A	0.16

$\sum u_i = 1.$

Algo: Arquitectura:

Centralizado

+ Operación

- Basado en tablas similares a estáticos

- Existe RCC <sup>ultracorto
extremo
cortado</sup> ^{nada especial de gestión}

sin. al prob. de env. en

→ C. de Envío a Sub. en

+ C/IMP envía inf periódica a RCC

subre:

- ^{sur} Vecinos activos <sup>pe. otros que no están
correctos con él</sup>

- Longitud de ^{las} colas <sup>se por separado ser los
19 h.a. por el área</sup>

- ^{y del} Tráfico ^{procedido por el} línea desde último reporte

⇒ RCC construye tablas ^{óptimas}

- La envía a todos los IMPs

+ Para adaptarse al tráfico, el alg. debe

- Se corre ^{ve} cada $\approx$ seg $\rightarrow$ <sup>genera mucho tráfico
en la red!</sup>

+ Para adaptarse a cambios de topología

- Se corre $\approx$ minuto

(cont. control de
impis)

Centralizado (cont'd)

+ Ventajas

- Se dispone de toda la información (RCC)
- Excelentes decisiones de "rutage"
- IMPs descargados del trabajo

+ Desventajas

+ Debe correrse con frecuencia

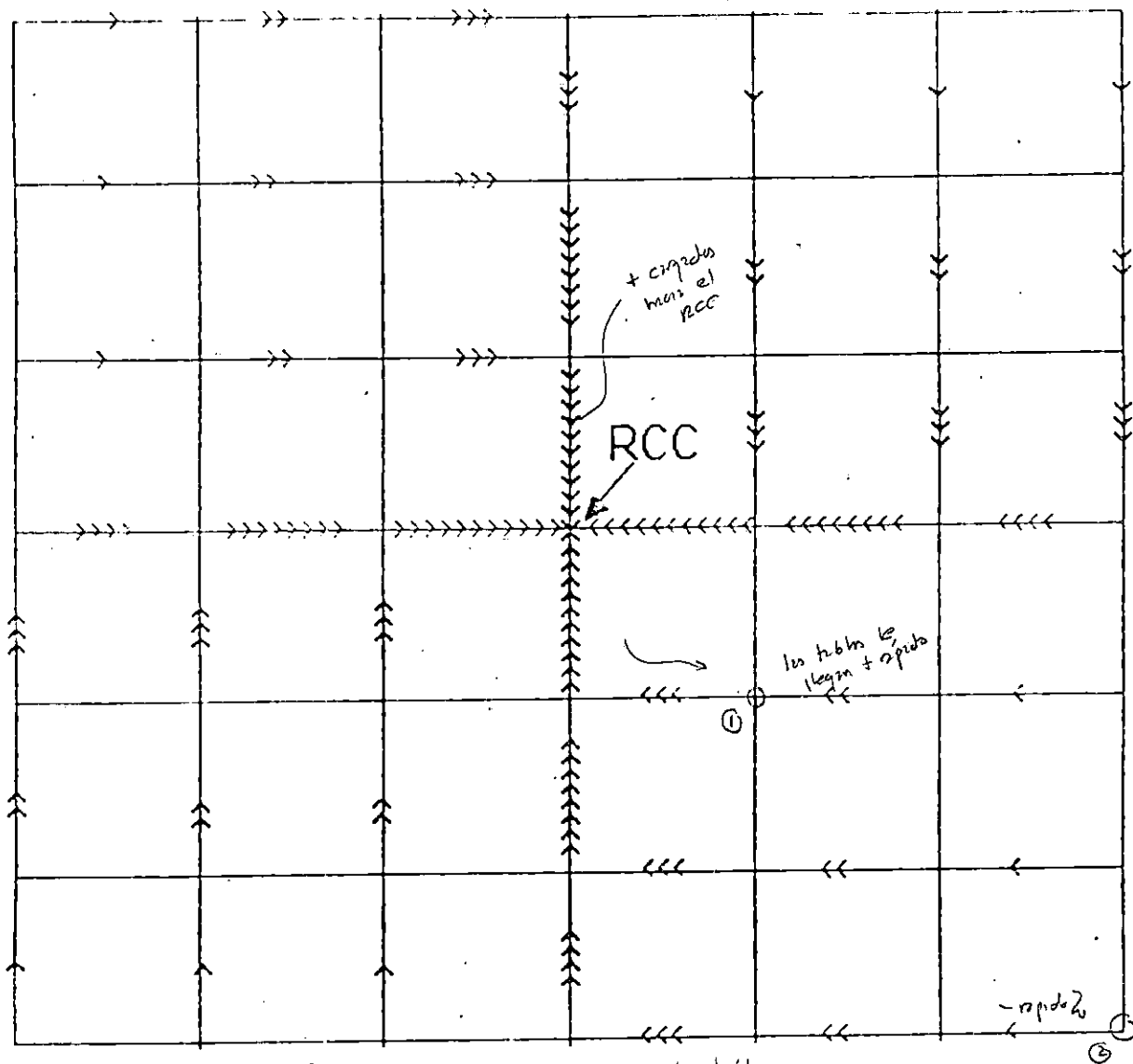
- Fuerte carga de proceso ← gran carga de la red.

+ Vulnerabilidad del RCC

- Fallas propias
- Fallas de sus líneas
- Sincronización en distribución de tablas
- Concentración del tráfico cerca a RCC

testeo al RCC son muy activos/cargados

unos llegan + repete q' otros



Puede ser funciones con diferentes estados de la table

Alq. de envh/
Aislados
 C/IMP hace lo q' quiere ; desde p'ro p'p'io ante la red

- + Filosofía
- + Papa Caliente
- + Mezcla de estático y aislado
- + Backward Learning
- + Enrutamiento delta

Why rec !
Filosofía C/IMP decide y es independiente.

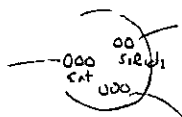
- Deciden con info manejada por C/IMP
- No intercambian info con otros IMPs
- Adaptables a cambios de tráfico
- Adaptables a cambio de topología

Papa Caliente

"a Pierre bntz"

- IMPs sacan pckts lo + rápido posible
- Los envían por línea de cola ^{de buffers} + corta
- No miran a dónde va la línea

real pg
pna (9-5
IMP).



Mezcla de estático y aislado

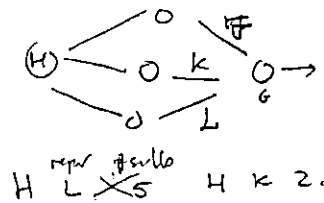
+ Considera ^{"mexote"} ⁽¹⁾ peso de línea y ⁽²⁾ long de su cola

+ Opciones :

- Usar mejor opción estática si cola ^{#pg. en cola.} $< X$
- Usar ^{ln} cola + corta ^(pg. en cola) si peso $\geq X$

+ Calificar líneas por peso ponderado

- Peso ponderado = Peso Est + long de cola



come el 2lg. rep. para H $\rightarrow$ 3

llega 1 paquete $\rightarrow$ 2

1 rep. es H

llega inf. de los pqs. q le entran

Backward Learning

Saber cómo está la ruta

tomar decisiones
en base en la
inf. de q' pqs
si entran

- Usa información de pckts que llegan

+ C/pckt incluye

prop. al list q' entra al paquete

- Número del IMP fuente (orig. de la pte).

- Contador de saltos (costo de la pte).

+ Si llega pckt de IMP I por línea K

- se examina su contador de saltos

semp
che 215 con la pte
y si se le quita en
la pte anterior?

+ Si $<$ anterior de I ent

- Mejor línea para $I = K$

pm $\rightarrow$ + Problemas:

- No reacciona a malas noticias

Si se cree un error
de esa línea?

- Qué pasa cuando la mejor ruta empeora?

+ Solución

- Reinicializar info periódicamente

esperar a
cambiar de
nuevo.

+ Si se hace muy frecuentemente

- Mucho enrutamiento de baja calidad
(con mucha inf.)

+ Sino

- Adaptación lenta

(considerar en vez de mala)
a los cambios y de topología.

Enrutamiento delta

- Idea: ^{enr/h/15}
- Mezcla ^{de} centralizado y aislado
 - RCC recibe información de IMPs
 - Para todo i, j ^{prop. de f. y destino:}
 - Calcula k ^{mejores del alg. de k caminos} mejores caminos de i a j
 - $C_{ijn} =$ ^{suma} ~~nava~~ mejor opción entre ij
 - + Si $C_{ijn} - C_{ij1} < \epsilon$ ^{oh padreño.}
 - $\Rightarrow$ camino n ^{equiv. a 1} equivalente a 1 (el repetido!)
 - ^{lujos} RCC maña lista de caminos equiv. a IMPs
 - + Basado en ^{su} información local ^(ver. la long. de camino de k caminos)
 - IMP escoge uno de ellos
 - + Ajustando K y ϵ se hace más centraliz o aislado
 - + ϵ pequeño deja poca elección a IMPs $\rightarrow$ RCC
 - Más centralizado
 - + ϵ grande genera muchos caminos "equivalentes"
 - Más aislado ^(lo de se toma local por el BR)
 - Si K grande, más ^{posibil} de caminos equiv
 - Con K y ϵ bien escogidos da buenos resultados

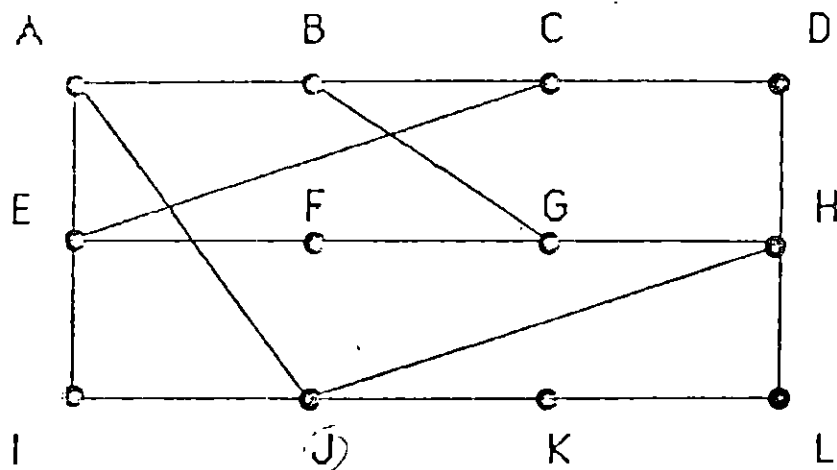
Distribuidos

Argument
Decret Proc 5
t elogiato

- # IMPs intercambian información con vecinos
- IMPs manejan tablas
- ^{q' hacen} Una entrada por cada IMP de la red
 - Línea de salida preferida para ese destino
 - Estimativo de ^{de demora} t o distancia ^{si hay a ver nudo por en línea}
 - El estimativo puede hacerse en
 - Número de saltos
 - t de demora estimado
 - #pckts en la cola
 - AB disponible → ^{Vel salida} # paquetes
- ^{para} C/IMP ^{que} conoce distancia a su vecino
 - Si # de saltos ent → 1
 - Si demora ent eco ^{no de paquetes} ^{si conocer el b d esp.} (det. la dist. ^{porq' puede q' no estan sincer. 2 entrs.})
 - ~~Ejemplo usando demora~~

EJEMPLO UTILIZANDO DEMORA

Ala de EVOLTA D.T.C.R.



Vecinos de J:

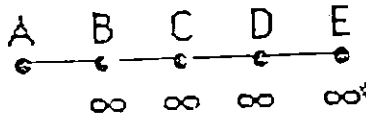
demora a c/u de los posibles!

	A	I	H	K	Nueva	
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25 <u>33</u>	18 <u>28</u>	19 <u>31</u>	36 <u>42</u>	28 <u>28</u>	I
D	40 <u>42</u>	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17 <u>25</u>	20 <u>30</u>	0 <u>12</u>	19 <u>6</u>	<u>12</u>	<u>H</u>
I	21	0	14	22	10	I
(J)	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K

$8 + 24 = 32$
 $10 + 20 = 30$
 $12 + 20 = 32$
 $6 + 11 = 17$
 $8 < 30 < 32 < 17$
 Demora de J a c/u. (Vecinos)
 rev I.

de J por I = C
vía A = 33 ms
vía E = 28 ms
vía H = 31
vía J = 42
por I
por I = 28 ms
por J = 42 ms

REACCION RAPIDA A BUENAS NOTICIAS



1^{er} ∞ ∞ ∞

1 2 ∞ ∞

1 2 3 ∞

1 2 3 4

A caído. ⇒ Todos lo saben

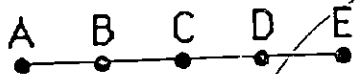
A Ok. ⇒ 1^{er} Intercambio. B informado

2^º Intercambio. C informado → esto es 2 pasos de A

3^º Intercambio. D Informado → " 3 "

4^º Intercambio. Problema Superado

REACCION LENTA A "MALAS" NOTICIAS



1 2 3 4

3 2 3 4

3 4 3 4

5 4 5 4

5 6 5 6

7 6 7 6

7 8 7 8

Todo OK

A se cae ⇒ Intercambian Info

Primera noticia a D

Primera Noticia a E

La info no se da a la vez a la red

después de 7 A está caído

Operación

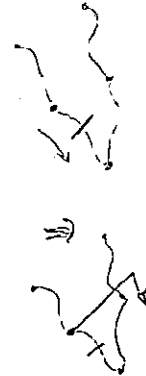
Funciona / del Dlg de CHU

no entrar con
"vs la consola"

Regla de CHU → Tener

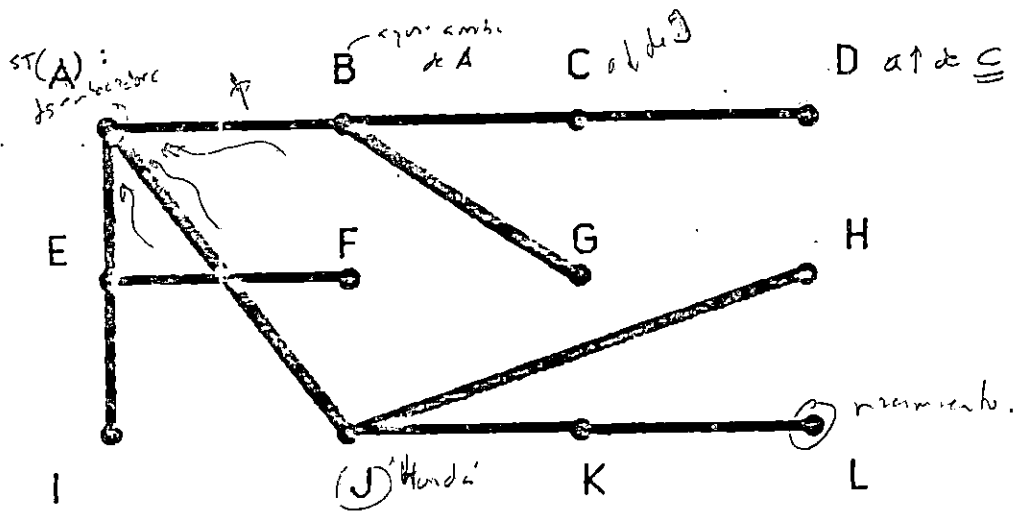
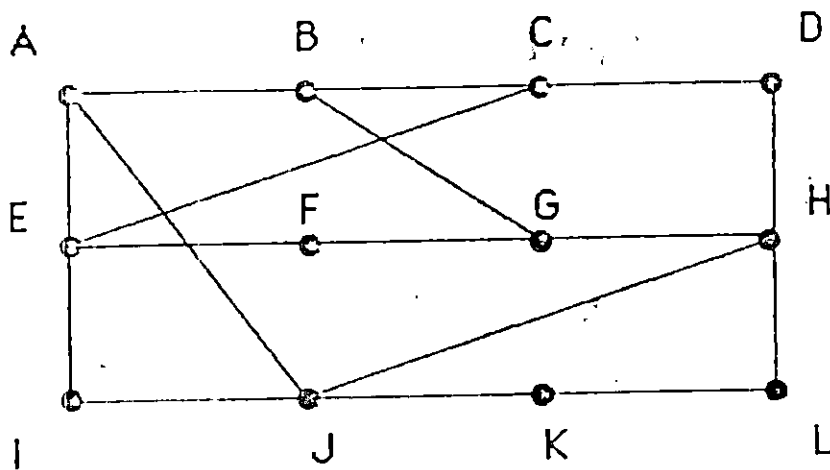
- c Si se cae una línea
- c IMP no puede mandar tráfico aguas arriba
- c Debe buscar otra rama del sink tree
- c Se mantiene una tabla
 - c Una fila por destino
 - c Una columna por línea *de salida*
 - c Cada elemento significa
 - c Para el destino
 - c Por la línea de salida
 - c # de saltos
 - c Mejor ruta encerrada en círculo
 - c Sink tree diferente para C/fila
 - c Elementos aguas arriba en blanco
- c Análisis de nodo G
 - c Qué pasa si línea GH falla?
 - c Columna de línea H no usable
 - c Busca rutas óptimas perdidas (E y H)
 - c Busca ruta alternativa para esos destinos
 - c Para E hay alternativa vía D
 - c Para H tenemos problemas
 - c Pide ayuda a vecinos

línea
kps ↑



EJEMPLO DE SINK TREE

= $\text{facto } CEA < CEA$
 shing new

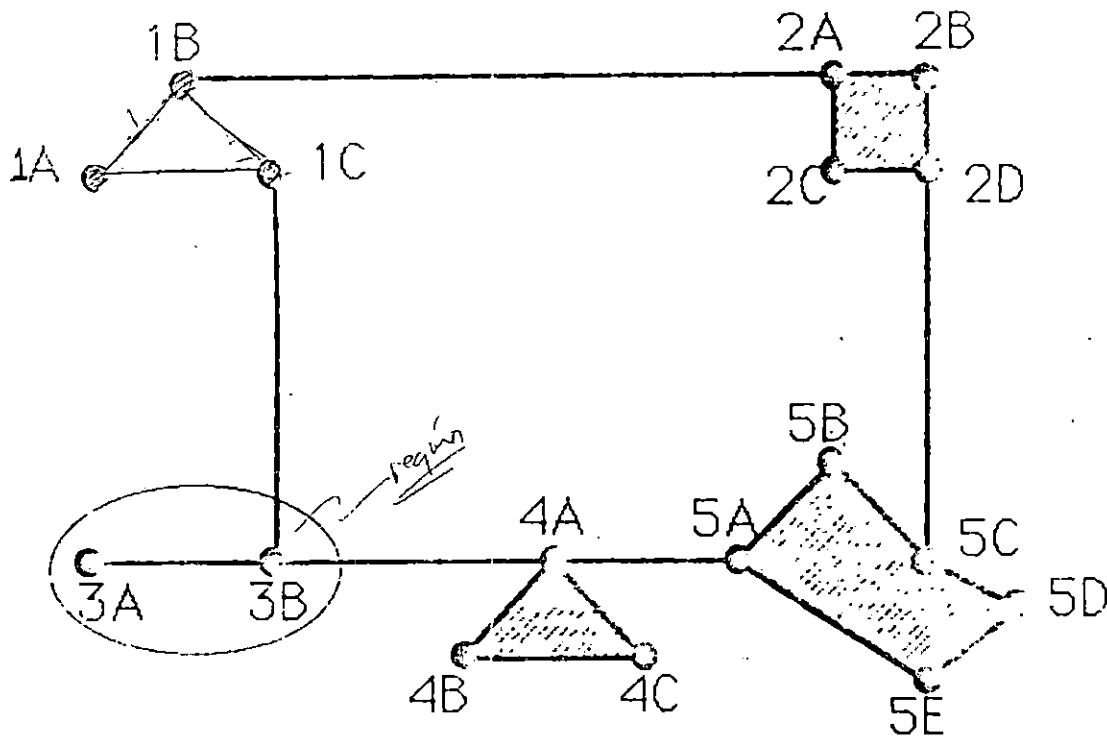


"Aguas f. & Aguas d"

con red => línea de enrutamiento y los

ENRUTAMIENTO JERARQUICO

la red se divide en regiones
propósito: reducir el tamaño de la tabla red. y el enrutamiento.



Dest Lín Salt

1A	--	--
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Dest Lín Salt

1A	--	--
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Además región 2 se divide por la línea

Tabla jerárquica para 1A

se reemplaza por "substitución"

Tabla plana para 1A

El enrutamiento jerárquico puede generar ineficiencias
Ejemplo: 1A-5C (así como red)

Heurística: tener "la n" muelles => el nro. enlaces por el grupo

Enr/ Broadcast

- 1^a parte
- ^{Alternativo} Mandar ^{A=} pckt a C/IMP! ← ineficiente! { dep. req.
 - ^{Alternativo} Inundación
 - Enrutamiento multidestino
 - Algoritmo basado en el sink tree
 - Reverse path Forwarding

para ver si se puede por un nodo
alguno de los que se pegan

de AB

q' a the. tenga
lista completa de
IMPs →

Mandar pckt a C/IMP

- Desperdicia AB
- Fuente necesita lista de IMPs

Inundación

- Candidato bueno
- Genera muchos paquetes
- Consume mucho AB

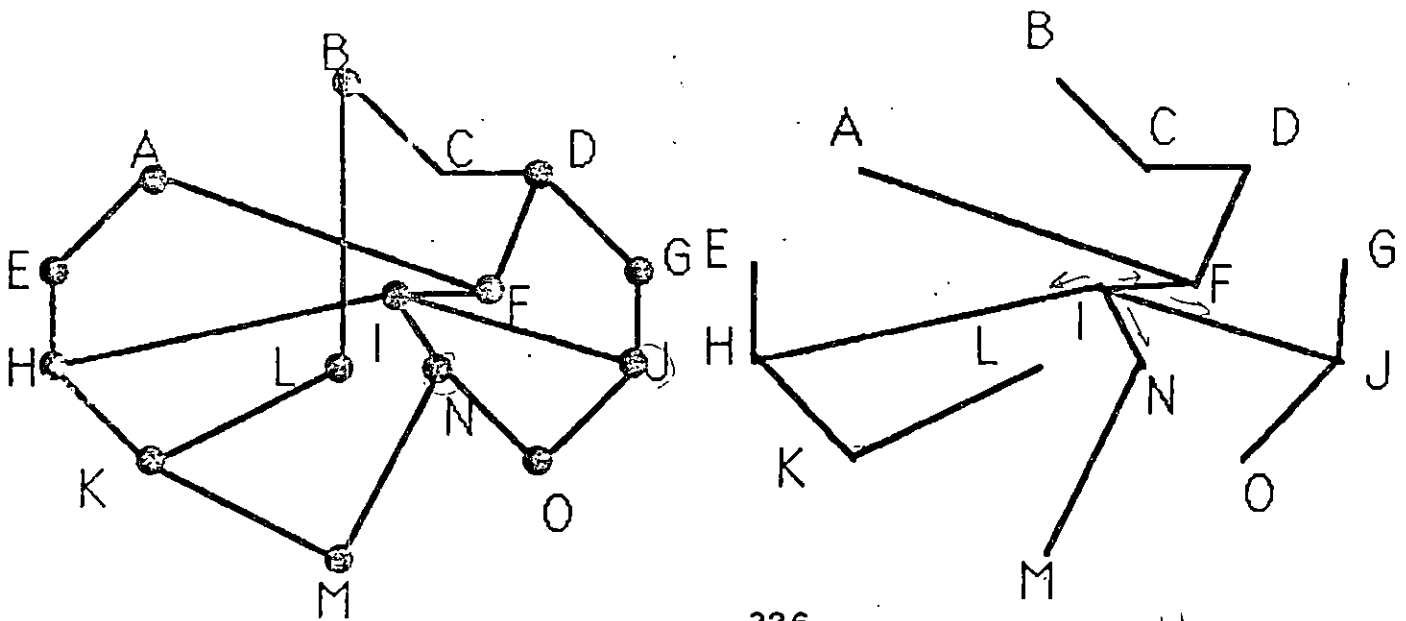
Enrutamiento multidestino

- C/pckt lleva lista de destinos
- C/IMP examina lista de destinos
- Genera copia por c/línea necesaria
- Parte lista de destinos

ej. - res. consistir la lista de destinos, y la parte de la red.

Algoritmo basado en el sink tree

- IMPs conocen sink tree de la fuente
- Retransm por todas ellas ^{excepto} por la que llegó
- Optimiza AB generando mínimo de pckts
- Desventaja: Necesidad de conocer ~~el~~ sink tree

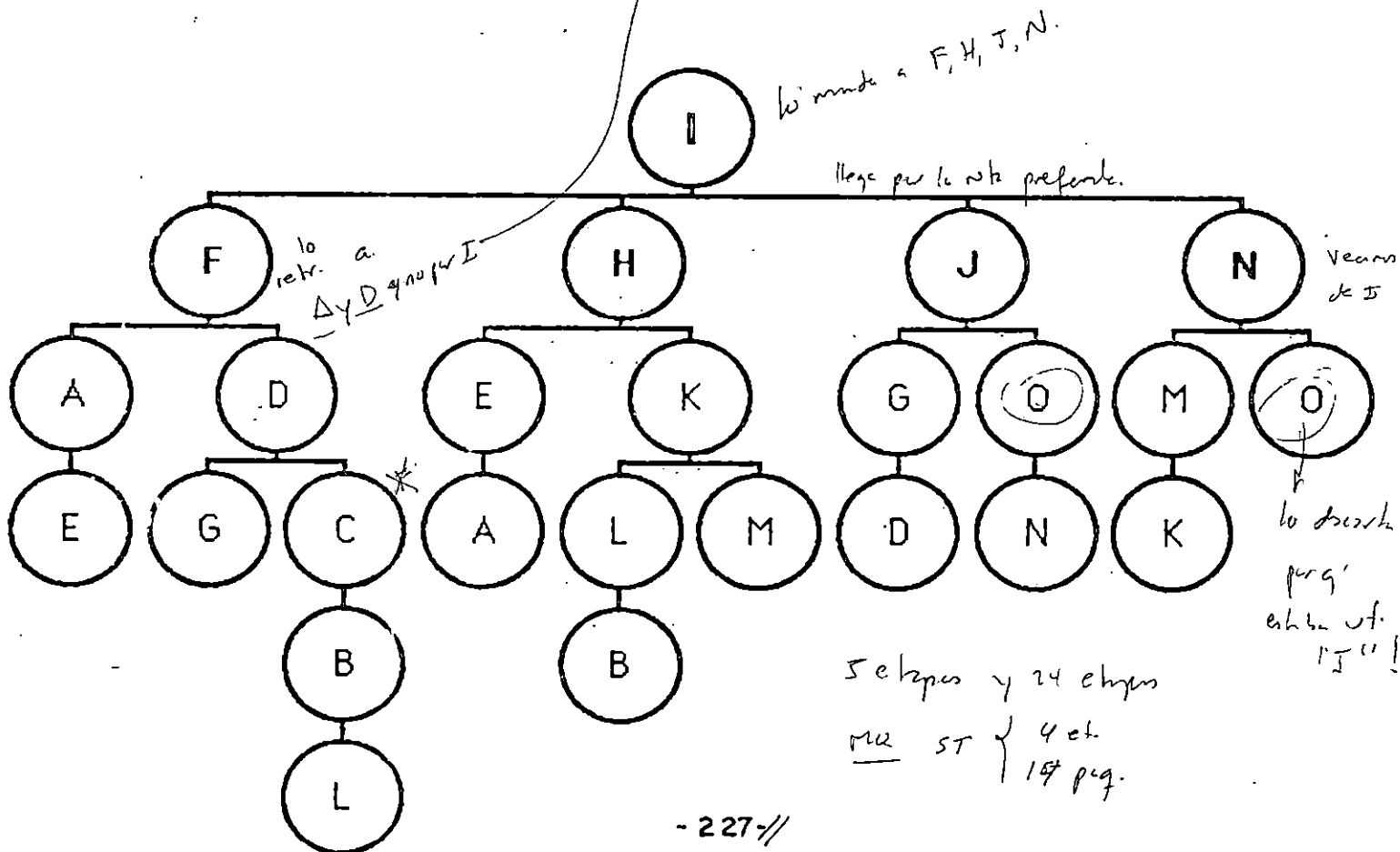


*ST del
nodo I*

Reverse path Forwarding

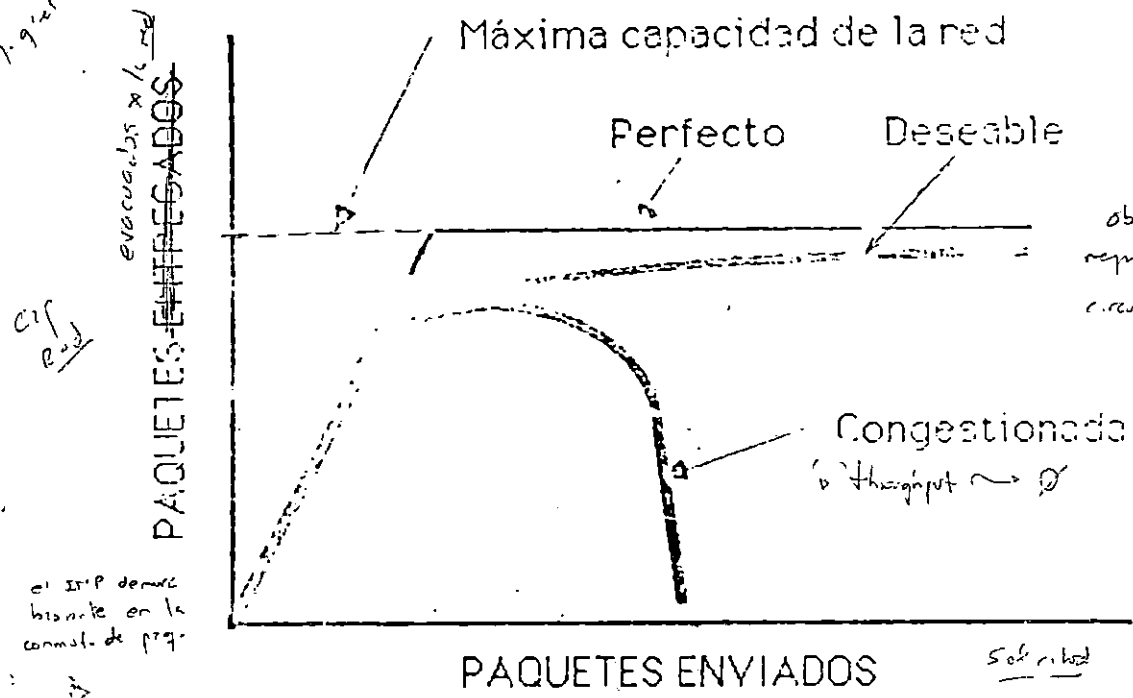
Idea → suponer q' en un momento da da.

- Para cada pckt que llega cada IMP
- Si llegó por lín preferida para enrutar a la fte
- Lo manda por todas la lín excepto la de llegada
- Sino lo descarta
- En ejemplo:
 - Ejecuta en 5 etapas
 - Genera 24 paquetes
 - Sink tree genera 14



Throughput = # pag / un. de t
 en red = red net
 en red = red net

CONGESTION



el IMP demora
 bastante en la
 comut. de pag.

paquete por
 tráfico de entrada
 >
 cap. de salida
 (entrada del
 IMP)

9.000

9.100

$\Rightarrow$ se agotan
 los buffers
 y se atoran en
 IMP.

$\Rightarrow$ muchos pag
 por tr.

y por comut

CONGESTION

- + Causas :
 - + IMPs Lentos *el la comut. del pag. dentro de ellos mismos*
 - Colas crecen aunque haya AB disponible
 - Tráfico Entrada > Σ AB salida
 - Consecuencia: Escasez de Buffers (Retroalim)
- + 5 Estrategias de control de congestión
 - Preasignación de recursos
 - Descarte de Paquetes
 - Restric del # de paquetes en la red
 - Control de flujo
 - Disminuc flujo de entr al detect cong

1) Preasignación de recursos

Técnica q' hace q' la CDP se presta a la CDC. | q'!

Solo es

cuando se trabaja con redes de

Util con Circuitos Virtuales

Por Cell Request ^{preg. en red para llamar} reserva buffers para cada CV.

Señal de "ocupado" si no hay buffers (señal de ^{cuando request} ~~no hay buffers~~)

Técnicas de SW. habrán de ut
a nivel de RED p'
control de flujo / empuje.

Si trabaja stop and wait
2 buffers/CV duplex
1 buffer/CV simplex

→ solo se puede enviar un p'q. a la
red y esperar confirm.

→ nes. reservar 2 buff en c/ SWP. } si SWW.

Al recibir paquete

No envía ack inmediatamente

Espera retransmisión

Asegura así buffer vacío

que haya c/er 1.

SWW
con el
ha ventura (SW)

Conn de paq con CV ^N Conn circuitos q' hacen:

• Asignación ^{estática} de recursos

independiente del tráfico → ^{reun p'} ^{preferir col'q.}

• Congestión ^{es} Imposible ^{porque si la puede haber}

• Uso ineficiente de recursos <sup>cuando al hacer
se detecta la llamada
congestión. (ocupado)
una vez cobrada la
tarifa, no hay congestión.</sup>

en carga baja

cuando la red
tiene va

Sección

Util en aplicaciones con demora crítica

P. Ej. Transmisión (digital) de voz ^{perpetua}

preg? → como se p' 1 buffer ⇒ garantía de trq. de y repto
de p'q's.

2) Descarte de Paquetes

DATAGRAMAS

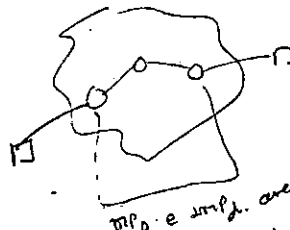
- Opuesta a la anterior
- Si llega pkt y $\rightarrow$ hay buffer $\rightarrow$ descarta paquete
"buffer" es el "redes con"
- Natural en datagramas (no le garantiza al N&T la entrega por los pags.).
- En CV $\rightarrow$ red conserva copia del pkt en alguna parte *para: si alguno IMP falta en pag, es responsabilidad de la red retransm.*
- Sería absurdo descartar pkt con ack *Imp. 2 al IMP fuente*
- Reserva 1 buf/lin en la linea de entrada
- Desperdicia AB por duplicados
- Graduación de timeout afecta cong.
- Para optimizar AB, se descartan pkts iniciando viaje *F.O. pags $\Rightarrow$ congestión. generan (xps, duplicados innos.)*

El IMP debe esperar confirm. de otro p' retrans. si se perdió.

p' debe provida a los pags

pero estos paquetes q' llegan a la red van a ser rechazados

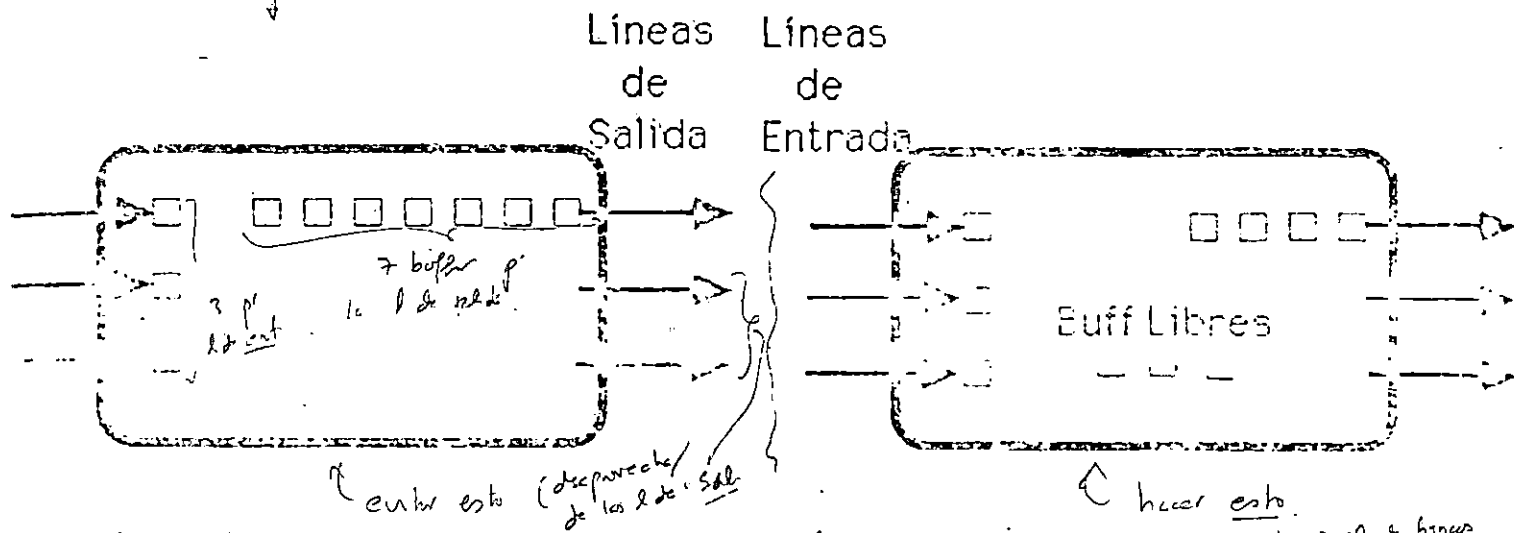
q' están al final del recorrido



IMP. e imp. avaran en particular desde el momento de inf. p' retrans. etc.

Objeto. Lim. en la long. de las colas

idea → limitar el tam. de las colas de salida. (logica 2) ^{haciendo} q' optimice el throughput de la red.



- Limitemos long de colas en líneas de salida ^{Ejemplo → alt de líneas}
- Línea con tráfico alto sigue funcionando a capacidad máx
- Otras líneas pueden funcionar
- Throughput duplicado o triplicado
- Paquetes descartados serán retransmitidos
Posiblemente llegarán antes de que línea quede libre
No habrá demora adicional
- Regla de dedo de Ireland

m = Tamaño máximo de la cola

k = # de buffers disponible para líneas de salida (1)

s = # de líneas de salida (13)

$m = k / \sqrt{s}$ ^{menor q'} $\frac{7}{13} \approx 0.54$

3) Distribución del # de pckts en la red

otra forma de medir de congestión.

Es → restricción de Ven.

Es decir, limitar el # pckts q' circulan en la red.

• Permisos circulantes (pckts especiales)

• Procedimiento

Host entrega pckt a IMP_p

→ IMP ^{debe} coger ^{un pck de} permiso antes de transm

IMP destino entrega pckt a host y

Regenera ^{el pck de} permiso p' q' otro pueda usarlo

• Desventajas del método.

Control global no local

Distribución pareja de permisos complicada

Malo en ^{inferno batch} transmisión de archivos ^{C/track de la estructura de permisos}

Permisos consumen AB y generan congestión

• Difícil detectar ^{una} pérdida de permisos

en el b, se puede perder (en los IMPs)

Si permito 1000 permisos en la red

no puedo hacer q' en x punto ⇒ congestión → truncado (imp) red de perm

4) Control de flujo mec. q' el receptor vibran y no permite al br. q' lo choque

mecc. q' el receptu ubi
p' no pante al br.
q' lo chage

- Confusión con control de congestión
Control de flujo = Cuello de botella receptor pone al transmisor para evitar ahogo
Ej: Stop-and-wait, sliding window (sensitivo) p' evitar congestión.
Control de congestión = Asignación de buffers a paquetes que llegan
→ función de ctrl de fluj. de XPS!
- Algunas redes usan control de flujo para controlar la congestión
- No es un buen sistema -
Tráfico de datos es intermitente
Si (control de flujo en promedio)
⇒ Mal servicio en picos
Si alto
⇒ No sirve para controlar congestión
*pqz: sensib. más "reactiva"
noches → tp
no sirven ante el tráfico es d.*
- Control de flujo puede aplicarse
a Procesos de usuario → restr. al # req. entrantes se h. rechazo.
entre Hosts
entre IMPs fuente y destino
lim. # de CV abiertos simultáneamente

rNA
Docnet.

- bueno*
Mejor mecanismo *solo de base* activado al detectar congestión
(Ej. 2 cm 1m Cam 6cm Pna!)

- idea! C/IMP monitorea ^{el} uso de c/ línea de salida
 y A cada línea le mantiene variable $0.0 \leq U \leq 1.0$
 Muestra ^{el} instantáneamente las líneas $f: (0 \text{ ó } 1)$
 $U = a * U + (1-a) * f$
 $a = \text{peso del pasado} \quad 0 \leq a \leq 1$

Si llega paquete que sale por línea en alerta ent

- marca paquete en tránsito para evitar generar nuevos choke pkts en IMPs siguientes

Quando host fuente recibe choke packet

⇒ Disminuye tráfico a ese destino en X%

Ignora choke pkts a ese destino durante t

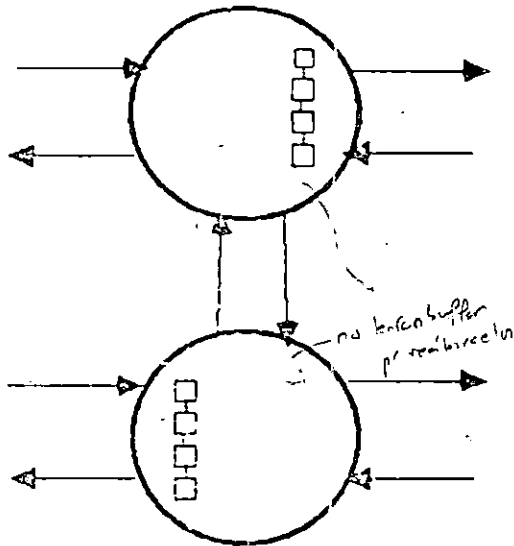
Don't Vuelve a buscar choke packets
Si llega otro ent.

disminuye más el tráfico
sino empieza a aumentar

→ gradar el control de flujo se cambia al botón de red!

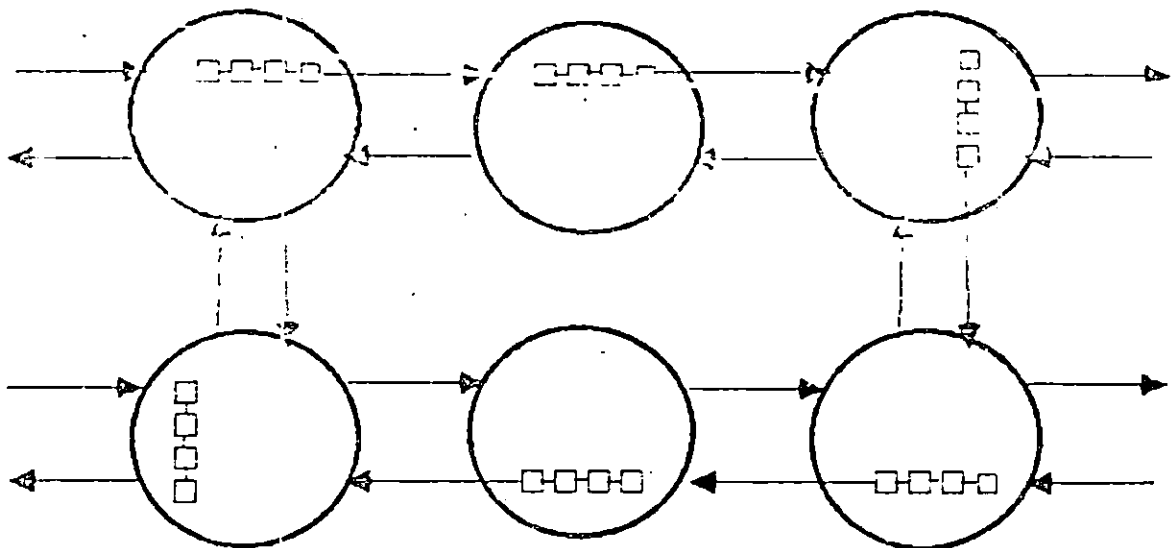
Abrazos Mortales

detectados en bñte dificultad en Impulset.

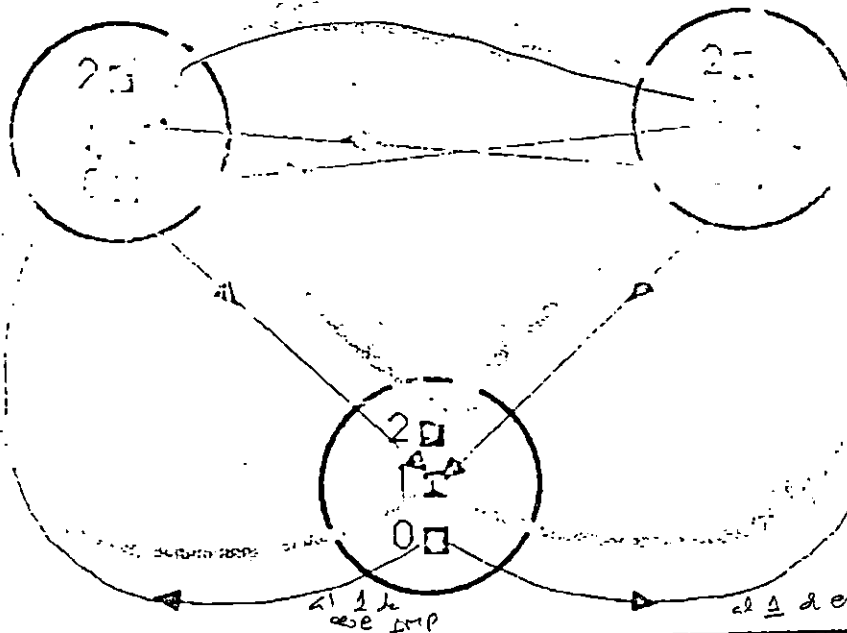


DIRECTO

INDIRECTO



MERLIN Y SCHWEITZER



Funcionamiento:

IMP acepta pág. de host si el buffer está $\emptyset$.

el pág. viaja por los rta. legales del grupo.

si llega a su destino, se entrega al host. pero si llega al buffer $\neq$ y no está en el host destino $\Rightarrow$ se descarta.

$\Rightarrow$ No permite A.M. (Arbitrio de Medio).

Estado del M -como buffer: descargado o contiene un pág. p' un host. local $\Rightarrow$ lo entrega. $\Rightarrow$ se descarta.

En todos los casos, buffer de M queda libre.

Págs. en M - buffers $\leq M \times L$ $\Rightarrow$ no puede haber A.M.

Ineficiencia del método: los buffers no están siendo uti. eficientemente.

MA S construimos un Grafo así

- Supongamos red con N IMPs
- Ruta más larga: M saltos
- Cada IMP necesita $M+1$ Buffers
- Los numeramos 0 a M ^{los buffers}
- $\Rightarrow$ Nodos = buffers, arcos = rutas válidas ^{entre los IMPs}
- Rutas: Buff i de c/nodo a $i+1$ de vecinos ^{vecinos}

A. Morán de Reensamblaje

• Ejemplo:

IMP con 20 buffers

4 líneas de salida a otros IMPs

Nivel de ~~red~~ ^{red} ~~red~~:

Sliding window

Ventana Transmisora = 7

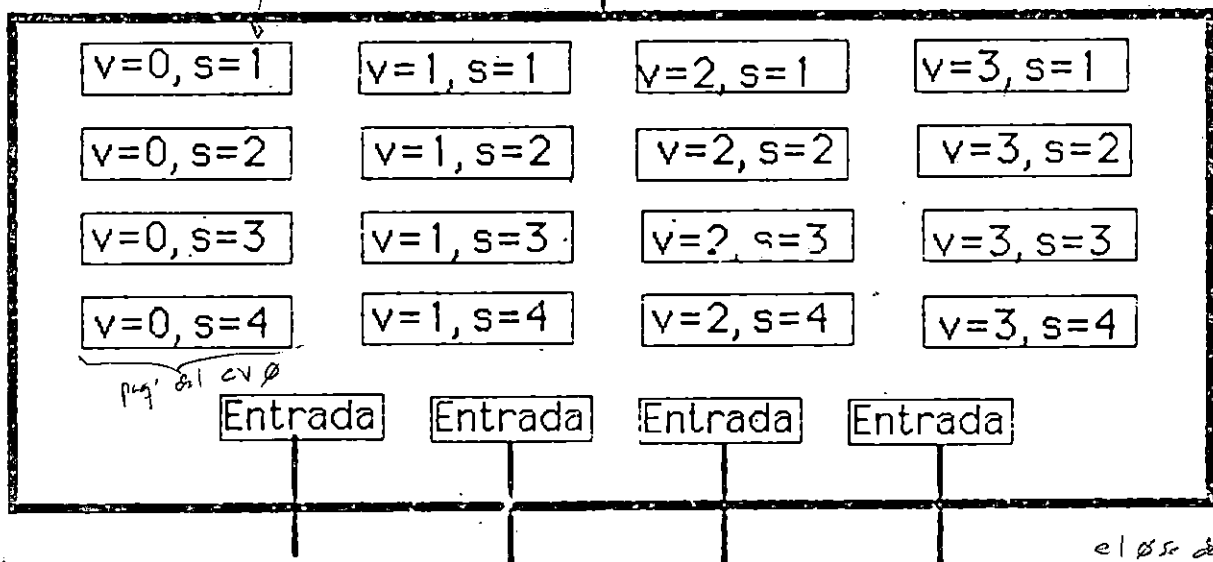
Ventana Receptora = ⑤

4 CV abiertos: 0, 1, 2, 3

Host Espera #seq 0 en todos los CV $\Rightarrow$ A. Morán

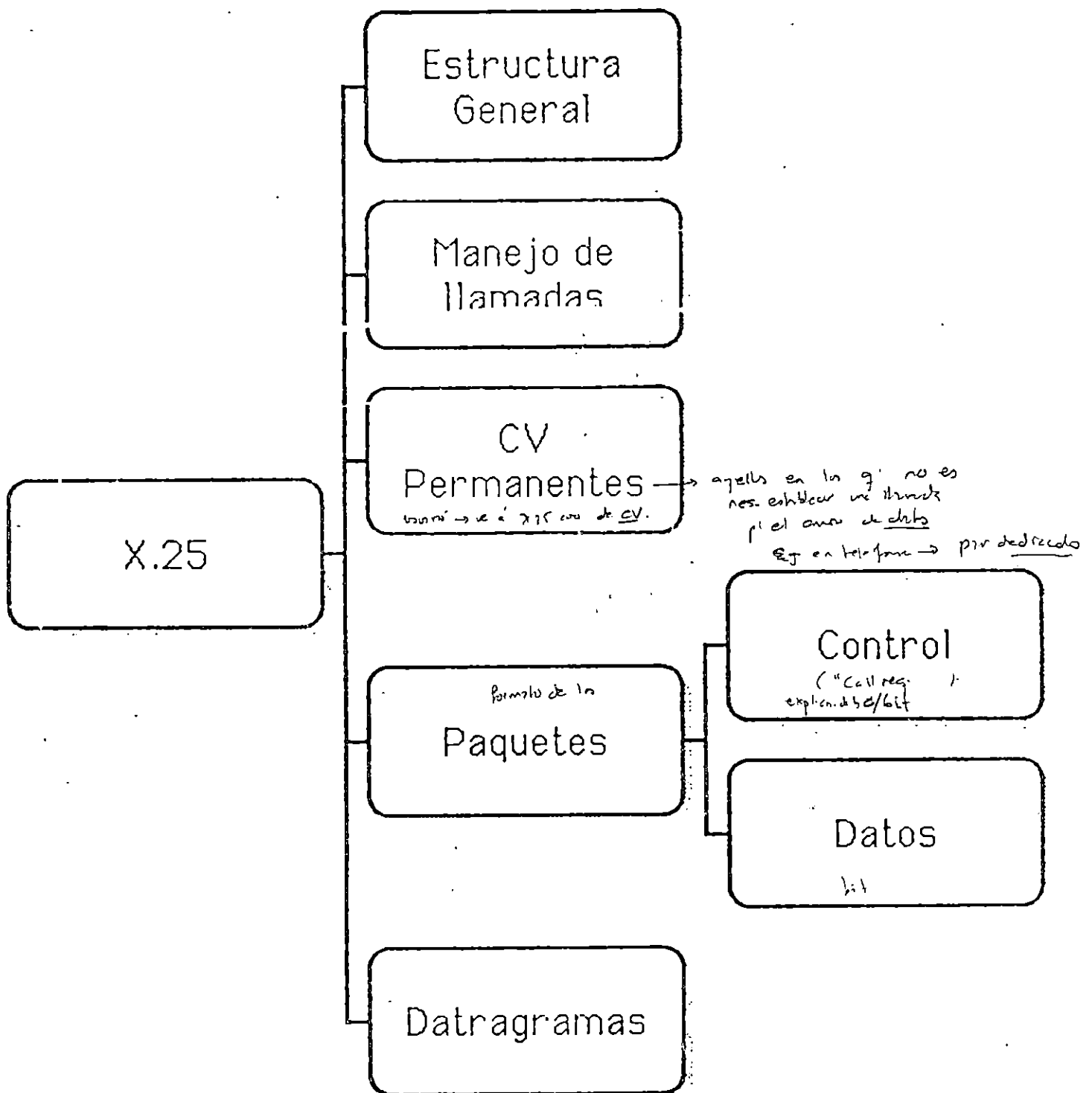
interfaz de CV $\rightarrow$ todo p' el CV en orden
 p' el CV $\rightarrow$ p' el CV
 est. red $\rightarrow$ DIG

AL HOST



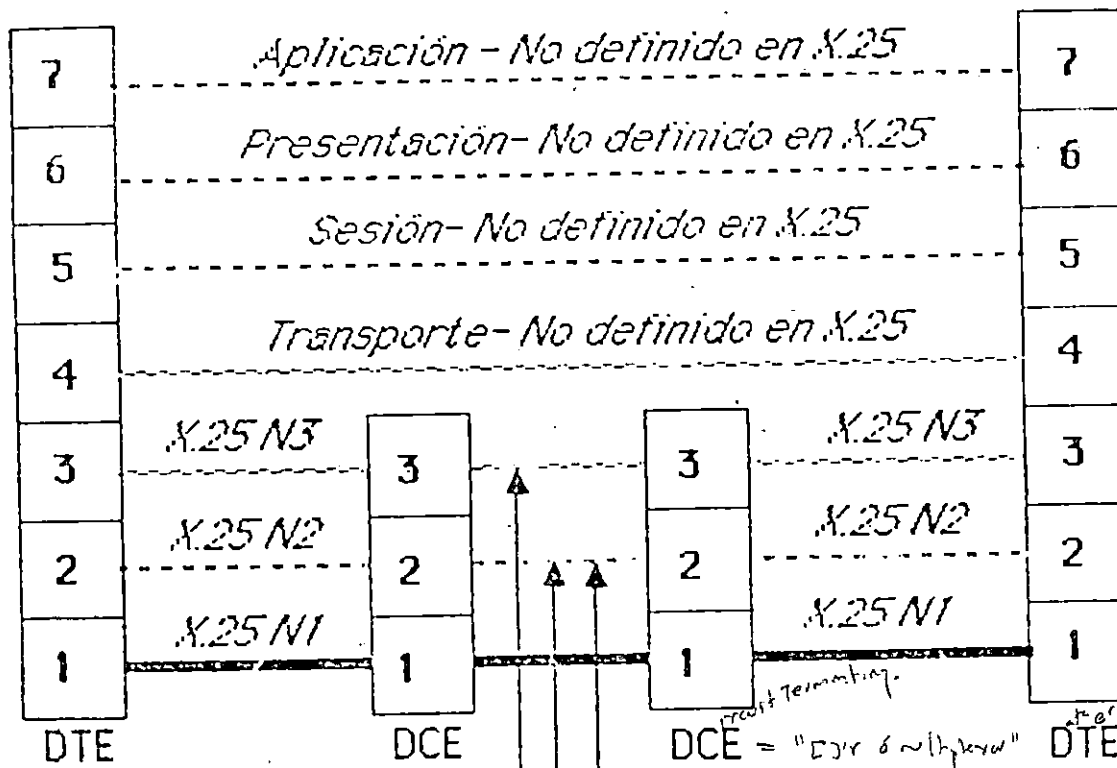
A LOS IMPs

el se detiene en
 todos los CV
 $\Rightarrow$ no le viene a
 pensar más
 (se agotan los buffers)



H.25

→ cubre los 3 primeros niveles del OSI.

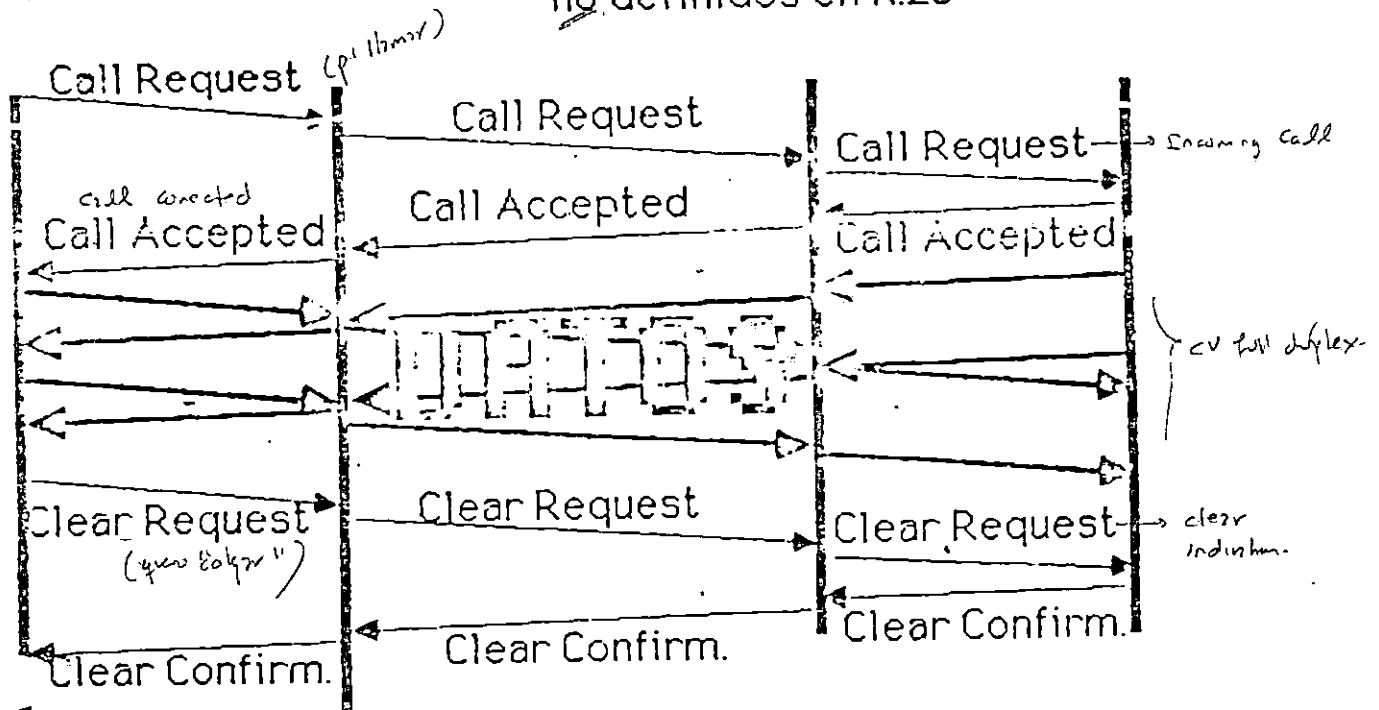


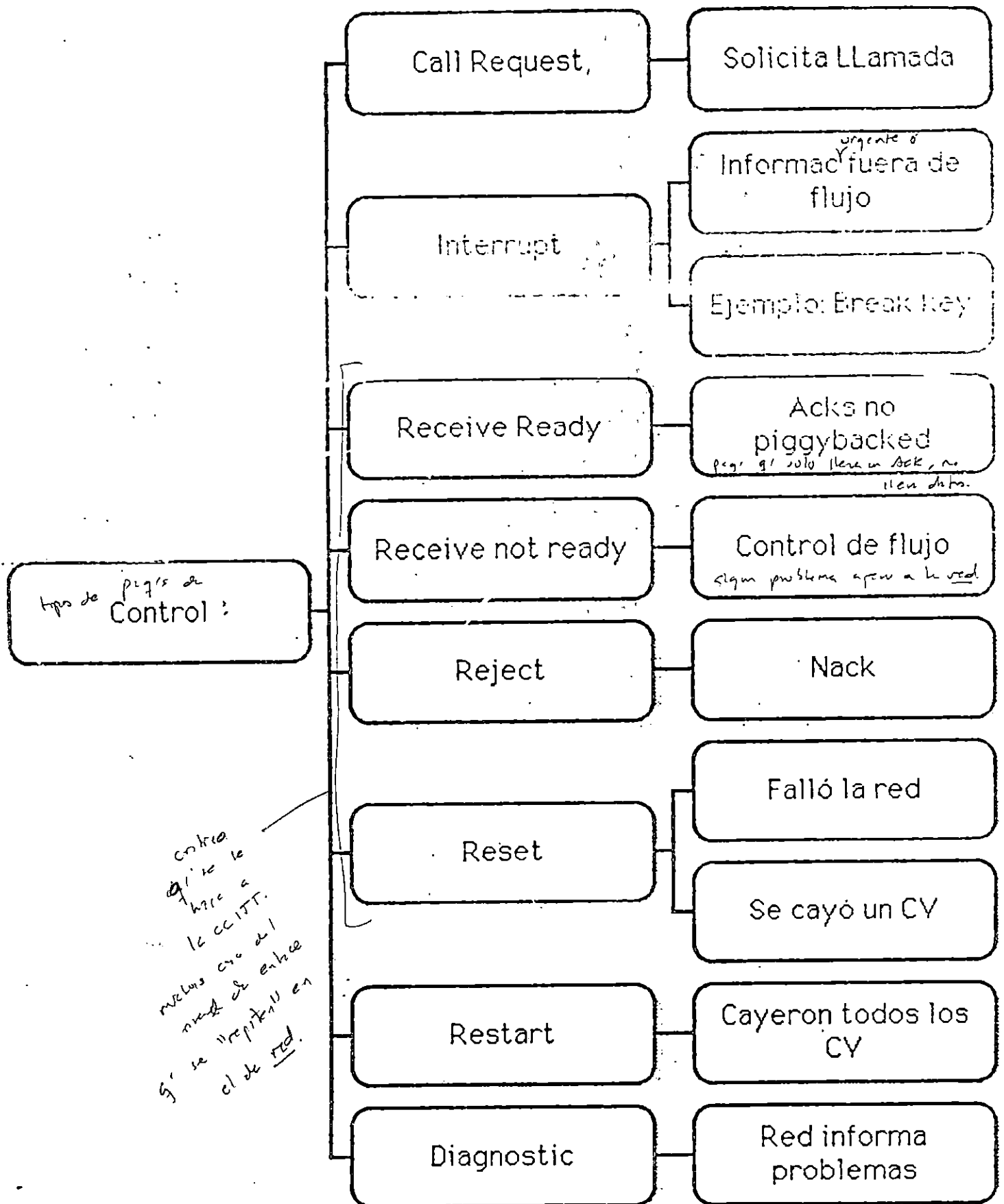
LAPB = ^{efecto} BLP _{protocol}

X.21, X.21bis

ES-732C

Protocolos Internos no definidos en X.25

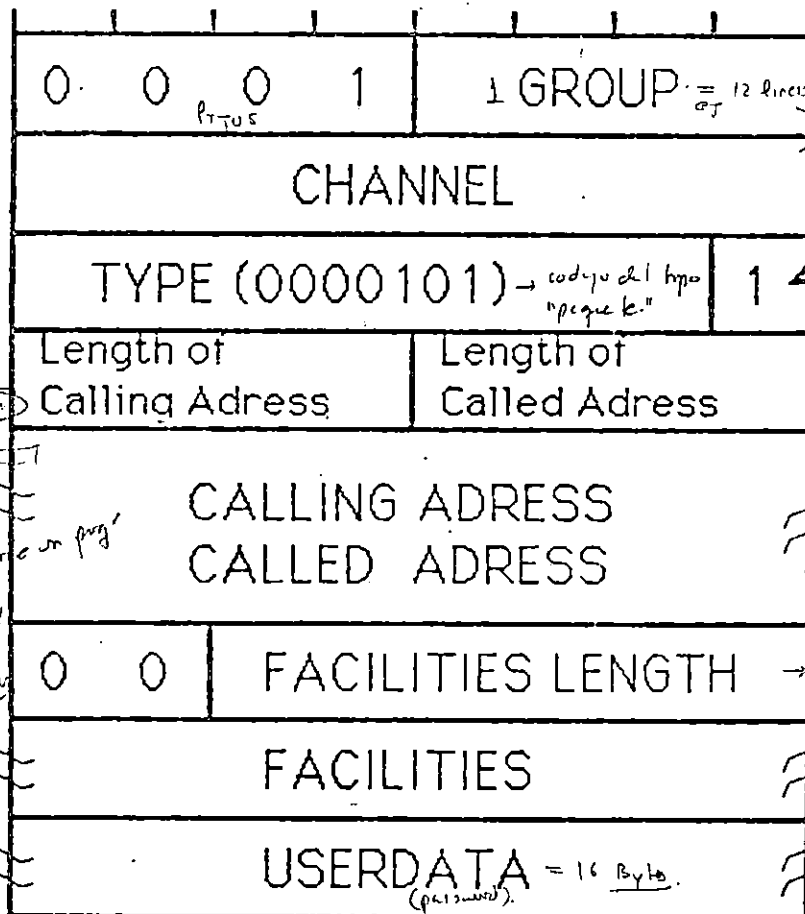




Call Request

- Solicita Llamada
- Group & Channel: #CV, Máx 4095, CV0 reserv
- Type: Tipo de comando
- + Facilities: Longitud de facilities
- Userdata: Significado acordado: Fj: Password

8 bits



del CV

Control

dice si es un prog.
de clrd/dato
y desp si se trata TYPE

X12 → problema:
no de p' interseccion
de cada locion
por eso TCP/IP

requisit
collect
req/vent.
CV si lex.

comando
vel.
var. plex.

Prd de direccion
Plan
ej: código

2 Valgu de lre dr. Jemayec/
de plan
de los dss

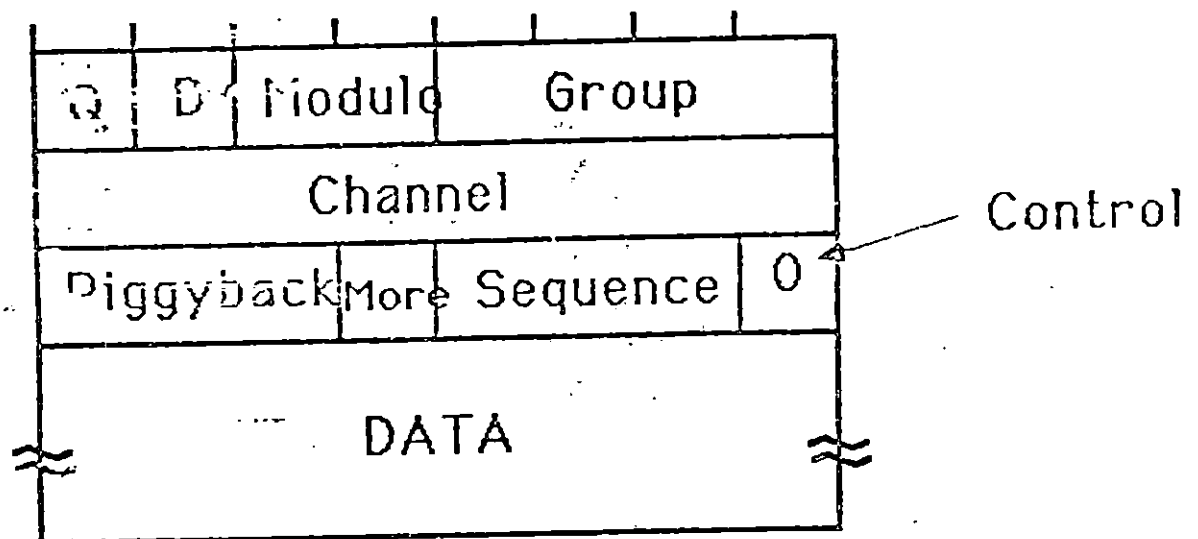
16 Asign. pred ser descono
se puede hacer con p
del usup
x de lre.

On Verifica:
habilidad de lectu
del código
→ se p' mienp
de entre por
"coligante" →

Se requiere
incluye remita
en el código
931-
indica
c/digito true un prog

Datos

- Q = Qualified Data
- Group & Channel = CV
- More: Mensajes Multipaquete
- Control = 0 en datos
- Piggyback: #seq de ack

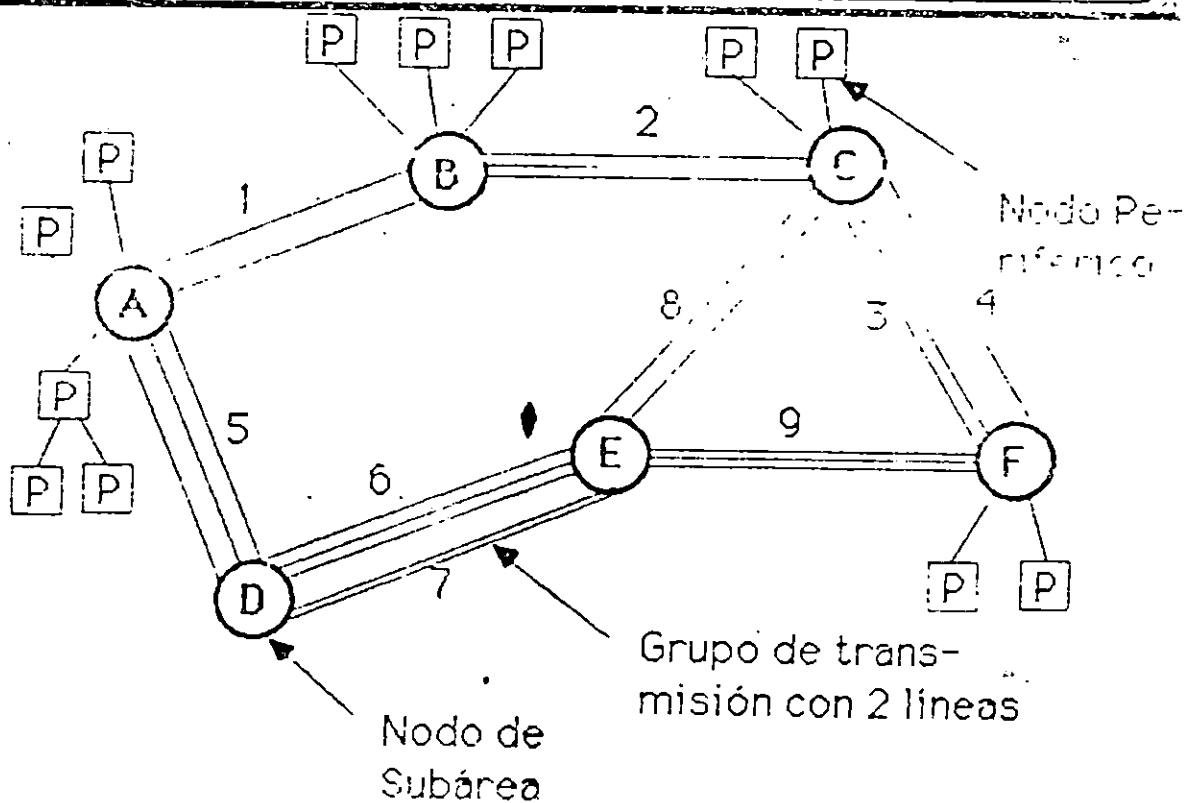


Datagramas

- Japón y USA mandaron propuestas
- Las dos fueron aceptadas
- Versión USA llamada DTGM Verdadero
 - Pckt de DTGM similar a Call Request
 - Byte de tipo reemplaz por seq y piggyback
 - Usados sólo para control de flujo
 - No para confirmación end-to-end
 - Campo de datos se amplía a 128 bytes
 - Los 1ºs 2 son identif del DTGM
 - En facilities se pide notificac de entrega
 - Notificación se hace con pckt espacial
 - Su nombre es: Datagram Service Signal
 - Usa Identificación del DTGM
 - En Caso negativo da razón:
 - Congestión
 - @ mala

SNA

- Nivel de red llamado Path Control
- Circuitos Virtuales interna y externamente
- + Dividida en subáreas
 - + C/U contiene 1 o + NAUs
 - C/NAU tiene 2 @s
 - Subárea
 - @ dentro de subárea
- C/subárea controlada por nodos tipo 4 ó 5
- + Líneas entre subáreas ÷s en grupos de transm
 - C/grupo tiene cola de transm propia
 - Grupos contienen líneas homogéneas
 - Líneas intercambiables dentro del grupo
 - Múltiples líneas/grupo dan > AB
- + Grupos aparte dan > confiabilidad
 - Más rutas alternas



SNA - Ruta virtual

- Lista de subáreas
- ADECF p. ej.
- + Elección de ruta virtual
 - Tablas manuales Estáticas
 - Carga actual
- + Tipo de servicio
 - Interactivo
 - RJE
 - Transferencia de Archivos
 - Seguridad
 - Tiempo real
- Servicios agrupados en tres clases
- Máximo 8 rutas/clase de servicio
- Total= 24 posibilidades
- + Ruta virtual se convierte en ruta explícita
 - Lista de grupos de transmisión
 - P. ej. 5683, 5783, 5684 y 5784 para ADECF
- Nivel de enlace escoge línea de grupo
- + Pckts pueden desordenarse de una subárea a otra
 - SNA resecuencia en cada subárea

SNA - Control de flujo

- + Se hace por ruta virtual
 - Mecanismo llamado Pacing
 - Transmisor recibe autorización
 - Envía ventana de n paquetes
 - El 1º contiene requerimiento de permiso para otros n
- + Si receptor tiene espacio
 - Envía permiso para otros n

SNA - Control de congestión

Basado en control de flujo
Se ajusta dinámicamente n de pacing
Cada ruta virtual tiene n mín y n máx
Cada paquete tiene 2 bits

- Transmisor los coloca en 0
- Si IMP intermedio está moderadamente congest
 - Coloca uno de los bits en 1
- Si está seriamente congestionado
 - Coloca el otro bit en 1

El recept sabe de estado de congestión

- Si moderada, baja $n--$
- Si grave coloca $n = n$ mín
- Si no hay congestión $n++$
- Siempre n mín $\leq n \leq n$ máx

Parecido a choke packets

SNA -Cambios de Topología

Si falla grupo de transmisión

Rutas explícitas que lo usan se cambian

Escoge otra ruta explícita dentro de ruta virtual

Si no hay se busca otra ruta virtual

Si no hay se aborta la sesión

Si grupo cae o renace se generan pckts de ctrl

Así se conocen rutas explícitas disponibles.

Ablocamiento de paquetes

+ Pckts no relacionados pueden ablocarse en un BTU

En siguiente subárea se separan

Se ponen en cola por separado

Ablocamto y desablocamto hecho en cada salto

Bloque puede tener pckts de $\neq$ s fntes y destinos

ARPANET

- + Generalidades
- + Mensajes
- + Protocolo IMPs destino y fuente
- + Control de congestión
- + Enrutamiento

Generalidades

- Datagramas con interfaz de CV
- $V_{\text{máx Host-IMP}} \approx 100.000\text{bps}$
- 24 bits de CRC por HDW y 16 por Soft
- Bit de versión para distribuir nuevos protoc
- Bit de Trace para estadísticas de uso de rutas

Mensajes

- Long. Máx = 8063 bits
- Header de transporte = 40 bits
- Host agrega leader de 96 bits
- Número de CV
- Host destinatario
- IMP local lo quita
- IMP dest pone uno nuevo
- IMP fte ÷ mens en 8 pckts de 1008

Protocolo IMPs destino y fuente

- Ack entre IMPs destino y fuente
- Se llama RFNM (Request For Next Message)
- Transmisor puede hacer timeout y retransm
- Adicional a acks de enlace
- Fuente conserva copia hasta recibir RFNM
- A los 30 seg fuente interroga a destino
- Dependiendo de respuesta retransmite

Control de congestión

- 1a V: 1 mensaje por CV
- 2a V: 4 mens máx por pareja IMPs Fte-Dest
- 3a V: 8 mens por pareja de hosts
- En mensajes simples descarta recordando requer
- Manejo de Abrazo mortal de reensamblaje

Abrazo mortal de reensamblaje

Si IMP fte tiene mens multipckt ent
envía packt reservando

IMP destino devuelve ALLOCATE

IMP fte transmite

IMP destino conserva buffers

Devuelve nuevo ALLOCATE con RFNM

ALLOCATE se mantiene por

250mseg

Enrutamiento.

- + Inicialmente: Distribuido
 - Hold Down
 - Intercambio c/640mseg
 - Paquetes de enrutamiento de 1160bits
 - Menos frecuente en líneas $V < 50\text{Kbps}$
- + Cambiado en 1979 porque:
 - Adaptación lenta
 - Pckts en loop
 - No usaba rutas alternas

Enrutamiento (cont'd)

- Arpanet muy grande.
- Congestión por muchos pckts de enrutam
- + Nuevo mantiene informac de toda la red
 - Calcula demoras a todos los nodos
 - Mide demoras en todas sus líneas
 - Saca promedio de últimos 10 seg
 - La envía a todos los otros IMPs
 - Usa enrutamiento por inundación

DECNET

- Datagrama puro
- Enrutamiento distribuido
 - Intercambios con vecinos
 - Periódicos
 - Especiales
 - Caída y recuperac de líneas e IMPs
- Control de Congestión por Choke packets
- Usa contador de saltos para loops
- Formato del Frame

Cntrl de Cong por Choke packets

CREDITO = Estándar

USO = Número pequeño

REPITA periódicamente

Si colas crecen rápido o uso alto de lín ENT

Envíe choke packet a IMP fuente

En IMP fuente: $USO = USO * kte$

SINO

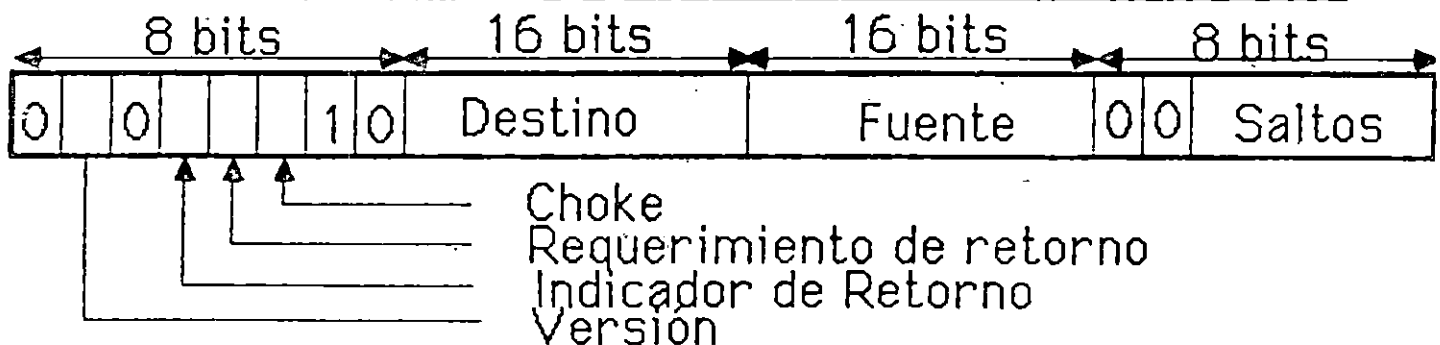
$CREDITO = CREDITO + kte1$

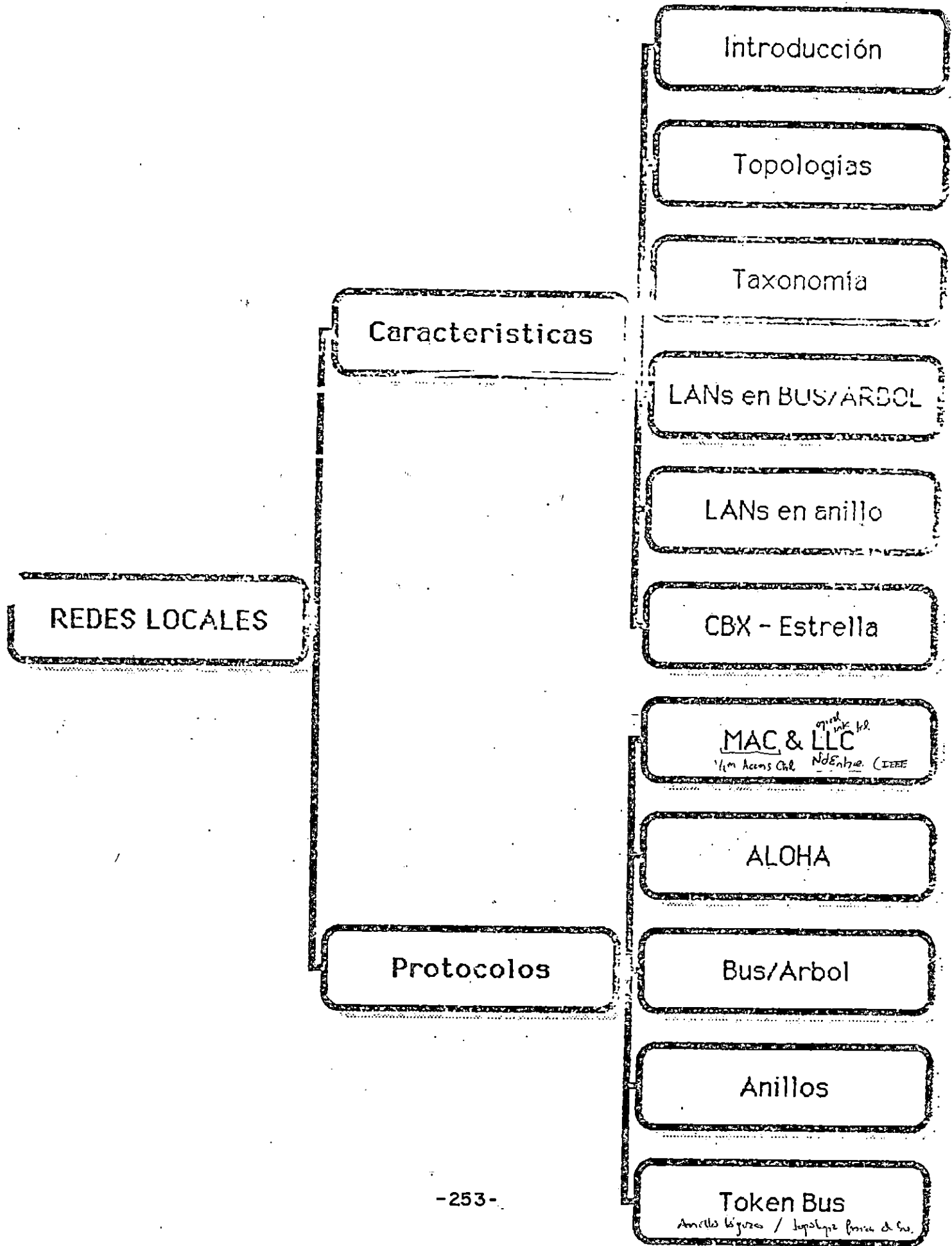
$USO = USO - kte2$

FSI

F-REPITA

Descarte de packts con Irland





Introducción

- Qué es una red local?
- Qué no es una red local?
- Tecnología Inmadura

Qué es una red local?

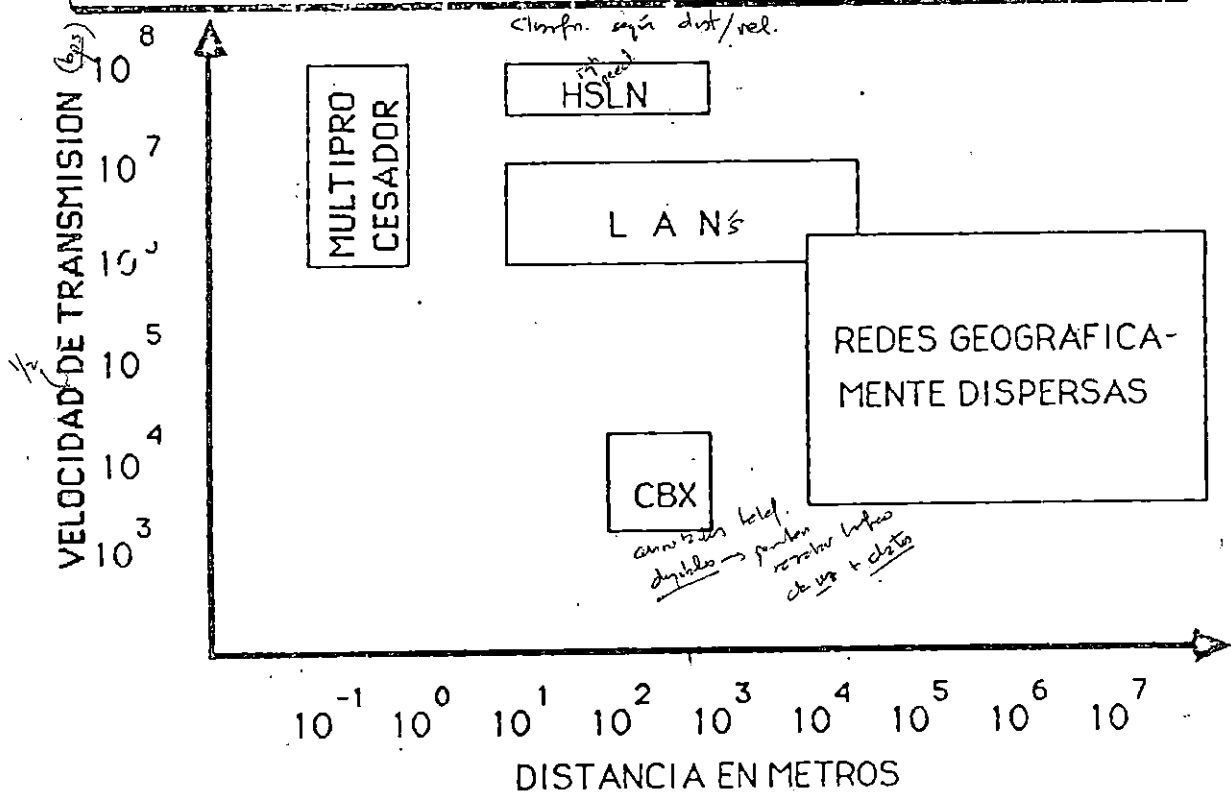
- No necesariamente área pequeña
grupos de m = edificios
- Generalmente ^{son} privadas (Cía)
- Topologías simétricas ^{bus/estrella/Anillo, etc.}
- ^{correcto, sino prop. de una red local} Nivel de enlace subdividido en MNC & LLC
- No solamente conectan computadores
sino: teléfonos, video, etc.
- Generalmente dentro de un edificio

Qué no es una red local?

- Sistema virtualizador de discos *duros*
- Sistema virtualizador de impresoras
- Sistema sin control de concurrencia

RL → Tecnología Inmadura ^{heer} ^{con} (incompleto)

- Productos incompletos
- "Carencia" de estándares ^{IEEE!} ^{← falta estándar}
^{recomendación → Tomar pelo}
^{q' no son IEEE!}
- Variaciones bruscas de estándares
- Carencia de software ^{BSD Distrib. en acceso en t.p.}
^{Compil. sin ctrl. de memoria}
- Fallas ^{con} en hardware y software
- Escasez de soporte



Topologías

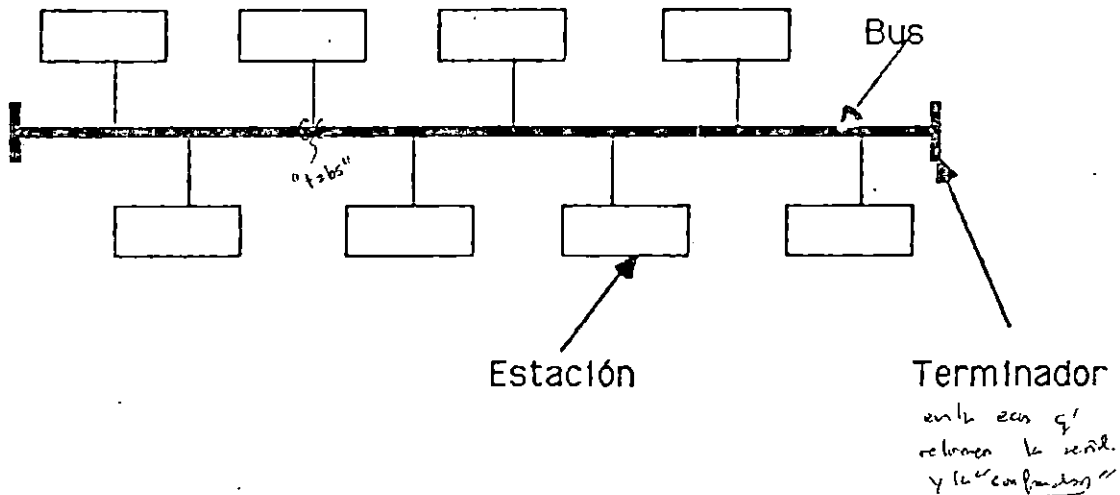
de RL's.

- Bus/Arbol
- Anillo
- Estrella

Bus/Arbol

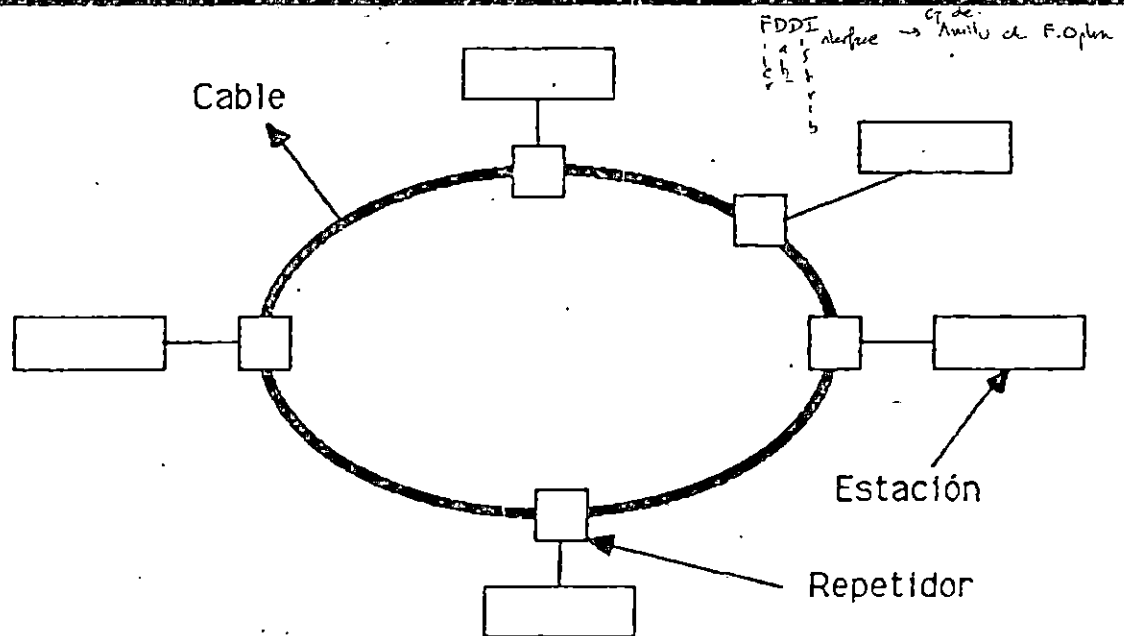
Los símbolos del bus y árbol.

- Más comunes
- Medio Pasivo
- Broadcast *todos oyen todo lo que se transmite*
- Arbitraje del bus
- Centralizado *señales únicas p' q' los datos concuerdan*
- Distribuido *señales de cada estación en q' se transmite*
- Paquetes con dirección *por ser en buses multipunto!*
- Fáciles de instalar
- Daño en Bus
- Aislamiento parcial



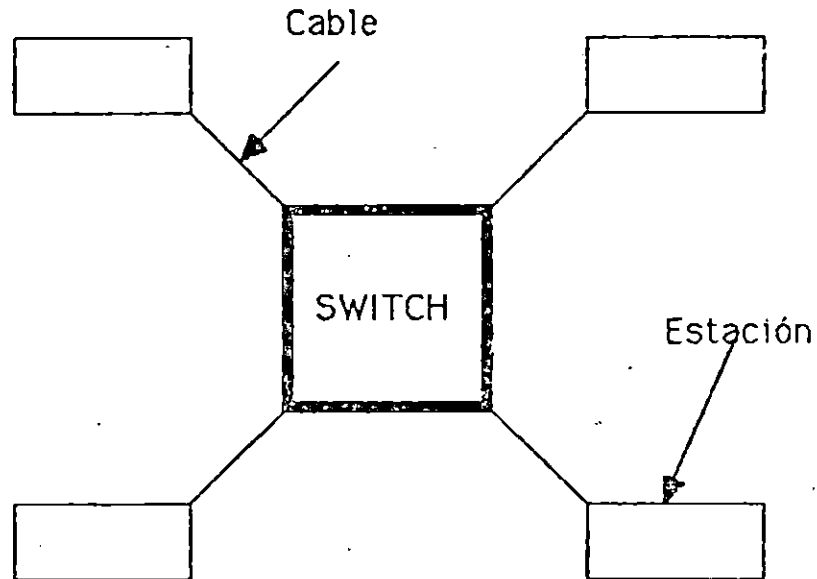
Anillo

- Repetidores punto a punto
- Buffer repetidor = 1bit
- ^{flujo de enf.} Unidireccional!
- Paquetes con dirección
- ^{se res en red} Control de acceso al medio
_{pl. de red que puede tr. en un m. dado.}
- Posibilidad de fibra óptica
- Falla en cable o repetidores
 - Implica caída de toda la red
- Medios de transmisión
 - Par enrollado
 - Cable coaxial
 - Fibra óptica → no usual en redes en bus
por no ser extensible en anillo! ^{get}



Estrella

- Estaciones punto a punto con switch
- Conmutación de circuitos y no de paquetes
⇒ (wz1)
- Carga al switch
 - Estaciones descargadas
- Integra bien voz y datos ej:
Edif. Banco de Bogotá
aproximar a telefonia



Taxonomía

- LAN
- HSLN
- CBX ^{mult. end change}
- Parámetros de decisión

LAN

- Area relativamente grande ^{era bps/seg}
 - ^{pred. 1100} Tráfico combinado
 - Voz
 - Datos } ^{en CBX.}
 - Gráficas
 - Video
 - Velocidades típicas: 1 a 20 Mbps
 - En micros
 - Velocidad típica = 1Mbps
 - Medio ^v = par enrollado
 - Estándar = IEEE802
- [↓]
 T.R. → 4 Mbps
 Ethernet → 10

HSLN

- ^{El gigante} Conexión rápida de mainframes
- Normalmente en bus con ^{cable} coaxial
- Velocidad típica = 50Mbps
- Distancia y número de dispositivos limitado
- Costosa ^(por nivel!) ----> no usada en minis ni micros ^(LAN)
- Típicamente dentro de un cuarto ^(cdc)
- Estandar: ANSI X3T9.5

CBX

- Integra voz y datos
- Topología de estrella
- Medio = Par enrollado
- Comutación de circuitos
- Conmutación digital por división de tiempo

Parámetros de decisión

- HSLN tiene campo aparte ^{# 200)}
- Gran competencia entre CBX y LAN
- CBX limitado en velocidad ^{pero es}
- CBX más barato ^(cable más barato y es más barato)

LANs en BUS/ARBOL

- Generalidades
- Baseband ^{tecnología}
- Broadband ^{p' RC en B/A.}

únicos
enlaces
multipunto
(sólo en el
q' se conecta
por los estremos)

Generalidades

- Unicas multipunto (instalable en todos los sitios)
- Problemas con balance de la señal
 - Potencia debe estar en cierto rango
 - Desviación grande de distancias
 - Solución: Amplificadores-Repetidores
- Control de acceso al medio
 - Centralizado -----> Polling ^{baseband (centralizado)}
 - Distribuido -----> Ethernet ^{baseband (distributed)}

ni
s'ha potente q' pueda
dañar otros estn.
ni ha débil q' otros
no puedan oír la señal
(no repetidores)
Broadcast: todos oyen,
"hello"
⇒ control y acceso al
medio p' entre estremos

XEROX → respaldado por el
DIGITAL → H. Packard
(etc)

TERMINADOR
(absorben la señal p' no genere
ECO q' el cable)

CABLE COAXIAL

CONTROLADOR
ETHERNET

TRANSCIVER

TRANSMITE/
RECIBE / recibe 1/2 de comput.
DETECTA SEÑAL
DETECTA COLISIONES

(cable + de 1 estn.
hacen malhecho 2
y se destruyen la
señal)

-261-
255

Separado del transceiver
por razones de manteni-
miento

RED
ETHERNET

E. Loc. en bus tokenring.

Baseband ^{es una Tecnología}

• Significado

• En general: Señal no modulada ^{d/a}

• En redes locales: Señalización digital ^o

• ^{la señal es} Bidireccional $\Rightarrow$

^{Ej: Arco hie, PC-Net}
la señal se propaga en ambos sentidos ($\neq$ broadcast)

• Distancias cortas $\approx 1\text{km}$ $\leftarrow$

señales digitales son más propensas a la atenuación q' las analógicas

• No ^{es} apta para árbol ^{topología de}

$\leftarrow$ ^{por el de pot de la señal (atenuación)}

• Medios de transmisión ^{usados:}

• Par enrollado

• instalación fácil

• Velocidad $\approx 1\text{Mbps}$ ($<$)

• Bajo costo

• Longitud $\approx 1\text{km}$

• ^{podemos colocar} Algunas decenas de estaciones

• Cable coaxial

• Velocidad $\approx 10\text{Mbps}$

• Número de estaciones ≈ 100

• Longitud máxima sin repetidores $\approx 500\text{m}$

• Número máximo de repetidores ≈ 4 ^{entre tramos de cable}

• Longitud Máxima total 2.5km

Fibra óptica NO por su dificultad de conexión!

Ej: señal q' va desde nuestro teléfono hacia la 1ª estn. (central telef) (señal analógica)

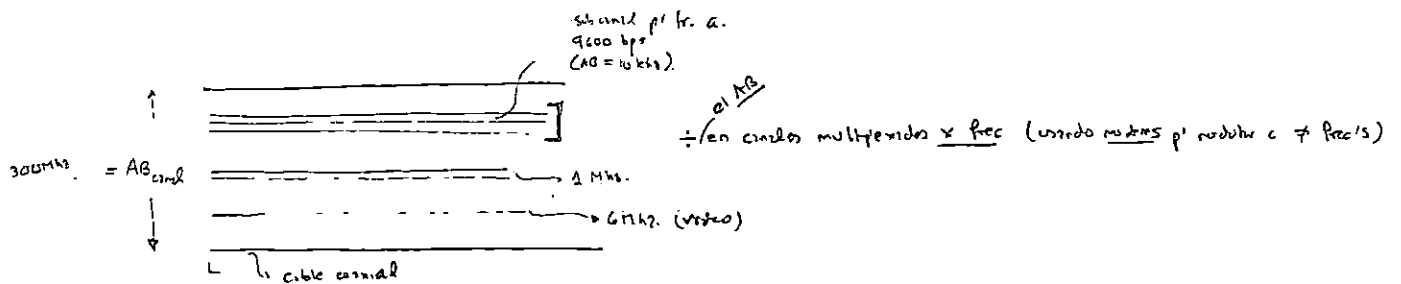
Central telef

Q

$\neq$

por el problema de balanceo de la señal!

2.5 km



Broadband

- Significado $\rightarrow$ En el contexto q'd se com'r canal con $AB \geq 4 \text{ kHz}$ (esp mayor q' la de un canal de voz)
En b. de R-Los Angeles: señales / módulos x frec y C. Coaxial
- Usa FDM (Multiplex / x frec)
- Longitud mayor: Decenas de kilómetros
- Apta para árbol (redes son) señales digitales no interfieren como las analógicas (pueden haber interferencias en el bus!)
- Unidireccional es, porq' nos. ampl. Residuos (señal es ANALOGA) y no repetidores pus la
- Cientos y aún miles de dispositivos
- Logran de 0.25 a 1bps/Hertz NIESTO $\rightarrow$ de c/ hertz se pueden sacar 2 bits
" 1 bit en el mayor de los casos.
- Formas de comunicación los cables pueden com. por 1/4 de un canal $\leq \frac{D}{C}$ AM.
- Instalación Complicada modems $\neq$ p' com. s. dig $\rightarrow$ analóg.
cable coaxial
- Maneja tráfico de distintos tipos dato de un cable se puede recibir video, voz, de otros tipos.
- Red eléctrica como medio de transm Ej: "path"

Unidireccional

Si nos el problema de q' la red es $\rightarrow$:

• Midsplit

• Ahorra cable (10-15% del costo total)

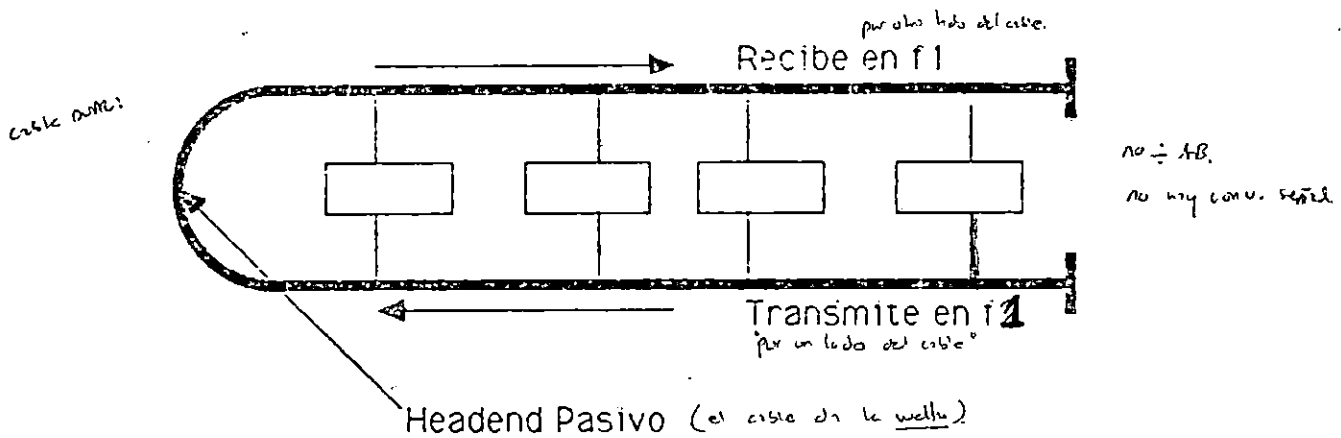
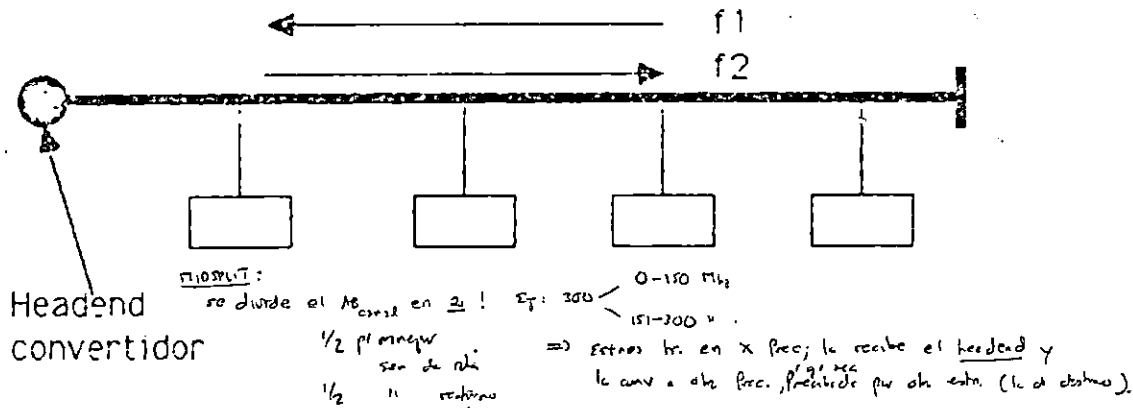
• Bueno si cable ya está instalado

• Convertidor redundante $\leftarrow$
por si se cae el headend!

• Cable dual

Desventajas:
dependen del
AB
el resultado es
problema de
zorro!

• El headend amador
es una "centralita"
del negocio de pollo
 $\Rightarrow$ tener duplicados!
(de pollo)

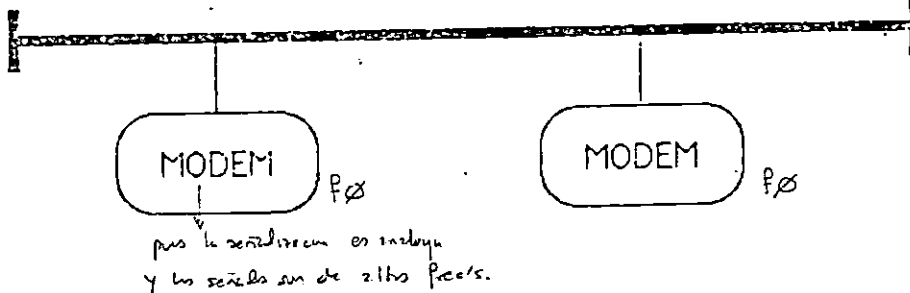


³ Formas de comunicación: ^{banda} (ancho)

entre estaciones

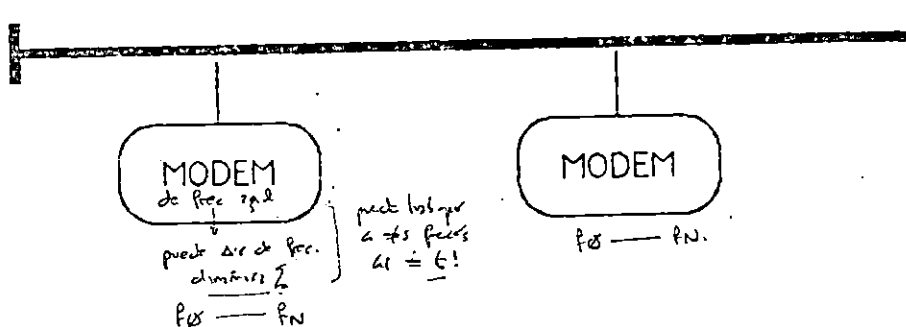
- ^{canal} Dedicado
- Conmutado
- ^{de} Acceso múltiple

Canal DEDICADO: p' com'r minifones a una frec. dada (alta $\Rightarrow$ ^{canal} modos). ^{por s. Analógico}



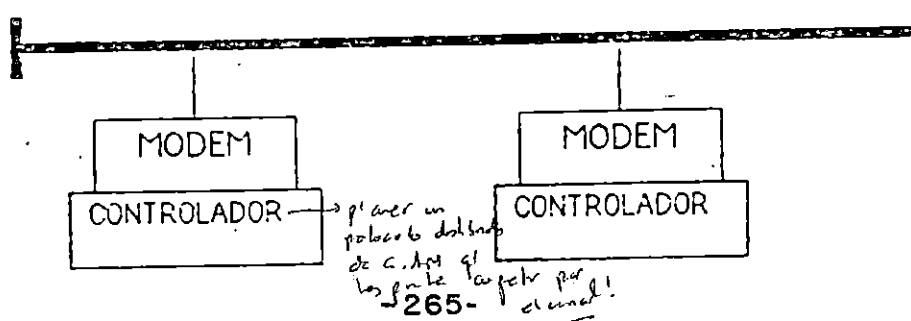
CONMUTADO

hay un medio controlador de conmutación de señal que se encarga / puede circular con x frec. p' establ com.



ACCESO MÚLTIPLE $\rightarrow$

estac. res comp p' coger en estac. (código).



Transmisión por Red Eléctrica

- Bajo costo
- Llamada red inalámbrica
- Interfaz con el medio
 - Tomas eléctricas normales (enchufes)
- Modulación ~~de~~ ^{por} fase
- Paso de banda: 200-400khz
 - ⇒ No interfiere con corriente (50-60hz) ^{redes}
 - Se sacan de 10 a 20 canales de 9600 bps
- Ventaja fundamental: Portabilidad
de la pta a la red.

Poco conocidos

Comparación Broad .vs. Base

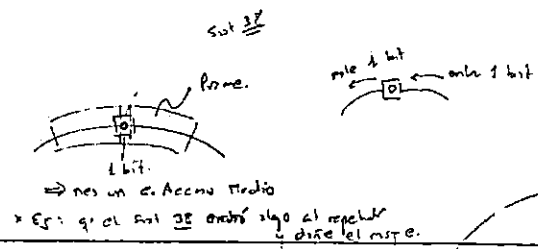
• Broadband

- Más poderosa
- ^{cable} Mayor distancia.
- ^{múltiple} Tráfico de varios tipos

• \$ (costo m. de 1 sec.)
c. común

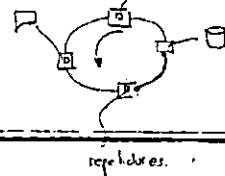
• Baseband

- Más barata
- Tecnología más simple
- Más fácil de instalar ^{(7 mds) / por cableado}



LANs en anillo

"TOKEN" → quien tenga la pizca lee el cable cuando sea vez de devolver la pizca



- Generalidades
- Estados del repetidor
- Ventajas de los anillos
- Problemas de los anillos
- Estrella Anillo
- Estrella Anillo Jerárquica

Generalidades

- Funciones de los repetidores
 - Insertan mensajes
 - Reciben mensajes (COPIAN msjos).
 - Remueven mensajes *recibir un msje > sacar un msje*
- Mensajes enviados en paquetes con @
- Remoción de paquetes *puede ser hecho por el*
 - Receptor
 - Transmisor
 - Permite confirmación automática
 - Permite multidirección *permite env en p.g. a mltos cohos en la red*

3 Estados del repetidor

1) Inspección

- Busca paquetes con dirección local si detecta dir local
- Copia paquetes con dirección local si los pega en buffer sin p' copia de sus enlaces.
- "Modifica bits al paso." p. ej. ack

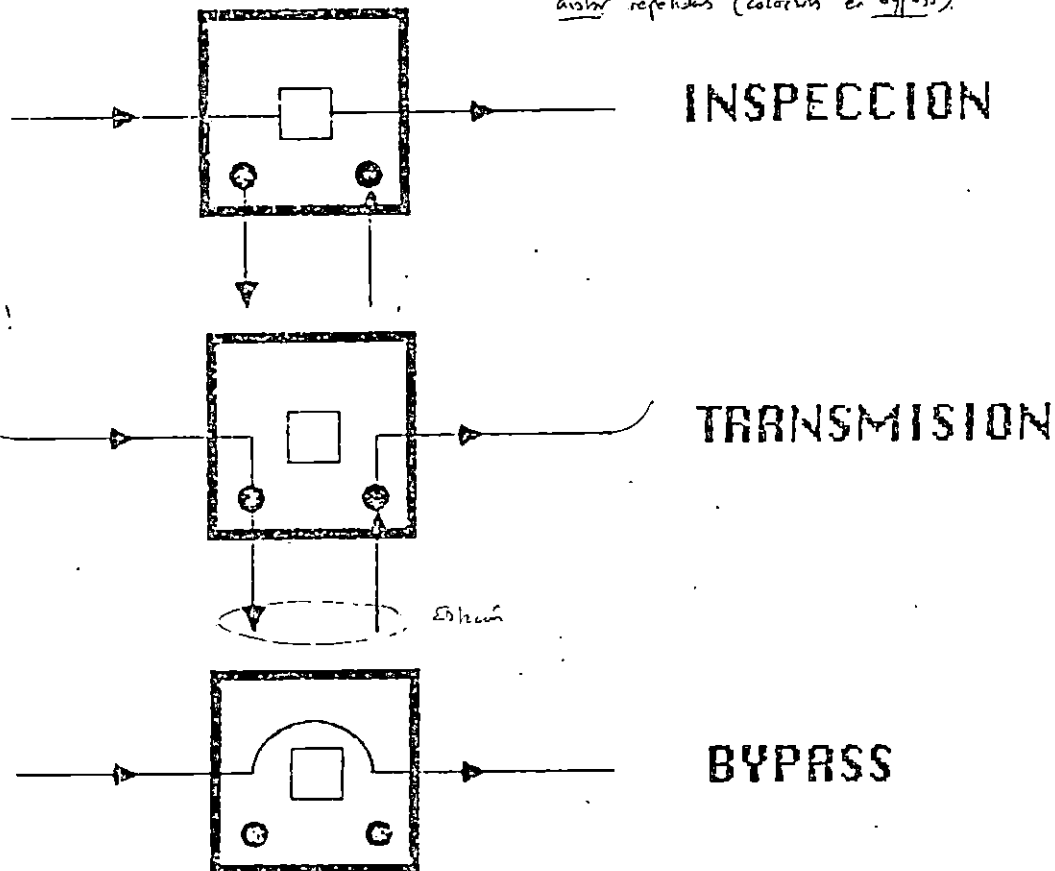
2) Transmisión

- Puede recibir mientras transmite en el momento en q' por el buffer por el repetidor, el packet cambia el estado de algún bit.
- Longitud del anillo < longitud del paquete
- Otro paquete en anillo multipaquete

3) Bypass

- Evita demora en estaciones no activas
 - Permite aislar repetidor fallando q' está o p' colocarlo en manteni.
- c/bit debe pasar en c/ repetidor q' haya => [un repetidor adicional => demora de n bits por rep.]

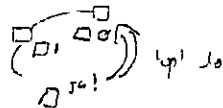
aislar repetidos (colocarlo en bypass)



Ventajas de los anillos

- Punto a punto y no multi-punto
- Regeneración de la señal ^{dupl.} en c/ repetidor
- Minimiza errores y
- ^{fácil} Distancias mayores ^{la señal solo "cae nueva" de c/ repetidor}
- Fibra óptica (por ser pto. a pto.)
- Mantenimiento más simple (").
- Aislamiento más simple que en bus ^{hyper. switch en los repetidores p' col en hipers.}
- Fácil reconocimiento de dirección duplicada
^{c/ hiper. Ethernet tiene una dir y # identif. unid.; en anillo no hay esa problema!}

para saber q' una = dir se le agrega a + de 1 estn. → solo: uno manda un paquete en un lat de tiempo en p' si alguien lo oye ^{eso} dir ⇒ $\frac{1}{2}$ ⇒ dir duplicada.



Problemas de los anillos

- Vulnerabilidad del cable y repetidores
- Bolsillo lleno de llaves → ^{Ex: 500 estn} ^{exceso de term. y 7 p' publicistas (res. de 1/2 p' por el canal)}
- Instalación de nuevos repetidores
 - Identificación de vecinos
 - Acumulación de cable
 - Geometría irregular ^{+ cuando la se tiene} a redonde q' la red se redondea
- Inicialización y recuperación complicadas
 - Quién empieza? ^{quien es el dueño del token}
 - Quién remueve paquete al caerse transmisor?



se en un
un. de
de estn.
tr a p' p'
y no lle
llaga ⇒
sabe q'
falta la
red p'
no en q'
p' p'
⇒ el adm.
de la red.
sabe q' un p' p'
bueno
p' ver en
q' estn p' p'
se p' p' (2)

Estrella-Anillo

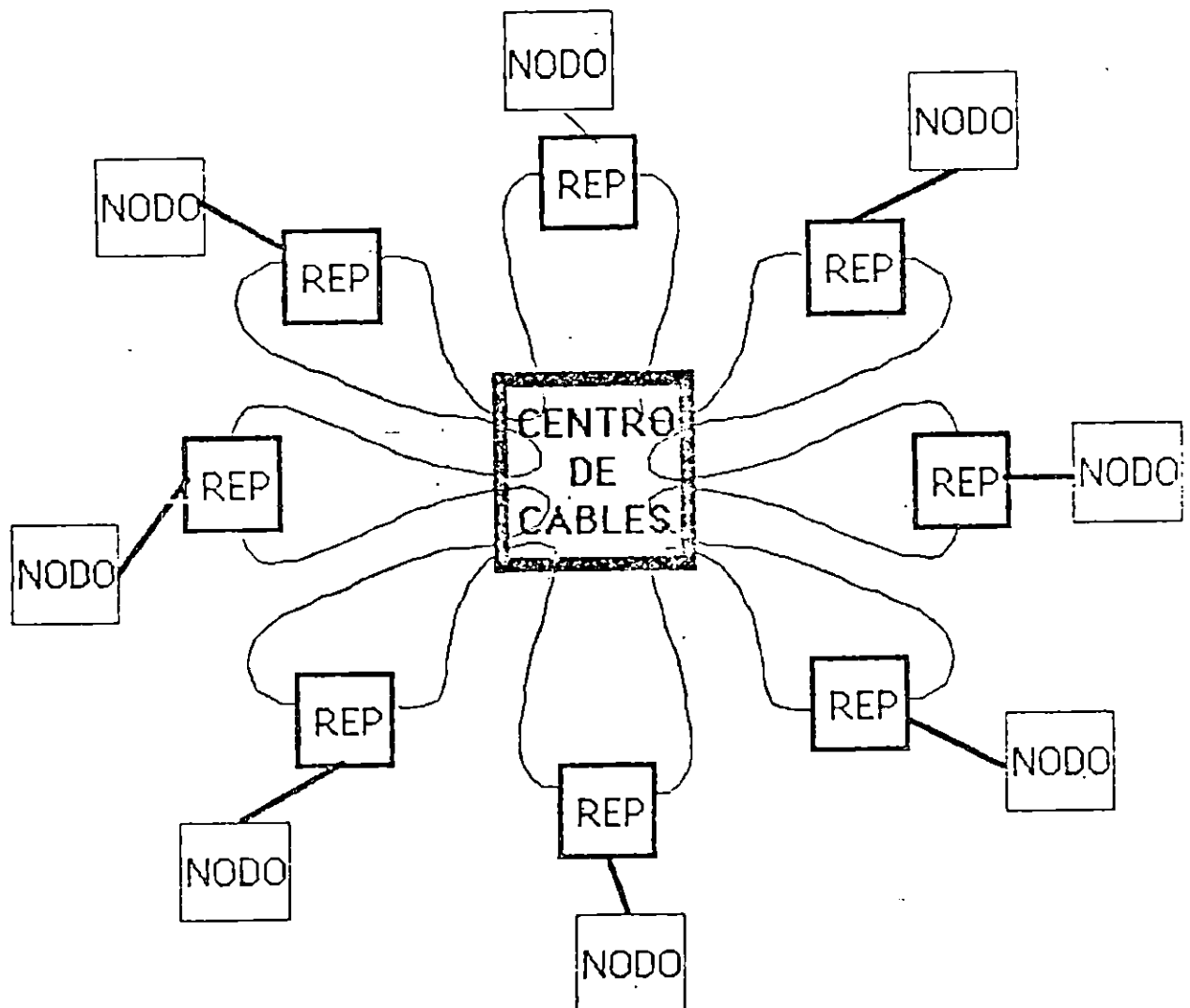
• Centro de Cables

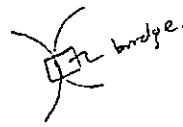
• Ventajas

- Fácil aislar estaciones entra algo y no se vuelve a salir
⇒ voy a tirar en el mtn. (eco)
- Soluciona bolsillo lleno de llaves
- Aislamiento en centro de cables
 - Distancia entre repetidores $\approx Kte$ ("puentando")

• Problemas

- Aunque simple, centro de cable puede fallar Centrale/ ⇒ negro!
- Mayor longitud de cable $\Rightarrow$ regeneración. slm:
E-A regenera



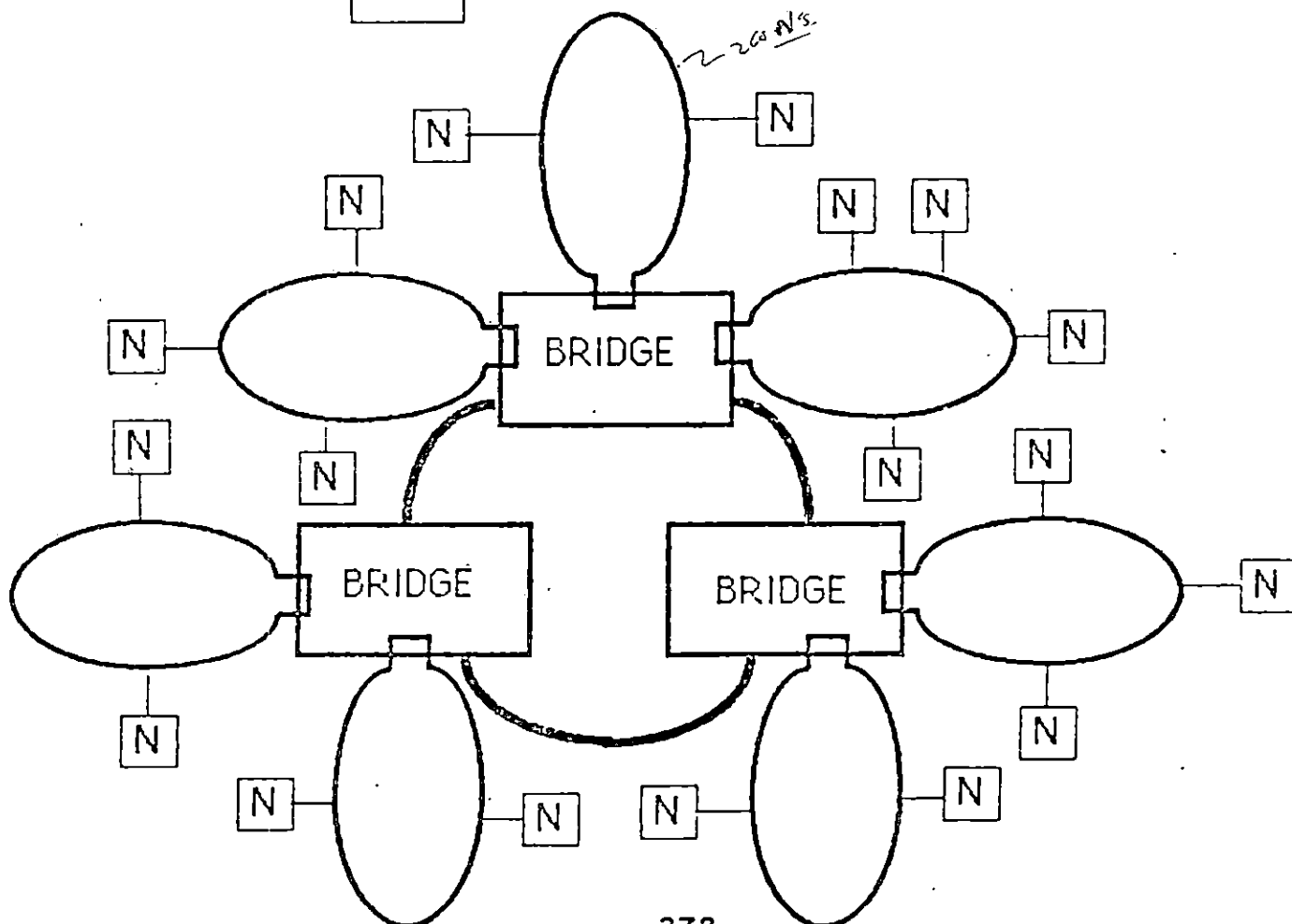
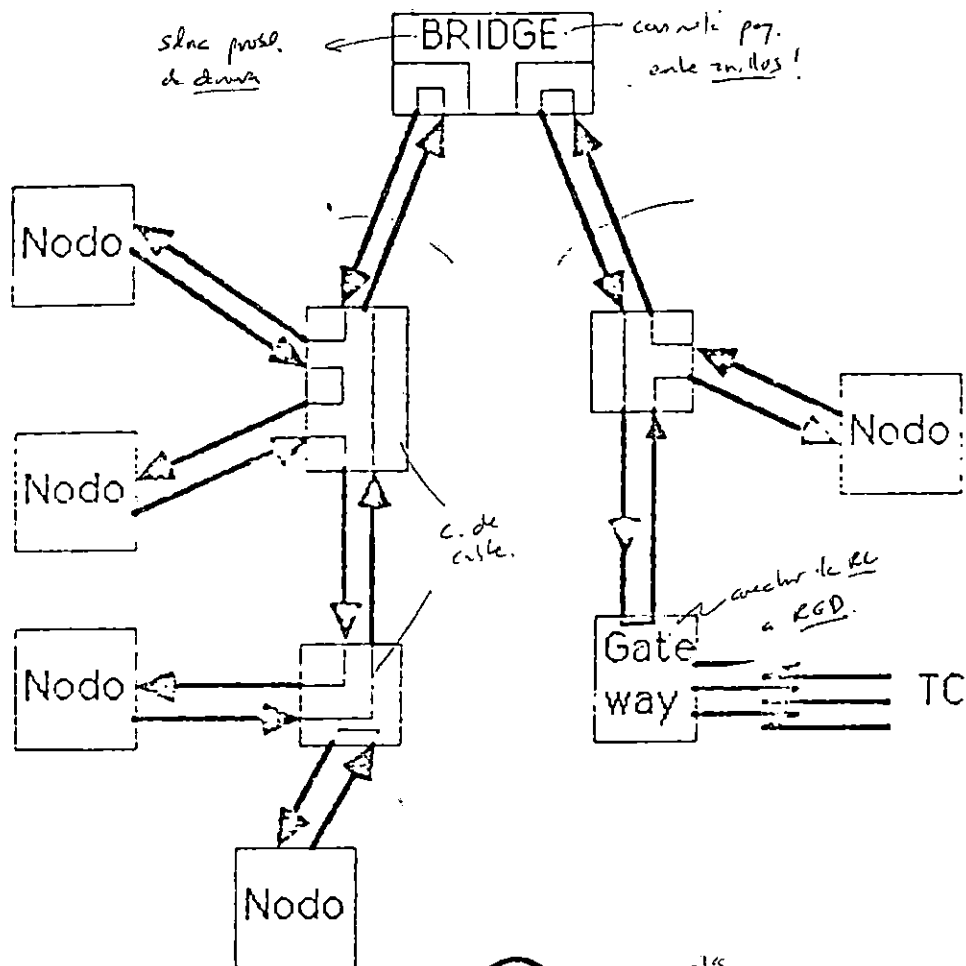


Estrella Anillo Jerárquica

"TOKEN Ring"

- La red se divide en varios anillos
- Esos anillos se ^{inter-}conectan a un ^{trás de} BRIDGE
- C/anillo tiene varios centros de cable (1 cable por c/anillo)
- El BRIDGE tiene como función:
 - Escoger tráfico interanillo
 - Colocarlo en buffer de entrada
 - Conmutar
 - Colocarlo en buffer de salida
 - Transmitir
 - Convertir velocidades (pueden haber anillos con vel $\neq$ 5.0).
- Varios bridges se pueden interconectar →
- Se debe minimizar tráfico interanillo c/anillo superior 200 nodos
- Ventajas
 - Mejora confiabilidad
 - Disminuye demora en c/anillo ----> redes...
- Desventajas
 - Se pierde posibilidad de confirmación...
 - Diseño más complejo
 - Se debe minimizar tráfico interanillo

↳ Anillo de bridges = protocolo
que → frames deben tener confirmación si hay tráfico interanillo

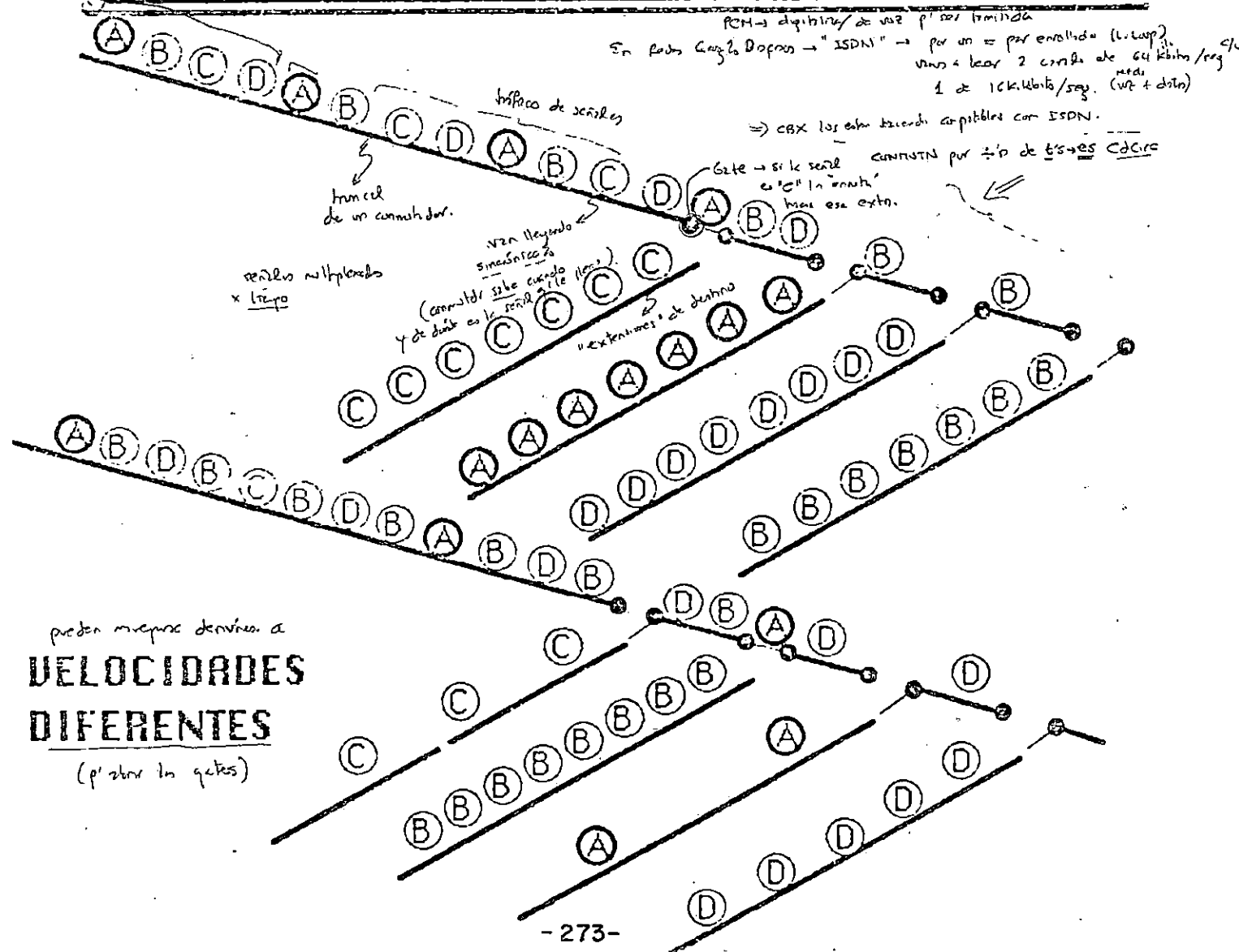


evolución de los PBX

CBX - Estrella

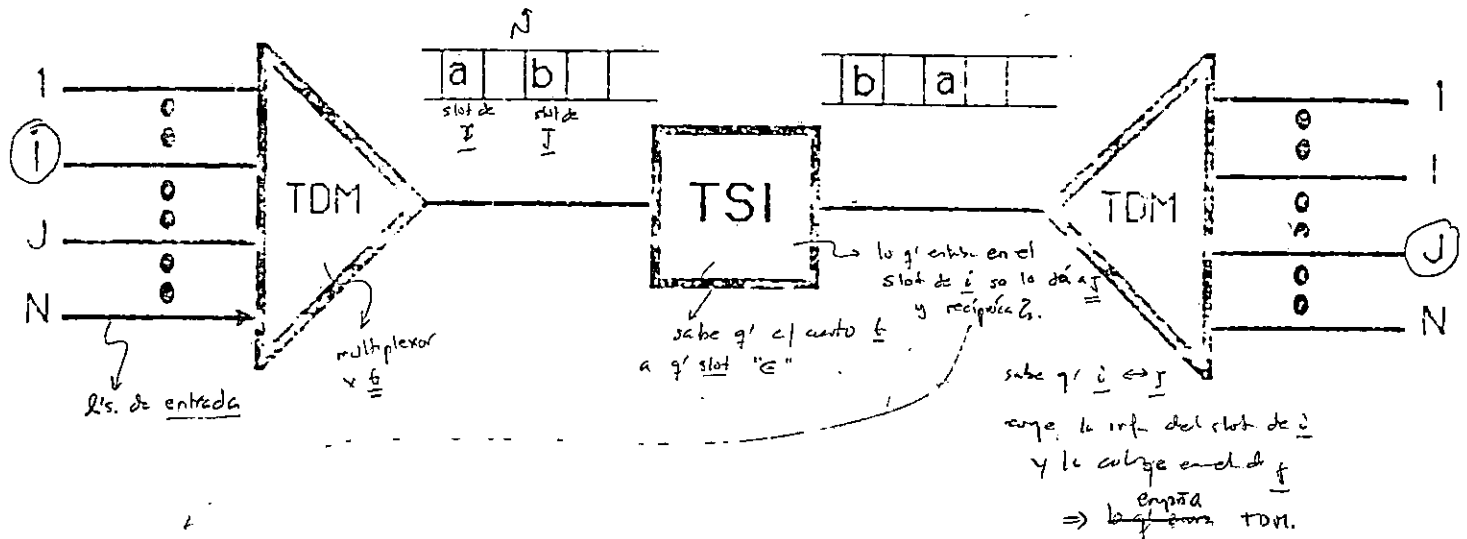
anuncios telef. digitales

- Computerized Branch Exchange
- Evolución de PBX
 - 1^{as}) PMBX = Private Branch Exchange ^{se telegrafista muelle de} _{chinos (publicos)}
 - PABX = Private Automatic Branch Exchange ^{conmutador telef. muelle}
 - PBX empiezan a usar conmutación digital
 - TDM  _{p. muelle - hora al día}
 - Time division switching
 - Maneja naturalmente líneas de velocidad...
 - Se implementa con TSI ---> Time Slot...



IMPLEMENTACION DE TSI (TIME SLOT INTERCHANGE)

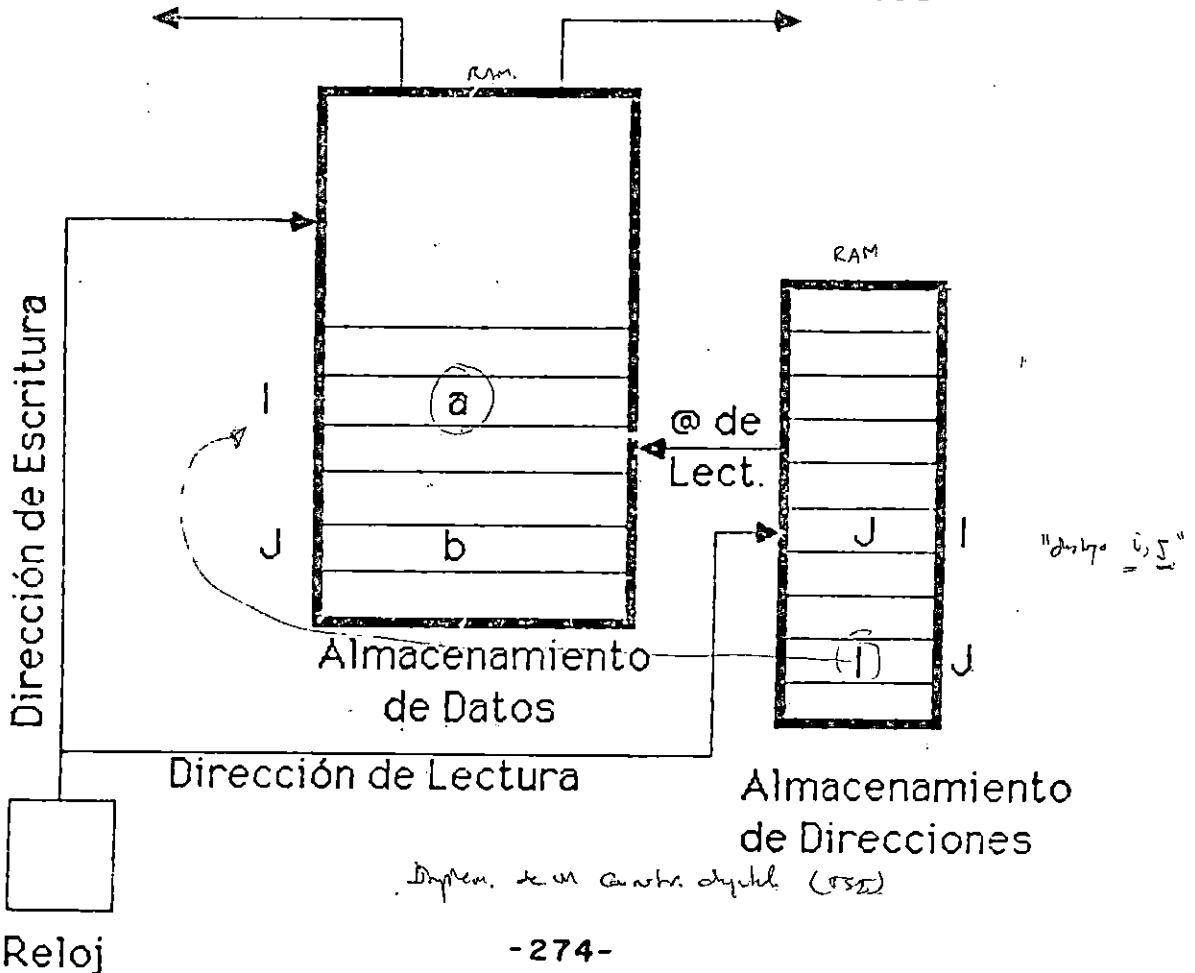
Plano multiplexador sincronizado.



"Interrupción":

Entrada de Datos

Salida de Datos



MAC → subcapa entre NF y NEnte
La OBT: control de acceso al 1/2 físico,
q' es compartido (acceso al bus,
etc.)

Protocolos

Modificación de OSI

Necesidad de Nivel de Red

MAC

Necesidad

Dónde?

ALOHA

precursor de ETHERNET

Bus/Arbol

CSMA ó LBT ^{origen: token de host}

CSMA/CD

Medium Access Control MAC

P. de C.A.M.

Slotted ^{origen}
de Anillo de Cambridge

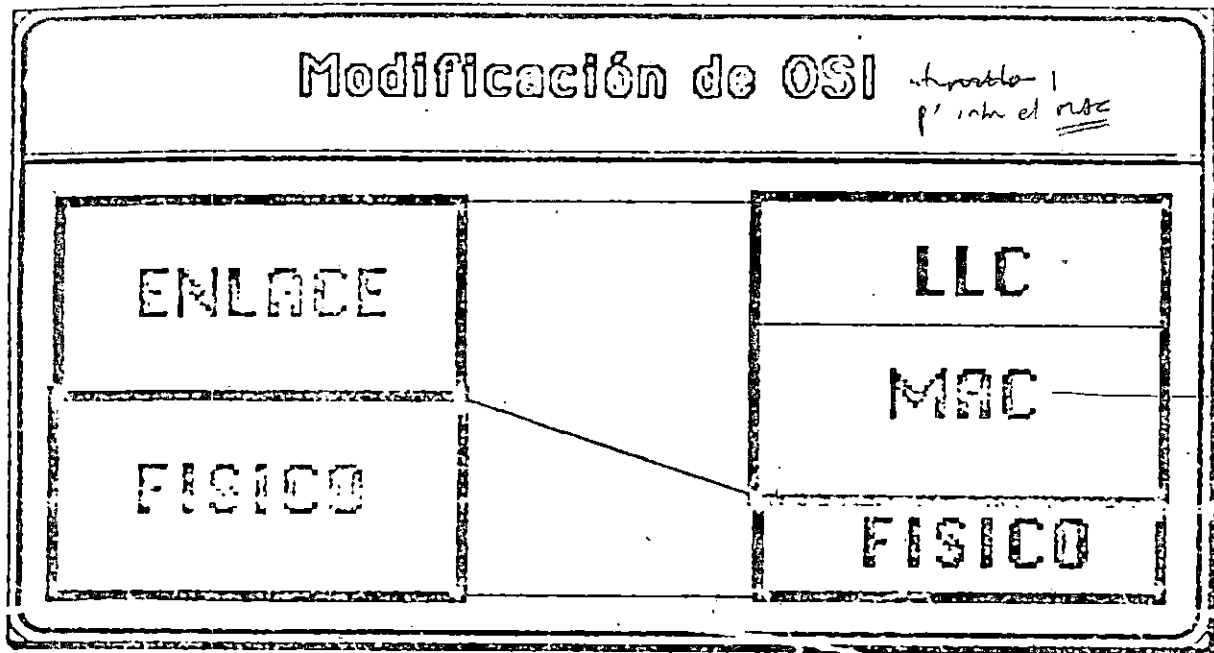
no! Register Insertion

Anillos

(SOM)
Token Ring

estándar:
IEEE 802.5

Token Bus



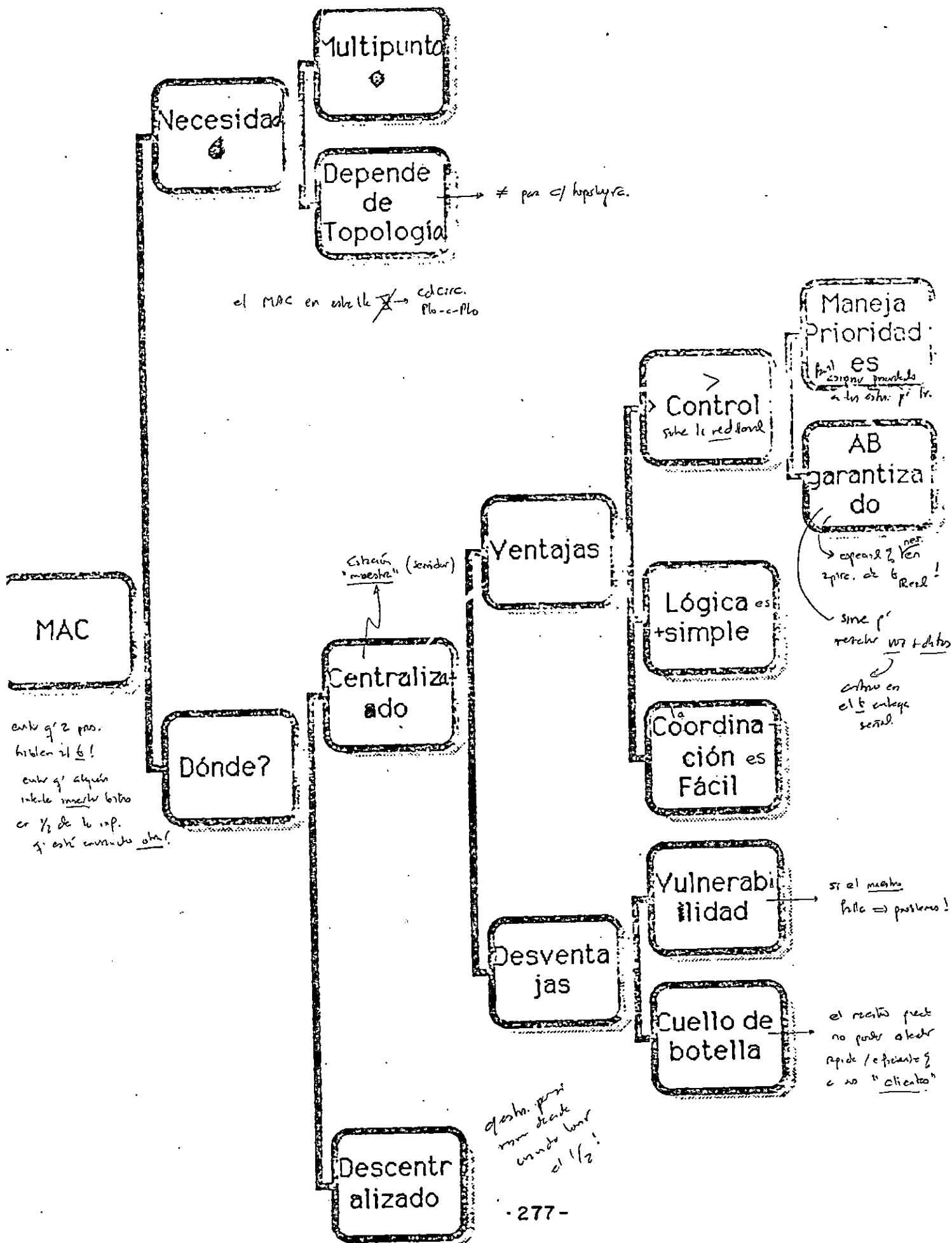
Necesidad del Nivel de Red en R.L.

si se req, es muy reducido

- No hay conmutación *excepto en T.R. ambiente bridge con bridge (R. Locales) hay conmutación en bridges*
- No hay enrutamiento *ambiente con*
- En Nivel 2 en un enlace físico
 - Control de Flujo
 - Control de Errores
 - Secuenciación
- Nivel 3 hace lo mismo en varios *entre*
- En LANs no hay sino uno *en un ambiente de inf. se maneja con un solo!*

R. Loc's.

*entre
excepto en bridge!*



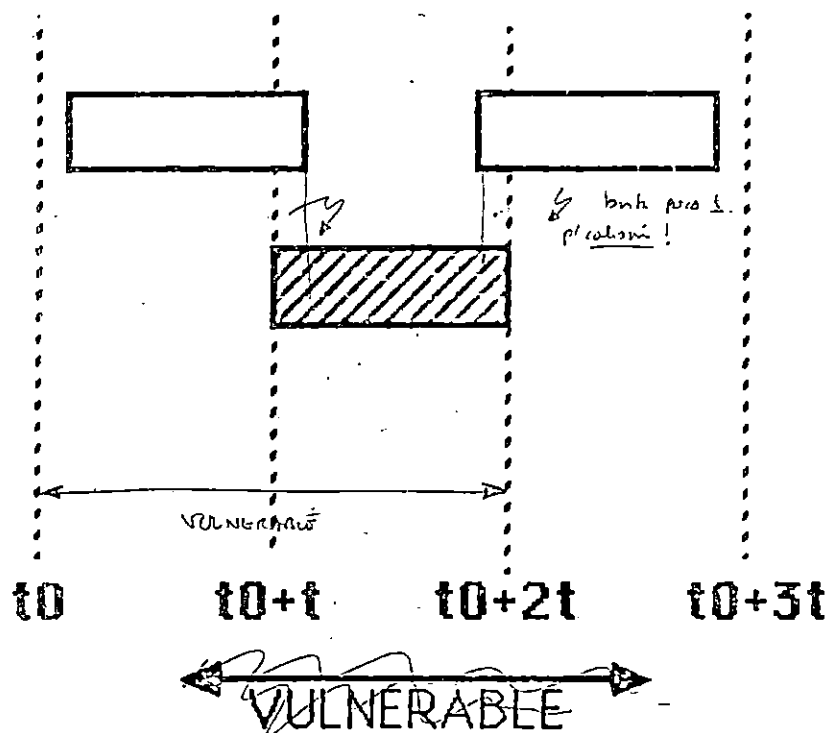
"ALOHA"

Las estaciones se comunican con radio al comp. central.

- Estaciones transmiten cuando quieren
- Oyen $2 \times t_{\text{máx}}$ de propagación p ver si les llega confirm. del otro lado.
- Si no ack entonces retransm después de t
- Aloha Puro Problema $\rightarrow$ t de vulnerabilidad de un paquete es $\uparrow$ $\rightarrow$ $p' < p$ / estm.

Utiliz máx del canal = 18%

- $\Rightarrow$ Slot
- Slotted Aloha: dividir el t en interv. discretos: solo se puede tr. el com de esos intervalos.
si periodo es com,
 que espere a q' termine el sigte intervalo.
 - Intervalos discretos
 - Utilización Máx del canal = 37%



chase, los paquetes o colisiones "completas" o no colisiones.

Síntesis $\rightarrow$ 1/2 de manchester

CSMA Ó LBT

Antes de iniciar, primero, y si
hay alguien hablando $\Rightarrow$ esperar!

envíe (frame)

ack = falso

REPITA

oigamedio(libre)

SI libre ENT

transmita

ack espere ($2 * t_{prop}$, ~~ack~~)

SINO

espere (t_1)

FSI

HQ ack

FIN envíe

solo tr. si el $\frac{1}{2}$ este libre

Colisión si 2 estac transmiten dentro de t_{prop}



tiempo de colisión es ∞ al t de propagación.

Utilización del canal

- Mejor que ALOHA *pero menos bueno si el $\frac{1}{2}$*
- Proporcional a long del frame *→ si frame fr. ∞ , aumenta el ser bit. de ser fr. este en la línea y a medida*
- Inversam proporc a t_{prop} *no lo que el $\frac{1}{2}$ $\Rightarrow$ rep uso del canal*

ETAPAS: *captó y en tr. intermedio (R. peg)*
ack en tr. bitoh (C. peg)

< tiempo p. tr. en peg y de.

Cómo escoger t_1 ?

→ cuando el canal está ocupado ¿qué?

- Non Persistent
- 1 persistent
- p persistent

3 Variantes de CSMA:

(1) Non Persistent

Estamos sin generador y no se dispone a esperar

envíe (frame)

ack=falso

REPITA

oigamedio (libre)

SI libre ENT

transmita (frame)

SINO espere ($2 * t_{prip}$, Ack)

escoja (t_1 , distprob)

espere (t_1)

FSI

HQ ack

FIN envíe

Problema → puede generar interferencias
varias estaciones esperando y el canal libre.

PROBLEMA:

Varias estaciones esperando y canal libre

1 persistent

envíe (frame)

ack=falso

REPITA

oigamedio (libre)

SI libre ENT

transmita

≠. **SINO**

REPITA

oigamedio(libre)

HQ libre

⇒ transmita (frame)!

FSI

espere ($2 * t_{prop, ack}$)

SI no ack ENT

escoja (t_1 , $distprop$)

espere (t_1)

FSI

HQ ack

FIN envíe

PROBLEMA: Estaciones "egoistas"

→ listas p/ entrar en el medio en
q' no haya q' desocupa el $\frac{1}{2}$.

p persistent

envíe (frame)
REPITA
oigamedio (libre)X
SI libre ENT
evento (p,trans) *cara y sello → si em " ⇒ fr.
si sello ⇒ no fr.*
SI trans ENT
transmita(frame) *⇒ callosi lmben*
espere (2* tprop, ack) *n estres. apen
no > 4 ⇒
callosura!*
SINO
espere (tprop) *f asignado $\frac{1}{2}$
d. n. r. 2?*
FSI
FSI
HQ ack
FIN envíe

REPITA

SI libre ENT

car y sello $\rightarrow$ si "car" $\Rightarrow$ tr.
si sello $\Rightarrow$ no tr.

```
transmit(frame)
```

$\Rightarrow$ Calvin haben

n estres. aprox 46, y,
np $> 1 \Rightarrow$
columnas!

espere (tprop.)

f. Angewandte Ethik
Diskussion

FSI

FIN envíe

p persistent (cont'd)

PROBLEMA:

Elección de p delicada para estabilidad

Si n estaciones quieren transmitir
y el canal está ocupado

cuando se desocupe

$n * p$ estaciones transmitirán

Si $n * p > 1$ ent $\rightarrow$ $\text{throughput} \rightarrow 0$! (el $\frac{1}{n}$ crece

colisión garantizada

de q'
espera!)

después de t_{prop}

más estac querrán transm

más colisiones

throughput $\rightarrow 0$

Si p muy pequeño

demoras innecesarias

CSMA/CD^{edición}

Lo es para estar oyendo a ver si se detecta

una colisión y si pasa, nos para de transmitir

- CSMA tiene una ineficiencia :

$$t_{tr} \gg t_{prop.}$$

- Si hay colisión
transm continúa hasta final del frame

- Desperdicio de AB

- Medio No Usable
~~Riesgo de colisión hasta que último bit termina de propagarse~~

- CSMA/CD oye mientras transmite

- Si detecta colisión
para inmediatamente.

Qué Persistencia Escoger?

SORPRESA:

La más usual es 1-persistent
Ethernet, Mitrenet, IEEE802.3

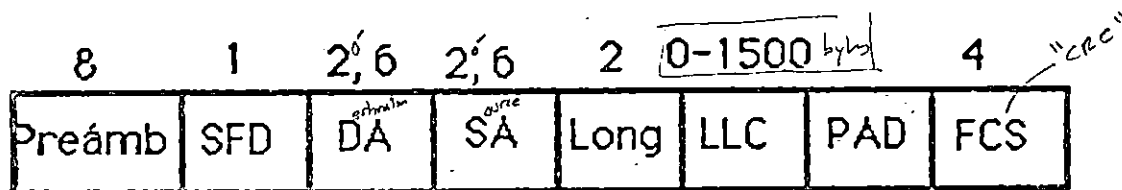
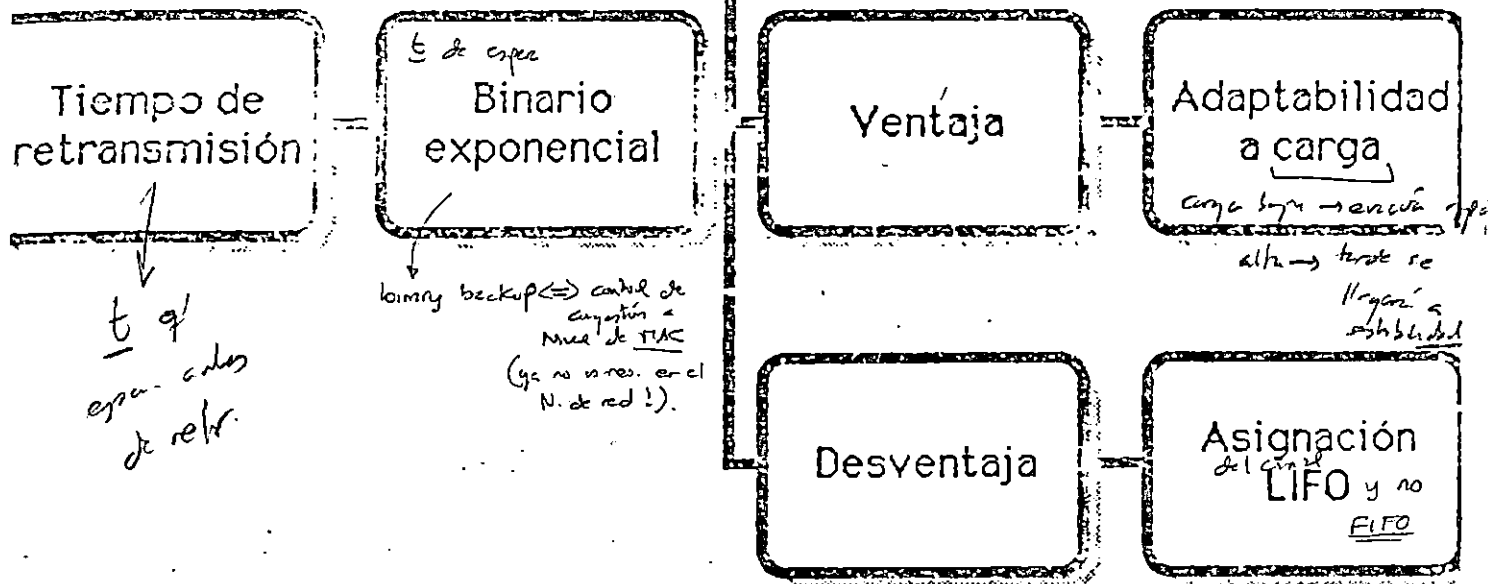
POR QUÉ?

t perdido en colisiones bajo si $T_{trans} \gg T_{prop}$
Si 1er bit ^{se} ha propagado completamente
no hay riesgo de colisión

valor de t_p :
 si el buffer es $\downarrow$, no se el canal + rápido
 (ETHERNET). o.k.
 T.R. - 2.

$\neq$ permiti $\rightarrow$ desahorro de la red

oper como colmón $\Rightarrow$ 2* t_p $\rightarrow$ $\uparrow$ y $\uparrow$...



CSMA/CD

1) Ethernet (refer bus!)

IEEE 802.3

(Ethernet).

tipos de Ethernet $\rightarrow$ @ se reconoce x H.W.

Problema de seguridad $\rightarrow$ se puede falsificar un tipo de (virus"!).

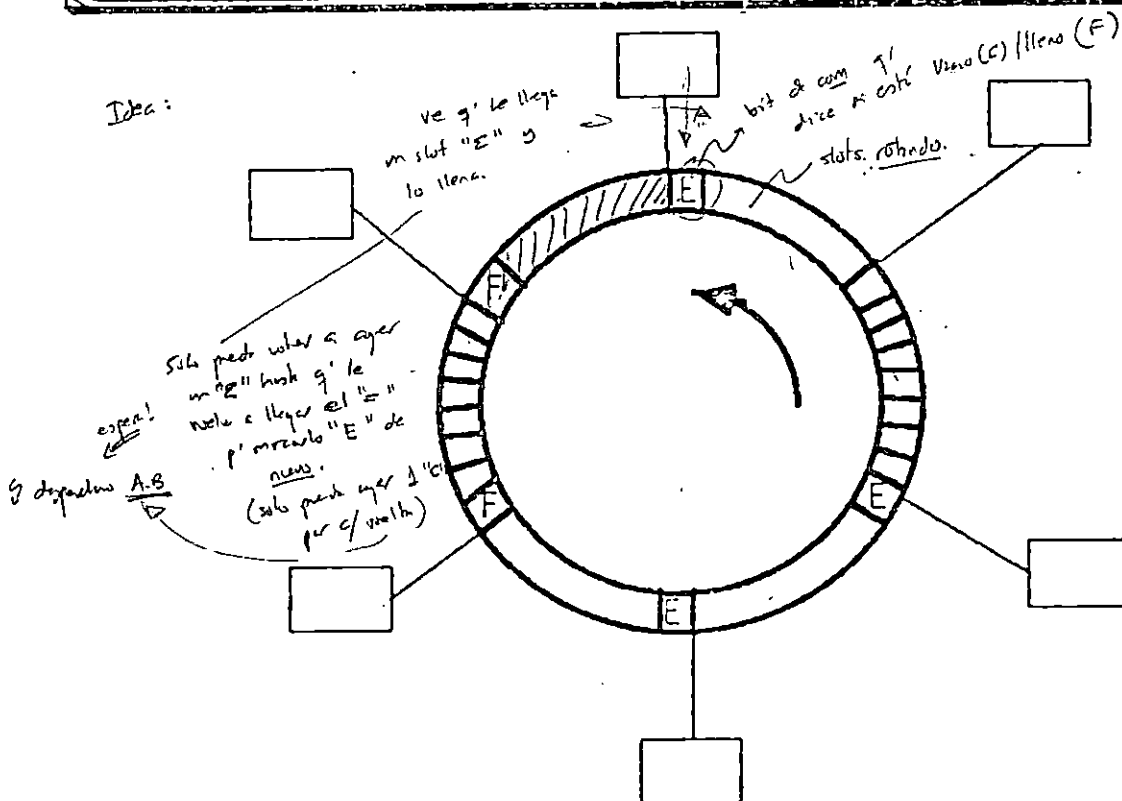
- AC = Access Control
- DA = Destination Address
- DSAP = Destination Service Access Point
- ED = Ending Delimiter
- FC = Frame Control
- FCS = Frame Check Sequence
- FF = Frame Format
- FS = Frame Status
- SA = Source Address
- SD = Starting Delimiter
- SFD = Start Frame Delimiter
- SSAP = Source Service Access Point

Slotted ^{ring}

ó Anillo de Cambridge

- Slots de long fija
- Para transmitir, la estación
 - Divide mens en pkcts con long máx de slot
 - Espera slot vacío
 - Lo marca como lleno
 - Inserta packt al paso de slot
 - No transm otra vez hasta que vuelva slot
 - Al llegar lo marca libre
- En anillo de Cambridge cada slot tiene
 - Un byte para @ fuente
 - Un byte para @ destino
 - 2 bytes de datos ! (2) *↑↑ inf & ctrl.*
 - Cinco bits de control
 - Total 37 bits
 - Overhead ≈ 50%
- Mal performance en bajo tráfico

*hace q' esperar
hasta q' llega al
= slot p' seguir tr.*

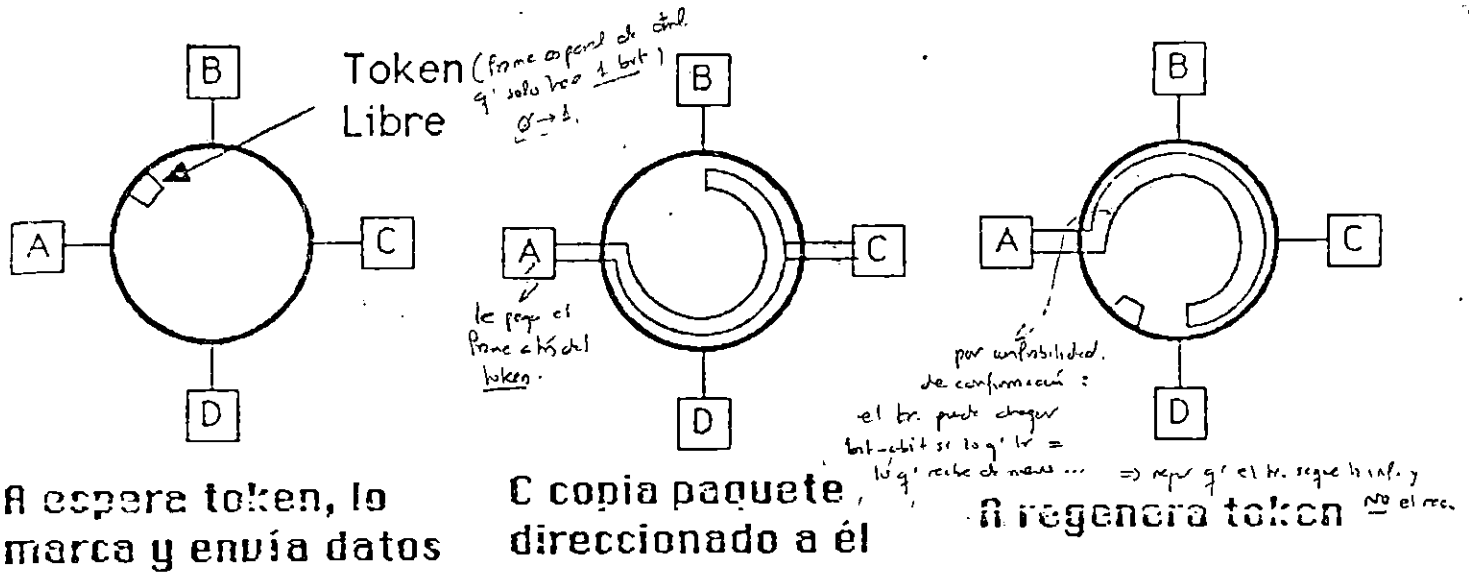


Capacidad del anillo

• Ejemplo

- V de transm = 10Mbps = $10 \times 10^6 \text{ bits/seg.} = 10 \times \frac{10^6 \text{ b}}{10^6 \mu\text{s}} = 10 \text{ b}/\mu\text{s}$
- V de prop = 200.000 km/s $\frac{2 \times 10^5 \text{ m}}{10^6 \mu\text{s}} = 200 \text{ m}/\mu\text{s}$
- Long anillo = 1.000 m
- 100 estaciones
- Se emiten 10 bits/ μs
- En un μs la señal avanza 200m y son 10b/ μs
- 1 bit ocupa 20 m !
- Caben 50 bits en el anillo (muy poco !!)
- En cada estación cabe 1 bit (!) $\rightarrow 100 \rightarrow 100 \text{ bits}$
en el km de cable. así
- Total = 150 bits $\approx 32 \text{ kbps}$
- Suficiente para 4 slots de Cambridge
- Qué pasa si se desactivan repetidores?
se ↓ la cap. del anillo

Token peak tr $\approx$ 4 Mbits/sec en 10 pseg $\Rightarrow$ puede tr 40 bits



Token Ring

Clase de curso el 1/2: "token" en un círculo por la red: el q' copia la ficha puede tr. dte. 10 mseg. (default).

Acorda el 1/2 $\rightarrow$ eliminación y no prohibición (ethernet) ; bueno p' recibir voz y datos!

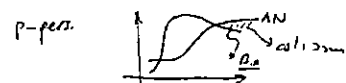
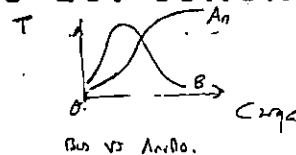
Desperdicia AB en tráfico bajo

Permite garantizar AB una parte sobre el tiempo "el token", se puede tr.

Problema: Mantenimiento del token

IEEE 802.5

- Formato del token
- Formato del Frame
- Generalidades



- Trasm regenera token al recibir ocupado
- Bits de prioridad
- Indicadores de reservación
- T máx de transm = Token Holder Timer
- Bits de ACK
- Mantenimiento del token (FC) $\rightarrow$
 - No token Timer (TNT) - hay un estm. de monitoreo q' se envía a mantenerlo
 - Valid Frame Timer (TVX) (se re cse, cuando un paquete p' q' (la recepción!) va a salir)
 - IBM tiene esquema centralizado
 - Monitor y bit monitor
 - Monitores activos y pasivos

FORMATO DEL TOKEN



delimiters → codifn. Manchester especial



paritas: El token lo copia los q' tienen > prioridad p' transmitir!

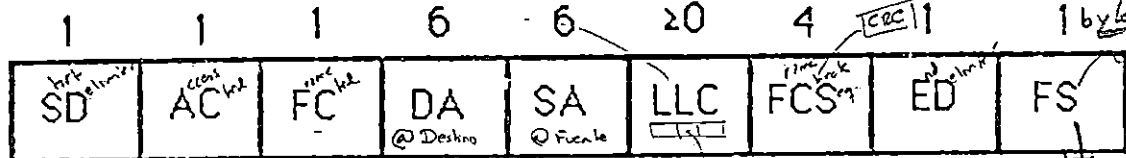
PPP = Prioridad (0-7)

T = Token

RRR = Reservación (0-7) → reservar un token con PPP h2!

original link local LLC → "Nivel de enlace" de estos datos

FORMATO DEL FRAME

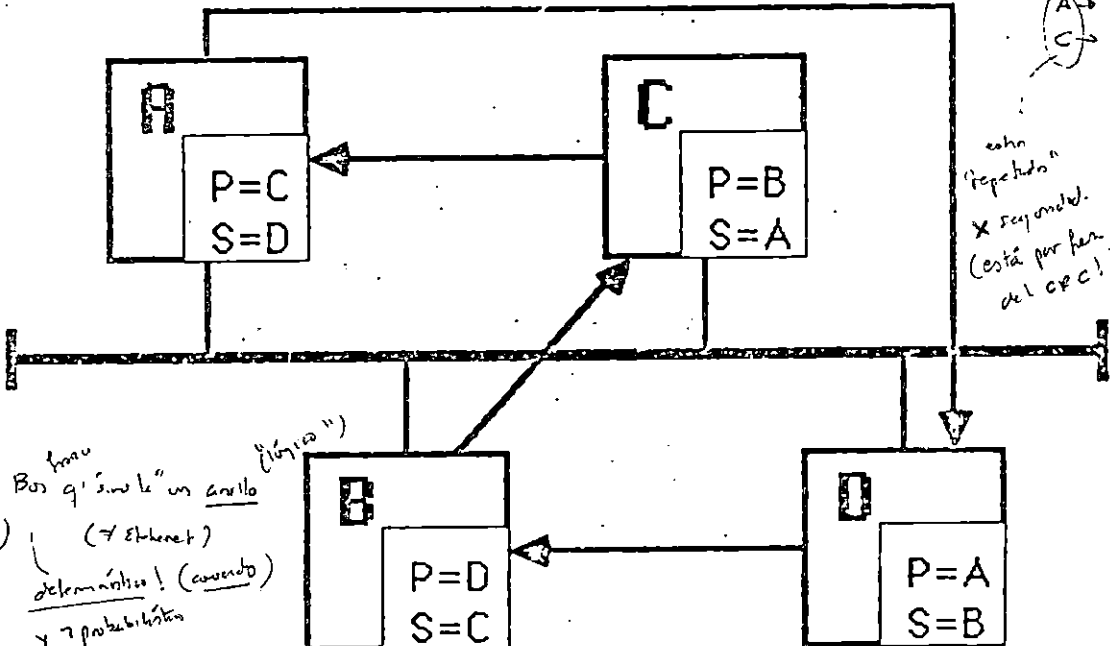


se inserta con inf al token

T. Ring → 2 bits de confirm.

FC → p' montar el token

TOKEN BUS

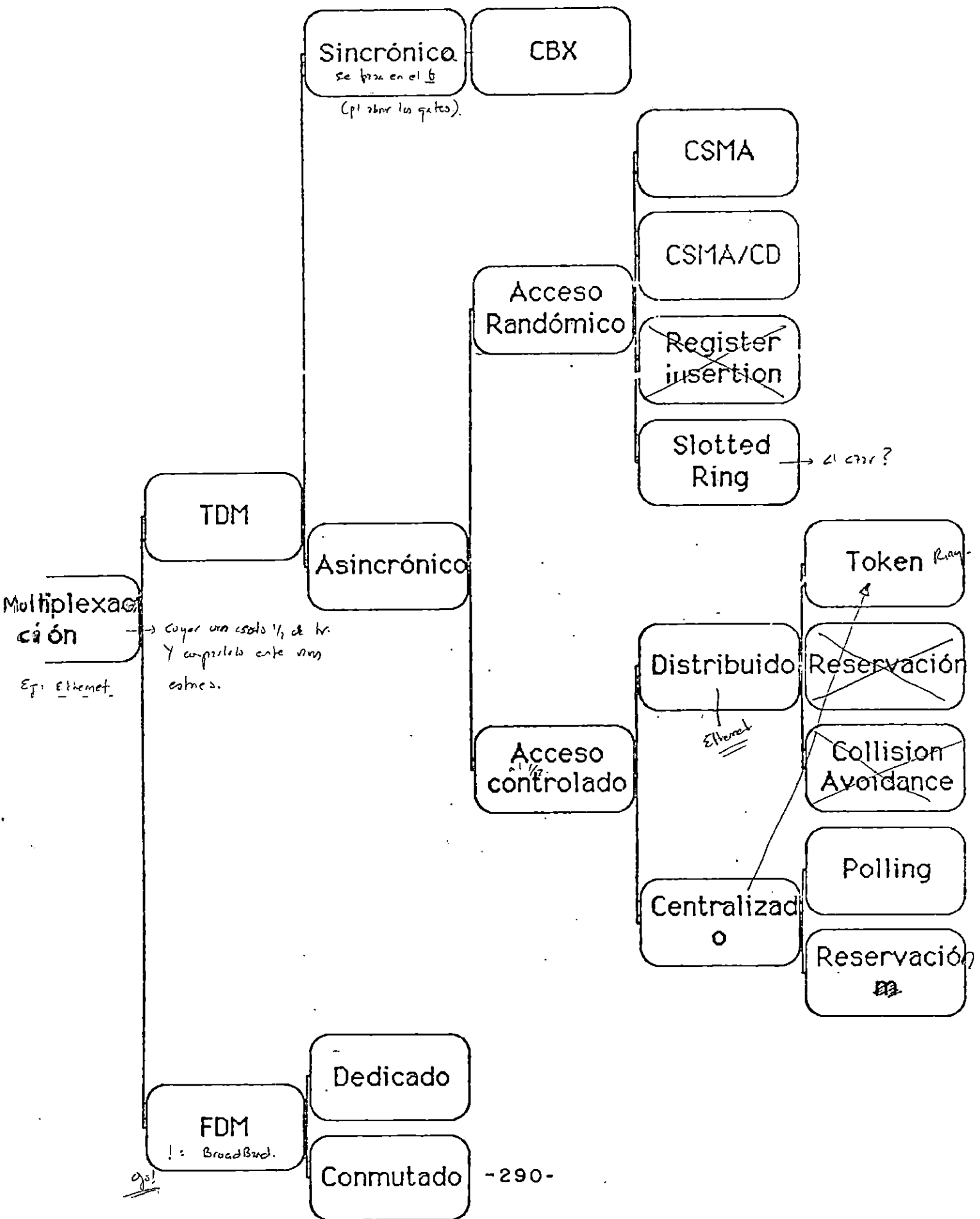


A → 1 si recon. dir. válida
C → 1 si copia el frame.

esta "repetida" X redundante. (está por fuera del crc!).

MAP → Bus físico "simple" un anillo (G. Metro) (≠ Ethernet) determinístico! (cuerpo) y 7 prioridades

P = Predecesor
S = Sucesor



Logical Link Control (LLC)

≡ Nivel de Enlace en R.Loslos -

- Debe soportar multiacceso
- Algunas funciones de nivel 3
- IEEE802.2 (LLC)

nombre estandarizado

802.3 → Ethernet

4 → Token Bus.

5 → Token Ring.

6 → Redes celulares

Algunas funciones de nivel 3

- Datagramas (connectionless Service)
- Simplemente incluir @s fte y destino
- CV (Connection Oriented Service)
- Establecimiento de conexión
- Basado en SAPs
- SAP análogo a # de CV de X25
- Multicast: Fácil por acceso múltiple

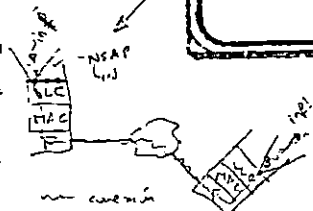
7 envíos
dir de
compañía

en el MAC
no a
nivel de red
(en todos
R.Loslos)

En T.Ring →
bien dir de compañía.
1 solo enlace
solución para W < 1/2 E.

envío de datos.

Pto de acceso al NdeEnlace.



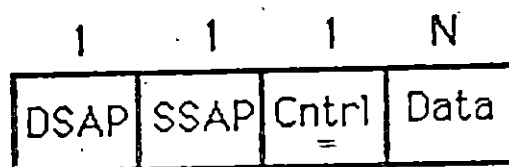
en X25 SAP ↔ # CV.

TSAP → conexión a nivel de
file

IEEE802.2 (LLC)

- Formato Frame
- Servicio al Suscriptor-Primitivas
- Protocolo-Funciones
- Interfaz con MAC

FORMATO DEL FRAME "LLC"



- 291 - bytes de dir.
de salida (#sec 1).

Servicio al Suscriptor-Primitivas

- Unacknowledged Connectionless Service
 - L-DATA.request
 - Nivel superior pasa frame a LLC
 - L-DATA.indication
 - LLC entrega frame a nivel superior
- Connection Oriented Service
 - Establecimiento de conexión
 - L-DATA-CONNECT.request *call req.*
 - L-DATA-CONNECT.indication
 - L-CONNECT.confirm *call accepted*
 - Finalización de conexión
 - L-DISCONNECT.request *clear req.*
 - L-DISCONNECT.indication
 - L-DISCONNECT.confirm *clear confirmed*
 - Reinicialización de conexión
 - L-RESET.request *reset.*
 - L-RESET.indication
 - L-RESET.confirm
- En Estudio
 - Acknowledged Connectionless Service
 - Management Service

Protocolo-^{de LLC}Funciones

≡ SDLC, HDLC pero con números #5

• Tipos de Frames

• 1) de Transferencia de Información

- Numeración y acks módulo 8

• 2) de Supervisión

- Receive Redy Ack en otros casos no hay info. de igno
- Receive-not-Ready
- Reject

• 3) de No-numerados

- Datagramas (UI) no numerados
- Establecimiento de conexiones (SABM) → call req.
- Terminación de conexiones (DISC)
- Prueba de la conexión (TEST)
- Confirmación no numerada (UA)
- Responde a SABM y DISC

Servicio o interfaz → interno cuando se va a usar un det. num. (llamada, etc.)

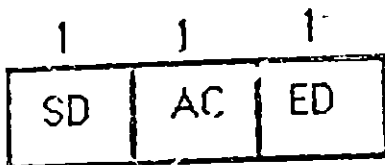
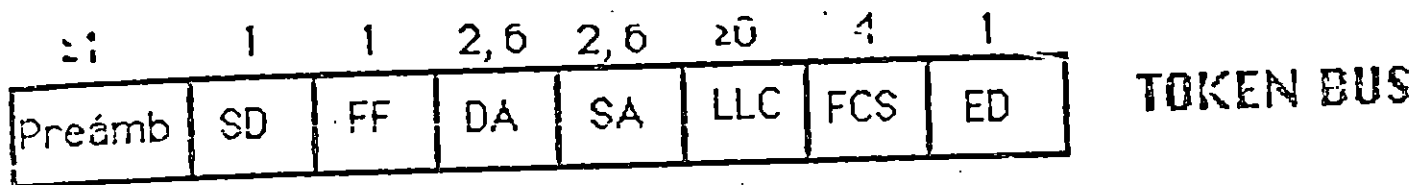
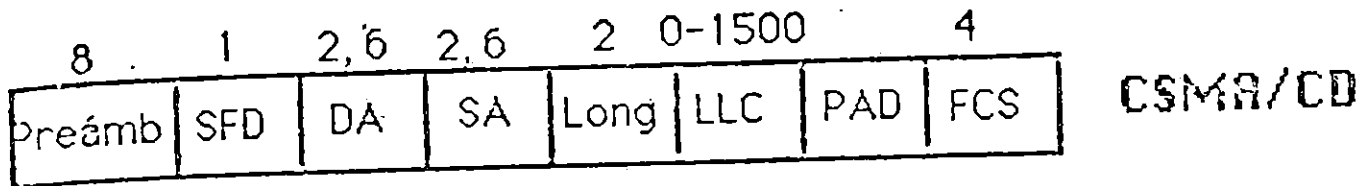
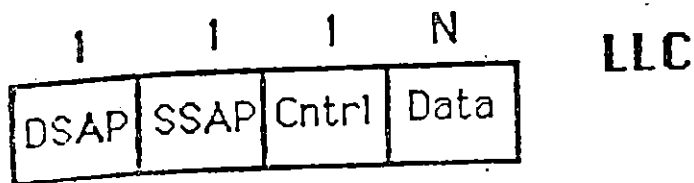
protocolo → se refleja en la info de chd y seq x la red. (↑ rec, ack's, etc.)

./.

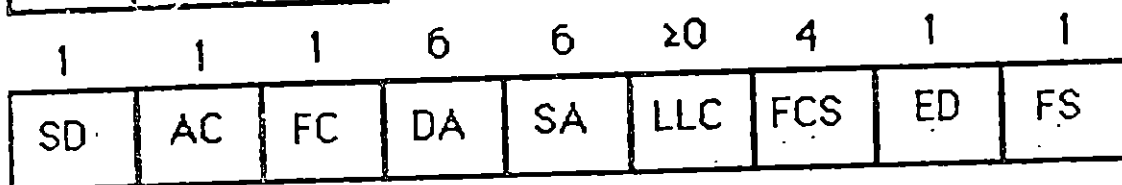
Interfaz^{de LLC} con MAC

- Trabaja con cualquier MAC y os + por c/ topología
 - CSMA/CD
 - Token Bus
 - Token Ring
- IEEE da modelo; no interfaz específica
- Primitivas (es ^{propio} Parce estable una conexión!).
 - MA-DATA.request
 - LLC solicita a su MAC servicio
 - MA-DATA.confirm
 - Respuesta de MAC local
 - No tiene sentido end-to-end
 - MA-DATA.indicate
 - MAC local avisa a LLC local que tiene frame
 - ~~MA-DATA.indicate~~

FORMATOS DE FRAMES (resumen).

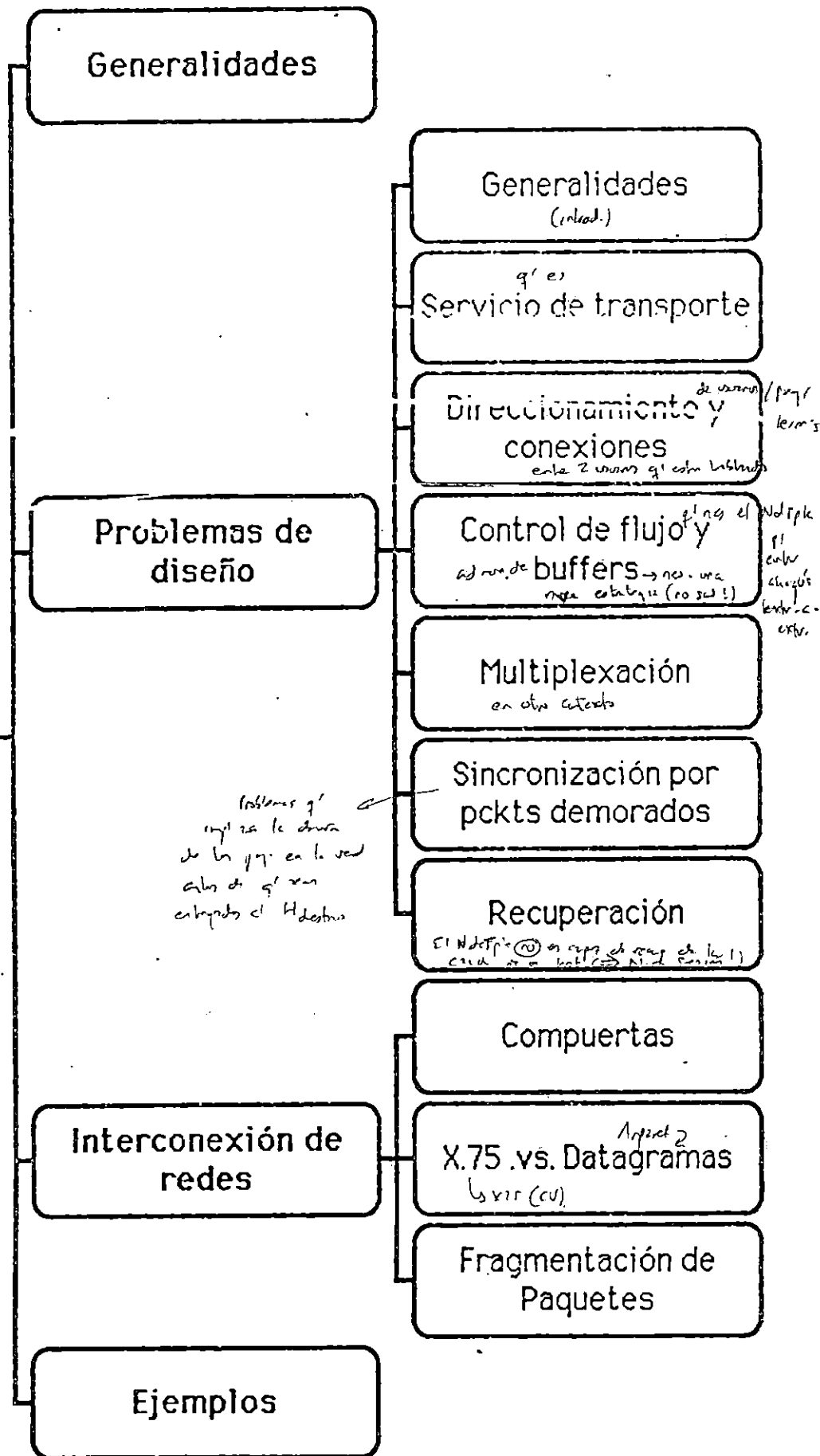


TOKEN RING



- AC = Access Control
- DA = Destination Address
- DSAP = Destination Service Access Point
- ED = Ending Delimiter
- FC = Frame Control
- FCS = Frame Check Sequence (CRC)
- FF = Frame Format
- FS = Frame Status
- SA = Source Address
- SD = Starting Delimiter
- SFD = Start Frame Delimiter
- SSAP = Source Service Access Point

VI. NIVEL DE TRANSPORTE



Niveles I, II y III

- Comunicación de máquinas adyacentes
- Corren en hosts e IMPs

Nivel de transporte

- ≠ Extremo-a-extremo (entre hosts) protocolo q' corre !2 en los hosts, la red ya no intere (no ac en DTG's).
- Comunica procesos de usuarios entre ⇒ nivel lógico.
- Corre sólo en hosts, ya no en IMPs (nodos i, 2, 4)
- Incluye funciones para (si no posibles de)
 - Direccionamiento de usuarios Nivel de sesión
 - Establecimiento de conexiones Programa q' invoca el N° de pte
 - Terminación de conexiones
 - Administración de buffers +
 - Control de flujo host-host
 - Multiplexación
 - Recuperación de errores
 - Sincronización x pags demandados

→ hacer ver la red como un tubo virtual. de com. entre 2 host en la red.

se encarga de "enviar" los datos de tecnología (hace tipo la com. a D's en niveles inferiores !).

es: UUCP (VAX),
transm. archivos VAX-VAX entre cualquier pte del mundo.
CU → envía de term. en UNIX.

Modelo interno afecta complejidad

- C. V. ---> + simple
- S. E. usuarios pueden desconfiar sin embargo
- Ejemplo: RESET en X.25
- DTG ---> + complejo la red no garantiza
 - Ordenamiento
 - Pérdidas
 - Duplicados

OSS → Redes A' (Locales) -

B (X.25)
con interfaz de CV
para intera con
redes y hosts.
(q' CV 1).

E (redes con interfaz de DTG !)

TCP/IP (Arg. Net)

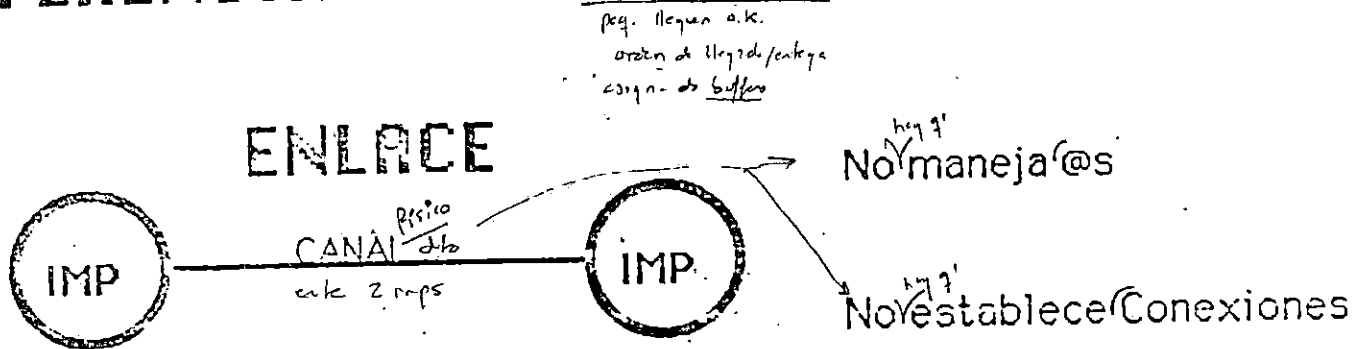
N° de pte. N° de red
Redes "C"

de los bucles de la red de CV

D < 2 de errores, pte.

N° de + C!

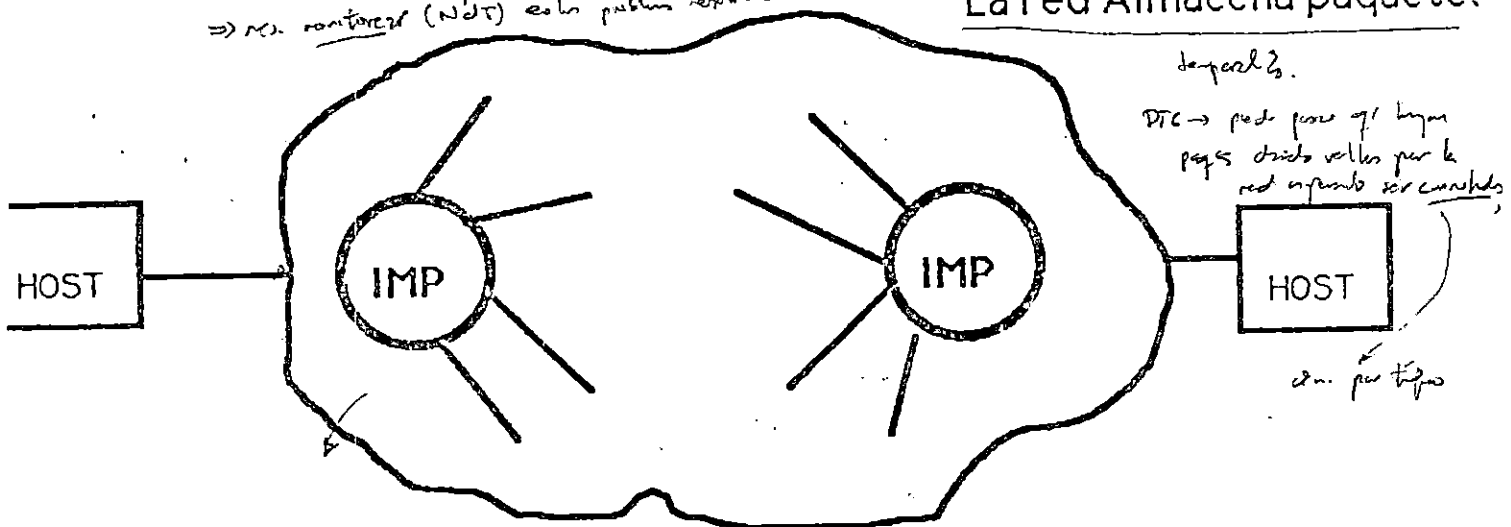
DIFERENCIA ENTRE ENLACE Y TRANSPORTE



DIFERENCIA: EN TRANSPORTE LA RED ESTÁ DE POR MEDIO ! , no un canal de comunicación.

NDE → chequeamos entre mensajes y/o
pero los pac. no están exentos de
los problemas de comunicación.
muchos veces hay error de conexión por descarte de pacq's.
" " " error de flujo p' error de flujo
⇒ res. monitoreo (NDE) es la solución a estos problemas

La red Almacena paquetes:



*problema → no se ocupa del problema de la red.
NDE.*

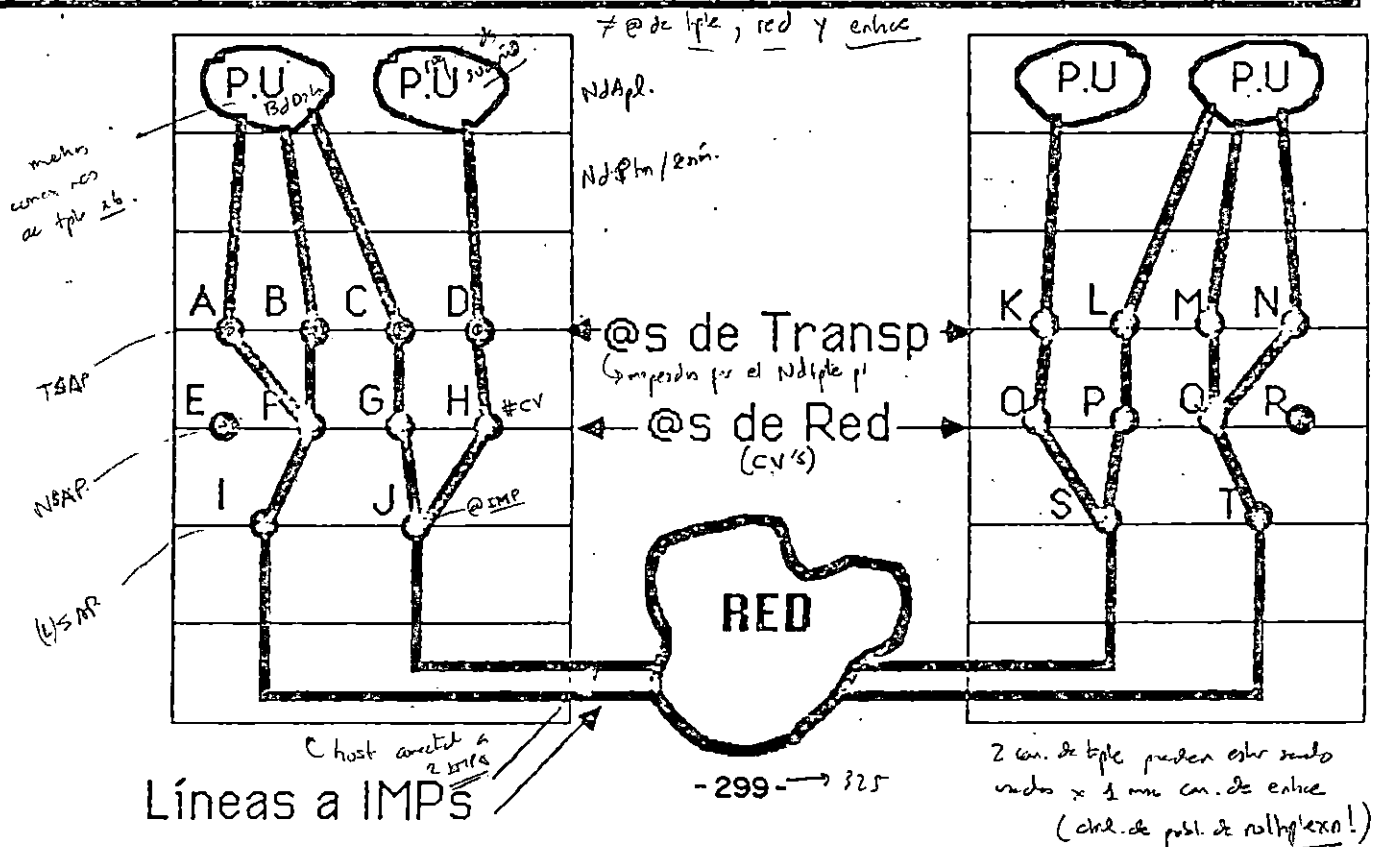
TRANSPORTE

Servicios de transporte

- Niveles 1 a 4 proveen servicio de transporte
- S.O. provee llamados o librería
- Cambios en la red → *no debe afectar la interfaz entre el N de transporte y los niv. sup. por sí el N de transporte*
- Se define un conjunto de @s de transporte *pl*
- Usuario pide conexión de transporte *exte. puede conectar*

Se define un conjunto de @s de Transporte

- # red & #host & #puerto
- ARPANET
 - 32 bits para #red y #Host
 - y 16 bits para #puerto *"proceso" / "puerto físico"*
- CCITT
 - 4 dígitos decimales para #red *host*
 - 10 dígitos decimales para red+puerto



Ejemplo

- Generalidades
- Funciones
- Interfaz con nivel de red
- Estructuras de datos
- Macro de establecim de conex
- Control de flujo

Generalidades

General de tipo ^{grande} S # cv.

esta de tipo = misma problem:

- Monousuario → 2 nivel de concurrencia, etc.
- Mensajes de cualquier long
- ~~Red~~ divide mensajes en pckts
- Trabaja con interfaz X.25

Ndt. ÷ mje en pckts → Nalred.

Nalred q'
lo q' hay algo
es una red X.25

Funciones

- o connum = CONNECT(local, remoto)
 - o Establece conexión
 - o Si OK, devuelve # de conex en connum
 - o Si fracaso # negativo en connum
 - o cuando:
 - o @local o remota inválida
 - o Host remoto caído
- o connum = LISTEN(local) → estoy dispuesto a establ. conexiones !
 - o Quien la llama, desea atender conexiones
- o status = CLOSE(connum)
 - o ^{cierra} Termina conexión de transporte
 - o Error si connum inválido
- o status = SEND(connum, buffer, bytes)
 - o Envía contenido de buffer por connum
 - o Buffer = apuntador al buffer de datos
 - o bytes = long del mensaje
 - o Errores:
 - o No hay conexión (con num 0)
 - o @ de buffer ilegal
 - o bytes < 0
- o status = RECEIVE(connum, buffer, bytes)
 - o Quien la llama desea recibir datos

may. local
se abre y
al recibir,
ret. en connum.



ult. pag. de un = $\pi \cdot r \cdot c (M)$

Interfaz con nivel de red

- FromNet y To Net con 6 parámetros
 - Identificador de conexión *después de q' con. se cierra el pág*
 - Bit Q para mensajes de control *control de datos → p' distinguir en Ndr, control de datos y control de cierre*
 - Bit M para mensajes multipkt *datos p' el Ndr pero control p' Ndr*
 - Tipo de pckt
 - Data, Call Request, Call Accepted
 - Clear Request, Clear Confirmation, Credit
 - Apuntador a los datos *para datos*
 - # de bytes de datos *p' indicar para q' control hay un Δ en la red, no hay res. de Δ Ndr. indicar la q' de byte de los*
 - Si ventana de red llena transporte a dormir *de hecho de replanear de la red*
 - EnableHost, DisableHost
 - Bit Q en 0 para datos y 1 para créditos *base de la fec. de datos de flujo*

pag. de red los pag. de cierre de Ndr).

Estructuras de datos

- Conn contiene el estado de la conexión:
 - @s de transporte local y remota
 - # de mensajes enviados y recibidos *por*
 - Estado actual *→ cerrado, ab, dormido, etc. en conexión*
 - @ del bffr de datos del usuario
 - Desplazamiento en buffer *dentro* *aviso mpe multipag.*
 - Bit indicador de close *aviso de con.*
 - timer para establecimiento de conex
 - Contador de Créditos *→ 5 datos - 7 se con. LISTEN.*

1 req. por y conex. estable.

Macro de establecim de conex

Usuario llama a CONNECT *pl. establ. la con.*

Nivel de red envía CALL REQUEST

Al otro lado al recibir CALL REQUEST

Si Estac de transporte ^{ha} llamó a LISTEN? ent

Red genera CALL ACCEPTED

sino

CALL REQUEST entra a timeout

Si en ese período ~~se~~ host hace LISTEN ent

Se establece la conexión

sino

Red envía CLEAR REQUEST

Control de flujo

- Receptor envía ^{debe} créditos ^{pag.} a transm
- 1 crédito = 1 mensaje ^{y no} _{a la pag.}

g: mpe my gde => a lugo!

*solo cuando llega
un crédito puede env
un mpe*

EJEMPLO DE UNA ESTACION DE TRANSPORTE SIMPLE

program estransp;

```
const MaxConn=...; MaxMsg=...; MaxPkt=...;
      TimeOut=...;
      q0 = 0; q1 = 1;
      m0 = 0; m1 = 1;
      cred=...;
      OK = 0;
      ErrFull = -1; ErrReject = -2; ErrClosed = -3; LowErr = -4;
```

```
type bit = 0..1;
      TransportAddress = integer;
      ConnId = 1..MaxConn;
      PktType = (CallReq, CallAcc, ClearReq, ClearConf, DataPkt, credit);
      CState = (idle, listening, waiting, queued, open, sending,
                receiving, closing);
      message = array[0..MaxMsg] of 0..255;
      msgptr = ^message;
      ErrorCode = LowErr..0;
      ConnIdOrErr = LowErr..MaxConn;
      PktLength = 0..MaxPkt;
      packet = array[PktLength] of 0..255;
```

```
var ListenAddress: TransportAddress;
     ListenConn: ConnId;
     data: Packet;
     conn: array[ConnId] of record
       LocalAddress, RemoteAddress: TransportAddress;
       MsgsSent, MsgsReceived: Integer;
       state: CState;
       UserBufferAddress: msgptr;
       ByteCount: 0..MaxMsg;
       ClrReqReceived: boolean;
       timer: Integer;
       credits: integer;
     end;
```

```
function listen (t: TransportAddress): ConnIdOrErr;
var i: integer; found: boolean;
begin i:=1; found:=false;
  while (i <= MaxConn) and not found do
    if (conn[i].state = queued) and (conn[i].LocalAddress = t)
    then found := true
    else i := i+1;
  if not found then begin ListenAddress := t; isleep; i := ListenConn; end;
  conn[i].state := open; conn[i].timer := 0;
  Listen := i;
  ListenConn := 0;
  TONet(i, q0, m0, CallAcc, data, 0)
end;
```

```

function connect(x, r: transport address): conn-obj + int;
begin
  i := MaxConn;
  while (conn[i].state < idle) and (i > 1) do i := i - 1;
  with conn[i] do
    begin
      MsgReceived := 0;
      LocalAddress := x;
      CtrReqReceived := false;
      Credits := 0;
      Timer := 0;
      Conn := ErrReject;
      State := Idle;
      ToNet(i, qd, md, ClearConn, data, d)
    end
  end
  else Conn := ErrFull
end;

function send(cid: connid; bufptr: msgptr; bytes:
var i: count; integer; m: bit;
begin
  with conn[i] do
    begin
      State := sending;
      CtrReqReceived := 0;
      Credits := 0;
      if (not CtrReqReceived) and (Credits = 0) then skip;
      if not CtrReqReceived then
        begin
          repeat
            if bytes - Byte(count) > MaxPkt
              then begin count := MaxPkt; m := 1 end
            else begin count := bytes - Byte(count); m := 0 end
            for i := 0 to count - 1 do data[i] := bufptr[bytecount + m];
            ToNet(i, qd, md, data, count);
            ByteCount := ByteCount + count;
            until ByteCount = bytes;
            MsgSent := MsgSent + 1;
            Credits := Credits - 1;
            send := OK
          end
        else send := Err(Closed);
        state := open
      end
    end
  end;
end;

```

```

function receive(cid: ConnId; bufptr: msgptr; var bytes: integer): error code;
begin
  with conn[cid] do
    begin
      if not ClrReqReceived then
        begin
          state := Receiving;
          UserBuffer Address := bufptr; Byte Count := 0;
          data[0] := cred;
          To Net (cid, q, m, cred, data, 1);
          shuf;
          Msgs Received := Msgs Received + 1;
          buf := Byte Count
        end
      if ClrReqReceived then receive := ErrClosed else receive := ...;
      state := open
    end
  end;
end;

```

```

function close(cid: ConnId): Error Code;
begin
  with conn[cid] do
    if ClrReqReceived then
      begin
        state := idle;
        To Net (cid, q, m, clear (conf, data, d))
      end
    else
      begin
        state := closing;
        To Net (cid, q, m, clear Req, data, d)
      end
    end;
  close := OK
end;

```

```

procedure Packet Arrival;
var
  cid: ConnId;
  q, m: bit;
  ptype: Pkt type;
  data: packet;
  count: Pkt Length;
  i: integer;

```

```

begin
  From Net (cid, q, m, ptype, data, count);
  with conn[cid] do
    case ptype of
      call req:
        begin
          Local Address := data[0]; Remote Address := data[1];
          if Local Address = Listen Address
            then begin Listen Conn := cid; state := open; wake up end
            else begin state := queued; timer := timeout end;
          Msgs Sent := 0; Msgs Received := 0;
          ClrReqReceived := false; credits := 0
        end
    end;
  end;

```

```

begin
  state := clean;
  waveup;
end;

Clear Reg:
begin
  Clear Recv := free;
  if state = cleaning then state := idle; & collision envelope
  if state in waiting, sending, then waveup;
end;

Clear Count:
state := idle;

credit:
begin
  credit := (credit + data[φ]);
  if state = sending then waveup;
end;

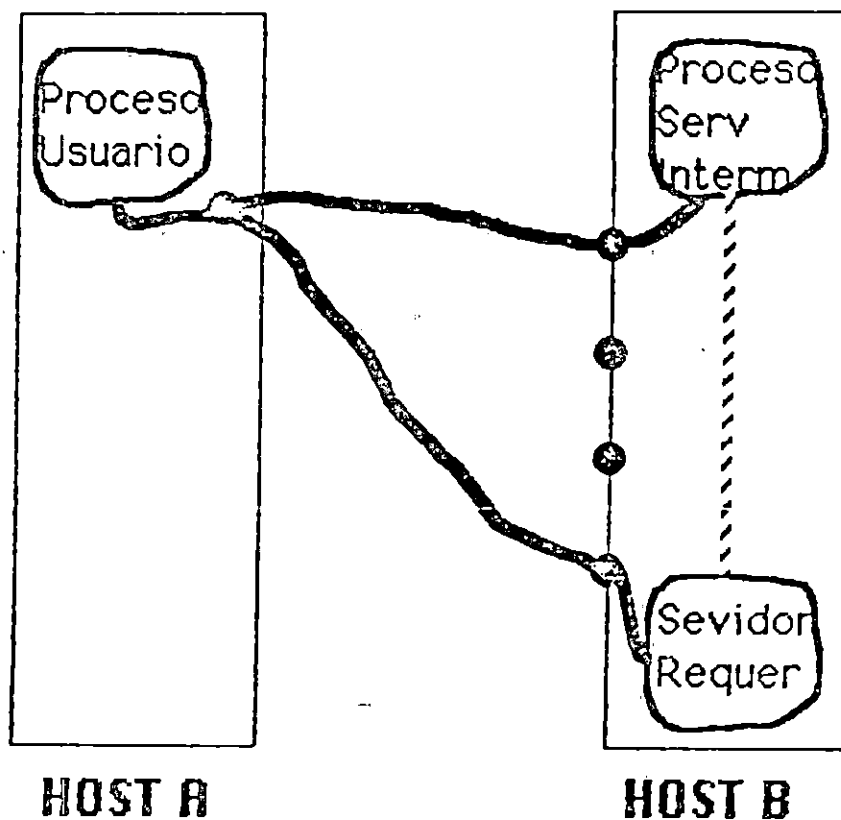
info:
begin
  for i := φ to count-1 do
    user buffer address := byte(count+i);
    byte count := byte(count+1);
    if i = φ then waveup;
  end;
end;

procedure clock:
begin
  for i := 1 to Max (can do
    with (count) do
      if timer > φ then
        begin
          timer := timer-1;
          if timer = φ then
            begin
              state := cleaning;
              to Ale + 1; φ, mφ, Clear Reg, data, φ)
            end
          end
        end
      end;
end;

```

Servidor Intermediario

- Servidor Interm oye @ conocida
- Usuario se conecta a esa @
- Serv Int crea proceso solicitado
- Le indica por qué @ oír
- Serv requerido oye por @ indicada
- Serv Interm informa a usuario @ creada
- Usuario se conecta por esa @
- Serv Interm vuelve a oír por @ conocida



Direccionamiento y conexiones

Host tiene recursos ^{adice} accequiblos

Ej. Archivos, compiladores, impresoras

C/U tiene proceso que lo administra: servidor

Cómo saber @ de transp de un servidor?

Servidor tiene nombre y @

Ej: Directorio telefónico

Una posibilidad: Well known addresses

@s se mantienen Ktes y se publican

Si host tiene miles de servicios

debemos tener miles de servidores

oyendo por miles de @s? ineficiente!

SOLUCION: SERVIDOR INTERMEDIARIO: ^{despachador}LOGGER

Arpanet

Control de flujo y buffers ^{anq. de}

- Similit y diferencias con nd enlace
- En redes con interfaz de dtg
- En redes confiables " " ^{<v}
- Asignación dinámica de buffers y no estática (seu → red)
- Separación de la asignac de buffers y acks
- Control de flujo por capacidad de la red y ^{mem.}

Similit y diferencias con enlace

- Similitudes
 - Objetivo común :
 - Evitar que host rápido ahogue a uno lento
- Diferencias
 - IMP maneja pocas líneas
 - Host maneja numerosas conexiones
 - ^{result} Impráctico usar técnicas de enlace pres
 - Se requerirían demasiados buffers ^{mem.}

~ En redes con interfaz de dtg ^(enlace)

- ^H Transm debe hacer buffering para retransm
- Receptor puede dedicar bufers a conexas ^{los} ó per
- Puede también tener ^{no} pool de bffrs para todas
 - Al llegar mensaje puede descartarse
 - Transmisor hace timeout retransmite
 - Hay peligro de abrazo mortal de reensamb

3 En redes confiables (CV)

los buffers los
hace el SMP y
no el host

Si receptor garantiza espacio p' la llegada de los pags
Transmisor no tiene que hacer buffering
Sólo conserva mensaje hasta ack de la red
Si recept hace buffering, cuál es el tamaño?

- Si poca variación en long de mensajes, podemos ver un pool.
- 1) Pool de buffers de longitud fija

• Si long de los msgs tiene alta variación

• Problemas con pool de long fija

• Si tan grandes como msg + largo posible

• Desperdicio de espacio

• Si + pequeño q' el msg + len

• Múltiples buffers para un mensaje

• Complejidad

• 2) Buffers de long variable

• Mejor utilización de memoria

• Mayor complejidad

• 3) Buffer circular grande por conexión

• Bueno para conexiones con alta carga

Entonces hacemos buffering en rcpt o trnsm

• Para tráfico interactivo (intermitente)

• En el transmisor Δ bin. buffer

• Para tráfico batch

• En el receptor phpr cle. de pag.

de un l/ho

Msg 1

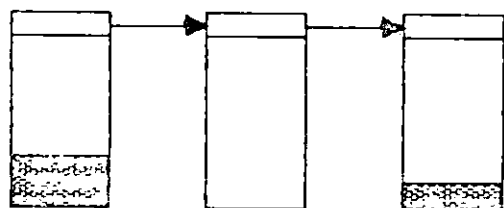
Msg 2

Msg 3

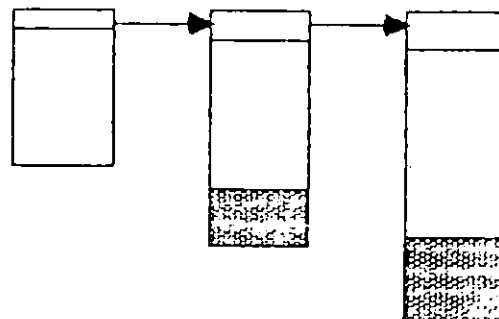
Msg 4

h/ho.

op!



1 buffer por q' mje.



4 Asignación dinámica de buffers

∞ en caso de conexión q' larga el receptor.

- Varía con apertura y cerrado de conex
- Protocolo debe permitir ^{alter} pedir buffers *100 - 1000*
- Debe permitir también ajustar asignación *2 buf*

4 SEPARACION DE LA ASIGNACION DE BUFFERS Y ACKS

A	MENSAJE	B	COMENTARIOS
→	<Pide 8 buffers>	→	A quiere 8 buffers
←	<ack=15, buf = 4>	←	B asigna sólo 4
→	<seq = 0, data = m0>	→	Le quedan 3 a A
→	<seq = 1, data = m1>	→	Le quedan 2 a A
→	<seq = 2, data = m2>	***	Pérdida A cree queda 1
←	<ack=1, buf = 3>	←	B conf 0 y 1. Acep <u>2,3 y 4</u>
→	<seq = 3, data = m3>	→	Le queda 1 buf a A
→	<seq = 4, data = m4>	→	A tiene 0. Debe parar
→	<seq = 2, data = m2>	→	A hace timeout y retransm
←	<ack=4, buf = 0>	←	Todo ack pero pare
←	<ack=4, buf = 1>	←	Puede mandar 5
←	<ack=4, buf = 2>	←	B encontró otro buffer <i>reloj A</i>
→	<seq = 5, data = m5>	→	Le queda un buffer a A
→	<seq = 6, data = m6>	→	A queda bloqueado otra vez
←	<ack=6, buf = 0>	←	Todo OK pero A bloqueado
***	<ack=6, buf = 4>	←	Posible abrazo mortal

AM → A pide exp. AdB!
6 " " msje!

→ si se pierde ⇒ A.17 B exp q' d' todas las mps es c

Don → B pide exp. A.17 B exp q' d' todas las mps es c
7 # se al. sí. A.17 B exp q' d' todas las mps es c

6 Cntrl de flujo por capac de la red

- Suposición hasta el momento
 - Limitante = capacidad de bffrs del recept
 - Memoria barata
 - Limitante no existirá más $Mem \infty$!
- Nuevo cuello de botella :
 - la Capacidad de la red $\cdot (p' \text{ primer } p' \text{ s})$
 - $X = \text{Máx \# de pkts entre IMPs/seg}$
 - $K = \text{\# de caminos disyuntos entre hosts}$
 - $\Rightarrow \text{Máximo throughput} = KX$
- Belsones propuso "sliding window" dinámico:
 - Transm ajusta tamaño de ventana dinámicam
 - Lo hace basdo en capacidad de la red
 - Capacidad de la red = $c \text{ msgs/seg}$
 - Ciclo de un msg = $r \text{ seg}$ $\rightarrow \text{transm}$
 - Tamaño de la ventana de transm = cr
 - Transmisor opera a tubo lleno
 - Cálculo de c y r por el tr:
 - $c = \text{\# de msgs ack en cierto } T/T$
 - Durante medición debe transmitir rápido;
 - $r = \text{media mantenida permanentemente}$

Σ: debe tener en ctn. el fin.

promedio de los n
paquetes.

Multiplexación

- Hacia Arriba
- Hacia Abajo

Hacia Arriba

- Varias conex de transp / CV
- ^{pres-}Ahorra costos ^{de asbleci/ de conexi- al CV}
- Util en tráfico interactivo

se ven
todas en
un = CV!
solo ✓.

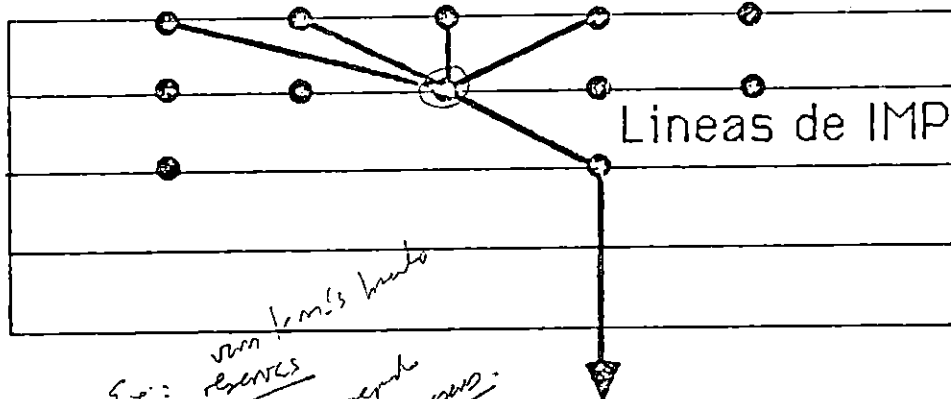
Nivel

4

3

2

1

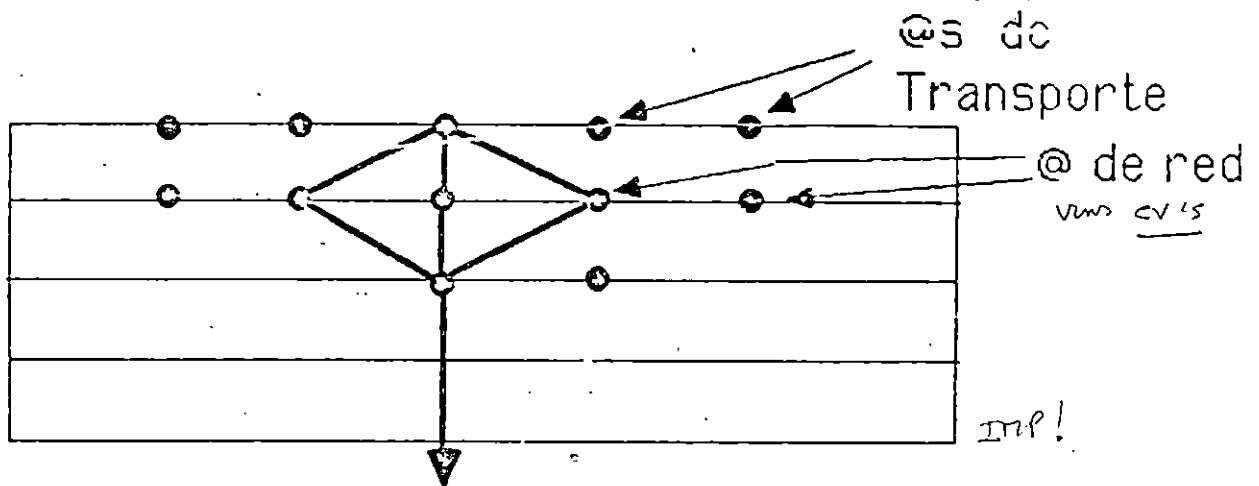


Ej: reservas
1 solo CV porque
todos los mens.

Nivel

Hacia Abajo

- Varios CV/conex de Transp
- Tráfico batch de gran veloc
- Línea host-IMP debe tener AB grande
- Util en líneas de satélite de vel.
- Ejemplo



Ejemplo

X.25:

Paquetes de 128 bytes *may 4K*

Número de secuencia de 3 bits *→ vent. de br. de 7*

Número máx de pckts en *no. J = 700* 540 msg = 7

70/5 ≈ Aprox 14 pckts/seg

$14 \text{ pck/sg} * 128 \text{ bytes/pck} * 8 \text{ bits/byte}$

V aprovechada ≈ 14 kbps

V típica en línea satelital ≈ 50Mbps

GRAN DESPERDICIO DE AB

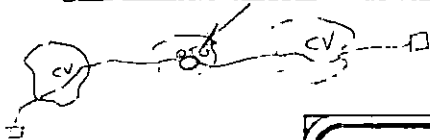
7 P / 540 msg

Sincroniz por pckts demorados

cap de elegir la pckts q' estn en la red

- Crítico en datagramas por looping
- CV deben manejarlo por interconex de redes
- Origen del problema
- Consecuencia
- Soluciones

no hay una vía para la pckts para que en la red → demora



Origen del problema

(del looping)

Traen control de flujo en la red

- Mal enrutamiento
- Congestión

Consecuencia

- #s de secuencia traslapados
- Crítico al reabrir conexiones
- Con mismo # de conexión
- Con misma combinación de @s
- Confusión en el receptor
- Cuál es original y cuál duplicado?

2 pckts con el = # seq.
en la = conexión
la red puede
enviarlos
rechazarlos
y enviarlos
otra vez

si se envía
se envía
5 7 0 1 2 3 4
y se envía
con el
el cuerpo →

duplicado

SOLUCIONES

Cach: *Asla. ^{redes} Schneider*

(1) No reusar @s

- Anteponerles la hora
- ^{hacer al mltiplo del} Logger $\rightarrow$ imposible
- Variantes ^{o posibilidades destino}
 - Reusar @ ^{frente.} fuente pero no destino
 - Prohibir reuso de combinac de @s

(2) Hacer #seq tan grande que... ^{nunca se recicle}

- ^{# sec de tr} 48 bits a 1000pck/seg recicla en 8000 años ^{ya se traspasa}

Problema común:

- Hosts deben mantener historia ^{de las anexas g' ya no se}
- Si host se cae ^{mem}
 - En qué número de seq iba?
 - Qué direcciones ya se usaron?

en (1) serie

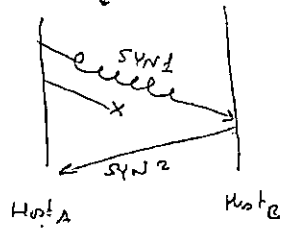
3) Limitar vida de pckts

- (1) Contador de saltos *p' q' no se queda dentro de valores por siempre en la pag.*
- (2) Tiempo: Relojes de IMPs sincronizados
 - Externamente con estación de radio
- Debe limitar también vida de acks
 - $T = \text{Múltiplo de } t_{\text{máx}} \text{ de vida de pckts}$
- Si usamos k bits para $\#seq$ 2^k
 - Host no debe emitir + de $2^{*} * k$ pckts en $T \rightarrow$
- Reusemos $@s$ o $\#conex$ sólo después de T *si $\pm$, problema todos duplicados*
 - Problemas
 - Demoras en establecimiento de conex
 - Mantenim de inform de conex muertas
- Variante
 - No inicializar $\#seq$ en 0

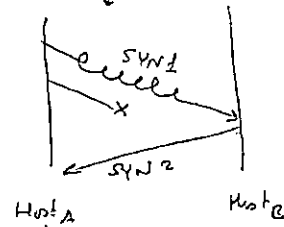
- Host recordará $\#seq$ durante T
- Partirá del siguiente
- No demora apertura de conex
- Requiere recordar inform de conex cerradas
- Alternativa
 - Incluir $\#$ de encarnación *otra g*
 - Se pierde en caída de host
- Problema común
 - Host debe guardar info de conex $\times$ cerradas
 - Si host cae *(según de la ISO)*
- Hay que esperar T al resucitar $\rightarrow$ *sin p' en la vida*
- T es difícil de estimar *com. a la red*
- Peor en interconexión de redes

*T = ...
se ...
y ...
después ...
no ...
se ...
por ...*

(4) Tomlinson (1975)

- Equipar hosts con relojes
 - Contador binario
 - No sincronizados
 - $\#bits(reloj) \geq \#bits(\#seq)$
 - Funcionan con host caído
 - Objetivo: Evitar 2 pckts con $\#seq =$ en la red
 - $\#seq$ inicial = kbits bajos del reloj ($\neq 0$!)
 - Cómo sincronizar establecim de conex
 - Handshake de 3 vias
 - Normal
 - SYN1 duplicado
 - SYN1 y datos duplicados
- 

 $Host A$ $Host B$
- $SYN1 = \#seq$, plus datos
 $SYN2 = ACK$ de $SYN1$.
 (plus orden)



SYN 1 = # dec., plus dest.
SYN 7 = ACK de SYN 0.

frag' puden)

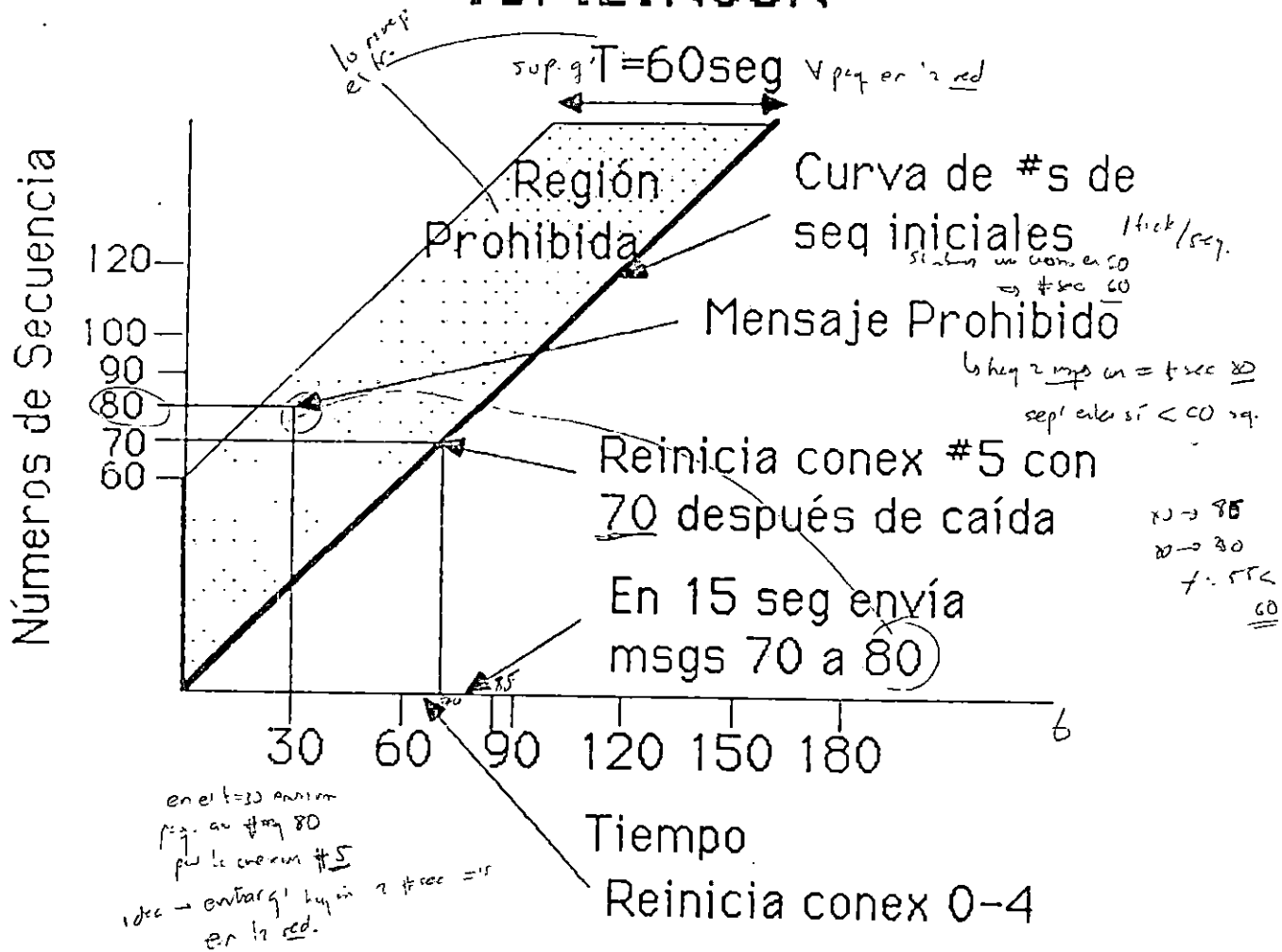
slr:

P⁴⁹ 321.
 H. S. de
 3 548

It is correct on 104 signed copy.

haber deploras.

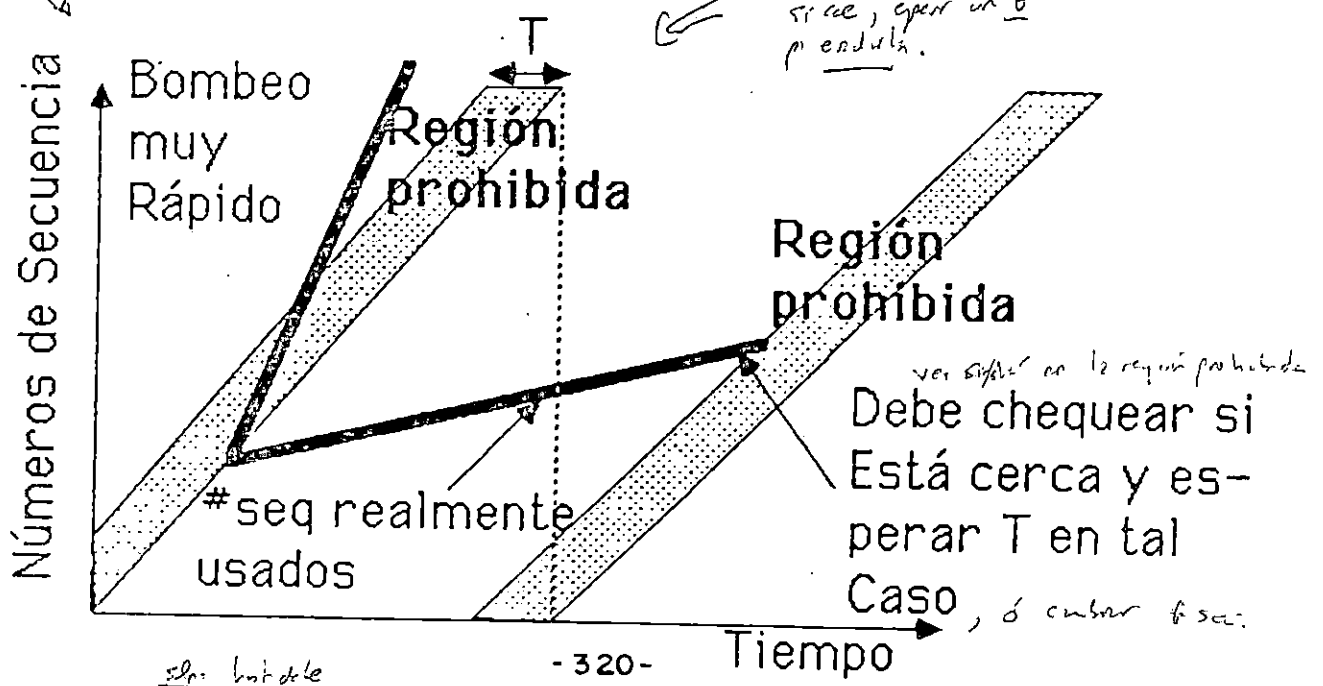
TOKLINSON



Frecuencia del Reloj = 1 tick/segundo

No se puede entrar + reg. de CI - dejarlo?

Concl:
tout est en fait le même,
p ne var en situ,
si ce, pour un 6
pendula.



V.S. de 2 vcs:

Normal

(bien hecho!)

A	PAQUETE	B	COMENTARIO
—>	<syn1,seq=x>	—>	A quiere empezar
<—	<syn2,seq=y,ack=x>	<—	B acepta
—>	<data,seq=x,ack=y>	—>	Empieza transm

SYN1 duplicado

A	PAQUETE	B	COMENTARIO
.....	<syn1,seq=x>	—>	syn1 <u>demorado</u>
<—	<syn2,seq=y,ack=x>	<—	B acepta
—>	<reject,ack=y>	—>	A no ha solicit nada

Se queda dicho well y se re envia

A no ha solicitado un duplicado de un paquete mas.

SYN1 y datos duplicados

A	PAQUETE	B	COMENTARIO
.....	<syn1,seq=x>	—>	syn1 dup demorado
<—	<syn2,seq= <u>y</u> ,ack=x>	<—	B acepta
.....	<data,seq=x,ack= <u>z</u> >	—>	Datos dup? Z≠Y
—>	<reject,ack=y>	—>	A no ha solicit nada

propone

confirma

"repro"

Recuperación

- Transmisión
- Caídas de la red
- Caídas del Host

Reset → se envía un flag
 Result → se envían datos
 NBT de los paquetes de
 estos!

Transmisión

- Mens de tranp puede incluir CRC Ej. cplz loc mz 2
- Reduce errores no detectados por red
- Se protege de bugs de la red
- Detecta errores de reensamblaje

Redo
 150
 A → b222
 B → x25
 C → red
 en abst de DTG.
 150. 5 paquetes
 de byte

Caídas de la red

- Reset, Restart
- Posible con buffering en transmisor
- Host A pregunta estado a B
- Ejemplo:
 - Tengo mensajes 2, 3, 4, 5 sin ack
 - Ha recibido alguno de ellos?

error de reensamblaje
 no envían los paquetes
 CRC → detecta esto.
 porq' la rec. es g.
 mayor la byte es #.

Caídas del Host

- Más complicadas e imposibles de resolver a nivel de Tpte.
- Ejemplo de por qué en G...
 - Transmisión de archivo
 - Stop-and-wait 1 solo msg. en tr.
 - En la mitad se cae receptor
 - Transmisor 2 estados posibles
 - S0: No hay mens pendiente
 - S1: Hay mens pendiente
 - Hilando fino hay un problema
 - Secuencias posibles
 - Pérdida de mensaje
 - Envío ACK
 - Caída
 - Grabación en disco
 - Mensaje duplicado
 - Grabo
 - Caída
 - Envío ack
- No puede ser transprte a capas super

cbcd. → no hay fin de
recup. de env.
problema a nivel de

soln: recup.
a nivel de sesión
1571 Check Point

Debilidad de OSI

Interconexión de redes

problema de red vs no de red IP.

- Generalidades
- Puertas
- X.75 vs. Datagramas
- Fragmentación de Paquetes

Nivel de red en 3

-----> subred de empresa / -----> copia en red no así y volviendo OSI
-----> red global
-----> subred IP (mucha el protocolo interred).

X.75 -> copia la tecnología de X.75 a inter-red.

- CIP -> protocolo a nivel de red de Apnet

Hosts con DTG's.
TCP -> protocolo a nivel de inter-red

Generalidades

- Se ^{debe} suponer ^{modelo} de red en DTG p' inter-red
- Aun si dos redes extremas con CV
- Red intermedia puede usar DTG
- No ^{se} debe afectar tráfico interno

c's:

- Unica
noção p/ $\rightarrow$ centrar
nos dir. $\rightarrow$ le. cogn. de
Jung.

$\begin{array}{ccc} 3 & 3 & 7 \\ \downarrow & \downarrow & \downarrow \\ \text{pers} & \text{ind. c.} & \rightarrow \neq \text{del } \underline{\text{abonado}} \end{array}$

com cédula de
B/2. se pode
votar em Chico

dir. plm → ~~reducer~~ poate este unghiul de $\frac{\pi}{2}$ de la 2. prin $\frac{\pi}{2}$.
→ se reg. de control la usq. de 15.

→ código kl. → es permanente

g' parler usages des l's de
mon indep x païs.

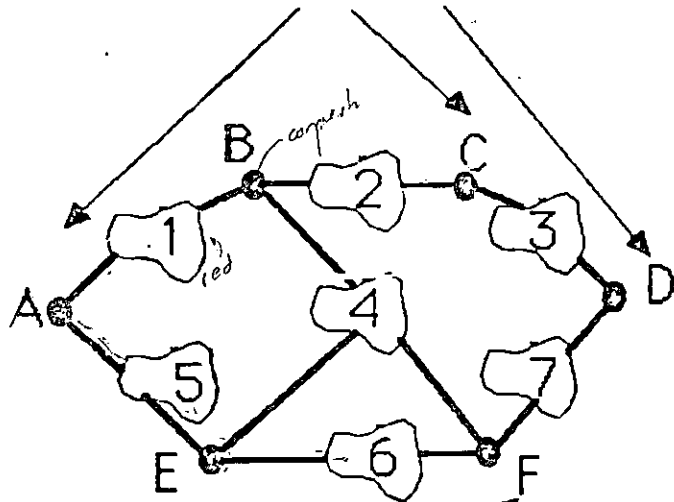
En $\mathbb{R}$, + de 7, la φ'
surde es φ' long $\varphi' \times$ de facto
(automatique)

Compuertas

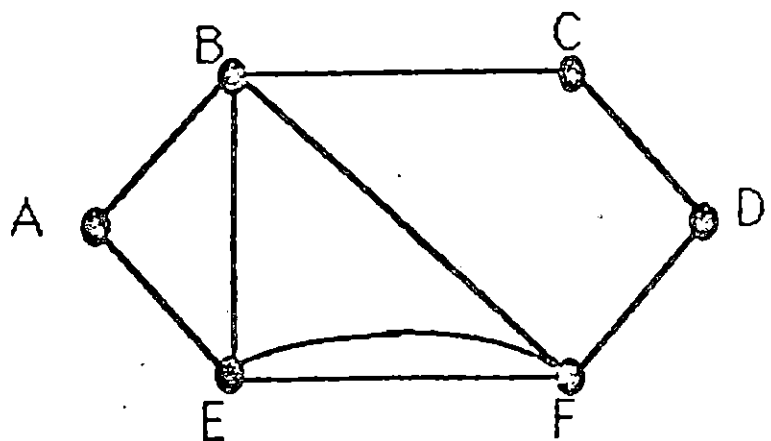
- Convierte pckts de un protoc a otro
- Puede ser Bilateral o Multilateral
- Se forman superredes
 - Nodos = compuertas (strp)
 - Arcos = redes (cables)
- E y F tienen arcos paralelos
- Tienen camino redundante
- Redes de Interconexión

bridge → compuerta de N. de enlace.
 repetidor en ethernet → comp. de N. físico.

Bilaterales



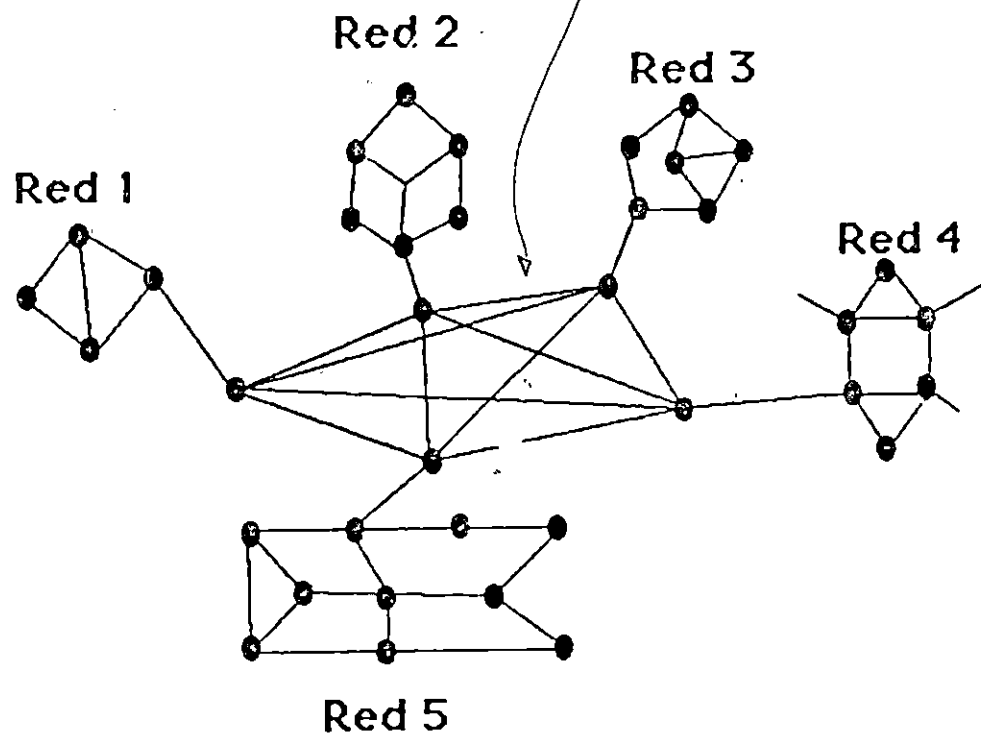
Trilaterales



si N grado
 de conectividad de la
 red → $N!$ caminos.

Redes de Interconexión

- Red nueva para tráfico inter-red
- Gateways son IMPs de red nueva
- Ejemplo: Euronet
- Implicaciones políticas
 - Carriers pierden control → no es rentable (H) - pierden



Programación

(ctrl. de mas q. tos.)

- Lo más simple
- Alternat: Protoc. interred común

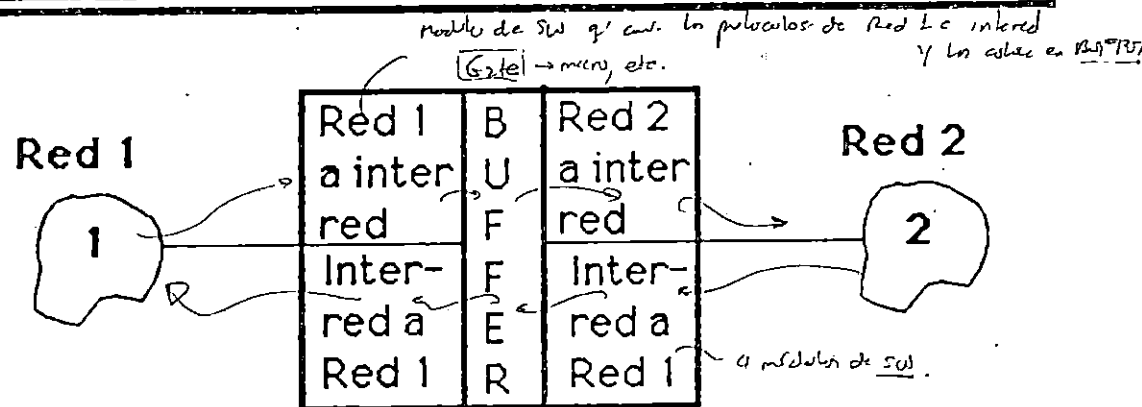
Lo más simple

- Conversión formato de una red a otra
- Compuerta bilateral: $2 \rightarrow 2'$ 2 conversiones
- Compuerta trilateral: $3 \rightarrow 3'$ 6 conversiones
- Si ^{el} ~~alta~~ conectividad en superred _{es 21(2)}
- Software inmenso

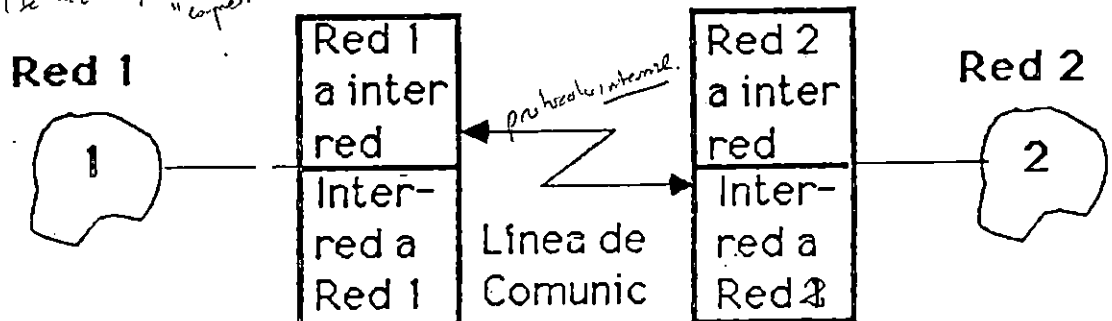
Alternat: Protoc interred común

Idea →

- Nadie modifica software interno
- Cuatro procesos en compuerta
- Para n redes ^{n^2} ~~$n(n-1)$~~
 - Requiere $2n$ rutinas de conversión
 - Antes requería $n(n-1)$
 - En ejemplo: 14 y no ~~44~~ rutinas *exped a local*
- ^{Problema} ¿Quién administra compuerta?
 - Claridad de responsabilidades
 - Half-Gateway
 - Se puede poner en IMPs o Hosts



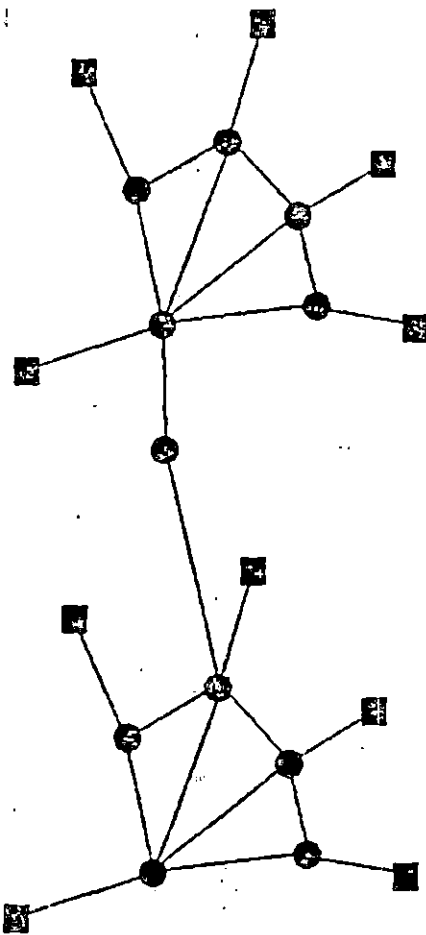
sin el problema puerto.
Lo q' se hace hacer es hacer "1/2"
"compuerta"



Nivel de Interconexión

Donde se colocan los gateways? en IMP o en Hosts.

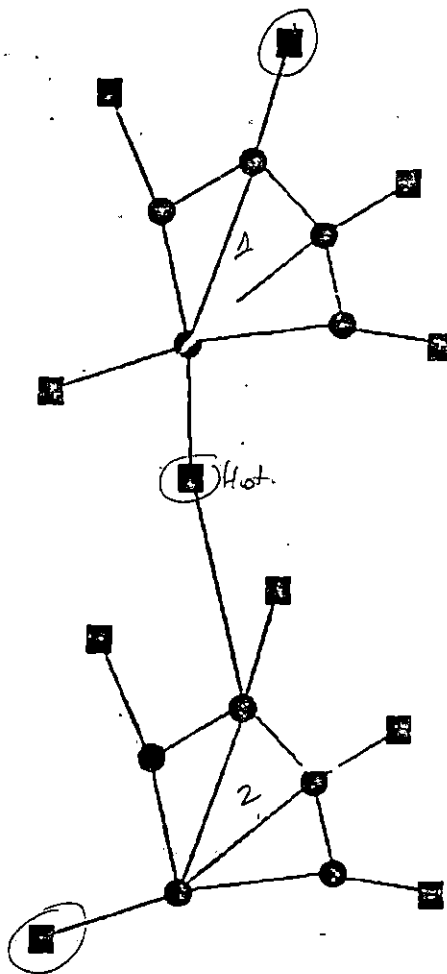
- Gateways en IMPs o en host?
- En IMPs
- En hosts



(1) COMPUERTA EN IMP

- Para que en RLIS más no en RED.
Ej: bridge de T-Bing.

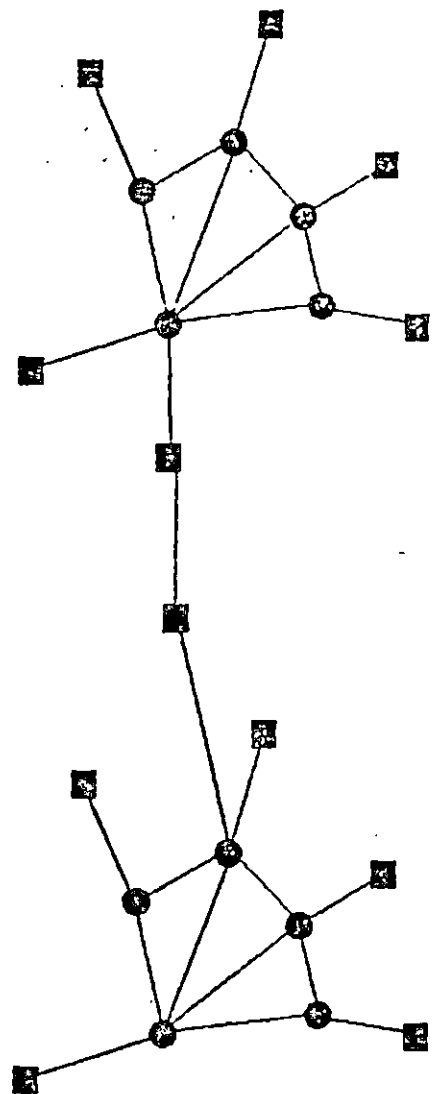
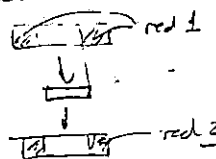
No debe interferir el tráfico interno
por routers en tráfico interred



(2) COMPUERTA EN HOST

(+ simple).
protocolo de los 2 redes

-330-



(3) MEDIAS COMPUERTAS EN HOSTS

En IMPs

- Si protocolos de subred =s
 - Problema de @miento
 - Para enviar pckt en red 1 a IMP de red 2
 - Si @s =s en las 2 redes
 - Bit indicandor de red
 - Si hay muchas redes se necesita header...
 - Sobrecarga a tráfico interno
- Si protocolos de subred ≠s
 - Una red puede no controlar dupl
 - Protocolo end-to-end?
- Conclusión: es practicam imposible

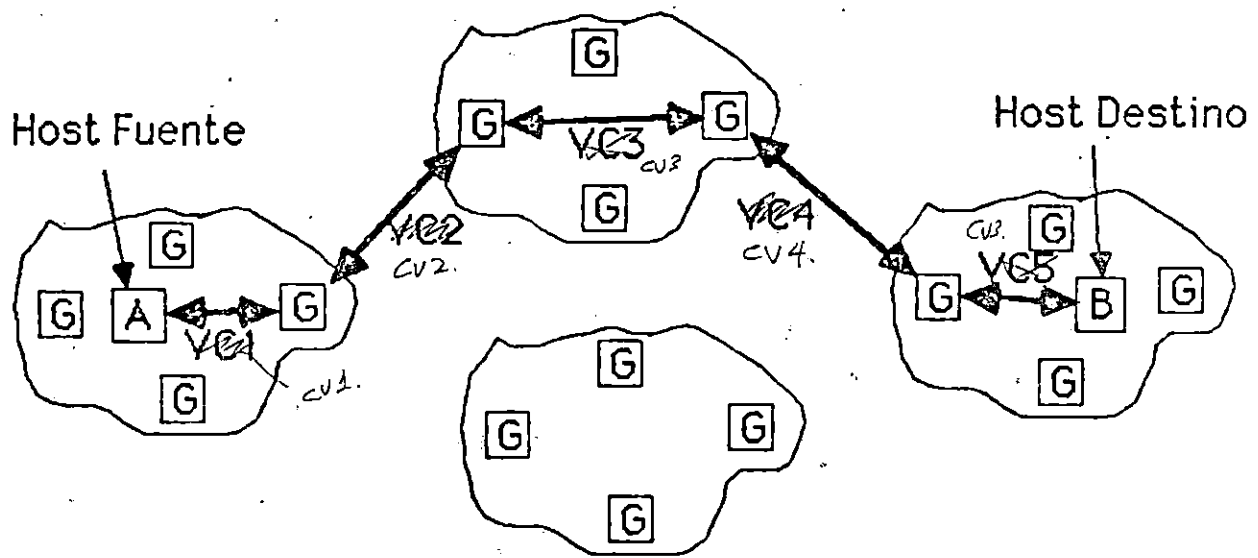
En hosts

- Mucho más fácil
- Se porta para cada red como host normal
- No hay molestias a protocolo interno
- Para mandar mensaje interred
 - Transp normal host fuente y comprta
 - Comprta transf protocolo (headers)
 - Transp normal comprta y host destino

X.75 .vs. Datagramas

- CCITT sacó X.75 basado en CV
- ARPANET y XEROX basado en DTG
- Enrutamiento

repetir x25
en la comparativa

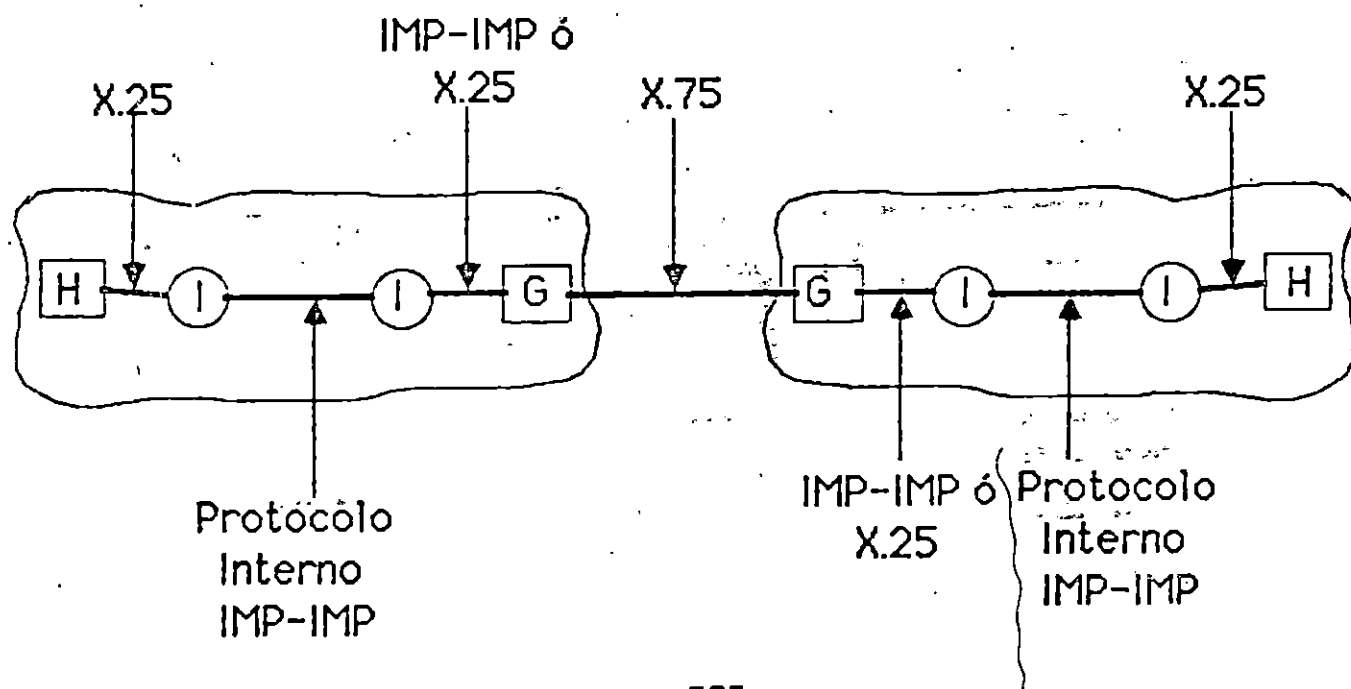


≠ x25 y x75 : En el pg. de Call Report 2 en campo UTILITIES
el red debe guardar su dir x121 en ese campo.

get 2
x.21
4 dígitos
3 per y 1 red
p' cubrir y p' seguridad.

CCITT sacó X.75 basado en CV

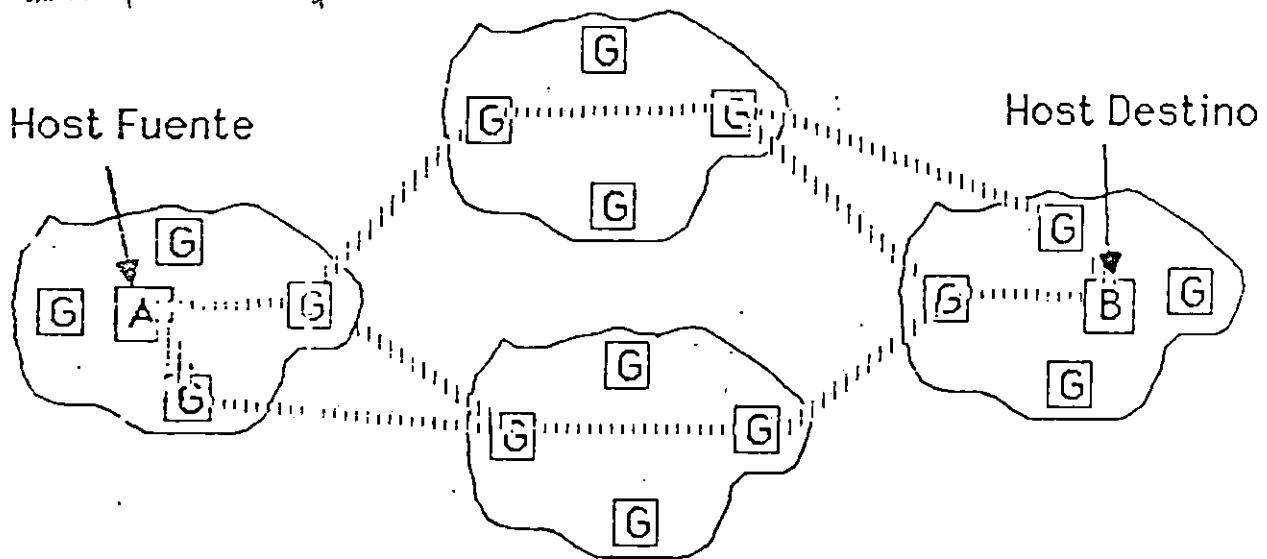
- Transporte hace CALL REQUEST
- Lo pasa al nivel de red del host
- Host o IMP escogen media comp^{ueh} de esa red
- Media comp busca siguiente media comp, etc
- Media comp de red destino localiza host dest
- CV funciona a nivel de compuerta no de IMP
- Entre una compuerta y otra puede haber DTG
- En fig VC1, 3 y 5 pueden ser internam DTG
- X.75 Prácticamente = a X.25
 - CALL REQUEST adiciona campo de Utilities
 - Empieza con byte diciendo su longitud
 - Exige que c/red grabe sus 4 dígitos X.121
 - Contenido del resto del campo no definido



ARPANET y XEROX basado en DTG

- Pckts no deben viajar por misma secuenc de comp
- Headers y trailers cambian
- Ver figura

usar DTG por la red de ~~gates~~



Enrutamiento en redes inter-red

+ c.

- Dificultad de conocer topologías remotas
- (1) Enrutamiento jerárquico?
 - Complicado
- (2) Host fuente escoge primera compuerta
 - Compuertas manejan el resto (organizan el tráfico por la red)
- (3) Enrutamiento fuentes (+ usado en redes inter-red)
 - En tráfico interno no tiene sentido
 - En tráfico internacional es atractivo
 - Razones políticas

Por seguridad
se prefieren redes
más aisladas
(como por Canadá
y por Cuba)

- Toda red tiene un $\text{MaxPckt}^{\text{fin max de}}$ por:
- Ejemplos :
- Si red intermedia tiene Maxpckt pequeño?

- H_{w} $\frac{1}{2}$ de $\frac{1}{2}$ ~~H_w~~: Slot de TDM
- S. O.: Todos los Bffrs de 512 bytes $\rightarrow$ pag's de 512.
- Protocolos: # de bits en campo de long. long. —
- Estándares
- Reducción de carga en retransmisiones_p
- Evitar que pkt ocupe canal mucho t _(tiempo interch.)

- HDLC: Infinito, en teoría, pero *no se usó mucho*
- X.25: 8192 bits hasta 1980
- *(4K)* 32768 bits en 1984
- ARPANET: 1008 bits
- ALOHANET: 640 bits

=> 2 pag's = medium

Si red interm tiene ^{pkts} ~~pk~~t pequeño? →

- Evitarlo enrutando por otras redes
- Qué pasa si el problema es en red destino?
- Fragmentar en compuertas

^{es red} ⇒ Fragmentar en compuertas

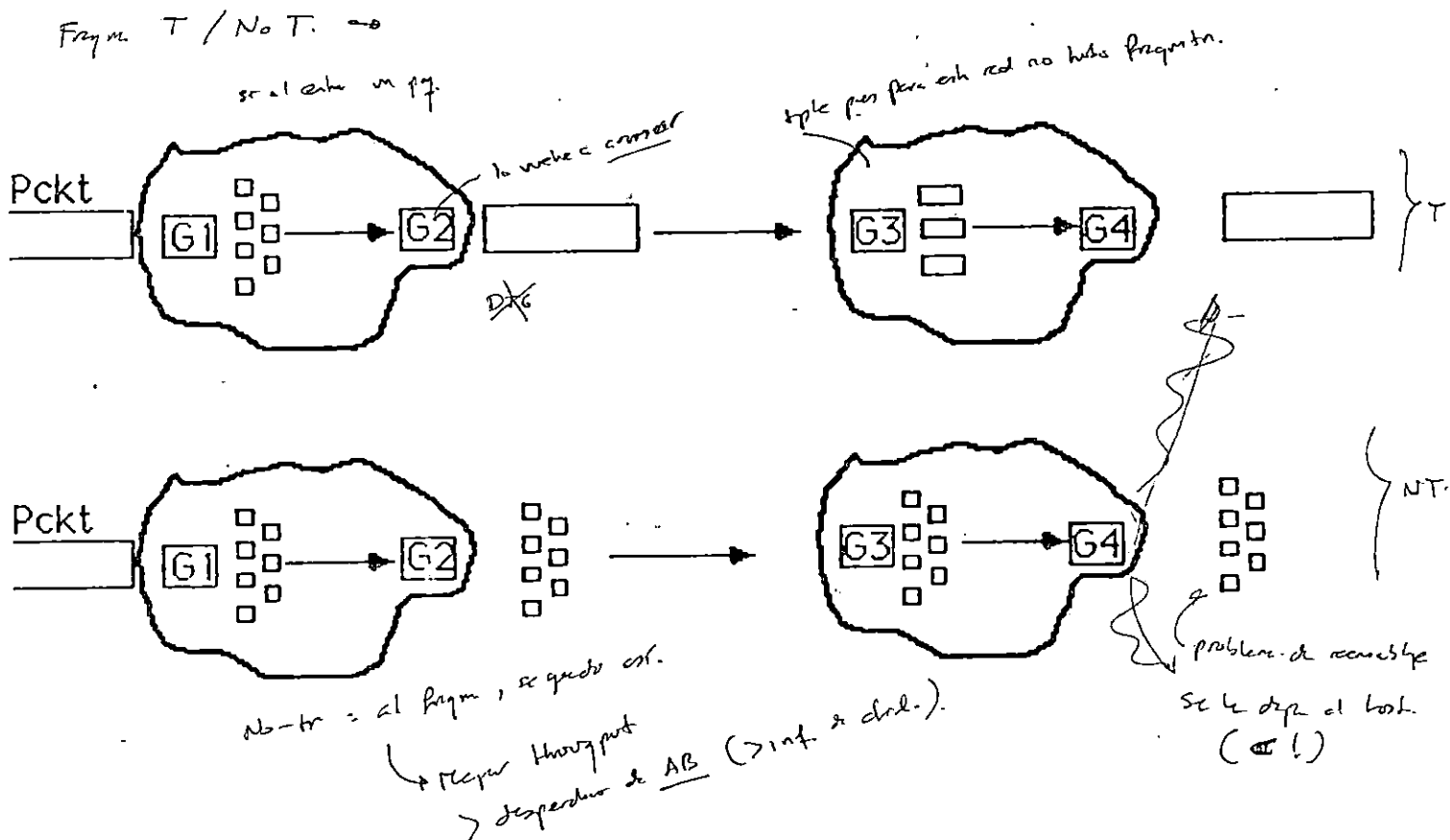
- ^{fragmentar ⇒}
- Problemas de reensamblaje: Ej. ARPANET
 - Dos estrategias
 - Fragmentación transparente
 - Fragmentación no transparente
 - Propuesta de Shoch (1979)

Fragmentación transparente

- Dentro de una red
 - Compuerta fuente fragmenta
 - Enruta todo a misma compuerta destino
 - Compuerta destino reensambla
- Redes siguientes no se enteran
- Problemas
 - Compuerta destino debe saber # de fragm
 - Se debe incluir contador o "end of pkt" bit
 - Todos deben salir por la misma compuerta
 - Se pierde performance
 - Varias fragmentaciones y reensamblajes

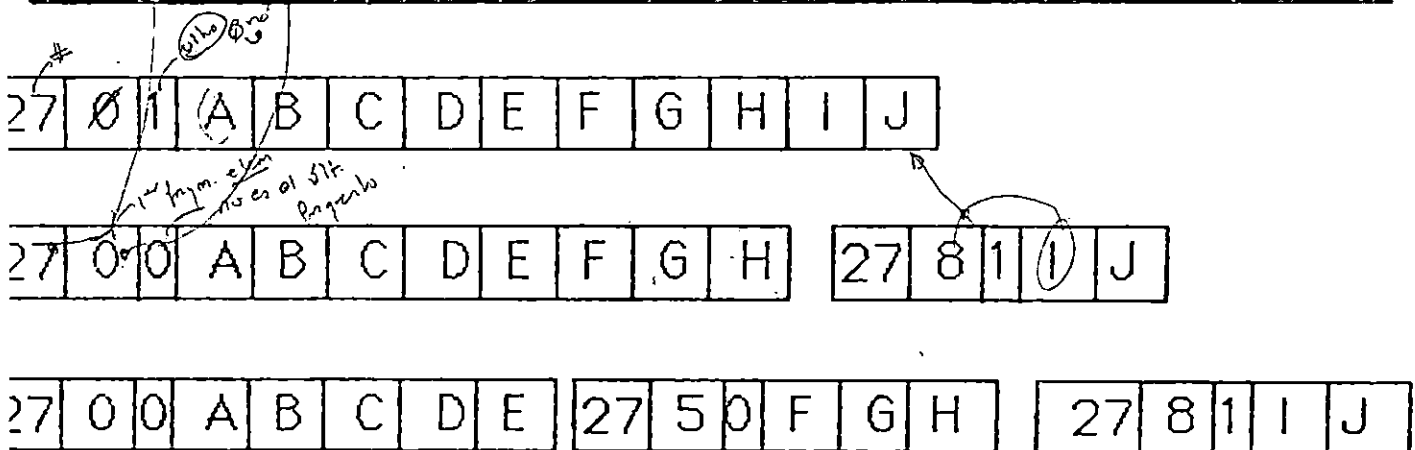
Fragmentación no transparente

- Problemas:
 - Los hosts deben reensamblar
 - Más información de control
- Ventajas
 - Múltiples compuertas. No útil en CV



Propuesta de Shoch (1979)

- Bit en pckts
- Dice si destino puede reensamblar
 - Sino, compuerta debe buscar otra que...
- Fragmentos deben numerarse
 - Una opción es usando un árbol
 - 0, 0.0, 0.1, 0.2, etc. $0 \leq \begin{matrix} 0.0 \\ 0.1 \\ 0.2 \end{matrix} < \dots$
 - Si fragmento 0.0 debe fragmentarse
 - 0.0.0, 0.0.1, 0.0.2, etc *problem: si pasa por una red de DTG y nos falta un paquete*
 - Si red intermedia pierde pckts?
 - Retransmisión puede tomar otra ruta *retr. paquete*
 - Fragmentación puede ser mayor
 - Reensamblaje puede ser equivocado
 - Otra opción mejor (Cerf y Kahn, 1974)
 - Definir fragmento elemental
 - Pasa por cualquier red
 - Header ~~interned~~ de c/fgmt debe tener 2 #s
 - Número original del pckt
 - # del 1er frgmt elemental contenido
 - Bit indicando ^{si es el} último fragmento
 - Fgmt elemental puede llevarse a bit o byte.
 - Algunos usan # absoluto de toda la transm



Nivel de Sesión

- Toma Servicio de transporte
- Adiciona servicios orientados a usuario

Adiciona serv orient a usuario

- Identificación de usuarios
- Asignar privilegios
- Liga usuarios a procesos
- Maneja alternación en diálogo interactivo
- Sincronización, checkpoints y recuperación
- Hace fallas de conexiones de transporte
 - transparentes a niveles superiores

NCP

- Network Control Protocol
- Supone circuitos virtuales
- Servicio confiable de la red
- Maneja conexiones simplex
- Conecta puertos
- Establecimiento de conexiones
- Conexión recibe # (link) de 8 bits
- Mensajes de datos tienen header de 40 bits
- Conexiones se cierran cruzando CLS
- Control de flujo
- INR e INS sirven como interrupción
- ECO y ERP sirve para chequear otros hosts
- ERR sirve para reportar errores detectados
- RST (ReSTart) y RRP(Restart RePly)

bits
0 8 16 24 32 40 48 56 64 72 80

INR	link
-----	------

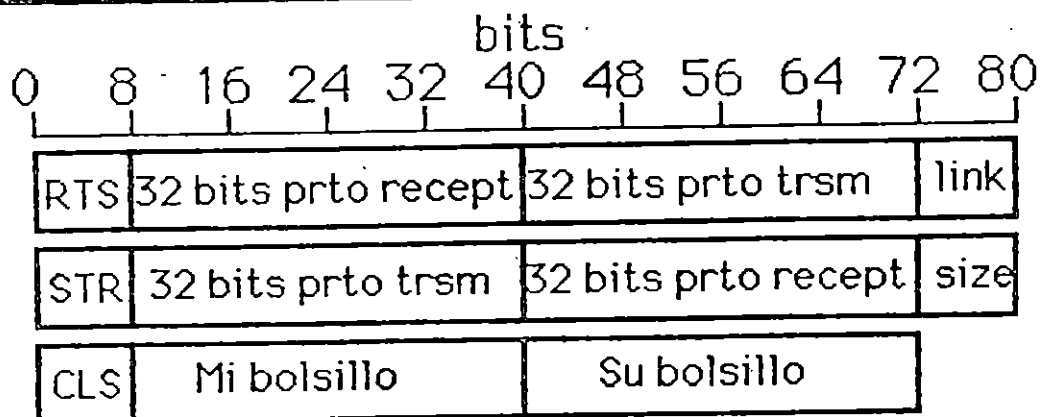
INS	link
-----	------

Conecta puertos

- Identificados por 32 bits
- Pares para recibir
- Impares para enviar
- Gender = bit menos significativo

Establecimiento de conexiones

- Dos mensajes
 - RTS: Receiver to Sender
 - STR: Sender to Receiver
- LISTEN
 - Oye pasivamente
 - Especifica puerto local
 - Opcionalmente puerto remoto
- INIT
 - Trata de establecer conexión
 - Dependiendo de genders
 - Se sabe sentido del tráfico
 - Envía STR ó RTS
 - Al llegar:
 - Si alguien estaba oyendo
 - Envía RTS o STR en respuesta
 - Sino
 - póngalo en espera o niegue CLS (close)



Conex recibe # (link) de 8 bits

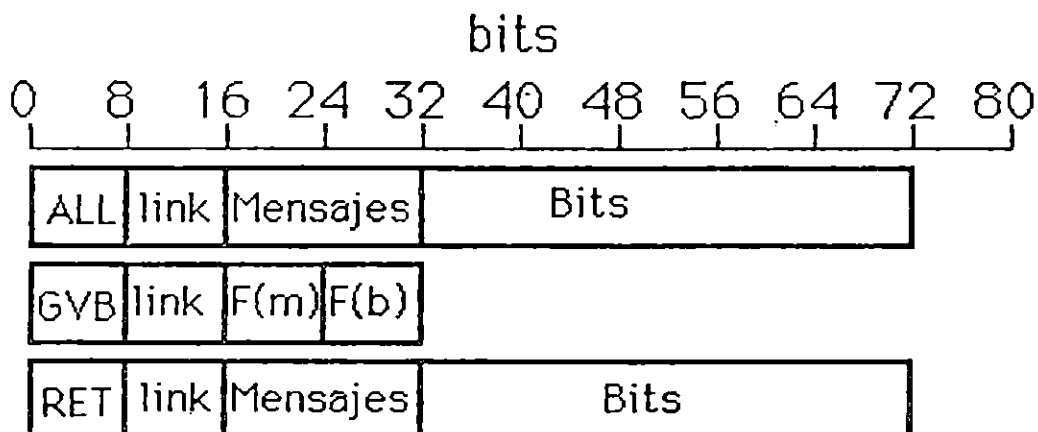
- Link se pasa como parámetro a nivel 3
- Nivel 3 lo envía como # de CV en header

Header de 40 bts en Msgs de dat

- 4 campos
 - 2 en cero
 - Número de bits por byte
 - Número de bytes en el mensaje

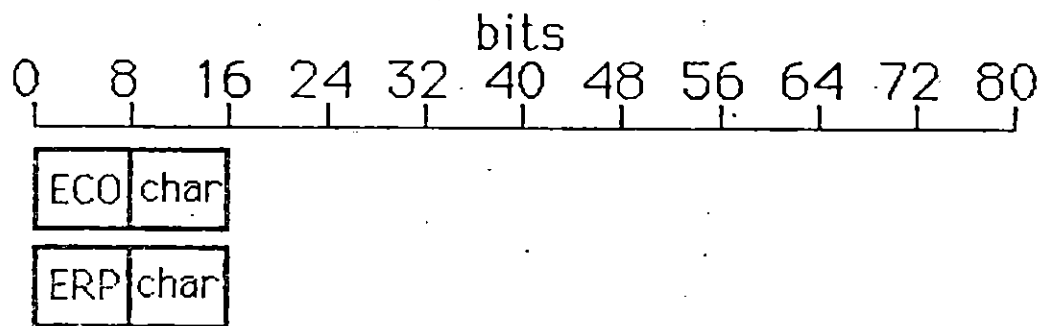
Control de flujo

- Usa tres mensajes de control
 - ALL (Allocate)
 - GVB (Give Back)
 - RET (Return)
- Al abrir conex host no puede enviar
- Debe esperar por un ALL
- ALL se envía en link 0
 - No pertenece a ninguna conex particular
 - Control de flujo a nivel de host
 - ALL especifica 2 límites
 - Número de mensajes
 - Número de bits
- GVB quita asignaciones previas
 - Rara vez se usa
 - Algunos hosts no lo tienen implementado
- Transmisor debe confirmar con RET



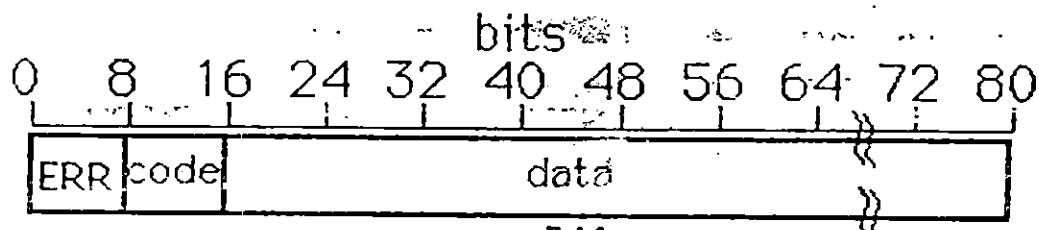
ECO y ERP chequean otros hosts

- Se da caracter de eco
- Sirve también para medidas de t



ERR reporta errores detectados

- En código se dice el tipo de rror
- En data se dan detalles
- Ejemplos
 - Msg de cntrl no definido en NCP
 - Msg de cntrl con pocos parámetros
 - Msg de cntrl con parámetros inválidos
 - Puerto o link inválidos
 - Header de transporte inválido



RST (ReSTart) y RRP(Restart RePly)

- Similar a X.25
- Purga todas las conex del host
- Se dan cuando un host se cae

bits
0 8 16 24 32 40 48 56 64 72 80

RST

RRP

TCP

- Necesario para interconexión de redes
- Redes con datagramas no son confiables
- Tiene asociado nivel 3 distinto
- Puede coexistir con NCP
- Acepta "letters" de los procesos de usuario
- Letters arbitrariamente largos
- Los fracciona en segmentos de 65K
- Envía cada fragmento como un datagrama
- Debe cuidar pérdidas, duplicados y desorden
- Cada byte tiene #seq propio
- #seq de 32 bits por duplicados viejos
- Usa handshake de 3 vías al establecer conex
- Header

Header

- Mínimo 40 Bytes
- TCP
- DATAGRAMA
- Por qué tan grande?

TCP

- Puertos destino y fuente
 - Identifican procesos en comunicación
- Longitud del header
 - # de palabras de 32 bits en header de TCP
- Flags
 - URG
 - 1 si apuntador urgente en uso
 - Apuntador urgente:
 - Desplaz a desde #seq donde viene inf urg
 - Usado a cambio de msgs de interrupc
- SYN y ACK
 - Establecim de conex de 3 vías
- EOM
 - Letters se dividen en DTG
 - DTG se dividen en fragmentos
 - Si está en 1

TCP (cont'd)

- Letter fué dividida en DTG
- Este es un DTG intermedio
- FIN
 - Cierra conexiones
 - Después de cerrar, receptor sigue...
- RST
 - Reset a conexiones con problemas
 - Confusión por SYNs demorados
 - Caídas de hosts
- Ventana
 - Control de flujo
 - Ventana de tamaño variable
 - Permiso para ventana bytes después de ack
- Opciones
 - Miscelanea P. Ej. tamaño de buffers

DATAGRAMA

- Versión distinta del nivel de red
- Tipo de servicio
 - Combinaciones de confiabilidad y V
 - Voz digital
 - Necesita velocidad de entrega
 - No corrección de errores
 - Transferencia de archivos.
 - Lo contrario
- Longitud total
 - Incluye headers y datos
 - Máximo 65k
- Número de datagrama
 - Si dtg fué fraccionado por red intermedia
 - C/fragmento debe pertenecer a un dtg
- DF (Don't Fragment)
 - Host destinatario no puede reensamblar

DATAGRAMA (cont'd)

- MF (More Fragments)
 - Si está en 1
 - Dtg fué fragmentado
 - No es el último fragmento
 - No confundir con EOM
- Desplazamiento del fragmento
 - Situación del fragmento dentro del DTG
 - Multiplo de 8 bytes
 - Fragmento elemental = 8 bytes
- Tiempo de vida
 - En segundos
 - Máximo 255 seg
- Protocolo
 - NCP ó TCP
 - Así sabe dóde buscar @ de transporte
- Checksum del header del dtg

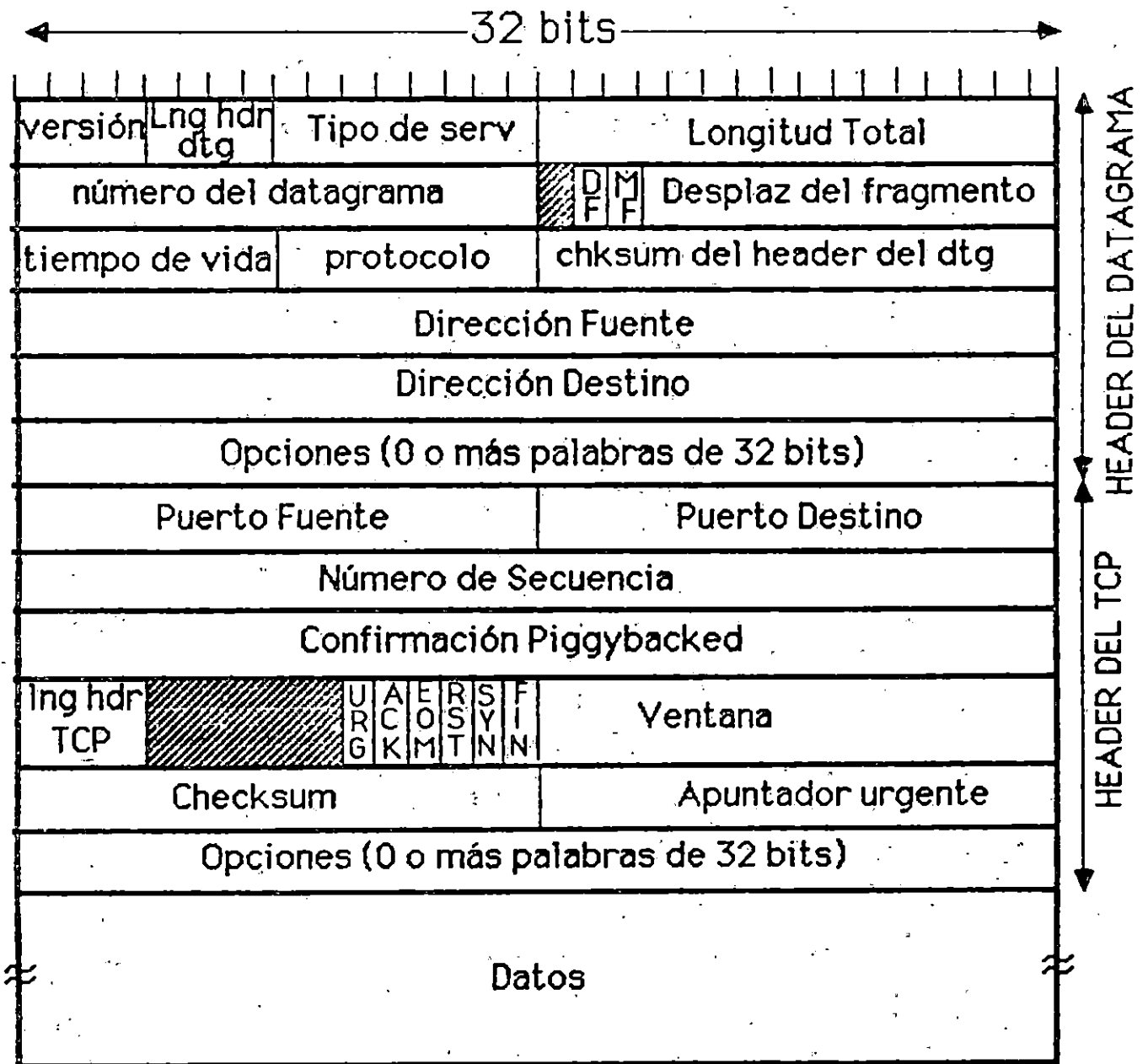
DATAGRAMA (cont'd)

- Util por cambios en header al fragmentar
- **@s fuente y destino**
 - Número de red (8 bits)
 - Número de Host (24 bits)
- **Opciones**
 - Ejemplos
 - Seguridad
 - Enrutamiento fuente
 - Manejo de errores
 - Estampillas de tiempo
 - Otros
 - Flexibilidad para futuras versiones
 - Experimentación
 - Campos raramente usados
 - Evita disponer bits en header
 -

Por qué tan grande?

- Investigadores tienen visión distinta a CCITT
- Investigadores
 - Muchos comput pequeños en red local
 - Tráfico local >> remoto
 - Redes locales tienen mucho AB
 - No vale la pena optimizar header
 - No vale la pena sacrificar flexibilidad.
- CCITT
 - Más preocupada por redes públicas
 - Típico: termin accediendo comp grandes
 - Transferencia de grandes archivos
 - Parecida a redes telefónicas (AB escaso)
 - Viabilidad económica muy importante
 - Interfaz con aplicaciones existentes
 - Todo lleva a CV y headers cortos

HEADER DE TCP



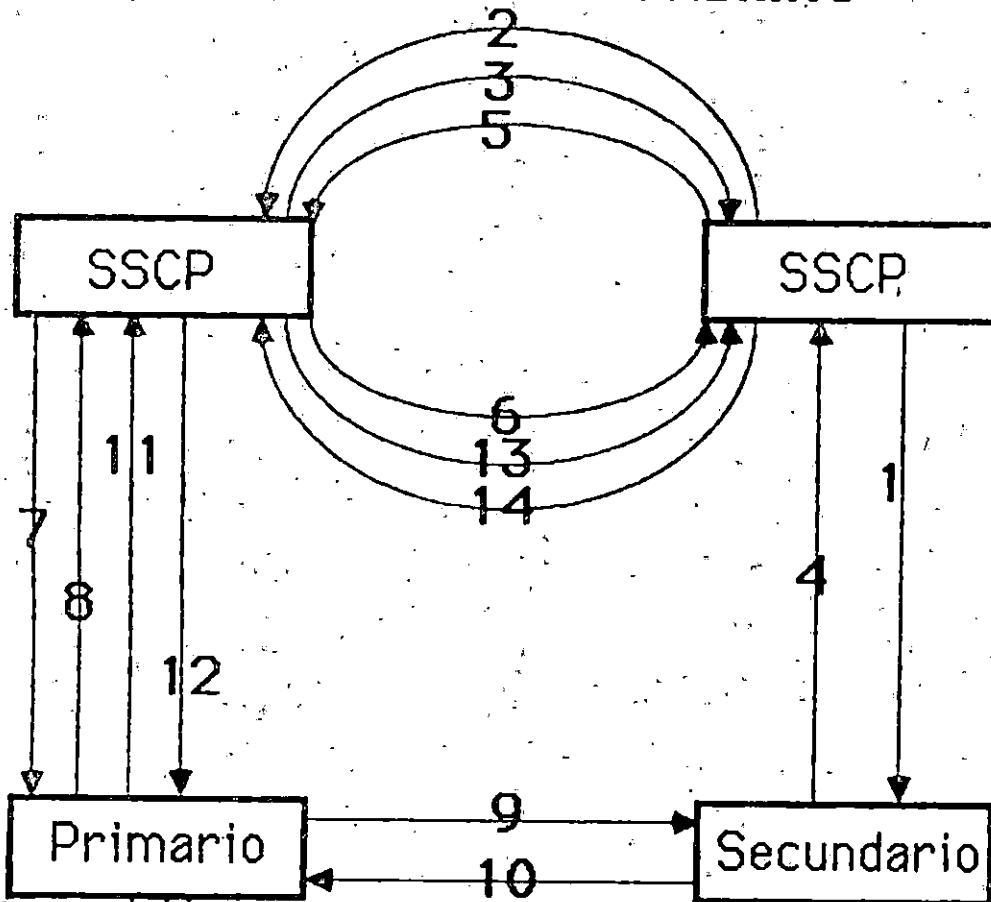
SNA (Transporte y Sesión)

- Manejo de conex en 2 niveles
- Data flow control layer
- RH header

Manejo de conex en 2 niveles

- Path control
 - Confiabilidad end-to-end
- Transmission control
 - Establecim y corte de conexiones (sesiones)
 - Cuando un proceso quiere comunic con otro
 - Debe solicitarlo al SSCP de su dominio
 - Proceso (NAU) siempre tiene sesión con SSCP
 - Hace solicitud a través de esa sesión
 - Si el proceso es remoto
 - SSCP busca dirección de red
 - Comunicación asimétrica
 - Primario y secundario
 - Asegura orden para capas superiores
 - Control de flujo
 - Igual que el de path control
 - Aplicado a sesiones no a máquinas

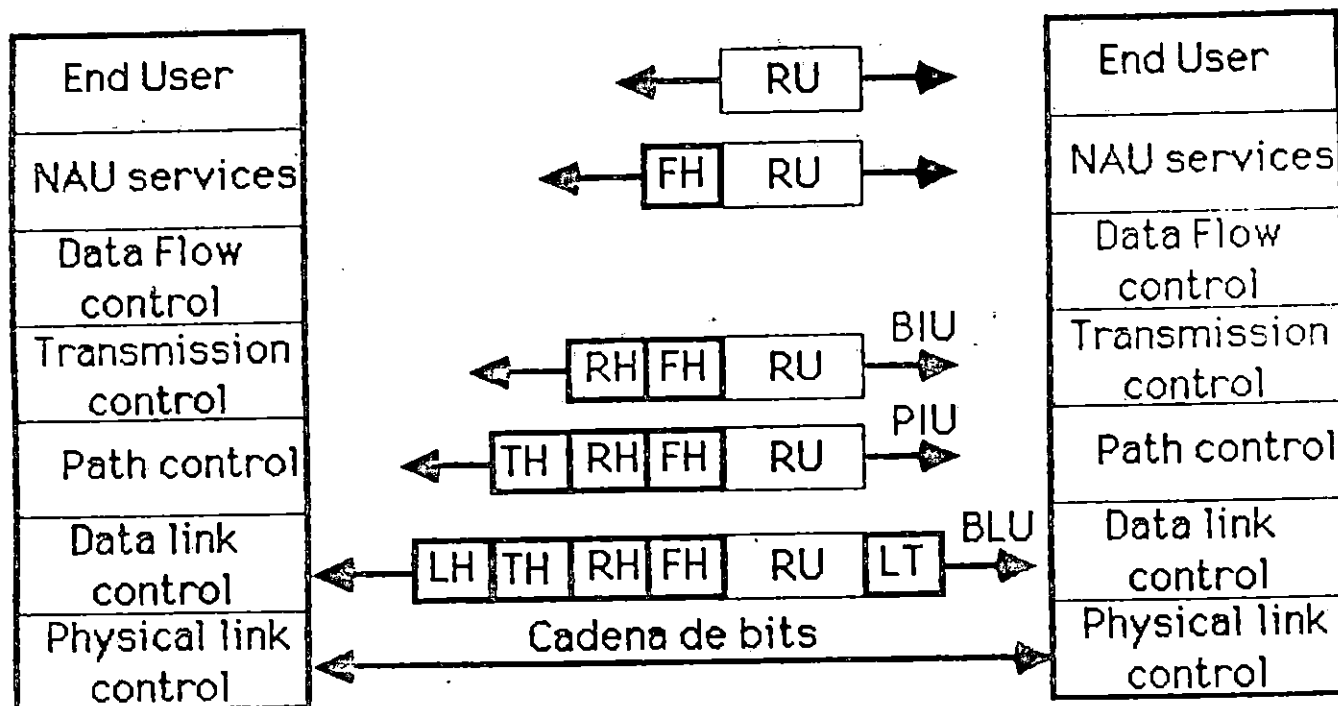
SNA - ESTABLECIMIENTO DE CONEXION REQUERIDA POR UN SECUNDARIO



1. Request Session
2. Request Session: Entrega Password
3. Acknowledgement: Si Password correcta, etc.
4. Acknowledgement
5. Initiate binding
6. Acknowledgement
7. Initiate binding
8. Acknowledgement
9. Bind: Siempre de primario a secundario. Parámetros de la sesión
10. Acknowledgement
11. Session Started
12. Acknowledgement
13. Session started
14. Acknowledgement

Data flow control layer

- Agrega servicios orientados al usuario
- Control de diálogo
 - Msgs que llegan pueden interr inoportunamte
 - Permite agrupar msgs en cadenas
 - En caso de fallas en medio de una cadena
 - Puede pedirse retr de cadena completa
 - Puede pedirse confirmación a nivel de cadena
 - Causa alternación estricta de cadena
 - Maneja transacciones atómicas
 - Agrupa conjunto de requerim y resp
 - Ejemplo: Reserva de avión
 - Control de concurrencia



LH = Link Header

LT = Link Trailer

TH = Transmission Header

RH = Request/Response Header

FH = Function Header

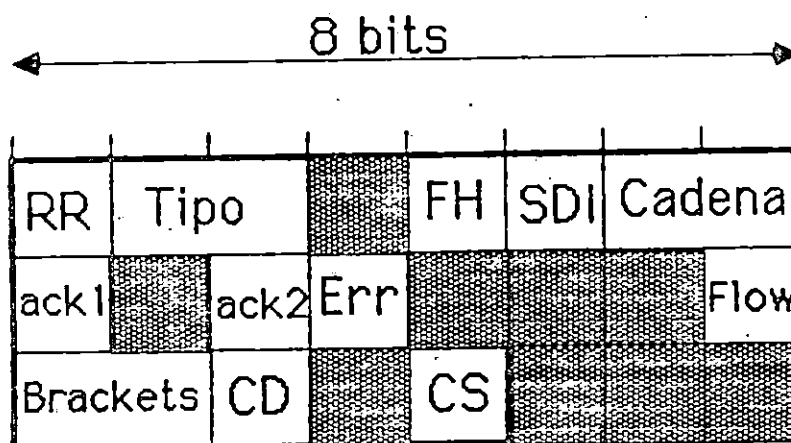
BLU = Basic Link Unit (=frame)

PIU = Path Information Unit (=Pckt)

BIU = Basic Information Unit (=Msg)

RU = Request/Response Unit

SNA - REQUEST RESPONSE HEADER



RH header

- Controla sesiones y diálogo
- RR: 0= requerimiento 1=respuesta
- Tipo: Indica qué hacer con el mensaje
 - 00: páselo a NAU services directo
 - 10: páselo a control de diálogo
 - 11: Msg de ctrl de esta sesión
 - 01: msg de ctrl de la red
- SDI: Indica si hay header FH
- Cadena
 - 10= principio de cadena
 - 00= mitad de la cadena
 - 01= final de la cadena
 - 11= cadena de un solo mensaje

- Campos de ack
 - Usados en requerimientos
 - Piden tipo de ack
 - Msg recibido
 - Msg recibido y procesam iniciado
 - Msg recibido y procesam completo
- Flow
 - Transm pide permiso para n msgs
 - Receptor concede o niega permiso
- Brackets
 - Controla transacciones atómicas
- CD
 - Cambia dirección del flujo en HDúplex

DECNET

- Protocolo llamado NSP
- Limpia y sencilla
- Usada directamente por los prgrs de usuario
- Comandos
- Tres opciones de Control de flujo

Comandos

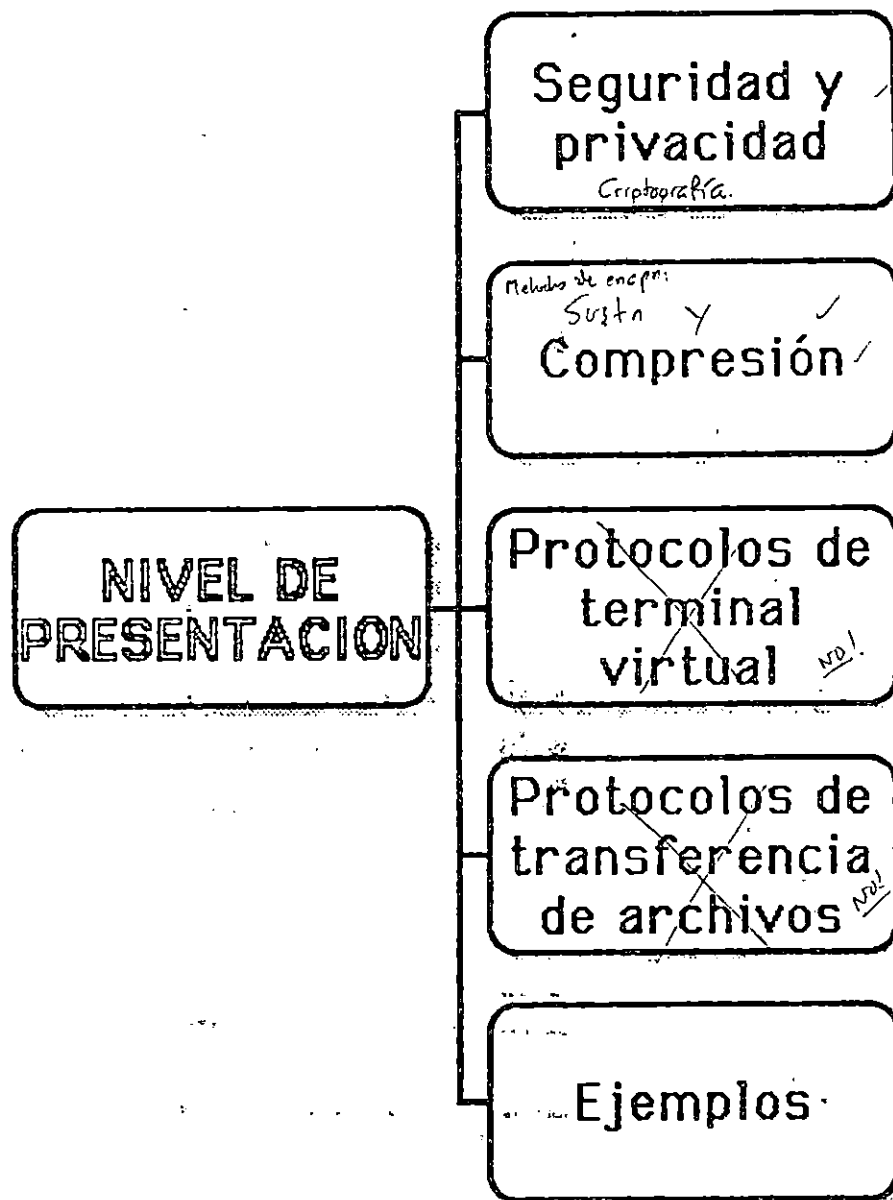
- **CONNECT REQUEST**
 - Parámetros
 - @ Máquina destino
 - Nombre del Proceso destino
 - Según convención del destinatario
 - Identificación de quien lo envía
 - Identificación de conexión
 - Para tráfico de salida
 - @ de bffr para datos de entrada
 - Otros
 - Genra envío de msg CONNECT INITIATE
- **LISTEN (RECEIVE CONNECT)**
 - Cuando quiere aceptar conexiones
- **ACCEPT CONNECT o REJECT CONNECT**
 - Usuario acepta o niega conexión

Comandos (cont'd)

- Si proceso receptor acepta conexión
 - Envía su #identif de conex
- Generan msgs
 - CONNECT CONFIRM
 - DISCONNECT INITIATE
- Para defenderse de duplicados viejos
 - Estac de transporte iniciadora
 - Debe devolver ack de estos msgs
- SEND (TRANSMIT)
 - Dos formas
 - Transf de msgs en bffr contiguo
 - Tamaño arbitrario
 - Transf de msgs por segmentos (pckts)
- Otros para cerrar e interrupciones

Tres opciones de Control de flujo

- Inexistente por nivel de transporte
- Por mensajes
- Por segmentos
 - Usa créditos igual que la anterior



Seguridad y privacidad

- + Introducción ←
- + Criptografía tradicional ←
- ✗ Estándar de encriptación de datos
- ✗ El problema de distribución de llaves
- ✗ Criptografía de llaves públicas
- ✗ Autenticación y firmas digitales

NDTple $\rightarrow$ no es capaz de resp. a las criden
del host; $\rightarrow$ NDSession

Energy $\left\{ \begin{array}{l} \propto \frac{1}{N} \propto \frac{1}{\rho} \\ \propto \frac{1}{N} \propto \frac{1}{\rho} \end{array} \right.$

- + ^{Problema → hay muchos pases en el mundo de encriptación = solución} Encriptación end-to-end

Criptografía tradicional

- + Introducción *Gschals*
- + Dos metodos tradicionales.

Introducción *Gschals*

- Desarrollada por cuatro gremios
- Modelo de Encripción
- Regla fundamental
- LLave
- Del lado del criptoanalista

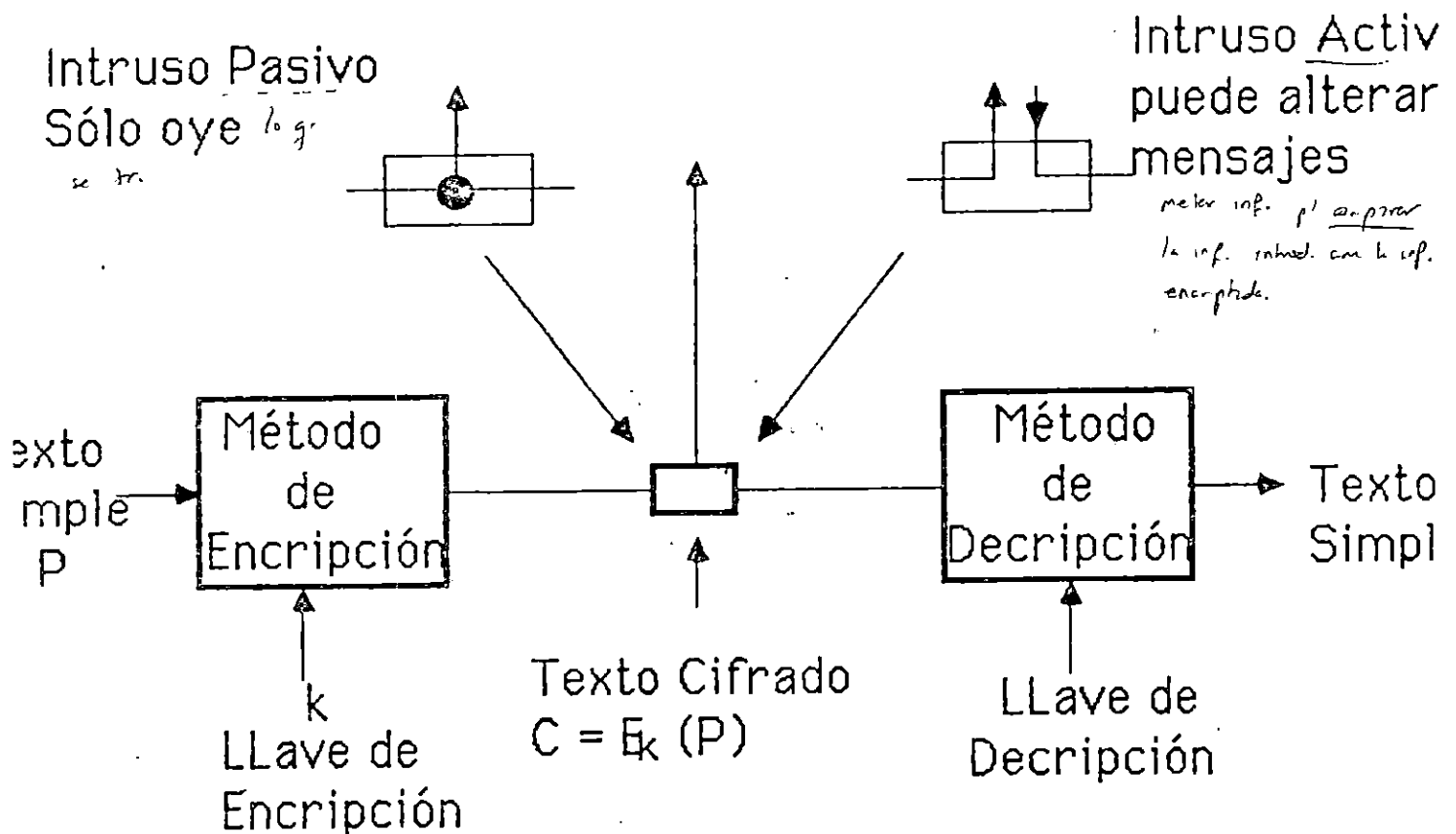
Desarrollada por cuatro gremios

- Militares
- Diplomáticos
- Periodistas
- Amantes

Modelo de Encripción

- Plaintext
- Ciphertext or cryptogram
- Criptología
 - Diseño de cifrados (Criptografía)
 - Criptanálisis: Arte de decifrar

MODELO DE ENCRIPCION



Regla fundamental

en criptografía

- Suponer que enemigo conoce método gral de *encriptación*.
- Dificultad de cambio de método

LLave → *es el estado de encriptación.*

- Cadena corta de caracteres
- Se puede cambiar frecuentemente *¿dinámico?*

Del lado del criptoanalista

3 niveles de posibilidad

- Ciphertext only → *solo hace texto encriptado o (Cifrado)*
- No sólo se debe proteger contra este caso
- Hay cadenas conocidas en sitios conocidos
 - PLEASE LOGIN
- Known Plaintext → *se conoce el texto original y el texto encriptado.*
- Chosen Plaintext → *meter mensajes en texto y recibir resp. por ellos y analizar*

Defensa por lo general de estos 3 prob's.

Dos metodos tradicionales

- (1) • Sustitución
- (2) • Transposición

(1) Sustitución

- Cambio de letras o grupo de letras *per otros*
- Ej: a* • Cifrado de César
- b* • Sustitución monoalfabética
- c* • Cifrado polialfabético
- d* • One time key
- e* • No siempre se hace sobre una letra
- f* • Códigos

(a) Cifrado de César

- *3 letras de en el alfabeto*
 $a \rightarrow D, b \rightarrow E, c \rightarrow F, d \rightarrow G, \text{ etc}$
- ataque = DWDTXH
- Generalización
- Desplazamiento *no es la clave* *sino 2* *displ.* variable (llave)

Sustitución monoalfabética

- Mapeo con letras desordenadas
- abcdefghijklmnopqrstuvwxyz
- qwertyuiop~~a~~sdfghjkl~~z~~xcvbnm
- LLave = cadena inferior
- ataque = QZQJXT
- 26! llaves distintas posibles
- Tomaría muchos años tratar todas
- Si 1 μ seg por llave --> 10**13 años
- Sin embargo es fácil de romper

vamos → letras q se repiten muy frecuent 2.

Sin embargo es fácil de romper

- Con texto cifrado
- Con estadíst de lenguajes naturales
- En inglés p. ej.
 - e es la letra más común seguida por t,a,o,n,i
 - Digramas comunes: th, in,er,re,an
 - Trigramas comunes: the,and,ion,ent
- Búsqueda de palabras esperadas
 - Mensaje entre cías financieras
 - CTBMN BYCTC BTJDS QXBNS GSTJC BTSWX
 - CTQTZ CQVUU QJSGS TJQZZ MNQJS VLNSX
 - VSZJU JDSTS JQUUS JUBXJ JDSKS UJSNT
 - KBGAQ JZBGY QTLCT ZBNYB NQJSW
- Búsqueda de "financiar"
 - i repetida con 4 letras en el medio
 - ocurre en 6,15, 27, 31, 42, 48, 56
 - 66, 70, 71, 76 y 83
 - sólo en 31 y 42 la n está ok
 - sólo en 31 la a está ok
- Después el trabajo es mucho más simple

Cifrado polialfabético

- Ej. Vigenère
 - Matriz cuadrada
 - 26 alfabetos de César
 - Fila A: abcde.....xyz
 - Fila B: ¹bcdef.....yza
 - Fila C: cdefg.....zab
 - Clave: Frase simple ↘
 - AUTOFANTASTICOAUTOFANTAST;
 - ⇒ I NVADANPARISPORELSURO ESTE
 - I trabaja con línea de la a
 - N con línea ^{fila} de la U, etc. ^{et de César}
- Mejora

Cifrado polialfabético (cont'd)

- Usar sustitución monoalfabética en c/fila
- Más poderoso pero difícil de usar
- Puede ser roto con ciphertext only :
 - Teniendo suficiente cantidad de texto cifrado
 - El truco es adivinar la longitud de la clave
 - Primero supone longitud k
 - Coloca texto cifrado en líneas k letras/línea
 - Si k correcto, distribución f de letras ok.
 - La más común 13%, sig 9%, etc.
 - Si no, cambia longitud de k .
 - Una vez se encuentra la longitud de la clave
 - C/fila se ataca como un cifrado monoalfab

$\swarrow$
 sr k ok

□	□	□	□	□	□
X	Q	T	V	E	S
A	P	W	Z	-	

4 ar.
 una clave

A U T O ← clave
 P O R F
 A V O R
 M I N C
 N L A E
 N T E A
 P A

estos letras fuer
 adividas en k =
 clave monoalfabética!
 -370-
 25- A → 132 veces
 }

One time key

idea:

- Usar una clave + larga que el texto.
- Inutiliza ataque anterior
- Ejemplo:
 - Construyamos cadena de bits al azar = llave
 - Convertamos texto simple en cadena de bits
 - Usando el código ASCII
 - XOR bit a bit de llave y texto simple
 - No se puede descifrar
 - Desventajas prácticas
 - Llave no memorizable. Copia escrita
 - Cantidad de datos limitado por tamaño de llave
 - Sincronización transm-recept con la llave
 - Podrían solucionarse empezando llave en c/msg
 - Ya no es una one time key
 - Puede romperse igual que polialfabética
 - Llave debe ser verdaderamente al azar
 - No usar pseudo generadores de #s al azar
 - Hay dependencia en la llave
 - Usar ruido cuántico de resistencias eléctricas

No siempre se hace sobre 1 letra

- Cifrado de Porta

- Matriz de 26X26 de letras *clave se tiene en fila y se el.*
- Codifica dos caracteres al tiempo
- 1er caracter = fila, 2º = columna
- Elemento de la intersec es el resultado
- Podría extenderse a 3 dimensiones
- Se codificarían trigramas
-

	a	e	i	o	u
a	e	o	a	i	u
e	<u>a</u>	a	e	e	o
i					
o					
u					

por la línea "ea"

-372-

ventaja → muy fácil de
usar desde el código secreto
sin de otro, etc.

$$\varphi: V^* \rightarrow N$$

Códigos

Edes → diccionarios de palabras y a c/u le damos un #.

- Encriptan textos del long variable
- Una parte
 - Palabras y código organizados igual
 - carga caribe cariño Carmen carnet carro --
 - 16142 16144 16149 16155 16160 16189
 - Usa un sólo catálogo a ambos lados → *el dir. lo da de ver el resp. y la parte se c!*
- ^{codig} Dos partes *(direct)* p' entrar q' se note q' hay orden.
 - carga caribe cariño Carmen carnet carro
 - 15202 40420 30012 80032 76290 39321
 - Necesita 2 catálogos a ambos lados
 - Mucho más difícil de llevar
 - Más bagaje
- Criptoanálisis
 - Código es como cifrado monoalf inmenso
- Desventajas: *Las palabras tienen estructura. lo q' pone es g'ra +!*
 - Requieren grandes catálogos
 - No son fáciles de reemplazar
- ^{monoc} Supercifrado
 - Combinac de códigos y cifrado polialfab
 - Texto simple se codifica
 - Dígitos de codificación se encriptan

(2) Transposición

- Cambia^{no las letras sino.} el orden de las letras
- Transposición columnar

Transposición columnar

- Criptografía
- Criptoanálisis

Criptografía

- Texto se escribe en filas y
- Se envía en columnas ^{de acuerdo a una clave.}
- Orden de columnas según clave

Transposición columnar

T R A P E C I O ← clave

8 7 1 6 3 2 4 5

M	I	N	E	N	E	L	C
A	N	A	L	D	E	L	A
M	A	N	C	H	A	C	O
N	B	O	M	B	A	S	T
I	P	O	Q	7	(I)	N	(I)
C	I	E	N	L	(A)	B	(O)
R	E	S	A	P	A	R	T
I	R	D	E	L	A	S	O
C	H	O	D	E	L	A	M
A	Ñ	N	A	D	E	M	A
Ñ	A	N	A	O	N	C	E
D	E	N	O	V	I	E	M
B	R	E					

TEXTO CIFRADO:

NANOOESDONNNEEEAAIAAALENN
 DHB7LPLDOVLLCSNBRSAMCECAOT
 IOTOMAEMELCMQNAEDAAOINABPIE
 RHÑAERMAMNICRICAÑDB

Criptoanálisis

con desenfoque!

- Descubrir ^{¿ sí es} ~~que es~~ transposición
 - Frecuencia de letras
 - Transposición no cambia letras
- Adivinar número de columnas
 - Palabra o frase probable
 - Sospecha: Inicien labores
 - Digramas IA, NB, IO, CR, IE, etc
 - Indican longitud de la clave
 - Si ^{long. es} clave 7 y no 8
 - IL, NA, IB, CO, hubieran ocurrido
- Ordenar las columnas
 - Si longitud de la clave es k
 - Pruebe $k(k-1)$ parejas de columnas
 - La que cuadre con estadíst de digramas es
 - Tome c/u de las otras a la derecha de las 2 1as
 - Busque mejor cuadre con estad de trigramas
 - Tome las C/u de las otras a la izq de las 2 1as
 - Repita
 - Es posible empezar a reconocer texto
 - Si ocurre INIICEN, la corrección es clara ^{¡ clar!}

QUIZZ DE CONGESTION Y EJEMPLOS DEL NIVEL DE RED

1. Cuál es la consecuencia final de la congestión?
 - a. Reinicialización (RESET) del circuito virtual
 - b. Escasez de buffers en los IMPs
 - c. Caída de uno o varios IMPs
 - d. Caída de uno o varios hosts
 - e. Ninguna de las anteriores
2. La técnica de control de congestión por preasignación de recursos es natural en redes:
 - a. Que usen como modelo interno los datagramas
 - b. Que usen como modelo interno los datagramas, pero que presenten interfaz de circuito virtual.
 - c. Que usen modelo interno de circuito virtual.
 - d. Que usen modelo interno de circuito virtual pero presenten interfaz de datagramas.
 - e. Ninguna de las anteriores
3. De las siguientes técnicas, cuál actúa solamente al detectar la congestión?
 - a. Preasignación de recursos
 - b. Control de Flujo
 - c. Restricción del número de paquetes en la red
 - d. Choke packets
 - e. Ninguna de las anteriores
4. Una de las siguientes técnicas de control de congestión ahorra memoria a cambio de mayor consumo de ancho de banda.
 - a. Preasignación de recursos
 - b. Descarte de paquetes
 - c. Restricción del número de paquetes en la red
 - d. Control de flujo
 - e. Ninguna de las anteriores

5. Cuál de las siguientes técnicas de control de congestión hace parecerse la conmutación de paquetes a conmutación de circuitos?
- a. Preasignación de recursos
 - b. Descarte de paquetes
 - c. Restricción del número de paquetes en la red
 - d. Choke packets
 - e. Ninguna de las anteriores
6. Una desventaja del control de congestión por limitación del número de paquetes en la red es:
- a. Hace control global de la red pero no garantiza control en algunos sectores de la red.
 - b. Consumo excesivo de memoria.
 - c. Mal funcionamiento en tráfico bajo
 - d. Consumo de ancho de banda en retransmisiones.
 - e. Ninguna de las anteriores
7. Una desventaja del control de congestión por control de flujo es:
- a. Mal funcionamiento en picos de una conexión aunque no haya congestión global.
 - b. Consumo excesivo de memoria.
 - c. Manejo complicado de buffers.
 - d. Consumo de ancho de banda en retransmisiones.
 - e. Ninguna de las anteriores
8. Qué técnicas garantizan que no se descarten paquetes?
- a. Choke packets
 - b. Control de flujo
 - c. Preasignación de recursos.
 - d. Restricción del número de paquetes en la red
 - e. Ninguna de las anteriores

9. El control de congestión por descarte de paquetes es natural usarlo en redes:
- a. Que usen como modelo interno los datagramas, pero que necesariamente presenten interfaz de circuito virtual.
 - b. Que usen modelo interno de circuito virtual.
 - c. Que usen modelo interno de circuito virtual pero necesariamente presenten interfaz de datagramas.
 - d. Que usen como modelo interno los datagramas
 - e. Ninguna de las anteriores
10. La razón para limitar el tamaño de las colas de salida en un IMP es:
- a. Aumentar la velocidad de propagación en las líneas de salida
 - b. Aumentar la velocidad de transmisión en las líneas de salida
 - c. Disminuir el número de paquetes descartados
 - d. Aumentar el throughput del IMP
 - e. Ninguna de las anteriores
11. La mejor forma de contar el número de permisos circulando en la red cuando se controla la congestión limitando el número de paquetes en la red es:
- a. Colocando un contador por cada línea en los IMPs.
 - b. Colocando un contador por cada buffer en los IMPs.
 - c. Durante el intercambio de información de enrutamiento
 - d. Haciendo enrutamiento centralizado
 - e. Ninguna de las anteriores
12. La manera como ARPANET resolvió el abrazo mortal de reensamblaje fué:
- a. Exigiendo al IMP fuente pedir permiso al destino antes de enviar un mensaje multipaquete.
 - b. Colocando un buffer de entrada por cada línea
 - c. Designando un pool de buffers de emergencia
 - d. No permitiendo mensajes multipaquete.
 - e. Ninguna de las anteriores

13. La red ejemplo que usa un sistema parecido a choke packets para control de congestión es:
- a. ARPANET
 - b. DECNET
 - c. SNA
 - d. X.25
 - e. Ninguna de las anteriores
14. El método de control de congestión de X.25 es:
- a. Preasignación de recursos
 - b. Descarte de paquetes
 - c. Restricción del número de paquetes en la red
 - d. Choke packets
 - e. Ninguna de las anteriores
15. Si la ruta más larga posible de una red ha sido fijada en M , cuántos buffers necesita cada IMP para poder implementar la técnica de Merlin y Schweitzer?
- a. M
 - b. $M-1$
 - c. $M+1$
 - d. $M/2$
 - e. Ninguna de las anteriores
16. Por qué el paquete de CALL REQUEST de X.25 contiene en uno de sus campos el número del circuito virtual, si no ha sido establecido?
- a. Por que un host siempre usa siempre el mismo número de circuito virtual
 - b. Porque el host que llama escoge el número del circuito virtual
 - c. Es una propuesta que la red confirma
 - d. Ese campo va vacío y la red lo llena y se lo devuelve
 - e. Ninguna de las anteriores

17. Qué sentido tiene en SNA usar varios grupos de transmisión entre dos IMPs?

- a. Aumentar el throughput transmitiendo por varios simultáneamente.
- b. Permitir un mejor control de flujo
- c. Permitir el uso de datagramas
- d. Adaptarse mejor al tipo de servicios
- e. Ninguna de las anteriores

18. En SNA una ruta explícita es:

- a. La sucesión de grupos de transmisión que conforman la ruta.
- b. La sucesión de subáreas que conforman la ruta.
- c. La ruta interna de una subárea
- d. El número de líneas que se van a utilizar de un cierto grupo de transmisión.
- e. Ninguna de las anteriores

19. DECNET:

- a. Usa circuitos virtuales interna y externamente
- b. Usa circuitos virtuales internamente pero presenta interfaz de datagrama
- c. Usa datagramas pero presenta interfaz de circuito virtual
- d. Usa datagramas interna y externamente
- e. Ninguna de las anteriores

20. ARPANET

- a. Usa circuitos virtuales interna y externamente
- b. Usa circuitos virtuales internamente pero presenta interfaz de datagrama
- c. Usa datagramas pero presenta interfaz de circuito virtual
- d. Usa datagramas interna y externamente
- e. Ninguna de las anteriores