

## Nivel enlace

Este nivel se encarga de: sincronización de tramas, control de flujos (velocidad de transmisión de las tramas) y de errores, direccionamiento, y gestión de enlaces.

El **control de flujos** es una técnica que controla que un emisor no desborde al receptor. Por ejemplo, si no hay control de flujos, un emisor puede enviar un mensaje a un receptor que tenga llena la memoria intermedia de datos, que tiene una longitud máxima, y antes de pasarlo al nivel superior ha de procesarlo, pero si está ocupado con datos anteriores, este mensaje sería eliminado.

Cada fila representa una única trama que transporta un enllaç de dades entre dues estacions. Les dades s'envien en una seqüència de trames, i cada trama conté una part de les dades i informació de control. El temps que triga una estació en emetre tots els bits d'una trama cap al mitjà és el **temps de transmissió**; aquest temps és proporcional a la longitud de la trama. El **temps de propagació** és el temps que triga un bit en travessar l'enllaç entre l'origen i la destinació.

Las tramas se envían en el orden que se reciben, y sufren retardos de manera independiente.

## Técnicas de control de flujo

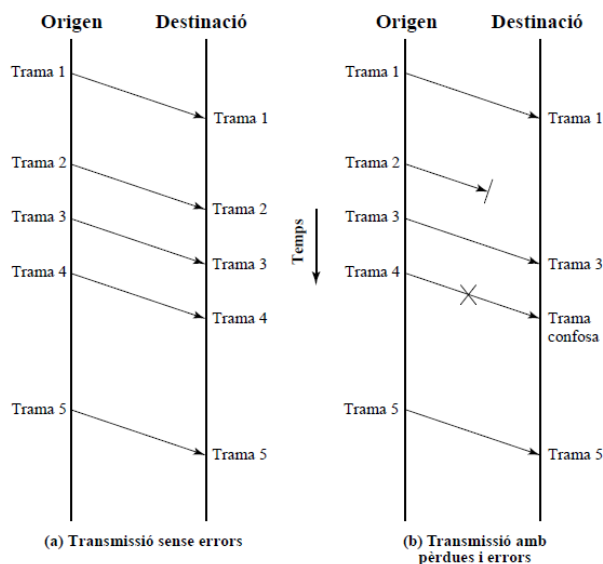
El control de fluxos és una tècnica per garantir que una entitat transmissora no desborda una entitat receptora amb massa dades.

### Stop & Wait Flow Control

Una entitat d'origen transmet una trama. Després que l'entitat de destinació rebí la trama, indica si vol acceptar una altra trama enviant un reconeixement a la trama que acaba d'arribar. L'origen ha d'esperar a rebre el reconeixement abans d'enviar la trama següent. Per tant, la destinació pot aturar el flux de dades si reté el reconeixement. Esto genera un problema, solo puede transmitir una trama a la vez. Al acabar de enviar una trama, el receptor envía un ACK.

Es menos eficiente si envía tramas grandes, por lo que suele dividir las en otras más pequeñas, esto es mejor así porque:

- El tamaño del buffer es limitado.
- Si se envían tramas grandes tarda más, y hay más probabilidad de fallo, además, en este caso, hay que volver a transmitirla en tramas más pequeñas.
- En un medio compartido es conveniente que no se ocupe la línea mucho rato.



Quan  $a$  és inferior a 1, el temps de propagació és inferior al temps de transmissió. En aquest cas, la trama és prou llarga i els primers bits de la trama arriben a la destinació abans que l'origen hagi finalitzat la transmissió de la trama.

Quan  $a$  és superior a 1, el temps de propagació és superior al temps de transmissió. En aquest cas, l'emissor completa la transmissió de tota la trama abans que els bits inicials de la trama arribin al receptor. Dit d'una altra manera, els valors més grans d' $a$  corresponen a velocitats de dades superiors i/o a distàncies més llargues entre estacions. En este caso la trama nunca utiliza la capacidad completa del enlace.

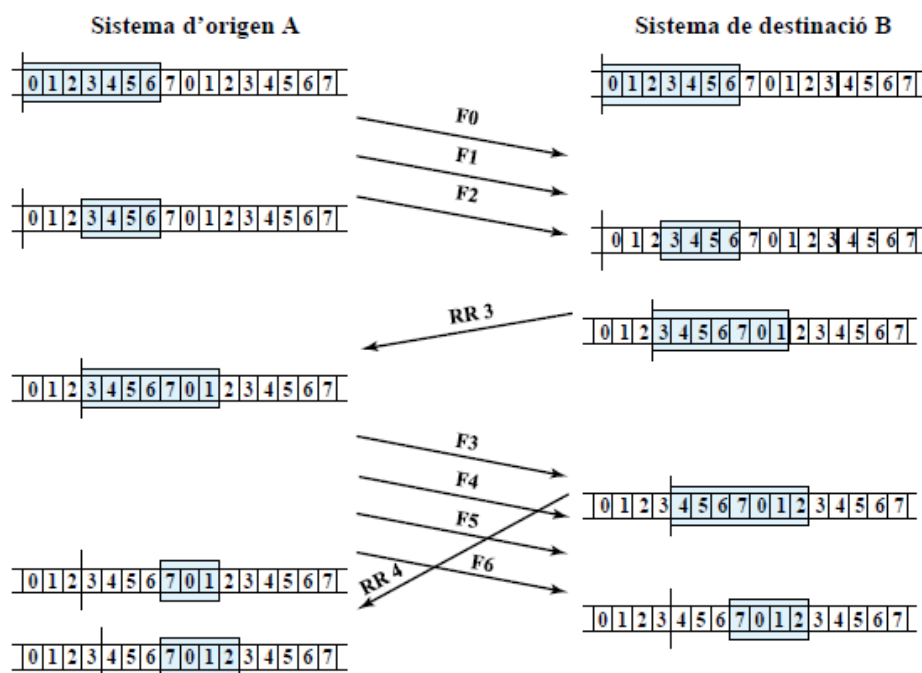
## Sliding Window Flux Control

Para  $a > 1$  ( $t_{prop} > t_{trans}$ ) se producen muchas deficiencias, esto se puede resolver enviando más de una trama a la vez. En este caso, cada trama se etiqueta con un número de secuencia, A envía a B un número  $x$  de tramas a la vez, A tiene una lista de tramas que enviar y B tiene una lista de tramas que espera recibir, estas listas se llaman **ventanas**. B puede hacer el reconocimiento de varias tramas a la vez, este tramo de reconocimiento (trama  $x, y, z$  por ejemplo).

**Para un tramo de  $k$  bits, la numeración de secuencia puede ir del 0 al  $2^k - 1$ , en módulo  $2^k$ . Este máximo es también el tamaño máximo de ventana.**

El flujo avanza, es decir, la ventana se reduce en uno y se aumenta según recibe reconocimientos, y el emisor va enviando tramas. El receptor puede parar la emisión con la señal RNR (receive not ready), que significa que ha recibido las anteriores pero ha de parar, por lo tanto RNR 5 indica que ha reconocido hasta la 4, aunque hayan llegado más. Para volver a abrir la emisión el receptor envía un reconocimiento.

El **piggybacking** es la técnica usada en full-duplex, en la que cada estación envía y recibe tramas a la vez, por lo que tienen dos ventanas cada uno. Cada **trama de dades** inclou un camp per al número de seqüència de la trama i un camp que té el número de seqüència utilitzat per al reconeixement.



## Técnicas de control de errores

El **control de errores** es la técnica que se utiliza para detectar y corregir errores en la transmisión de una trama. Hay dos tipos de errores:

- Trama perdida: una trama no llega al destinatario, por ejemplo debido a una ráfaga de ruido.
- Trama errónea: una trama ha llegado pero sus bits han sido afectados durante la transmisión.

Les tècniques més comunes per al control d'errors es basen en alguns d'aquests elements, o en tots ells:

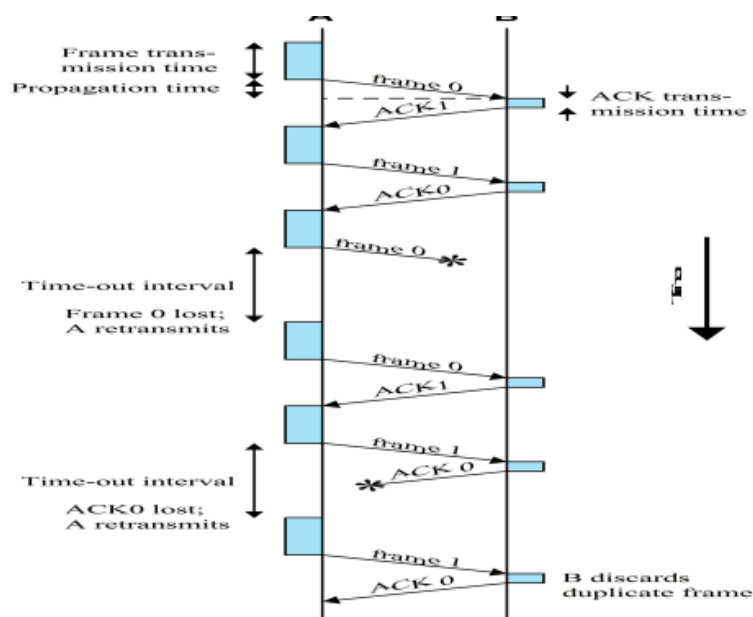
- • **Detecció d'errors:** HEC.
- • **Reconeixement positiu:** la destinació retorna un reconeixement positiu per a trames que arriben correctament i sense errors.
- • **Retransmissió després d'excedir el temps d'espera:** l'origen transmet una trama que no s'ha reconegut després d'un període de temps determinat.
- • **Reconeixement negatiu i retransmissió:** la destinació retorna un reconeixement negatiu de trames si es detecta un error. L'origen retransmet aquestes trames.

Estos mecanismos los lleva a cabo la técnica de **ARQ**, del cual hay tres versiones: **Stop-Wait ARQ**, **Go-Back-N ARQ**, i **Selective-reject ARQ**. Tienen como objetivo convertir un enlace no fiable en uno fiable.

### Stop-Wait ARQ

L'estació d'origen transmet una única trama i després ha d'esperar un reconeixement (ACK). No es pot enviar cap altra trama de dades fins que la resposta de l'estació de destinació no arriba a l'estació d'origen. Si no se recibe la trama tras el tiempo temporizador, el origen vuelve a enviar la trama. También sufre el problema de que solo puede enviar una trama a la vez.

Pueden producirse dos errores, una trama errònea, que es descartada y el origen, al no recibir ACK, la vuelve a enviar, o un reconocimiento erròneo, que es un RR que no llega, y el origen vuelve a enviar la trama, por lo que se aceptan dos copias de una trama, para evitar esto hay ACK0 y ACK1, y se etiquetan las tramas con 0 o 1, para que al enviar ACK0, se entienda que espera una trama 1.



## Go-Back-N ARQ

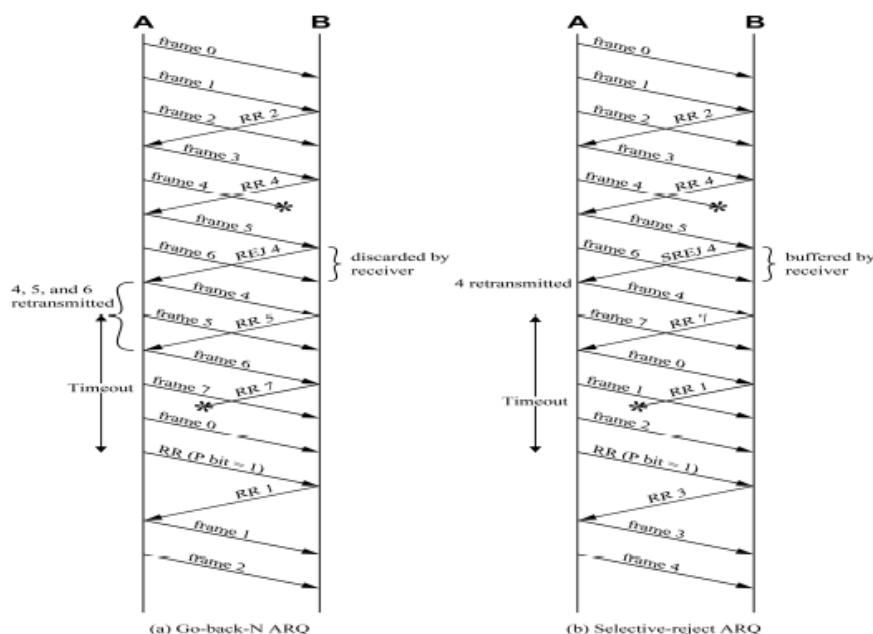
Sería la técnica con ventana que permite enviar muchas tramas a la vez. Es igual que la técnica de control de flujo, pero si hay una trama errónea, envía un REJ y descarta todas las tramas que llegaron en esa ola, por lo que el emisor ha de volver a enviar esas tramas. El tamaño de ventana vuelve a ser el de  $0 - 2^k - 1$ .

Los posibles errores son:

- Una trama errónea: que es descartada y el origen, al no recibir ACK, la vuelve a enviar. Puede pasar que:
  - i) la envíe desordenada, en tal caso recibe REJ y ha de enviar esa y las posteriores.
  - ii) A no envía nada, y B tampoco, en tal caso se acaba el temporizador y A envía RR y  $P=1$ , que indica que B ha de enviar un RR indicando la trama que espera, por lo que B envía RR  $i$  y A le manda la trama indicada.
- RR erróneo:
  - i) B envía RR y no llega a A, por lo que el siguiente RR es de una trama posterior, y A lo recibe antes de recibir la que falta.
  - ii) A no recibe un RR, y envía un RR, iniciando el temporizador P, pero sigue sin recibir nada, una y otra vez hasta que acaba inicializando.
- REJ erróneo: si se pierde la trama: se acaba el temporizador y A envía RR y  $P=1$ , que indica que B ha de enviar un RR indicando la trama que espera, por lo que B envía RR  $i$  y A le manda la trama indicada.

## Selective-Reject ARQ

Amb l'ARQ de rebuig selectiu, les úniques trames que es retransmeten són les que reben un reconeixement negatiu, en aquest cas anomenat SREJ, o les que sobrepassen el temps d'espera. El transmissor també ha de tenir una lògica més complexa per poder enviar una trama fora de seqüència. Aquestes complicacions fan que l'ARQ de rebuig selectiu s'utilitzi molt menys que l'ARQ de retrocedir N. El rebuig selectiu és una bona opció per a l'enllaç satèl·lit perquè implica un retard estès de propagació.



## Control de enlaces de datos

### HDLC (High Level Data Link Control)

Este protocolo define tres **tipos de estaciones**:

- **Estació principal:** és la responsable de controlar el funcionament de l'enllaç. Les trames que emet aquesta estació s'anomenen ordres.
- **Estació secundària:** opera sota el control de l'estació principal. Les trames que emet una estació secundària s'anomenen respostes. L'estació principal manté un enllaç lògic diferent per a cada estació secundària de la línia.
- **Estació combinada:** combina les característiques de les estacions principal i secundària. Una estació combinada pot emetre ordres i respostes.

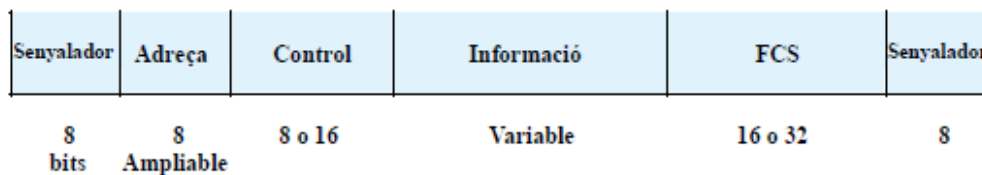
Les dues **configuracions d'enllaç** són:

- **Configuració no equilibrada:** consisteix en una estació principal i una o diverses estacions secundàries i admet tant la transmissió de dúplex complet i semidúplex.
- **Configuració equilibrada:** consisteix en dues estacions combinades i admet la transmissió de dúplex complet i semidúplex.

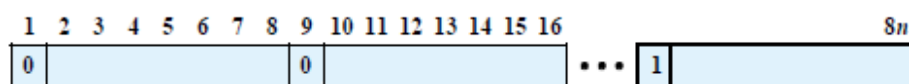
El modo de transferencia que utiliza LLC es:

- **Mode equilibrat asíncron (ABM):** s'utilitza amb una configuració equilibrada. Qualsevol de les estacions combinades pot iniciar la transmissió sense rebre permís de l'altra estació combinada.

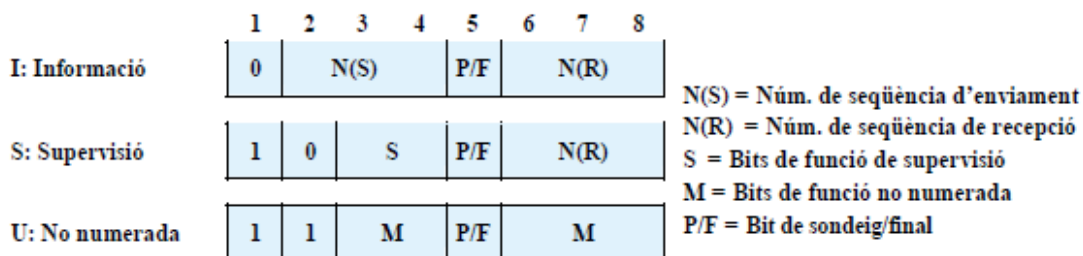
### Estructura de la trama



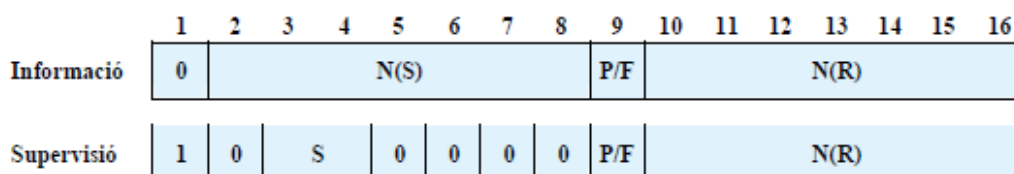
(a) Format de trama



(b) Camp d'adreça ampliada



(c) Format de camp de control de 8 bits



(d) Format de camp de control de 16 bits

- **Señalador:** Se usa para sincronismo con la secuencia fija 01111110. Como no es seguro que a lo largo de la trama no salga esta frecuencia, se detecta si hay una serie de 1's y se modifica la trama.
- **Dirección:** indica el identificador de la estación secundaria que ha emitido la trama o la ha de recibir. No es necesario en enlaces punto a punto.
- **Control:** Defineix tres tipus de trames, cadascuna amb un format de camp de control diferente:
  - o Les **trames d'informació** (trames I) inclouen les dades que cal transmetre per a l'usuari. Si se usa ARQ piggyback, usa dos listas.
  - o Les **trames de supervisió** (trames S) proporcionen el mecanisme ARQ quan no s'utilitza l'encavalcament.
  - o Les **trames no numerades** (trames U) proporcionen funcions addicionals de control d'enllaç.

Tots els formats de camp de control inclouen el bit de sondeig/final (P/F). A les trames de ordre, el bit P indica que necessita rebre un RR amb la trama següent. A les trames de resposta, el bit F indica la trama de resposta a la sol·licitud de P.

- **Información:** solo en tramas I y algunas U.
- **Secuencia de comprobación de trama:** contiene un código de detección de errores que se calcula a partir del resto de bits de la trama.

## Funcionamiento (más en cap.7, p.263 del libro)

El funcionament del protocol HDLC consisteix a intercanviar trames I, trames S i trames U entre dues estacions.

### 1) Inicialización

Uno de las dos estaciones inicializa el enlace para permitir el intercambio de tramas con un **SABM**. Indica a l'altre costat que s'ha sol·licitat la inicialització. Si se acepta la sol·licitud, la otra estación envía una trama no numerada **UA**, si no, envía una de modo desconectado **DM**.

### 2) Transferencia de datos

Las dos estaciones intercambian datos de usuario e información de control para llevar a cabo el control de flujos y de errores.

Tots dos costats comencen a enviar dades d'usuari en trames I, començant pel número de seqüència 0.

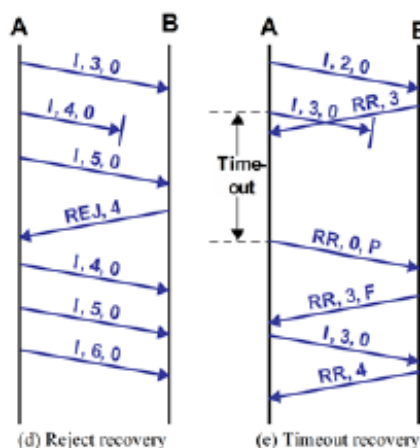
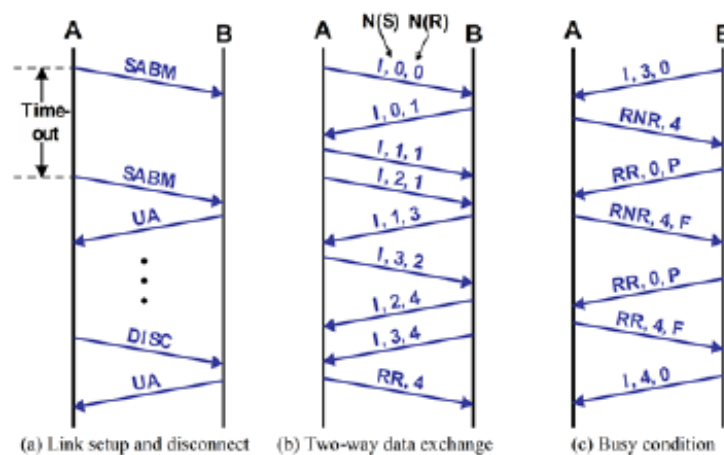
Es col·locarà el número de seqüència a **N(S)**. **N(R)** és el reconeixement per a les trames I rebudes, és la última trama confirmada.

Les trames S també s'utilitzen per al control de fluxos i el control d'errors. La **RR** (preparat per rebre) reconeix la darrera trama I rebuda indicant quina és la propera trama I que espera. La trama RR s'utilitza si no hi ha trànsit de dades d'usuari invers (trames I) per dur un reconeixement. La trama **RNR** (no preparat per rebre) reconeix una trama I, amb RR, però també demana que l'entitat igual suspengui la transmissió de trames I. Quan l'entitat que ha emès la trama RNR torna a estar a punt, envia una trama RR. **REJ** inicia l'ARQ de retrocedir N. Indica que la darrera trama I que s'ha rebut s'ha rebutjat i que cal tornar a transmetre totes les trames I que comencen pel número N(R). El rebuig selectiu (**SREJ**) s'utilitza per sol·licitar la retransmissió només d'una trama.

### 3) Desconnexió

Qualsevol dels mòduls HDLC pot iniciar una desconnexió per iniciativa pròpia si hi ha algun tipus d'error o a petició de l'usuari de la seva capa superior. Per emetre una desconnexió, el mòdul HDLC envia una trama de desconnexió (DISC). Per acceptar la desconnexió, l'entitat remota ha de respondre amb una UA i ha de notificar l'usuari de la capa tres que la connexió ha finalitzat amb una trama I no reconeguda.

Nom	Ordre/ resposta	Descripció
<b>Informació (I)</b>	O/R	Intercanviar dades de l'usuari
<b>Supervisió (S)</b>		
Preparat per rebre (RR)	O/R	Reconeixement positiu; preparat per rebre trama I
No preparat per rebre (RNR)	O/R	Recepció positiva; no preparat per rebre
Rebuig (REJ)	O/R	Recepció negativa; retrocedir N
Rebuig selectiu (SREJ)	O/R	Recepció negativa; rebuig selectiu
<b>No numerada (U)</b>		
Definir resposta normal/mode estès (SNRM/SNRME)	O	Definir mode; estès = números de seqüència de 7 bits
Definir resposta asíncrona/estès (SARM/SARME)	O	Definir mode; estès = mode de números de seqüència de 7 bits
Definir equilibri asíncron/estès (SABM/SABME)	O	Definir mode; estès = mode de números de seqüència de 7 bits



## Capa 2 LLC/MAC

### LLC (Logical Link Protocol)

LLC se basa en HDLC, y tiene como objetivo el control de flujo y de errores, se preocupa de la transmissió d'una PDU de nivell d'enllaç entre dues estacions. És compatible amb el multiaccés i el mitjà compartir. A més, no té tots els detalls d'accés a l'enllaç del nivel de MAC. L'encaminament en l'LLC implica especificar els usuaris d'LLC d'origen i de destinació. Normalment, un usuari és un protocol de capa superior o una funció de gestió de xarxa a l'estació. Es fa referència a aquestes adreces d'usuari d'LLC com a punts d'accés a servei (SAP), mantenint la terminologia d'OSI per a l'usuari d'una capa de protocol.

Especifica els mecanismes per encaminar estacions al llarg del mitjà i per controlar l'intercanvi de dades entre dos usuaris. Es proporcionen tres serveis com a alternatives per als dispositius connectats fent servir l'LLC:

- **Unknowledged conectionless Service:** aquest servei és un servei d'estil datagrama. És un servei molt simple que no implica cap dels mecanismes de control de fluxos o errors. (**Operación de tipo 1**).
- **Connection-mode Service (CS):** aquest servei és semblant al que ofereix l'HDLC-ABM. Es configura una connexió lògica entre dos usuaris que intercanvien dades, i es proporciona control d'errors i de fluxos. (**Operación de tipo 2**).
- **Acknowledge conectionless Service:** és una combinació entre els dos serveis anteriors. Proporciona els datagrames que cal reconèixer, però no es configura cap connexió lògica anterior. (**Operación de tipo 3**).

### Estructura de las tramas:

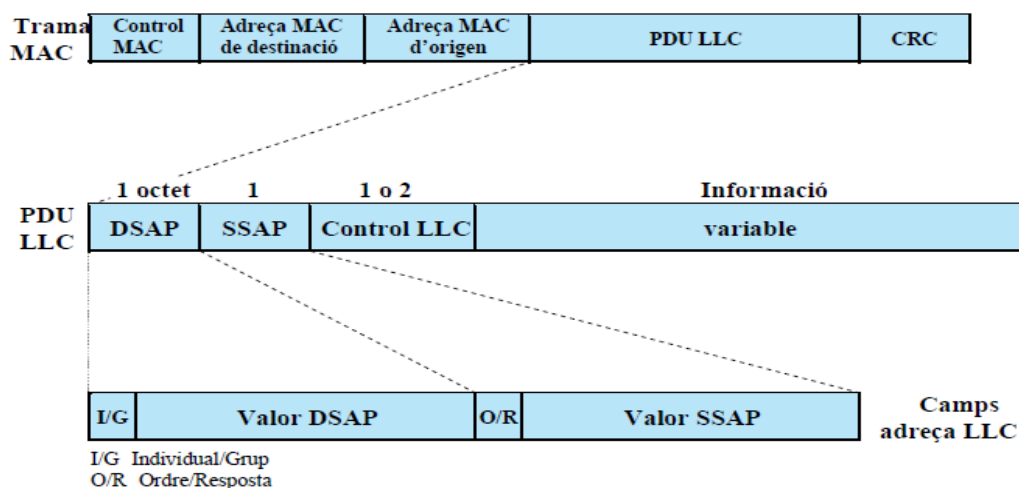


Figura 15.7 PDU LLC en un format de trama MAC genèric

La información de control se lleva a cabo en la trama LLC, que luego pasa la información a la MAC y añade a los lados su información de control.



#### En las tramas LLC:

- **DSAP (punt d'accés de servei de destinació) i SSAP (punt d'accés de servei d'origen):** contenen cadascun una adreça de 7 bits, que especifica els usuaris d'origen i de destinació de l'LLC.
- **I/G:** Un bit del DSAP indica si el DSAP és una adreça d'una persona o d'un grup.
- **O/R:** Un bit del SSAP indica si la PDU és una PDU d'ordre o de resposta.

Las tramas de operación 1 (UI) transportan datos de usuario.

Las tramas de operación 2 (SABM, UA, DM y DISC) se encargan de establecer la conexión lógica. SABM pide la conexión y el UA confirma, o DM indica un rechazo. Cuando la conexión termina el destino emite un DISC.

Las tramas de operación 3 (AC) utilizan un bit para evitar pérdida de tramas.

#### **MAC**

Su función es ensamblar y desensamblar paquetes, detectar errores, reconocer direcciones y dirigir el acceso al medio de la LAN.

Cal proporcionar alguns sistemes per controlar l'accés al mitjà de transmissió, per a un ús eficaç i ordenat d'aquesta capacitat. Aquesta és la funció d'un **protocol de control d'accés a mitjans (MAC)**. En un esquema centralitzat, es designa un controlador que té l'autoritat de garantir l'accés a la xarxa. Una estació que vulgui transmetre ha d'esperar fins que rebí el permís del controlador. En una xarxa descentralitzada, les estacions duen a terme col·lectivament una funció de control d'accés als mitjans per determinar dinàmicament l'ordre de transmissió de les estacions.

Como LLC utiliza MAC, ya no necesita sincronismo. Y LLC proporciona a MAC poder realizar circuitos virtuales.

La capa MAC rep un bloc de dades de la capa LLC i és responsable de dur a terme funcions relacionades amb l'accés als mitjans i per transmetre les dades.

#### Las tramas MAC:

- **Control de MAC:** aquest camp conté qualsevol informació de protocol de control necessària per al funcionament del protocol MAC. Per exemple, aquí podria indicarse un nivell de prioritat.
- **Adreça MAC de destinació:** el punt de connexió física de destinació a la LAN per a aquesta trama.
- **Adreça MAC d'origen:** el punt de connexió física d'origen a la LAN per a aquesta trama.
- **LLC:** les dades LLC de la capa immediatament superior.
- **SRC:** el camp de comprovació de redundància cíclica (també conegut com a camp de seqüència de comprovació de trama, FC). Aquest és un codi de detecció d'errors, com hem vist a l'HDLC i a d'altres protocols de control d'enllaços

La capa MAC és responsable de la detecció d'errors i l'eliminació de les trames que es troben en situació d'error. Opcionalment, la capa LLC fa un seguiment de les trames que s'han rebut correctament i retransmet les trames no correctes.

La trama MAC original consta de:

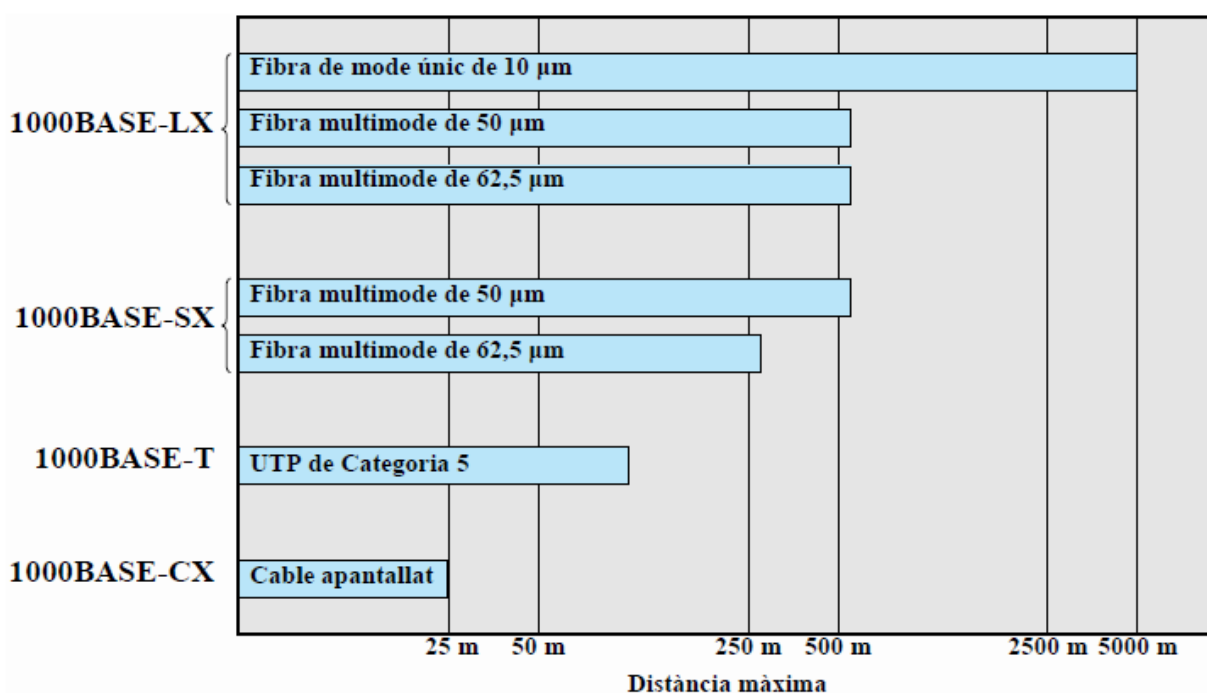
- **Patró:** un patró de 7 octets de 0 i 1 alternatius que el receptor utilitza per establir la sincronització de bits.
- **Delimitador de trama d'inici (SFD):** la seqüència 10101011, que indica l'inici real de la trama i permet que el receptor localitzi el primer bit de la resta de la trama.
- **Adreça de destinació (DA):** especifica les estacions a les quals està destinada la trama. Pot ser una adreça física única, un grup d'adreces o una adreça global.
- **Adreça d'origen (SA):** especifica l'estació que ha enviat la trama.
- **Longitud/Tipus:** longitud del camp de dades LLC en octets, o camp de tipus d'Ethernet, en funció de si la trama és compatible amb l'estàndard IEEE 802.3 o l'especificació Ethernet anterior. En cada cas, la mida de trama màxima, sense incloure el patró i l'SFD, és de 1518 octets.
- **Dades LLC:** unitat de dades proporcionada per l'LLC.
- **Coixí:** octets que s'afegeixen per assegurar que la trama és suficientment llarga per a una operació adequada del CD.
- **Seqüència de verificació de trama (FCS):** una comprovació de redundància cíclica de 32 bits

Tambien existen otros tipos de tramas MAC: la trama tagged que permite usar VLAN, la trama QinQ que permite crear circuitos virtuales y anidar VLAN. Y Mac in Mac que permite hacer tunneling porque permite crear niveles de conmutación diferentes, y jerarquizarlos.

## Ethernet

**Taula 16.2** Alternatives de mitjà de capa física IEEE 802.3 a 10 Mbps

	10BASE5	10BASE2	10BASE-T	10BASE-FP
Mitjà de transmissió	Cable coaxial (50 ohm)	Cable coaxial (50 ohm)	Parell trenat sense apantallar	Parell de fibra òptica de 850 nm
Tècnica de senyalització	Baseband (Manchester)	Banda de base (Manchester)	Banda de base (Manchester)	Manchester/ activat-desactivat
Topologia	Bus	Bus	Estrella	Estrella
Longitud de segment màxima (m)	500	185	100	500
Nodes per segment	100	30	—	33



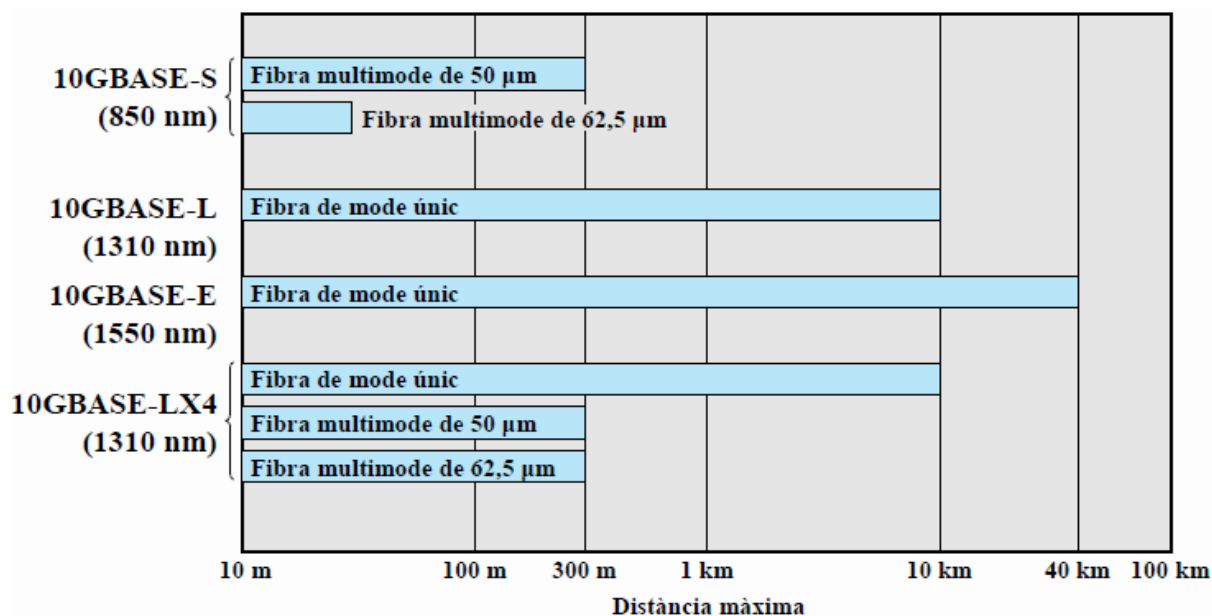


Figura 16.7 Opcions de distància d'Ethernet a 10 Gbps (escala de registre)

## Media Options for 40-Gbps and 100-Gbps Ethernet

	40 Gbps	100 Gbps
1m backplane	40GBASE-KR4	
10 m copper	40GBASE-CR4	1000GBASE-CR10
100 m multimode fiber	40GBASE-SR4	1000GBASE-SR10
10 km single mode fiber	40GBASE-LR4	1000GBASE-LR4
40 km single mode fiber		1000GBASE-ER4

Naming nomenclature:

Copper: K = backplane; C = cable assembly

Optical: S = short reach (100m); L - long reach (10 km); E = extended long reach (40 km)

Coding scheme: R = 64B/66B block coding

Final number: number of lanes (copper wires or fiber wavelengths)

## MPLS

Es un protocolo de nivel 3 que se encarga de forwardear paquetes con etiquetas en lugar de IPs. Esto permite: garantizar calidad de servicio a tiempo real, un cierto delay y valores de pérdidas para voz y video. También permite hacer VPN's, traffic engineering, y soporte multiprotocolo. Los componentes son:

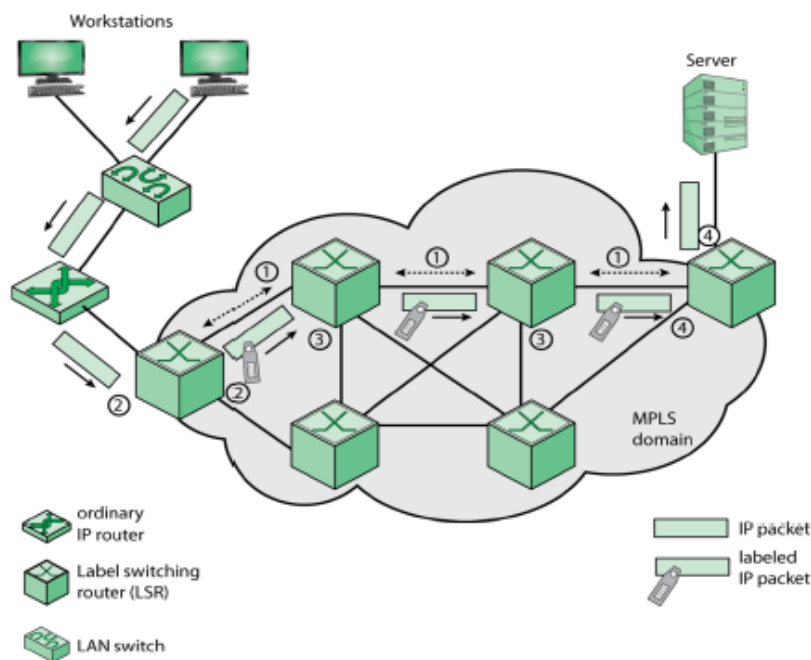
- **LSR (Label Switched Router):** routers que pasan paquetes e identifican los flujos entre end points. Cada uno construye una tabla que describe cómo los paquetes han de ser tratados, esta tabla se llama **LIB (Label Information Base)**
- **FEC (Forwarding Equivalence Class):** describe una serie de flujos que irán por un mismo path, por lo que compartirán las mismas etiquetas y serán tratados de la misma forma. Un flujo está definido por: IP orig. y dest., un puerto orig. y dest.,
- **LSP (Label Switched Path):** describe un camino.
- **LDP (Label Distribution Protocol):** protocolo para label distribution.

Todos los paquetes de un flujo FEC serán dirigidos por el mismo camino origen-destino, a menos que se especifique otra cosa.

Es posible que el tráfico siga diversos FECs para distintos requerimientos de QoS.

La **ingeniería de tráfico** es la habilidad de definir rutas dinámicamente, repartir recursos bajo demanda, y optimizar el uso de la red. Si se hace bien, se puede aumentar la capacidad útil de la red. Suele hacerse cuando hay congestión.

### Funcionamiento:



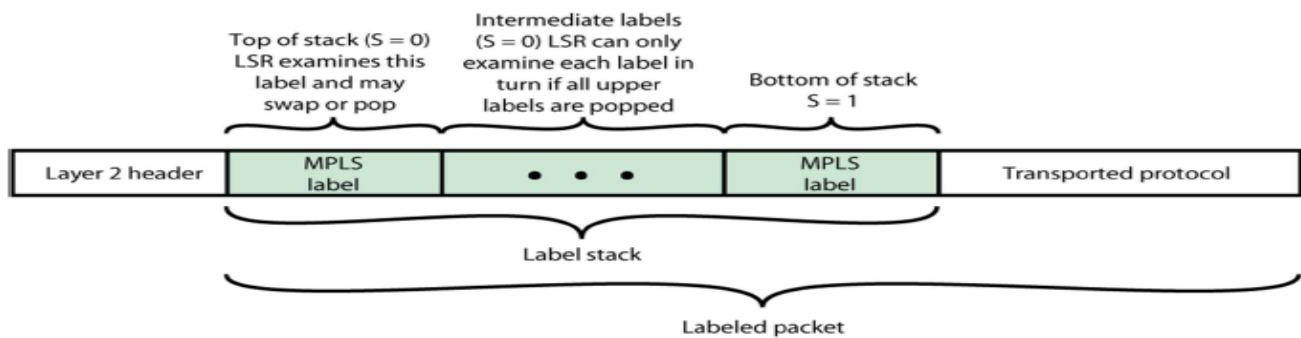
- 1- Ha de estar determinado los paths y los flujos. Cada flujo se define con unos parámetros de QoS, que indican: 1) cuantos recursos soporta el path, 2) la política de queueing y de descarte en cada LSR. Para esto se ha de intercambiar información entre routers, que para ello usan dos protocolos: a) un protocolo interno como OSPF, para intercambiar información de enrutamiento, b) etiquetas asignadas a los paquetes para un FEC particular, y que solo tienen uso de manera local (entre dos LSR). Para asignar las etiquetas usa el protocolo LDP.
- 2- El LSR que está al borde del dominio MPLS y que usan los paquetes para ingresar en este se llama **ingress Edge**. El último nodo por el que pasa el paquete y sale del dominio es el **egress Edge**. El ingress se encarga de leer la IP destino, buscar la destinación, y asignar un FEC y una etiqueta al

paquete, y luego lo pasa al siguiente nodo. Si no existe un path para el FEC, el LSR coopera con los demás para crearlo.

- 3- Este, quita la etiqueta del ingress y pone la suya, y luego lo pasa al siguiente LSR.
- 4- Finalmente, el egress quita la etiqueta, lee el header IP y lo manda a la destinación designada.

Cuando un paquete no tiene FEC asignado, el ingress lee el IP header, localiza la destinación, y asume un next-hop, le solicita una etiqueta, y esta request se propaga por la red, cuando el egress recibe la request, la responde en el orden inverso hasta el ingress, es decir, cada LSR recibe una etiqueta del router anterior (desde el egress al ingress).

El **label stacking** es una técnica que consiste en apilar etiquetas en una pila stack. Para introducir se usa la operación **push**, y para eliminar **pop**. Para cambiar la etiqueta y forwardear se usa **swap**. Al acumular etiquetas en un mismo path LSP, se puede crear túneles. De esta manera, un mismo LSR pondrá una etiqueta sobre los paquetes de distintos LSPs, llevándolos por el túnel. Al final de este, el LSR elimina la etiqueta y lee la etiqueta original que quedó en la base del stack (la de s=1).



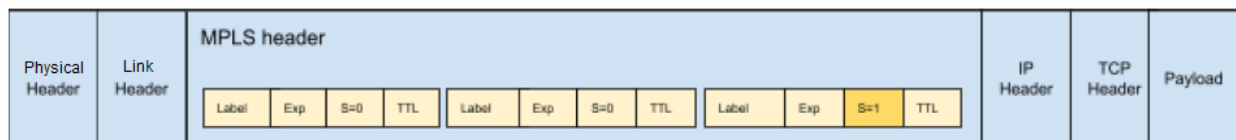
Esta técnica permite agregar paths de distintos proveedores.

Este stack se sitúa entre el header de enlace, y el header de red.

Se pone una label en el stack por cada túnel anidado, es decir, si solo hay un túnel (que puede abarcar varios LSRs hasta el final), hay dos etiquetas, los LSRs intermedios no tocan esa etiqueta, porque le hacen swap. Así, siguen haciendo sus cosas con sus etiquetas normales.

El **formato de la trama** de LDP es una etiqueta de 32bits que contiene:

- **Label value:** de 20 bits, los valores 0-15 están reservados.
- **Traffic class:** de 3 bits, define el QoS.
- **S:** el bit de stack, que indica si se hace o no. Solo lo tiene a 1 el nodo más primitivo, los demás están a 0, es decir indica el inicio del stack.
- **TTL:** de 8 bits, el tiempo de vida, para evitar bucles infinitos.



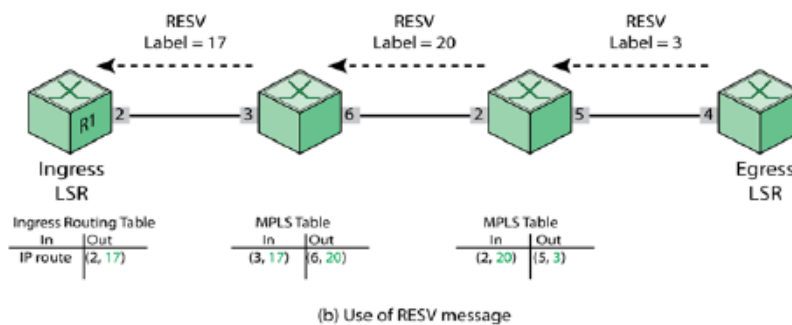
Esta trama se encuentra entre el header de enlace nivel 2, y el header de red de nivel 3.

## MPLS Traffic Engineering

Se encarga de dos cosas: i) maximizar el uso de la mayor parte de la capacidad de la red, ii) asegurarse de que los paquetes toman el camino más provechoso según los parámetros de calidad. Para implementarlo se usa:

- **OSPF:** para la distribución de información sobre la topología (ej: saber si cae algún link).
- **Cálculo de path:** el camino más corto.
- **Path setup:** reserva de recursos para el tráfico de flujo. RSVP para túneles o CR-LDP.
- **Forwarding de tráfico**

RSVP se usa para reservar recursos y para asignar túneles determinados a los flujos.



Un nodo de ingreso utiliza el mensaje RSVP “PATH” para solicitar que se defina un LSP a lo largo de una ruta explícita. El mensaje incluye un objeto de solicitud de etiqueta y un objeto de ruta explícito (ERO). La ERO define la ruta explícita a seguir por el LSP. El nodo de destino del LSP responde al LABEL\_REQUEST incluyendo un objeto LABEL en su respuesta RSVP RESV. El objeto LABEL se inserta en la LIB inmediatamente después. El mensaje RESV se envía de vuelta hacia arriba hacia el origen, siguiendo la ruta creada por el mensaje de ruta.

## VPN

Los túneles pueden tener distintos canales virtuales (CV) gracias al label stacking.

Los componentes son:

- **Customer Edge:** provee del servicio de VPN.
- **Provider Edge:** comunica a los CE con el resto de CE.
- **Provider (Core):** el resto de routers de la red, que forwarden paquetes.

El tráfico destinado a una VPN solo puede ir de una VPN origen a un destino en la misma VPN.

El L2VPN es VPN a nivel 2, y se utilizan los headers del protocolo de enlace.

## SDN

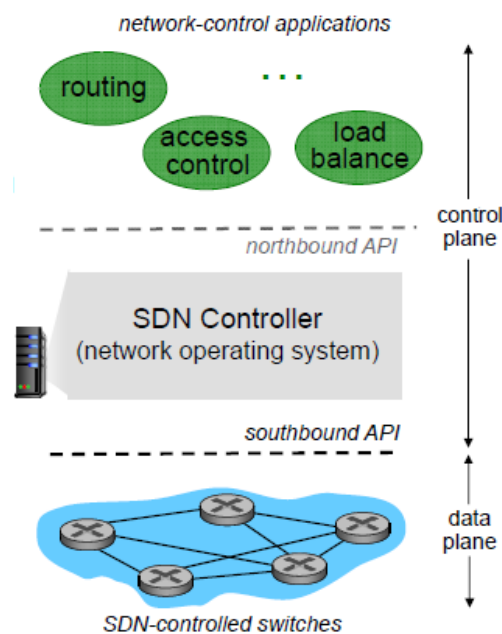
SDN se encarga de gestionar una red de manera automática en lugar de configurar los routers a mano. Se divide el plano de datos (routers y función de forward) del plano de control (controlador que envía órdenes a estos routers, y función de routing).

En el **plano de datos** se encuentran los switches, que se encargan del forward y están configurados por el controlador, el cual les instala las Flow tables.

En el **plano de control** se encuentran el SDN controller, que se comunica con la capa de arriba y abajo, gestiona los switches, controla el estado de la red, la escalabilidad, la robustez, etc. Está compuesto por:

- Capa de network control apps: se comunica con dicho nivel, contiene herramientas como el grafo de la red, restfull api...
- Capa intermedia: base de datos distribuida con información sobre los links de la red, switches, servicios... contiene Flow tables, estadísticas...
- Capa de comunicación: se comunica con el nivel de abajo, con los switches. Contiene la herramienta OpenFlow, etc.

Por último, están las aplicaciones de control, que calculan las funciones de más bajo nivel (routing, control de acceso...).

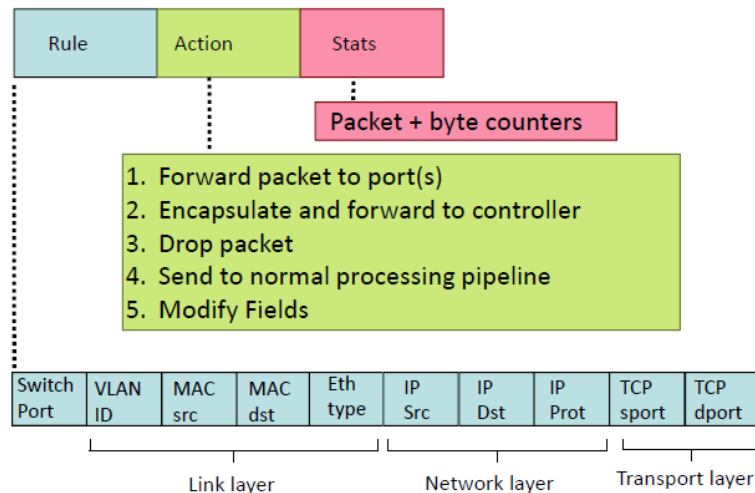


Para ello especifica 3 tipos de tablas en la arquitectura lógica. Una **Flow table** relaciona los paquetes de llegada con un flujo particular y especifica qué funciones hay que aplicar sobre estos. Una Flow table dirige el flujo a un **group table**, que lanza varias acciones sobre uno o más flujos. Una **meter table** puede lanzar una variedad de acciones sobre un flujo relacionadas con el rendimiento.

En el plano de **datos**, encontramos dos elementos: un flujo y un protocolo de forward, que consta de: un patrón que relaciona las cabeceras de paquetes, acciones que tomar sobre estos, prioridad, y contadores de bytes y de paquetes.



Las **entradas del Flow table** serían así:



- **Rule:** la información de routing o de flujo (patrón)
- **Action:** la acción a tomar sobre este flujo (acciones)
- **Stats:** (contadores)

### Ejemplos:

- Router
  - *match*: longest destination IP prefix
  - *action*: forward out a link
- Switch
  - *match*: destination MAC address
  - *action*: forward or flood
- Firewall
  - *match*: IP addresses and TCP/UDP port numbers
  - *action*: permit or deny
- NAT
  - *match*: IP address and port
  - *action*: rewrite address and port

El **match-action** es una especie de relación entre dos tipos de devices y las acciones a tomar sobre esta, por ejemplo:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

*IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6*

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

do not forward (block) all datagrams sent by host 128.119.1.1



## OpenFlow

El protocolo OpenFlow es el que se utiliza para comunicar el controlador con los switches. Se utiliza el protocolo TCP para enviar mensajes.

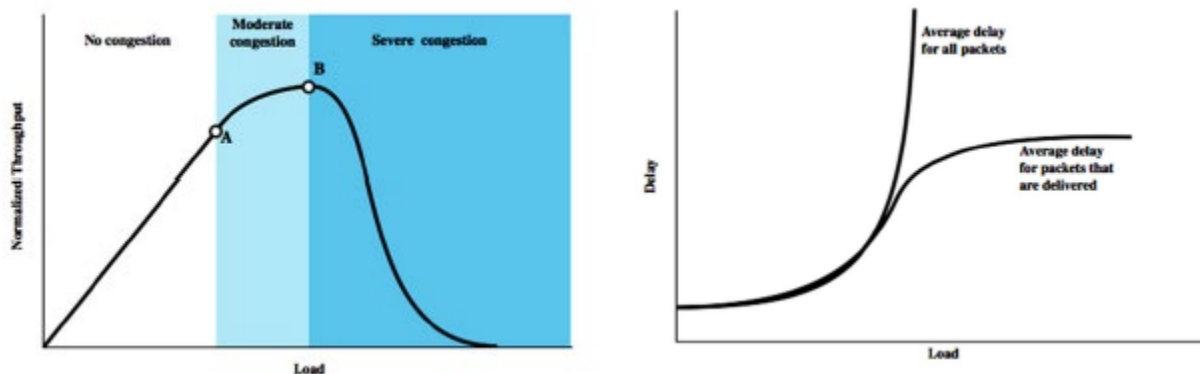
Los mensajes de OpenFlow entre el SDN y los switches son:

- **Controller – Switch:** permite al controlador gestionar los switches. A veces requiere una respuesta del switch:
  - **Features:** asignar características.
  - **Configure:** asigna parámetros de configuración del switch.
  - **Modifica estado:** añade, borra o modifica una entrada en una de las tablas.
  - **Packet-out:** el controlador envía un paquete a un puerto específico.
- **Switch – Controller:**
  - **Packet-in:** envía un paquete al controlador, y recibe su packet-out.
  - **Flow-removed:** indica que se eliminó una entrada en la Flow table.
  - **Port status:** indica el estado de un puerto.
- **Asíncrono:** El switch los usa para enviar paquetes al controlador sin que este se lo haya solicitado.
- **Simétrico:** Se envían desde el controlador o desde el switch sin que nadie lo solicite. Se utilizan para hacer Hello para asegurar las conexiones, el ancho de banda, etc.

## Control de congestión

La congestión ocurre cuando la cantidad de paquetes siendo transmitidos a través de una red comienza a acercarse a la capacidad de paquetes de dicha red. El objetivo del control de la congestión es mantener el número de paquetes dentro de la red por debajo del nivel en el que el rendimiento caería dramáticamente.

Colas en los nodos: Una red de datos o Internet es una red de colas. En cada nodo (conmutador de red de datos, router), hay una cola de paquetes para cada canal outgoing. Si la velocidad a la que llegan los paquetes y se ponen en cola supera la velocidad a la que los paquetes se transmiten, el tamaño de la cola crece sin límites y el retraso experimentado por un paquete va al infinito. Incluso si la tasa de llegada de paquetes es menos que la tasa de transmisión de paquetes, la longitud de la cola crecerá dramáticamente a medida que la tasa de llegada se aproxima a la tasa de transmisión, y experimenta un delay.



Podemos considerar que, en cada nodo, cada puerto tiene dos colas, una de incoming y otra de outgoing. A veces en lugar de esto tenemos una pool que se encarga de las funciones de queueing. En cualquier caso, los paquetes que llegan se almacenan en la cola de incoming del puerto por el que llegaron, el router lo examina, hace sus funciones de routing y lo mueve a la cola outgoing correspondiente. Si los paquetes llegan demasiado rápido como para procesarlos o más rápido de lo que se borran los de salida, no habrá memoria disponible para ellos.

Para estos casos hay dos estrategias. La primera, descartar los paquetes de incoming para los que no tenga memoria disponible. La segunda es ejercer algún tipo de control de flujo sobre sus vecinos para que el tráfico siga siendo manejable.

Además, existe el delay de propagación que se produce a través de la red de origen a destino y que se suma al delay de procesamiento del nodo.

En todo caso, aunque la capacidad de carga sea más grande que la capacidad de envío, no podrán enviarse más del máximo de la capacidad. Así, el rendimiento de la red (o envío de paquetes) aumenta hasta acomodar la carga a la capacidad máxima ofrecida por la red, luego el rendimiento queda constante en la capacidad máxima de carga. Sin embargo, el delay de procesamiento crece conforme aumenta el rendimiento.

Como podemos ver a la izquierda, a medida que la carga continúa aumentando, se alcanza un punto (punto A) más allá del cual el rendimiento de la red (envíos) aumenta a un ritmo más lento que la tasa a la que se incrementa la carga ofrecida. Esto se debe a una congestión moderada. En esta área se siguen enviando paquetes a pesar del delay (que como vemos a la derecha, incrementa). A medida que aumenta la carga la red intentará equilibrarla enrutando paquetes a áreas (puertos) menos congestionadas, pero los **nodos** (routers) no se reparten la carga, por lo que unos pueden estar muy saturados y otros no tanto. A partir del punto B, el rendimiento cae en picado debido a la severa congestión con respecto a la capacidad de carga, pues los buffers tienen capacidad limitada y

se ha visto saturada, por lo que han empezado a descartar paquetes y estos han de ser enviados de nuevo.

### **Mecanismos de control de congestión:**

- **Backpressure:** Si un nodo se llena, reduce o corta el flujo proveniente de sus nodos vecinos para recibir menos carga de estos, y hasta que consiga liberarse un poco. Si persiste, los nodos vecinos han de reducir el flujo también.
- **Choke packet:** Es un paquete que envía un nodo congestionado a un nodo vecino origen para indicarle que reduzca el flujo, hasta que deja de recibir estos mensajes, uno por cada datagrama que se descarta. Se puede anticipar y enviarlos antes de llegar a la congestión total, en este caso aún sigue recibiendo paquetes, pero con menos frecuencia.
- **Implicit Congestion Signaling:** Consiste en que todos los nodos origen detecten cuando están enviando flujo a un nodo destino congestionado, porque este aplica un delay a los paquetes o directamente los descarta. De esta manera, se anticipan a la congestión total y reducen el flujo de datos hasta que se alivie la congestión.
- **Explicit Congestion Signaling:** La red alerta a los nodos de una congestión y estos han de gestionar el flujo para evitar un colapso. Puede trabajar de dos maneras:
  - **Backward:** notifica al origen mediante bits alterados del encabezado de paquete o con paquetes aparte que aplique el procedimiento en sentido contrario a la señal en caso de congestión.
  - **Forward:** notifica al destino que aplique el procedimiento en la dirección de la señal en caso de congestión.

La señalización se puede hacer de tres maneras:

- **Binario:** se establece un bit (0 ó 1) en los paquetes para indicar que debe reducir el flujo.
- **Credit based:** se ofrece un crédito de octetos o paquetes, y si se alcanza, el origen debe esperar a otro crédito adicional.
- **Rate based:** se establece un límite de tasa de datos, si se detecta una congestión cualquier nodo de la ruta puede reducir la velocidad para controlarla.

Al aplicar los mecanismos, se tienen en cuenta: justicia (si un flujo congestiona la red, se producen descartes más a menudo en el buffer con más tráfico, y se procura que todos los buffers descarten por igual), QoS (los flujos de tráfico se tratan distinto, según el tipo de paquetes, las prioridades, de la congestión...), y reservas (al establecer el link, se establecen sus características, si el flujo se pasa, se descarta).

### **QoS**

La calidad de servicio tiene dos categorías: las que se refieren a la tecnología, y las que se refieren a la percepción del usuario. Estudia distintos aspectos:

- Delay
- Delay jitter
- Throughput
- Packet loss
- Reliability

Pero los **parámetros** que tiene en cuenta son:

- Delay: retraso del paquete en el camino de origen a destino.
- Jitter: la variación del delay.
- Loss: paquetes perdidos por el camino.

Categorías de servicio:

Tiempo real:

- CBR (constant bit rate): establece ancho de banda fijo en el momento de la conexión. No tolera delay. Ej: transmisión de video
- Rt.VBR (real time variable bit rate): especifica una tasa de rendimiento y otra tasa sostenida, pero los datos no se envían uniformemente. Este en concreto se usa para aplicaciones en tiempo real.

No tiempo real:

- Nrt.VBR (non real time variable bit rate): igual. Este se usa para tráfico de ráfagas y velocidad variable.
- UBR (unspecified bit rate): usa el ancho de banda disponible porque no asegura los niveles. Tolerar delays. Ej: email

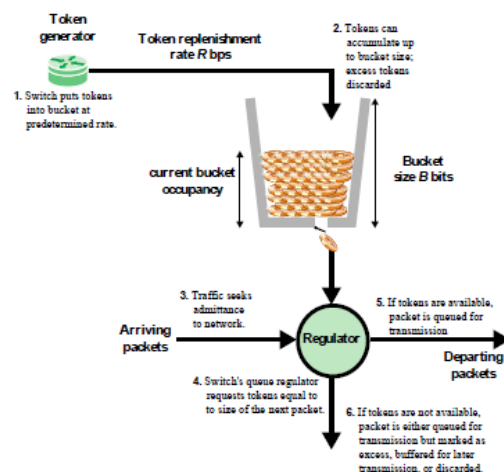
El **traffic policing** se encarga de asegurar que el tráfico cumple con los contratos prenegociados, sino, los descarta o los clasifica como no negociados. Diferencia entre paquetes que cumplen o no el QoS. Los paquetes que no cumplen el QoS les da menos prioridad, los etiqueta como no-conforme, o los descarta.

El **traffic shaping** tiene como objetivo suavizar la acumulación de paquetes. Se encarga de controlar la tasa y volumen de tráfico que ingresa y transita por la red por cada flujo. La empresa responsable de la conformidad tiene un buffer con los paquetes no negociados y los guarda hasta que se aprueban. Para reducir la congestión, suaviza el flujo de salida, si el buffer de entrada está lleno, hace que el buffer de salida tenga un flujo más regular.

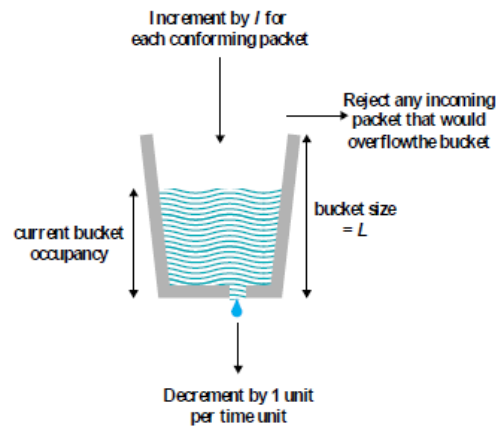
### Mecanismos

**Token Bucket:** Consiste en dos parámetros, un token rate  $R$ , que indica el data rate medio en largos períodos de tiempo, y un bucket  $B$ , el ratio que el data rate puede excederse durante cortos períodos de tiempo. En un período  $T$ , el data rate no puede excederse de  $RT+B$ .

El switch va dando un token al bucket cada cierto tiempo y este se llena de tokens  $R$ , que se van asignando a los paquetes que han de enviarse, que se transmiten si hay tokens suficientes requeridos para dicho paquete, o si no, se etiquetan como excesos y se llevan a la cola, al buffer o se descartan. Esta técnica se usa para traffic shaping.

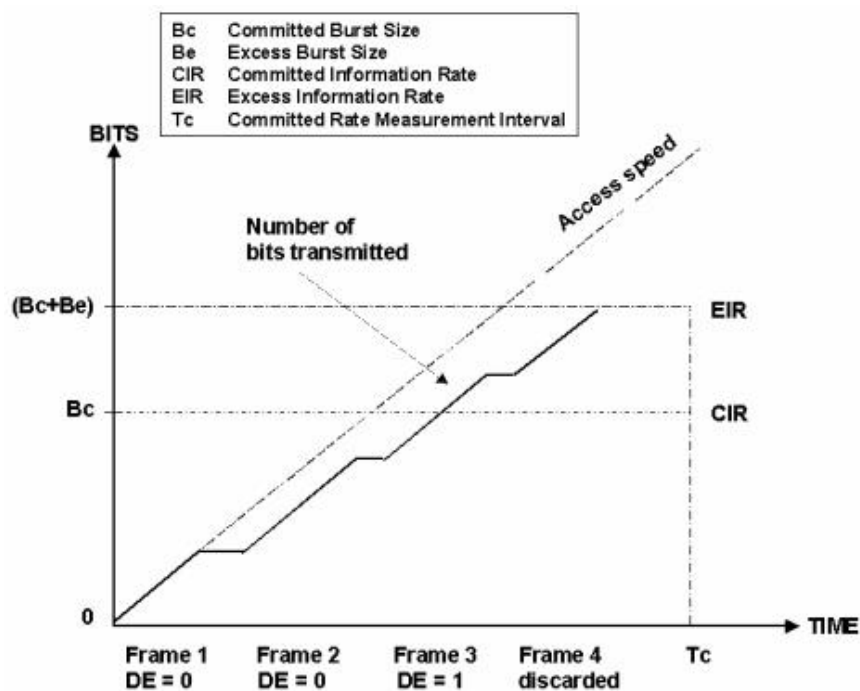


**Leaky Bucket:** Se cuenta de manera acumulativa la cantidad de datos en un contador. Por cada paquete que llega el contador sube la cantidad del tamaño del paquete, el valor máximo de contador es  $L$ . Si un paquete excede el contador, el paquete se etiqueta como no-conforme. El contador decrementa con el tiempo (de 1 en 1 cada cierto tiempo).



## Frame Relay

[https://es.wikipedia.org/wiki/Frame\\_Relay](https://es.wikipedia.org/wiki/Frame_Relay)



CIR: velocidad recomendada =  $B_c/T_c$

$T_c$ : período de tiempo

$B_c$  (committed burst)

$B_e$  (excess burst)

DE: descartables

Los paquetes que superan  $B_c$  se marcan como DE para ser los primeros descartados si se produce una congestión, si superan  $B_c+B_e$  directamente se descartan.

Cuando hay congestión se usa la técnica de control Leaky Bucket con dos cubos, el intervalo  $0+CIR$ , y el  $CIR+EIR$ .

## OTROS

**ATM** es un **protocolo** central utilizado sobre la red troncal SONET/SDH de la red telefónica pública conmutada y la red digital de servicios integrados (RDSI), pero su uso está disminuyendo a favor de la red de siguiente generación en la cual la comunicación se basa en el **Protocolo IP**.

**Frame relay** es un protocolo que define cómo se direccionan las tramas en una red de paquetes rápidos en función del campo de dirección de la trama. **Frame relay** aprovecha la fiabilidad de las redes de comunicaciones de datos para reducir al máximo la comprobación de errores que efectúan los nodos de red.

Frame Relay proporciona conexiones entre usuarios a través de una red pública, del mismo modo que lo haría una red privada punto a punto, esto quiere decir que es orientado a la conexión.

Las conexiones pueden ser del tipo permanente (PVC, *Permanent Virtual Circuit*) o conmutadas (SVC, *Switched Virtual Circuit*). Por ahora solo se utiliza la permanente. De hecho, su gran ventaja es la de reemplazar las líneas privadas por un solo enlace a la red.

El uso de conexiones implica que los nodos de la red son conmutadores, y las tramas deben llegar ordenadas al destinatario, ya que todas siguen el mismo camino a través de la red, puede manejar tanto tráfico de datos como de voz.

Al contratar un servicio Frame Relay, contratamos un ancho de banda determinado en un tiempo determinado.