Enllaç de dades

José M. Barceló Ordinas
Llorenç Cerdà Alabern

P06/19026/01892

UOC
www.uoc.edu

Índex

Introducció 5

Objectius 6

1. Descripció del nivell d'enllaç 7
1.1. Muntatge de les trames 8
1.2. Sincronisme de trama 8
1.2.1. Mecanisme de transparència 10

2. Detecció d'errors 12
2.1. Distància de Hamming 13
2.2. Capacitat de detecció d'una ratxa d'error 14
2.3. Probabilitat que una combinació arbitrària de bits sigui acceptada com a paraula vàlida 14
2.4. Control de paritat 15
2.5. Codis de paritat longitudinal i transversal 16
2.6. Codis detectors d'errors CRC 18
2.7. Codis correctors d'errors 20

3. Control d'errors 22
3.1. Diagrames de temps 23
3.2. Avaluació dels protocols: eficiència 25
3.3. Protocol *Stop & Wait* 26
3.3.1. Protocol *Stop & Wait* amb retransmissions implícites 27
3.3.2. Protocol *Stop & Wait* amb retransmissions explícites 28
3.3.3. Necessitat de numerar les trames i les confirmacions 29
3.3.4. Càlcul de l'eficiència del protocol *Stop & Wait* 30
3.4. Protocols de transmissió contínua 32
3.4.1. Retransmissió selectiva 33
3.4.2. Protocol *Go back N* 36

4. Control de flux 38
4.1. Protocols de finestra 38
4.2. Dimensionament de la llista de transmissió 39
4.3. Finestra de recepció 39
4.4. Finestra òptima 40
4.5. Dimensionament del camp dels identificadors de trama 41

5. Altres consideracions sobre el nivell d'enllaç 44
5.1. *Piggybacking* 44
5.2. Sincronisme de protocol 45

5.3. Importància del nivell d'enllaç segons el context 46
5.4. Multiplexació al nivell d'enllaç 48

6. Exemple: HDLC i ATM 50
6.1. HDLC 50
6.2. ATM 53

Resum 55

Exercicis d'autoavaluació 57

Solucionari 59

Glossari 63

Bibliografia 64

Annex 65

Introducció

Tant a les xarxes de gran abast com a les xarxes d'àrea local hem parlat de connexions punt a punt. Es tracta de l'enllaç que uneix directament entre si dues entitats sobre el qual es transmeten dades. També tenen sentit els enllaços punt a punt per si mateixos i no com a elements d'una xarxa, com ara quan es tracta de connectar un ordinador i una impressora.

La connexió punt a punt es fa amb medis de transmissió que, com ja sabem, en general no són perfectes i estan sotmesos a diferents fenòmens físics que acaben produint errors de transmissió. Dit d'una altra manera, els bits que es reben a l'altre extrem d'una connexió no són exactament els mateixos que s'han transmès des de l'altre.

El nivell d'enllaç és l'encarregat de detectar aquests errors i intentar d'alguna manera solucionar-los, corregint-los o sol·licitant la retransmissió de les trames que s'han rebut errònies. Veurem els diferents mecanismes de retransmissió i en quins casos és millor aplicar cada un.

El control d'errors comporta una certa feina de processament en transmissió i recepció. Això fa que calgui afegir un control de flux, per a poder aturar el procés en cas que les trames arribin a un terminal amb una velocitat més alta que la seva velocitat de processament.

Objectius

L'estudi dels materials didàctics d'aquest mòdul ha de permetre que l'estudiant assoleixi els objectius següents:

1. Assimilar els conceptes de trama i sincronització de trama.

2. Aprendre els principis bàsics de la detecció d'errors i conèixer-ne les tècniques més habituals.

3. Conèixer els protocols que es fan servir per a la recuperació d'errors: *Stop & Wait*, *Go Back N* i retransmissió selectiva.

4. Avaluar l'eficiència dels protocols que es fan servir per a la recuperació d'errors a un nivell bàsic, assumint que no es produeixen errors de transmissió.

5. Conèixer els mecanismes de control de flux, estudiar els conceptes de finestra de transmissió i finestra de recepció i saber calcular la finestra òptima.

1. Descripció del nivell d'enllaç

El nivell d'enllaç o DLC* és el programari que controla la transmissió d'informació sobre el flux no fiable de bits del nivell físic. En aquest apartat aprofundirem aquest concepte i la unitat de transmissió d'informació del nivell d'enllaç: la **trama**.

* DLC és la sigla de l'expressió anglesa *data link control*.

Igual que en els altres nivells de l'arquitectura de comunicacions (excepte el nivell físic), el **nivell d'enllaç** consisteix en dos programes o processos que s'executen en computadors diferents i que es comuniquen entre si. Perquè aquests dos processos es puguin comunicar és necessari establir el següent:

- Un **format** per a la informació que s'intercanvien.
- Un **conjunt de regles de comportament**.

Procés

En el context informàtic s'anomena *procés* un programa en execució.

Aquest concepte es fa servir normalment en entorns multitasca; és a dir, en aquells en què hi pot haver molts programes executant-se simultàniament en el computador.

El DLC està format pel conjunt de protocols necessaris per a la transmissió de dades sobre un enllaç de comunicació punt a punt. En aquest mòdul ens referirem a cada un d'aquests processos com a *DLC*, *entitat de nivell d'enllaç* o simplement *nivell d'enllaç*.

Una de les funcions bàsiques del nivell d'enllaç és el **control d'errors**, perquè assumim que el medi de transmissió que hi ha "per sota" no és perfecte i introdueix errors de transmissió. Per a fer el control d'errors és necessari destinar una part dels bits que s'intercanvien els DLC a la detecció i la posterior gestió dels errors. Això implica que el DLC ha de ser capaç de distingir els bits que formen part de les dades d'informació dels que formen part de les dades de control. Per a aconseguir-ho, el DLC organitza les dades en unes estructures anomenades *trames* (*frames*, en anglès).

Les **trames** són estructures ordenades de bits organitzats en camps. Segons la posició dels bits en la trama, el DLC pot reconèixer el camp a què pertanyen. Hi ha camps amb bits d'informació i camps amb bits de control.

Es pot donar el cas que el nivell d'enllaç rebi les trames més ràpidament del que és capaç de processar-les. En aquest cas s'ha de "frenar" el nivell d'enllaç remot per tal d'evitar que la memòria temporal en què hi ha emmagatzemades les trames pendents de processar s'esgoti. Aquesta funció es coneix com a ***control de flux***.

Les tasques bàsiques del nivell d'enllaç són les següents:

- Gestió de les trames.
- Control d'errors.
- Control de flux.

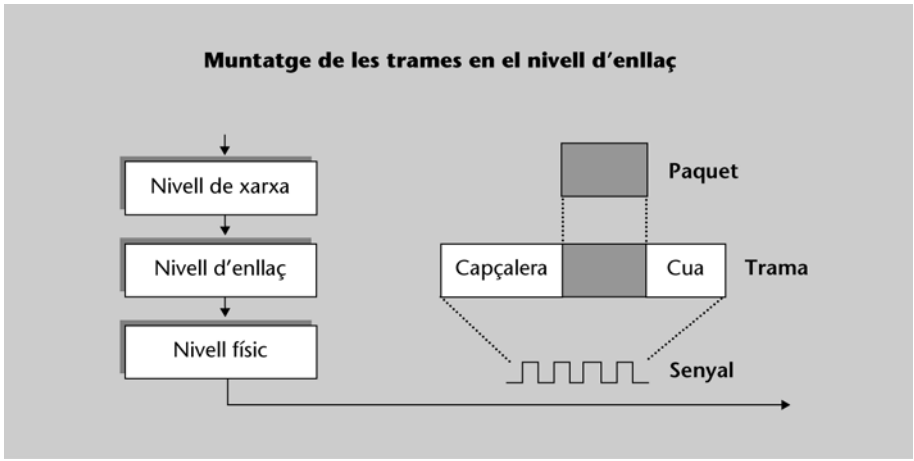
1.1. Muntatge de les trames

Les trames contenen bits de dades i bits de control. Aquesta informació de control s'estructura en camps, uns quants dels quals es troben abans de les dades, la **capçalera**, i uns altres al darrera, la **cua**.

Quins camps hi ha, de quina mida són i el seu significat depèn de les implementacions del nivell d'enllaç. En general, hi haurà un camp on hi hagi la longitud de la trama, un camp amb el tipus de dades que transporta i, com dèiem abans, la informació addicional necessària per a fer el control d'errors.

Podeu veure un exemple concret de DLC, l'HDLC, en l'apartat 6 d'aquest mòdul didàctic.

En el context de les arquitectures de protocols com OSI, el nivell d'enllaç està per sota del nivell de xarxa. Això vol dir que les dades que van inserides a les trames són paquets de xarxa.



1.2. Sincronisme de trama

Fins ara hem parlat sempre del nivell físic com un medi capaç de transportar un flux de bits. En molts casos, aquesta és una descripció correcta de la visió que té el nivell d'enllaç del nivell físic. Hi ha, però, alguns medis físics que tenen com a unitat de transmissió el **caràcter**, que es defineix com un bloc fix de bits*. Aquest cas es coneix amb el nom de **transmissió orientada a caràcter**.

* Normalment cada caràcter és de 8 bits.

De fet, es parla de **dos tipus de protocols**, un per a cadascun dels tipus de transmissió esmentats:

- Un **protocol orientat a caràcter** és un protocol de nivell d'enllaç dissenyat per anar sobre una transmissió orientada a caràcter.

- Un **protocol orientat a bit** és un protocol de nivell d'enllaç dissenyat per anar sobre una transmissió orientada a bit.

El **sincronisme de trama** (en anglès, *framing*) és el mecanisme que fa servir el nivell d'enllaç per a determinar l'inici i el final d'una trama dintre del flux de bits o caràcters que arriba del nivell físic.

Per a poder determinar l'inici i el final de trama és necessari marcar-los d'alguna manera. La solució adoptada depèn del tipus de transmissió:

1) Per a indicar l'**inici de trama**:

- Si la transmissió és orientada a caràcter, l'inici de trama s'indica amb un caràcter especial anomenat **caràcter d'inici de trama**.

- Si la transmissió és orientada a bit, s'indica amb una combinació especial de bits anomenada **indicador (*flag*) d'inici de trama**.

2) Per a indicar el **final de trama** hi ha dues tècniques:

- Utilitzar un caràcter especial anomenat **caràcter de final de trama** o una combinació especial de bits anomenada **indicador (*flag*) de final de trama**, segons si la transmissió és orientada a bit o a caràcter.

- Utilitzar un **camp de longitud** que indica la mida de la trama.

Alguns protocols fan servir totes dues tècniques conjuntament per a dur a terme el control d'errors. Així, si el caràcter de final de trama o l'indicador, segons el cas, no arriben al final de la trama indicat pel camp de longitud, es detecta un **error de delimitació de trama**, o **error de *framing***.

Exemples de sincronització de trama

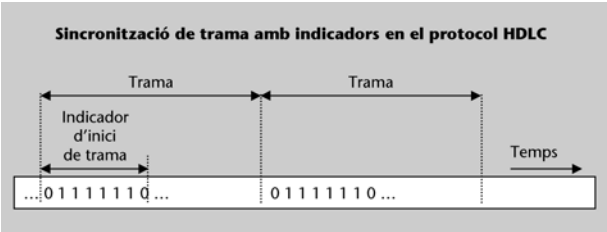
Presentem un parell de casos en què es fa servir la sincronització de trama, segons el tipus de transmissió de què es disposi:

1) Sincronització de trama en una transmissió orientada a caràcter

Tant el codi ASCII com el codi EBCDIC tenen els caràcters de control STX (de l'anglès *Start of TeXt*) i ETX (de l'anglès *End of TeXt*). Alguns dels protocols orientats a caràcter més estesos fan servir aquests caràcters en el sincronisme de trama. La figura següent mostra com seria el sincronisme de trama amb aquests dos caràcters:

Protocol BSC

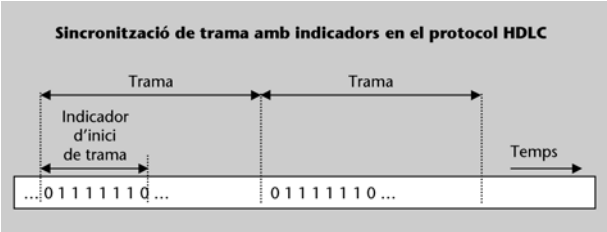
Un dels protocols orientats a caràcter més coneguts el va desenvolupar IBM. El nom d'aquest protocol és *binary sinchronous control* (*bysinc* o BSC). Aquest protocol fa servir els caràcters STX i ETX en la sincronització de les trames.



2) Sincronització de trama en una transmissió orientada a bit

Com a exemple de sincronització de trama, considerarem la figura següent, que mostra els indicadors que fa servir el protocol HDLC:

Vegeu una descripció del protocol HDLC en l'apartat 6 d'aquest mòdul didàctic.



En el protocol HDLC es defineix l'indicador de final de trama amb el mateix conjunt de bits que el d'inici de trama. El protocol permet que, si hi ha dues trames consecutives, l'indicador de final de trama sigui també el d'inici de trama de la següent, i així s'estalvia la transmissió d'aquest indicador.

1.2.1. Mecanisme de transparència

Atès que les dades d'informació que transporta la trama són arbitràries, si no es fa servir el camp de longitud es pot donar el cas que, en una transmissió orientada a caràcter, un dels caràcters de dades coincideixi amb el caràcter de final de trama. De la mateixa manera, en una transmissió orientada a bit es pot donar el cas que un conjunt de bits de dades coincideixi amb l'indicador de final de trama.

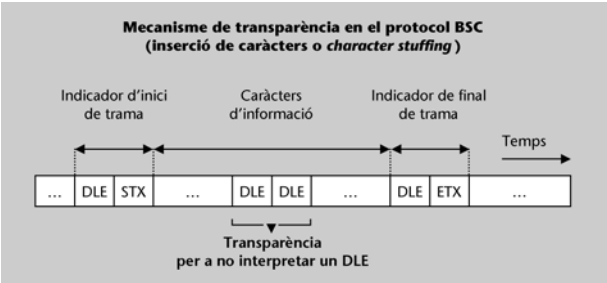
Qualsevol de les dues situacions esmentades al paràgraf anterior s'interpretaria erròniament com el final de trama. Per a evitar-ho es fa servir un **mecanisme de transparència**.

Exemples de mecanismes de transparència

Segons els dos casos que acabem de plantejar, podem trobar mecanismes de transparència com els que presentem a continuació:

1) Mecanisme de transparència en una transmissió orientada a caràcter

L'exemple de la figura següent mostra el mecanisme de transparència del protocol BSC. Per a aconseguir la transparència es fa servir un tercer caràcter, el DLE (de l'anglès *data link escape*).



Tal com mostra la figura, els caràcters STX i ETX van precedits d'un DLE. A més, cada vegada que el transmissor troba un DLE entre les dades d'informació, independentment del caràcter següent, l'envia inserint-hi prèviament un altre DLE. Aquesta tècnica per a aconseguir la transparència es coneix amb el nom d'**inserció de caràcters** (en anglès, *character stuffing*).

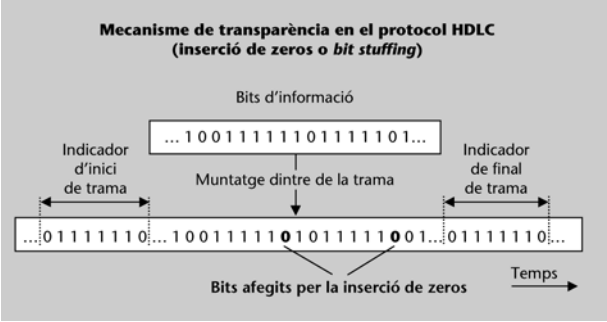
Pel que fa a la recepció, després d'haver rebut els caràcters DLE-STX, s'investiga cada un dels caràcters d'informació rebuts. Si és un caràcter DLE seguit d'un altre caràcter DLE, es descarta un dels caràcters DLE i es processa el caràcter següent (sense interpretar que anava precedit d'un caràcter DLE). Si és un caràcter DLE seguit d'un caràcter ETX, s'interpreta com a final de trama. D'aquesta manera:

a) El receptor elimina tots els caràcters de final de trama (DLE) sobrers inserits pel transmissor.

b) Si resultés que dintre de les dades d'informació hi hagués una seqüència DLE-ETX, la inserció del caràcter DLE faria que el receptor no interpretés el caràcter ETX com a final de trama.

2) Mecanisme de transparència en una transmissió orientada a bit

La figura següent mostra el mecanisme de transparència del protocol HDLC. En aquest cas l'indicador de delimitació de trama és 01111110:



La tècnica consisteix a afegir un 0 (els marcats en la figura) quan hi ha cinc 1 seguits, independentment del bit que hi hagi a continuació. La inserció de bits "extres" per a aconseguir la transparència es coneix amb el nom d'*inserció de zeros* o, en anglès, *bit stuffing*.

En recepció, si arriben cinc 1 seguits d'un 0, el 0 s'elimina i es processa el bit següent sense tenir en compte els cinc 1 anteriors. D'aquesta manera:

a) El receptor elimina tots els 0 "extres" inserits pel transmissor.

b) Si es produís la coincidència que dintre dels bits d'informació hi hagués l'indicador 01111110, la inserció del bit 0 faria que el receptor no ho interpretés com el final de trama.

2. Detecció d'errors

Suposem que el nivell d'enllaç és capaç de delimitar perfectament les trames del flux de bits que rep del nivell físic. Ara ens queda el problema de detectar quines d'aquestes trames tenen un o més bits erronis. En aquest apartat veurem alguns conceptes bàsics de la detecció d'errors.

Lectura complementària

Per a obtenir més informació sobre la detecció d'errors, consulteu l'obra següent , que trobareu referenciada a la bibliografia:

D. Bertsekas; R. Gallager. *Data Networks*.

Les trames poden tenir una combinació de bits arbitrària. Per tant, si hi ha error en un o més bits de la trama, la nova combinació és una altra possible trama. Així, doncs, només mirant els bits de la trama no és possible d'esbrinar si n'hi ha cap d'erroni. És necessari afegir-hi bits extres que ens permetin la detecció d'errors.

Els bits extres es calculen a partir dels bits que cal protegir, tal com s'explica a continuació. Fixeu-vos, però, que en afegir aquests bits estem fent servir més bits dels estrictament necessaris per a transmetre el paquet. Per això es diu que els codis detectors d'errors es basen a afegir redundància als bits de dades que es volen protegir.

Suposem que la mida de les dades que volem protegir és de P bits, i que n'hi afegim Q per poder fer la detecció d'errors.

En aquest context s'anomena **codi detector d'errors**, o simplement **codi**, la transformació que s'aplica als P bits que volem protegir per a convertir-los en la nova combinació de $P + Q$ bits. Ens referirem a aquesta nova combinació com a **paraula codi**.

Aquesta transformació és biunívoca; és a dir, a cada combinació determinada dels P bits que cal protegir li correspon una única combinació de $P + Q$ bits, i viceversa.

En aquest apartat ens referirem a les trames com a *paraules codi*, que és el terme que es fa servir en el context de la detecció d'errors.

Fixeu-vos que el **nombre de paraules codi vàlides** és igual al nombre de combinacions possibles dels bits de les dades que volem protegir (2^P). En canvi, el nombre possible de paraules codi és de 2^{P+Q}. Les $2^{P+Q} - 2^Q$ combinacions restants són **paraules codi no vàlides** i no es faran servir mai en transmissió (perquè no poden resultar mai de la transformació que aplica el codi als P bits que s'han de protegir).

Detectem que hi ha un error quan els bits erronis transformen una paraula codi vàlida en una de no vàlida. Si com a conseqüència dels bits erronis resulta una altra paraula codi vàlida, no detectarem l'error.

Hi ha nombrosos codis detectors d'errors (els més utilitzats s'explicaran tot seguit). Per a comparar-los, la mesura ideal seria determinar la probabilitat que es produís un error no detectat en la transmissió d'una trama. Desafortunadament, aquesta probabilitat depèn de les característiques del medi de transmissió i del codi detector d'errors i, en general, és molt difícil de determinar.

Per a mesurar empíricament la probabilitat d'una trama errònia no detectada hauríem de comptabilitzar totes les trames errònies no detectades i totes les trames errònies que es produeixen en l'enllaç. La relació entre aquests dos valors seria la probabilitat buscada.

No confongueu...

... la probabilitat d'error en una trama amb la probabilitat d'una trama errònia no detectada. La probabilitat d'error en una trama depèn exclusivament del medi. La probabilitat d'una trama errònia no detectada és molt més difícil de calcular perquè depèn, a més, del codi detector d'errors.
El codi detector ideal detectaria totes les trames errònies.

A causa de les dificultats que comporta determinar la probabilitat d'una trama errònia no detectada, considerarem els tres paràmetres per a mesurar la **robustesa d'un codi detector d'errors** que s'expliquen a continuació:

- La distància mínima del codi (distància de Hamming del codi).
- La capacitat de detecció de ratxes d'error (*burst detecting capability*).
- La probabilitat que una combinació arbitrària de bits sigui acceptada com a paraula vàlida.

Com veurem a continuació, per a prendre aquestes mesures no hem de tenir en compte el tipus d'errors que introdueix el medi de transmissió. És a dir, aquestes mesures donen idea de la facilitat que té un codi per a determinar certs tipus d'errors. A l'hora de triar un codi o un altre s'haurà de tenir en compte quin tipus d'error introdueix el medi de transmissió per a escollir-ne el més adient, és a dir, el que minimitzi la probabilitat de tenir una trama errònia no detectada.

2.1. Distància de Hamming

Per a definir la distància de Hamming d'un codi, primer és necessari introduir el concepte de *distància de Hamming entre dues paraules codi*.

Distància de Hamming entre dues paraules codi

Entre les paraules codi següents, hi ha 2 bits de diferència; per tant, la seva distància val 2:

- **1**00100**1**01
- **0**00100**0**01

La **distància de Hamming entre dues paraules codi** es defineix com el nombre de bits diferents que hi ha entre aquestes paraules.

La distància mínima d'un codi, o **distància de Hamming d'un codi**, es defineix com la distància més petita que hi ha entre dues paraules vàlides del codi.

D'aquesta definició es dedueix que si la distància de Hamming d'un codi val D_H, qualsevol combinació de n bits erronis que compleixi $n < D_H$ es detectarà amb probabilitat 1.

De la definició de la distància de Hamming d'un codi deduïm també que un mètode exhaustiu per a calcular-la seria considerar totes les parelles possibles de paraules vàlides, mirar quants bits diferents hi ha i agafar-ne el mínim. En la pràctica, generalment no s'aplica aquest mètode, sinó que el càlcul es fa a partir de les propietats del codi.

Com més gran és la distància de Hamming, més bits erronis hi ha d'haver perquè es produeixi un error no detectat i, per tant, el codi detector d'errors serà millor.

2.2. Capacitat de detecció d'una ratxa d'error

Moltes vegades els errors no es produeixen en bits aïllats, sinó que són originats per espurnes que afecten diversos bits consecutius. Normalment, però, una espurna no introdueix errors en tots els bits que coincideixen amb la seva duració. Segons les variacions elèctriques de la intensitat de l'espurna, alguns bits canvien, amb la qual cosa es produeix un error, i alguns altres no canvien.

En una trama es defineix la **ratxa d'error** com el nombre de bits que hi ha entre el primer bit erroni i l'últim, ambdós inclosos.

La **capacitat de detecció d'una ratxa d'error** es defineix com l'enter més gran, diguem-ne *B*, tal que el codi és capaç de detectar totes les ratxes d'error més petites o iguals que *B*.

Exemple de ratxa d'error

En la trama
101000000000000
els bits erronis són els que estan marcats. Com que entre el primer bit erroni i l'últim (ambdós inclosos) hi ha 7 bits, diem que la ratxa d'error val 7.

Evidentment, com més gran sigui la capacitat de detecció de ratxes d'error, millor serà el codi detector d'errors.

La capacitat de detecció de ratxes d'error és especialment important quan el medi de transmissió té tendència a introduir els errors en forma de ratxes. En aquest cas, com més gran sigui la capacitat de detecció de ratxes, més petita serà la probabilitat de tenir una trama errònia no detectada.

2.3. Probabilitat que una combinació arbitrària de bits sigui acceptada com a paraula vàlida

Hem vist que si el nombre de bits erronis d'una trama no excedeix la distància de Hamming ni la capacitat de detecció de ratxes, la trama errònia es detectarà amb probabilitat 1. En cas contrari, hi ha dues possibilitats:

a) La paraula codi corresponent a la trama errònia coincideix amb una altra paraula codi vàlida i, per tant, no es detecta l'error.

b) La paraula codi resultant és una paraula no vàlida i es detecta l'error.

El càlcul exacte de la probabilitat que la trama errònia no sigui detectada no és obvi. Això no obstant, podem deduir-ne de manera intuïtiva un valor aproximat fent la suposició següent: que la paraula codi corresponent a la trama errònia passa a ser, amb la mateixa probabilitat, qualsevol altra paraula codi. Això equival a suposar que es tria una combinació arbitrària de bits. Si aquesta combinació és una paraula codi vàlida, no es detectarà l'error; si no ho és, l'error es detectarà.

Atès que una paraula codi té una mida de $P + Q$ bits, cada una de les combinacions arbitràries possibles es pot rebre amb una probabilitat de $1/2^{P+Q}$. Com que hi ha 2^P paraules codi vàlides, la probabilitat que una combinació arbitrària de bits sigui acceptada com a paraula vàlida serà $2^P/2^{P+Q} = 2^Q$.

La probabilitat que una combinació arbitrària de bits sigui acceptada com a paraula vàlida és 2^{-Q}, on Q és el nombre de bits que afegeix el codi detector d'errors.

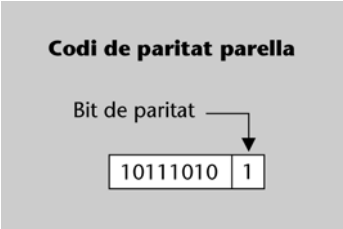
Com més gran sigui Q, més petita serà aquesta probabilitat i millor serà el codi. Això demostra que, com més bits afegeix el codi detector d'errors, més difícil és que es produeixi un error no detectat.

2.4. Control de paritat

El codi detector d'errors més senzill que hi ha és l'anomenat **control de paritat**. El control de paritat consisteix a afegir un sol bit al bloc de bits que es vol protegir, anomenat **bit de paritat**.

El valor del bit de paritat es tria de manera que la suma dels 1 de la paraula codi sigui un nombre parell (**paritat parella**). Si el criteri és que la suma sigui un nombre senar, es parla de **paritat senar**.

El gràfic següent mostra un exemple de paritat parella (la suma de tots els 1 ha de ser un nombre parell):



a	*b*	*a* XOR *b*
0	0	0
0	1	1
1	0	1
1	1	0

Càlcul de la paritat parella

La paritat parella se sol calcular fent l'operació OR exclusiva (XOR) dels bits que es volen protegir.

El codi detector d'errors més habitual

La paritat és el codi detector d'errors que més es fa servir en transmissions orientades a caràcter. Per exemple, la transmissió pel port sèrie d'un PC és orientada a caràcter. El dispositiu que controla el port sèrie (anomenat *UART*, *Universal Asynchronous Receiver Transmitter*) afegeix automàticament un bit de paritat a cada caràcter transmès. Pel que fa a la recepció, l'UART també controla automàticament que el bit de paritat sigui correcte; en cas contrari, es produeix una condició d'error.

Paritat

Quan es fa servir la paritat com a codi detector d'errors, no ens hem d'imaginar que tota la trama és protegida amb un sol bit de paritat. La paritat normalment es fa servir en transmissions orientades a caràcter, i les trames d'aquestes transmissions estan formades per més d'un caràcter, cada un amb el seu bit de paritat.

Si en la transmissió de la paraula codi es produeix un sol error (un 1 passa a valer 0 o un 0 passa a valer 1), la paritat de la paraula codi canviarà i no coincidirà amb la del bit de paritat. Per tant, es detectarà l'error. Però si hi ha dos errors, la paritat serà la mateixa i l'error no es detectarà. Deduïm, doncs, que amb el bit de paritat el codi permet detectar un nombre senar de bits erronis.

Ara deduirem quant valen els tres paràmetres que hem introduït per a mesurar la robustesa d'un codi:

a) Per a qualsevol paraula vàlida, si es canvia un bit s'obté una paraula no vàlida, i si se'n canvien dos s'obté una altra paraula vàlida. Deduïm, doncs, que la diferència mínima entre dues paraules codi vàlides és de dos bits i, per tant, que la distància de Hamming val 2. En conseqüència, el codi és capaç de detectar amb probabilitat 1 totes les combinacions de bits erronis inferiors a 2 (és a dir, el codi detecta un bit erroni, com havíem vist anteriorment).

b) Atès que el codi no detecta una ratxa d'error igual a 2 (dos bits consecutius erronis), la capacitat de detecció de ratxes val 1.

c) Finalment, la probabilitat que una combinació arbitrària de bits sigui acceptada com a paraula vàlida és: $2^{-Q} = 2^{-1} = 0,5$; és a dir, dels caràcters que tinguin molts bits erronis, només se'n detectaran la meitat.

2.5. Codis de paritat longitudinal i transversal

Una manera de millorar la robustesa del codi detector d'errors per bit de paritat consisteix a organitzar els bits que cal protegir en una matriu, i calcular la paritat de les files i de les columnes.

La paritat de les files s'anomena **paritat transversal** (o horitzontal), i la de les columnes, **paritat longitudinal** (o vertical).

Així, doncs, la informació es transmet organitzada en blocs amb la seva respectiva paritat longitudinal i transversal. La transmissió del bloc es fa per files,

de manera que els últims bits transmesos són els bits de la paritat longitudinal. La figura següent mostra un d'aquests blocs:

Codis de paritat longitudinal i transversal

Paritat transversal o horitzontal (parella)

	1	0	1	1	0	1	0
Paritat	1	0	1	1	1	0	0
longitudinal	0	1	1	0	1	0	1
o vertical	1	1	0	0	0	1	1
(parella) →	1	0	1	0	0	0	0

El codi de paritat longitudinal i transversal s'acostuma a fer servir en transmissions orientades a caràcter. El dispositiu transmissor afegeix automàticament un bit de paritat a cada caràcter, i d'aquesta manera la trama queda formada per un conjunt de caràcters als quals s'afegeix un caràcter amb la paritat longitudinal.

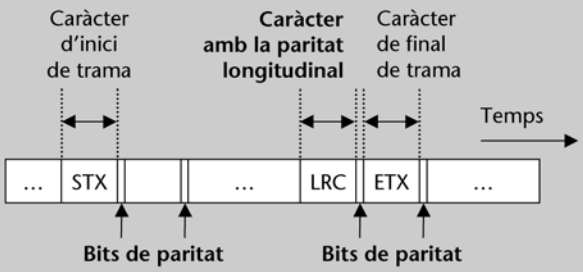
La figura següent ens mostra la disposició que acabem d'explicar dels bits de paritat dins de la trama. A causa d'aquest caràcter extra, aquest codi detector d'errors es coneix també com a **LRC*** o **BCC****.

*** LRC és la sigla del terme anglès *longitudinal redundancy check.***
**** BCC és la sigla del terme anglès *block check character.***

Càlcul del caràcter LRC

El caràcter LRC se sol calcular fent l'operació XOR (paritat parella) dels caràcters que es volen protegir.

Transmissió d'una trama amb els bits de paritat i el caràcter de paritat longitudinal LRC



Per analitzar aquest codi investigarem primer els casos en què una combinació d'errors no seria detectada. De la definició de paritat deduïm que aquest codi detectarà totes les combinacions de bits erronis que tinguin un nombre senar d'errors en alguna fila o columna. És a dir, no es detectaran les combinacions de bits erronis que tinguin un nombre parell d'errors en totes les files i columnes simultàniament. El cas més senzill és el que mostra la figura següent:

Combinació d'errors que no seria detectada

1	0	1	1	0	1	0
1	**1**	1	**0**	1	0	0
0	1	1	0	1	0	1
1	**0**	0	**1**	0	1	1
1	0	1	0	0	0	0

Deduïm el valor dels paràmetres introduïts per a mesurar la robustesa d'un codi:

a) Si en un bloc canvia un dels bits que cal protegir canviaran, a més, les paritats transversal, longitudinal i la paritat de les longitudinals: canvien 4 bits, per tant, $D_H = 4$.

Bits que canvien entre dues paraules codi vàlides consecutives

1	0	1	1	0	1	0	
1	1	1	1	1	0	1	← Paritat transversal
0	1	1	0	1	0	1	
1	1	0	0	0	1	1	
1	1	1	0	0	0	1	← Paritat de les paritats longitudinals
	↑ Paritat longitudinal						

b) Per a determinar la capacitat de detecció de ratxes hem de trobar la ratxa mínima no detectada. A partir de la figura "Combinació d'errors que no seria detectada" és fàcil deduir que la ratxa mínima no detectada es produeix quan els quatre bits erronis són adjacents, i la seva mida és igual a la longitud d'una fila més dos. Així, doncs, la capacitat de detecció de ratxes és la longitud d'una fila més u.

c) La probabilitat que una combinació arbitrària de bits sigui acceptada com a paraula vàlida és $1/2^{\text{longitud fila+longitud columna-1}}$.

2.6. Codis detectors d'errors CRC

Com hem comentat, els codis detectors amb bit de paritat estan indicats per a transmissions orientades a caràcter. Per a transmissions orientades a bit no són útils, perquè les tires de bits on es podria aplicar la paritat són molt més llargues i perdrien efectivitat. En comptes del bit de paritat, s'utilitzen els anomenats *codis CRC** o *polinomials*.

*** CRC és la sigla de l'expressió anglesa *cyclic redundancy check*.**

Els codis detectors d'errors polinomials es basen en el càlcul d'un nombre binari, que és el que realment és el CRC, resultat d'una certa operació matemàtica efectuada amb els bits que s'han de protegir. Aquest nombre es posa en el camp de control d'errors de la trama. En recepció es repeteix el càlcul i s'interpreta que hi ha o no error en funció de si coincideix o no amb el CRC rebut.

Per a explicar el càlcul del CRC és convenient representar les seqüències de bits com a polinomis, tal com s'explica a continuació. Ens referirem als bits d'una seqüència s de K bits com a $s_{K-1, sK-2, \ldots, s0}$. Definim la representació polinomial $s(x)$ de la seqüència s de la manera següent:

Exemple

La representació polinomial de la seqüència "1001001" és: $x^6 + x^3 + 1$

$$s(x) = s_{k-1}x^{k-1} + s_{k-2}x^{k-2} + \ldots + s_1x + s_0$$

L'objectiu de les potències x^j és distingir el pes del bit s_j dins de la seqüència. Suposem que la seqüència s és la formada pels bits de la trama que volem protegir. Als bits s de la trama s'afegeix una seqüència de Q bits, diguem-ne c, per a la detecció d'errors. Aquests bits es coneixen com el *CRC de la trama*. La representació polinomial de la trama, diguem-ne t, formada pels bits s i el CRC serà, doncs:

$$t(x) = s(x) \cdot x^Q + c(x)$$

El càlcul del CRC s'especifica en funció d'un polinomi generador $g(x)$, de manera que el polinomi $c(x)$ es defineix com el residu de la divisió en mòdul 2 entre els polinomis $s(x) \cdot x^Q$ i $g(x)$, és a dir: $c(x) = \text{Residu}[(s(x) \cdot x^Q) \ / \ g(x)]$. En mòdul 2, la resta és igual a la suma, que a la vegada és l'operació XOR binària.

Exemple de càlcul del CRC

Suposem que la seqüència de bits que cal protegir és "11001", amb un CRC de tres bits, i que el polinomi generador és $g(x) = x^3 + 1$. Tenim que $s(x) = x^4 + x^3 + 1$, $s(x) \cdot x^3 =$ $= x^7 + x^6 + x^3$, i la divisió en mòdul 2 de $(s(x) \cdot x^Q) / g(x)$ és:

$$\begin{array}{l|l}
x^7 + x^6 + x^3 & x^3 + 1 \\ \cline{2-2}
x^7 + x^4 & x^4 + x^3 + x \\ \cline{1-1}
\quad x^6 + x^4 + x^3 & \\
\quad x^6 + x^3 & \\ \cline{1-1}
\qquad x^4 & \\
\qquad x^4 + x & \\ \cline{1-1}
\qquad\quad x &
\end{array}$$

S'obté, doncs, $c(x) = x$, i, per tant, el CRC que caldria afegir seria "010".

Per saber si hi ha hagut errors pel que fa a la recepció, es repeteix la divisió: si el residu coincideix amb el CRC, s'entén que la trama s'ha rebut correctament.

El grau del polinomi generador $g(x)$ no és arbitrari, sinó que és determinat pel nombre de bits que es desitja que tingui el CRC. El residu de la divisió per un polinomi de grau p és un polinomi de grau menor o igual a $p - 1$.

Si volem que el CRC tingui Q bits, és a dir, que la seva representació polinomial tingui un grau menor o igual a $Q - 1$, haurem de triar un polinomi generador de grau Q. Amb altres paraules, el grau del generador ha de ser igual al nombre de bits del CRC.

Les propietats del codi CRC depenen del polinomi generador. En general, però, es pot demostrar que si triem un polinomi generador adequat de grau Q (és a dir, amb un CRC de Q bits):

a) La distància de Hamming del codi és major o igual a 4.
b) La capacitat de detecció de ratxes d'error és major o igual a Q.
c) La probabilitat que una combinació arbitrària de bits sigui acceptada com a paraula vàlida val 2^{-Q}.

Polinomis generadors estandarditzats

Hi ha polinomis generadors molt emprats que han estat estandarditzats. En són alguns, per a CRC de 16 i 32 bits, els següents:

a) CRC-16: $g(x) = x^{16} + x^{15} + x^2 + 1$
b) CRC-CCITT: $g(x) = x^{16} + x^{12} + x^5 + 1$
c) CRC-32: $g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$

2.7. Codis correctors d'errors

Hi ha codis que, a més de detectar errors, permeten corregir alguns bits erronis. Per exemple, en un codi de paritat longitudinal i transversal, qualsevol error d'un sol bit es pot corregir buscant la fila i la columna amb la paritat canviada (tal com mostra l'esquema **a)** de la figura següent). Però si l'error es produeix en dos bits, el codi ja no és capaç de corregir-lo. Això es pot veure en l'esquema **b)**. En aquesta figura els bits erronis estan marcats en negre. Si l'error es produís en els bits marcats en blanc, es tindria el mateix error longitudinal i transversal. Així, doncs, el codi de paritat longitudinal i transversal no seria capaç d'esbrinar quina de les dues possibilitats s'hauria de corregir.

La tècnica FEC

La tècnica de fer servir un codi corrector d'errors per a recuperar errors (en comptes de sol·licitar la retransmissió de la trama) es coneix en anglès com a ***forward error correction*** (FEC).

Correcció de bits amb un codi amb paritat transversal i longitudinal

a)

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

b)

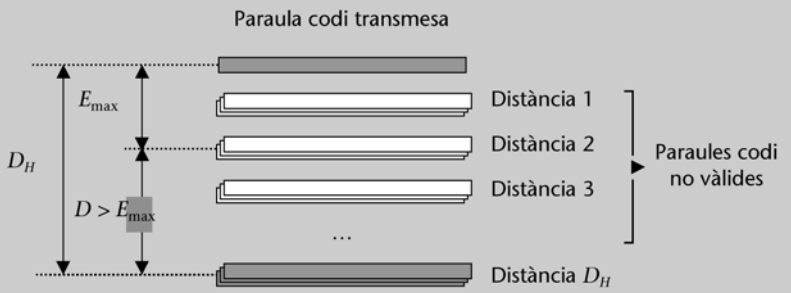
1	0	1	1	0	1	0
1	1	1	1	1	0	0
0	1	1	0	1	0	1
1	1	0	1	0	1	1
1	0	1	0	0	0	0

Correcció d'un bit

Un codi amb paritat transversal i longitudinal és capaç de corregir l'error en un bit (*a*), però no en dos (*b*).

2Amb la finalitat de fer una anàlisi més formal dels codis correctors es fa servir el concepte de *distància de Hamming*, que ja hem introduït. En cas d'error, la correcció consisteix a suposar que la paraula codi transmesa és la paraula codi vàlida més propera a la paraula rebuda, segons el concepte de *distància* (criteri de la distància mínima). Per tant, serà la que tingui menys bits de diferència.

Codi corrector segons el criteri de la distància mínima



La figura anterior és una representació gràfica de la idea que tot just acabem d'exposar. En aquesta figura podem veure una possible paraula codi transmesa, i després hi trobem agrupades totes les paraules amb el nombre de bits de diferència (és a dir, que disten): 1, 2... fins a les paraules vàlides més properes a D_H bits de distància. Si es rep una de les trames que es troba a una distància 1, 2, ..., $E_{\max}$, és a dir, una de les paraules que no tenen una altra paraula vàlida més propera que la transmesa, el codi corregirà l'error.

Per saber quants bits és capaç de corregir el codi amb probabilitat 1, suposem que, en la figura, D_H és la distància mínima entre dues paraules vàlides (és a dir, la distància de Hamming del codi).

Sigui $E_{\max}$ el nombre de bits erronis; de la figura deduïm que el criteri de distància mínima corregirà l'error si $E_{\max} < D$, on $D = D_H - E_{\max}$.

En definitiva, si la distància de Hamming d'un codi és D_H, fent servir el **criteri de la distància mínima** es pot corregir qualsevol combinació de $E_{\max}$ bits erronis que compleixi $E_{\max} < D_H/2$.

Codi amb paritat transversal i longitudinal

Si es fa servir un codi amb paritat transversal i longitudinal, la distància de Hamming val $D_H = 4$. Per tant, es pot corregir $E_{\max} < 2$, és a dir, 1 bit, tal com havíem deduït a l'inici d'aquest subapartat. Un codi amb $D_H = 5$ pot corregir $E_{\max} < 2{,}5$, és a dir, 2 bits.

3. Control d'errors

En cas que es rebi una trama amb errors, el nivell d'enllaç pot adoptar una de les solucions següents:

1) Descartar la trama.
2) Intentar corregir els errors (si es fa servir un codi corrector adient).
3) Sol·licitar la retransmissió.

La solució que s'adopti depèn de factors virtuals a l'ús de l'enllaç o al tipus d'aplicació. Més endavant parlarem d'aquests factors. Aquí només apuntarem que per a aplicar la tercera solució és necessari que la comunicació sigui bidireccional (*half-duplex* o *full-duplex*).

Vegeu els motius que condueixen a una solució determinada en els subapartats 5.2 i 5.3 d'aquest mòdul didàctic.

La primera solució només és factible en aplicacions que tolerin un cert grau d'error en la informació rebuda. L'avantatge de la segona solució respecte de la tercera és que s'evita el retard que implica sol·licitar la retransmissió de les trames. Per contra, els codis correctors d'errors necessiten afegir molta redundància (bits extres) i, per tant, la transmissió és més ineficient (la relació entre els bits d'informació i control disminueix). Més endavant aprofundirem el concepte d'eficiència.

Vegeu el concepte d'*eficiència* en el subapartat 3.3.4 d'aquest mòdul didàctic.

Un cas típic on és preferible un codi corrector d'errors abans que un codi detector acompanyat de retransmissions és en comunicacions via satèl·lit. Surt més a compte pagar en ineficiència per l'increment de bits redundants, que en temps de retransmissions, perquè la distància a recórrer és molt gran.

Els motius anteriors expliquen perquè, quan es necessita fer una recuperació d'errors, s'adopta la tercera solució sempre que es pot. Els protocols que fan servir aquesta tècnica es coneixen com a **protocols ARQ**, de l'anglès ***a**utomatic **r**epeat re**q**uest*. En concret, els protocols que veurem són els següents:

- *Stop & Wait*
- Retransmissió selectiva
- *Go back N*

Per estudiar aquests protocols suposarem:

1) Un **nivell físic** genèric en el qual:

- Les trames poden arribar amb error o es poden perdre. Direm que la trama es perd quan els errors es produeixen de tal manera que impedeixen que el receptor en pugui delimitar i interpretar la recepció.

- Les trames arriben en el mateix ordre en què s'han transmès.

2) Un **nivell per sobre del nivell d'enllaç** en el qual:

- El transmissor sempre té dades d'informació a punt per a transmetre. El nivell d'enllaç, però, té la llibertat d'acceptar-les en instants arbitraris.

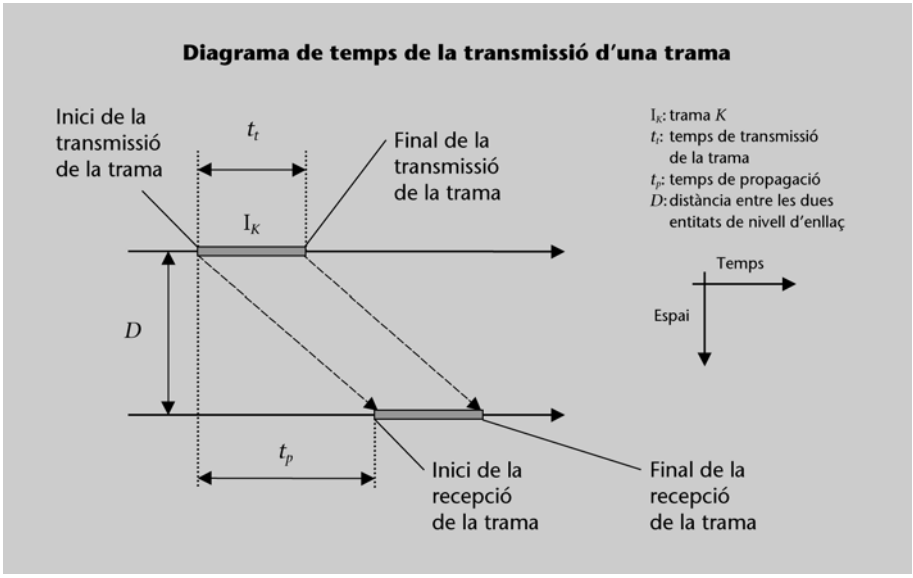
- El receptor sempre està disposat a acceptar dades del nivell d'enllaç.

A més de descriure els protocols, els avaluarem. L'**avaluació dels protocols** consisteix en el càlcul dels paràmetres de més interès, com ara l'eficiència. Per a la descripció i avaluació d'aquests protocols és molt útil l'ús d'un diagrama de temps. Així, doncs, abans de començar amb la descripció dels protocols, explicarem aquesta eina.

3.1. Diagrames de temps

Els diagrames de temps que farem servir en aquesta assignatura són una representació espaciotemporal de la transmissió de trames entre les dues entitats de nivell d'enllaç. Aquest tipus de diagrames també es fa servir per a l'estudi de la comunicació entre entitats en molts altres contextos.

Com a exemple d'aquest tipus de diagrames, la figura següent mostra la transmissió d'una trama anomenada I_K. La interpretació de l'índex K és la següent: si enumerem la seqüència de trames transmeses amb els números {1, 2, 3, ...}, I_K representa la trama K de la seqüència transmesa.



En la figura anterior hi ha dos eixos temporals; sobre cada un es representa els esdeveniments que tenen lloc en cada una de les estacions: en l'eix superior, els que es duen a terme en l'estació que transmet la trama, i en l'inferior, els que es duen a terme en l'estació que la rep.

En els diagrames de temps que representarem farem aquestes suposicions:

- Assumirem que el **temps de processament** és zero. És a dir, que quan una entitat de nivell d'enllaç té una trama preparada per a transmetre, el nivell físic la comença a transmetre immediatament.

- Suposarem que la **velocitat de transmissió** de la línia és fixa i l'anomenarem v_t (mesurat en bits per segon o bps).

- Suposarem (si no s'indica el contrari) que la **mida de les trames**, L, és fixa i es mesura en bits.

- Suposarem que la **distància entre les estacions**, D, i la **velocitat de propagació en el medi**, v_p, són fixes.

Velocitat de propagació

En el buit, un senyal electromagnètic es propaga a una velocitat de $v_p \approx 3 \cdot 10^8$ m/s. En qualsevol altre medi la velocitat de propagació és inferior.

En funció de v_t i L, el **temps de transmissió d'una trama** (t_t) ve donat per:

Equació 1

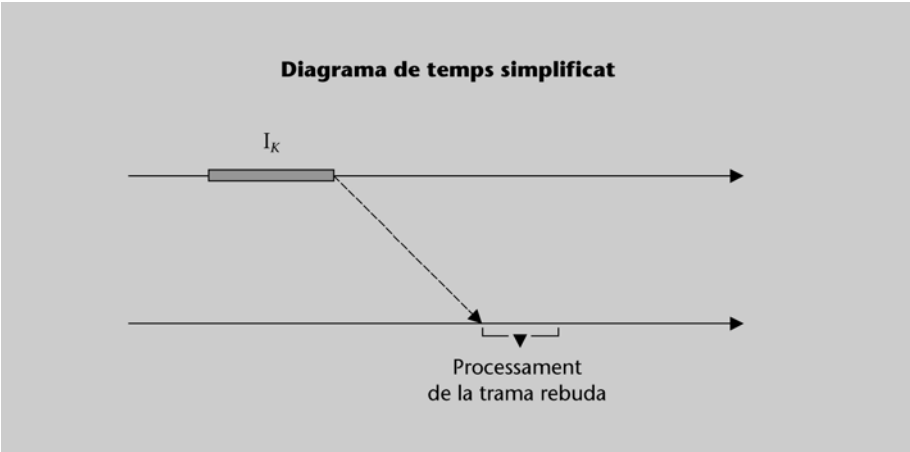
$$t_t(\mathrm{s}) = \frac{L(\mathrm{bit})}{v_t(\mathrm{bps})}$$

En funció de D i v_p, el **temps de propagació** (t_p) es pot obtenir a partir de:

Equació 2

$$t_p(\mathrm{s}) = \frac{D(\mathrm{m})}{v_p(\mathrm{m/s})}$$

D'ara endavant farem servir una versió simplificada del diagrama de temps de la figura anterior, en què indicarem només el temps de transmissió de la trama transmesa, el temps de propagació i l'instant d'arribada de l'últim bit de la trama, tal com es mostra en la figura següent:



Fixeu-vos que el receptor només pot començar a processar la trama rebuda (mirar si hi ha errors) després d'haver rebut l'últim bit.

3.2. Avaluació dels protocols: eficiència

El principal criteri d'avaluació dels protocols ARQ és l'**eficiència**.

> L'eficiència quantifica la pèrdua de capacitat de transmissió d'informació. És a dir, és la relació entre la velocitat mitjana a què es transmeten els bits d'informació i la velocitat màxima a què es podrien transmetre.

La **velocitat de transmissió màxima** (v_t) és la velocitat de transmissió de la línia; és a dir, la velocitat a què el nivell físic és capaç de transmetre els bits:

Equació 3

$$v_t = \frac{1}{\text{temps de transmissió d'un bit}}$$

La velocitat que ens interessa, però, és la velocitat mitjana a què es pot transmetre la informació tenint en compte tots els factors que redueixin la velocitat de transmissió màxima. Aquesta velocitat l'anomenarem **velocitat de transmissió eficaç** o **velocitat de transmissió efectiva** (v_{ef}):

Equació 4

$$v_{ef} = \frac{\text{bits d'informació}}{\text{temps de transmissió}}$$

L'eficiència es defineix com la relació entre la velocitat efectiva i la velocitat de transmissió; és a dir:

Equació 5

$$E = \frac{v_{ef}}{v_t} = \frac{\text{bits d'informació}}{\text{temps de transmissió}} \Big/ \frac{1}{\text{temps de transmissió d'un bit}}$$

Atès que:

bits d'informació × temps de transmissió d'un bit =
= temps en què s'han transmès els bits d'informació,

l'equació 5 per a l'eficiència es pot expressar també com es mostra a continuació:

Equació 6

$$E = \frac{\text{temps transmetent informació}}{\text{temps de transmissió}}$$

De vegades és interessant tenir l'eficiència en funció dels bits transmesos, en comptes del temps invertit en la seva transmissió. Tenint en compte la igualtat següent:

$$\frac{\text{temps de transmissió}}{\text{temps de transmissió d'un bit}} = \begin{array}{c}\text{Nombre de bits que es podrien haver}\\ \text{transmès en el temps de transmissió.}\end{array}$$

l'equació 5 també es pot expressar de la manera següent:

Equació 7

$$E = \frac{\text{bits d'informació}}{\text{bits que caben en el temps de transmissió}}$$

3.3. Protocol *Stop & Wait*

El protocol *Stop & Wait*, que podem traduir per 'para i espera', és el protocol ARQ més senzill.

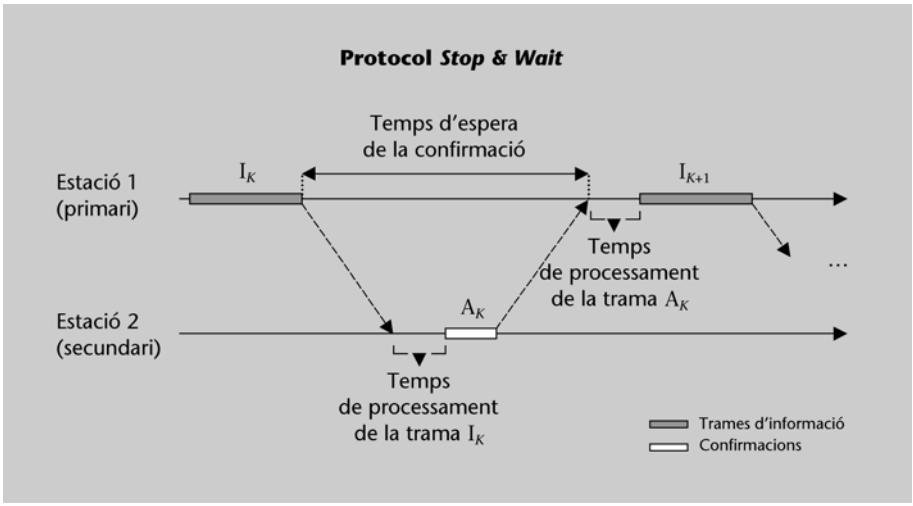
El principi de funcionament del protocol *Stop & Wait* és "no transmetre una trama nova fins que no es té la certesa de la recepció correcta de la trama anterior".

Per a aconseguir-ho es defineixen dos tipus de trames:

a) Les **trames d'informació** o **trames**, que porten la informació que s'intercanvien les entitats del nivell superior.

b) Les **trames de confirmació** o **confirmacions**, que formen part del protocol del nivell d'enllaç i que no porten informació dels nivells superiors.

La figura següent mostra el funcionament d'aquest protocol. En aquesta figura l'estació 1 envia trames d'informació a l'estació 2, i l'estació 2 les confirma. Per distingir l'estació transmissora de trames d'informació de l'estació transmissora de les confirmacions, parlarem de **primari** i **secundari**, respectivament:



Notació

En els diagrames de temps indicarem les trames d'informació amb una *I* i les confirmacions amb una *A* (del mot anglès *acknowledgement*).

El diagrama de la figura s'ha d'interpretar de la manera següent: quan el secundari rep completament la trama I_K, comprova si l'ha rebuda correctament i, en cas afirmatiu, envia la confirmació A_K.

Fixeu-vos que tant el secundari com el primari inverteixen un cert temps en el processament de les trames rebudes. Això implica que, tot i que el temps de propagació és constant, a causa del temps de processament de les trames en el secundari, el **temps d'espera de les confirmacions** és variable.

Per simplificar-ho, suposarem generalment que el temps de processament de les trames és zero, excepte quan sigui necessari tenir-lo en compte.

La **mida de les trames de confirmació** és normalment molt més petita que la de les trames d'informació, atès que no porten dades del nivell superior. A més, s'ha de tenir en compte que el codi detector d'errors també s'ha d'aplicar a les confirmacions per tal d'assegurar-ne la recepció correcta.

En aquest subapartat suposarem que el codi detector d'errors és prou robust per a poder detectar qualsevol error que hi pugui haver en la transmissió d'una trama.

En cas que hi hagi error, el secundari sol·licita la retransmissió de la trama d'informació. Això es pot fer mitjançant les retransmissions implícites i les explícites.

3.3.1. Protocol *Stop* & *Wait* amb retransmissions implícites

En aquest cas, les regles que segueixen el primari i el secundari són les següents:

1) El secundari envia confirmacions positives, A, de les trames que rep sense errors.

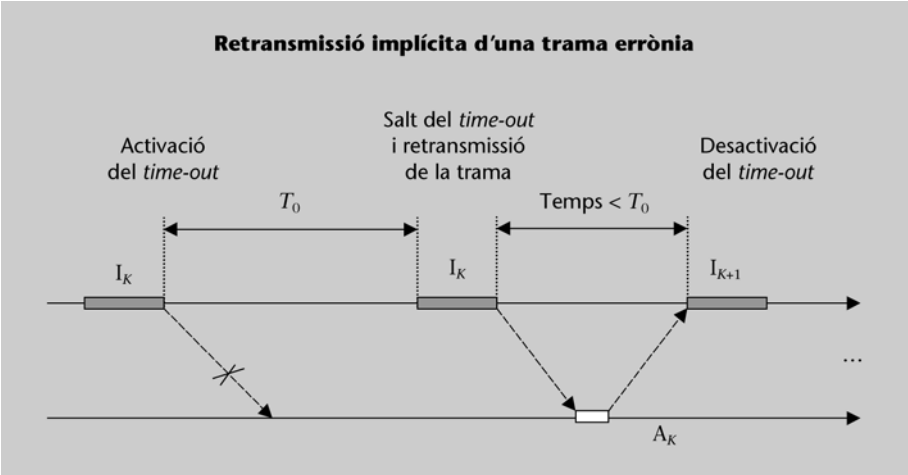
2) El primari, després d'enviar una trama I_K, espera un temps T_0 (activa un temporitzador o *time-out*) per a rebre la confirmació positiva, A_K.

a) Si un cop esgotat el temps T_0 la confirmació no ha arribat, retransmet la trama I_K i torna a activar el *time-out*.

b) Si rep la confirmació positiva, A_K, desactiva el *time-out*, accepta noves dades del nivell superior, munta una nova trama I_{K+1} i repeteix aquest procés.

La figura següent mostra la recepció d'una trama errònia i la seva retransmissió implícita. Evidentment, el valor T_0 del *time-out* s'ha de fixar de manera que el primari tingui temps de rebre la confirmació (és a dir, $T_0 > t_a + 2t_p$, on t_a és el temps

de transmissió de les confirmacions i t_p és el temps de propagació). Si el valor del *time-out* fos massa petit, es produiria la retransmissió innecessària de les trames:

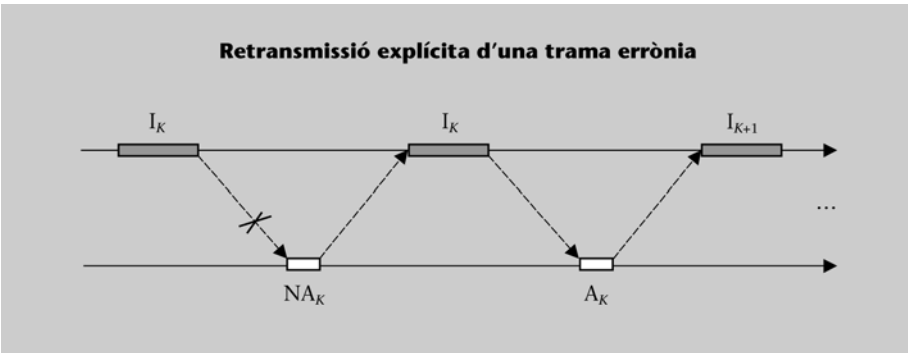


Notació

En els diagrames de temps indicarem la recepció d'una trama errònia amb una creu.

3.3.2. Protocol *Stop* & *Wait* amb retransmissions explícites

En aquest cas, les regles són les mateixes que en la retransmissió implícita, amb la diferència que el secundari també envia confirmacions negatives (NA_K) si rep una trama amb errors:

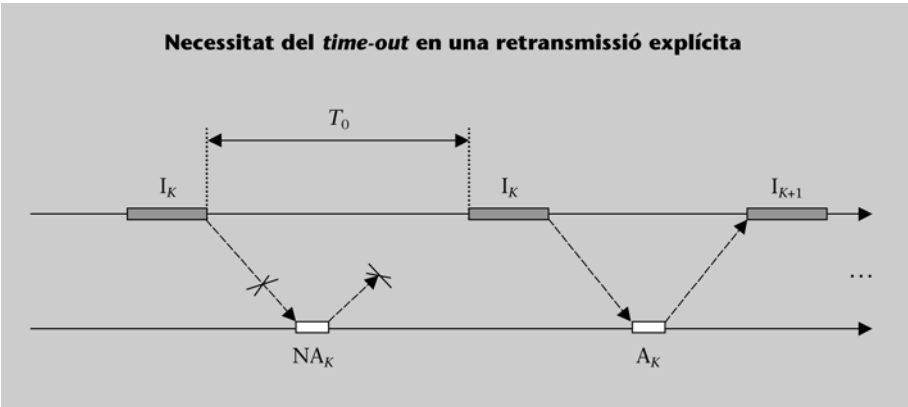


Notació

En els diagrames de temps indicarem les confirmacions negatives amb la sigla NA (*no acknowledgement*).

En general, la confirmació negativa permet una retransmissió més ràpida de la trama errònia, perquè no cal esperar que s'esgoti el *time-out*. Ara bé, en la transmissió també és necessari emprar un *time-out*, ja que les confirmacions poden arribar amb errors o es poden perdre*. Aquesta condició es reflecteix en la figura següent:

*** Les confirmacions es poden perdre en cas que els errors no permetin reconèixer la recepció de la trama.**



Notació

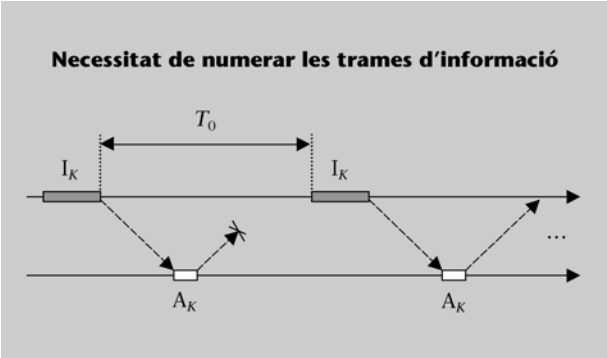
Indicarem que una trama es perd (és a dir, que els errors impedeixen reconèixer-ne la recepció) amb una fletxa que no arriba a l'eix contrari i una creu.

3.3.3. Necessitat de numerar les trames i les confirmacions

Sabem que les trames porten un camp de control que serveix per a detectar possibles errors. En aquest subapartat veurem que, per al funcionament correcte del protocol, és necessari numerar les trames d'informació i les confirmacions (positives o negatives). Així, doncs, un altre camp de control de les trames és un **número de seqüència** que permet identificar-les.

Necessitat de numerar les trames d'informació

La figura següent evidencia la necessitat de numerar les trames d'informació:



En aquest cas, es perd la confirmació d'una trama. Atès que el primari no rep la confirmació, salta el *time-out* i retransmet la mateixa trama I_K. Si la trama no estigués numerada, el secundari no podria detectar la recepció duplicada de la trama. Gràcies al número de seqüència el secundari detecta aquesta condició, descarta la trama duplicada i torna a enviar la confirmació perquè el primari continuï la transmissió de la trama d'informació següent (la trama I_{K+1}).

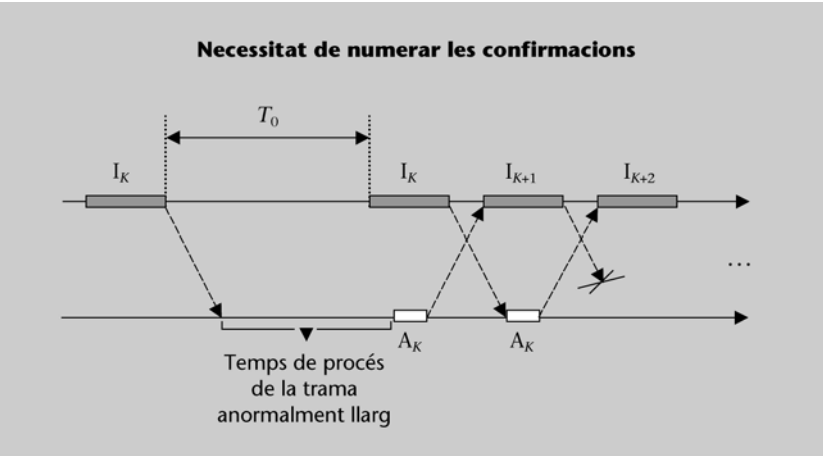
Fins ara, en els diagrames de temps hem considerat que el temps de processament de les trames era igual a zero. Ara tindrem en compte que, en una situació real, aquest temps pot ser variable. Fins i tot, pot tenir valors anormalment grans. Aquest fet pot respondre a múltiples causes. Per exemple, si el programari que executa el DLC és un procés en un computador multitasca, en el cas que el computador estigués molt carregat, el temps que dedicaria al DLC podria ser molt variable.

Exemple

Si el programari que executa el DLC resideix en un computador multitasca, pot passar que el computador estigui molt carregat i el temps que dediqui al DLC sigui molt variable.

Necessitat de numerar les confirmacions

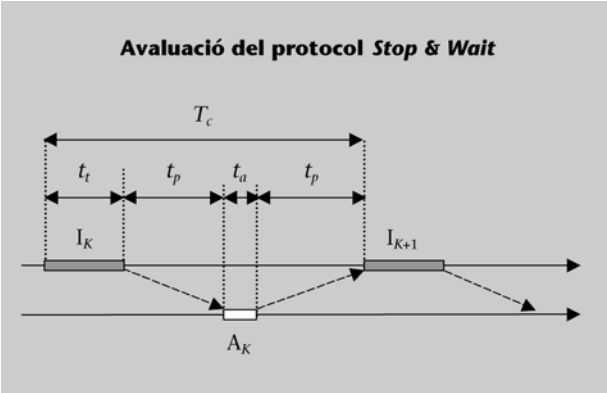
La figura següent és un exemple que evidencia la necessitat de numerar les confirmacions:



En el cas que presentem, la confirmació A_K arriba al cap d'un temps anormalment llarg (a causa d'un temps de processament excessiu en el secundari), fet que provoca que abans que arribi la confirmació salti el *time-out* i la trama I_K es retransmeti. Per aquest motiu, arriben dues confirmacions de la trama I_K. Si no es numeressin, la segona confirmació s'interpretaria com la confirmació de la trama I_{K+1}. En cas que la trama I_{K+1} es perdés (com indica la figura), el primari no la retransmetria.

3.3.4. Càlcul de l'eficiència del protocol *Stop & Wait*

En el cas del protocol *Stop & Wait,* la pèrdua de capacitat de transmissió es deu al temps que es perd en l'espera de confirmacions i retransmissions de trames. Hi ha també altres factors que afecten l'eficiència, com ara els camps de control en les trames d'informació. A l'hora de calcular l'eficiència d'un protocol, sempre hem de tenir en compte totes les dades que tinguem a l'abast i que resultin d'una reducció de la capacitat de transmissió. Per tal de calcular l'eficiència considerarem el diagrama de temps següent:



Recordem que t_p és el temps de propagació. Suposarem que totes les trames tenen la mateixa mida, igual a L bits (si la mida fos variable, agafaríem com a L el valor mitjà). El temps necessari per a la transmissió de la trama el dóna l'equació 1. De la mateixa manera, si les confirmacions són de M bits, el **temps de transmissió d'una confirmació** (t_a) és:

Vegeu l'equació 1 en el subapartat 3.1 d'aquest mòdul didàctic.

Equació 8

$$t_a(\mathrm{s}) = \frac{M(\mathrm{bit})}{v_t(\mathrm{bps})}$$

Segons això, el cicle indicat com a T_c en la figura es repetirà en la transmissió de cada trama. Aplicant l'equació 6, l'eficiència del protocol vindrà donada per la relació entre el temps de transmissió de la trama (t_t) i el temps invertit per a la transmissió de la trama (T_c). De la figura anterior deduïm que T_c és $T_c = t_t + t_p + t_a + t_p$; així:

Equació 9

$$E = \frac{t_t}{T_c} = \frac{t_t}{t_t + t_a + 2t_p}$$

Sovint la mida de les trames d'informació és molt més gran que la de les confirmacions. Aleshores $t_t >> t_a$ i, per tant, l'equació 9 es pot reescriure de la manera següent:

$a = t_p/t_t$

Equació 10

$$E \approx \frac{t_t}{t_t + 2t_p} = \frac{1}{1 + 2t_p/t_t} = \frac{1}{1 + 2a}$$

En funció de l'eficiència, podem calcular la velocitat efectiva (v_{ef}):

Equació 11

$$v_{ef} = v_t \cdot E \approx \frac{v_t}{1 + 2a}$$

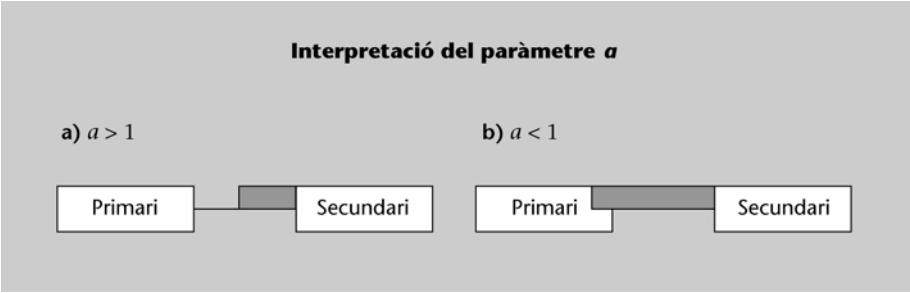
El paràmetre a, segons la mida de la trama, té la interpretació física següent:

$$a = \frac{t_p}{t_t} = \frac{t_p}{L/v_t} = \frac{t_p \cdot v_t}{L} = \frac{\text{nombre de bits que caben en l'enllaç}}{\text{mida de la trama}}$$

És a dir, podem interpretar ***a*** com el **nombre de trames que caben en l'enllaç**. Per tant, es dóna un d'aquests casos:

- Si $a > 1$, quan el primer bit de la trama arriba al secundari, el primari ja ha acabat de transmetre la trama.

- Si $a < 1$, quan el primer bit de la trama arriba al secundari, el primari encara no ha acabat de transmetre la trama.



Cal tenir en compte les observacions següents:

a) L'eficiència normalment es dóna en tant per cent, és a dir, multiplicant el resultat de l'equació 9 per 100.

b) En cas que hi hagi errors, l'eficiència és inferior perquè cal considerar les retransmissions. Per tal d'evitar confusions parlarem de l'**eficiència màxima** per a referir-nos a l'eficiència que podem aconseguir en el millor dels casos, és a dir, sense errors.

c) Tal com hem comentat abans, s'han de tenir en compte totes les dades que afectin l'eficiència i que tinguem a l'abast, per exemple els camps de control de les trames d'informació. Suposem que sabem que dels L bits de les trames d'informació n'hi ha L_d d'informació i Lc de control. Aplicant l'equació 7, l'eficiència deguda als bits de control de la trama (diguem-ne E_t) s'obtindria a partir de l'equació següent:

Notació

En el càlcul de l'eficiència farem servir les notacions següents:

- Si hem de considerar l'eficiència deguda a diversos factors, ens referirem a cada un d'ells amb una E amb un subíndex, i amb E_{tot} al producte de tots.
- Si només considerem una eficiència, ens hi referirem amb la lletra E sense cap subíndex.

Equació 12

$$E_t = \frac{L_d}{L}$$

Evidentment, l'**eficiència total** (E_{tot}) és el producte de les eficiències parcials:

Equació 13

$$E_{tot} = E_p \cdot E_t$$

E_P és l'eficiència del protocol, donada per les equacions 9 i 10.

Activitat

Calculeu l'eficiència màxima que es pot aconseguir en un enllaç punt a punt que fa servir un protocol *Stop & Wait* en els casos següents:

a) Enllaç entre dues estacions d'una xarxa de dimensions reduïdes amb les dades següents:
- Distància entre les estacions = 1 km
- Velocitat de transmissió de l'enllaç = 10 Mbps
- Velocitat de propagació en l'enllaç = $2 \cdot 10^8$ m/s
- Mida mitjana de les trames d'informació = 5 kbit
- Mida de les confirmacions negligible

b) Enllaç via satèl·lit amb les dades següents:
- Temps de propagació entre estacions = 270 ms
- Velocitat de transmissió de l'enllaç = 56 kbps
- Mida mitjana de les trames d'informació = 4 kbit
- Mida de les confirmacions negligible

Solució

a) $t_p = D/v_p = 10^3 / 2 \cdot 10^8 = 0{,}5 \cdot 10^{-5}$ s
$t_t = L/v_t = 5 \cdot 10^3 / 10 \cdot 10^6 = 5 \cdot 10^{-4}$ s
$E(\%) = t_t / (t_t + 2t_p) \cdot 100 = 98\%$

b) $t_t = 4 \cdot 10^3 / 56 \cdot 10^3 = 71{,}5$ ms, d'on $E = 12\%$.

Un cop hàgiu fet aquesta activitat, veureu que per a aconseguir eficiències elevades amb el protocol *Stop & Wait*, s'ha de complir que $a << 1$, és a dir, que el temps de propagació sigui petit en comparació amb el temps de transmissió d'una trama d'informació.

3.4. Protocols de transmissió contínua

Acabem de veure que el protocol *Stop & Wait* pot arribar a ser molt ineficient, a causa que el primari no pot transmetre durant el temps d'espera de les confirmacions.

Als protocols de transmissió contínua es deixa que el primari transmeti contínuament trames d'informació mentre s'esperen les confirmacions.

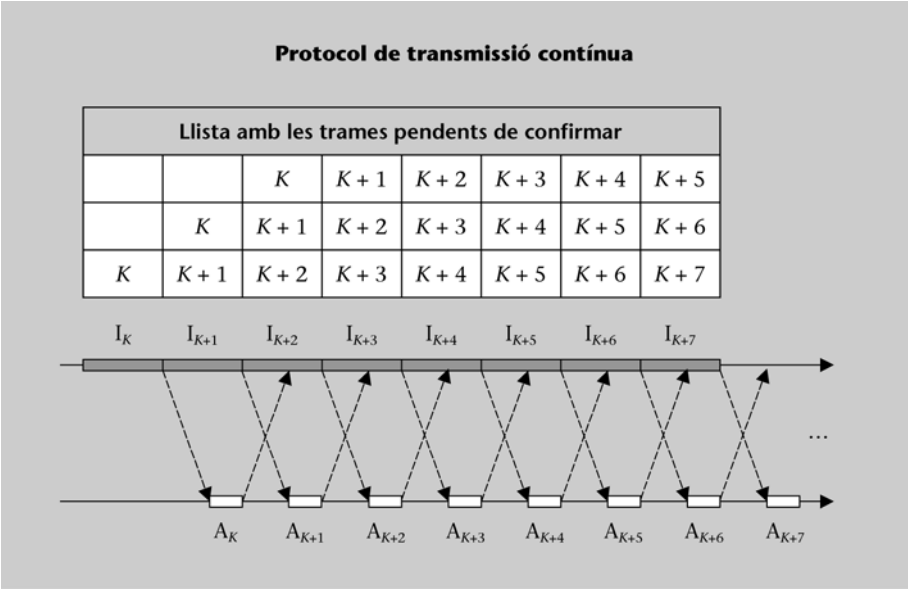
Evidentment, això només és possible si l'enllaç és *full-duplex*, perquè el primari i el secundari han de poder ocupar simultàniament el canal. En el cas del protocol *Stop & Wait* això no és necessari perquè el primari i el secundari ocupen el canal alternativament.

En cas d'error, el primari es pot veure obligat a retransmetre una trama anterior a l'última transmesa.

Així, doncs, en un protocol de transmissió contínua és necessari que el primari guardi en una llista les trames que s'han enviat i que estan pendents de confirmar, per tal de poder-les retransmetre en cas d'error. Aquesta llista s'anomena **llista de transmissió**.

La figura següent mostra el principi de funcionament d'un protocol de transmissió contínua. Cada trama transmesa es guarda en una llista i s'hi extreu quan n'arriba la confirmació. Es pot observar que el primari no es queda mai parat; és a dir, en absència d'errors l'eficiència d'un protocol de transmissió contínua és del 100%.

Podeu consultar el càlcul de l'eficiència en presència d'errors en l'annex d'aquest mòdul didàctic.



En cas d'error es fa servir una de les dues estratègies que s'expliquen a continuació: la retransmissió selectiva i el protocol *Go back N*.

3.4.1. Retransmissió selectiva

En l'estratègia de retransmissió selectiva el secundari només sol·licita la retransmissió de les trames errònies. En funció del conveni que es fa servir per a sol·licitar la retransmissió, hi ha dues versions d'aquest protocol: la implícita i l'explícita (anàlogues a les que hi ha en el protocol *Stop & Wait*).

Vegeu les versions implícita i explícita del protocol *Stop & Wait* en els subapartats 3.3.1 i 3.3.2 d'aquest mòdul didàctic.

Retransmissió selectiva implícita

Les regles que ha de complir una retransmissió selectiva implícita són les següents:

a) El secundari només envia confirmacions positives.

b) El secundari envia una confirmació per cada trama d'informació rebuda.

c) Si el primari rep dues confirmacions no consecutives, dedueix que la confirmació que manca correspon a una trama errònia i la retransmet.

d) Com en el protocol *Stop & Wait*, el primari ha d'activar un *time-out* per a la retransmissió de les trames pendents de confirmar. Això és així perquè una trama pot arribar amb errors en successives retransmissions, però el primari, amb la no-seqüència de les confirmacions, només en detectarà el primer.

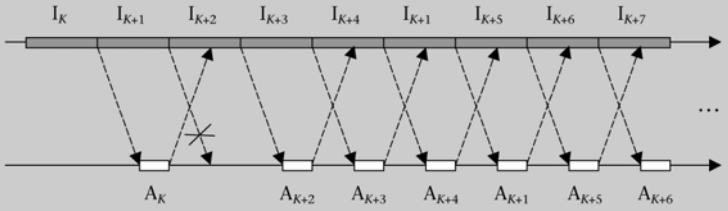
Tal com ja s'ha comentat, el nivell d'enllaç ha de lliurar les dades d'informació que porten les trames no tan sols sense errors, sinó també en ordre.

Si en una retransmissió selectiva es produeix un error, el secundari ha de memoritzar en una llista les trames que arriben després de la trama amb errors per tal de poder lliurar-les en ordre. Aquesta llista es coneix com a **llista de recepció**.

En la figura següent es mostra una condició d'error en una retransmissió selectiva implícita i la llista de recepció que cal que el secundari elabori per poder reordenar les trames:

Condició d'error en una retransmissió selectiva implícita

Llista amb les trames pendents de confirmar								
				$K+1$				
		K	$K+1$	$K+2$	$K+1$	$K+1$	$K+1$	$K+5$
	K	$K+1$	$K+2$	$K+3$	$K+3$	$K+4$	$K+5$	$K+6$
K	$K+1$	$K+2$	$K+3$	$K+4$	$K+4$	$K+5$	$K+6$	$K+7$



Llista per a poder reordenar les trames en cas d'error								
			$K+2$	$K+3$	$K+4$			
				$K+2$	$K+3$			
					$K+2$			

Nota

En la llista de recepció també podríem considerar la posició ocupada per la trama en fase de recepció.

- La trama I_{K+1} arriba amb errors i, per tant, el secundari no envia la confirmació corresponent.

- Quan la confirmació A_{K+2} arriba al primari, aquest s'adona de l'error en la trama I_{K+1} perquè no ha rebut la confirmació, i retransmet la còpia que guarda en la llista de trames no confirmades.

Retransmissió selectiva explícita

Les regles que ha de complir una retransmissió selectiva explícita són les següents:

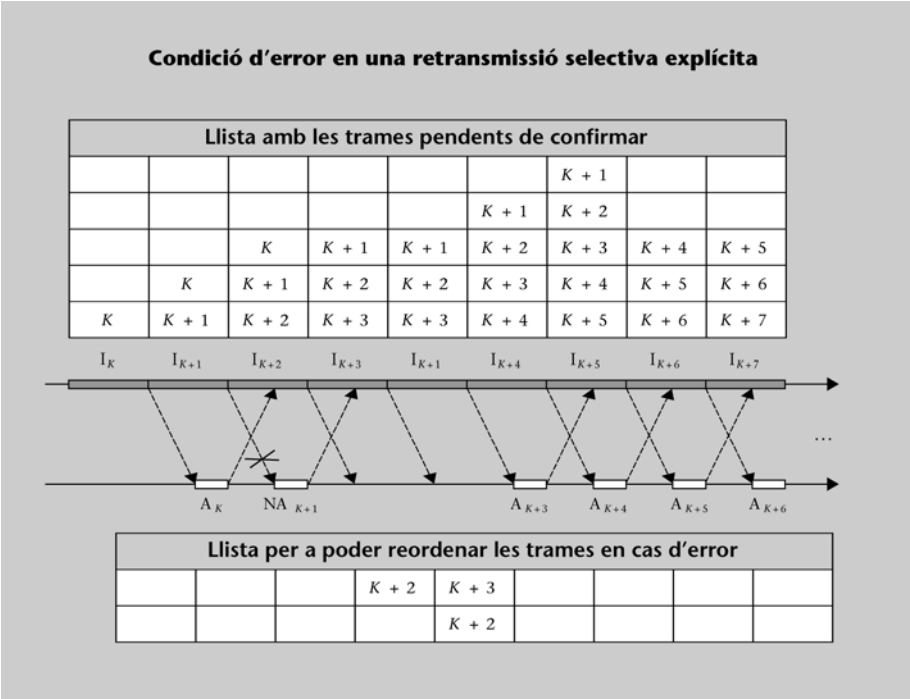
a) El secundari envia confirmacions positives quan la trama és correcta i de negatives quan rep una trama amb errors o fora de seqüència.

b) Si el primari rep una confirmació negativa NA_K, ha de retransmetre la trama I_K i interpreta que totes les trames anteriors a I_K s'han rebut correctament: una confirmació A_K confirma la trama I_K i totes les anteriors.

Una conseqüència d'aquesta regla és que, mentre el secundari espera la retransmissió d'una trama *K*, deixa d'enviar confirmacions positives o negatives de trames posteriors a la *K*.

c) El secundari ha d'activar un *time-out* per a la retransmissió de les confirmacions negatives de les trames no rebudes, ja que una confirmació negativa NA_K es podria perdre o arribar amb errors, i el primari no retransmetria la trama I_K.

La figura següent mostra el funcionament de la retransmissió selectiva explícita:



- La trama I_{K+1} arriba amb errors i, per tant, el secundari envia la corresponent confirmació negativa NA_{K+1} i deixa de transmetre confirmacions.

- Quan NA_{K+1} arriba al primari, aquest retransmet la trama I_{K+1}.

- Quan I_{K+1} arriba al secundari, aquest envia A_{K+3} i confirma totes les trames rebudes correctament fins aleshores.

3.4.2. Protocol *Go back N*

En molts enllaços la probabilitat d'error en una trama és relativament baixa, de manera que la retransmissió de trames és un esdeveniment poc freqüent. Aprofitant això, el protocol *Go back N* simplifica el mecanisme de recepció de les trames eliminant el problema de la reordenació.

Per tal d'eliminar el problema de la reordenació, en cas d'error el primari retransmet totes les trames a partir de la trama errònia. Així, quan el secundari detecta un error, ho comunica al primari i descarta totes les trames que li arriben fins a la nova recepció de la trama errònia, que anirà seguida de les trames descartades.

En la pràctica, el protocol *Go back N* només existeix en la versió explícita. Així, en cas d'error el secundari descarta les trames fins que el primari retransmet la trama errònia i, per tant, no tindria sentit enviar confirmacions de les trames descartades. Així, doncs, les regles del protocol *Go back N* són les següents:

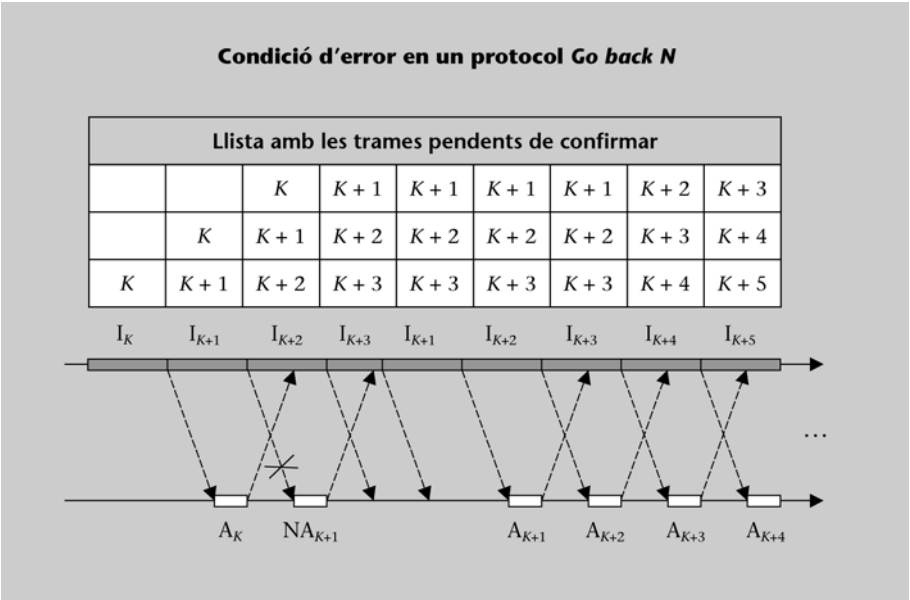
a) Quan el secundari rep una trama amb errors (o fora de seqüència), sol·licita la retransmissió de la trama errònia (o de la trama que falta) enviant una confirmació negativa.

b) El primari sempre transmet les trames en seqüència, de manera que si rep una confirmació negativa NA_K, retransmet la trama I_K i totes les posteriors, encara que ja les hagi transmès.

c) Una confirmació positiva A_K confirma la trama I_K i totes les anteriors. Així, doncs, la pèrdua d'una confirmació no té cap efecte perquè la confirmació següent ratifica la perduda. Una confirmació negativa de la trama *K* també ratifica totes les trames anteriors a *K*.

d) El secundari activa un *time-out* per a la retransmissió de les confirmacions negatives de les trames no rebudes.

La figura següent mostra el funcionament del protocol *Go back N*:



a) La trama I_{K+1} arriba amb errors i, per tant, el secundari envia la corresponent confirmació negativa NA_{K+1}.

b) Quan NA_{K+1} arriba al primari, aquest avorta la transmissió en curs de la trama I_{K+3} i comença a retransmetre immediatament la trama I_{K+1}.

Atès que el secundari no ha de reordenar les trames, no necessita una llista per a la reordenació.

4. Control de flux

El control de flux consisteix a regular la velocitat de transmissió de les trames. La necessitat del control de flux s'explica pels motius següents:

1) Pot ser que el mateix nivell d'enllaç no sigui capaç de processar les trames a la velocitat amb què arriben.

2) En els protocols de transmissió contínua, el primari necessita una memòria on emmagatzemar la llista de trames pendents de confirmar. Si no imposem cap limitació a la velocitat de transmissió de les trames, no serem capaços de dimensionar la memòria necessària.

3) Fins ara hem suposat que el nivell superior sempre està disposat a acceptar les trames del nivell d'enllaç. En la pràctica, però, és possible que això no sigui així. En aquest cas es podria esgotar la memòria on el secundari emmagatzema les trames rebudes.

Típicament, en el nivell d'enllaç l'algorisme de control de flux està lligat a l'algorisme de control d'errors. Per exemple, el protocol *Stop & Wait* serveix sense modificacions per a fer un control de flux. Simplement, el secundari ha de retardar les confirmacions per aconseguir disminuir la velocitat de transmissió del primari.

Control de flux X-ON/X-OFF

En algunes transmissions orientades a caràcter (on la unitat de transmissió del nivell físic és el caràcter) es fa servir el mecanisme de control de flux anomenat X-ON/X-OFF. En aquest mecanisme hi ha un caràcter especial (X-OFF) que s'envia per parar l'equip remot. Quan un dels equips rep un caràcter X-OFF, es para fins que rep el caràcter d'activació X-ON.

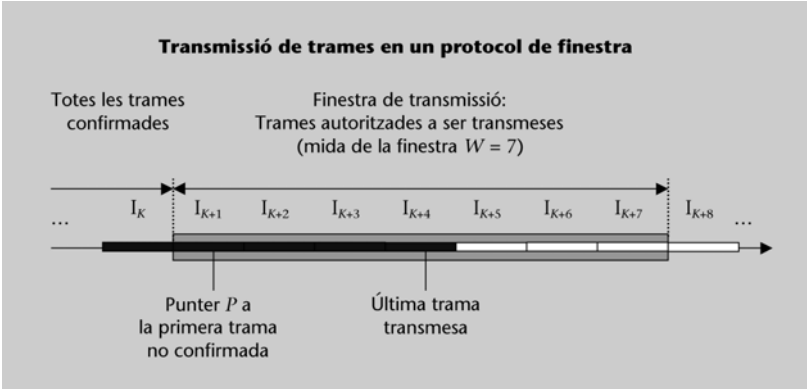
4.1. Protocols de finestra

Els algorismes de transmissió contínua no han d'esperar les confirmacions per a la transmissió de noves trames. Així, doncs, deixar d'enviar confirmacions no permet fer un control de flux. Per a fer-ho, es limita el nombre de trames que es poden transmetre sense confirmar. Els protocols resultants es coneixen com a *protocols de finestra*, perquè aquesta limitació es pot interpretar com una finestra que es desplaça sobre el flux de trames del primari a mesura que arriben les confirmacions.

Els **protocols de finestra** són protocols de transmissió contínua en què el primari pot transmetre fins a W trames després de la trama K, estant la trama K i totes les anteriors confirmades. El paràmetre $\boldsymbol{W}$ s'anomena **mida de la finestra**.

El transmissor manté un punter P a la primera trama no confirmada (totes les trames amb identificador més petit que P estan confirmades). Aquest punter s'actualitza a mesura que arriben les confirmacions.

El primari està autoritzat a transmetre trames amb identificadors < $P + W$. Les trames P, $P + 1$, ..., $P + W - 1$ es coneixen com a **finestra de transmissió**. Per exemple, si en les condicions de la figura següent no arriben més confirmacions, després de la transmissió de la trama I_{K+7} el primari es quedarà parat:



Per fer el control de flux en un protocol de finestra, el secundari simplement deixa d'enviar confirmacions. En aquest cas, el primari esgotarà la finestra i es pararà.

La finestra mínima que es pot definir és $W = 1$. En aquest cas, el primari no pot transmetre una nova trama fins que no s'ha confirmat l'anterior; és a dir, un protocol de finestra amb una finestra de mida W = 1 és equivalent a un protocol *Stop & Wait*.

4.2. Dimensionament de la llista de transmissió

Una propietat important de la finestra de transmissió és que permet dimensionar la memòria necessària per a emmagatzemar la llista de trames del primari pendents de confirmar.

El raonament que seguirem per a arribar a aquesta conclusió és el següent: si totes les trames amb identificador més petit que P (on P és el punter a la primera trama no confirmada) han estat confirmades, podem assegurar que el secundari no en sol·licitarà la retransmissió de cap i, en conseqüència, no cal guardar-les en la llista. Atès que el primari no pot haver transmès cap trama amb identificador més gran o igual que $P + W$ i que en la llista només hi ha guardades les trames transmeses, deduïm que en la llista de trames no confirmades del primari d'un protocol de finestra hi pot haver com a màxim W trames.

4.3. Finestra de recepció

Per analogia amb la finestra de transmissió, la **finestra de recepció** es defineix com el nombre màxim de trames que ha d'emmagatzemar el secundari.

Quan es defineix la finestra de recepció s'assumeix que el nivell d'enllaç treu de la llista de recepció totes les trames que no estan pendents de ser rebudes amb un identificador anterior. D'aquesta manera, es garanteix que les dades que porten aquestes trames passen sense errors i de manera seqüenciada al nivell superior. A més, en la finestra de recepció s'inclou la trama que es troba en fase de recepció.

Així, doncs, la finestra mínima en recepció és igual a 1 (per a emmagatzemar la trama en fase de recepció).

També s'han d'emmagatzemar les trames que es poden haver de reordenar. Com que en els protocols *Stop & Wait* i *Go back N* no existeix el problema de la reordenació de les trames, tenen una finestra de recepció que val 1.

El problema de la reordenació només sorgeix en el cas de la retransmissió selectiva. Per conèixer quin és el nombre màxim de trames que es poden haver de reordenar en el secundari basarem el raonament en la figura anterior. Atès que totes les trames amb identificadors més petits o iguals que K han estat confirmades, han d'haver estat rebudes sense errors pel secundari i, per tant, aquest les ha d'haver tret de la llista de recepció. Així, doncs, resulta que les úniques trames que pot haver d'ordenar el secundari són les $K + 1 \ldots K + W$ trames que el primari ha estat autoritzat a transmetre (en total W trames).

Vegeu la figura esmentada en el subapartat 4.1 d'aquest mòdul didàctic.

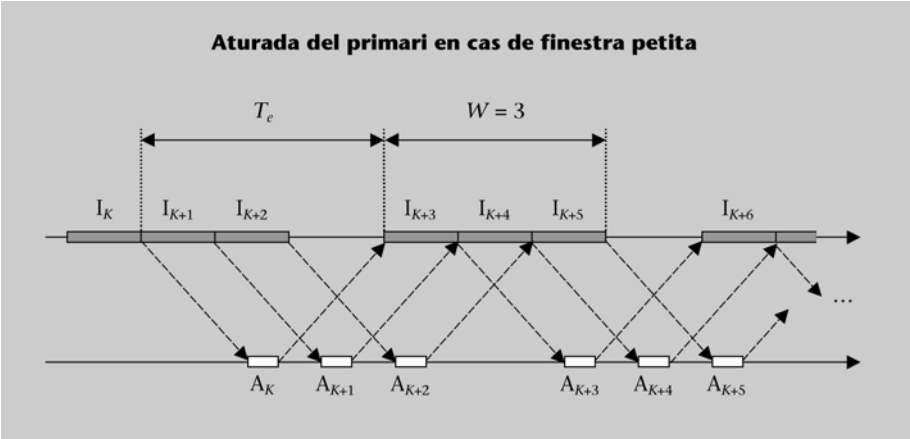
Per tant, en una retransmissió selectiva la mida de la finestra de recepció és igual que la mida de la finestra de transmissió.

La taula següent resumeix la mida de les finestres en els tres protocols:

Mida de les finestres de transmissió i recepció		
Protocol	**Finestra de transmissió**	**Finestra de recepció**
Stop & Wait	1	1
Go back N	*W*	1
Retransmissió selectiva	*W*	*W*

4.4. Finestra òptima

Hem vist que l'eficiència dels protocols de transmissió contínua és del 100% gràcies al fet que el primari no roman mai parat tot esperant confirmacions. En un protocol de finestra, aquesta condició pot no donar-se; és a dir, el transmissor es pot quedar parat si la mida de la finestra no és prou gran.



Finestra de transmissió petita

En la figura, la mida de la finestra és $W = 3$ (es poden transmetre tres trames sense confirmar). És a dir, després de transmetre una trama, el primari està autoritzat a transmetre'n dues més abans de rebre la primera confirmació. S'hi observa, però, que el temps d'espera de la confirmació (T_e) és major que el temps de transmissió de les dues trames i el primari es queda parat.

Activitat

Doneu el valor de l'eficiència obtinguda en la figura anterior en funció dels paràmetres t_t, t_a i t_p definits igual que en la figura "Avaluació del protocol *Stop & Wait*".

Vegeu la figura esmentada en el subapartat 3.3.4 d'aquest mòdul didàctic.

Solució

En cada interval de temps igual a $t_t + T_e$ (on $T_e = 2t_p + t_a$) es transmeten tres trames d'informació. Així, doncs: $E = 3t_t / (t_t + T_e) = 3\ t_t / (t_t + 2t_p + t_a) \approx 3 / (1 + 2a)$, on recordem que $a = t_p/t_t$ i s'ha suposat que $t_a << t_t$.

En l'activitat anterior podríem aconseguir una eficiència del 100% amb una finestra de transmissió més gran.

La **finestra òptima** (W_{opt}) és la finestra mínima que permet aconseguir una eficiència del protocol del 100%.

En l'exemple de la figura anterior hem calculat que la finestra òptima (W_{opt}) val el següent:

$$W_{opt} = 1 + \lceil T_e/t_t \rceil = 1 + \lceil (2t_p + t_a) / t_t \rceil,$$

on $\lceil x \rceil$ és l'enter major o igual que x.

4.5. Dimensionament del camp dels identificadors de trama

Fins ara hem suposat que les trames van numerades perquè es puguin identificar, sense preocupar-nos de les limitacions d'aquesta seqüència de nombres. En la trama hi haurà un camp, suposem que de n bits, destinat al número de seqüència. Atès que amb n bits podem generar 2^n identificadors diferents, numerant seqüencialment les trames es repetiran cíclicament els mateixos identificadors cada 2^n trames.

Amb *n* xifres binàries...

... es pot representar el conjunt d'enters que hi ha entre 0 i $2^n - 1$. Per tant, si el camp dels identificadors de trama és de n bits, disposem de 2^n identificadors diferents.

Així, doncs, hi ha el perill que en sol·licitar una retransmissió A_K hi hagi ambigüitat: es pot tractar de la trama I_K actual o de la del cicle anterior. Igual-

ment, s'ha d'evitar que el secundari interpreti com a trames duplicades les que arriben amb el mateix identificador, però que són de cicles diferents.

Per saber quants identificadors necessitem com a mínim per a evitar ambigüitats, farem una deducció informal per al cas del protocol *Stop & Wait*. Suposarem les mateixes condicions que les que ja hem enumerat abans: les trames poden arribar amb retards arbitraris, però arriben en l'ordre en què s'han transmès.

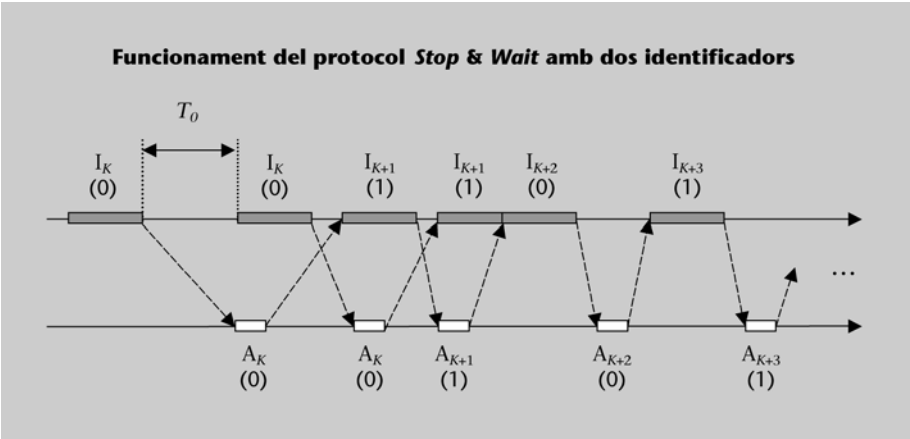
Vegeu les condicions que suposem respecte a la recepció de les trames en l'apartat 3 d'aquest mòdul didàctic.

Atès que en aquest protocol no es trenca mai la seqüència dels identificadors en la transmissió de les trames d'informació, aquestes sempre arribaran en ordre. D'aquest fet es dedueix que les confirmacions també arribaran sempre en ordre.

Suposem que el primari ha rebut la confirmació de la trama *K*, però no la de la *K* + 1. Com que la finestra val 1 i les trames d'informació arriben en ordre, en el secundari només poden arribar trames I_K i I_{K+1}. Atès que les confirmacions arriben en ordre, al primari no li pot arribar cap confirmació anterior a la *K*. Conseqüentment, només li poden arribar les confirmacions A_K i A_{K+1}. Deduïm, doncs, que en el cas del protocol *Stop & Wait* només hi pot haver ambigüitat entre dues trames i, per tant, només són necessaris dos identificadors diferents.

En la figura següent hi ha un exemple del funcionament del protocol *Stop & Wait* amb dos identificadors (entre parèntesis en la figura). En aquest cas, la confirmació de la trama I_K tarda un temps anormalment llarg i es retransmet (salta el *time-out* T_0). Això provoca que la trama I_K i la confirmació A_K arribin duplicades amb l'identificador* (0). Tant el primari con el secundari detecten la duplicació perquè la trama següent té l'identificador (1). Quan arriba la confirmació de la trama I_{K+1}, el primari avorta la segona transmissió d'aquesta trama i comença la transmissió de la següent (I_{K+2}).

*** L'identificador que es posaria en cada trama apareix entre parèntesis en la figura.**



La deducció del nombre d'identificadors necessaris en el cas dels protocols *Go back N* i retransmissió selectiva és més complexa i no la farem. El resultat és

Lectura complementària

Per a més detalls sobre la deducció del nombre d'identificadors necessaris en cada protocol es pot consultar l'obra següent, que trobareu referenciada a la bibliografia:

D. Bertsekas; R. Gallager. *Data Networks*.

que, si la mida de la finestra és W, es necessiten $W + 1$ identificadors diferents en el protocol *Go back N* i $2W$ en el de retransmissió selectiva.

El nombre d'identificadors necessaris, com es pot veure a la taula següent, coincideix amb la suma de la mida de les finestres de transmissió i de recepció:

Identificadors necessaris i finestra màxima en un protocol de finestra		
Protocol	**Nombre d'identificadors necessaris**	**Mida de la finestra màxima si es fan servir *n* bits per als identificadors**
Stop & Wait	2	1
Go back N	$W + 1$	$2^n - 1$
Retransmissió selectiva	$2W$	2^{n-1}

Finalment, la **mida de la finestra màxima W_{max}**, segons el nombre de bits (n) que es facin servir en el camp de l'identificador de trama, serà de $W_{max} + 1 = 2^n$ en el protocol *Go back N* i $2W_{max} = 2^n$ en el de retransmissió selectiva.

5. Altres consideracions sobre el nivell d'enllaç

Fins ara hem vist les funcions més importants associades al nivell d'enllaç, que són la sincronització de les trames, la detecció i el control d'errors, i el control de flux. En aquest apartat s'expliquen alguns detalls que també podem trobar en un protocol de nivell d'enllaç.

L'objectiu d'aquest apartat és, a més, donar exemples dels contextos on es pot trobar el nivell d'enllaç.

5.1. *Piggybacking*

Fins a aquest moment hem considerat sempre que hi havia una estació que transmetia trames d'informació (el primari) i una altra que les confirmava (el secundari). Ara bé, hi ha casos en què les dues estacions s'intercanvien trames d'informació recíprocament i, per tant, actuarien com a primari i secundari alhora i haurien d'alternar les trames d'informació amb les confirmacions.

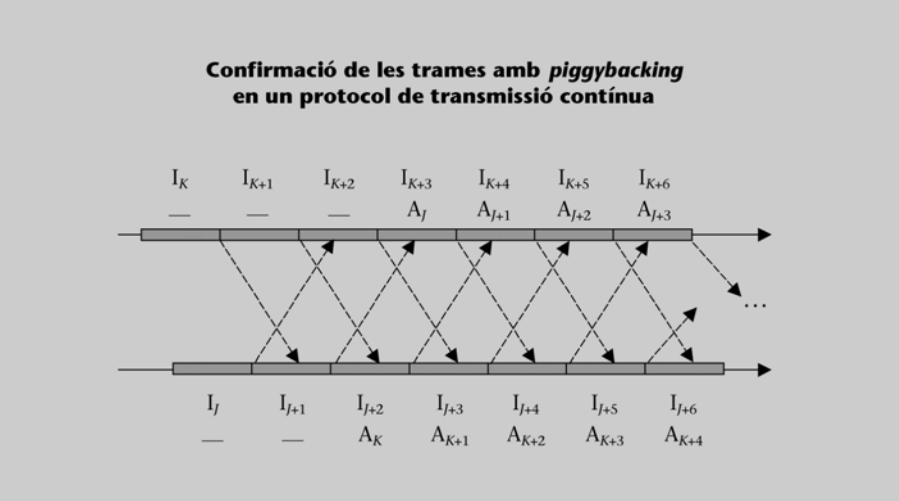
Les trames de confirmació solen tenir una mida petita perquè l'única informació rellevant és la de l'identificador de la trama que confirmen. La major part dels bits d'aquestes trames els ocupen els camps de control (indicadors de sincronització, CRC, etc.).

No ens hem preocupat d'això fins ara perquè l'eficiència que hem mesurat és l'obtinguda en el sentit primari-secundari. Però si les dues estacions envien informació, ens interessarà aconseguir que l'eficiència en tots dos sentits sigui tan alta com sigui possible.

Una manera d'augmentar l'eficiència en aquesta situació és incorporar les confirmacions a les trames d'informació (així ens estalviem la transmissió dels altres camps de control de les confirmacions).

La manera de fer-ho és destinant un camp de la trama d'informació a l'identificador de la trama que es vol confirmar del primari contrari. Aquesta tècnica es coneix com a ***piggybacking***.

La figura següent mostra el diagrama de temps que s'obtindria fent servir el *piggybacking*:



5.2. Sincronisme de protocol

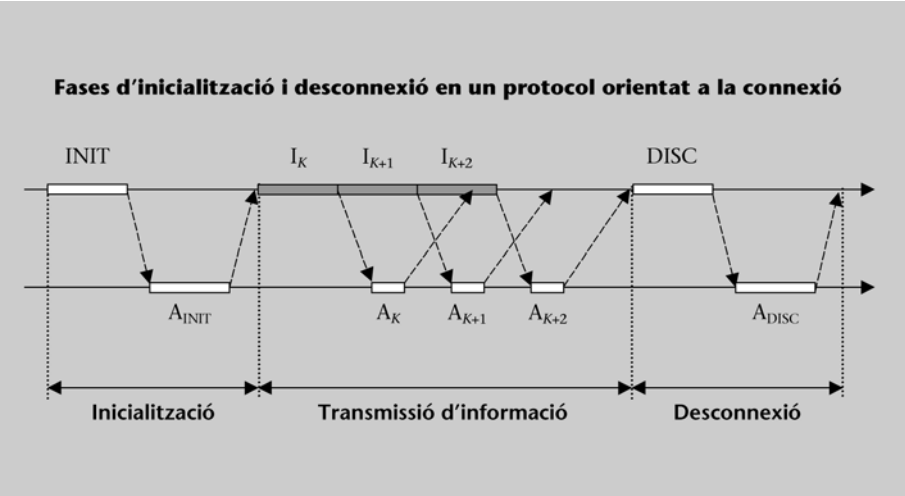
En la descripció que hem fet fins ara dels protocols de recuperació d'errors i control de flux, hem suposat implícitament que els protocols estaven correctament inicialitzats. És a dir, hem suposat que en els diagrames de temps que hem fet servir no hi havia trames d'informació ni confirmacions anteriors a les mostrades que poguessin introduir cap mena d'ambigüitat en els identificadors. En la pràctica això no és així. Abans que les dues entitats de nivell d'enllaç es puguin començar a intercanviar trames d'informació, és necessària una **fase d'inicialització**.

Es podria pensar que només cal fer la inicialització quan les dues entitats es posen físicament en marxa per primera vegada. Això no és possible perquè, en un cas real, pot passar que l'execució d'una de les entitats de nivell d'enllaç es quedi temporalment fora de servei, la qual cosa faria que es perdessin les trames emmagatzemades en la memòria de transmissió i recepció, i es perdria la sincronització; és a dir, s'hauria d'executar de nou la fase d'inicialització.

A més de la fase d'inicialització també hi ha una **fase de desconnexió**, en què s'acorda la finalització de l'enllaç. Aquesta fase permet alliberar els recursos que se li dediquen, com ara les memòries intermèdies per a emmagatzemar les llistes de transmissió i recepció.

Un protocol que té una fase d'inicialització i una fase de desconnexió es coneix com a **protocol orientat a la connexió** (*connection oriented*). Altrament, parlaríem de **protocol no orientat a la connexió** (*connectionless*).

La figura següent mostra la fase d'inicialització i la de desconnexió. La trama INIT sol·licita la inicialització, que és confirmada per la trama A_{INIT}. Les trames DISC i A_{DISC} funcionen de manera anàloga per a la desconnexió. Aquestes trames no estan numerades (per això se les coneix com a **trames no numerades** o *unnumbered frames*). En les trames hi ha un camp de control que indica el tipus de trama i que permet diferenciar-les:



Atès que per a poder fer una recuperació automàtica d'errors necessitem numerar les trames, aquests protocols hauran de ser protocols orientats a la connexió.

Però no tots els protocols de nivell d'enllaç són orientats a la connexió. Segons el tipus de xarxa o aplicació, és possible que no es faci servir un protocol de recuperació d'errors en el nivell d'enllaç i que, per tant, les trames no s'hagin de numerar. En aquest cas tindríem un nivell d'enllaç no orientat a la connexió que faria la detecció d'errors, però que descartaria les trames errònies en comptes d'intentar recuperar-les.

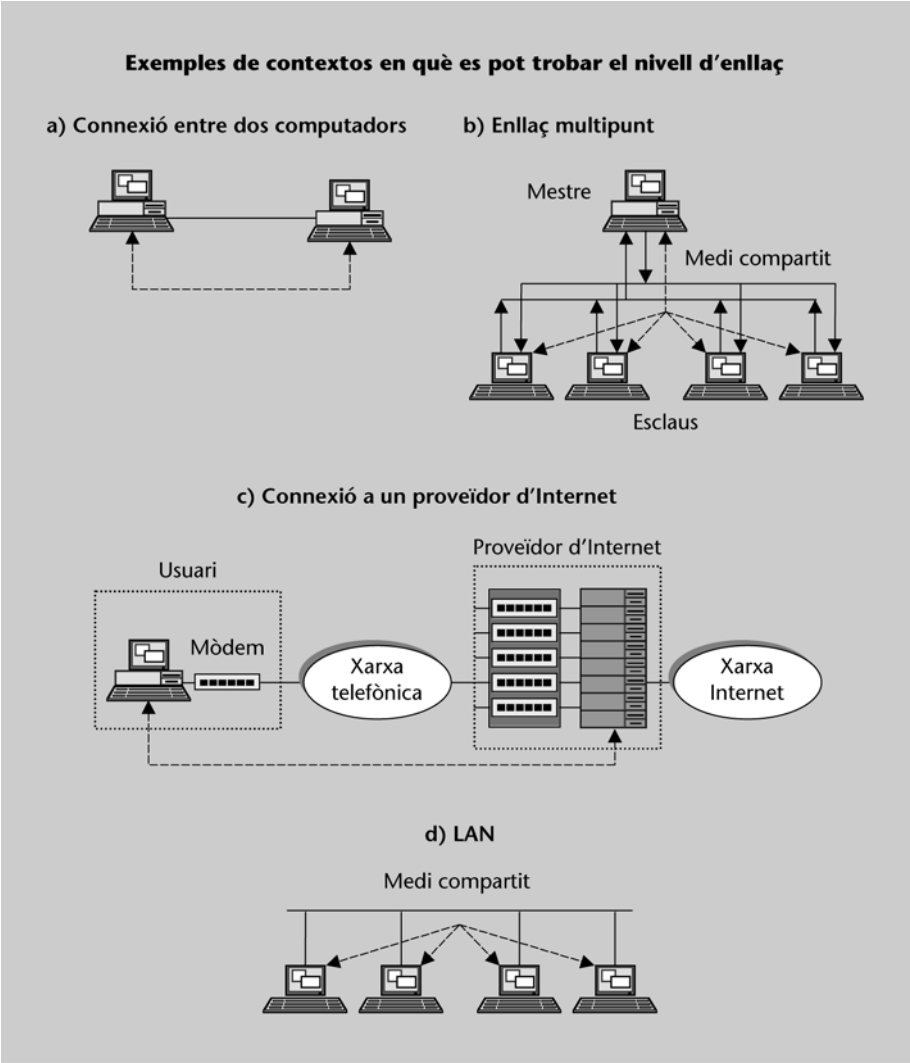
Hi ha motius que poden justificar un nivell d'enllaç no orientat a la connexió: en aplicacions en temps real en què es transmet veu o vídeo, per exemple, és possible que el retard que es necessitaria per a poder fer la recuperació d'errors no sigui acceptable.

5.3. Importància del nivell d'enllaç segons el context

Fins a aquest moment hem explicat les funcions que podem trobar en el nivell d'enllaç. És important entendre, però, que el nivell d'enllaç no sempre efectua totes les funcions explicades. Això depèn molt del context on es trobi el nivell d'enllaç.

Recuperació d'errors en el nivell de transport

Hi ha xarxes en què el nivell d'enllaç no fa una recuperació d'errors, sinó que simplement descarta les trames errònies. En aquest cas, si és necessari, la recuperació d'errors la fan els nivells superiors, normalment el nivell de transport.



Notació

En aquesta figura, les línies discontínues connecten els dispositius entre els quals s'estableix el protocol de nivell d'enllaç.

A continuació, descriurem les diferents situacions representades en la figura:

a) Comunicació punt a punt entre dos computadors; per exemple, la comunicació pel port sèrie entre dos PC per a poder fer una transferència de fitxers. A causa del reduït nombre d'elements que intervenen en aquest cas, normalment tota l'arquitectura de comunicacions estarà integrada en el mateix programa. Evidentment, aquí no hi haurà nivell de xarxa i el nivell d'enllaç serà responsable de la recuperació d'errors.

b) Enllaç multipunt. Aquesta situació és semblant al cas anterior, però el que entra en comunicació és una comunitat de computadors i ho fa a través d'un medi compartit. Per a poder comunicar-se sense interferir-se entre ells, n'hi ha un (el **mestre** en la figura) que s'encarrega d'arbitrar les comunicacions. Totes les comunicacions són entre el mestre i un altre dels computadors (els **esclaus**), o viceversa.

Exemples d'entorn amb enllaç multipunt

Una comunitat de terminals d'un departament universitari, o un conjunt d'instruments intel·ligents d'una planta industrial controlats per un computador central.

Aquesta condició és representada per les fletxes que hi ha entre el mestre i cada un dels esclaus (en tots dos sentits). En aquest entorn podem interpretar que hi ha un enllaç punt a punt entre el mestre i cada un dels esclaus, i que el mes-

tre selecciona alternativament un dels enllaços possibles segons algun algorisme d'arbitratge.

c) Entorn d'accés a Internet. Actualment és un dels més habituals. Milions d'usuaris el fan servir per a accedir-hi. En aquest entorn el protocol de nivell d'enllaç s'estableix entre el computador de l'usuari i el computador del proveïdor d'Internet. Aquí l'usuari es connecta al proveïdor per mitjà d'un mòdem. El proveïdor disposa d'una bateria de mòdems perquè múltiples usuaris s'hi puguin connectar simultàniament.

d) Xarxa d'àrea local (LAN). Una característica d'aquest tipus de xarxes és que la comunitat de computadors comparteix el medi de transmissió, igual que en el cas **b)**. La diferència fonamental, però, és que ara no hi ha un àrbitre que selecciona un dels enllaços possibles, sinó que l'algorisme d'accés està distribuït entre els computadors que accedeixen al medi.

5.4. Multiplexació al nivell d'enllaç

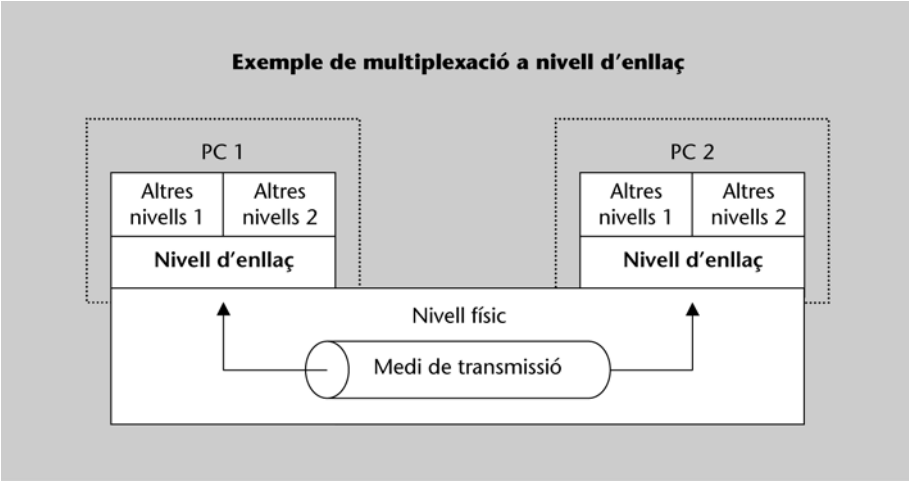
El concepte de *multiplexació* ja s'ha introduït anteriorment. La idea de la multiplexació és fer servir un únic medi per a la transmissió de diferents fluxos d'informació. Òbviament, cal definir un mecanisme que permeti separar els diferents fluxos pel que fa a la recepció.

Ara veurem que la multiplexació també es pot trobar en un protocol de nivell d'enllaç. De fet, és habitual que aquesta tècnica es faci servir en qualsevol nivell de l'arquitectura de comunicacions.

Primer de tot, vegem per a què pot ser útil tenir multiplexació en el nivell d'enllaç. El primer exemple és l'enllaç multipunt, que ja hem introduït (vegeu l'esquema **b)** de la figura anterior). El fet de tenir el medi compartit el podem interpretar com si hi hagués un únic enllaç on es multiplexen les trames de les diferents estacions. En aquest cas l'algorisme d'arbitratge que fa servir el mestre el podríem interpretar com el mecanisme de multiplexació.

Dintre d'aquest enllaç compartit necessitarem distingir cadascuna de les estacions. Això s'aconsegueix amb una **adreça de nivell d'enllaç**. És a dir, un dels camps de control de les trames és una adreça que identifica cada estació.

Un altre exemple de multiplexació a nivell d'enllaç és el que mostra la figura següent. En aquest exemple hi ha un nivell físic i d'enllaç comú i, per sobre, dues arquitectures de comunicacions (conjunt de protocols) diferents. En aquest cas el nivell d'enllaç porta les unitats d'informació de l'arquitectura 1 o 2 a la seva arquitectura parella 1 o 2, respectivament.



Aquest cas no és estrany en la pràctica, ja que hi ha nombrosos protocols de comunicacions que pot interessar que convisquin dintre d'un mateix entorn i que comparteixin el mateix enllaç; per exemple, en una xarxa d'àrea local.

Per a poder distingir l'arquitectura que transmet les trames i, per tant, l'arquitectura a la qual cal lliurar-les també es fan servir **adreces de nivell d'enllaç**.

6. Exemple: HDLC i ATM

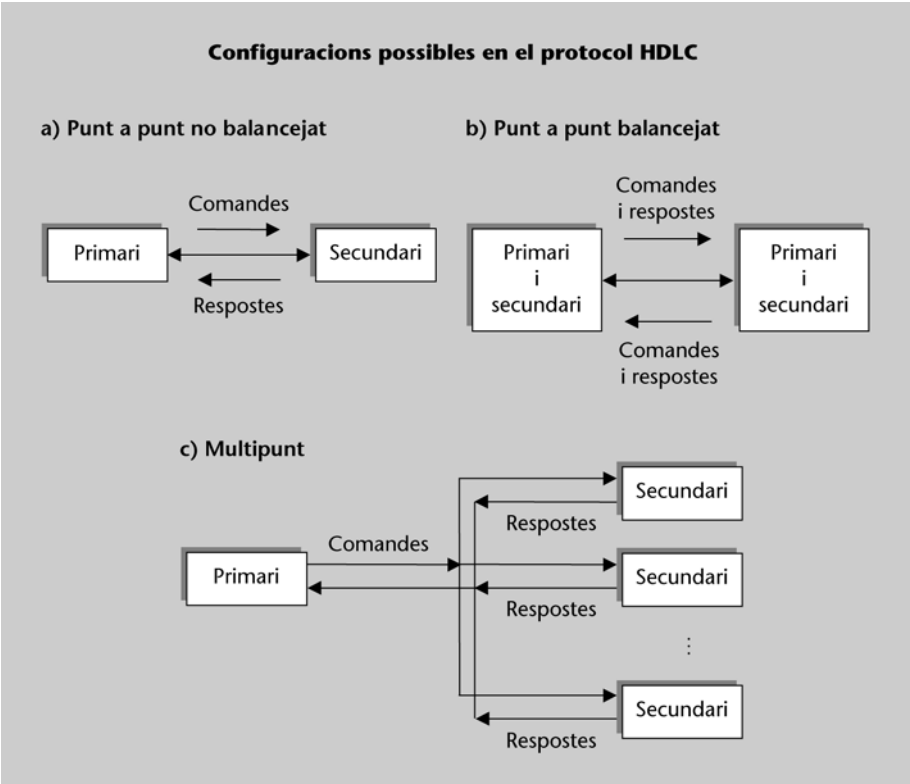
Com a exemple d'un estàndard de nivell d'enllaç, en aquest apartat descriurem algunes de les característiques de l'HDLC*. i de l'ATM.** HDLC ha estat estandarditzat per l'organisme ISO i molts altres protocols de nivell d'enllaç en deriven, mentre que ATM és un estàndard de l'ITU-T.

*** HDLC és la sigla del terme anglès *high-level data link control.***
**** ATM és la sigla d'*asynchronous* transfer mode.**

6.1. HDLC

L'HDLC ha estat dissenyat com un protocol de nivell d'enllaç de propòsit general que es pot fer servir en nombroses situacions. Per això s'han definit les tres configuracions de la figura següent. En aquesta figura els noms ***primari*** i ***secundari*** no fan referència a l'estació que envia les trames d'informació i les confirmacions, com fins ara, sinó que aquesta terminologia es fa servir per a distingir certs privilegis que té el primari. El privilegi principal de l'HDLC és que pot fer d'àrbitre en un enllaç multipunt (el que hem anomenat *mestre*).

Vegeu el concepte de mestre en el subapartat 5.3 d'aquest mòdul didàctic.



Exemples de protocols derivats de l'HDLC

Alguns protocols que s'han derivat del protocol HDLC són els que esmentem a continuació:

- LLC: es fa servir en les xarxes d'àrea local (LAN).
- PPP: s'utilitza en Internet.

- LAPM: és emprat en el camp de la comunicació entre mòdems.
- LAPD: es fa servir com a canal de senyalització (canal D) en l'accés a la xarxa telefònica digital ISDN.
- LAPB: s'utilitza per a connectar una estació a una xarxa de gran abast X.25.

En l'HDLC les trames que envia el primari es coneixen com a **comandes**, i les que envia el secundari, com a **respostes**. Si en la configuració totes les estacions són primari i secundari alhora, es diu que és una **configuració balancejada**; altrament, es parla de **configuració no balancejada**. Evidentment, una configuració multipunt ha de ser no balancejada, perquè només hi pot haver una estació que sigui primari (que faci d'àrbitre).

L'HDLC permet **tres modes d'operació diferents**:

a) ***Normal response mode*** (**NRM**): aquest mode s'utilitza en configuracions no balancejades (casos *a* i *c* de la figura anterior). En aquest mode d'operació els secundaris (o esclaus) només poden transmetre trames en resposta a les comandes enviades pel primari (o mestre). Aquest mode es fa servir típicament en l'enllaç entre un computador i un o més perifèrics.

b) ***Asynchronous response mode*** (**ARM**): aquest mode es fa servir en configuracions punt a punt no balancejades (cas *a* de la figura). A diferència del mode NRM, el secundari pot transmetre sense que el primari ho sol·liciti. Aquest mode és el menys utilitzat en l'actualitat.

c) ***Asynchronous balanced mode*** (**ABM**): aquest mode es fa servir per a configuracions punt a punt balancejades (cas *b* de la figura). Els protocols derivats de l'HDLC més utilitzats, com ara el LABM, el LABD i el LABB, deriven d'aquest mode.

Independentment del mode, en l'HDLC es defineixen **tres tipus de trames**:

a) Trames d'informació: equivalen a les trames d'informació a què ens hem referit fins ara. Per a la recuperació d'errors i control de flux es fa servir el *Go back N* o la retransmissió selectiva (tot i que molts dels derivats implementen només el *Go back N*). Per a les confirmacions s'utilitza la tècnica del *piggybacking*.

b) Trames de supervisió: equivalen a les trames de confirmació a què ens hem referit fins ara. S'utilitzen per a les confirmacions quan no s'envien trames d'informació.

c) Trames no numerades: es fan servir en les fases d'inicialització i de desconnexió i per a enviar informació de control suplementària.

L'HDLC és un protocol amb una estructura de les trames com la que es mostra en la figura següent. Els indicadors d'inici i final de trama valen 01111110, i es fa servir la tècnica de la inserció de zeros per a la transparència. La compro-

Hem vist la tècnica de la inserció de zeros en l'exemple de mecanisme de transparència en el subapartat 1.2.1. El control CRC s'explica en l'apartat 2.6 d'aquest mòdul didàctic.

vació d'errors es realitza amb codis CRC, concretament amb el polinomi generador CRC-CCITT.

El **contingut del camp d'adreça** depèn del mode d'operació; per exemple, en un enllaç multipunt treballant amb el mode NRM només hi ha un primari i cada secundari té una adreça única. En aquest cas, en el camp d'adreça es posa sempre l'adreça del secundari. En el mode ABM, en canvi, es posa l'adreça pròpia quan es vol enviar una comanda (l'estació actua com a primari) i l'adreça contrària quan s'envia una resposta (l'estació actua com a secundari).

Els **bits de control** són els següents:

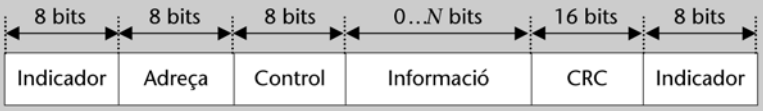
- **SN**: porta el número de seqüència d'una trama d'informació.
- **RN**: porta el número de seqüència de la trama confirmada.
- **P/F (*poll/final* bit)**: aquest bit té molts usos, alguns dels quals són els següents:

– El primari posa aquest bit a 1 per sol·licitar una resposta urgent d'un secundari.

– El secundari posa el bit a 1 per indicar el final d'una resposta.

- **Tipus**: el tercer i quart bits codifiquen el tipus de trama de supervisió, i els sisè, setè i vuitè, el tipus de trama no numerada.

Estructura de les trames en el protocol HDLC



Tipus de trama	Bits de control							
	1	2	3	4	5	6	7	8
Informació	0	SN			P/F	RN		
Supervisió	1	0	Tipus		P/F	RN		
No numerada	1	1	Tipus		P/F	Tipus		

L'estàndard descriu el tipus de trama i el comportament que s'ha de seguir en totes les situacions possibles.

En aquest apartat només s'han presentat els ingredients bàsics de l'estàndard perquè l'estudiant es faci una idea de tots els punts que involucra l'estandardització d'un protocol de nivell d'enllaç.

Lectura complementària

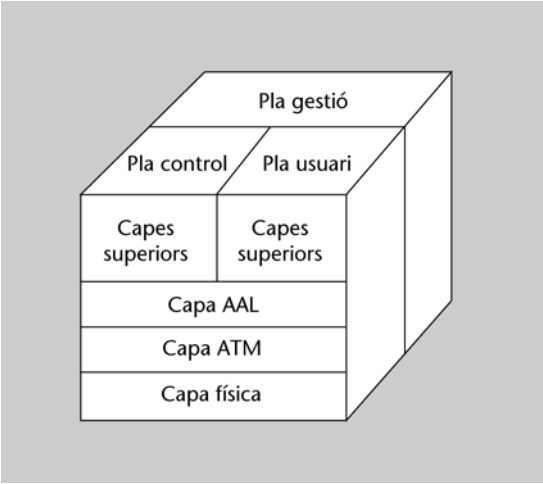
Per a obtenir més detalls sobre el protocol HDLC, consulteu l'obra següent:

F. Halsall (1998). *Comunicación de datos, redes de computadores y sistemas abiertos* (cap. 5). Argentina: Addison-Wesley Iberoamericana.

6.2. ATM

ATM és una tecnologia de commutació de paquets que es basa en l'ús de línies de transmissió ràpides i fiables. Això li permet aconseguir altes velocitats de transmissió, no només perquè les línies ho permeten, sinó també perquè els protocols intermedis (transport, xarxa, enllaç) poden ser molt més simples que els altres que hem vist, gràcies a la fiabilitat de les línies.

A continuació descrivim la funcionalitat del nivell d'enllaç a ATM, per poder comparar-ho amb l'HDLC.



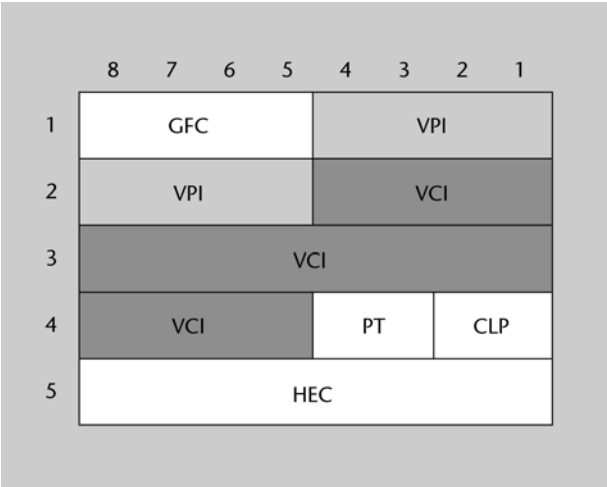
A la figura anterior es pot veure l'arquitectura de protocols d'ATM. La capa ATM és on es duen a terme les tasques que hem associat al nivell d'enllaç en terminologia OSI.

La capa ATM defineix que la transmissió de dades sobre el medi físic es faci en paquets o cel·les de mida fixa. Aquesta mida es va fixar en 53 bytes, 48 d'informació i 5 de capçalera. Una mida fixa permet un encaminament més eficient, i una mida petita permet que les cel·les amb alta prioritat pateixin retards molt petits a les cues.

Cel·la

A ATM el grup de bytes bàsic (el que en altres protocols és una trama o un paquet) es diu cell (cel·la).

A la figura següent es pot veure el format de la cel·la ATM:



Els camps de la capçalera són els següents:

- GFC (*generic flow control*). Aquest camp només es fa servir en la interfície usuari-xarxa, i no en les connexions dins de la xarxa. Permet fer el control de flux en aquesta interfície per a diferents qualitats de servei. En ATM les transmissions poden ser controlades o no controlades. Els diferents valors d'aquests 4 bits permeten especificar diferents tipus de control.

- VPI (*virtual path identifier*). És un camp d'encaminament.

- PT (*payload type*). Aquest camp indica el tipus d'informació que transporta la cel·la. El primer bit a 0 vol dir que és informació d'usuari, i llavors el segon bit indica si s'ha experimentat congestió o no. Si el primer bit val 1 la cel·la transporta informació de gestió de la xarxa.

- CLP (*cell loss priority*). Es fa servir en cas de congestió. Si aquest bit val 0, la cel·la no hauria de ser descartada, tret que no hi hagi més remei, mentre que un valor 1 indica que pot ser descartada.

- HEC (*header error control*). Aquests 8 bits serveixen per a protegir els 32 bits restants de la capçalera. Això vol dir suficient redundància per a no només detectar errors, sinó també per a intentar corregir-ne.

Resum

En aquest mòdul hem estudiat les **funcions del nivell d'enllaç**. Per sota d'aquest nivell es troba el nivell físic, que amb una probabilitat més gran o més petita introduirà errors en els bits transmesos. La tasca fonamental del nivell d'enllaç és **detectar aquests errors de transmissió**.

Amb aquesta finalitat el nivell d'enllaç organitza la informació que s'ha de transmetre en **trames**. Les trames són estructures organitzades amb bits d'informació i bits de control, alguns dels quals serveixen per a detectar si la trama ha arribat amb errors. El sincronisme de trama és un mecanisme que fa servir el nivell d'enllaç per a poder determinar on comencen i on acaben les trames dins del flux de bits que rep del nivell físic.

El nivell d'enllaç, cada vegada que rep una trama, comprova si hi ha errors. En cas afirmatiu pot descartar la trama o solucionar-ho. El fet d'escollir un protocol del primer tipus o del segon depèn del context on es fa servir el nivell d'enllaç, com ara el tipus de xarxa o l'aplicació.

Quan es fa una **recuperació d'errors**, el protocol normalment sol·licita la retransmissió de la trama errònia. Els protocols d'ús més generalitzat, en aquest sentit, són l'***Stop & Wait***, el ***Go back N*** i la **retransmissió selectiva**. Aquests protocols tenen diferents graus d'eficiència i complexitat. Segons la probabilitat d'error existent i els retards que es produeixin, serà més adient utilitzar-ne un o altre.

Una altra funció del nivell d'enllaç és regular la velocitat de transmissió de les trames, cosa que es coneix com a **control de flux**. El mateix protocol de control d'errors se sol fer servir, amb petites modificacions, per a fer un control de flux. Els protocols resultants es coneixen com a **protocols de finestra**.

Exercicis d'autoavaluació

1. En un enllaç es transmet la paraula codi 011001 i es rep 111101.
a) Quina distància de Hamming ha de tenir el codi per a assegurar la detecció de l'error?
b) Quina capacitat de detecció de ratxes d'error de bits necessitem?
c) Quina distància de Hamming ha de tenir el codi per a assegurar la correcció de l'error?

2. Un codi detector d'errors només té les tres paraules codi vàlides que es mostren a continuació: 110000, 001100 i 000011.
a) Quina és la distància de Hamming del codi?
b) Si es rep la paraula codi 001000 i s'hi aplica el criteri de correcció de la distància mínima, quins bits corregiríeu?

3. En un nivell d'enllaç que fa servir un codi de paritat longitudinal i transversal es rep la paraula codi següent:

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

a) És una paraula codi vàlida?
b) Si s'hi aplica el criteri de correcció de la distància mínima, quins bits corregiríeu?

4. Volem comparar la capacitat de detecció de ratxes d'error d'un codi de paritat longitudinal i transversal amb un codi CRC. Considereu un exemple de cada cas amb el mateix nombre de bits d'informació i bits de control d'errors. Quina és la capacitat de detecció de ratxes d'error en cada un dels casos? Quin és, doncs, el codi més robust per a la detecció de ratxes d'error?

5. Calculeu la velocitat efectiva a què es transfereixen les dades en un enllaç amb una velocitat de transmissió de 9.600 bps si les trames són de 800 bytes amb el 85% de bytes d'informació i el 15% de control. Suposeu que a causa del protocol de nivell d'enllaç hi ha un interval de repòs de 10 ms entre dues trames consecutives.

6. El temps de propagació entre dues entitats de nivell d'enllaç és de 10 ms. La velocitat de transmissió és de 28 kbps. La mida de les trames és de 800 B.
a) Quina és l'eficiència màxima que es pot aconseguir si es fa servir un protocol *Stop & Wait*?
b) Quina pot ser la mida màxima de les confirmacions perquè l'eficiència no sigui inferior al 90%?

7. En un enllaç on la mida de les trames d'informació és molt més gran que les trames de confirmació, raoneu en quines condicions triaríeu un protocol de nivell d'enllaç dels tipus que esmentem a continuació:
a) *Stop & Wait*
b) *Go back N*
c) Retransmissió selectiva

8. En un enllaç es fa servir un protocol *Stop & Wait*. La mida de les trames d'informació és variable, però la capçalera és fixa i de 50 bits. Les confirmacions són totes de 100 bits. La distància de l'enllaç és d'1 km i la velocitat de propagació de medi és de $2 \cdot 10^8$ m/s. La velocitat de transmissió és de 10 Mbps.
a) Dibuixeu el diagrama de temps indicant-hi els aspectes més rellevants en els casos següents:
- Transmissió d'una trama amb èxit.
- Transmissió d'una trama que es perd amb la corresponent retransmissió. Suposeu que la retransmissió és deguda a un *time-out*. Feu una breu explicació del mecanisme del *time-out*.

b) Calculeu el nombre de bits de dades que hauria de portar cada trama d'informació perquè l'eficiència fos com a mínim del 90%.

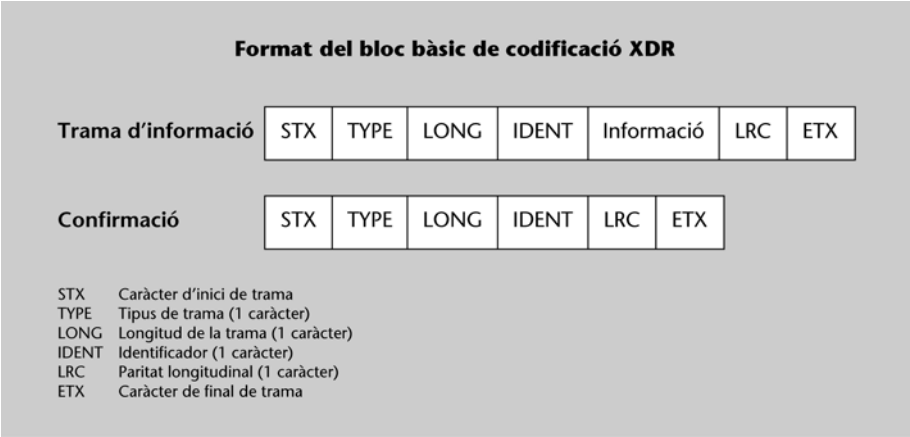
9. En una fàbrica hi ha un enllaç a 10 Mbps de 600 m de distància i velocitat de propagació de $2 \cdot 10^8$ m/s. Es fa servir un protocol *Stop & Wait* explícit amb trames de 1.500 bits de longitud i confirmacions de 40 bits. Una màquina propera produeix espurnes a una freqüència de 50 Hz que impedeixen la recepció correcta de la trama que s'estava transmetent en l'instant de l'espurna.
a) Dibuixeu el diagrama de temps amb els aspectes més rellevants quan es produeix l'espurna. Suposeu que l'espurna coincideix sempre amb la transmissió d'una trama d'informació.
b) Calculeu l'eficiència màxima quan no hi ha errors.
c) Calculeu l'eficiència màxima tenint en compte les retransmissions degudes a la interferència.

10. Indiqueu quant val la finestra òptima en un enllaç amb retransmissió selectiva, si el temps de propagació és negligible, en cadascun dels casos següents:
a) Si la mida de les confirmacions és menor que la de les trames d'informació.
b) Si es fa servir el *piggybacking* amb trames de la mateixa mida.

11. Dues estacions de característiques iguals es comuniquen amb un enllaç *full-duplex* de 100 kbps. Les estacions generen trames d'informació de mida variable entre 1 i 5 kbit. En l'enllaç es fa servir un mecanisme de finestra *Go back N*. Per a les confirmacions es fa servir la tècnica del *piggybacking* sempre que sigui possible, però si hi ha alguna confirmació pendent per un temps igual a 100 ms i l'estació continua inactiva, s'envia una trama de confirmació de 200 bits. El temps de propagació entre les estacions es considera negligible.
a) Calculeu la finestra òptima dibuixant el diagrama de temps que feu servir en el càlcul.
b) Quants bits de la capçalera s'haurien de dedicar als identificadors de les trames?

12. Es vol transmetre un fitxer d´1 Mbit entre dos PC que disten 10 km. El nivell d'aplicació divideix el fitxer en segments d'1 kB i els passa al nivell d'enllaç. El nivell d'enllaç envia un segment en cada trama d'informació i fa servir un protocol *Stop & Wait* per a fer la recuperació dels errors. L'enllaç té, a més, les característiques següents:
- Transmissió orientada a caràcter, asíncrona amb 8 bits per caràcter, 1 bit d'arrencada, 1 bit de parada i 1 bit de paritat.
- Velocitat de transmissió de 115.200 bps.
- Velocitat de propagació de 2 _ 10^8 m/s.
- Probabilitat d'error en un bit $p_b = 10^{-5}$.

La figura següent mostra el format de les trames i de les confirmacions:



a) Dibuixeu el diagrama de temps de la transmissió d'una trama.
b) Enumereu els factors que influeixen en l'eficiència. Comptabilitzeu la pèrdua d'eficiència deguda a cada un d'aquests factors.
c) Doneu la velocitat de transmissió eficaç.
d) Calculeu el temps mitjà de transmissió del fitxer.

Solucionari

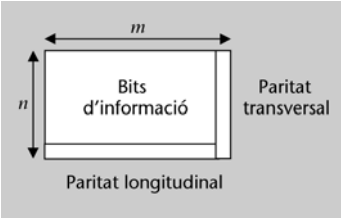
1.
a) En la paraula codi rebuda hi ha 2 bits erronis; això implica que l'error es detectarà amb probabilitat 1 si la distància de Hamming és més gran que 2.
b) La ratxa d'error és de 4 bits; això implica que l'error es detectarà amb probabilitat 1 si la capacitat de detecció de ratxes d'error és més gran o igual que 4.
c) Per a corregir n bits erronis amb probabilitat 1 s'ha de complir que $D_H/2 > n$, on D_H és la distància de Hamming del codi. Així, doncs, $D_H > 4$.

2.
a) Entre qualsevol de les paraules codi hi ha 4 bits de diferència. Així, doncs, la distància de Hamming val 4.
b) El criteri de la distància mínima consisteix a agafar la paraula codi vàlida amb menys bits de diferència. És la paraula codi 001100.

3.
a) No és una paraula codi vàlida perquè la paritat longitudinal i la transversal indicades no són correctes.
b) El bit que cal corregir aplicant el criteri de distància mínima és el que hi ha en l'encreuament de les paritats longitudinal i transversal incorrectes.

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

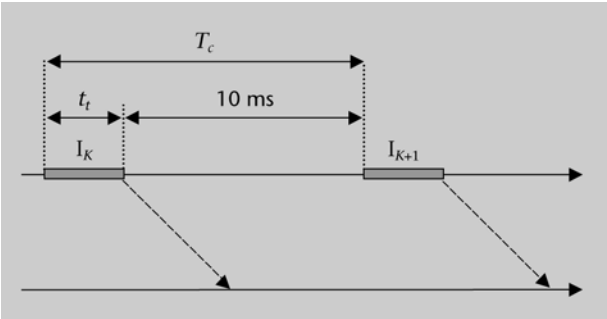
4. Suposem que tenim un codi de paritat longitudinal i transversal amb un bloc de $n \times m$ bits d'informació com el de la figura:



La capacitat de detecció de ratxes d'error (diguem-ne CDR) d'aquest codi és la mida d'una fila més 1, és a dir, $m + 2$. El nombre de bits dedicats al control de paritat és $n + m + 1$. Per tant, un codi CRC amb el mateix nombre de bits dedicats al control d'errors tindria un CDR de $n + m + 1$ bits. Així, doncs, només en el cas de $n = 1$ el codi de paritat longitudinal i transversal i el codi CRC tindrien un mateix CDR. Amb $n = 1$, però, la redundància és molt elevada (hi ha més bits dedicats al control d'errors que bits d'informació!). Així, doncs, cal suposar que n és més gran que 1 i que el codi CRC té més capacitat de detecció de ratxes d'error.

5. L'eficiència de les trames és del 85%. Per a calcular l'eficiència del protocol s'ha de tenir en compte l'interval de repòs. La figura que veieu a continuació mostra el diagrama de temps:

Vegeu la figura "Avaluació del protocol *Stop & Wait*" en el subapartat 3.3.4 d'aquest mòdul didàctic.



on tenim el següent*:

* Es farà servir el signe ≡ per a indicar que hi ha una conversió d'unitats.

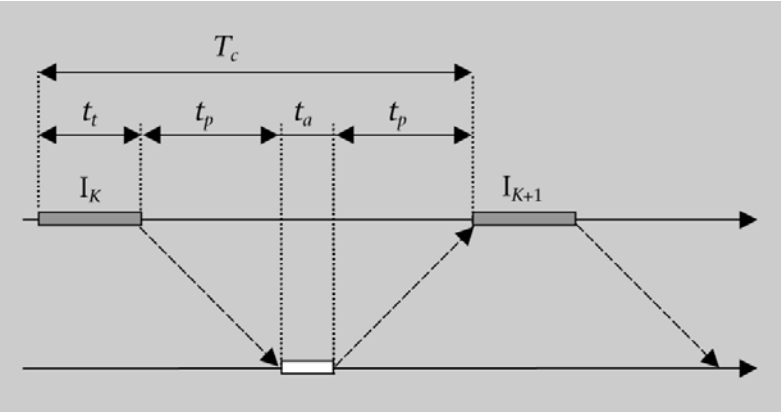
- $t_t = \frac{L}{v_t} = \frac{800 \cdot 8}{9.600} \equiv 666{,}7$ ms

- $E_P = \frac{t_t}{T_C} = \frac{t_t}{t_t + 10 \text{ ms}} \equiv 98{,}5\%$

- $E_{tot} = E_P \cdot E_t = 0{,}985 \cdot 0{,}85 \equiv 83{,}7\%$

- $v_{ef} = v_t \cdot E_{tot} = 9.600 \cdot 0{,}837 \equiv 8.035$ bps

6.
a) La figura següent mostra el diagrama de temps que es correspon al cas que estem considerant a l'enunciat:



Se suposa que $t_a \approx 0$, $t_t = L/v_t = 800 \cdot 8 / 28 \cdot 10^3 \equiv 228{,}6$ ms; així, doncs, obtenim:

$$E_P = \frac{t_t}{T_C} = \frac{t_t}{t_t + 2t_p} = \frac{228{,}6}{228{,}6 + 2 \cdot 10} \equiv 92\%$$

b) Tenim:

$$E_P = \frac{t_t}{t_t + 2t_p + t_a} > 0{,}9 \ ,$$

on $t_a < t_t\,(1/0{,}9 - 1) - 2\,t_p \equiv 5{,}4$ ms. Per tant, si denotem amb K la mida de les confirmacions, aleshores $K/v_t < 5{,}4 \cdot 10^{-3}$, $K < v_t \cdot 5{,}4 \cdot 10^{-3} = 151{,}2$ i finalment obtenim el resultat $K < 152$ bits.

7.
a) *Stop & Wait*: si l'enllaç és *half-duplex*, o si $t_t >> t_p$. En aquest cas el paràmetre $a = t_p / t_t \approx 0$ i l'*Stop & Wait* té un comportament semblant al *Go back N* i a la retransmissió selectiva; vegeu la figura "Avaluació del protocol *Stop & Wait*".

Vegeu la figura esmentada en el subapartat 3.3.4 d'aquest mòdul didàctic.

b) *Go back N*: si l'enllaç és *full-duplex*, no es compleix $t_t >> t_p$ i hi ha pocs errors. En aquest cas el *Go back N* i la retransmissió selectiva tenen un comportament semblant.
c) Retransmissió selectiva: si l'enllaç és *full-duplex*, no es compleix $t_t >> t_p$ i hi ha molts errors. En aquest cas la retransmissió selectiva té un comportament notablement millor que els altres protocols.

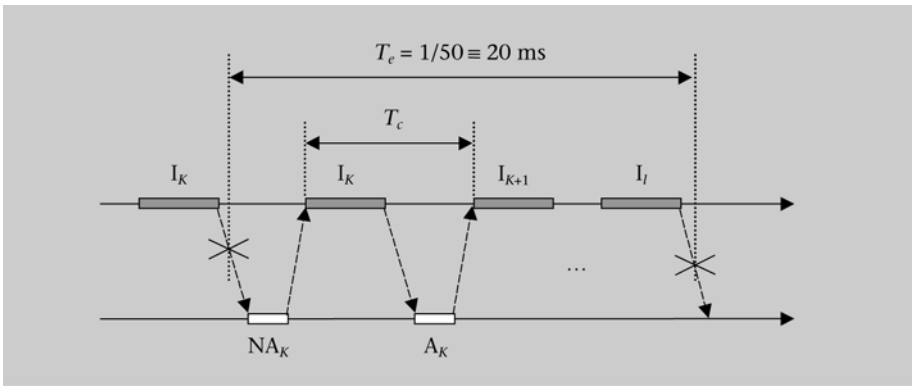
8.
a) Transmissió d'una trama amb èxit: igual que la figura de l'exercici 5.
b) $t_p = D/v_p = 10^3/2 \cdot 10^8 \equiv 5$ µs, $t_a = L_a/v_t = 100/10 \cdot 10^6 \equiv 10$ µs

Podem trobar una explicació del *time-out* amb la figura corresponent en el subapartat 3.3.1 d'aquest mòdul didàctic.

$$E = E_P \cdot E_t = \frac{t_t}{t_t + 2t_p + t_a} \cdot \frac{L}{L + 50} > 0{,}9 \ ,$$

on $t_t = (L + 50) / v_t$. Substituint, s'obté $L > 2.250$ bits.

9.
a)



b) $t_p = D/v_p = 600/2 \cdot 10^8 \equiv 3$ µs, $t_t = L/v_t = 1.500/10 \cdot 10^6 \equiv 150$ µs, $t_a = L_a/v_t = 40/10 \cdot 10^6 \equiv 4$ µs

$$E_P = \frac{t_t}{T_C} = \frac{t_t}{t_t + 2t_p + t_a} = \frac{150}{150 + 2 \cdot 3 + 4} \equiv 93{,}75\%$$

c) Aplicant l'equació 14:

$$E = \frac{t_t}{N_t(t_t + 2t_p + t_a)} = \frac{E_p}{N_t}$$

Per a calcular N_t hem de tenir en compte que cada T_e (20 ms) hi ha una trama que es transmet dues vegades. En T_e es poden transmetre T_e/T_C trames, on $T_c = t_t + 2\ t_p + t_a \equiv 160$ µs. Així, doncs:

$$T_e/T_C = 20 \cdot 10^3\ /\ 160 \equiv 125 \text{ transmissions,}$$

de les quals n'hi ha una de repetida. Es transmeten, doncs, 124 trames de cada 125 transmissions i, per tant: $N_t = 125/124$.

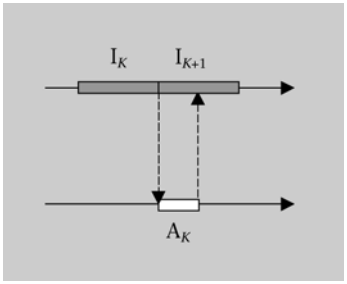
També es pot obtenir el mateix resultat amb el raonament següent: atès que de cada 125 transmissions n'hi ha una d'errònia, la probabilitat d'error en una trama és de $p_t = 1/125$. De l'equació 18 tenim que:

$$N_t = 1\ /\ (1 - p_t) = 1\ /\ (1 - 1\ /\ 125) = 125\ /\ 124$$

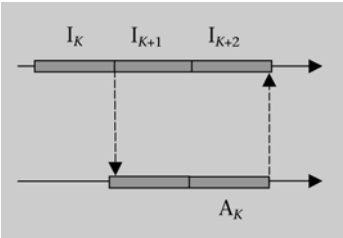
Substituint en la fórmula de l'eficiència anterior, s'obté:

$$E = \frac{E_p}{N_t} = \frac{0{,}9375}{125/124} \equiv 93\%$$

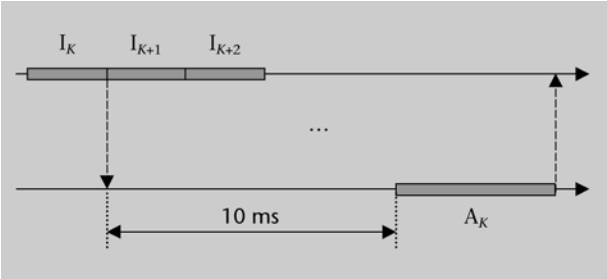
10.
a) El diagrama de temps és el que es veu al marge, d'on $W_{opt} = 2$.



b) Suposarem que les dues estacions transmeten contínuament. El cas més desfavorable és quan la trama d'informació I_K acabi d'arribar a l'estació remota just quan aquesta ha començat a transmetre una trama d'informació. La confirmació de IK anirà dins de la trama següent, com mostra el diagrama de temps que es veu a continuació. Així, doncs, $Wopt = 3$.



11.
a) El cas més desfavorable és el que mostra el diagrama de temps següent. Aquí les trames I_{K+1}, I_{K+2}, ... són trames d'1 kbit, i just quan s'esgoten els 100 ms d'inactivitat, l'estació remota decideix enviar una trama de 5 kbit.



El temps de transmissió d'una trama d'1 kbit és de $10^3/100 \cdot 10^3 \equiv 10$ ms. El d'una trama de 5 kbit és de $5 \cdot 10^3/100 \cdot 10^3 \equiv 50$ ms. Així, doncs:

$$W_{opt} = 1 + \lceil (100 + 50) / 10 \rceil = 16 \text{ trames}$$

b) Per a un protocol *Go back N* amb una finestra W i n bits per als identificadors s'ha de complir $W + 1 \leq 2^n$. Substituint $W = 16$ s'obté:

$$n \geq \log_2 17 = \log_{10} 17/\log_{10} 2 = 4{,}08, \text{ d'on } n \geq 5 \text{ bits}$$

12.
a) El diagrama de temps és el mateix que el de l'apartat *a* de l'exercici 7.
b) Influeixen en l'eficiència:

- Els bits d'arrencada, parada i paritat (E_{Tx})

$$E_{Tx} = \frac{\text{bits d'informació per caràcter}}{\text{bits transmesos per caràcter}} = \frac{8}{8+1+1+1} \equiv 66{,}7\%$$

- Els caràcters de control de les trames (E_t)

$$E_t = \frac{\text{caràcters d'informació per trama}}{\text{caràcters transmesos per trama}} = \frac{1.000}{1{,}006} \equiv 99{,}4\%$$

- El temps d'espera de les confirmacions (E_p)

$$E_p = \frac{t_t}{t_t + 2t_p + t_a} \equiv 98\%$$

- Les retransmissions degudes als errors (E_e)

$$E_e = \frac{1}{N_t} = 1 - p_t \approx 1 - L \cdot p_b = 1 - 1.006 \cdot 8 \cdot 10^{-5} \equiv 92\%$$

Per a calcular E_p s'ha fet servir el següent:

t_t = bits per trama / $v_t = 1.006 \cdot 8 / 115.200 \equiv 69{,}86$ ms
t_a = bits per confirmació / $v_t = 5 \cdot 8 / 115.200 \equiv 0{,}35$ ms
t_p = distància / $v_p = 10 \cdot 10^3 / 2 \cdot 10^8 \equiv 0{,}5$ ms

Per a calcular E_e s'ha tingut en compte el següent:

$$p_t = 1 - (1 - p_b)^L,$$

on L és el nombre de bits d'una trama. Atès que $p_b \cdot L \approx 0{,}08 << 1$, es pot aproximar per:

$$(1 - p_b)^L \approx 1 - p_b \cdot L = 1 - 10^{-5} \cdot 1.006 \cdot 8 = 0{,}92$$

L'eficiència total és el producte de totes les eficiències anteriors:

$$E_{tot} = E_{Tx} \cdot E_t \cdot E_p \cdot E_e = 0{,}667 \cdot 0{,}994 \cdot 0{,}98 \cdot 0{,}92 \equiv 59{,}77\%$$

c) $v_{ef} = E_{tot} \cdot v_t = 0{,}5977 \cdot 115.200 \equiv 68{,}8$ kbps

d) Temps de transmissió = mida del fitxer / $v_{ef} = 10^6 / 68.800 \equiv 14{,}5$ s

Glossari

codi detector d'errors *m* Transformació que s'aplica a la trama per obtenir el valor dels bits que s'hi han afegit per poder detectar si una trama arriba amb errors. Els codis més utilitzats són la paritat, la paritat longitudinal i transversal i els codis CRC.

control de flux *m* Mecanisme que fa servir el nivell d'enllaç per a evitar que les trames arribin més aviat del que és capaç de processar. Normalment es fa servir un mecanisme de finestra.

diagrama de temps *m* Representació espaciotemporal de la comunicació entre dues entitats. Permet representar gràficament un protocol i el temps invertit en cada una de les accions.

distància de Hamming *f* Nombre de bits de diferència que hi ha entre dues paraules codi.

enllaç multipunt *m* Situació en què una comunitat de computadors es comuniquen a través d'un enllaç compartit. Per a poder comunicar-se sense interferir-se entre ells n'hi ha un que s'encarrega d'arbitrar les comunicacions.

finestra de recepció *f* Nombre màxim de trames que ha d'emmagatzemar el secundari.

finestra de transmissió *f* Nombre de trames que el primari està autoritzat a transmetre sense confirmar, atès que totes les anteriors estan confirmades.

finestra òptima *f* Finestra mínima que permet assolir una eficiència del protocol del 100%.

framing *m* *Vegeu* sincronització de trama.

inserció de caràcters *f* Mecanisme de transparència que es fa servir en les transmissions orientades a caràcter.

inserció de zeros *f* Mecanisme de transparència que es fa servir en les transmissions orientades a bit.

llista de recepció *f* Llista en la qual el secundari guarda les trames per poder lliurar-les en ordre.

llista de transmissió *f* Llista on el primari guarda les trames que s'han enviat i que estan pendents de confirmar per poder-les retransmetre en cas d'error.

mecanisme de transparència *m* Mecanisme que es fa servir per a no confondre les dades de la trama amb l'identificador de final de trama.

nivell d'enllaç *m* Nivell que hi ha immediatament per sobre del nivell físic. La seva funció més important és transmetre informació de manera fiable fent servir el flux no fiable de bits del nivell físic.

paraula codi *f* En el context dels codis detectors d'errors, conjunt de $P + Q$ bits que resulten de la transformació que aplica el codi als P bits que s'han de protegir.

piggybacking *m* Tècnica que consisteix a enviar les confirmacions dins d'un camp de les trames d'informació.

primari *m* Estació que envia les trames d'informació. En el protocol HDLC, és l'estació que pot enviar comandes.

protocol *m* Format i conjunt de regles que es defineixen per a la comunicació entre dues entitats del mateix nivell en l'arquitectura de comunicacions.

protocol no orientat a la connexió *m* Protocol que no té fase d'inicialització ni de desconnexió.

protocol orientat a la connexió *m* Protocol que té una fase d'inicialització i una fase de desconnexió.

ratxa d'error *f* Nombre de bits que hi ha entre el primer bit erroni d'una trama i l'últim, ambdós inclosos.

recuperació automàtica d'errors *f* Tècnica que consisteix a sol·licitar la retransmissió d'una trama quan es detecta un error. Els protocols més coneguts que fan servir aquesta tècnica són l'*Stop & Wait*, el *Go back N* i la retransmissió selectiva.

redundància *f* Bits addicionals que afegeix el codi detector d'errors.

secundari *m* En el context del mòdul, estació que envia les trames de confirmació. En el protocol HDLC, és l'estació que envia les respostes.

sincronització de trama *f* Mecanisme que fa servir el nivell d'enllaç per a determinar l'inici i el final d'una trama dins del flux de bits o caràcters que arriba del nivell físic.
en *framing*

trama *f* Estructura en què el nivell d'enllaç organitza els camps de control i d'informació perquè puguin ser identificats dins del flux de bits del nivell físic.

velocitat de transmissió efectiva o eficaç *f* Velocitat mitjana a què es transmeten els bits d'informació. Es defineix com a $v_{ef} = E \cdot v_t$, on E és l'eficiència de l'enllaç, i v_t, la velocitat de transmissió màxima.
sigla: v_{ef}

Bibliografia

Benedetto, S.; Biglier, E.; Castellani, V. (1987). *Digital Transmission Theory*. Madrid: Prentice Hall.

Bertsekas, D.; Gallager, R. (1992). *Data Networks* (2a. ed.). Prentice Hall.

Stallings, W. (2000). *Comunicaciones y redes de computadores*. Englewood Cliffs: Prentice Hall.

Annex

Càlcul de l'eficiència dels protocols ARQ en presència d'errors

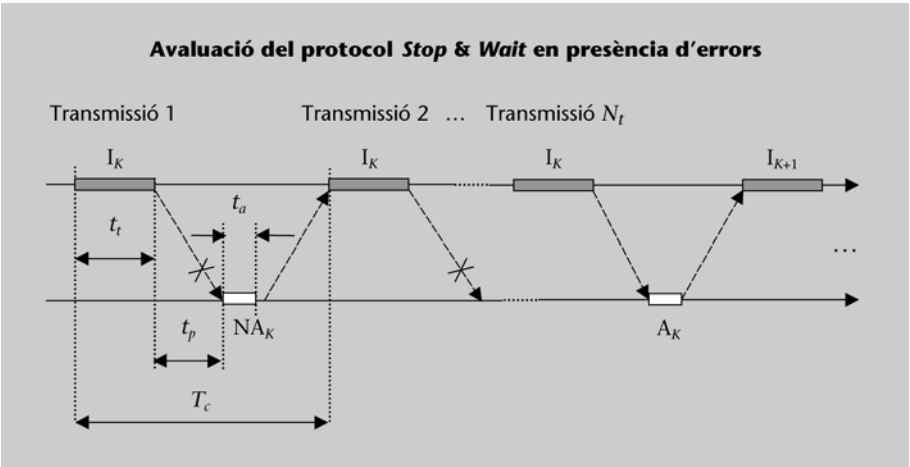
Hem vist que en absència d'errors l'eficiència del protocol *Stop & Wait* és determinada per l'equació 9, mentre que per als protocols de transmissió contínua és del 100%. Ara veurem com podem arribar a expressions tancades per a expressar aquesta eficiència suposant que es produeixen errors de transmissió.

Vegeu l'equació 9 en el subapartat 3.3.4 d'aquest mòdul didàctic.

Començarem suposant que coneixem el nombre mitjà de transmissions necessàries per a la transmissió amb èxit d'una trama (N_t).

1) Eficiència en el protocol *Stop & Wait*

Considerem el cas d'un protocol *Stop & Wait* explícit. El diagrama de temps que farem servir per a calcular l'eficiència és el següent:



La trama I_K es transmet N_t vegades, $N_t - 1$ amb error i l'última amb èxit. En la figura es veu que hi hem esmerçat un temps igual a $N_t \cdot T_c$ per poder transmetre-la.

Així, doncs, aplicant l'equació 6 arribem a aquesta expressió:

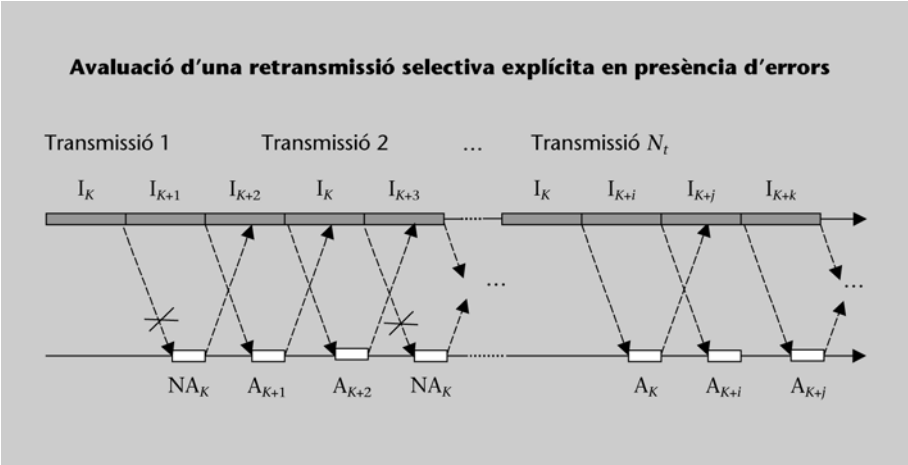
Equació 14

$$E = \frac{t_t}{N_t \cdot T_c} = \frac{t_t}{N_t(t_t + t_a + 2t_p)} \approx \frac{1}{N_t(1 + 2a)},$$

on $a = t_p/t_t$, i hem fet la suposició que $t_t >> t_a$.

2) Eficiència en una retransmissió selectiva

Considerem una retransmissió selectiva explícita. El diagrama de temps per a calcular l'eficiència és el de la figura, amb les mateixes suposicions que abans:



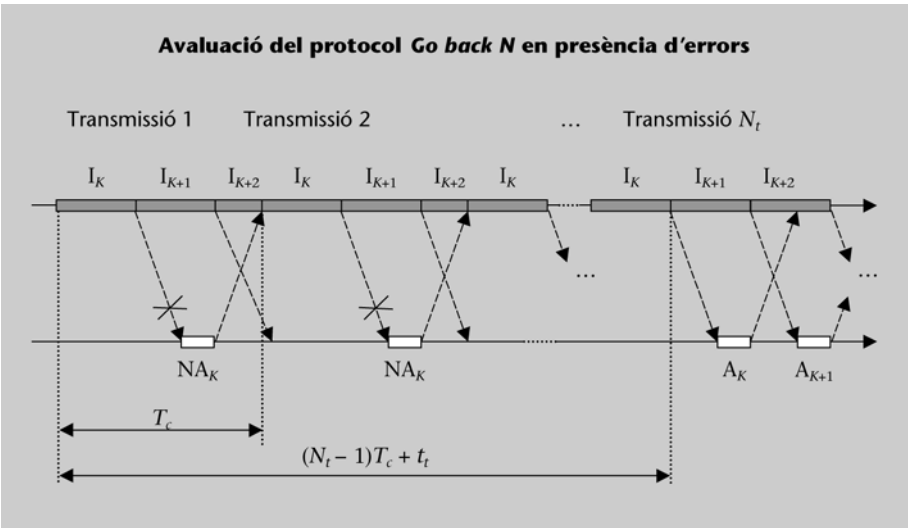
Aquí, el temps invertit en la transmissió és de $N_t \cdot t_t$, atès que el primari transmet altres trames mentre espera les confirmacions. Per tant:

Equació 15

$$E = \frac{t_t}{N_t \cdot t_t} = \frac{1}{N_t}$$

3) Eficiència en un protocol *Go back N*

Ara considerarem el diagrama de temps de la figura següent:



Tal com es veu en la figura, suposem que quan arriba una confirmació negativa s'avorta la transmissió de la trama en curs i es torna a començar a partir de la

trama errònia. Cada vegada que hi ha un error es perden les trames transmeses, tret del cas de l'última transmissió (que es fa amb èxit). Així, doncs, el temps invertit en la transmissió és:

$$N_t \cdot T_c - T_c + t_t = (N_t - 1) \cdot T_c + t_t$$

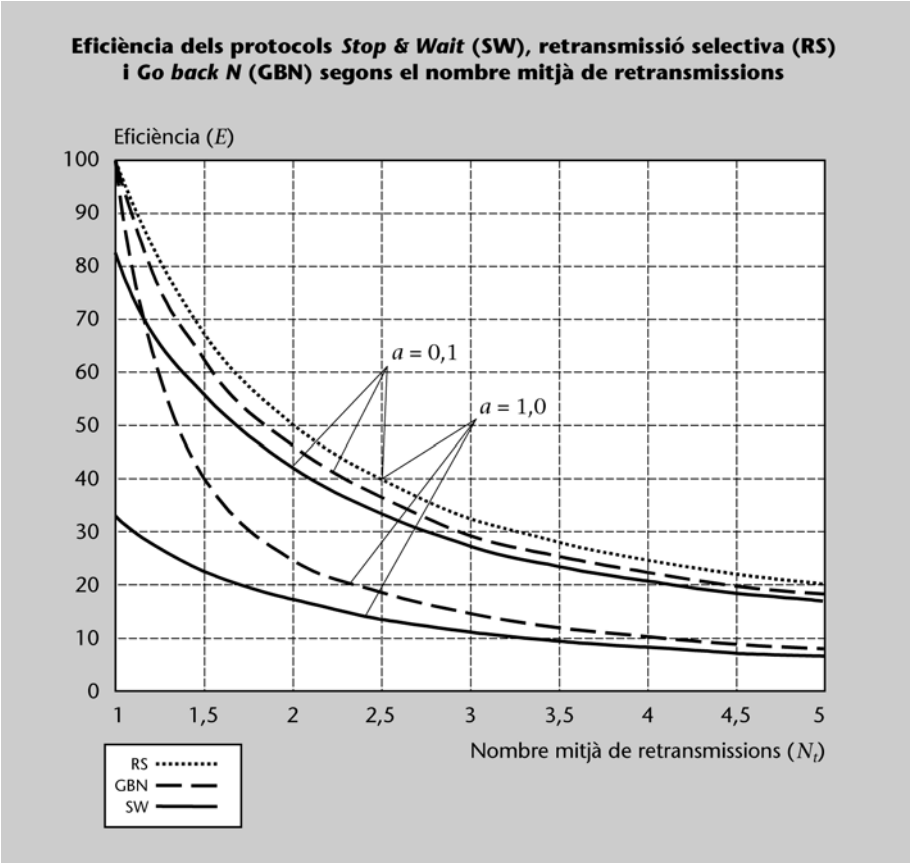
i, per tant:

Equació 16

$$E = \frac{t_t}{(N_t - 1) \cdot T_c + t_t} = \frac{t_t}{(N_t - 1)(t_t + t_a + 2t_p) + t_t} \approx \frac{1}{2a(N_t - 1) + N_t}$$

En aquesta equació s'ha fet la suposició habitual de considerar la hipòtesi $t_t >> t_a$. Si la probabilitat d'error en una trama és petita $\gtrapprox$, aleshores $N_t \to 1$, i l'equació 16 s'aproxima a $E \approx 1/N_t$, igual que en el cas de la retransmissió selectiva (equació 15). Aquest fet explica que en la pràctica, on la retransmissió de les trames sol ser poc freqüent, el protocol *Go back N* (GBN) és el mecanisme més utilitzat.

La figura següent mostra un **gràfic comparatiu de les eficiències** obtingudes amb els protocols *Stop & Wait* (SW), *Go back N* (GBN) i la retransmissió selectiva (RS) (equacions 14, 16 i 15, respectivament). Les corbes s'han dibuixat per a dos valors del paràmetre $a = t_p/t_t$ ($a = 0{,}1$ i $a = 1{,}0$):



De la figura anterior podem deduir les característiques més importants dels tres protocols de nivells d'enllaç explicats fins ara:

- L'eficiència amb retransmissió selectiva és independent de a.
- L'eficiència amb retransmissió selectiva i *Go back N* és del 100% si la probabilitat d'error en una trama és 0 ($N_t = 1$). En canvi, en el protocol *Stop & Wait* decreix ràpidament en augmentar a.
- Si la probabilitat d'error en una trama és petita ($N_t \gtrapprox 1$), la retransmissió selectiva i el *Go back N* tenen una eficiència semblant.
- Si la probabilitat d'error és alta ($N_t >> 1$), l'eficiència obtinguda amb els tres protocols és semblant.

Càlcul del nombre mitjà de retransmissions en funció de la probabilitat d'error en el bit

En les equacions 14, 15 i 16 hem donat el valor de l'eficiència en funció del nombre mitjà de retransmissions d'una trama (N_t). En la pràctica, és habitual disposar de la **probabilitat d'error en un bit** (***p_b***). Ara deduirem quina és la relació que hi ha entre p_b i N_t. Primer calcularem la **probabilitat d'error en la trama** (***p_t***) en funció de p_b. El nombre de bits de la trama l'anomenarem L. Suposant que la probabilitat d'error en els bits de la trama sigui independent, tenim que la probabilitat d'error en la trama és la que veiem a continuació:

$$p_t = 1 - \mathrm{P}\{\text{no error en la trama}\} = 1 - \mathrm{P}\{\text{no error en un bit}\}^L,$$

d'on:

Equació 17

$$p_t = 1 - (1 - p_b)^L$$

Atès que si $n \cdot x << 1$, podem fer el desenvolupament $(1 - x)^n \approx 1 - n \cdot x$, i l'equació 17 es pot aproximar per:

$$p_t \approx L \cdot p_b \text{ , si } L \cdot p_b << 1$$

Suposem a continuació que la probabilitat d'error en les trames és independent per a cada trama. Aleshores, si el nombre de retransmissions d'una trama és K, vol dir que hi ha hagut $K - 1$ trames amb error i una trama sense error. D'aquest fet es dedueix que la probabilitat que hi hagi K retransmissions és aquesta:

$$\mathrm{P}\{K \text{ retransmissions}\} = p_t^{K\text{-}1}\,(1 - p_t)$$

El nombre mitjà de retransmissions és donat per l'expressió següent:

$$N_t = \sum_{K \geq 1} K \cdot P\{K \text{ retransmissions}\}\,;$$

d'aquí:

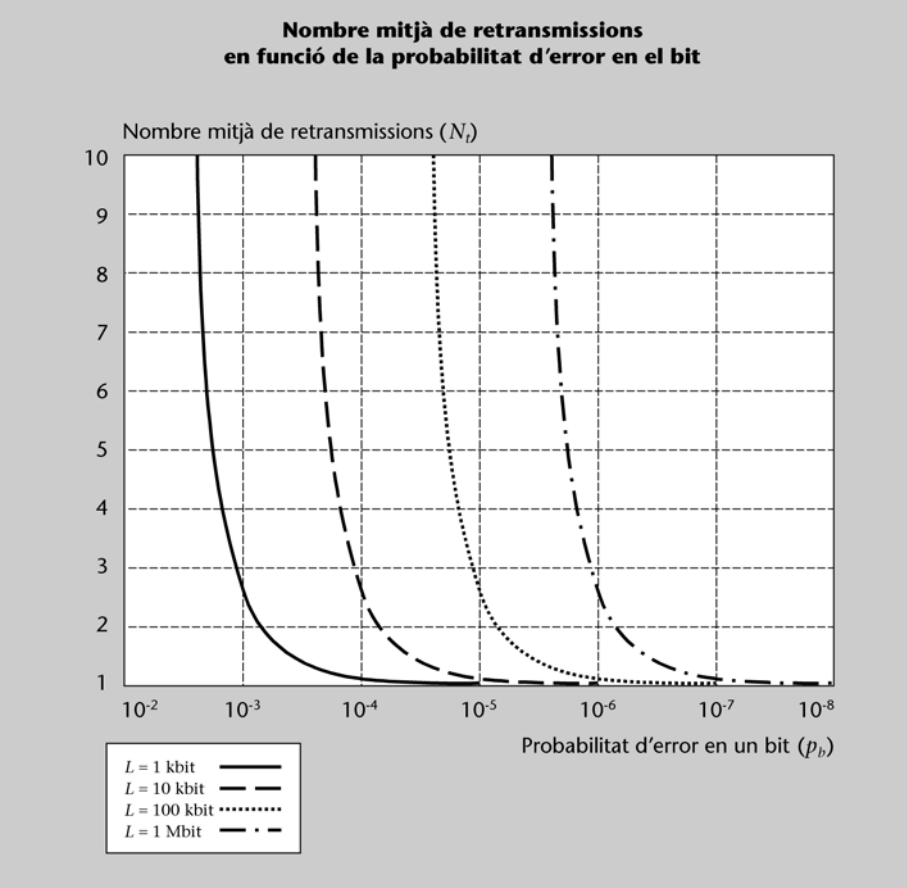
$$N_t = \sum_{K \geq 1} K \cdot p_t^{K-1}(1-p_t) = (1-p_t)/(1-p_t)^2 ,$$

on s'ha fet servir la relació $\sum_{n \geq 1} n \cdot x^{n-1} = 1/(1-x)^2$. Finalment, s'arriba a:

Equació 18

$$N_t = 1 / (1 - p_t)$$

Si substituïm l'equació 17 en l'equació 18, ens queda el nombre mitjà de retransmissions de cada trama en funció de la probabilitat d'error en el bit. La figura següent mostra un gràfic de l'equació resultant per a una mida de la trama L = 1 kbit, 10 kbit, 100 kbit i 1 Mbit:



Del gràfic deduïm la relació orientativa següent: per a tenir un nombre mitjà de retransmissions $N_t \approx 1$ fent servir trames de mida 10^x, la probabilitat d'error en el bit ha de ser $p_b < 10^{-(x+1)}$.

Com més gran sigui la mida de la trama més gran serà el nombre mitjà de retransmissions necessari per a transmetre-la i, per tant, menor serà l'eficiència de l'algorisme de recuperació d'errors. La probabilitat d'error en el bit limita la mida màxima de trama que podem fer servir perquè el nombre mitjà de retransmissions sigui aproximadament 1.