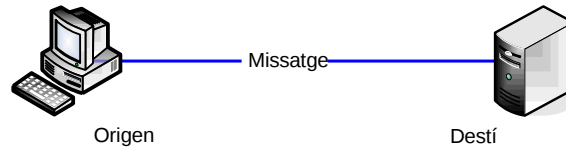


1.- Introducció.

Considerem dos equips que es transmeten informació digital, tal com mostra la figura adjunta. Tenint present que tota dada transmesa de forma digital es representa amb un conjunt de bits d'acord amb unes determinades regles que conformen el codi emprat en la comunicació, la informació transmesa la podem considerar com una seqüència de **paraules de codi** que constitueix el que anomenarem **missatge**.



Naturalment, per tal d'aconseguir que la màquina origen comuniqui la informació a la màquina destinatària cal:

- Transmetre la seqüència de paraules de codi que formen el missatge.
- Que els dos extrems interpretin de la mateixa forma cadascuna de les paraules de codi.

Això últim s'aconseguirà sempre i quan origen i destí utilitzin el mateix codi, és a dir, quan comparteixen el que anomenem la **codificació de font**¹. El que es vol indicar a través d'aquesta expressió és que el missatge a transmetre –la font– que té un determinat significat des del punt de vista del llenguatge humà necessita ser transformat a una seqüència de bits, és a dir, necessita ser codificat.

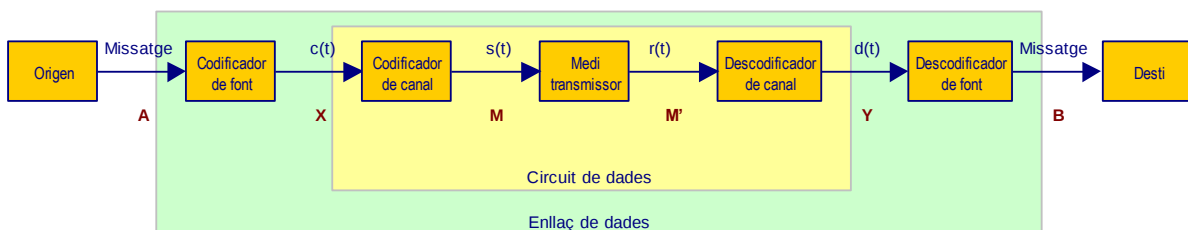
Des d'un punt de vista lògic, es pot modelar el procés de comunicació tal com s'observa en la figura. El missatge transmès per l'origen arribarà al destí a través d'un medi de transmissió –fil elèctric, fibra òptica, aire,... Com no es poden posar directament els bits en el medi, cal convertir-los en senyals electromagnètics, la qual cosa fa aparèixer la necessitat d'un element transmissor en la banda de l'origen de la informació i d'un element receptor en la banda de la destinació.



Sovint es considera que el transmissor té cura de dues qüestions:

- Convertir la informació a transmetre en una seqüència de bits d'acord amb la codificació de font emprada.
- Convertir aquesta seqüència de bits en els senyals adequats en funció del medi de transmissió utilitzat. Aquesta tasca s'anomena **codificació de canal**, ja que cal traduir cada bit d'informació en un senyal adient al canal físic emprat en la comunicació, i qui la porta a la pràctica rep, naturalment, el nom de codificador de canal.

El model complet seria el del gràfic següent. Els elements situats entre els punts **X** i **Y** formen part de l'anomenat **circuit de dades**. Aquest terme es fa servir quan es vol fer abstracció dels elements i es desitja considerar que aquest circuit rep una seqüència de bits per a lliurar-la al destí.



De forma anàloga, quan el que fan el codificador i el descodificador de font tampoc és important, es diu que el que hi ha entre **A** i **B** és l'**enllaç de dades**. Respecte la figura anterior cal remarcar dues qüestions:

- Amb la notació $s(t)$ i $r(t)$ es vol expressar el fet que el senyal rebut pot ser que no sigui exactament el transmès, la qual cosa s'explica per les pertorbacions que introdueix el medi de transmissió.

¹ Per exemple, si es transmet un fitxer de text, la codificació de font podria ser el conjunt de caràcters del codi ASCII. Si es transmet una imatge, es podria emprar el model RGB per a codificar el color d'un determinat punt.

- Anàlogament, pot ser que la descodificació obtinguda **-d(t)-** no coincideixi amb la codificació que es lliura al codificador de canal **-c(t)-**. Quan això succeeix, és perquè s'ha produït un error en la transmissió.

1.1.- L'enllaç de dades.

Dins el model d'arquitectura de comunicacions OSI², el segon nivell –per la banda baixa– que es contempla és l'anomenat **nivell d'enllaç**. La seva missió és controlar l'enllaç de dades. En aquest nivell resideix el programari que controla la transmissió d'informació entre els equips que conformen l'enllaç. Es tracta, bàsicament de controlar el flux de bits que es transmeten en el nivell físic.

Com es veurà en tractar del model OSI, igual que en els altres nivells, el nivell d'enllaç consisteix en un conjunt de programes que s'executen en computadors diferents i que es comuniquen entre si. Com s'ha esmentat anteriorment, per a que això sigui possible cal:

- Un format que permeti l'intercanvi d'informació.
- Un conjunt de regles –els **protocols**– que marquin el comportament dels interlocutors.

També s'ha esmentat anteriorment que el medi de transmissió introdueix errors en la transmissió. Per tant, una de les feines bàsiques del nivell d'enllaç és el **control d'errors**, la qual cosa implica disposar de mecanismes de detecció. Hem vist en el capítol dedicat a aquest tema que les dades associades a la detecció de l'errada s'incorporen al conjunt de la informació a transmetre, originant la unitat d'informació bàsica del nivell d'enllaç: la trama.

L'altra feina encarregada al nivell d'enllaç és el **control del flux**. Es tracta, bàsicament, d'evitar que el transmissor sobrecarregui la memòria temporal del receptor i permeti a aquest processar les trames a una velocitat adequada als recursos disponibles.

2.- El control del flux.

Com s'ha esmentat en l'apartat anterior, en un sistema de comunicacions el receptor sol reservar un espai de memòria temporal –un *buffer*– per tal d'emmagatzemar les dades que li arriben del transmissor abans de passar-les al nivell superior. També cal indicar que normalment l'emissor especeja el bloc de dades a transmetre en trames de grandària inferior al bloc per tot un conjunt de motius, entre els que podem esmentar:

- El *buffer* del receptor és limitat i pot ser de menys grandària que el missatge a transmetre.
- Com més llarga és una transmissió, més elevada és la probabilitat d'error i, cas de produir-se una errada, cal enviar novament totes les dades. Si la dimensió d'aquestes és elevada, l'enviament i reenviament ocuparan molt de temps.
- En entorns de xarxes d'àrea local cal evitar que una estació mantingui ocupat el mitjà molta estona.

El control del flux és una tècnica adreçada a assegurar que el transmissor no sobrecarregui el *buffer* del receptor, ja que si no existís aquest control es podria ultrapassar la capacitat d'aquesta memòria temporal i perdre alguna de les dades transmeses.



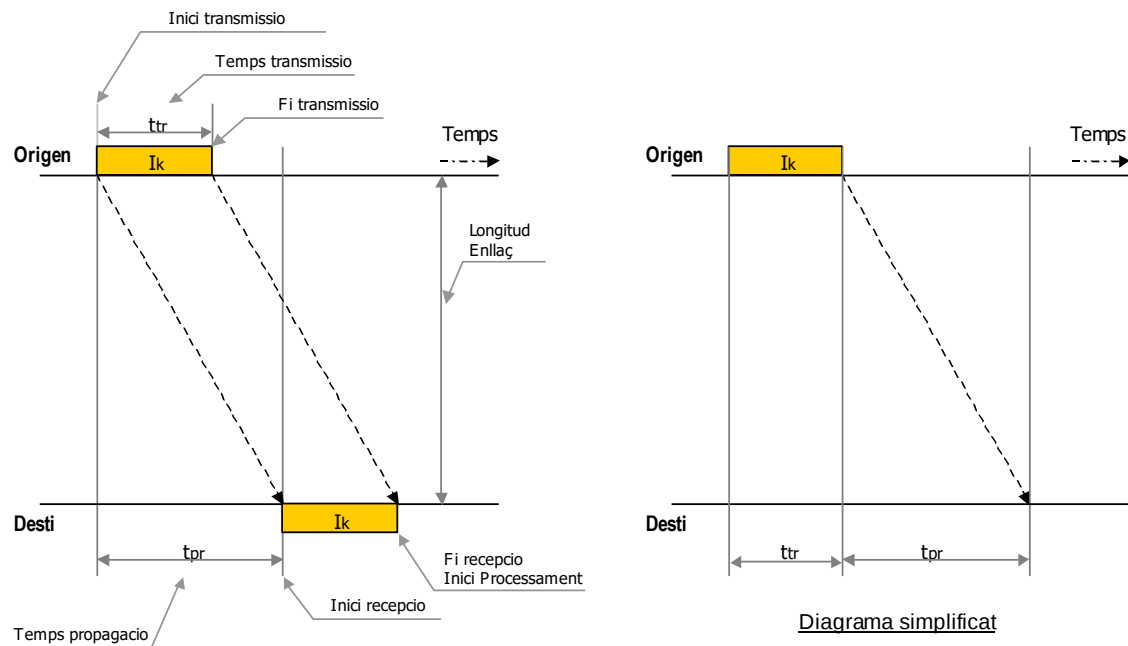
La figura mostra l'enllaç de dades entre un dispositiu transmissor i un de receptor, sistemes que es transmeten informació que, com ja s'ha esmentat, es pot considerar formada per un conjunt de trames que contenen tant les dades a transmetre com elements de control. Per tal de poder analitzar els diferents modes de control, s'usen models com els representats en la figura de la pàgina següent, on es pot observar la distribució temporal dels distints esdeveniments associats a la trama: la transmissió, la propagació i el processament.

D'acord amb la figura, si la velocitat de transmissió, la mida de trama i la distància de l'enllaç són fixes, tenim:

$$\text{○ Temps de transmissió: } t_{tr}(s) = \frac{\text{Mida trama (bit)}}{\text{velocitat transmissió (bps)}} = \frac{L \text{ (bit)}}{v_{tr} \text{ (bps)}}$$

² El model OSI es tractarà en el capítol **Arquitectura de Protocols**.

- **Temps de propagació:** $t_{pr}(s) = \frac{\text{Longitud enllaç (m)}}{\text{velocitat propagació (m/s)}} = \frac{D (m)}{v_{pr} (m/s)}$



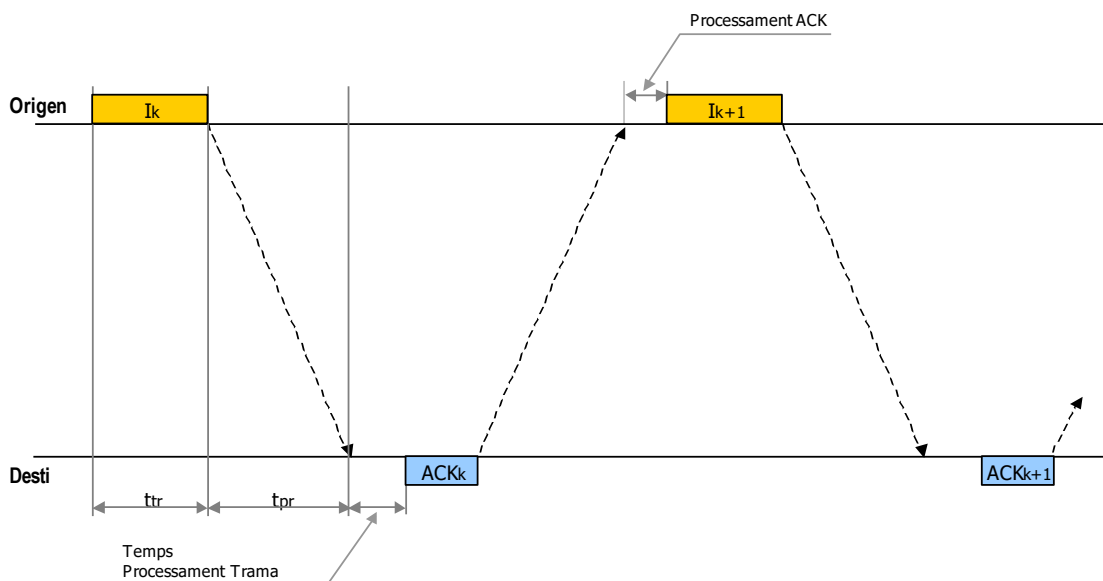
Un altre concepte a definir és l'anomenada **eficiència** de l'enllaç. Es defineix com la relació entre la velocitat mitjana a que es transmeten les dades i la velocitat màxima a que es podrien transmetre. És fàcil comprovar que es pot expressar de la forma següent:

$$E = \frac{\text{Temps transmetent informació}}{\text{Temps total transmissió}} = \frac{\text{Bits d'informació transmesos}}{\text{Total bits que es podien haver transmes}}$$

2.1.- Control del flux per aturada i espera –*Stop and wait*.

Aquest sistema de control, representat en la figura opera d'acord amb els següents principis:

- L'emissor transmet una trama i espera.
- El receptor rep la trama i la processa



- Si el receptor vol que el transmissor li envii una altra trama, ho comunica adreçant-li una **trama de confirmació** –*ACKnowledge*.

Per tant, per tal de controlar el flux el que ha de fer el receptor és retenir les confirmacions. El sistema presenta, a més les següents característiques:

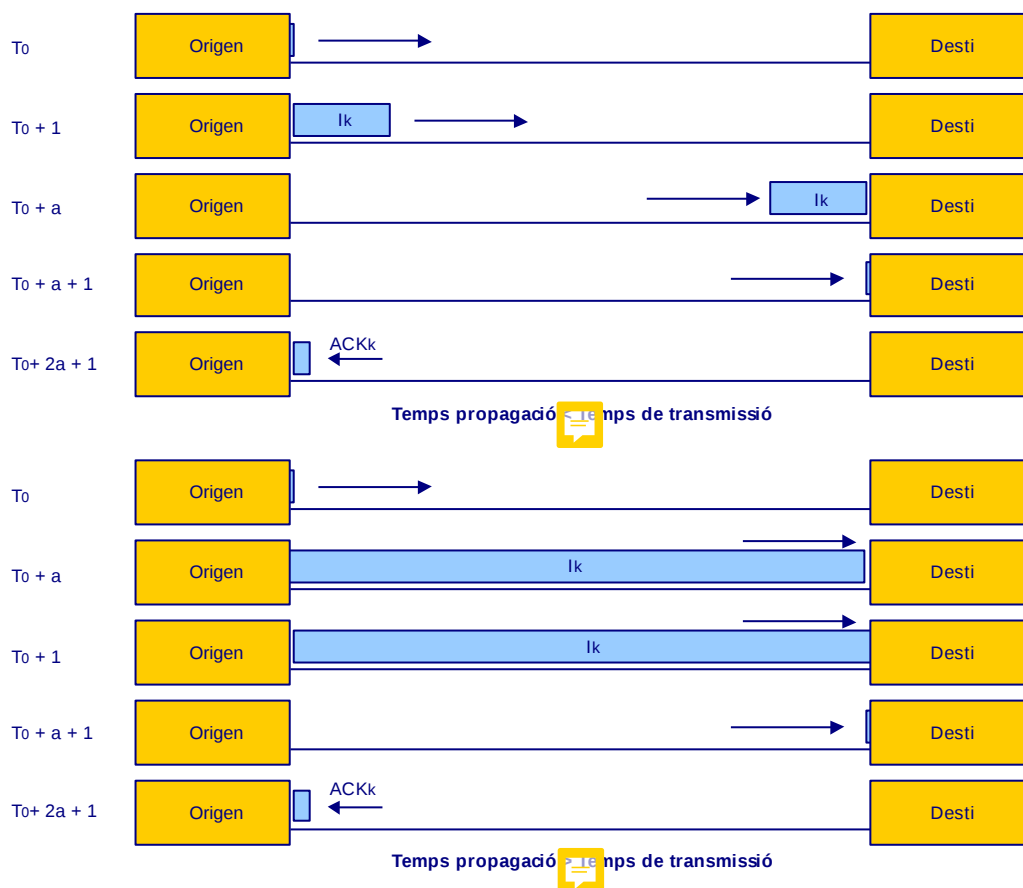
- Circula una única trama per l'enllaç.
- Com s'acaba d'indicar, es controla el flux a partir de les confirmacions
- Presenta un bon rendiment quan el temps de propagació és molt baix comparat amb el de transmissió. Podríem dir que s'envia un nombre reduït de trames de mida elevada –en comparança amb la longitud de l'enllaç.
- És ineficient quan la aquesta longitud és superior a la de la trama.

Per a confirmar les dues darreres conclusions, ens fixarem en què d'acord amb el diagrama anterior, si considerem negligibles els temps de processament i el de transmissió de la confirmació, l'eficiència vindria donada per:

$$E = \frac{\text{Temps transmetent informació}}{\text{Temps total transmissió}} = \frac{t_{tr}}{t_{tr} + 2t_{pr}} = \frac{1}{1 + 2t_{pr}/t_{tr}} = \frac{1}{1 + 2a}$$

on **a** és la relació existent entre el temps de propagació i el de transmissió d'una trama, i es pot interpretar com el nombre de trames que caben dins l'enllaç.

D'acord amb l'expressió anterior, si prenem el temps de transmissió de trama com a unitat de temps, és clar que el temps de propagació valdrà **a**. La figura adjunta mostra què succeeix en els casos en que **a > 1** –temps de transmissió inferior al de propagació– i quan **a < 1** –cas contrari.



En el primer cas, la línia es troba infrutilitzada, la qual cosa es pot comprovar si suposem el següent cas:

- Enllaç WAN entre dues estacions per fibra òptica emprant ATM amb les següents característiques:
 - Distància: 1.000 km

- Longitud de trama: 424 bit
- Velocitat de transmissió: 155,52 Mbps
- Velocitat de propagació en l'enllaç: 300.000 km/s

D'acord amb aquestes dades, tenim:

$$E = \frac{1}{1+2a} = \frac{1}{1+2t_{pr}/t_{tr}} = \frac{1}{1+2 \cdot 10^6 \text{ m} / 3 \cdot 10^8 \text{ m/s} / 424 \text{ bit} / 155,52 \text{ Mbps}} = \frac{1}{1+2445,28} = 0,0004088$$

En canvi, si es considera el següent supòsit:

- o Enllaç per parell trenat entre dues estacions en una LAN:
 - Distància: 100 m
 - Longitud de trama: 1000 bit
 - Velocitat de transmissió: 10 Mbps
 - Velocitat de propagació en l'enllaç: 300.000 km/s

D'acord amb aquestes dades, tenim:

$$E = \frac{1}{1+2a} = \frac{1}{1+2t_{pr}/t_{tr}} = \frac{1}{1+2 \cdot 10^2 \text{ m} / 3 \cdot 10^8 \text{ m/s} / 1000 \text{ bit} / 10 \text{ Mbps}} = \frac{1}{1+0,00667} = 0,9934$$

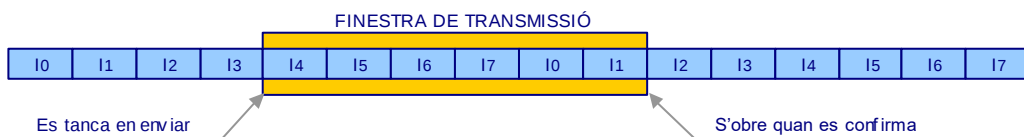
s'obté una eficiència propera al 100%. Observem que per a que aquest sistema sigui eficient cal que **a** sigui molt petita –**a** << 1–, és a dir, que el temps de propagació sigui molt petit respecte al de transmissió, la qual cosa és difícil d'aconseguir en enllaços a gran distància.

2.2.- Control del flux per finestra lliscant –*Sliding window*.

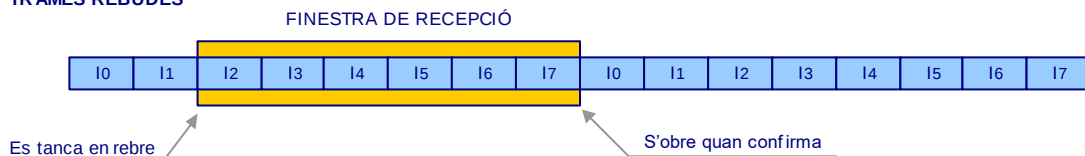
Tal com s'acaba de veure, el control per aturada i espera pot arribar a ser molt ineficient ja que l'estació origen no pot tornar a transmetre fins que rep la confirmació a la trama prèviament enviada. Com a alternativa, es presenta un sistema de transmissió contínua: el control per finestra lliscant, anomenat *sliding window* en anglès. Aquest sistema es basa en fer transitar alhora diverses trames d'informació per l'enllaç, sense esperar a rebre confirmació immediata per a cadascuna d'elles. Actua d'acord amb els següents principis:

- o El receptor disposa de la memòria suficient per a emmagatzemar **n** trames, per la qual cosa pot acceptar-les sense necessitat de confirmar-les una a una.
- o Per aquest motiu, el transmissor pot enviar **n** trames sense que calgui rebre confirmació immediata per a cadascuna.
- o Per tal de poder controlar les confirmacions –saber quines trames han estat reconegudes–, cadascuna s'etiqueta amb un número seguint un ordre seqüencial. Si s'utilitzen **k** bits –en la figura, k=3– per a emmagatzemar el número de trama, aquestes es numeren en mòdul **2^k**.
- o El receptor confirma una trama –i les anteriors que no havien estat confirmades– enviant un senyal de confirmació –**RRi**, *Receive Ready*– que inclou el número de la següent trama que espera rebre, la qual cosa indica també a l'origen que està preparat per rebre noves trames.

TRAMES TRANSMESSES

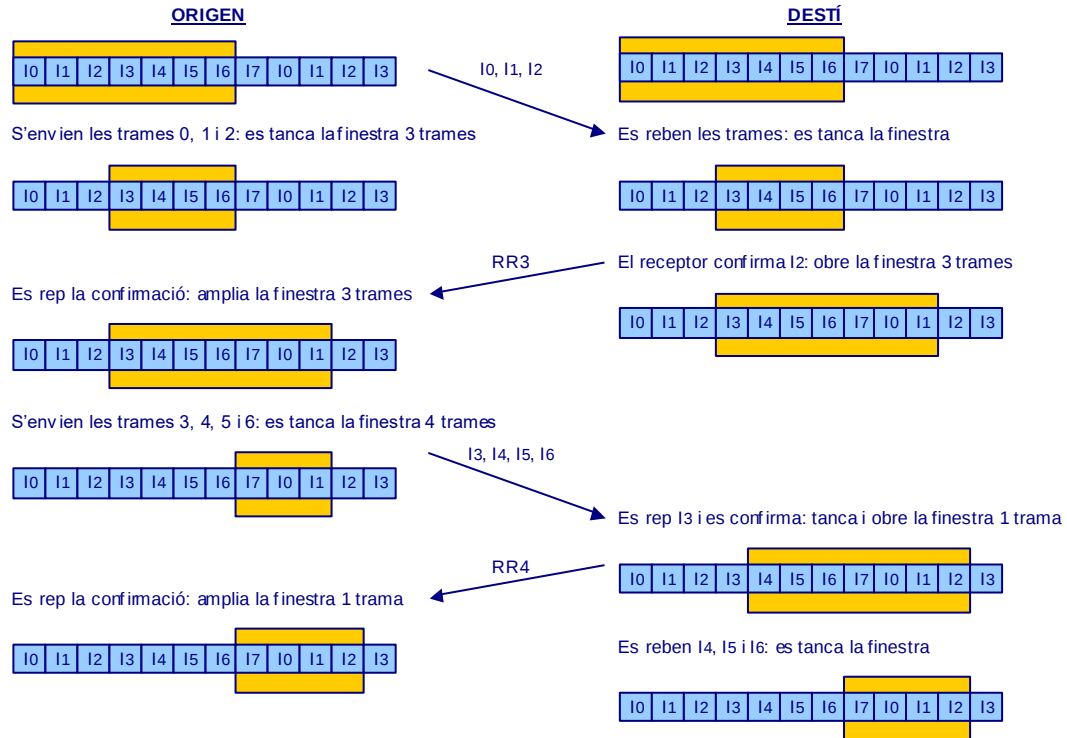


TRAMES REBUDES



Per tal de poder controlar tots aquests fets, els dos interlocutors mantenen, cadascun d'ells, una llista: el transmissor amb totes les trames que pot transmetre, el receptor amb tots els números de trama que espera rebre. Aquestes dues llistes s'anomenen finestres de trames i, conforme es va transmetent i rebent, van canviant de lloc i grandària. D'aquí el nom de **finestra lliscant**.

La figura mostra un exemple del funcionament del sistema en el cas en que $k=3$ i la longitud màxima de finestra és de 7 trames.



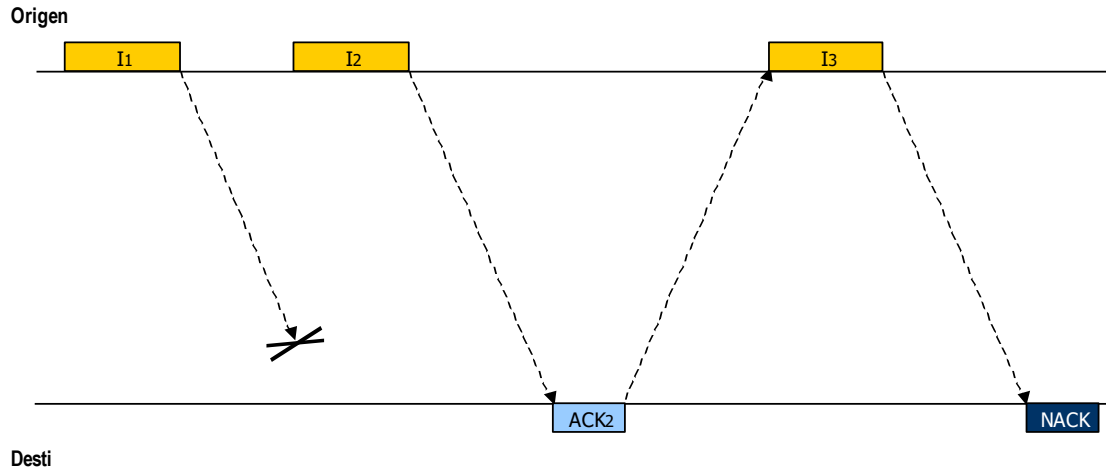
A més de tot el que s'ha esmentat, cal indicar el següent:

- Els protocols que utilitzen aquest mètode permeten que una estació pugui tallar momentàniament la transmissió enviant una trama de confirmació del tipus *Receive Not Ready* –**RNRi**. Mitjançant aquesta trama es confirmen les anteriors i s'indica a l'origen que no envii més trames. Així, **RNR5** confirmaria que s'ha rebut bé fins la trama 4 i que es vol suspendre momentàniament la transmissió.
- Per a desbloquejar la transmissió s'envia un *Receive Ready*. En l'exemple anterior, atraçaríem a l'origen un **RR5**.
- Quan es treballa en modalitat *full-duplex* –cosa normal si es desitja poder enviar confirmacions mentre es transmeten trames d'informació–, cada estació ha de mantenir dues finestres, una per a la recepció i l'altra per a la transmissió. Per tal d'aprofitar els canals, es solen incorporar les confirmacions en camps addicionals de les trames que contenen informació.

3.- El control d'errors.

El control d'errors tracta de com detectar i corregir els errors que apareixen en la transmissió de les trames. En principi, tal com mostra la figura següent, els errors són de dos tipus:

- Els que corresponen a trames que es perden, és a dir, aquelles que no arriben al receptor –cas de la trama 1.
- Els causats per les trames danyades, que són aquelles que no arriben al receptor tal com les ha enviat el transmissor –trama 3.



En cas de rebre una trama amb errors, el nivell d'enllaç pot actuar d'acord amb alguna de les següents polítiques:

- **Descartar la trama**, la qual cosa només és possible en els casos en què l'aplicació tolera un determinat nivell d'errades.
- **Intentar corregir l'error**, per la qual cosa fa falta disposar del codi corrector d'errors adequat. Tal com s'ha esmentat en el tema dedicat a la detecció d'errors, aquest mètode inclou molta redundància en les trames transmeses i és força ineficient.
- **Demandar la seva retransmissió**. Aquesta solució, que és la més emprada, necessita que l'enllaç sigui bidireccional i treballa segons les pautes següents:
 - Els dos interlocutors han d'estar d'**acord** en el mètode **detector d'errors** a emprar: BCC, CRC,...
 - El receptor ha de **confirmar positivament** la informació que rep quan aquesta és correcta.
 - El transmissor ha de **tornar a transmetre les trames no confirmades** un cop s'ha **exhaurit** un determinat **temps d'espera**.
 - Si el receptor rep una **trama errònia**, pot enviar una **confirmació negativa** a la que el transmissor respon **reenviant la trama**.
- Les tècniques esmentades en els darrers paràgrafs s'identifiquen normalment amb el mnemònic **ARQ** –*Automatic Repeat Request*–, és a dir, **Requeriment de Repetició Automàtica**, i n'existeixen tres variants normalitzats:
 - ARQ per aturada i espera –ARQ *stop & wait*.
 - ARQ endavant-endarrere-N –ARQ *go back N*.
 - ARQ amb retransmissió selectiva –ARQ *select reject*.

3.1.- ARQ per aturada i espera.

Com és fàcil suposar, aquesta variant està relacionada amb el control del flux anomenat també per aturada i espera, i es basa en no transmetre cap trama si l'anterior no ha estat confirmada positivament. Per tal d'aconseguir-ho, es disposa de dos tipus de trames:

- les d'informació,
- i les de confirmació,

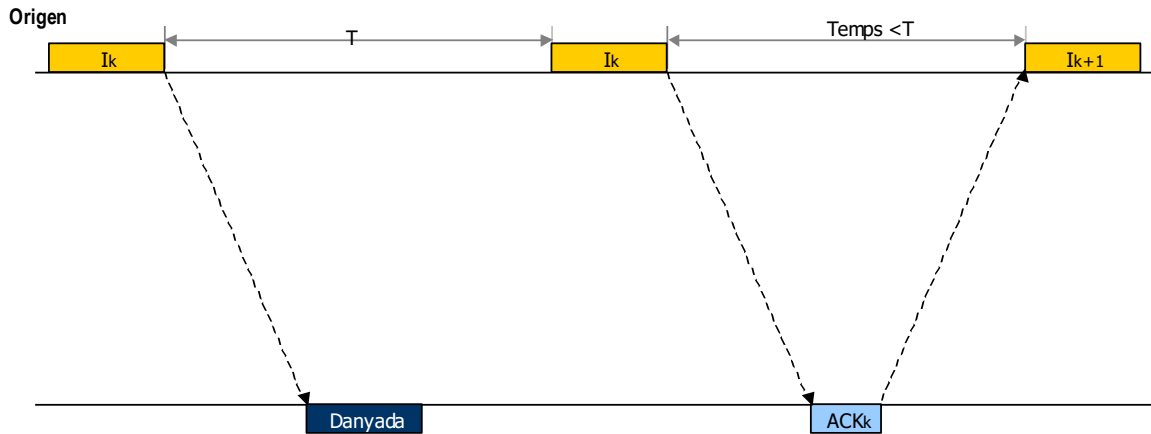
treballant el protocol d'acord amb les pautes següents:

- L'emissor transmet una trama i es posa a l'espera fins rebre la confirmació de la recepció correcta.
- Mentrestant, posa en marxa un control de temps –*time-out*.
- Si el temps s'exhaureix i no s'ha rebut confirmació, el transmissor torna a enviar la trama i activa novament el temporitzador.

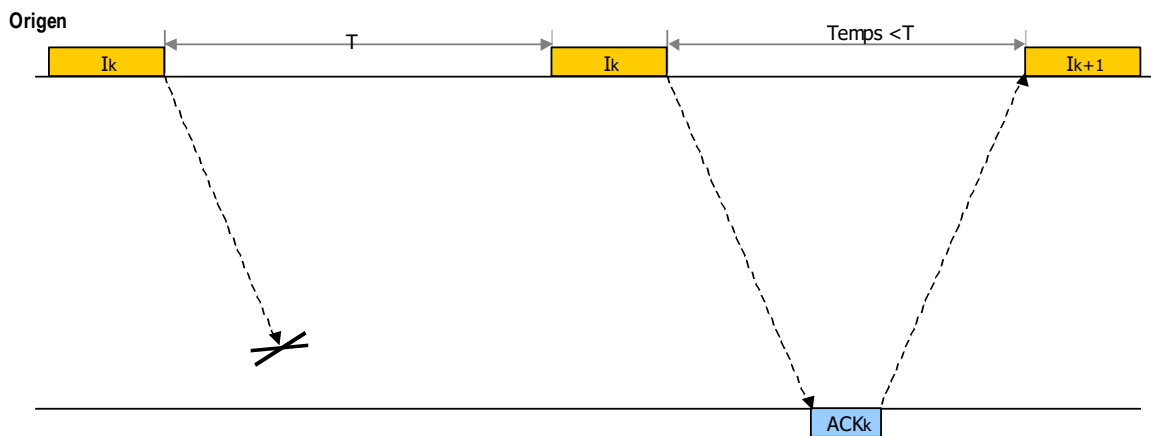
- Si el receptor rep la trama i la considera correcta, envia al transmissor una trama de confirmació, la qual cosa fa que aquest desactivi el temporitzador i iniciï el procés de transmissió d'una nova trama si encara n'hi ha per enviar.

Cal fixar-se en què el receptor no envia res en cas de considerar que la trama és incorrecta. Ara bé, aquestes pautes fan possible detectar qualsevol problema en el sentit de la transmissió emissor-receptor. En efecte, l'error es pot produir per dos motius:

- **Trama que arriba amb danys:** en aquest cas, el receptor ho detecta, descarta la trama i no envia la confirmació. En no rebre-la, el transmissor reenvia la trama un cop ha acabat el temps d'espera.
- **Trama perduda:** en no rebre la trama, el receptor no envia la confirmació, per la qual cosa el transmissor reenviarà la trama quan s'exhaureixi el temps.

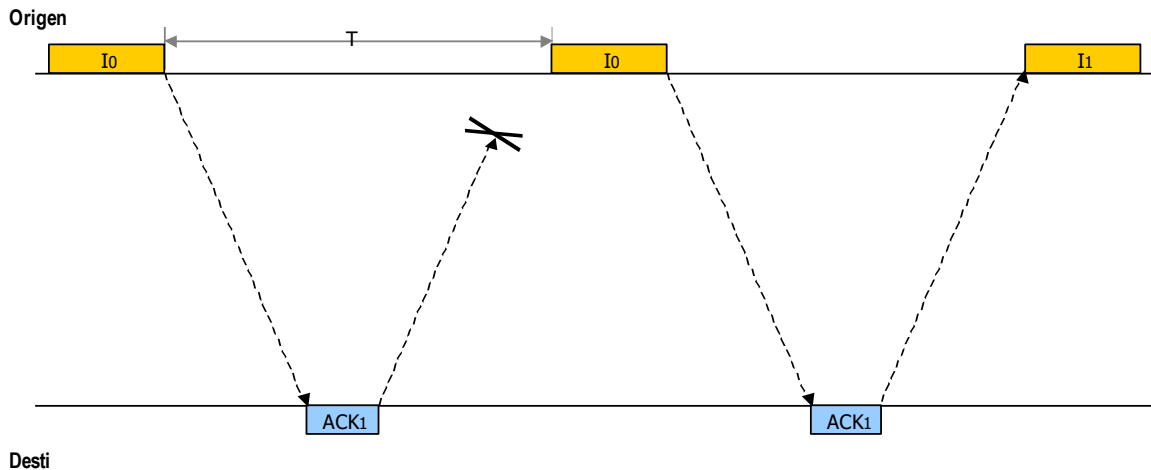


Desti



Desti

L'error pot afectar al senyal de confirmació, ja que aquest també es pot perdre o deteriorar. En aquest cas, quan s'acabi el temps de time-out, el transmissor tornarà a enviar la trama, amb la qual cosa el receptor en tindrà dues iguals.



Per tal d'evitar aquesta duplicat, tant les trames com les confirmacions es numeren, d'acord amb el següent:

- Les trames es numeren alternativament amb els nombres **0** i **1**.
- Les confirmacions es designen, també alternativament, com **ACK₀** i **ACK₁**, de forma que **ACK₁** confirma la recepció de la trama **I₀** –demana la trama **I₁**, i **ACK₀** confirma la recepció de la trama **I₁** –indica que vol la següent **I₀**.

Aquesta variant del protocol s'anomena *stop & wait* amb retransmissions implícites. N'hi ha una altra: *stop & wait* amb retransmissions explícites. En aquest cas, l'emissor envia una confirmació negativa –*Negative Acknowledge*– que detecta una trama errònia. Aquest fet permet guanyar temps ja que no cal esperar que s'acabi el temps de *time-out*. No evita, però, la utilització del temporitzador ja que sempre es poden perdre trames.

3.2.- ARQ endavant-endarrere-N.

Aquest sistema utilitza el control del flux per finestra lliscant i es basa en les següents pautes:

- Com en el cas anterior, cada vegada que es transmet una trama se li associa un control de temps.
- Si no es produeix cap errada, el receptor va enviant trames de confirmació positives, *Receive Ready* –**RR_j**– i *Receive Not Ready* –**RNR_j**–, per tal de controlar el flux. Mitjançant aquestes trames es confirma una trama concreta i totes les anteriors encara per confirmar.
- Si es detecta algun error de trama o bé es rep una trama fora de seqüència, el receptor envia una trama de confirmació negativa, anomenada *Reject* –**REJ_j**–, en què rebutja específicament la trama que conté l'error o sol·licita la trama que falta.
- A l'hora, el receptor descarta la trama errònia i totes les que la segueixen fins que torna a rebre correctament la trama malmesa.
- Per aquest motiu, quan el transmissor rep un **REJ_j** ha de tornar a transmetre la trama errònia i totes les que havia enviat posteriorment.

D'aquí el nom del sistema: la finestra es pot desplaçar tant endavant –si tot va bé– com enrere –quan hi ha errors– en la seqüència de trames a transmetre. A més, per tal de poder controlar les confirmacions, el transmissor ha de mantenir una llista amb les trames pendents de confirmació.

Tot seguit s'analitzaran una sèrie de casos per a veure com reacciona el sistema davant possibles anomalies. En primer lloc, suposarem que arriba al receptor una trama **I_k** i s'hi detecta un error, havent rebut prèviament la trama **I_{k-1}** de forma correcta.

En aquest cas, es produeixen els següents fets:

- En detectar l'error, el receptor envia un **REJ_k**.
- En rebre'l, l'emissor torna a transmetre la trama **I_k** i les que havia enviat posteriorment.

Si el que succeeix és que una trama **I_k** es perd, s'actua com s'indica tot seguit:

- En principi, cal tenir present que el transmissor pot haver enviat la trama següent **I_{k+1}**.

- Quan el receptor rep aquesta trama, com esperava rebre I_k , envia un **REJk** reclamant la trama perduda.
- Per tant, quan l'emissor el rep torna a transmetre la trama I_k i les posteriors ja enviades.

En el cas que una trama I_k es perdi i el transmissor no n'envii d'addicionals, i s'hagin confirmat les trames anteriors, succeirà el següent:

- Com el receptor no ha rebut la trama, no envia ni el **RRk+1** ni el **REJk**.
- Per tant, al cap del temps fixat el temporitzador de control del transmissor es dispara. Aquest fet causa que el transmissor transmeti una trama especial **RR** amb l'anomenat bit de sondeig – *polling*– actiu: **RR(P=1)**. Mitjançant aquesta trama, li demana al receptor que li doni alguna resposta.
- Aquest respon a aquest fet amb una trama **RRk**, amb la qual cosa el transmissor retransmet la trama I_k .

Finalment, esmentarem que l'amplada de la finestra en aquest tipus de ARQ és de 2^k-1 trames, essent k el nombre de bits emprat per a numerar-les.

3.3.- ARQ amb retransmissió selectiva.

Com l'ARQ anterior, aquest sistema fa servir el control del flux per finestra lliscant. En aquest cas, però, l'amplada de la finestra és igual a 2^{k-1} trames.

El sistema s'anomena de retransmissió selectiva perquè només es retransmeten les trames que expressament es rebutgen, la qual cosa es fa a través de la transmissió d'una trama de supervisió anomenada *Selective Reject* – **SREJ**, i aquelles per a les que s'exhaureix el temps de *time-out*.

Bibliografia.

Llibres:

Íñigo, J.; [et al.] (2004) *Estructura de xarxes de computadores*. (1a ed.) UOC

Stallings, W. (1998) *Comunicaciones y redes de computadores*. (5a ed.) Prentice Hall.

Pàgines web:

William Stallings. <http://williamstallings.com/>