

Considerazioni...



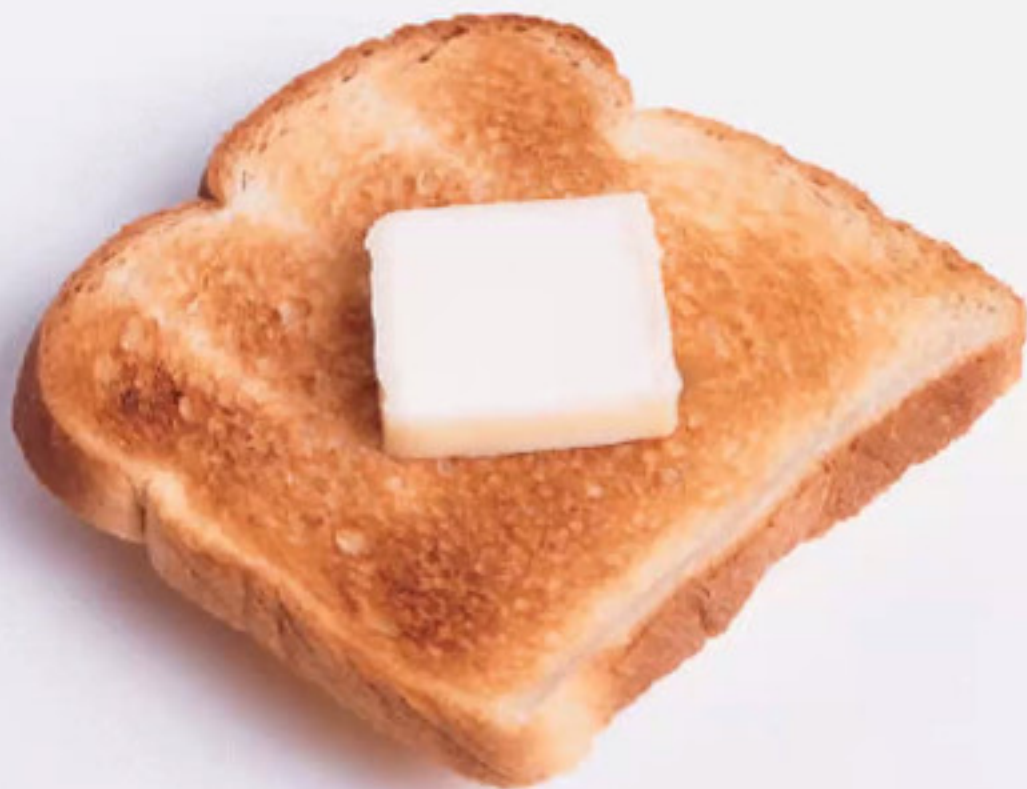
- ◆ Non è solo sfortuna...!
- ◆ Murphy ***si può quantificare!!***
- ◆ Pensate ai doppi errori nelle RAM, ai codici di errore, carte di credito etc



La sfortuna non esiste...?



??



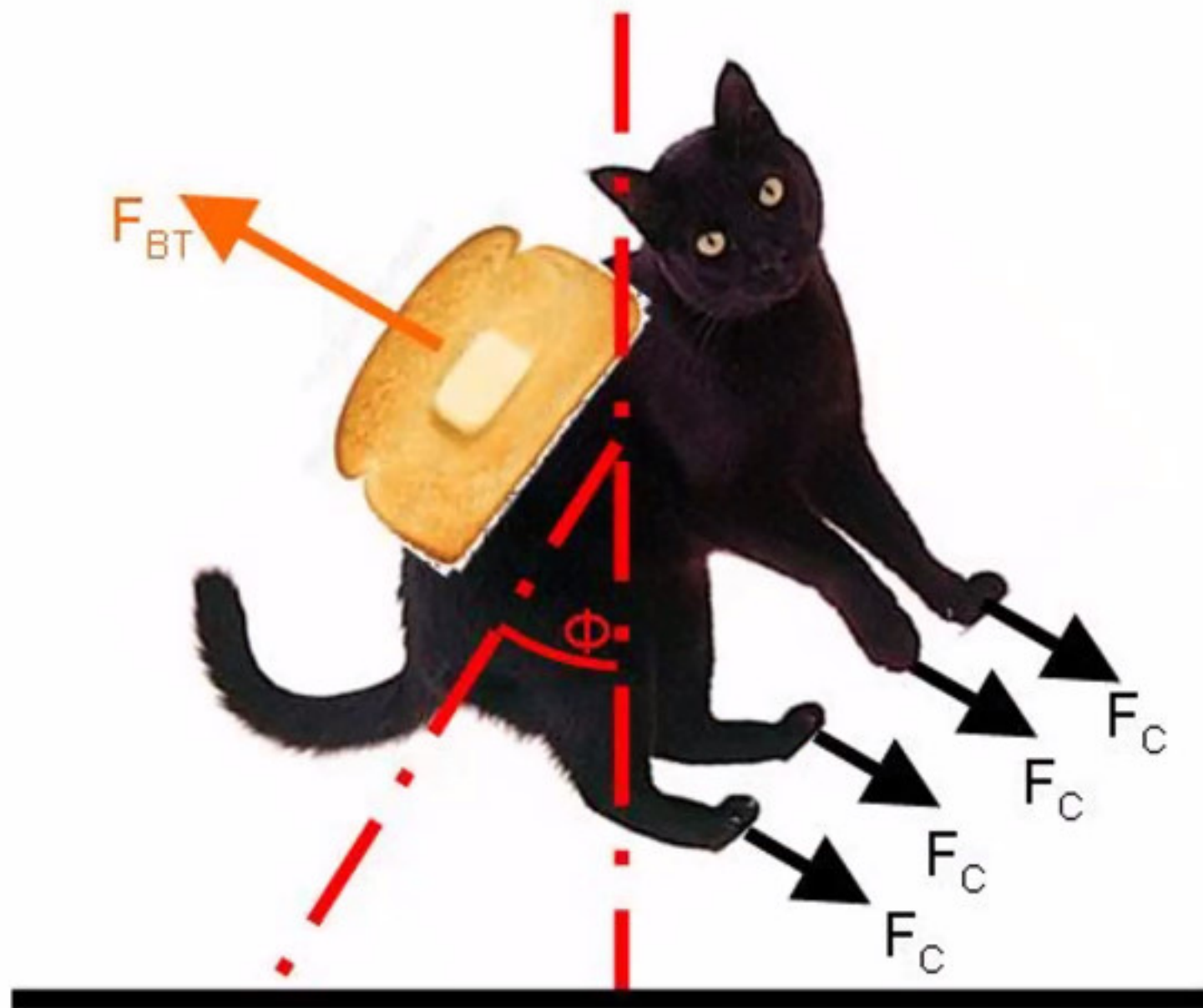


Fig. 1

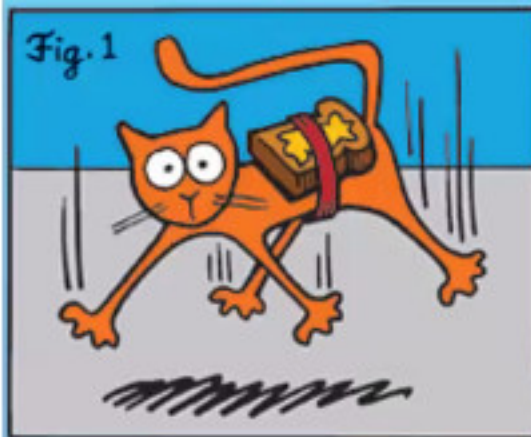


Fig. 2

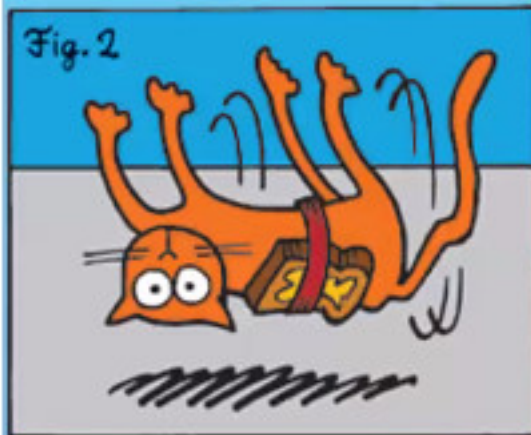
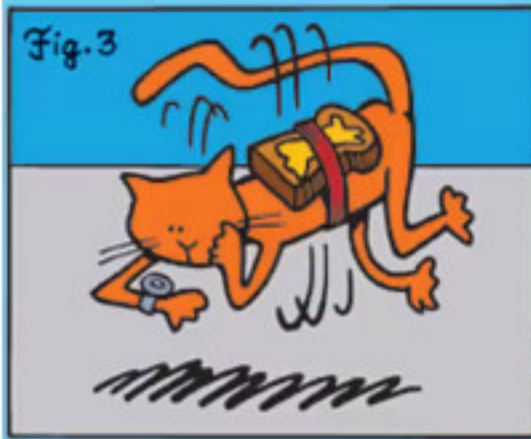


Fig. 3



Qui...



- ◆ Spiegazione informale/semplicata:
- ◆ Il problema delle reti (informatiche), rispetto a Murphy, è la loro velocità...!
- ◆ Per quanto la probabilità di un evento isolato sia bassa (quasi 0), quindi ***Murphy-like...***
- ◆ ... quando lo stesso evento si ripete per molte volte, la probabilità che accada sale rapidamente a 1 (certezza)

Quindi...

- ◆ Visto che il progresso tecnologico ci porta a reti sempre più veloci...
- ◆ ... → gli eventi improbabili di Murphy ***sono sempre più probabili !!!***



Altri Problemi delle Sliding Windows

- ◆ Se vi ricordate, abbiamo detto che quando si fa piggybacking c'è un problema quando il traffico è ***sbilanciato***: rischiamo di aspettare troppo per mandare l'**ack**



Soluzione?

- ◆ Ci sono comunque delle soluzioni: ad esempio, possiamo avere un altro ***timer supplementare*** anche nel ***receiver***. quando il timer scade, rimandiamo, anche senza piggybacking



Però...

- ◆ Occorre stare attenti, ovviamente, a settare bene la durata di questo timer: in particolare, dovrebbe essere più breve del timer usato per ritrasmettere i frame (!)



I timer del sender

- ◆ Gli altri timer da considerare sono quelli del sender: quale scelta fare?
- ◆ Se il tempo di ack è poco variabile, allora ci basta settare il timer a un pò di più del tempo medio di attesa
- ◆ Il problema è se il tempo di ack è invece ***molto variabile*** (!)



Che fare?

- ◆ Ci sono vari modi per scamparla, vediamone uno:
- ◆ Quando il tempo è variabile, si setta il timer del sender a un valore abbastanza largo, però si dà un'altra arma al receiver



II NAK!

- ◆ Il receiver misura il rischio che ci siano stati errori, e se questo rischio è troppo alto, si può sollecitare il sender, inviandogli un messaggio speciale:
- ◆ non un ACK, ma un **NAK (Not AcK)**

NAK

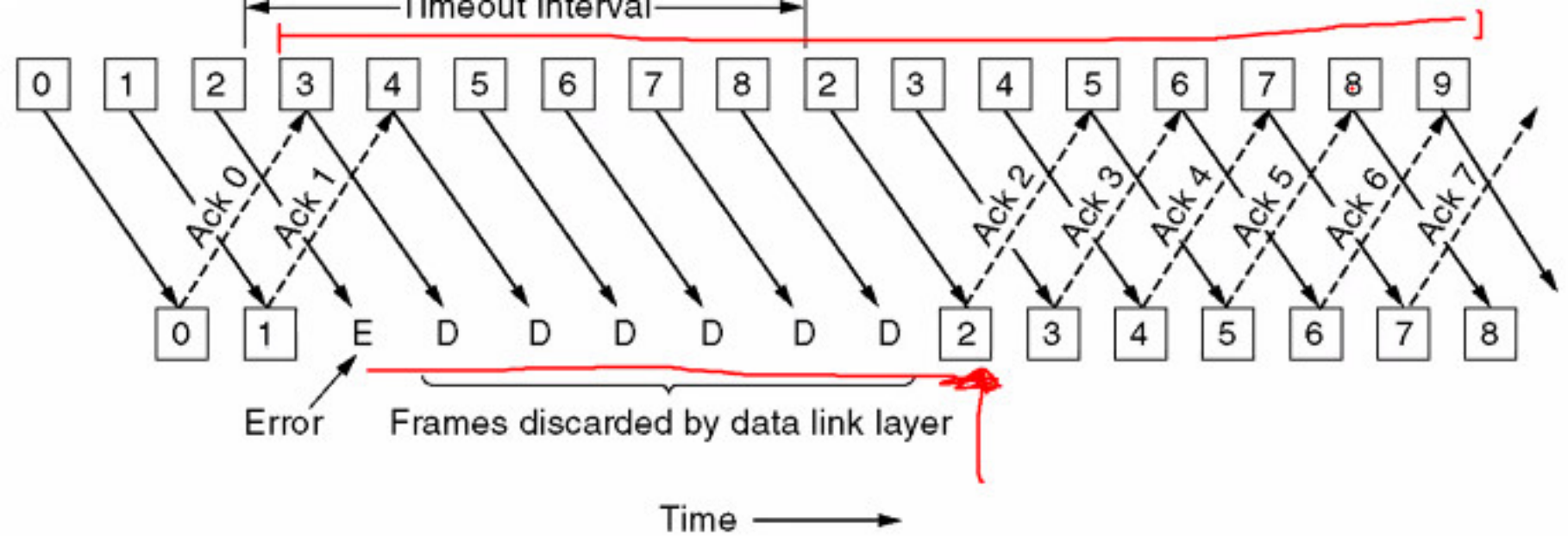
- ◆ Il NAK quindi è un avviso che per qualche motivo, non è stato ricevuto il pacchetto corrispondente
- ◆ In questo modo, il sender sa che non serve aspettare col suo timeout generoso: può ritrasmettere subito!
- ◆ → Un buon uso del NAK può velocizzare moltissimo una rete!



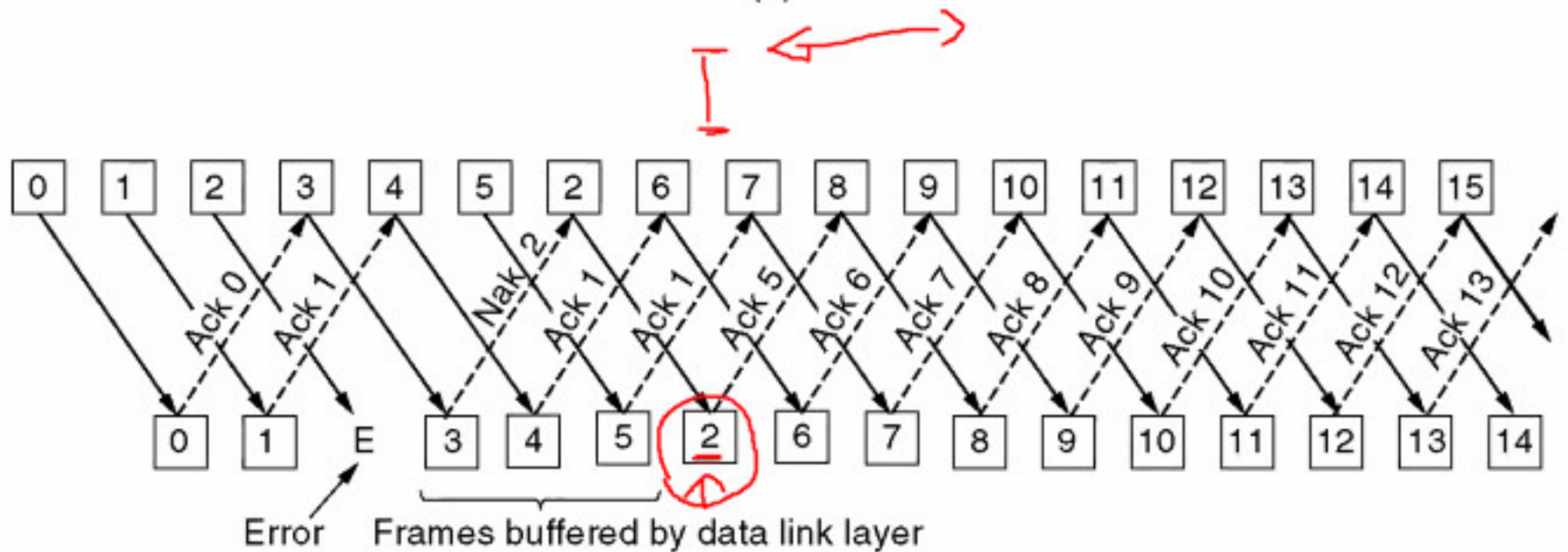
Quando si attiva un NAK?

- ◆ Il NAK si può attivare tramite uno speciale ***timer nak***, per ogni slot della sliding window, che tenga conto dell'aspettativa di ricevere il pacchetto col numero corrispondente
- ◆ Oppure anche in altre condizioni: come ad esempio ricevere un pacchetto "fuori flusso"...





(a)





La struttura del frame (stuffing)

Bits

8

8

8

≥ 0

16

8

0 1 1 1 1 1 1 0	Address	Control	Data	Checksum	0 1 1 1 1 1 1 0
-----------------	---------	---------	------	----------	-----------------

Bits

8

8

8

0 1 1 1 1 1 1 0	Address	Control
-----------------	---------	---------

≥ 0

16

8

Data	Checksum	0 1 1 1 1 1 1 0
------	----------	-----------------

Vediamo meglio le componenti

- ◆ La parte **Data** è il *payload*, i dati effettivi
- ◆ La parte **Checksum** è calcolata usando CRC

≥ 0

16

8

Data

Checksum

0 1 1 1 1 1 1 0

Il Frame HDLC

- ◆ La parte di **Address** serve per la componente di indirizzamento, se ci sono terminali multipli dentro la stessa rete che devono essere distinti

8

8

8

0 1 1 1 1 1 1 0	Address	Control
-----------------	---------	---------

Il Frame HDLC

- ◆ La parte di Control è quella più interessante

8

8

8

0 1 1 1 1 1 1 0

Address

Control

Control del Frame

◆ Essenzialmente, ci possono essere tre tipi di Frame:

◆ Information

◆ Supervisory

◆ Unnumbered

Control del Frame

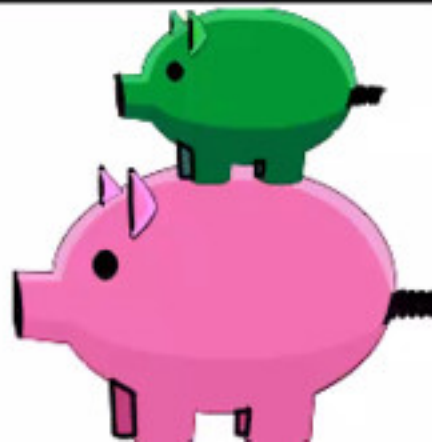
- ◆ Il controllo di flusso avviene tramite una ***sliding window*** di grandezza massima **3 bit** (8 "spicchi")



Il Frame **Information**

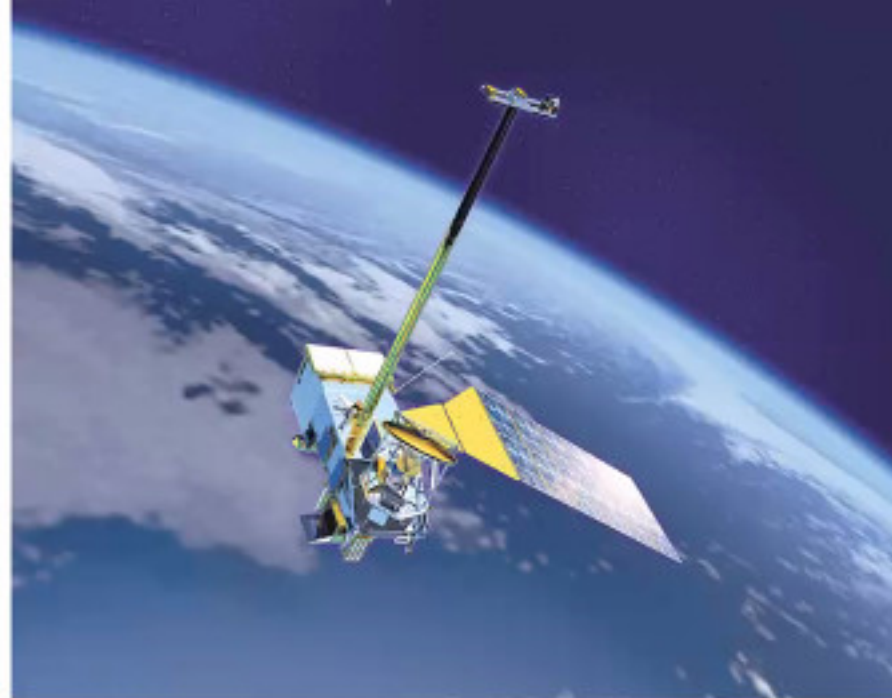
- ◆ **Seq** contiene il numero del controllo di flusso della *sliding window*

Bits	1	3	1	3
(a)	0	Seq	P/F	Next



Notare

- ◆ C'è anche una variante dove il controllo di flusso si fa con una sliding window grande **128** "spicchi", usata per le comunicazioni satellitari



Frame Information (cont.)

- ◆ P/F sta per **P**oll/**F**inal:
- ◆ Quando il bit indica **P** (Poll), si chiede al ricevente di iniziare la trasmissione
- ◆ Quando il bit indica **F** (Final) si segnala che la trasmissione va conclusa

Bits

(a)

1	3	1	3
0	Seq	P/F	Next

Supervisory (cont.)

Type 0 : ACK (in questo protocollo, detto **RECEIVE READY**)

si usa quando il flusso è ***sbilanciato*** e non si può fare ACK con piggybacking (ricordate il timer extra)



	0	Type	P/F	Next	
--	---	------	-----	------	--

Supervisory (cont.)

◆ **Type 1 : REJECT**

- ◆ E' un **NAK** generalizzato, segnala che vanno ritrasmessi tutti i frame a partire da quello indicato in poi nella sliding window
- ◆ Qui **Next** indica il primo frame



(b)

1	0	Type	P/F	Next
---	---	------	-----	------



Supervisory (cont.)

◆ **Type 2 : RECEIVE NOT READY**

- ◆ Questo è qualcosa di concettualmente nuovo: segnala che ci sono problemi di congestione nel receiver, e quindi la trasmissione va bloccata, finchè il receiver non rimanda un ACK

(b)

1	0	Type	P/F	Next
---	---	------	-----	------

Supervisory (cont.)



◆ **Type 3 :**
SELECTIVE REJECT

◆ Questo è il classico **NAK**

(b)

1	0	Type	P/F	Next
---	---	------	-----	------

Il Frame Unnumbered

- ◆ Usato per ulteriori comandi di controllo, non vediamo tutti i dettagli perché le implementazioni nella famiglia HDLC variano, vediamo solo qualche comando

(c)

1	1	Type	P/F	Modifier
---	---	------	-----	----------

I comandi di Unnumbered

- ◆ **DISC:** sta per ***DISConnect***, segnala che la macchina sta uscendo dalla rete in maniera definitiva (quindi, diverso dal frame Supervisory di tipo 2, che segnalava un problema temporaneo)



I comandi di Unnumbered

- ◆ **SNRM**: è il comando duale, segnala che una nuova macchina è entrata nella rete
- ◆ SNRM = **S**et **N**ormal **R**esponse **M**ode
→ indica un canale asimmetrico, dove il nuovo entrato è meno importante
- ◆ Retaggio storico, quando c'era sempre e solo un server centrale e terminali locali molto stupidi



I comandi di Unnumbered

◆ **SABM: Set Asynchronous Balanced Mode**

- ◆ E' il comando (introdotto successivamente nello standard) che invece crea una connessione bilanciata, dove chi entra ha gli stessi diritti degli altri