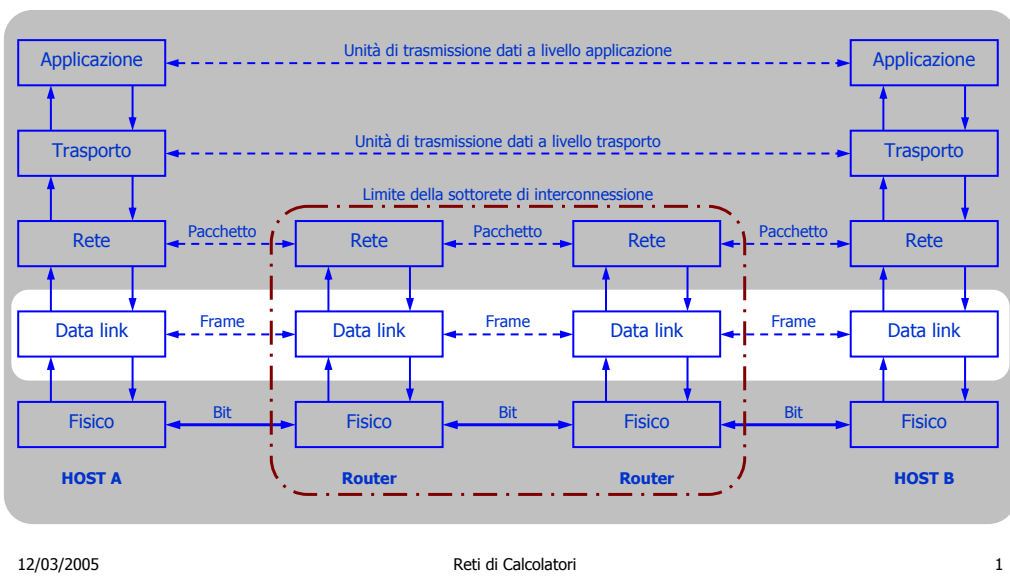


Livello Data Link



Data Link Layer Design Issues

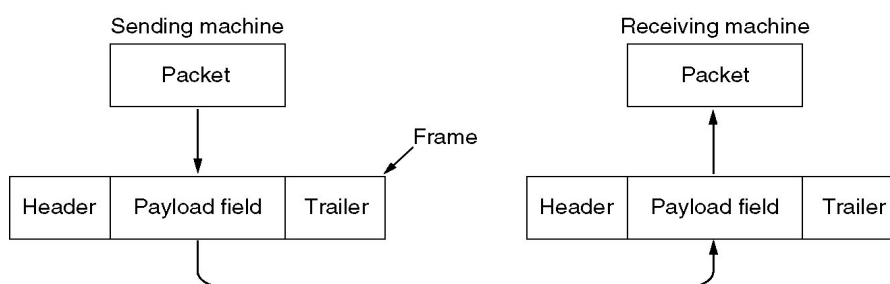
[Tanenbaum – Computer Networks - Chapter 3]

- Services Provided to the Network Layer
 - Framing
 - Error Control
 - Flow Control
- Functions of the Data Link Layer
 - Provide service interface to the network layer
 - Dealing with transmission errors
 - Regulating data flow
 - Slow receivers not swamped by fast senders

Functions of the Data Link Layer

[Tanenbaum – Computer Networks - Chapter 3]

Relationship between packets and frames.



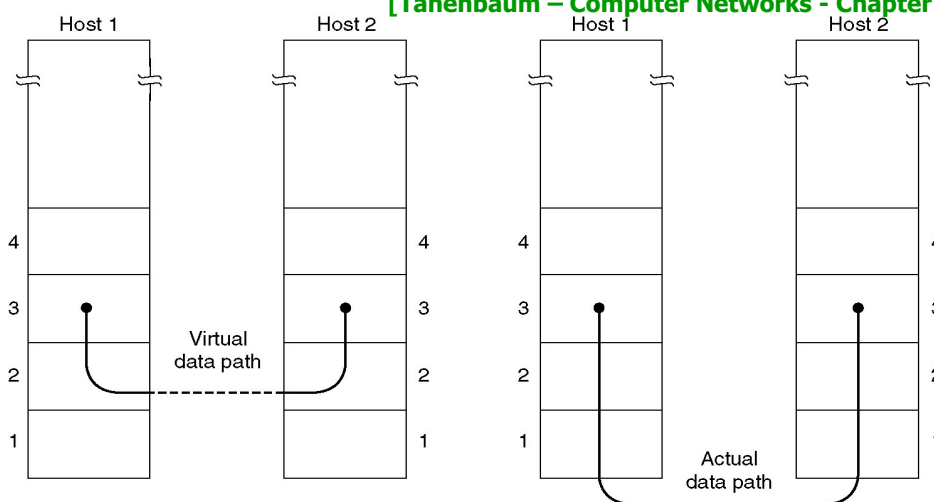
12/03/2005

Reti di Calcolatori

3

Services Provided to Network Layer

[Tanenbaum – Computer Networks - Chapter 3]



Virtual

communication

Actual

communication

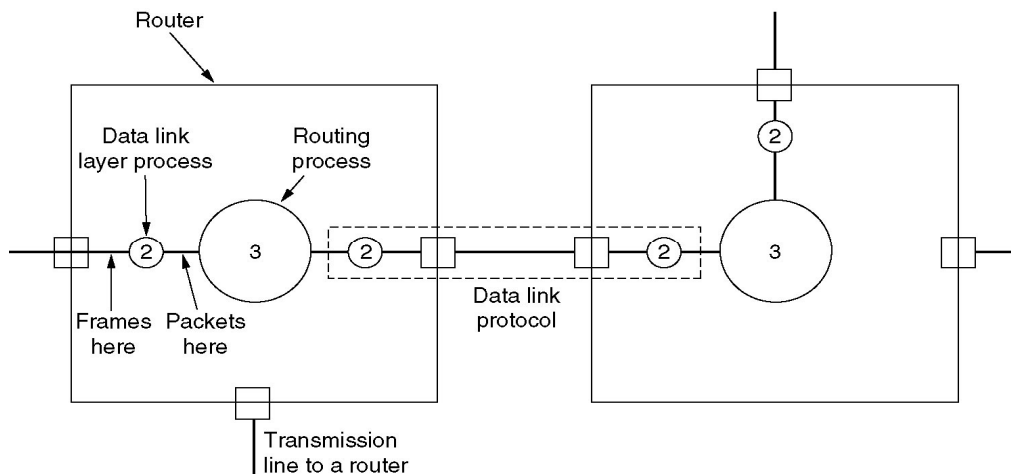
12/03/2005

Reti di Calcolatori

4

Services Provided to Network Layer

[Tanenbaum – Computer Networks - Chapter 3]



Placement of the data link protocol.

12/03/2005

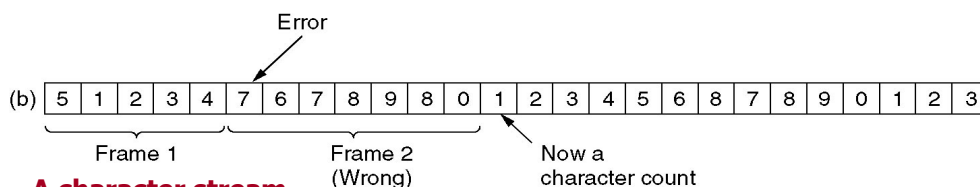
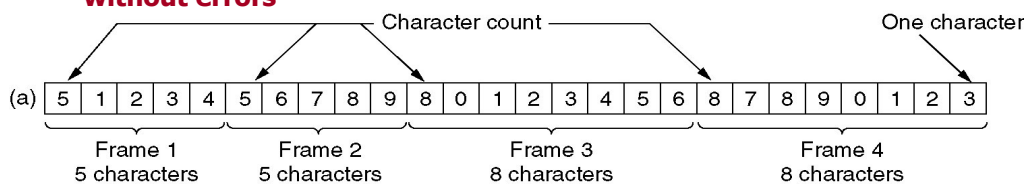
Reti di Calcolatori

5

Framing

[Tanenbaum – Computer Networks - Chapter 3]

A character stream without errors



A character stream with one error

12/03/2005

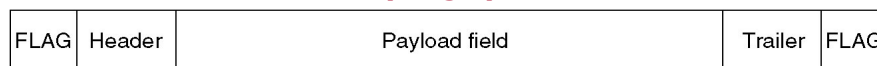
Reti di Calcolatori

6

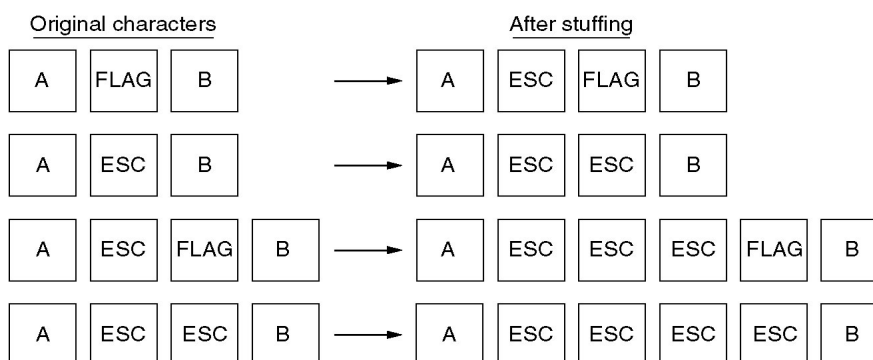
Framing (2)

[Tanenbaum – Computer Networks - Chapter 3]

A frame delimited by flag bytes



Four examples of byte
sequences before and
after stuffing



12/03/2005

Reti di Calcolatori

7

Framing (3) – bit stuffing

[Tanenbaum – Computer Networks - Chapter 3]

The original data

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

The data as they appear on the line

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

Stuffed bits

The data as they are stored in receiver's memory after destuffing

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

12/03/2005

Reti di Calcolatori

8

Errors, Error Detection and Error Control

[White - Data Communications and Computer Networks - Chapter 6]

- Noise is always present.
- If a communications line experiences too much noise, the signal will be lost or corrupted.
- Communication systems should check for transmission errors.
- Once an error is detected, a system may perform some action.
- Some systems perform no error control, but simply let the data in error be discarded.

12/03/2005

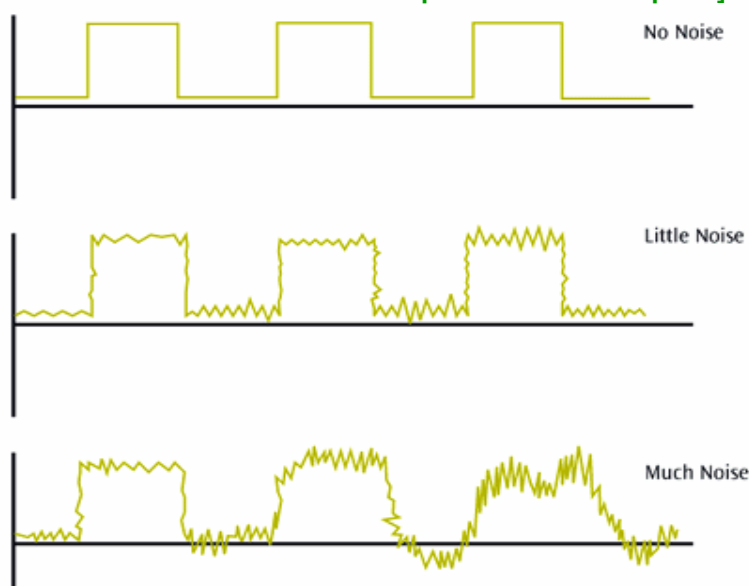
Reti di Calcolatori

9

Noise and Errors – White Noise

[White - Data Communications and Computer Networks - Chapter 6]

- Also known as thermal or Gaussian noise
- Relatively constant and can be reduced.
- If white noise gets too strong, it can completely disrupt the signal.



12/03/2005

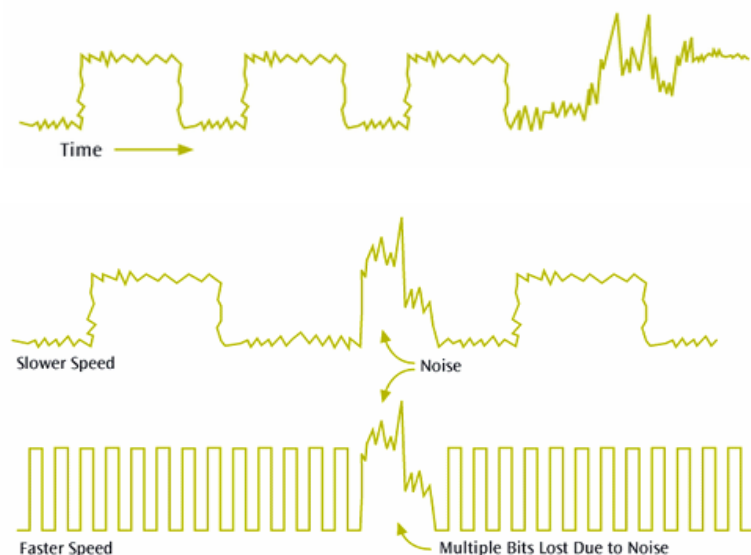
Reti di Calcolatori

10

Noise and Errors – Impulse Noise

[White - Data Communications and Computer Networks - Chapter 6]

One of the most disruptive forms of noise.
Random spikes of power that can destroy one or more bits of information.
Difficult to remove from an analog signal because it may be hard to distinguish from the original signal.
Impulse noise can damage more bits if the bits are closer together (transmitted at a faster rate).



12/03/2005

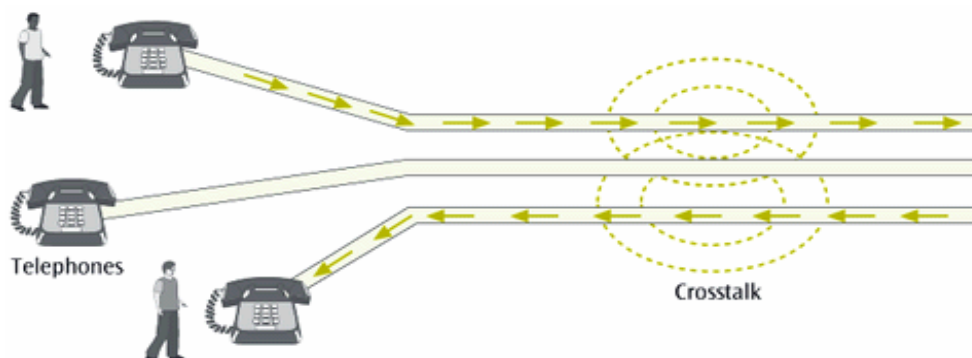
Reti di Calcolatori

11

Noise and Errors – Crosstalk

[White - Data Communications and Computer Networks - Chapter 6]

- Unwanted coupling between two different signal paths.
- For example, hearing another conversation while talking on the telephone.
- Relatively constant and can be reduced with proper measures.



12/03/2005

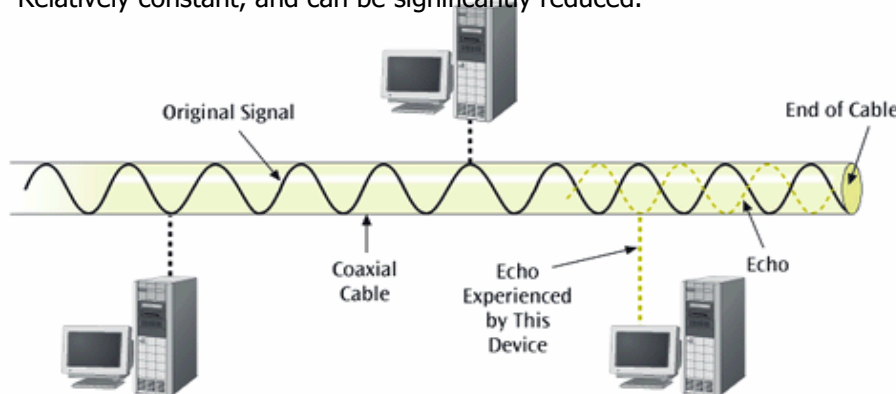
Reti di Calcolatori

12

Noise and Errors – Echo

[White - Data Communications and Computer Networks - Chapter 6]

- The reflective feedback of a transmitted signal as the signal moves through a medium.
- Most often occurs on coaxial cable.
- If echo bad enough, it could interfere with original signal.
- Relatively constant, and can be significantly reduced.



12/03/2005

Reti di Calcolatori

13

Error Prevention

[White - Data Communications and Computer Networks - Chapter 6]

- To prevent errors from happening, several techniques may be applied:
 - Proper shielding of cables to reduce interference
 - Telephone line conditioning or equalization
 - Replacing older media and equipment with new, possibly digital components
 - Proper use of digital repeaters and analog amplifiers
 - Observe the stated capacities of the media

12/03/2005

Reti di Calcolatori

14

Errori

- Errore: il dato letto da una cella di memoria è **diverso** da quello che vi era stato scritto.
 - Errore **singolo**: un solo bit è sbagliato.
 - Errore **doppio**: ci sono due bit errati.
 -
- **Codici** per la **rilevazione/correzione** degli errori:
 - a una parola di **m** bit di dato si aggiungono **r** bit di controllo e si ottiene una parola di codice di **n = m + r** bit;
 - quando si legge una parola gli **r** bit in eccesso vengono controllati per vedere se si è verificato un errore;
 - **rilevare** l'errore → **invalidare** il dato perché errato!
 - **correggere** l'errore → **sostituire** il dato con quello esatto!

12/03/2005

Reti di Calcolatori

15

Distanza di Hamming

- Indica il **numero di bit differenti** che ci sono tra due parole di uguale lunghezza
 - la distanza di Hamming tra 1000 1001 e 1011 0001 è 3;
 - può essere calcolata eseguendo l'**OR esclusivo** tra le due parole (il numero di bit a 1 del risultato indica la distanza)
 - 1000 1001 XOR 1011 0001 = 0011 1000, in cui sono presenti 3 uno.
- Se due parole di codice hanno distanza pari a **d**, ci vorranno **d** errori per convertire l'una nell'altra.
 - un errore singolo provoca la lettura di un dato errato che ha distanza di Hamming pari a 1 dal dato esatto;
 - un errore doppio provoca la lettura di un dato errato che ha distanza di Hamming pari a 2 dal dato esatto;
 -

12/03/2005

Reti di Calcolatori

16

Distanza di Hamming

- Codice rilevatore/correttore di errori:
 - parola di **n** bit → **2^n** configurazioni **possibili**;
 - parte dati di **m** bit → **2^m** configurazioni **legali**.
- La **distanza di Hamming** di un codice è il minimo delle distanze tra tutte le configurazioni legali.
- La proprietà di rilevazione/correzione di un codice dipende dalla sua distanza Hamming:
 - per rilevare un errore di cardinalità **d** serve un codice con distanza **$d + 1$** (**d** errori non possono cambiare una parola di codice legale in un'altra parola di codice legale);
 - per correggere un errore di cardinalità **d** serve un codice con distanza **$2d + 1$** (**d** errori trasformano una parola legale in una parola illegale che è comunque più vicina alla parola legale originaria che non ad altre parole legali).

12/03/2005

Reti di Calcolatori

17

Rilevazione errori: parità

- Codice di parità: ai dati si aggiunge un **bit di parità**
 - **parità pari**, il numero di 1 nella parola di codice è pari (e.g. al dato **0010** si aggiunge un **1** per ottenere **00101**);
 - **parità dispari**, il numero di 1 nella parola di codice è dispari (e.g. al dato 0010 si aggiunge uno 0 per ottenere 00100).
- La distanza tra due parole di codice è almeno **2** (distanza di Hamming del codice)
 - un **errore singolo** produce una parola di codice con **parità sbagliata** (e.g. un errore sul II bit cambia **00101** in 01101);
 - ci vogliono **due errori** per passare da una parola di codice valida a un'altra parola di codice valida (per passare da **00101** a **11101** sia il I che il II bit debbono essere errati).
- Questo codice consente di **rilevare** gli **errori singoli**.

12/03/2005

Reti di Calcolatori

18

Cyclic Redundancy Checksum

[White - Data Communications and Computer Networks - Chapter 6]

- The CRC error detection method treats the packet of data to be transmitted as a large polynomial.
- The transmitter takes the message polynomial and using polynomial arithmetic, divides it by a given generating polynomial.
- The quotient is discarded but the remainder is "attached" to the end of the message.
- The message (with the remainder) is transmitted to the receiver.
- The receiver divides the message and remainder by the same generating polynomial.
- If a remainder not equal to zero results, there was an error during transmission.
- If a remainder of zero results, there was no error during transmission.

12/03/2005

Reti di Calcolatori

19

Cyclic Redundancy Checksum

[White - Data Communications and Computer Networks - Chapter 6]

Type of Error	Error Detection Performance
Single bit errors	100 percent
Double bit errors	100 percent, as long as the generating polynomial has at least three 1s (they all do)
Odd number of bits in error	100 percent, as long as the generating polynomial contains a factor $x + 1$ (they all do)
An error burst of length $< r+1$	100 percent
An error burst of length $= r+1$	probability = $1 - (\frac{1}{2})^{(r-1)}$
An error burst of length $> r+1$	probability = $1 - (\frac{1}{2})^r$

12/03/2005

Reti di Calcolatori

20

Error Control

[White - Data Communications and Computer Networks - Chapter 6]

- Once an error is detected, what is the receiver going to do?
 - Do nothing
 - Seems like a strange way to control errors but some newer systems such as frame relay perform this type of error control.
 - Return an error message to the transmitter
 - Return a message has three basic formats:
 - Stop-and-wait ARQ
 - Go-back-N ARQ
 - Selective-reject ARQ
 - Fix the error with no further help from the transmitter
 - For a receiver to correct the error with no further help from the transmitter requires a large amount of redundant information accompany the original data.
 - This redundant information allows the receiver to determine the error and make corrections.
 - This type of error control is often called forward error correction.

12/03/2005

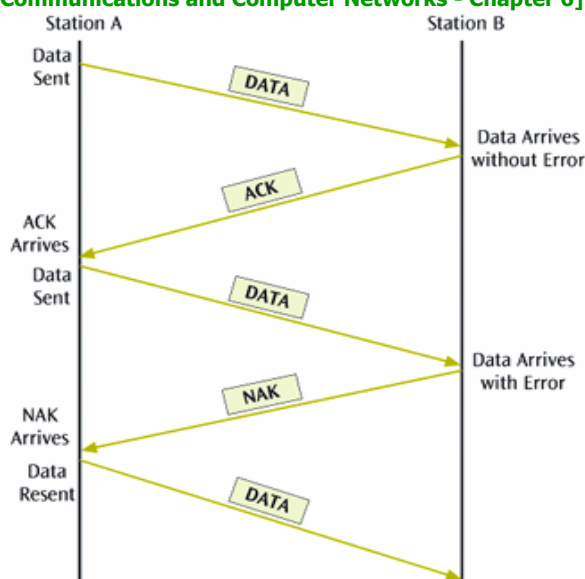
Reti di Calcolatori

21

Error Control – Stop-and-wait ARQ

[White - Data Communications and Computer Networks - Chapter 6]

- Stop-and-wait ARQ is the simplest of the error control protocols.
- A transmitter sends a frame then stops and waits for an acknowledgment.
 - If a positive acknowledgment (ACK) is received, the next frame is sent.
 - If a negative acknowledgment (NAK) is received, the same frame is transmitted again.



12/03/2005

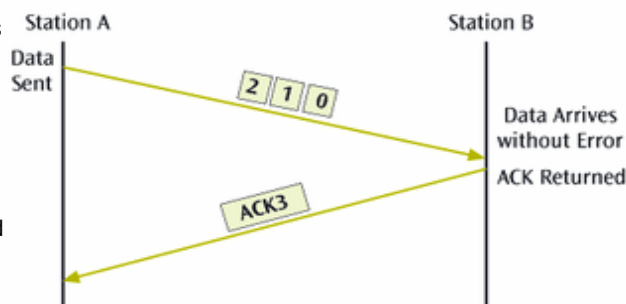
Reti di Calcolatori

22

Go-back-N ARQ and selective reject

[White - Data Communications and Computer Networks - Chapter 6]

- Go-back-N ARQ and selective reject are more efficient protocols.
- They assume that multiple frames are in transmission at one time (sliding window).
- A sliding window protocol allows the transmitter to send up to the window size frames before receiving any acknowledgments.
- When a receiver does acknowledge receipt, the returned ack contains the number of the frame expected **next**.
- Using the go-back-N ARQ protocol, if a frame arrives in error, the receiver can ask the transmitter to go back to the Nth frame and retransmit it.
- After the Nth frame is retransmitted, the sender resends *all* subsequent frames.



12/03/2005

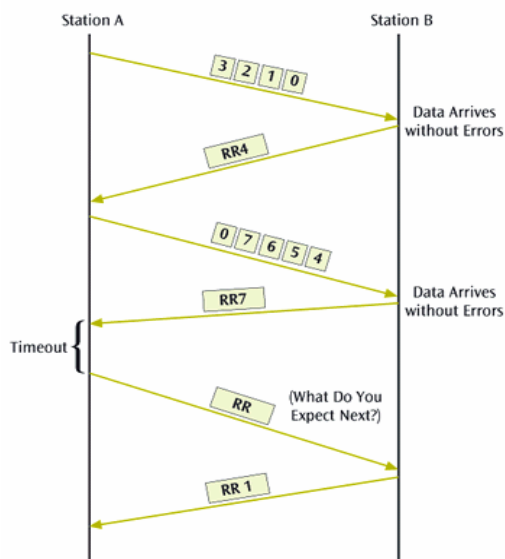
Reti di Calcolatori

23

Error Control – Selective-reject ARQ

[White - Data Communications and Computer Networks - Chapter 6]

- Selective-reject ARQ is the most efficient error control protocol.
- If a frame is received in error, the receiver asks the transmitter to resend **ONLY** the frame that was in error.
- Subsequent frames following the Nth frame are not retransmitted.
- Example of a normal transmission of frames with no errors.



12/03/2005

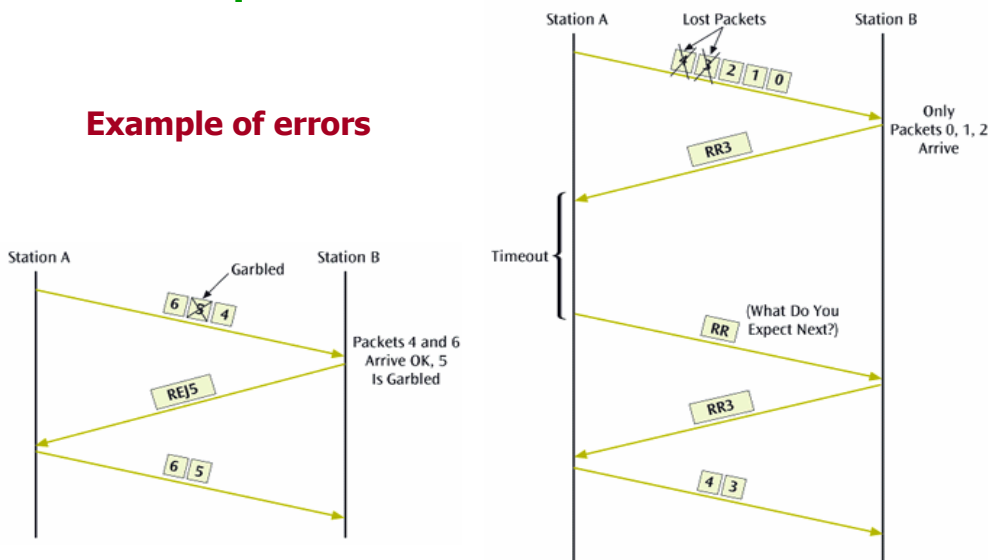
Reti di Calcolatori

24

Error Control – Selective-reject ARQ

[White - Data Communications and Computer Networks - Chapter 6]

Example of errors



12/03/2005

Reti di Calcolatori

25

Correzione dell'errore

➤ Esempio di codice per la correzione dell'errore

- il codice ha quattro parole valide:

A = 000000000

B = 0000011111

C = 1111100000

D = 1111111111

- il codice ha una **distanza di Hamming pari a 5**, perciò è in grado di correggere errori doppi.

➤ Consideriamo un esempio di correzione:

- si legge **X = 0000000111** che non appartiene al codice;
- viene calcolata la distanza tra **X** e le parole "legali" del codice:
AX=3, BX=2, CX=8, DX=7;
- si presume che in origine la parola fosse la più vicina, cioè **B**
 - se l'errore è doppio l'ipotesi è valida e permette di correggere l'errore;
 - se l'errore è triplo (o superiore) l'ipotesi non è valida e l'errore non viene corretto, ma viene "legalizzato";
 - bisogna assicurarsi che la probabilità di errore triplo sia molto bassa!!

12/03/2005

Reti di Calcolatori

26

Correzione di errori singoli

- Consideriamo che la **parola dati sia di m bit** e che quindi ci siano al massimo **2^m parole "legali"**;
- aggiungiamo **r bit di controllo** ed otteniamo **$n=m+r$ bit complessivi**, ogni parola "legale" è di **n bit**;
- nell'ipotesi di errore singolo, ognuna delle **2^m parole "legali"** ha **n parole "illegali"** a **distanza 1**, quindi richiede **$n+1$ configurazioni** (altrimenti non sarebbe possibile correggere l'errore singolo);
- con n bit ci sono al massimo **2^n configurazioni**, quindi **$(n+1) \cdot 2^m \leq 2^n$** , cioè **$(m+r+1) \leq 2^r$** ;
- fissato **m** , questo vincolo definisce il **limite inferiore del numero di bit di controllo**.

12/03/2005

Reti di Calcolatori

27

Overhead per la correzione

Bit x parola (m)	Bit x controllo (r)	Bit totali (n)	Overhead (percentuale)
8	4	12	50%
16	5	21	31%
32	6	38	19%
64	7	71	11%
128	8	136	6%
256	9	265	4%
512	10	522	2%

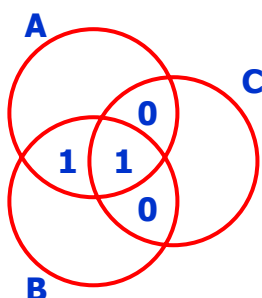
12/03/2005

Reti di Calcolatori

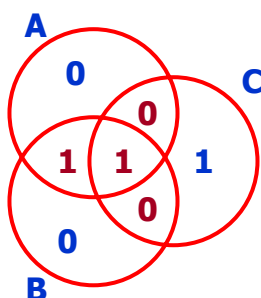
28

Esempio di codice correttore d'errore in parole di 4 bit

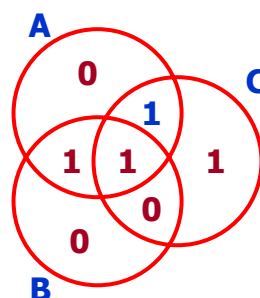
Codifica di 12_{dieci}
come 1100_{due}



Parità pari su
gruppi di 3 cifre
(001 1100)



Errore in AC



12/03/2005

Reti di Calcolatori

29

Codice di Hamming

/1

- La parola **dati** è di **m bit**, cui vengono aggiunti **r bit di codice** formando una parola di **n=m+r bit**.
- I bit vengono numerati da **sinistra a destra** a partire da 1 (il bit più a sinistra è al posto **1**, quello più a destra al posto **n**).
- I bit che si trovano in una posizione corrispondente a una **potenza di 2** ($2^0=1$, $2^1=2$, ...) sono bit di **parità**, gli altri contengono i dati:
 - si consideri per esempio **m = 16** ed **r = 5** (quindi **n = 21**);
 - i bit di **codice** si trovano in posizione **1, 2, 4, 8 e 16**;
 - i bit di **dati** si trovano in posizione **3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20 e 21**.

dato

1	2	3	4
1	1	1	1

5	6	7	8
0	0	0	0

9	10	11	12
1	0	1	0

13	14	15	16
1	1	1	0

codice

1	2	3	4
			1

5	6	7	8
1	1	1	

9	10	11	12
0	0	0	0

13	14	15	16
1	0	1	

17	18	19	20
0	1	1	1

21
0

12/03/2005

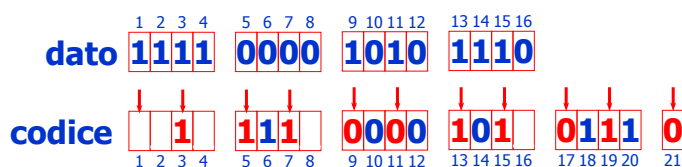
Reti di Calcolatori

30

Codice di Hamming

/2

- Ogni posizione viene scritta come **somma di potenze di due**:
 - per esempio, la posizione **5** corrisponde a **1+4** cioè 2^0+2^2 ;
 - si tratta della rappresentazione binaria della posizione ($5_{\text{dieci}}=101_{\text{due}}$);
- Ciascuna posizione è controllata da tutti i **bit di codice** che compaiono nella **sua rappresentazione binaria**:
 - la **posizione 5** è controllata dai bit di codice in posizione **1** e **4**.
 - il bit di **codice** in posizione **1** **controlla** tutte le **posizioni dispari**, cioè **1, 3, 5, 7, 9, 11, 13, 15, 17, 19** e **21**;
 - il bit n. **2** **controlla** i bit n. **2, 3, 6, 7, 10, 11, 14, 15, 18** e **19**.
- Ciascun bit di **codice** è **configurato** in modo da garantire la **parità pari sull'insieme che controlla**:



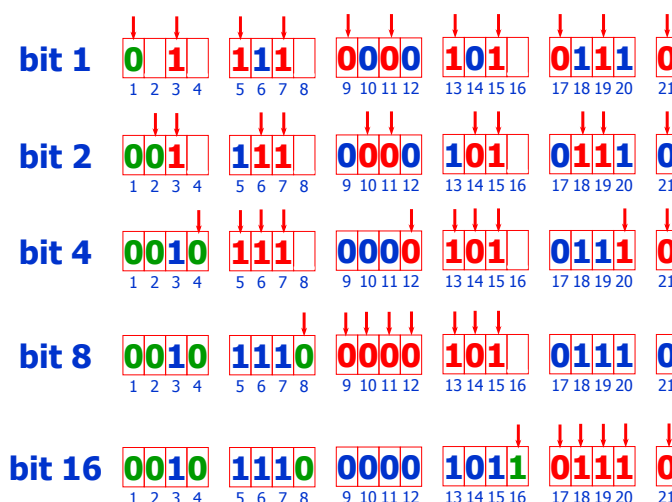
12/03/2005

Reti di Calcolatori

31

Codice di Hamming

/3



12/03/2005

Reti di Calcolatori

32

Correzione degli errori

- In caso di errore singolo, **uno o più bit** di parità **risultano errati**.
- Il **bit da correggere** si trova nella **posizione** indicata dalla somma dei **bit** di parità **errati**.

errore nel bit 5

bit 1

bit 2

bit 4

bit 8

bit 16

errore corretto

errore corretto

errore corretto

I bit di parità errati sono il n. 1 e il n. 4, si corregge il bit $4+1=5$.

È l'unico bit compreso negli insiemi controllati dai bit 1 e 4 ed esterno a tutti gli altri insiemi.