

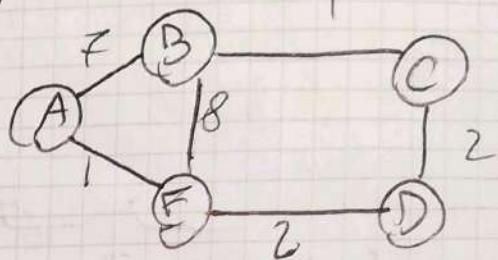
## Distance Vector (Protocollo Autobitato)

Soprattutto Bellman - Ford  
ogni nodo ha una tabella con i costi (miglia)

Colonne: Urte  
Righe: destinazioni

Nelle celle mettiamo i costi (miglia)

	A	B	D	Mia
Dest	1	12	5	
Tabella E	7	8	5	
	6	9	3	
	3	11	2	



Per raggiungere A urto come uscire B o il costo 15 che è il minimo

## Esempio Costruzione Tabella

Miglioramento zero: un giro vuole , di fare parte di vicini se ne prendono conto immediatamente e comunicano ai vicini le modifiche e affinche i valori vengano aggiornati.

Esempio su stile

Miglioramento: più complicato Esempio su stile

Problema: Se la variazione tra il nuovo e vecchio giro è grande diventerà tutti i giri prima che i corregga. Potremo in certi casi rivedere la tabella con le celle a 00

Quanto tempo è detto passare perenne

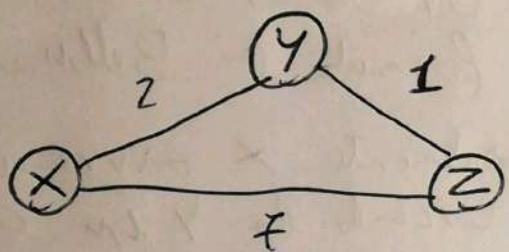


tabella X

X	Y	Z
Y	$\infty$	$\infty$
Z	$\infty$	$\infty$

Controllare la diagonale  
è  
basta

Esercizio DV vsc te

tabella  
A

Det

A	B	C
B	...	...
C	...	...

X	Y	Z
Y	2	$\infty$
Z	$\infty$	f

tabella Y

Y	X	Z
X	$\infty$	$\infty$
Z	$\infty$	$\infty$

tabella Z ...

Z	X	Y
X	f	$\infty$
Y	$\infty$	1

Controllare DV relativo a X: basta creare un vettore  
contenente Y, Z (le righe) e associare il valore  
minimo della rispettiva riga.

X	
Y	2
Z	f

Y	
X	2
Z	1

Z	
X	f
Y	1

Il fonte X invia il suo DV ai suoi vicini (Y e Z)  
così possono sfruttare le informazioni per migliorare  
la tabella.

Il router X prende come input i DV di Y e Z e fa i calcoli sfruttando la formula di Bellman-Ford

X	Y	Z
Y	$\infty$	
Z	$\infty$	7

Y	
X	2
Z	1

Z	
X	7
Y	1

Allora X arriva in Z uscendo da Y in 2  
Però raggiunge Y in 2 e Y raggiunge Z in 1

$2+1 < \infty$ ?  $\rightarrow$  TRUE: il valore 2+1 è scritto al posto di  $\infty$  nella best Z uscita Y

X	Y	Z
Y	2	$\infty$
Z	3	7

Allora X arriva in Y uscendo da Z in  $\infty$  più Z raggiunge Y in 1 e X raggiunge Z in 7.

$7+1 < \infty$ ?  $\rightarrow$  TRUE  $7+1$  è scritto al posto di  $\infty$  in best Y uscita Z.

X	Y	Z
Y	2	8
Z	3	7

Gli stessi ragionamenti sono fatti per la tabella Y che ha i DV di X e Z e per la tabella Z che ha i DV di X e Y.

N.B. Se aggiornato una riga c'è un nuovo minimo il DV va aggiornato

X		
Y	2	
Z	3	

Pointer Y

Y	X	Z
X	2	$\infty$
Z	$\infty$	1

X
Y
Z
3

Z
X
Y
1

Y va in X visto Z in  $\infty$  quindi Y va in Z in  
 1 è Z va in X in 7  
 $7+1 < \infty$ ? TRUE ma non basta perché  $\infty$

Y	X	Z
X	2	8
Z	$\infty$	1

Y va in Z visto X in  $\infty$  visto  
 Y va in X in Z e X va in Z  
 in 3

$3+2 < \infty$ ? TRUE 5 è scritto al posto  
 di  $\infty$

Y	X	Z
X	2	8
Z	5	1

Dorsale: Il DV di Y dopo questi  
 calcoli rimane il medesimo,  
 deve comunque inviare se si sono  
 due vicini?

NO, implica  
 che non ci sono  
 miglioramenti;

Y
X
Z

Tortet z

z	x	y
x	7	$\infty$
y	$\infty$	1

x	
y	2
z	3

y	
x	2
z	1

z va in x uscita y in  $\infty$  già z va in y  
in 1 e y va in x in 2

$2+1 < \infty$ ? TRUE: 2+1 è meno del punto N  $\infty$

z	x	y
x	7	3
y	$\infty$	1

z va in y uscita x in  $\infty$  già  
va in x in 7 e x va in y in 2

$7+2 < \infty$ ? TRUE 7+2 è meno

del punto di  $\infty$

z	x	y
x	7	3
y	9	1

DV di z  $\rightarrow$

z	
x	3
y	1

Questo DV va

da inserire ai tortet x e y



Tontek X (2<sup>a</sup> volta, è stato inviato un DV da Z)

X	Y	Z
Y	2	8
Z	3	7

X	Y
Y	2
Z	1

Z	3
X	1
Y	1

X arriva in Y con M<sub>010</sub> z in 8, arriva in Z in 7 e Z va in Y in 1

7+1 < 8? NO nessuna sostituzione

X arriva in Z con M<sub>010</sub> Y in 3 ma va in Y in 2 e Y va in Z in 1

2+1 < 3? NO nessuna sostituzione.

Il Tontek X v'è stoppato (il DV non è controllato)

Tontek Y (2<sup>a</sup> volta, è stato inviato un DV da Z)

Y	X	Z
X	2	8
Z	5	1

X	Y
Y	2
Z	3

Z	3
X	1
Y	1

Y va in X con M<sub>010</sub> Z in 8

Y va in Z in 1, Z va in X in 3

1+3 < 8? TRUE: 3+1 è sotto il limite di 8

Y	X	Z
X	2	5
Z	5	1

Y va in Z con X in 5

Y va in X in 2 e X va in Z in 3

3+2 < 5? NO nessuna sostituzione. Il DV rimane ed neaderemo. L'algoritmo è finito

Configurazioni finali

X	Y	Z
Y	2	8
Z	3	7

Y	X	Z
X	2	5
Z	5	1

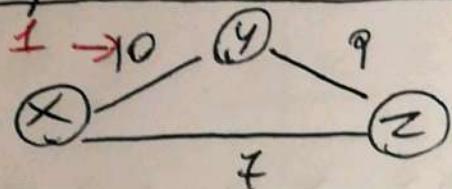
Z	X	Y
X	7	3
Y	9	1

X		
Y	2	
Z	3	

Y		
X	2	
Z	1	

Z		
X	3	
Y	1	

Moltiplicazione di un zero



X	Y	Z
Y	10	16
Z	19	7

Y	X	Z
X	10	16
Z	17	9

Z	X	Y
X	7	19
Y	17	9

Il DVK deve dare zero 10 e 1

X e Y si rinviano subito a sinistra, fornendo l'one ogni aggiornamento nello stesso tabella e iniziate il loro DV ai vicini per continuare l'algoritmo.

X	Y	Z
Y	1	16
Z	19	7

Y	X	Z
X	1	16
Z	17	9

X
Y
Z
1

Y
X
Z
1

Z va in X uscendo da Y in 19 già Z va in Y  
in 9 e Y va in X in 1

$q+1 < 19$ ? TRUE viene scritto 10 al posto di 19

Z	X	Y
X	7	10
Y	17	9

Z va in Y uscendo da X in 17  
già Z va in X in 7 e X va in Y in 1

$7+1 < 17$ ? TRUE: viene scritto 8 al posto di 17

Z	X	Y
X	7	10
Y	8	9

Z
X
Y
8

Questo DV viene inviato ai vicini

<del>X</del>	Y	Z
Y	1	16
Z	19	7

	Y
X	1
Z	9

	Z
X	7
Z	8

$X \leq 2$  in Y  $\leq 16$  e  $Z \leq 16$  per  $X \leq 2$  in Z in 7 e  $Z \leq 2$  in Y in 8

$7+8 < 16$ ? TRUE

<del>X</del>	Y	Z
Y	1	15
Z	19	7

$X \leq 2$  in Z  $\leq 16$  per Y in 19 per  $V_2$  in Y in 1 e Y  $\leq 2$  in Z in 7.

$1+7 < 19$ ? TRUE

<del>X</del>	Y	Z
Y	1	15
Z	10	7

DV inviato

Y	X	Z
X	1	16
Z	17	9

	X
Y	1
Z	7

	Z
X	7
Y	8

$Y \leq 2$  in X  $\leq 16$  per  $Z \leq 16$  per  $V_2$  in Z in 9

e  $Z \leq 2$  in X in 7: nessuna variazione

$Y \leq 2$  in Z  $\leq 16$  per  $X \leq 17$  per  $V_2$  in X in 1 e  $X \leq 2$  in Z in 7;

$7+1 < 17$ ? TRUE

<del>X</del>	X	Z
X	1	16
Z	8	9

1) DV inviata

1) 10 in X  
dove 9 ←

	Y
X	1
Z	8

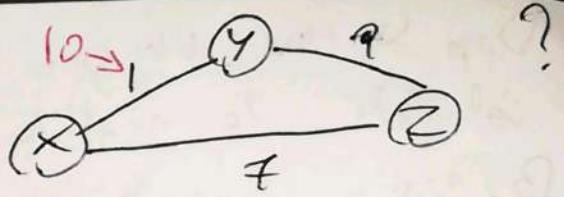
Viene inviato a X e Z che riferiscono che non ci salvo in questo caso di variazioni

## Regruppamento di numeri

X	Y	Z
Y	10	16
Z	9	F

Y	X	Z
X	10	16
Z	8	9

Z	X	Y
X	F	10
Y	8	9



X e Y fanno segno iniziale del condimento e inviano il loro DV a Z.

Z	X	Y
X	F	10
Y	8	9

X	Y
Y	10
Z	7

Y	10
X	10
Z	8

Azzino a  $(10-1)$  a tutti i valori in un punto di truci.

Z	X	Y
X	F	9
Y	11	9

X	Y	Z
Y	10	16
Z	9	F

Z	X	Y
X	10	16
Y	8	9

X	Y
Y	10
Z	7

X	Y	Z
Y	10	15
Z	18	F

Qui si nota che le formule

Y	X	Z
X	10	16
Z	17	9

Y		
X	10	
Z	8	

Z	X	Y
X	F	19
Y	17	9

Z	X	Y
X	7	10
Y	8	9

Z		
X	7	
Y	8	

→ Si può vedere come un avvaggiato a tutte le voci in cui partecipa 1 (come le formule di Bellman Ford)

Dopo di che vengono riuniti i DV e le loro  
Salgono la riga di classe

Bisogna fare in modo che venga indicato quale  
casi non sono rientri nelle tabelle

	X	Y	Z
Y	10	15	
Z	18	F	

	X
Y	10
Z	7

	X	Z
Y	10	16
Z	17	9

	X	Y
Z	7	19
X	7	19

	Z
X	X
Y	Y

	X	Y	Z
Y	10	16	
Z	19	7	

	Y	X	Z
X	10	16	
Z	17	9	

	Z	X	Y
X	7	19	
Y	17	9	

Il 15 e il  
18 erano già  
gli obiettivi  
(ma non Bellman)

Facendo i  
calcoli per  
queste due tabelle

non ci sono  
contraddizioni

S	X	X
X	0	X
X	X	5

X	X	5
X	X	X
X	X	X

X	X	5
X	X	X
X	X	X

<u>LINK</u>	<u>State</u>	<u>Routing</u>
tucols	in use	

LINK

Questo protocollo è usato per trasmettere anche dati istantanei.

Link: il controllo lo stato di link in uscita, dell'informazione viene mandata a tutti i nodi transitanti il forwarding. Dopo aver già tutti i nodi conoscono il grado.

A questo punto si può applicare Diffusion (in tutti i nodi) Envygo in slide.

Nel LS+ non trasmettono informazioni relative ai propri vicini sulla loro situazione dei link.

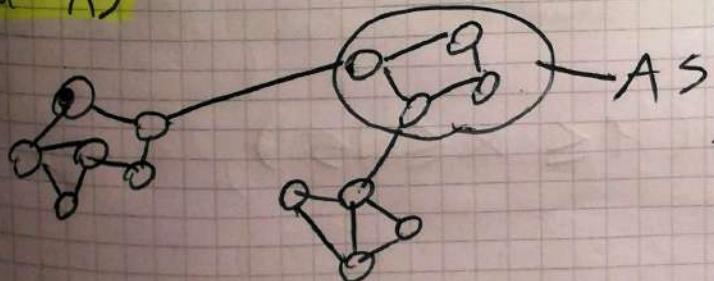
L'obiettivo oscilla poiché potrebbe tenere o lasciare mago.

## Confronto DV e LSR

DV	LSR
multicast	globale
Messaggi di vicini	messaggi broadcast
Info sulle reti.	info sui link vicini
Messaggi di variazioni di metriche	Messaggio traffico di messaggi
Info sulle reti	rd do

l'istituto è organizzato in zone autonome (AS)

im una AS C may rivel maceline  
contaminanti come l'etile o tra AS o ~~tra~~ beta  
all'AS



I zw che collegare  
sono detti di bordo

## RIPASSO DIJKSTRA

Dijkstra(s)

Initialize-single-source(s)

$Q := \text{Build-heap}(V, d)$

While  $Q \neq \emptyset$  do

$v := \text{Extract-min}(Q)$

SCAN(v)

Initialize-single-source(s)

for  $v \in V$

$d[v] = +\infty$

$\text{Pred}[v] = \text{NIL}$

$d[s] = 0$

SCAN(v)

for  $u \in \text{Adj}[v]$

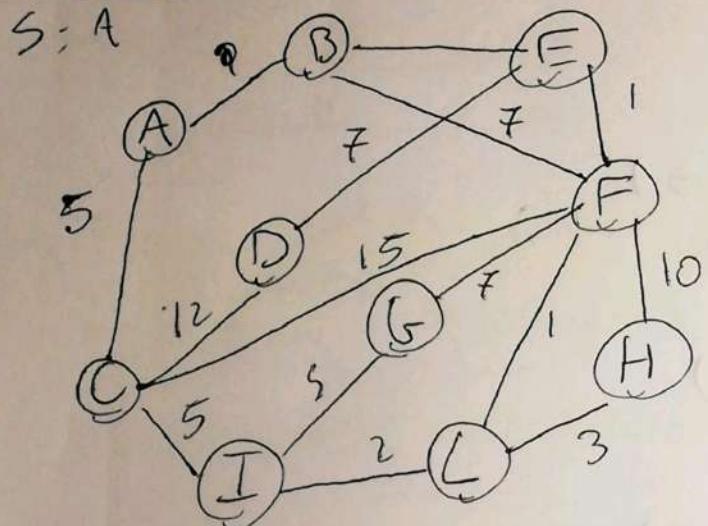
Relax(v, u)

Relax(v, u)

if  $d[u] > d[v] + w(v, u)$

$Q.\text{decreaseKey}(d[u], d[v] + w(u, v))$

$\text{Pred}[u] = v$



Si crea vettore dei Prod lungo  
 $|V|$  e vettore di n. d  
 Prod

X	X	X	X	X	X	X	X	X	X	X
A	B	C	D	E	F	G	H	I	J	L

0	0	0	0	0	0	0	0	0	0	0
A	B	C	D	E	F	G	H	I	J	L

La inizializza mettere a tutto 0 infinito tranne  
 A = 0 e Prod tutto a NIL X

Si organizza una coda di priorità dei nodi che  
 si gestisce con Key = d

Nel ciclo relazionario (extraiamo) il minimo e  
 appliciamo in tutti i nodi adjacenti la formula

Extraiamo A (non c'è già nella coda)

$$\begin{array}{l} (B) \quad (A) \quad (W(A,B)) \\ Q > 0 + 0 \end{array} \quad \begin{array}{l} \text{TRUE : } A[B] \leftarrow 0 + a \\ \text{Prod}[B] \leftarrow A \end{array}$$

FALSE : nulla

Facciamo così anche per C

e poi inizia la prima iterazione col nuovo min

Ottieni un nuovo grafo e  $S = A$

Prod

X	A	A	E	F	B	I	L	C	F
A	B	C	D	E	F	G	H	I	L

d

0	5	9	20	13	12	18	16	15	13
A	B	C	D	E	F	G	H	I	L

G: ACIG

D: ABFED

- B: AB
- C: AC
- F: ABF
- L: ABFL
- E: ABFE
- I: ACT
- H: ABFLH

## RIP (Routing Information Protocol)

Bottino in DN è una come metrica HOP, i salti.

Max HOP = 15 sono limiti massimi del grafo

sotto all'AS

Oltre tale valore il router è considerato un magnete.

Le routing table sono aggiornate inviando i vettori

Dopo 30 secondi

Se un pacchetto per un router non è stato aggiornato entro 180 s la distanza è più a

Dopo altri 120 s il router è eliminato dalla RT

dal altro router

Request: richiede info vicini vicini

Response: invia info di routing

## Informazioni nello RT

- indirizzo destinazione
- distanza della destinazione in HOP
- Next hop (router successivo o via quale pacchetto)
- timeout
- garbage-collection timer

## Dettagli su RIP v1

Hanno un lungo variabile fino a 512 byte (max 25 reti di destinazione)

## Nel pacchetto:

- Command: 1 = richiesta, 2 = aggiornamento
- Version: versione protocollo
- Address family id: 2 per id IP
- IP address

metric: hop count ( $1 \leq x \leq 15$ )

## RIP v2

- broadcastato con mostre
- autenticazione messaggi
- specifica next hop
- next horizon, poison reverse

## Diagramma RIP v2

Si aggiungono i campi Autentication, Authentication type, Route tag, subnet mask e next hop

RIPng protocollo aggiunto a IPv6 del livello utente, manca di campi autenticazione

OSPF Open Shortest Path First usa Link State Routing

- Usato per sostituire RIP
- è basato sul LSR
- è buono per reti grandi
- autenticazione in messaggi
- permette di usare altre metriche

OSPF si compone di 3 parti

- Hello: scoperta e verifica vicini
- Exchange: sincronizzazione iniziale DB
- Flooding: aggiornamento DB

Il giro nel ~~graph~~ è dato dal tempo di attesa ritardo tra due nodi

Ogni 30 minuti i forni le operazioni per aggiornare il DB, il flooding è bloccato tranne 10%

**BGP**: protocollo di interconnessione tra AS che non tiene conto di metriche ma ha "politiche"

C'è tra i router N border: router dell'AS al confine

Hanno BGP ← Router speaker: interni all'"esterno" (sono connessi quale N border)

**BGP neighbors**: coppia di speakers che

condividono info di interconnamento inter-AS

Se due AS comunicano lo fanno tramite il BGP  
che evita e risolve problemi di compatibilità,  
le due AS potrebbero avere protocolli interni  
diversi

## DLL

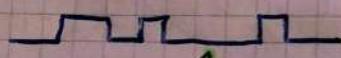
Si occupa di fornire al livello di Rete un servizio  
di trasmissione di frame a bit

Compiti principali:

- raggruppa bit per formare pacchetti (framing)
- gestisce l'accesso al mezzo fisico
- fornisce un recepto affidabile
- gestisce gli errori avvistati ed corrende
- regola il flusso di dati traffic e limita

Nel caso lo connette o c'è o non c'è.

Abbiamo 2 stati:



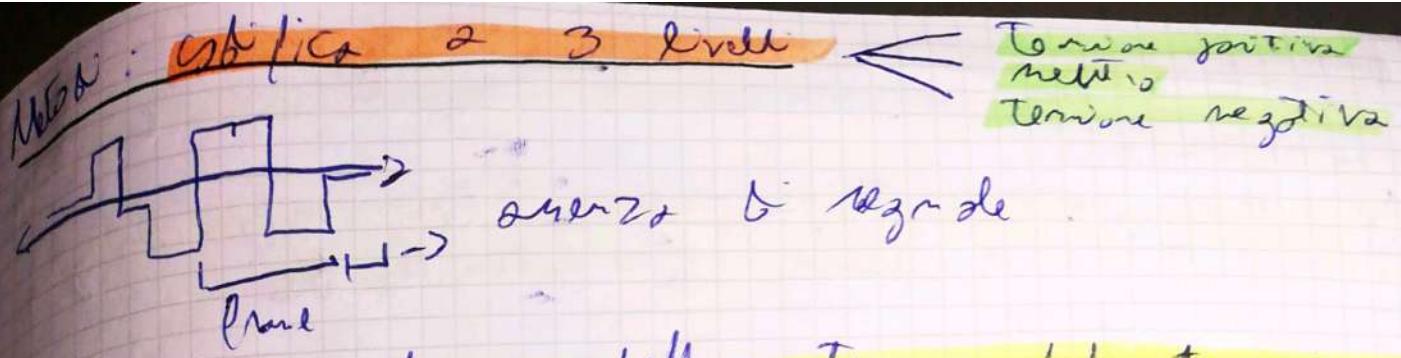
Se c'è: bit 1

Se non c'è: bit 0

bit 0 o assenza di comunicazione?

"Soluzione": creare un intervallo in cui si è obbligati a trasmettere sempre (intervallo minimo)

Nel intervallo minimo: ho il problema legato a "quando inizia" e "quando finisce"



Si cerca un terzo filo potenza del terzo filo  
potere fare un intera a 5 livelli e quindi  
un quinto della potenza del mezzo

### Codifica 5B5B

5 bit del DLL corrispondono a 5 bit nel filo  
Nella tavola ci sono anche le reg. per definire  
l'inizio e la fine di frame  
Conto: per 5 bit ne uso 5 = 20% di spazio  
Nona la codifica tutti 0 nel filo per l'antigiti

### Codifica 8B10B

8 bit DLL corrispondono con 10 fili  
Abbiamo numero di bit non poiché ovviamente  
dai bit a 0 e metà a 1 la funzione meglio  
il sistema  
Sai che è una codifica elicoidale  
dato che il numero di 1 è il numero di 0  
vengono tenuti uguali  
Invece di 8 bit c'è bisogno in

5 bit → 6 fili  
3 bit → 6 fili

Il sistema cerca di comporre il ~~numero~~ il numero  
di fili con meno errori (0.05%) ~~per~~ il numero  
di bit a 1 uguali al numero di bit a 0  
Questi sistemi compiono sync

Quindi conviene usare 5 bit in 6 fili invece  
di 3 bit in 6 fili perché il tutto già mentre  
permesso in base di valore/segnale nella reg di 3 bit

## Rilevazione e lo connessione ECRN

Troviamo una sequenza di segnali e ci sono interferenze

La rilevazione si ottiene tramite impianti più ribombanti che fanno da controllori.

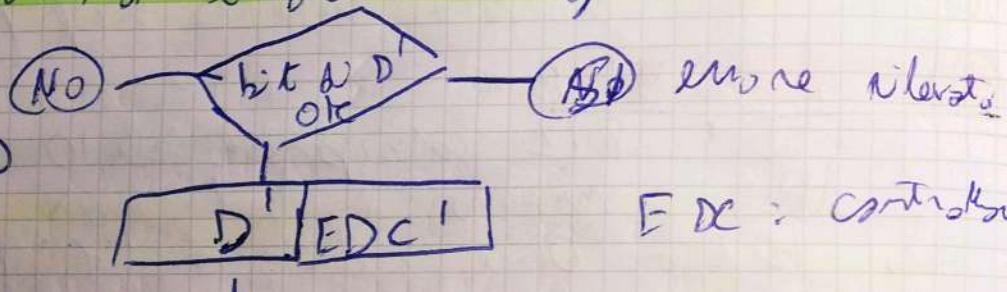
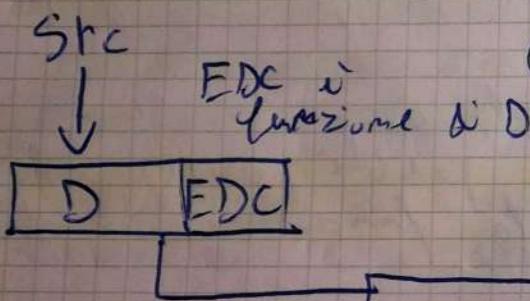
### Controllo surface

#### Porta

Si usa nei codi che hanno margine di errore massimo a 0

Il controllo di porta dice solo se i bit vanno bene o no.

E dice che i bit sono sbagliati oppure i bit sono giusti non è fatto in questo



EDC: controllo

### Porta a 2 linee orizzontali

scrivendo stampe e creare una matrice

$A_{1,1} \dots A_{1,j}$  | bit controllo righe

$A_{2,1} \dots A_{2,j}$

$A_{i,1} \dots A_{i,j}$

bit controllo colonne

Se c'è un solo errore la riga e la colonna a trave e anche a congegno

Con anche 2 errori il intera riga o colonna

CRC codice di controllo di parità  
bitorrente codice di errore correttore  
flessione errori off-line  
rapida implementazione

D: dati da memorare      R: r bit CRC

Funzione di bit e si scrivono con un polinomio  
esempio:

$$1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1$$

$$x^7 + x^5 + x^2 + x^1 + 1$$

$$x^6 + x^5 + x^3 + x + 1$$

$$x^6 + x^5 + x^3 + x + 1$$

Norma:

$$(x^6) : x^7 + x^6 + x^3 + x^2$$

Segno progetto  
 $\oplus$   
 $\times^2$

(ANO):

$$x^{13} + x^{11} + x^8 + x^7 + x^5 + x^3 + 1$$

### Esercizio scommesse $\mathbb{Z}^2$

$$x^7 + x^5 + x^2 + x + 1 \quad x^6 + x^5 + x^3 + x + 1$$

Si fa lo XOR tra i monomi e si sommano

$$x^7 : 1 \otimes 0 = 1$$

$$x^6 : 0 \otimes 1 = 1$$

$$x^5 : 0 \otimes 0 = 0$$

$$x^4 : 1 \otimes 1 = 0 \rightarrow x^7 + x^6 + x^3 + x^2$$

$$x^3 : 0 \otimes 1 = 1$$

$$x^2 : 1 \otimes 0 = 1$$

$$x : 1 \otimes 1 = 0$$

$$1(x^0) : 1 \otimes 1 = 0$$

### Esercizio problema $\mathbb{Z}^2$

Stesi polinomi di gradi 2.  $x^F$  modifica tutti gli  $a$  di  $A$  e solo via

$$x^7 : x^{13} + x^{11} + x^{10} + x^8 + x^7 +$$

$$x^6 : x^{10} + x^8 + x^7 + x^5 + x^3 +$$

$$x^2 : x^8 + x^6 + x^5 + x^3 + x^2 +$$

$$x : x^7 + x^5 + x^3 + x^2 + x +$$

$$1 : x^6 + x^5 + x^3 + x + 1$$

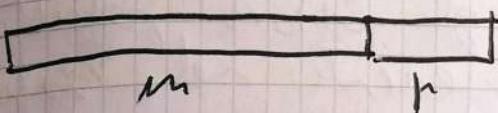
Ottiene questo binomio lungo e procede con degli XOR  
 $(x^m + x^n = 0 \times^m = //)$

Dunque i monomi che sono bipari (di  $x^8$  ce sono 3)  
 riformano rimane uno, quelli pari 0

$$R_1 : x^{13} + x^{11} + x^8 + x^7 + x^5 + x^3 + 1$$

~~D invia~~ polinomio  $M(x)$ : polinomio da inviare  
~~D riceve~~ polinomio generatore  $G(x)$

$R(x) = x^r M(x) \bmod G(x)$  Il trattante usa  $R(x)$  come ribombardore



Questo è quello che viene fatto  $x^r M(x)$ . Abbiamo aggiunto un "0"

Se le parole da testo diversa da 0 c'è stato un errore (non so dove)

Il  $G(x)$  è scelto così degli standard

del CRC rimaneva 2 interruzioni già erano, formano ~~"rotazione"~~; belli.

Facciamo nella pratica lo XOR delle sequenze dei bit entrambi con il polinomio generatore

$$\begin{array}{r} 11010 \\ 10011 \\ \hline 01001 \end{array} \quad \begin{array}{r} 11011110 \\ 10011 \\ \hline 01001110 \\ 10011 \\ \hline 000000 \end{array}$$

Regola: Se il 1° bit non è 0 allora reg è 1  $\rightarrow$  XOR con 0 e shift  $\dots$  al 1°  $\otimes$  bx

Se il 1° bit è 0 faccio solo shift

Se alla fine il reg è tutti 0 non ci sono errori

Nella pratica è un simbolo registrato o scritto

### Correzione errori

Distanza di Hamming: Distanza tra due codeword

10001100 XOR

$$11000100 = 0\textcircled{1}00\textcircled{1}000 \quad d=2$$

La distanza è 2 dato che per generare da un codeword altri altri servono 2 cambi di bit

Vocabolario: insieme di codeword

- 1) 000 000 Questo è un codeword da 3 codeword
- 2) 000 111 Distanza del vocabolario: calcolare tutte le distanze nel vocabolario, la distanza
- 3) 111 000
- 4) 111 111 Il vocabolario è la minima.

Questo è un vocabolario da 3 codeword e  $d=3$

L'idea è che il trasmettente invia una parola del vocabolario, ma per gli errori potrebbe arrivare una parola non presente nel vocabolario.

La correzione si basa sul fatto che

$$P(e=1) \gg P(e=2) \gg P(e=3) \dots$$

Ovvero la probabilità che ci sia un solo errore è molto maggiore della probabilità di 2 errori e così via.

Dunque se arriva un input/codeword non presente nel vocabolario gioiamo sul fatto che la probabilità che ci sia stato un errore e che la parola originale è quella con la distanza minore (distanza con la codeword in input)

Per esempio se è vero che un vocabolario con

$$\delta = 2e + 1$$

per  $n$  valori si necessita un vocabolario con  $\delta = e + 1$

la connessione sarà fatta in base probabilistica

In un vocabolario per  $m$  bit di dati, se  $t$  bit di controllo

$$m+t = n$$

I comb possibili sono  $2^m$ , quelle valide sono  $2^{m-t}$

Perche' 10001, è ~~ottenuto~~ una codifica valida

quelle a distanza 1 (sono  $m$ ) le restano non valide

$$\delta = 3 \text{ dunque } (m+1)2^m \leq 2^n$$

$$\rightarrow m+1 \leq 2^{t-f}$$

10 vocabolari seguire questa regola

E se  $\delta = 5$ ?

le formule non va bene

abbiamo un codice  $001100000000$

bit controlli

Dati originali  $10010001100$

$$\begin{array}{ccccccccc} \times & \times & | & \times & 001 & \times & 0001100 \\ 1 & 2 & 3 & 4 & & & 8 \end{array}$$

$$b_1 = 3 \otimes 5 \otimes 7 \otimes 9 \otimes 11 \otimes 13 \otimes 15$$

$$b_2 = 3 \otimes 6 \otimes 7 \otimes 10 \otimes 11 \otimes 15 \otimes 15$$

$$b_3 = 5 \otimes 6 \otimes 7 \otimes 12 \otimes 13 \otimes 15 \otimes 15$$

$$b_8 = 9 \otimes 10 \otimes 11 \otimes 12 \otimes 13 \otimes 15 \otimes 15$$

(sono bit mille per-potenze 2)

$$\text{Troriamo } b_1 = 1 \quad b_5 = 1$$

$$b_2 = 0 \quad b_8 = 0$$

key:  $101100100001100$

Sintesi combit ridotti  
bit di controllo  
controlli sono ridotti  
(mille per-potenze del 2)

Scrittura in  
base 2

$$3 = 1 + 2$$

$$5 = 1 + 4$$

$$6 = 2 + 4$$

$$7 = 1 + 2 + 4$$

$$9 = 1 + 8$$

$$10 = 2 + 8$$

$$11 = 1 + 2 + 8$$

$$12 = 4 + 8$$

$$13 = 1 + 4 + 8$$

$$15 = 2 + 4 + 8$$

$$15 = 1 + 2 + 4 + 8$$

Supponiamo di ricevere il messaggio con un errore

$101100100101100$

Il ricevente sa quindi che i bit si controllano ( $1, 2, 3, 8$ ) e i bit 1 e 8 sono errati perché non coincidono con le parole (messaggio corretto) che il ricevente ha ricevuto (sono errori).

$$b_1 = 1 \otimes 0 \otimes 1 \otimes 0 \otimes 0 \otimes 1 \otimes 0 = 1$$

$$b_2 = 1 \otimes 0 \otimes 1 \otimes 1 \otimes 0 \otimes 0 \otimes 0 = 1 \quad \text{non coincide con } b_0 \text{ in } 2$$

$$b_3 = 0 \otimes 0 \otimes 1 \otimes 1 \otimes 1 \otimes 0 \otimes 0 = 1$$

$$b_8 = 0 \otimes 1 \otimes 0 \otimes 1 \otimes 1 \otimes 0 \otimes 0 = 1 \quad \text{non coincide con } b_0 \text{ in } 8$$

Il ricevente vede che  $b_2$  e  $b_8$  non coincidono con  $b_0$  quindi ha problemi.

Entro 1 bit; uno è uno sbaglio, che influenza gli elementi i bit entro 2 e 8?

Sì, comunque la tabella delle scorrizioni in potenza due si rende conto che il bit 10 influenza gli bit 2 e 8.

Il ricevente è tenuto ad accorgersi che il bit 10 è errato.

$101000100001100$   
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Scary.

$$3 = 1 + 2$$

Errore nel  
Bit di  
controllo

$$b_1 = 3 \times 5 \times 7 \times 9 \times 11 \times 13 \times 15$$

$$5 = 1 + 4$$

$$b_2 = 3 \times 6 \times 7 \times 10 \times 11 \times 13 \times 15$$

$$6 = 2 + 4$$

$$b_3 = 5 \times 7 \times 12 \times 13 \times 15 \times 15$$

$$7 = 1 + 2 + 4$$

$$b_8 = 9 \times 10 \times 11 \times 12 \times 13 \times 15 \times 15$$

$$9 = 1 + 8$$

XOR:  $0 \oplus 0 = 0$

$$10 = 2 + 8$$

Avendo -1

$$11 = 1 + 2 + 8$$

$$12 = 3 + 8$$

$$13 = 1 + 2 + 8$$

$$14 = 2 + 3 + 8$$

$$15 = 1 + 2 + 3 + 8$$

$$b_1 = 1 \times 0 \times 1 \times 0 \times 0 \times 1 \times 0 = 1$$

$$b_2 = 1 \times 0 \times 1 \times 0 \times 0 \times 0 \times 0 = 0$$

$$b_3 = 0 \times 0 \times 1 \times 1 \times 1 \times 0 \times 0 = 1$$

$$b_8 = 0 \times 0 \times 0 \times 1 \times 1 \times 0 \times 0 = 0$$

Le tre N bit non corrispondono ai bit di controllo

Erre 1 bit; Uno è uno, ma che influenza sul b3? NO (vedi Tossicyn.)

Allora b3 è errato.

**Come si fanno le file**

Pretendiamo una reg. di codeword da trasmettere, da  
scriviamo per colonne e trasmettiamo in questo modo  
in ricezione (facendo tutte tecniche di buffering e ribacchiare)  
se una codeword arriva con tutti errori la fila non  
c'è problema.

Infatti si ritroviamo la colonna e righe e vediamo  
che ogni riga avrà un solo errore forse  
comenzando da una fila  
la fila è spalmarie tutti errori di file in più corrispondenti  
per correggerli

## Codici di Hamming

xx1x110  
xx0x011  
xx0x110  
xx1x000  
xx0x011  
xx1x100  
xx0x001  
xx0x110  
xx1x011  
xx1x110  
xx1x001  
xx0x011

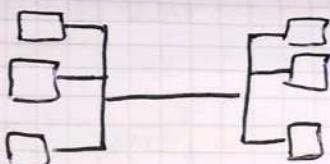
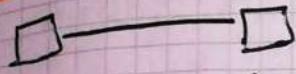
xxxxxxxxxxxxxx100101001110xxxxxxxx101001  
010100111010011101010010101011

xxxxxxxxxxxxxx100101001110xxxxxxxx101001  
01010000010111101010010101011

xxxxxxxxxxxxxx100101001110xxxxxxxxxxxx101001  
010100000101111101010010101011

xx1x100  
xx0x001  
xx0x100  
xx1x010  
xx0x001  
xx1x110  
xx0x011  
xx0x110  
xx1x011  
xx1x110  
xx1x001  
xx0x011

## Access ai link



access multiuso

Un'ente cogne se la linea è già occupata la trasmissione si va avanti

Protocollo Aloha: è un protocollo di accesso esclusivo

- enti di comunicazione che ne sfruttano uno al "caso" che fa ripetizione. Le onde di comunicazione creano un campo e l'estensione del campo "trasmette" campo

Idea: trasmetti, se va bene va bene, se non va bene ti riporta.

Il gme Aloha risulta al gme il 18,5% delle sue potenzialità.

Cercano di migliorare le prestazioni cercando lo stesso Aloha.

- stabilisce che la linea ha fini e fini, degli slot in cui inserire le frame per l'invio (slot temporali)

Con questo accorgimento le prestazioni fanno salto

N.B.: ogni ente gress solo si progr. slot

L'entità comincia una transm. perché una sua linea è pronta (lo usa già da un tempo "iniz. slot")

CSMA: gli enti che inv. il canale tra troppe "stazioni" (non-persistent) trasmissione appena respondo lo portano (di fatto)

P-persistent si riduce il tempo della precedente trasmissione, si trasmette con probabilità p

Non-persistent stazione risulta un tempo random e ripete a molla il canale

CSMA/CD - molto reperito (Ethernet) (CD = collision detection)

Risolvono i problemi legati a collisione

Fibra: Entrambe le c'è pulsanti, se c'è segnali

Si prevedono "slot"

L'idea di CSMA è vedere se il canale è  
vettivo vuoto e in caso non di tempo le istruzioni  
Per le istruzioni il ricezione rientra avere il messaggio  
Il CSMA CD vuole interrompere le collisioni per evitare  
e non intuire il canale di collisioni.  
Quanto è la probabilità di Ethernet  
Averci molti robot la collisione libera il canale  
Per riguadagnare la comunicazione dopo questo periodo di  
transmissione in ogni slot il contention period,  
ogni due macchine comunicanti si contendono di slot  
Una macchina vincerà la contesa e trasmetterà, l'altra  
non trasmetterà.

Non sono che i tentativi aumentano, aumentando gli slot per aumentare le probabilità che qualcuno vinca la contesa.

La dimensione degli slot è pre determinata  
Nel CSMA CD sempre c'è uno i giochi di transmissione  
giocando la collisione / contesa

La contesa è fatta generando numeri.

A gess 0 o 1, B gess

0: ~~prova~~ <sup>dopo t=0</sup>  $t = \frac{1}{2} \cdot 2^0$  Se entrambi 0 o 1 lo rifiutiamo

In caso abbiamo rifiuto resterà

l'altro di saltare binaria esponenziale

Alla fine i regole tra 0 e 1, alla seconda  $T_{12}$   
0, 1, 2, 3, ... (n è 3 oggetti t=3)

Si fa max 10 volte (0, 1, 2, ..., 1023)

t: tempo per inviare 512 bit

## Protocollo a mappa bit entro le collisioni

Ritardo delle macchine e le scateni, già in macchina ci sono i bit.

La macchina che ha premuto lo slot è la cui slot.

Ogni slot è 1 bit.

Fatto la prenotazione tutti sanno chi deve parlare e con quale ordine.

Poi dopo la comunicazione (regenerazione) c'è di nuovo il periodo di prenotazione.

La distribuzione non è equa: macchine a numeri alti sono privilegiate nel basso traffico.

Se ho progettato per 10 macchine e invia l'offerta come riorganizzo il tutto? La vorrei comunicare a tutti, non è semplice.

## Conteggio binario e ritardi CANBUS entro le collisioni

0: assenza di segnale

1: segnale

Se due macchine dicono 1, dicono lo stesso cosa (no prob).

Se una dice 0 e

l'altra 1, vince chi

dice 0. Res: sul canale c'è solo 1.

Il voto: momento le macchine che inseriscono il proprio id sulla rete

Come si posso avere il MAC Address

Se una macchina grande fa contesa lascia momentaneamente il canale

(a rx)

Il voto non è equo: chi ha molti 1 è molto vantaggioso

## Protocollo a token

entra le collisioni

## Token bus

può essere usato nelle centrali

Cresce il latency e un token che fa il  
gioco di forbire, il token già per forza  
è tutto il gioco di forbire.

Problema: il token è un messaggio e ha tutte le  
sue problematiche

Se arriva un nuovo ento nel canale come fa a  
decidere di forbire per inviare senza il token?

Se una macchina è spenta e gli "arriva" il token  
il token è perso.

Se il token si perde ci deve essere qualcuno che lo  
genera, i più gestire tramite timer e altre tecniche

Il token bus crea bisogno di trovare un  
coordinamento che è già molto complicato.

ESMA CD / ethernet ~~contatta~~ prova, se va bene ok