

in o Lo STRATO DI CONEGAMENTO (DATA-LINK)

In una rete i nodi sono collegati tramite i Link che possono essere di vari tipi: cavo, wireless, lan, ... Per Data-Link si intende tutto ciò che trasporta l'informazione da sorgente a destinazione. I protocolli di questo strato operano sezione per sezione e si occupano del trasferimento dei pacchetti lungo un singolo canale. Le unità di dati sono le TRAMES o FRAMES. I servizi che vengono offerti dallo strato sono:

1- **FRAMING**: Il protocollo definisce il formato della PDU, cioè delle Trame. Quindi decide la lunghezza dello SDU (spazio utile) e della PCI (intestazione di controllo). Nella sua versione incapsulato il pacchetto fornito dalla strato di rete.

2- **MAC (Medium Access Control)**: È un protocollo che si attiva solo quando sono presenti più terminali. Il suo compito è controllare l'accesso al mezzo, cioè identificare il destinatario corretto.

3. **Controllo di Flusso**: Controlla l'arrivo dei dati in modo da non saturare il nodo ricevente, non dipende dalla velocità del canale.

4. **RIVELAZIONE E CORREZIONE DEGLI ERRORI**: Il nodo ricevente rileva la presenza di errori ed eventualmente li corregge. Gli errori non sono causati solo dal trasmesso sul mezzo trasmissivo, infatti possono essere di natura naturale:

- Ricevo al posto di uno 0 un 1 o viceversa.

- Ricevo meno bit di quelli innati.

- Ricevo tutti i bit ma in disordine.

In realtà lo strato Due è una combinazione tra Hardware e Software. È una scheda elettronica chiamata SCHEDA DI RETE (o Network INTERFACE CARD / NIC) che permette la trasmissione fisica dei dati e processa le informazioni secondo i servizi già descritti sopra.

FRAMING: Il suo compito è di incapsulare la PDU nella SDU e delimitare i vari frame. Infatti il nodo ricevente deve, senza ambiguità riconoscere l'inizio e la fine di una trama.

Per dividere i vari frames nella parte di controllo, PCI, viene inserito un FLAG che poi venga ripetuto alla fine dell'unità. Questo flag deve essere riconosciuto sia dal mittente che dal

destinatario. Un flag, essendo una sequenza di bit, potrebbe essere confusa con una parte di informazioni. Per evitare ciò si inserisce un bit di controllo (BIT STUFFING) che viene inserito dopo una sequenza "sospetta". ESEMPIO:

FLAG: 01111110 → bit stuffing: inserisco uno 0 dopo 5 1

segnale originale: 10111111111101101111100

segnale trasmesso: 10111110~~1~~11110~~0~~1101111100

Quando il segnale viene ricevuto, si opera il DE-STUFFING. Cioè si contano i bit delle serie sospetta (cinque 1) se il bit seguente è come nella flag allora il frame è terminato (dopo cinque 1 trovo un altro 1) altrimenti elimino quel bit (dopo cinque 1 trovo uno 0 allora quest'ultimo lo elimino).

Stesso discorso puo' essere fatto con i byte. Alcuni protocolli, come PPP (point-to-point protocol) al posto di un bit stuffing usano un byte stuffing cioè una sequenza di 0 e 1. Anche in questo caso il modo ricevente opera il De-Stuffing.

Controllo d'ERRORE:

E' la funzione chiave dello strato di collegamento, anche se alcuni protocolli dello strato 4 operano questi controlli ma agiscono da estrema ad estrema.

Come sappiamo la trasmissione di un segnale puo' indurre ad errori. Il Bit Error Rate (BER) e' il rapporto tra bit sbagliati e bit totali. Il controllo di errore assicura un determinato livello di accuratezza nel trasferimento di dati.

Esistono due approcci possibili:

1. ERROR DETECTION & RETRANSMISSION (A RQ → Automatic Repeat Request): rivelazione erriore e ritrasmissione del messaggio.

2. FORWARD ERROR CORRECTION (FEC): rivelazione errori e correzione automatica.

Il principio che li accomuna è l'organizzazione dei dati in codeword (Parole di codice o correzione). Se il blocco ricevuto non è un codeword allora è considerato errore.

Vengono inseriti anche dei bit di controllo (overHEAD). Anche questo procedimento però non è esente da errori.

Un primo modo per effettuare questo principio base è il controllo di PARITÀ SINGOLA.

La mia informazione ha K bit. Creo uno codeword di $K+1$ bit in cui il $K+1$ -mo è 1 se gli uni della mia informazione sono dispari, è zero 0 se sono pari.

Quindi il numero di 1 sarà sempre pari.

Chi ricevere sa, per convenzione, che il bit di controllo è alla fine.

FLAG: SEGNALE + FLAG	DATA	OVERHEAD
0/1		

Le limitazioni di questo controllo sono:

- rileva gli errori ma non si sa in che posizione
- il sistema fallisce se il segnale subisce 2 o più errori

Un altro modo è il controllo di PARITÀ BI-DIMENSIONALE:

Spazzetto le righe e le colonne. Alla fine di ogni riga e colonna inserisco un bit di parità. Questo metodo è chiamato anche controllo di parità a blocchi.

1001000
0100011
1001000
1101100
1001111

Con questo metodo sono rilevabili 1 (anche la posizione), 2 o 3 errori ma da 4 in su non sempre.

Naturalmente in questo caso l'overhead è maggiore rispetto al controllo mono-dimensionale. La percentuale di over-head è il rapporto tra i bit extra e i bit totali. Si capisce che più è bassa questa percentuale, più il controllo è in linea ad errore.

Ci si può chiedere con che probabilità un codice può rivelare degli errori. Quando creo una parola di codice, posiziono i bit di redundanza (overhead). Quanto più sono questi bit tanto più aumentano le possibili codeword. Per distingue tra parole di codice ci intende il numero di bit necessari per fraintenderle.

Nel caso del controllo di parità singola se avevo 1 errore ciò che ricevereo era una noncodeword, se avevo 2 errori allora ottenevo un'altra codeword diversa.

Perciò in questo codice la distanza tra codewords è bassa.

Più alto è la distanza tra parole più è buono il codice, ci sono vari studi per maximizzarla. Essa è chiamata "distanza di Hamming".

Ricapitolando i codici di parità singola hanno scarse prestazioni, quelli bidimensioniali richiedono un numero elevato di overhead.

Però i codici più usati sono: 1) INTERNET Checksums

2. Codici Polinomiali a Ridondanza Ciclica (CRC)

1. Codice implementato da IP, TCP e UDP. È specificato nel RFC 1071 di Internet.

E' un codice che non protegge l'informazione stessa ma inserisce due bit di controllo (CHECKSUM) per rilevare errori nell'header (intestazione).

Per creare uno checksum si procede: (caso IP)

- Si divide la stringa da proteggere in L parole da 16 bit. Proprio 16 perché l'intestazione di IP è formata da righe di 32 bit.

- Sommo le varie parole $\rightarrow x = b_0 + b_1 + \dots + b_{L-1}$ e poi applico modulo $2^{16} - 1$

$$\text{Il checksum } b_L = -x \bmod 2^{16} - 1$$

Perciò l'intero blocco trasmetto deve rispettare la seguente proprietà:

$$b_0 + b_1 + \dots + b_{L-1} + b_L \bmod 2^{16} - 1 = 0$$

Esempio con stringhe di 4 bit

$$b_0 = 1100 \text{ (12)}, b_1 = 1010 \text{ (10)} \rightarrow b_0 + b_1 = (22) = 7 \bmod 15$$

$$b_2 = -7 = 8 \bmod 15 = 1000$$

STRINGA INVIATA

$$\begin{array}{c} \underline{1100 \ 1010 \ 1000} \\ \text{SDU} \qquad \text{PCI} \end{array} \rightarrow \text{CONTENUTO FINALE} \quad \begin{array}{c} 1100 \\ 1010 \\ 1000 \end{array}$$

Extra informazione

$$\textcircled{1} \ 1110 - 1111 \text{ che in } \bmod 15 \text{ è } 0$$

Anche se viene rispettata la proprietà, non c'è certezza che non sia stato commesso un errore.

2. La sequenza di K bit da inviare viene rappresentata come un polinomio $P(x)$

di grado $K-1$. Sia il nodo emittente che il ricevitore conoscono ed utilizzano un polinomio comune $G(x)$ di grado Z. [Fa parte del protocollo e non ha nulla a che fare con P(x)]

L'entità emittente opera questa espressione: $x^Z P(x) + R(x)$ che sarà uguale a $Q(x) \cdot G(x) + R(x)$ dove $Q(x)$ è il quoziente ed $R(x)$ è il resto.

Ciò che viene inviato è $T(x) = x^Z P(x) + R(x)$. Essa è una sequenza di bit, che viene inserita in un apposito campo della PDU chiamato campo CRC

$T(x)$ rappresenta una PAROLA DI CODICE che ha grado $K+Z-1$.

OPERAZIONI TRA POLINOMI IN BINARIO

- RAPPRESENTAZIONE: $x^5 + x^3 + x^2 + 1 \rightarrow 101101$
- SOMMA: L'addizione e la sottrazione sono operazioni identiche, equivalgono ad un XOR ($0+0=0 / 0+1=1 / 1+0=1 / 1+1=0$)

$$(x^5 + x^4 + 1) + (x^5 + x^2) = 110001 + 100100 = 010101 = x^4 + x^2 + 1$$

- MOLTIPLICAZIONE: Equivale a "shiftare" il primo polinomio per tanti posti quanto è il grado massimo del secondo $\cdot (x^2 + 1) \cdot x^2 = x^4 + x^2 = 10100$

$$\cdot (x+1)(x^2+x+1) = x^3 + x^2 + x^2 + x + x + 1 = x^3 + 1 = 1001$$

- DIVISIONE: (metodo Euclideo)

TRA POLINOMI \rightarrow EUCLEO In BINARIO

$x^6 + x^5$	$x^3 + x + 1$	1100000	1011
$\underline{x^6 + x^4 + x^3}$	$\underline{x^3 + x^2 + x}$	$\underline{1011}$	
$\underline{x^5 + x^4 + x^3}$	$\underline{x^3 + x^2 + x}$	$\underline{111000}$	
$x^5 + x^3 + x^2$	$\downarrow Q(x)$	1011	
$\underline{x^4 + x^3 + x^2}$		$\underline{10100}$	
$x^4 + x^2 + x$		1011	
$\underline{\underline{x}}$		$\underline{\underline{110}}$	
			$R(x)$

IL ricevitore, quando riceve $T(x)$ compie l'operazione $\frac{T(x)}{G(x)}$.

Ma $T(x) = \frac{x^2 P(x) + R(x)}{G(x)} = \underline{\underline{Q(x)}} \rightarrow$ quindi se facendo queste operazioni si genera

N.B. addizione e sottrazione mi equivalgono testo, allora è stato commesso errore durante la trasmissione, se invece è zero non ho la sicurezza che non ci siamo stati sbagli.

Esaminiamo il caso in cui ci fosse errore: il ricevitore riceve $T'(x) = T(x) + E(x)$

dove $E(x)$ è il polinomio che rappresenta l'errore.

Capire come è fatto $E(x)$ permette di rilevarlo e correggerlo. Un tipo di errore può

essere a BURST cioè un 1 iniziale, altri 0 e 1 e poi un 1 finale quindi se avesse

Lunghezza n sarebbe $E(x) = x^i (x^{n-i} + \dots + 1)$ dove i rappresenta la posizione dell'errore dall'estremità destra delle PDN.

Se $E(x)$ fosse uguale o multiplo di $G(x)$ il ricevitore non si accorgerebbe dell'errore.

Anche qui vengono effettuati studi su i possibili tipi di errori e quindi vengono creati $G(x)$ (polinomi generatori) standard con barre probabilità di errore.

MAC (Medium Access Control)

Sono dei protocolli che gestiscono l'accesso al mezzo.

Esistono due tipi di collegamenti di rete: 1. Collegamento Punto-Punto (PPP)

$(0 = 1 + 0) \wedge 1 = 0 + 1 \wedge 0 = 1$ Impiegato nelle connessioni telefoniche

2. Collegamento BROADCAST (canale comune)

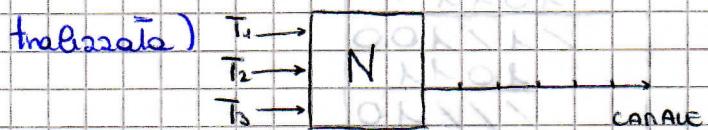
Stato iniziale: $x = 00000000$ e $y = 00000000$ (ogni bit ha durata infinita oppure radio)

Esempio: Ethernet e Wireless LAN

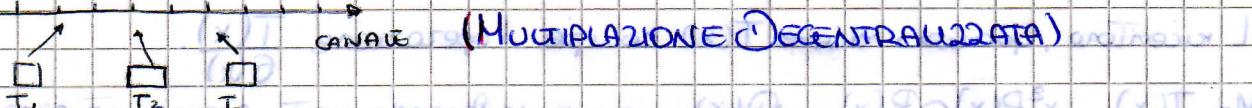
Solo nel caso 2, in cui ci sono più terminali, bisogna occuparsi dell'accesso multiplo.

Infatti i nodi che accedono ad un canale broadcast devono coordinarsi per evitare una collisione, cioè che vengano inviati più frame contemporaneamente da terminali diversi. La conciliazione può avvenire:

- Attraverso un'entità centralizzata che gestisce l'uso dell'essere (Multiplicazione Centralizzata)



- Autoregolazione: cioè i nodi distribuiti accedono al canale in maniera autonoma



Nel secondo caso sono necessari PROTOCOLLI DI ACCESSO MULTIFIO che fissino le regole per l'invio di pacchetti su un canale Broadcast da parte dei terminali, nodi o stazioni.

I protocolli sono suddivisi in due categorie

1. A suddivisione del canale (canalizzazione statica):

Dividere il canale in parti più piccole (slot di tempo, frequenza o codice) e ogni slot è assegnato in maniera esclusiva ad un nodo.

2. Ad Accesso Dinamico (le risorse vengono frazionate in base alle esigenze):

Questa categoria è suddivisa in:

a. Ad Accesso Controllato (Controlled Access): vi è un controllo preventivo che impedisce la trasmissione su un'area che è già occupata. I nodi trasmettono durante il proprio turno, il quale non ha una durata prefissata.

b. Ad Accesso Casuale (Random Access): I nodi inviano, senza controllare, sul canale

Ciò comporta delle collisioni e quindi le stazioni coinvolte ritrasmettono ripetutamente i pacchetti.

ESEMPI: (DA SAPERE!)

Protocolli di Suddivisione del Canale (1):

- TDMA (Time Division Multiple Access) [Accesso Multiplo a Divisione di Tempo]

Funziona come il TDM (Time Division Multiplexing), cioè il canale è suddiviso in diversi slot di tempo ed ognuno di essi è assegnato ad un utente. La grande differenza tra TDM e TDMA è che nel primo tutti gli utenti accedono alle risorse alla volta di un unico nodo che ha il compito di controllare che non ci siano sovrapposizioni, nel secondo invece sono le stazioni stesse che autocontrollano il proprio accesso.

Con questi protocolli ci possono essere sprechi di risorse.

- FDMA (Frequency Division Multiple Access) [Accesso Multiplo a Divisione di Frequenza]

Equivalenti al TDMA ma si opera sulle frequenze. Un esempio di FDMA è per radio, invece di FDM è un telefono che gestisce due frequenze e decide su quale delle due trasmettere la chiamata.

- CDMA (Code Division Multiple Access) [Accesso Multiplo a Divisione di Codice]

Molto diffuso in America. Gli utenti trasmettono nello stesso tempo e sulla stessa frequenza. Per differenziare le comunicazioni, i dati vengono moltiplicati per un codice, chiamato CHIP, individuale per ogni trasmissione.

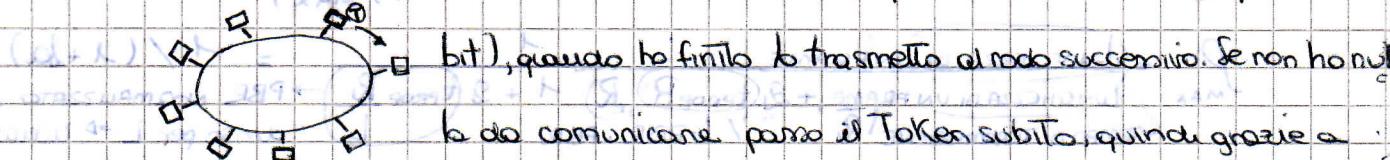
E' come ad una festa in cui tutte le coppie parlano, ma tutte comunicano con un linguaggio diverso. I diversi idiomi sono i codici.

Protocolli Dinamici Ad Accesso Controllato (2.a):

- TOKEN-Passing (Descritto anche in seguito)

Ogni utente trasmette uno per volta ma il controllo non avviene in maniera centralizzata (da un unico nodo) bensì distribuita (ogni stazione si auto-regola).

Crema una configurazione di rete ad anello (TOKEN-RING) in cui accedono le varie stazioni in maniera distribuita. Posso trasmettere solo chi possiede il TOKEN (sequenza di



questo protocollo non vi è rischio di collisione ma anche di spreco di risorse. Il TOKEN-RING (Rete ad anello che implementa il Token-Passing) è un esempio di protocollo ad accesso multiplo, controllato (si trasmette uno per volta), distribuito (gli utenti accedono direttamente al canale) e dinamico (i turni di trasmissione non sono prefissati).

Protocolli Dinamici ad Accesso Casuale (2 b):

Una stazione trasmette quando è pronta su un canale broadcast (cioè comunica ai tutti). Però si possono verificare delle collisioni se due nodi comunicano contemporaneamente, quindi bisogna rtrasmettere i dati.

C'è rischio di collisione ma sono i più facili da implementare ed in più ogni trasmissione arriva alla massima velocità del canale (R bit/s → bit Rate).

- ALOHA

- SLOTTED ALOHA

- CSMA, CSMA/CD, CSMA/CA

} Descritti in seguito

Protocolli ad Accesso Dinamico Casuale

Un protocollo è caratterizzato da un parametro chiamato PROBABILITÀ BANDA RITARDO (PBR):

$$PBR = \frac{\text{banda del canale (bit)}}{\text{s}} \cdot \text{ritardo di propagazione (bit)} = \frac{\text{(bit)}}{\text{(bit)}}$$

Questo valore rappresenta il numero di bit che si trovano contemporaneamente sul canale. Il PBR è legato alla difficoltà di coordinamento e quindi alle collisioni.

Esempio di collisione

1° NO COLLISIONE



A trasmette all'istante $t=0$ e B non trasmette fino all'istante $t=t_{\text{prop}}$

2° COLLISIONE



A trasmette all'istante $t=0$ ma B inizia a parlare a $t=t_{\text{prop}}$. E quando ancora non aveva ricevuto il messaggio da A, quindi dopo t_{prop} di tempo e sarà una collisione. A si accorgere della collisione a $t=2t_{\text{prop}}$.

Il tempo $2t_{\text{prop}}$ si chiama INTERVALLO DI VULNERABILITÀ ed ha un peso importante nell'EFFICIENZA DI UTILIZZAZIONE DEL MEZZO (P_{MAX})

$$P_{\text{MAX}} = \frac{\text{LUNGHEZZA FRAME}}{\text{BIT RATE}} \cdot \left(\frac{1}{L + 2t_{\text{prop}}} \right) = \frac{L}{L + 2t_{\text{prop}}R} = \frac{1}{1 + 2 \frac{t_{\text{prop}}R}{L}} = \frac{1}{1 + 2a}$$

PBR normalizzato ad L cioè dunque per $L \rightarrow \infty$ chiamato a

Se si trasmiscono su un canale ideale $P_{\text{max}} = \frac{1}{1+Q}$ $\Rightarrow Q = 2 \text{ ebit/s}$

$P_{\text{max}} = \frac{1}{1+2\alpha} < 1$ E' max perché si misura al massimo dell'utilizzo

aumenta se: - t_{prop} aumenta $\rightarrow t_{\text{prop}} = \frac{\text{distanza da terminali}}{\text{velocità di propagazione}}$ \rightarrow distanza aumenta
- R aumenta (si riduce il tempo utile $= \frac{L}{R}$)

Affinché si abbia la certezza che il messaggio arrivi, la lunghezza dei frame deve avere una dimensione minima cioè $L_{\text{min}} (\text{bit}) \geq 2 \cdot t_{\text{prop}} \cdot R$. Con queste misure ha la sicurezza che sia passato il tempo di vulnerabilità.

Il THROUGHPUT MASSIMO cioè il bit rate effettivo $R_{\text{eff}} = \frac{L}{t_{\text{prop}} + 2t_{\text{prop}}} = \frac{1}{3} R (\text{bit/s})$

Se è presente il tempo di rivelazione di collisione (t_c) $\rightarrow L_{\text{min}} \leq 2t_{\text{prop}} + t_c$

• Protocollo ALOHA

Sviluppato da un ricercatore all'università delle Hawaii. Fa parte della famiglia dei protocolli ad accesso dinamico casuale.

Il principio su cui si basa è molto semplice: quando il frame è pronto, trasmetti.

Naturalmente c'è un alto rischio di collisione. La stazione che trasmette aspetta un

ACKNOWLEDGE (avvertimento) dal destinatario, per un certo tempo (Timeout);

esso è un ampio intervallo di ricezione e quindi assenza di collisione.

Se invece, allo scadere del Timeout, non riceve alcuna ACK, calcola il Tempo di ritrasmissione (BACKOFF TIME) ed allo scadere di questo, rinnova le informazioni.

Le collisioni possono avvenire anche tra comunicazioni diverse (altri utenti) che hanno

inviato i loro pacchetti contemporaneamente o qualche istante prima.

PRIMA TRASM.

BACKOFF TIME

TRITRASMISSIONE

to-x

to

to+x

to+x+2t_{prop}

to+x+2t_{prop}+3

ISTANTE DI INNODI

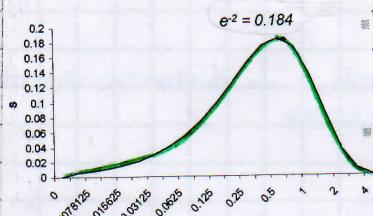
Intervallo di vulnerabilità (momento durante il quale si può ricevere la trasmisone).

Per calcolare il Throughput, cioè la capacità di trasmissione effettivamente utilizzata esegui il prodotto fra: P_{load} , cioè il numero medio di tentativi di trasmissione effettuati in X secondi, e la probabilità che la trasmessa non subisca collisioni.

Quindi $S = \text{Throughput} = P_{\text{load}} \times \text{Probabilità} = G \cdot P_{\text{successo}}$. Si può dimostrare

che per un n° di tentativi che tende ad oo $P_{\text{successo}} = e^{-2G}$ ($e = \text{n}^{\circ}$ di Nepero = 2,7182)

Quindi $S = Ge^{-2G}$, se ci segniamo questa funzione



Possiamo notare che se G è basso, S anch'esso è basso ma

è in crescita. Quando accade a $G=1$, $S=e^{-2}=0,184$. E poi
avrà inizio a decrescere. E quindi per valori elevati di G ,

S , il throughput o anche efficienza, tende a 0 in quanto si verificherebbero un maggior numero di collisioni.

In conclusione l'efficienza massima (o throughput) in un protocollo ALOHA è del 18%.

• PROTOCOLLO Slotted ALOHA

Si basa sullo stesso principio del protocollo ALOHA, però queste volte i pacchetti hanno tutta stessa dimensione, ma soprattutto il tempo è suddiviso in slot. Le stazioni che accedono al canale sono sincronizzate quindi conoscono l'istante temporale d'inizio e la durata dello slot. Il funzionamento è semplice: appena il frame è pronto, il nodo lo spedisce sul canale all'inizio dello slot. Se non si riferiscono collisioni, si prepara ad inviare il secondo frame nell'intervallo successivo. Se invece si verifica una collisione, il nodo fa ruire prima della fine dello slot e ritrasmette il suo pacchetto durante gli slot successivi.



Gli invii avvengono in maniera casuale, l'unico controllo avviene sull'anno del pacchetto all'inizio del frame.

Questo protocollo consente l'invio dei frame alla massima velocità del canale però molte frasi di tempo rimangono inattive ed altre sprecate a causa delle collisioni.

Calcolando la probabilità di successo con un numero elevato di utenti e quindi un valore

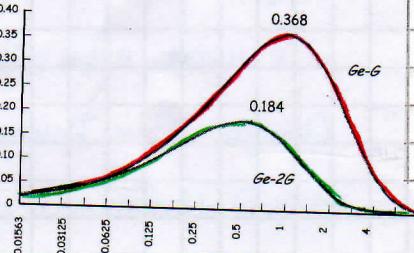
elevato di invii si ha che $P_{\text{success}} = e^{-G}$ (La mancanza del 2, comenell'ALOHA, è dovuta alla sincronizzazione che rende meno probabile la collisione).

Perciò $S = G \cdot P_{\text{success}} = Ge^{-G}$.

Ridisegnando la curva, possiamo verificare che S al massimo

può raggiungere il 36% circa il doppio del throughput ALOHA.

(È un protocollo molto semplice e solitamente viene usato).



to in reti con pochi utenti e con poco traffico.

Soltanente lo Slotted Aloha è usato da altri protocolli per permettere ai multipli utenti di prenotarsi. Cioè l'asse dei tempi è formato da cicli, ognuno dei quali è suddiviso da dei mini-slot per effettuare le prenotazioni e da degli slot per la trasmissione vera e propria. Le stazioni usano il protocollo slotted ALOHA per userarsi. L'uso esclusivo del resto del ciclo per inviare i propri frame.

• PROTOCOLO CSMA

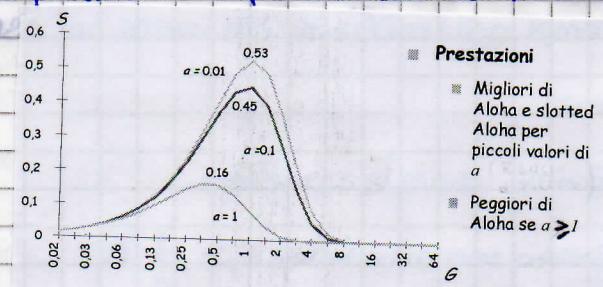
CSMA (Carrier Sensing Multiple Access) cioè Accesso Multiplo A Rilevazione della Portante (mezzo) è un protocollo che applica il principio: "ascolta il mezzo prima di trasmettere". Il nodo trasmette solo se sa che il canale è libero. Se, invece, è occupato calcola l'algoritmo di backoff e quindi attende.

Questo protocollo, purtroppo non è esente da collisioni, infatti il mio pacchetto ha un tempo di trasmissione pari a t_{prop} , se un nodo inizia a trasmettere qualche micron secondo prima di $t=t_{prop}$ i frame si sovrappongono.

Se il canale è occupato ci sono due algoritmi che i nodi applicano per rilevare il canale:

- a- 1-persistent CSMA:
 - non appena il canale è libero inizia la trasmissione
 - non esistono "tempi morti", c'è un basso ritardo ma anche bassa efficienza
 - è utile quando la rete ha pochi utenti

Le sue prestazioni dipendono da α (formula efficienza) = $\frac{PBR}{L \cdot t_{prop} \cdot R}$

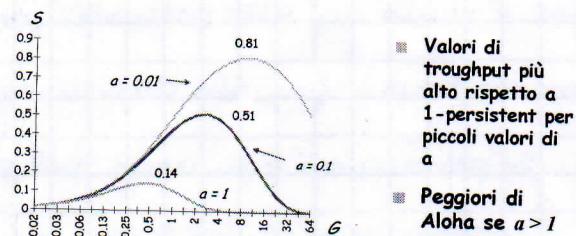


b- Non-persistent CSMA: il nodo applica un backoff, quindi attende un paio di tempi e poi effettua un nuovo carrier sensing

- c'è un alto ritardo ma anche un'alta efficienza

• conveniente in presenza di molti utenti

Nel grafico è riportata la variazione di S al variazione di a .



Per avere i vantaggi sia dell'algoritmo 1-Persistent che Non-Persistent, si è pensato un terzo modo:

- p-persistent CSMA:
 - il nodo attende fino a che il canale si libera poi con probabilità p trasmette, con $(1-p)$ attende un breve periodo (mini-slot) ed effettua di nuovo il carrier sensing
 - il ritardo e l'efficienza possono essere modulate.

Nel protocollo CSMA l'efficienza $E = \frac{1}{1 + 2t_{\text{prop}} \cdot R \cdot e}$

↳ n° di Nepero che rappresenta il massimo dei tentativi effettuati.

-RIPASSO-

$$\left\{ \begin{array}{l} \text{Il Tempo di trasferimento } \frac{L_{\min}}{R} \geq 2t_{\text{prop}} + t_{\text{RN}} \\ \text{quindi } L_{\min} = 2t_{\text{prop}} \cdot R = 2 \frac{d}{v_p} \cdot R \end{array} \right. \quad \text{Tempo rivelazione collisione (TRASCURABILE)}$$

Il CSMA è stato migliorato nel CSMA with Collision Detection (CSMA/CD).

Il nodo ascolta prima di trasmettere ma anche mentre parla, e se rileva una collisione

smette di comunicare. Inizierà a riunire dopo un intervallo di backoff.

Con questo protocollo si riducono i tempi delle collisioni che invece nel CSMA erano pari ai tempi di trasmissione di un frame.

Affinché ci si accorga di una collisione devo trasmettere

$$\frac{L_{\min}}{R} \geq 2t_{\text{prop}} + t_R \quad \text{e} \quad L_{\min} = \text{lunghezza min di un frame} = 2 \frac{d}{v_p} R$$

Si nota che più è ampia o veloce la rete tanto più il frame deve essere lungo, ciò è dovuto al fatto che neve più tempo per rivelare eventuali collisioni.

Un esempio di standard che utilizza il protocollo CSMA/CD è Ethernet.

Ethernet è molto usato nelle reti LAN sia nella sua forma Wireless (Wifi) sia Wired (Cavo cablato)

E' 1-persistent CSMA, ha un bit rate standard di 10 Mbit/s e un Tprop pari a 51.2 microseconds. Facendo i calcoli si trova che la lunghezza di un frame minimo è di 512 bits, cioè 64 byte slot, la distanza massima è 2,5 Km.

L'algoritmo per calcolare il tempo da attendere per il rinvio dei frame si chiama:

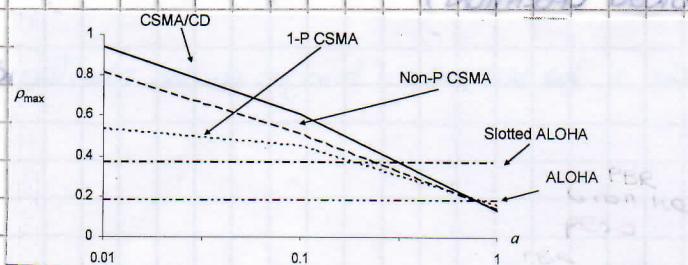
TRUNCATED BINARY EXPONENTIAL BACKOFF:

Il tempo di backoff è scelto random, dopo l'attesa, tra i valori $\{0, 1, \dots, 2^k - 1\}$ dove $k = \min\{n, 10\}$.

Ad esempio:

$n=1 \rightarrow \{0, 1\}$ posso scegliere tras subito (0) o dopo uno slot

CONFRONTO TRA PROTOCOLLI AD ACCESSO MULTIPLO CASUALE



Alotta e Slotted Alotta non vengono condizionati dal PBR.

Per valori piccoli di a il migliore è CSMA-CD, per valori grandi sono Slotted ALOHA e ALOHA.

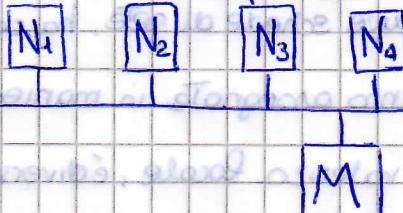
Protocolli ad Accesso Dinamico Controllato

Sono dei protocolli abbastanza complessi ma abbassano lo spreco delle risorse (alto nei protocolli di suddivisione del canale) ed evitano le collisioni (presenti nei protocolli ad accesso casuale).

Protocollo POLLING

Un nodo, chiamato Master, decide il turno di chi trasmette. È dinamico in quanto le sessioni diurne non sono prefissate e distribuito in quanto i frame vengono immessi direttamente sul canale.

Grazie a questo protocollo si eliminano le collisioni e gli slot vuoti ma si introduce il rischio di polling e soprattutto il funzionamento dipende dal master, quindi se si guasta il canale resta inattivo.



Protocollo TOKEN-PASSING

Tra i nodi circola un Token, cioè un messaggio di controllo, seguendo un ordine prefissato. La stazione che possiede il Token parla per un tempo non deciso e primi (dinamico).

Quando ha terminato la sua comunicazione lascia il token al nodo successivo.

Lo svantaggio è che un guasto di un terminale potrebbe mettere fuori uso l'intero canale. Come però, invece, è che non è centralizzato ma soprattutto ha un'alta efficienza.

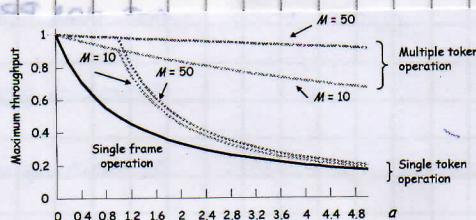
$$P_{\max} = E = \frac{\text{Tempo Utile}}{\text{Tempo Utile} + \text{Passaggio Token}} = \frac{T_U}{T_U + \tau}$$

Il token naturalmente deve essere conosciuto da tutte le stazioni. Esso è un insieme di bit; all'inizio il terminale, che è in fase di ascolto, è settato su FREE TOKEN, quando è il suo turno, modificando uno o più bit, passa a BUSY TOKEN ed inizia ad inviare. Appena ha finito ritorna a Free.

Questo protocollo è stato ampliato con l'introduzione di più token e quindi con passaggi e mini di pacchetti paralleli (Multi-Token operation).

Dal grafico si evince che al crescere di α , si ha sempre più bisogno di una strategia.

MULTI-TOKEN.



INDIRIZZAMENTO DELLO STRATO DI CONNESSIONE: Indirizzi MAC

Quando un terminale opera su un canale broadcast, i suoi pacchetti vengono fisicamente ricevuti da tutti, però in realtà sono indirizzati solo a uno o ad alcuni nodi.

Per capire a chi sono destinati i frame, nelle intestazioni di controllo di strato due viene inserito un MAC-ADDRESS che identifica univocamente le schede di rete del destinatario.

Nello standard Ethernet l'indirizzo MAC è formato da 6 byte esadecimale.

L'indirizzo broadcast (destinato a tutti) è FF-FF-FF-FF-FF-FF

Una società che produce schede di rete, ha comprato un blocco di spazio di indirizzi (come le targhe) che saranno assegnate in maniera unica.

L'indirizzo MAC ha validità locale, è diverso per ogni scheda (di rete, wifi...).

Quando comunque è difficile sapere il Mac-Address dell'interlocutore, qui entra in gioco il livello superiore, quello di rete, che introduce un nuovo indirizzo, chiamato IP. Quest'ultimo è come l'indirizzo postale (Mac ≈ codice fiscale).

quindi permette la comunicazione a livello di rete e non locale.

L'IP (Internet Protocol) address è una serie di numeri anche se difficile da ricordare. Per comunicare l'utente inserisce l'INDIRIZZO MNEMONICO (sito, email, ...), il DNS, cioè un protocollo, lo trasforma in IP address poi il passaggio da IP a Mac viene fatto da ARP (Address Resolution Protocol).

Funzionamento ARP: (Vedi "APPROFONDIMENTO ARP")

- L'indirizzo IP individua un host che fa parte di una sottorete LAN.

- ARP chiede con un messaggio broadcast chi è quell'IP specifico e il terminal coinvolto risponde con il proprio indirizzo Mac. (Tramite un frame di dimensione standard)

Quando riceverà delle richieste, vengono registrate in una Tabella ARP.

Il protocollo ARP è efficiente, ma facilmente ingannevole, se un host risponde positivamente alle sue domande, anche se il suo IP è un altro rispetto a quellorichiesta, riceverà tutti i pacchetti inviati dal mittente.

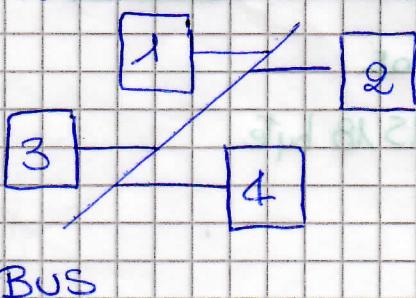
Senza Mac e senza IP non possiamo comunicare con altri Terminali.

STANDARD SPECIFICO PER LE RETI LAN: ETHERNET

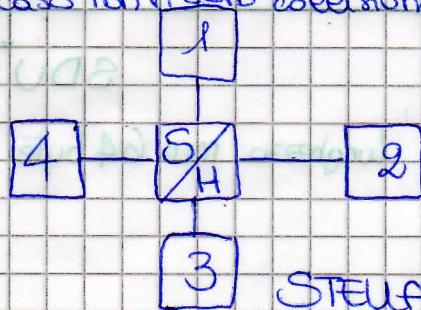
Lo sviluppo di Ethernet inizia negli anni '70 con il progetto ALOHA. Viene standardizzato nel 1985 da IEEE, la sua famiglia è 802 e .3 è la sua specifica. In questo documento nasceva Ethernet ed aveva una velocità di 10 Mbps.

Negli anni successivi lo hanno migliorato ed oggi sono molto diffuse LAN che usano lo Gigabit Ethernet.

La topologia originale di una rete locale era quella al bus, dove tutti i terminali erano collegati ad un mezzo comune. Successivamente essa è stata sostituita dalla topologia a stella, in cui il singolo capo dell'host è collegato ad un elemento (hub o switch), il quale si trova al centro della stella e ha la funzione di commutazione delle frame sui rami. In questo caso non vi sono collisioni.



BUS



STELLA

Ethernet utilizza il protocollo CSMA/CD, in tutte e due le topologie.

Lo slot time è il tempo necessario per inviare un impulso alla massima di stessa.

Si era stato fissato a 51,2 μ sec, quindi la durata dell'impulso di 512 bit ad una velocità di 10 Mbit/s.

Il terminale che trasmette i propri pacchetti è in attesa di una collisione, continua ad inviare dei bit per far rivelare anche agli altri utenti del problema. Il segnale che continua ad inviare si chiama JAM e il MAC Jam Time è il tempo di rivelazione. Un altro modo per accorgersi che n'è stato uno scontro è il fatto che l'utente che invia ha potenza più alta. Lo standard Ethernet stabilisce anche la potenza "normale".

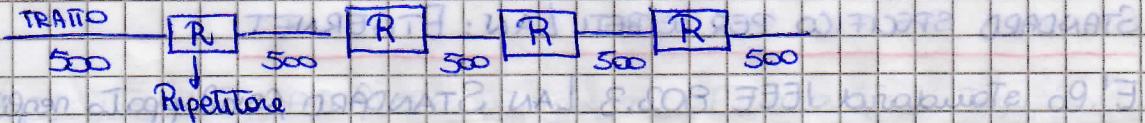
PARAMETRI ORIGINALI:

$$P_s = 10 \text{ Mbit/s}$$

Lunghezza minima Frame = 512 bit = 64 byte

$$\rightarrow \text{Slot Time} = 51,2 \mu\text{sec}$$

ESTENSIONE MAX = 5,12 Km \rightarrow poter avere una topologia del genere



$$5 \text{ tratti da } 500 \text{ metri} + 4 \text{ ripetitori} = 2500 \text{ m} (\times 2 = 5 \text{ Km})$$

Se aumento il bit rate, devo diminuire la lunghezza massima.

UNITÀ DI DATO: FRAME ETHERNET

PREAMBLO	S/D	DEST ADDRESS	SOURCE ADDRESS	LENGTH	DATA	PAD	CRC
7 byte	1 byte	6 byte	6 byte	2 byte	DATI DI UN FRAME	4 byte	

PCI, parte di

informazione

SDU, parte dati

Lunghezza min 64 byte max 1518 byte

Ethernet implementa un servizio senza connessione, cioè il mittente invia i propri dati senza aspettarsi nemuna risposta dal ricevitore. Questo metodo non è affidabile, in quanto potrebbero verificarsi delle perdite di informazione, perciò è molto semplice eseguire.

Protocollo CSMA/CD di ETHERNET (Ripasso)

1. Scheda di rete prepara un frame Ethernet

2. Se il canale è libero inizia ad inviare, altrimenti attende che si liberi.

3. Durante la trasmissione verifica se sono presenti eventuali segnali provenienti da altri adattatori.

4. Se non rileva disturbi allora il pacchetto è stato spedito.

5. Se riceve ulteriori segnali allora c'è stata una collisione e spedisce un SEGNALE DI DISTURBO (JAM) dopo aver interrotto l'emissione del frame. Il JAM è un avviso a tutti gli adattatori che stanno trasmettendo, è lungo 48 bit.

6. La scheda di rete calcola il tempo di BackOff e pronta a riinviare.

Livello Fisico

Ethernet è uno standard, che oltre a definire il protocollo MAC e il formato delle frame (STRATO DI COLEGAMENTO), stabilisce i mezzi trasmissivi (STRATO FISICO).

Le sigle indicano la velocità di trasmissione, la frequenza e il mezzo (cavo o fibra).

• Il primo Ethernet (10 Mbit/s) : 10 base 5, 10 base 2, 10 base T, 10 base Fx
in memoria della lunghezza massima bandi mezzo = cavo Fibra Optica

• Fast Ethernet (100 Mbit/s) : 100 base T, 100 base Fx

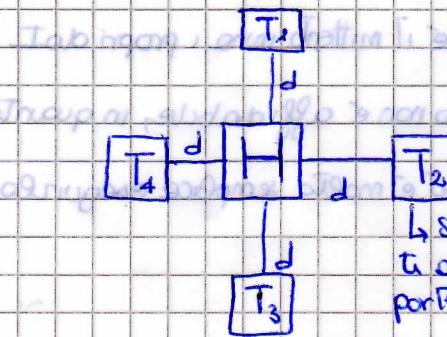
↳ prevista solo tipo logico a stella, standard più usato

• Gigabit Ethernet (1 Gbit/s) : 1000 base Sx, 1000base Lx, 1000base X, 1000base T

↳ la lunghezza dello frame è aumentata a 512 byte, il CSMA-CD è praticamente abbandonato

TOPOLOGIA A STELLA

All'inizio del ramo era posizionato un HUB, che in pratica è un ripetitore. Esso lavora solo a livello fisico, riceve il segnale e lo trasmette a tutte le interfacce perciò rimane un dominio broadcast, anche chiamato unicast o collisione.



L'estensione della rete, cioè la massima

distanza che intercorre tra due terminali, è $2d$.

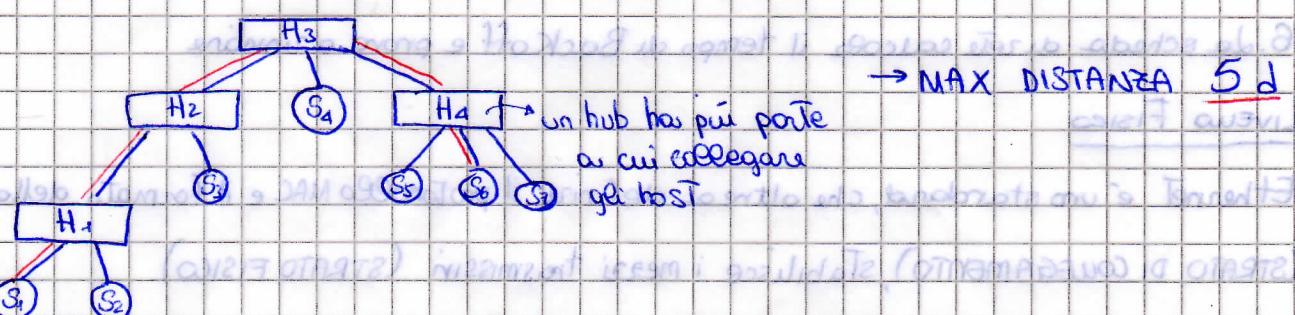
Si accorgono di eventuali collisioni ma le ignora.

Permette di aumentare le dimensioni di una LAN ma bisogna sempre rispettare i limiti teorici imposti da CSMA/CD e i limiti del numero di repetitori massimo utile.

FAST ETHERNET (100 Mbit/s) d_{max} è 100 m

Successivamente gli hub furono sostituiti da dispositivi leggermente più intelligenti: gli SWITCH. Essi permettono di unire più domini.

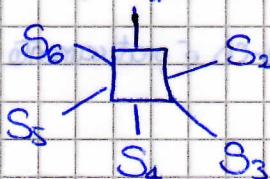
Con gli hub si crea un solo dominio.



Lo switch oltre allo strato fisico, possiede anche lo strato di collegamento quindi ha la capacità di cogliere ciò che c'è scritto all'interno delle frame.

Quindi riesce a capire l'indirizzo MAC e perciò a quale Terminal è destinato il pacchetto. In più, implementando lo strato 2, opera il CSMA/CD entro le collisioni. Eseguendo queste azioni crea diversi domini di collisione, uno per ogni porta.

Le domande: \times Se è un hub → 1 solo dominio



Se è uno switch → 6 domini

Per inviare solo all'indirizzo giusto, lo switch ha bisogno di una TABELLA DI INOLTRO chiamata TABELLA DI COMMUTAZIONE (SWITCH TABLE), in cui memorizza i vari indirizzi MAC e associati alle porte.

La tabella è composta da:

(97c) indirizzo MAC, interfaccia, TTL

Indirizzo MAC	Interfaccia	TTL
---------------	-------------	-----

Intervallo di tempo (time to live)

time to live (tempo di vita)

ogni tot scade e si cancella la regola (utile se eventualmente spostò il terminale su un'altra porta)

Questo non è compilato dall'utente ma direttamente dallo switch (chiamato anche BRIDGE) tramite il PLUG-AND-PLAY.

Compilazione:

• All'inizio è vuota, arriva un frame e memorizza il MAC-ADDRESS della porta mittente

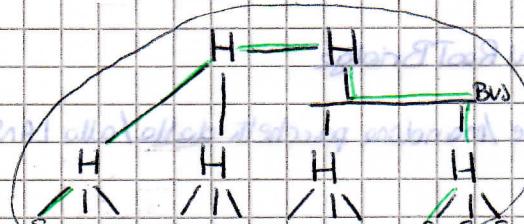
• Invio a tutti il pacchetto

• Memorizza un nuovo indirizzo MAC quando riceverà un dato da una nuova stazione

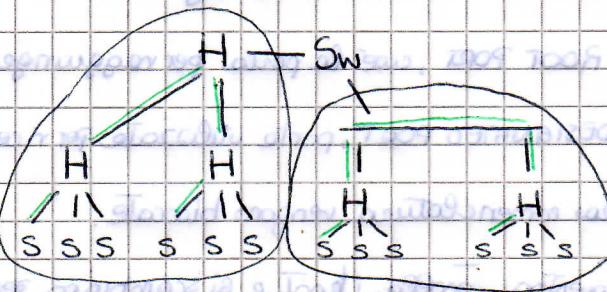
La tabella si dice in regime quando ha memorizzato tutti gli indirizzi.

Gli switch permettono di estendere la rete senza limiti, grazie al fatto che considerano ogni porta un dominio di collisione diverso

SOLO HUB



SWITCH - Hub



1 solo dominio di collisione

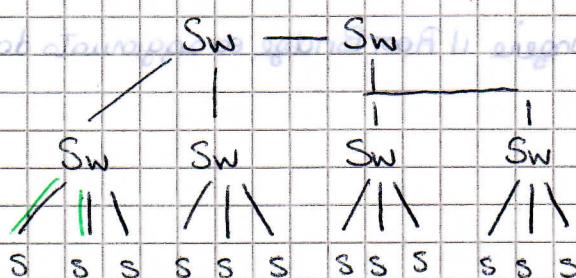
1 sola efficienza

d_{MAX}

2 domini di collisione

2 efficienze 2 distanze max

Solo SWITCH



1° efficienze

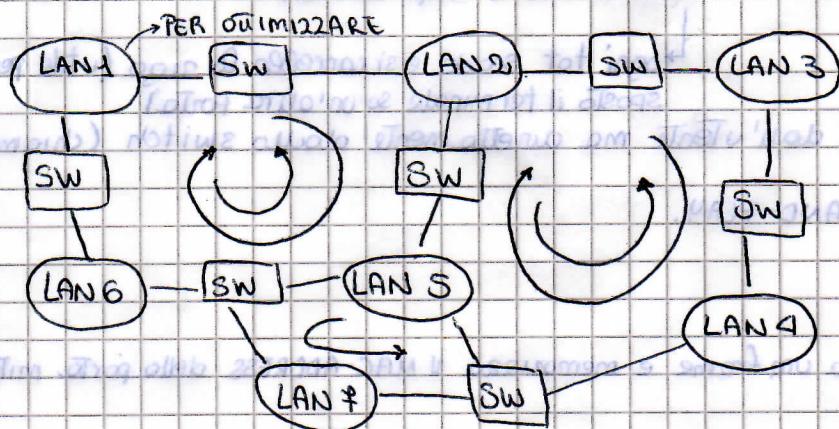
d_{MAX} (dove si recalcola per ogni dominio
ma considera il vero perimetro)

Per evitare che le tabelle diventino troppo estese consideriamo le varie reti LAN. Per

identificare considero un indirizzo di livello superiore (STRATO DI RETE), per questo sono stati introdotti i ROUTER che operano anche a livello 3.

Spanning Tree Protocol (STP)

L'STP è un protocollo introdotto per evitare dei cicli infiniti di instradamento. Alla fine il pacchetto arriva anche al destinatario, ma prima ha girato indefinitivamente per la rete.



Lo spanning tree protocol fa parte degli IEEE 802.1D e ha il compito di disattivare quei link ("tagliare i rami") che creano loop nella rete. In questo modo ha il compito di determinare un albero che copre l'intera rete.

Per far ciò bisogna compiere 3 fasi:

1. Determinare lo switch RADICE (Root Bridge)
2. Selezionare le ROOT PORT, cioè la porta per raggiungere il Root Bridge
3. Selezionare le DESIGNATED PORT, porte utilizzate per ricevere / mandare pacchetti dalla / alle LAN

Tutte le porte senza nomenclatura vengono bloccate.

Per implementare questo protocollo, i bridge si scambiano periodicamente delle trame di controllo specifiche chiamate BRIDGE PROTOCOL DATA UNIT (BPDU). Esse contengono le informazioni fondamentali:

- a. Root_ID (nome identificativo del Bridge candidato a diventare Root bridge)
- b. Switch_ID (identificativo del bridge che trasmette le BPDU)
- c. Root Path Cost (conto totale del percorso per raggiungere il Root Bridge ed aggiornato da ogni switch)

Ora vediamo in dettaglio le 3 fasi:

1. ELEZIONE DEL Root Switch:

Ogni switch, all'inizio, crede di essere la radice quindi trasmette una BPDU con Root_ID = SWITCH_ID. Se riceve una BPDU da un altro bridge, verifica se il suo identificativo è minore del Root_ID contenuto. Se così fosse, cambierebbe il campo interendoci il proprio nome,

altrimenti, rtrasmetterebbe le BPDU, modificando solo lo switch-id, ai nodi adiacenti.

Quindi il principio su cui si basa è: "Il switch con identificativo più basso, vince"

2. SELEZIONE Root-Port

Uno switch ogni volta che riceve una BPDU, somma al campo ROOT PATH COST il costo percorso associato alle porte di ricezione. La root port è la porta da cui ricevere pacchetti che partono dalla radice percorrendo il percorso con costo minimo.

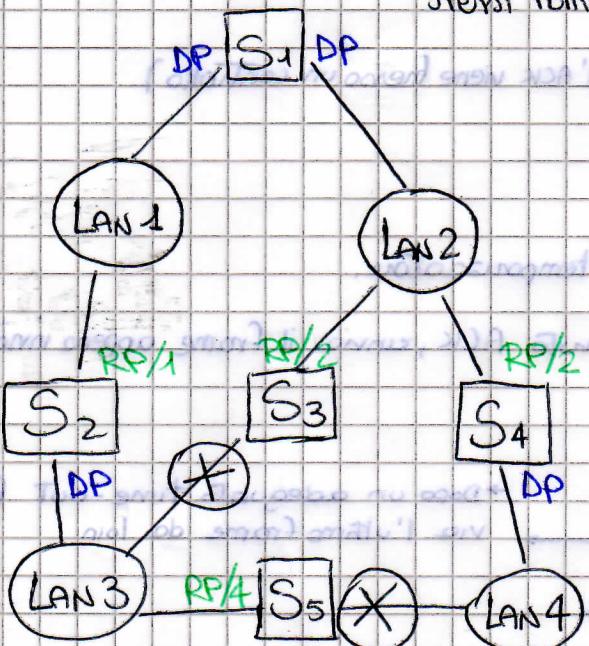
Il costo potrebbe dipendere da quante LAN si attraversano.

3. SELEZIONE DESIGNATED Port

Sono le porte da cui non si raggiunge la radice, necessarie per raggiungere le LAN ai cui sono interfacciate. Se più switch sono collegati ad una stessa sottorete, la DESIGNATED PORT è quella che percorre un percorso con costo minore.

Le porte senza nomenclatura vengono disabilitate. Ciò non significa che non potrebbe essere riattivata in un secondo momento.

APPlicazione STP (il costo delle LAN e gli identificativi dei bridge sono i loro stessi nomi)



1. Root BRIDGE \rightarrow S1
 2. Root Port / Costo \rightarrow RP/numero
 3. Designated Port \rightarrow DP
- (X) ELIMINAZIONE RAMI

Controlli d'ERRORE

In questo strato esistono due tipi di controlli d'errore. Uno, descritto in precedenza, si chiama FORWARD ERROR CORRECTION, che racchiude il CRC, i bit di parità, ... L'altro, descritto in questo paragrafo, è l'AUTOMATIC REPEAT REQUEST (ARQ), di cui fanno parte tre diversi protocolli (STOP-AND-WAIT, Go-Back N, SELECTIVE REPEAT).

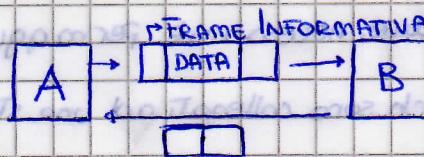
Essi si occupano di richiedere e ricevere un frame accettato con ercione, in più si assicura che la sequenza delle PDU sia in ordine e senza ripetizioni.

Il controllo ARQ non avviene solo nello strato 2, quindi sezione per reazione, ma anche al livello 4 (Trasporto), implementato dal protocollo TCP questa volta operando estremo ad estremo.

Come scritto sopra, delle modalità ARQ esistono 3 procedure:

1. STOP-AND-WAIT ARQ:

• PRINCIPIO: A invia un frame ed aspetta un ACK (riscontro positivo).

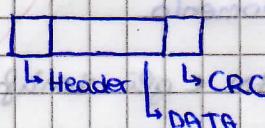


• FRAME CONTROL (ACK / NACK)

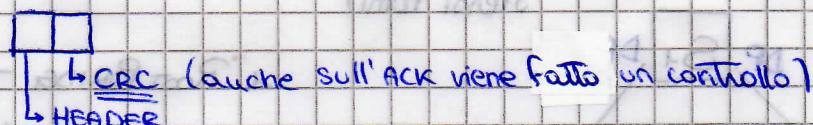
→ riscontro negativo

• COMPOSIZIONE FRAME:

• INFORMATIVA



• CONTROLLO



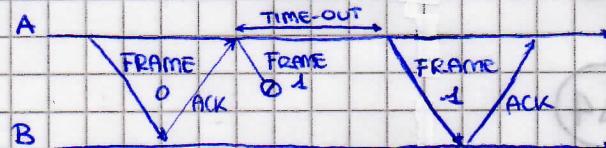
• FUNZIONAMENTO:

- Quando A invia un frame, si attiva un temporizzatore.

- Allo scadere del Time-Out, se non ha ricevuto ACK, invia il frame appena inviato.

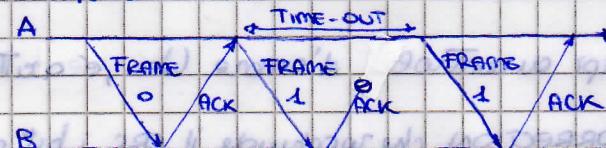
Questo rinvio può avvenire in 3 casi:

a. IL FRAME NON È STATO RICEVUTO



→ Dopo un adeguato time-out A invia l'ultimo frame

b. L'ACK NON È ARRIVATO



→ A non riceve l'ACK relativa alla frame 1 quindi lo rinvia. Però B ha 2 volte la frame 1, se non ci fosse "1" non capirebbe che sono uguali (duplicazione). Per evitare ambiguità nel pacchetto viene inserito un numero di sequenza.

c. ESAURIMENTO PREMATURO DEL TIME-OUT



→ Il time-out è troppo breve, quindi c'è una sbagliata interpretazione degli ACK. Infatti, nell'esempio, A invia Frame 1 prima che B riceva il primo ACK.

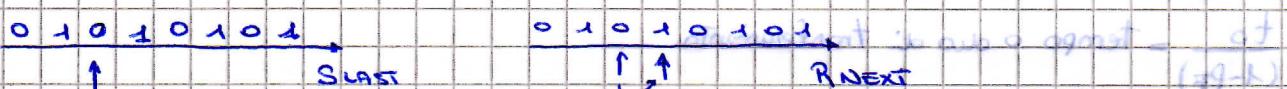
MA QUANDO RICEVE L'ACK HA GIÀ INVIA
PER LA SECONDA VOLTA F2. B RINVIÀ UN
ACK MA A HA APPENA SPEDITO F1, IL QUALE
LE NON È ARRIVATO, MA INTERPRETA L'ACK
RICEVUTO COME UNA CORRETTA RICEZIONE,
PERCÒ INVIA F2.

Dal caso C si deduce l'importanza dei Numeri di sequenza sia nei frame informativi che negli ACK.

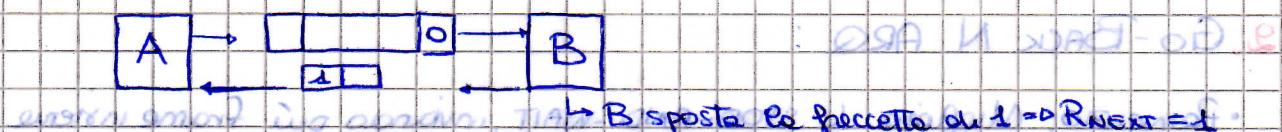
Un modo semplice per numerare le unità è quello ad 1 bit:

I frame possono essere 0 o 1 o 0.

Nella Trans. informativa è inserito il numero SLAST, in quella di Controllo è contenuto RNEXT, cioè il numero di sequenza della prossima frame che il ricevitore si aspetta.

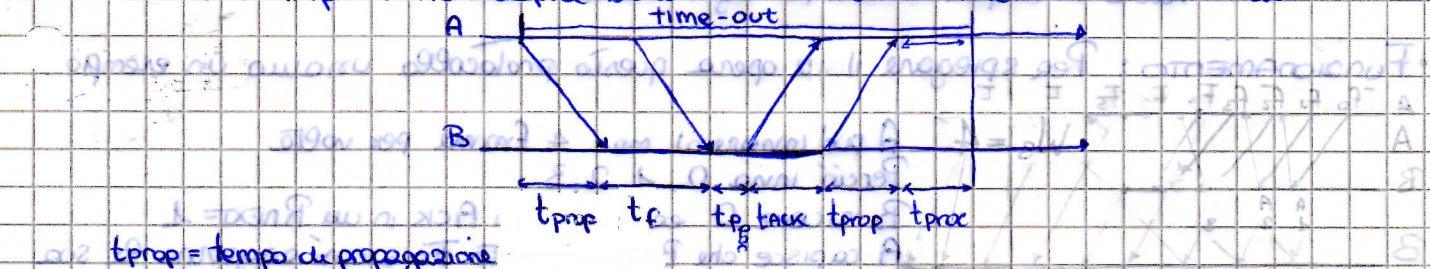


B si aspetta di ricevere 0 ed A è pronto ad inviare 0



Potendo sempre del caso C, è da notare che vi è un doppio invio di una stessa frame non necessario, quindi c'è uno spreco di risorse che potranno essere evitato.

Quindi è importante scoprire bene il minimo di time-out necessario.



t_{prop} = tempo di propagazione

t_f = tempo di frame / t_{ack} = tempo di ack

t_{proc} = tempo di processamento (controllo vari)

t_o = $2 t_{prop} + 2 t_{proc} + t_f + t_{ack}$

In questa formula il time out inizia quando A invia il frame, in alcuni esercizi, il tempo potrebbe essere calcolato alla fine dell'invio, naturalmente sarà specificato nel testo.

Ora calcoliamo l'Efficienza in due situazioni:

1. Efficienza su un canale senza errori

$$E = \frac{\text{TUTTO}}{R} = \frac{n_f}{R}$$

$$\text{TUTTO} = \frac{n_f}{R} + 2t_{\text{PROP}} + 2t_{\text{PROC}} + \frac{n_a}{R}$$

dove n_f = lunghezza frame
 n_a = "ack
 R = bit Rate

L'efficienza diminuirà al crescere della distanza (tempo di propagazione) o del

bitrate. Su lunghe distanze, questa procedura non è buona, ha troppi tempi morti.

Se voglio calcolare il throughput (cioè la percentuale di utilizzo del canale) basta che moltiplico E per R . Nel caso di operazioni estremo ad estremo, R è il Bitrate medio.

2. Efficienza su un canale con errori

In questo caso consideriamo una probabilità P_F che dipende da quanto sbaglia il canale.

Quindi $1 - P_F$ è la probabilità che il frame arrivi senza errori.

$\frac{1}{(1 - P_F)}$ = numero medio di trasmissioni necessarie per avere una frame corretta

$\frac{t_0}{(1 - P_F)}$ = tempo medio di trasferimento

$$E_{CE} = E_{SE} \cdot (1 - P_F)$$

2. Go-BACK N ARQ :

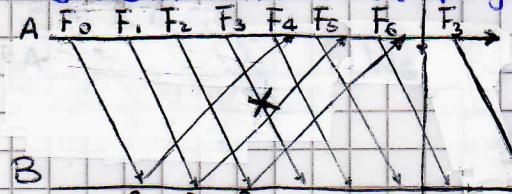
• Principio: Migliora lo STOP-AND-WAIT, inviando più frame insieme.

• Componenti: il numero max di frame inviati insieme è W_s

• Utilizza una finestra di trasmissione di ampiezza W_s

• Poi c'è un temporizzatore per ogni frame inviato.

• Funzionamento: Per spiegare come opera questo protocollo usiamo un esempio



A può inviare al max 4 frame per volta

Perciò invia 0, 1, 2, 3

B riceve 0 ed invia un Ack in cui $R_{NEXT}=1$

A capisce che B ha ricevuto F_0 e così aggiorna la sua finestra

Sposta le sue finestre in modo tale che S_{LAST} (cioè il primo numero) sia lo stesso numero contenuto nell'Ack ricevuto.

Il nuovo numero che compare (4) è chiamato S_{RECENT} ed è quello che A invia.

Dopo l'invio di F_4 , A riceve Ack 2 perciò aggiorna la sua finestra e invia F_5 . Gli invia Ack 3 e manda, dopo l'aggiornamento, F_6 .

Se non ci fossero mai errori, A invierebbe frame senza perdere tempo. Ma nell'esempio F_3 non arriva.

B riceve un qualcosa con errore e non capisce cosa sia quindi non puo' inviare un NACK (il CRC ci dice che c'è errore ma non dove).

A attende l'ACK 4 però non arriva, perciò allo scadere del Time-out relativo ad F_3 invia tutti i frame dell'ultimo

0	1	2	3	RICEVE	1	2	3	4
↓					↓			
S _{LAST}	2	3	4	5	→	3	4	5

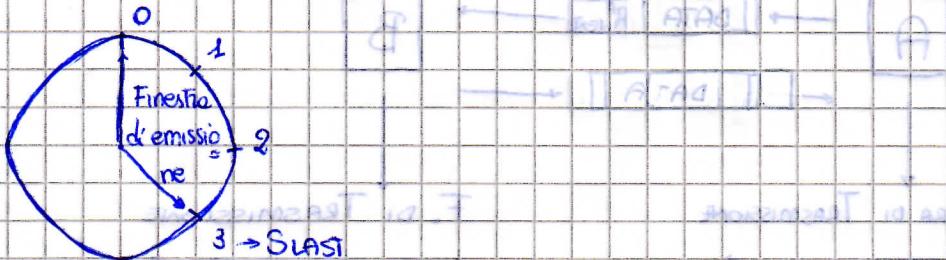
ACK2 ACK3

aggiornamento delle finestre di trasmissione, cioè

[3 4 5 6]

Infatti B, aspettando F_3 , quando riceve F_4, F_5, F_6 li scarca essendo fuori sequenza.

La finestra di trasmissione è chiamata Sliding Window, finestra scorrevole. Può essere rappresentata anche su un cerchio.



Nell'esempio gli ack vengono inviati per ogni frame ricevuto, in alcuni casi, B può decidere di inviare il riscontro dopo un tot di frame (Ack cumulativo).

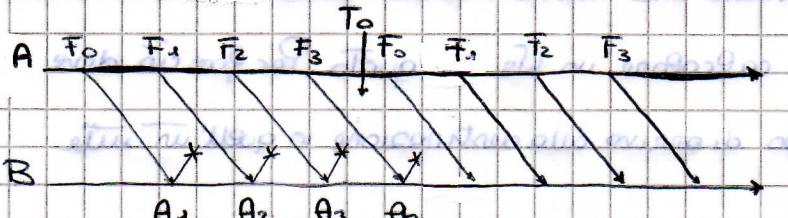
Quando A riceverà l'ack aggiungerà la finestra in modo che $SLAST = \text{numero contenuto nel riscontro}$.

Questo protocollo è un po' scaduto in quanto se il timeout scade A ritorna indietro e invia tutti i frame con numero di sequenza contenuti nella finestra.

Sia nell'esempio ma anche negli esercizi, i frame avranno un numero progressivo che tende all'infinito, invece in realtà la macchina ha a disposizione un certo numero di bit.

Per evitare ambiguità, la finestra massima non sarà uguale a $2^m = M = W_s$ ma bensì a $2^m - 1$.

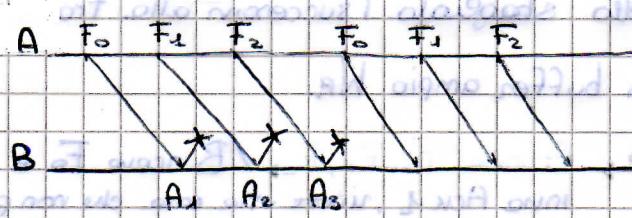
Infatti se $W_s = M = 2^m = 4$



Per B, F_0, F_1, F_2, F_3 sono arrivate correttamente, A invece non lo sa in quanto non ha ricevuto riscontri.

Perciò A, allo scadere del timeout invia tutto, accorciato a B non se ne sono frame nuovi o duplicazioni.

Invece se $W_s - 1 = M - 1 = 2^m - 1 = 3$ (W_s sempre = 4)



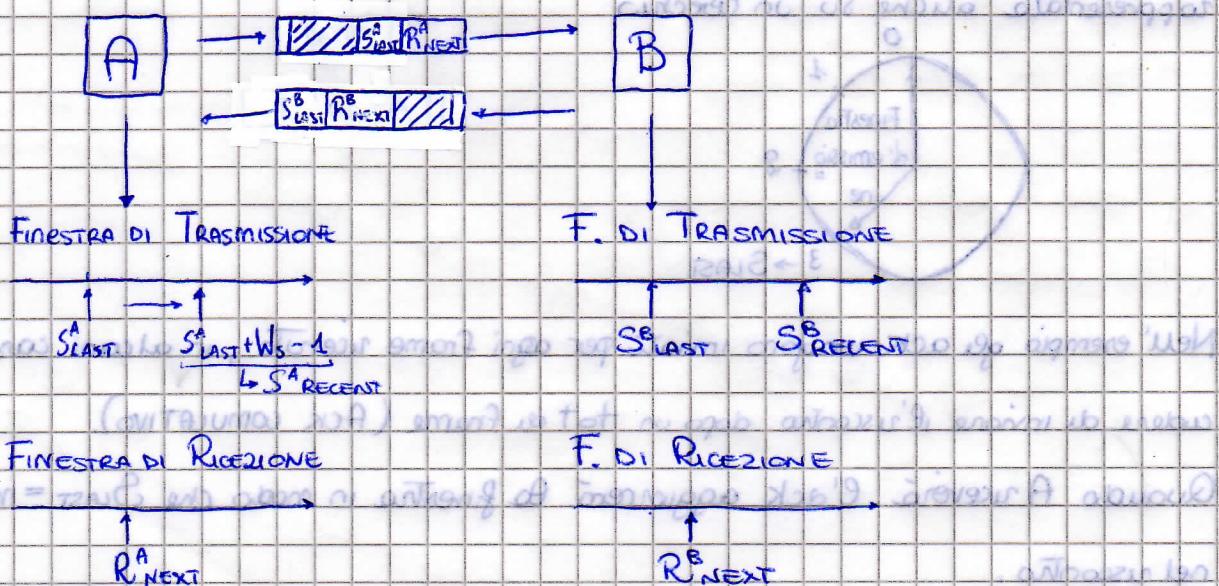
Questa volta A invia solo 3 frame quindi dopo F_2 B si aspetta F_3 . Quando riceve di nuovo F_0 capisce che è una duplicazione quindi lo scarica.

Come detto prima i riscontri possono avvenire dopo un numero (W_s) di frame.

Oltre questo l'ACK potrebbe non essere inviato da solo ma "sulla schiena" di un'altra unità.

Questo modalità è chiamata Piggybacking ed è utilizzata quando sia A che B devono trasmettere a B sia B ad A.

Dunque esistono due finestre di trasmissione, una per B, una per A.



L'efficienza di questo protocollo dipenderà da W_s . Infatti W_s deve essere grande abbastanza da poter tenere il canale occupato durante tutto il TOUT.

Per calcolarlo faccio questo rapporto: $W_s = \lceil \frac{TOUT}{TFRAME} \rceil$ (calcolo la parte intera oltre del rapporto tra Time out e Tempo di frame) (Insomma calcolo quanti frame invio in un timeout).

Oltre a W_s , l'efficienza dipende dal prodotto Banda-Ritardo e dalle probabilità di errore. Go-Back N è molto efficiente se P_f è basso.

Il protocollo TCP ogni volta che inizia una nuova sessione dovrà valutare la distanza e il bit rate e di conseguenza calcolare un W_s adeguato. Per farciò deve stimare il Round Trip Time, cioè il tempo di ritorno alla destinazione in quell'istante.

3. Selective Repeat ARQ

Principio = migliora il Go-Back N, quando deve trasmettere, non invia tutti i frame contenuti nella finestra ma solo quello sbagliato. I successivi alla trame sbagliata sono stati memorizzati in un buffer ampio W_s .

Funzionamento =

$$W_s = 3$$



A invia F_0, F_1, F_2 / B riceve F_0 e invia ACK 1, riceve una cosa che non può leggere quindi non invia nulla, invia F_3 e perciò spedisce un ACK richiedendo F_2 e mette nel buffer F_3 .

Intanto F_1 viene mandato da A, B lo mette nel Buffer e manda un ACK 2.

A ricevere il NACK perciò rispedisce SOLO F_1 , B lo legge e lo rispedisce allo strato superiore insieme a tutti i frame bufferizzati.

Naturalmente ogni volta che A riceve un ACK, aggiorna anche la propria finestra.

In questo protocollo, il ricevitore invia solo un frame sia nel caso che il time out è scaduto (rispettando la trama relativa) sia il caso dell'esempio, gli invia un NACK.

Una domanda che potrebbe sorgere è: "Quale il valore di WR?"

WR, come scritto su, è l'ampiezza delle finestre di ricezione e del buffer di memorizzazione. Ricordando che m è il numero di bit necessari per rappresentare i numeri di sequenza (Parliamo del caso "reale" in cui i frame non hanno un numero di sequenza progressivo che tende all'infinito).

Il valore massimo di WR deve essere tale che $WR + \text{W}_S = 2^m$, solo così non ci saranno ambiguità.

Naturalmente l'efficienza è la migliore rispetto agli altri due protocolli, ma è da sottolineare che è anche il più complicato da scrivere e gestisce più informazioni.

Anche in questo caso E dipende in maniera deterministica dalle probabilità di errore.