4° PAROLA | S.A (source address) | 32 bit

5° PAROLA | D.A | 32 bit.

In tutto sono 20 byte di Header. Per quanto riguarda il TOS abbiamo:

PPP DTR XX

D = DELAY, T = THROUGHPUT, R = RELIABILITY

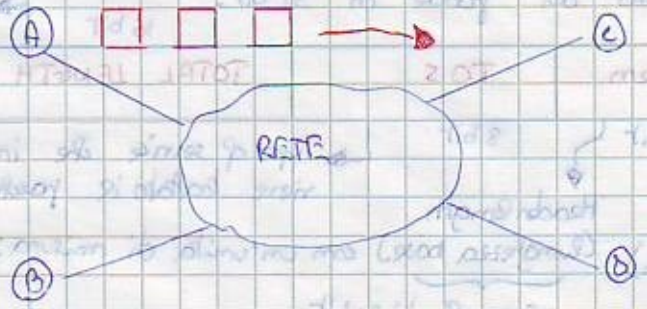
8 livelli di priorità. Un router così sceglie quale pacchetto smistare prima.

Per quanto riguarda TTL (Time to live):

8 bit, decrementati ogni volta che router.

Il campo PROTOCOL è fondamentale. Consideriamo la seguente situazione:

Quando l'IP riceve il pacchetto a quale entità superiore lo manda? Guarda in questo il campo protocol. L'HEADER CHECK SUM invece è un codice di parità che serve in ricezione per vedere se ci sono errori di bit sull'header. Per quanto riguarda il campo FLAGS esiste un flag speciale chiamato D (DON'T FRAGMENT). Supponiamo per esempio di avere:

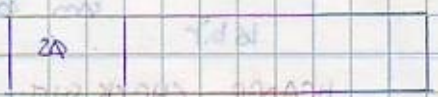


Affinché dalla rete i pacchetti IP.

Quindi la macchina C si vede arrivare dei pezzi di pacchetti che deve riassemblare.

Nasce un problema. Quando a C giunge un frammento esso deve capire di quale pacchetto IP è. Si risolve tale problema analizzando il campo IDENTIFICATION. Se A, B spediscono un pacchetto con lo stesso IDENTIFICATION nello stesso istante, bisogna analizzare il S.A. Quindi si analizza sempre la coppia (SA, IDENTIFICATION).

Vediamo ora come avviene la numerazione dei frammenti. Supponiamo di avere:

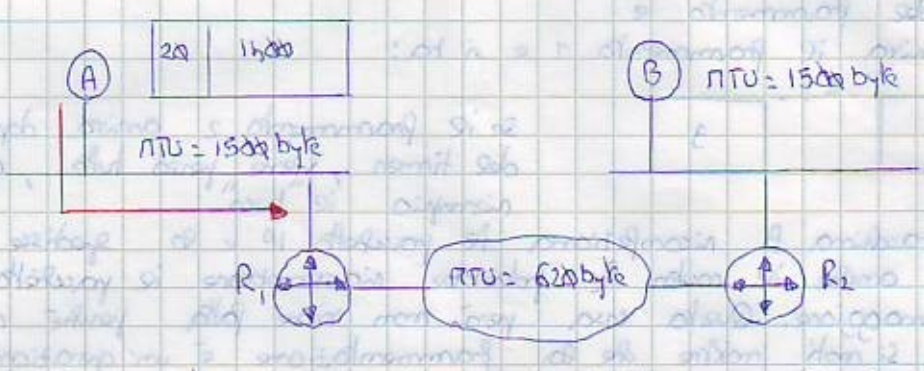


e supponiamo che la MTU del DLC sia di 620 byte. Bisogna frammentare il 1399 pacchetto IP.

Si creano così 3 pacchetti IP diversi (frammenti):

Ogni frammento deve viaggiare indipendentemente sulla rete. Ci chiediamo ora come è e' header dei tre frammenti. E' uguale in tutti e tre i casi tranne che per i campi FLAG, FRAGMENT OFFSET, HEADER CHECKSUM, TOTAL LENGTH. Il campo IDENTIFICATION è uguale.

Per esempio:



Il router R_1 frammenta il pacchetto. Noi ipotizziamo che il bit DON'T FRAGMENT = 0. Quindi:

R_1:	ID	D	MORE	FRAGMENT OFFSET	TOTAL LENGTH
PACCHETTO ORIGINALE	10721	0	0	0	1420
FRAMMENTO (1)	10721	0	1	0	620
FRAMMENTO (2)	10721	0	1	75	620
FRAMMENTO (3)	10721	0	0	150	220

ROUTING TABLE

NB: FRAMMENTO (N) è una mera notazione.

NB: More è un bit che: { 0 se il frammento è l'ultimo del pacchetto; 1 viceversa }

Il flag D e il flag MORE sono i due bit più importanti per la frammentazione. Fragment offset contiene quella che è la posizione del 1° byte contenuto nel frammento. Siccome un payload si può frammentare in gruppi di 8 byte (ottetti), si divide il fragment offset in byte per 8. Quindi:

1° FRAMMENTO → 0 = FRAGMENT OFFSET

2° FRAMMENTO → 600/8 ...

3° FRAMMENTO → 1200/8 ...

(24)

Si noti che se la divisione non trova un numero intero allora si tronca e l'ultimo byte cioè B si sposta verso il successivo frammento. Quindi i tre frammenti giungono alla macchina B. Vediamo adesso come B riassembla i frammenti. Supponiamo che a B gli giunga prima il frammento 3.

In caso spiacevole, Rare = 0, come fa B a capire che gli è giunto un frammento e non un pacchetto? Quando Rare e fragment offset. In particolare se:

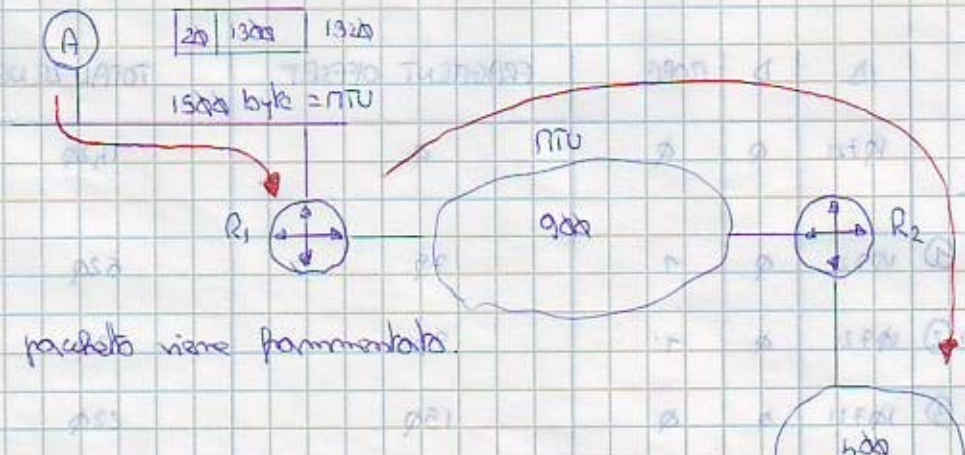
RARE = 0
FRAGMENT OFFSET ≠ 0

allora l'unità in questione è un pacchetto.

La macchina (B) quindi si posiziona tra 1200 e 1399, crea un buffer nel quale si attende l'arrivo del frammento e setta un timer. Poi arriva il frammento 1 e si ha:

se il frammento 2 arriva dopo lo scatto del timer viene perso tutto, altrimenti riempio il "buco".

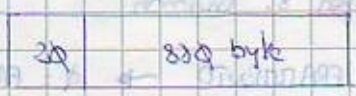
A questo punto la macchina B ricompatta il pacchetto IP e lo spedisce ai livelli superiori. Si noti che anche il router R_2 potrebbe riassemblare il pacchetto ottenendo anche un'efficienza maggiore. Questa cosa però non viene fatta perché non bisogna sovraccaricare i router. Si noti inoltre che la frammentazione è un'operazione ricorsiva. Vediamo un esempio:



Nel router R_1 il pacchetto viene frammentato. Quindi:

	ID	D	M	FO	TL
FRAMMENTO 1	19721	0	1	0	980
FRAMMENTO 2	19721	0	0	110	440

Guardiamo ora il frammento 1. Si ha:



ma il successivo MTU = 400 byte ⇒ nuova frammentazione.

Quindi:

20	376 byte

FRAMMENTO 1.1 — 2° LIVELLO DI FRAMMENTAZIONE

Si ricordi infatti che ci stanno 47 gruppi di 8 byte e quindi:

47·8 = 376 byte nel payload.

Quindi:

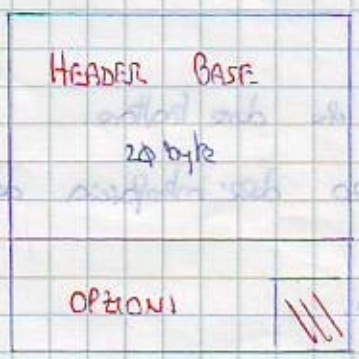
	ID	D	M	FO	TL
FRAMMENTO 1.1	10721	0	1	0	396
FRAMMENTO 1.2	10721	0	1	47	396
FRAMMENTO 1.3	10721	0	0	94	148 byte

Analogamente per il frammento 2:

	ID	FO	M	TL
FRAMMENTO 2.1	10721	110	1	396
FRAMMENTO 2.2	10721	157	0	64

2.1	20	376
2.2	20	44

La macchina destinataria a questo punto fa l'assemblamento. Vediamo ora le possibili opzioni dell'header IP:



Quindi se HEADER LENGTH = 5 ⇒ HEADER BASE.

se HEADER LENGTH > 5 ⇒ OPZIONI IN PIÙ.

NB: le opzioni seguono l'header base e sono multipli interi di 4 byte.

↳ bit non significativi da mettere in coda per fare in modo che le opzioni siano un multiplo di 4 byte.

Vediamo il 1° byte di opzioni:

OCO	C	C 1	C 2	N 1	N 2	N 3	N 4	N 5

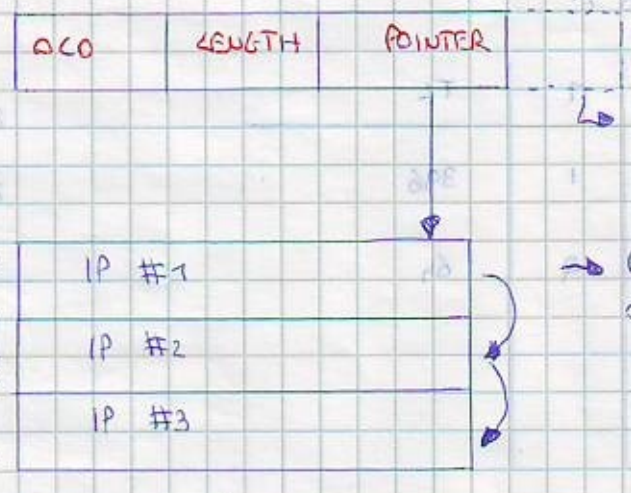
OCO = OPTION CODE OCTET

(26)

Analizziamone i campi:

- C = bit copy. Se C=1 ⇒ le opzioni vengono copiate in tutti gli header dei frammenti.
- (C 1) (C 2) sono le OPTION CLASS. Im particolare: 0 0 ⇒ Network control
  1 0 ⇒ debugging o measurement
- N N ... sono i numeri di opzione (32 numeri possibili).
  1 2

Vediamo ora un esempio di opzione. Guardiamo la RECORD ROUTE OPTION. Tale opzione serve per registrare nell'header il percorso che si sta seguendo (attraverso le interfacce dei routers).



Quindi il router fa le seguenti azioni:

1) Guarda se ci sono opzioni nell'header.
2) Guarda OCO per vedere il tipo di opzione che deve trattare.
3) Analizza il POINTER che punterà all'indirizzo dell'interfaccia del router successivo da attraversare.
4) Stampa nel suo indirizzo IP il next hop.
5) Incrementa il puntatore.

Un'altra opzione è STRICT SOURCE ROUTING che fa dell'override delle tabelle di routing. Si ha:

OCO	LENGTH	POINTER	—
IP #1			
IP #2			

} Parole di 32 bit.

⋮

Il router R_i "guarda" il pointer e lo incrementa di 1. Così facendo il pointer punta all'indirizzo IP del prossimo router (la sua interfaccia) che si dovrà attraversare. Ecco quindi che questa opzione la usiamo per fissare uno specifico percorso all'interno della rete. Si noti infine che se il router "vede" un next hop non valido, blocca il pacchetto, cioè lo cancella. Un'altra opzione piuttosto interessante è il time stamp.

OCO	LENGTH	POINTER	—
IP #1			
TIMESTAMP #1			
IP #2			
TIMESTAMP #2.			

⋮

Questa opzione non sempre si può usare. Serve principalmente per registrare un percorso e anche per fissare un router a mettere un'indicazione dell'istante di tempo in cui lo stesso ha ricevuto il pacchetto. Si fa ciò per vedere quanto un percorso è lento o veloce.

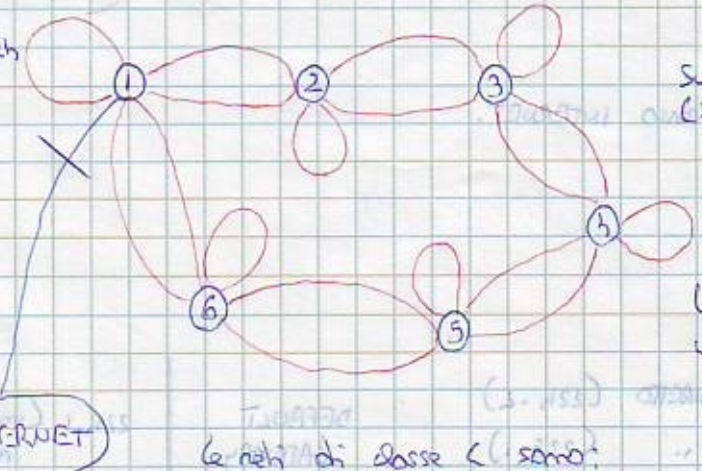
Si noti che se i router non sono sincronizzati i relativi timestamp risulteranno essere dei numeri casuali privi di senso. Ma come si sincronizzano i router? Ci sono sostanzialmente due modi:

1) GPS, cioè si installa un GPS su un router

2) NETWORK TIME CONTROL, che permette ai router di spargere delle informazioni di sincronizzazione all'interno della rete.

Vediamo ora un esempio conclusivo:

235/24



Supponiamo di aver ricevuto il seguente CIDR (indirizzo classless):

172.172.224.0 /19

Usiamo indirizzi ufficiali e quindi non usiamo NATBOX. In binario si ha:

10101100.10101100.111 00000.00000000 /19

Le reti di classe C sono:

$2^5 = 32$ reti.

OCO	LENGTH	POINTER	—
IP #1			
IP #2			

} Parole di 32 bit.

⋮

Il router R_i "guarda" il pointer e lo incrementa di 1. Così facendo il pointer punta all'indirizzo IP del prossimo router (la sua interfaccia) che si dovrà attraversare. Ecco quindi che questa opzione la usiamo per fissare uno specifico percorso all'interno della rete. Si noti infine che se il router "vede" un next hop non valido, blocca il pacchetto, cioè lo cancella. Un'altra opzione piuttosto interessante è il time stamp.

OCO	LENGTH	POINTER	—
IP #1			
TIMESTAMP #1			
IP #2			
TIMESTAMP #2.			

⋮

Questa opzione non sempre si può usare. Serve principalmente per registrare un percorso e anche per fissare un router a mettere un'indicazione dell'istante di tempo in cui lo stesso ha ricevuto il pacchetto. Si fa ciò per vedere quanto un percorso è lento o veloce.

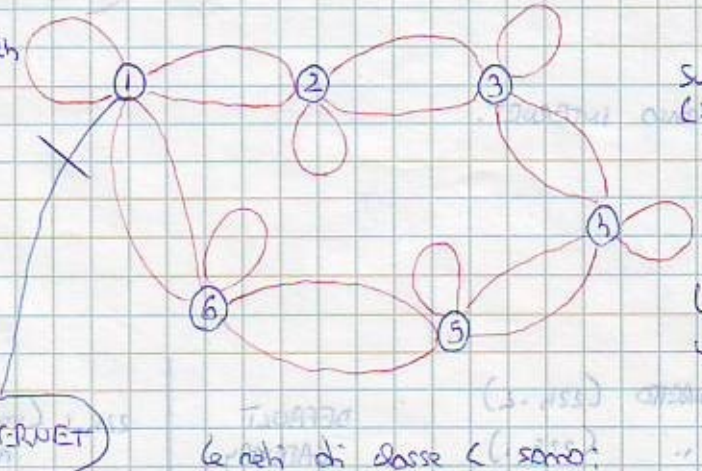
Si noti che se i router non sono sincronizzati i relativi timestamp risulteranno essere dei numeri casuali privi di senso. Ma come si sincronizzano i router? Ci sono sostanzialmente due modi:

1) GPS, cioè si installa un GPS su un router

2) NETWORK TIME CONTROL, che permette ai router di spargere delle informazioni di sincronizzazione all'interno della rete.

Vediamo ora un esempio conclusivo:

235/24



Supponiamo di aver ricevuto il seguente CIDR (indirizzo classless):

172.172.224.0 /19

Usiamo indirizzi ufficiali e quindi non usiamo NATBOX. In binario si ha:

10101100.10101100.111 00000.00000000 /19

Le reti di classe C sono:

$2^5 = 32$ reti.