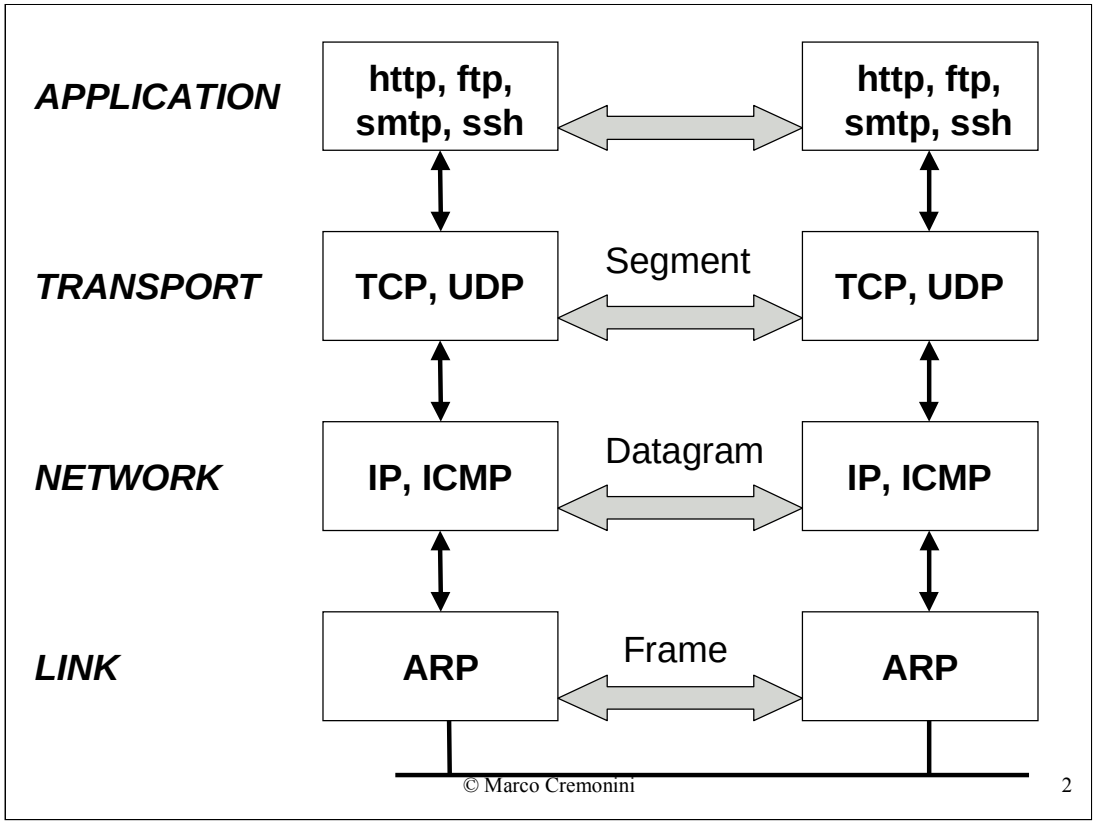
TCP/IP

Transmission Control Protocol/ Internet Protocol

© Marco Cremonini 1



Modello Internet dello STACK TCP.

- versione semplificata rispetto il modello OSI (7 livelli);
- modello a livelli (layers);
- ogni livello rappresenta un modello logico di comunicazione tra host;
- la comunicazione avviene attraverso PROTOCOLLI.

APPLICATION: a questo livello vengono definiti i protocolli applicativi (es. http, ftp, telnet, smtp, etc.), quindi sono i protocolli che ogni applicazione deve utilizzare nella comunicazione tra mittente e destinatario (es. Browser Web comunica via http, ad esempio, con un Server Web);

TRANSPORT: slide successiva

NETWORK: slide successiva

LINK: slide successiva

TRANSPORT: a questo livello i protocolli definiscono la modalita' secondo la quale mittente e destinatario comunicano.

TCP (Transaction Control Protocol) : Connection-oriented, si crea una sessione di comunicazione tra mittente e destinatario che viene instaurata con un accordo iniziale, vengono scambiati I dati e viene conclusa. E' affidabile (ritrasmissione dei pacchetti persi).

UDP (User Datagram Protocol) : Connection-less, non viene creata una sessione di comunicazione, non esiste un accordo preliminare tra mittente e destinatario , ovvero i pacchetti vengono inviati direttamente dal mittente senza garanzia di consegna.

© Marco Cremonini 3

NETWORK: a questo livello viene gestito lo spostamento dei pacchetti all'interno di una rete che normalmente vede la presenza di router tra mittente e destinatario.

IP (Internet Protocol) :in particolare permette l'instradamento corretto di ogni pacchetto inviato dal mittente fino al destinatario. A questo livello vengono gestiti gli INDIRIZZI IP, ed utilizzati da ogni router per determinare il corretto instradamento.

ICMP (Internet Control Message Protocol) : e' un protocollo utilizzato per segnalare condizioni di errore o testare la connettivita'.

© Marco Cremonini 4

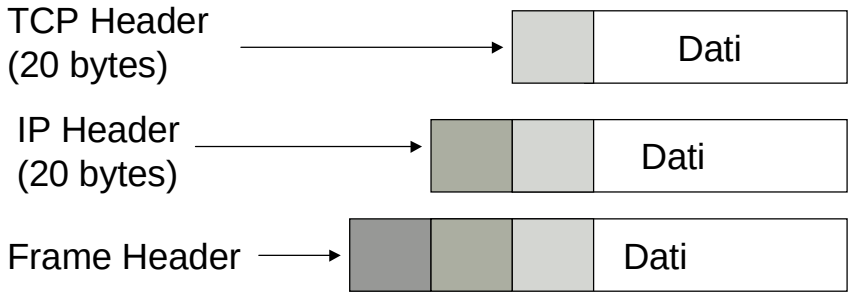
LINK: questo livello ha il compito di interfacciarsi con lo strato di rete fisico (es. Ethernet) e gestire la comunicazione tra schede di rete.

ARP (Address Resolution Protocol) : e' il protocollo usato per mappare gli indirizzi IP (livello Network) su identificativi fisici delle schede di rete (MAC). Questo protocollo viene utilizzato per l'indirizzamento dei pacchetti tra host (mittente-destinatario, mittente-router, router-destinatario) appartenenti ad uno stesso ramo di rete.

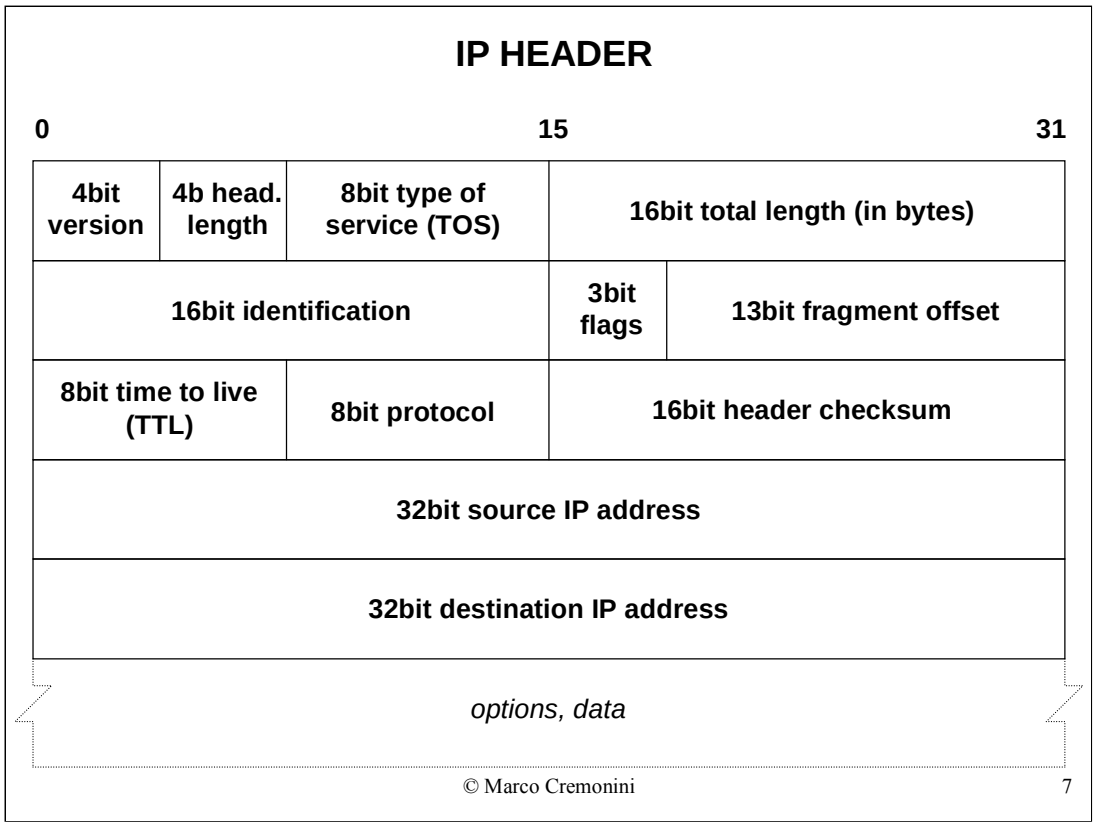
© Marco Cremonini

INCAPSULAMENTO

Implementazione del modello logico a livelli dello stack TCP.



© Marco Cremonini 6



Version : attualmente IP V. 4. Esiste anche IPv6 (versione 6), poco diffuso ancora;

Header Length : definisce la lunghezza IN 32-BIT (4 BYTES). Valore Normale = 5 => header IP 20 bytes;

TOS : (non ci interessa, minimizza il ritardo, massimizza il numero di pacchetti inviati, etc.,) molte delle attuali implementazioni del TCP non supportano queste opzioni;

Total Length : lunghezza in bytes dell'intero datagram IP trasmesso (header + opzioni + dati). Valore massimo = 655535 bytes (2**16). Questo valore e' logico non necessariamente fisico, perche' al livello di link esistono limiti piu' stretti derivanti dalla tecnologia di rete (es. Ethernet max 1500 bytes a pacchetto);

Identification : numero identificativo dei datagram IP, ad ogni trasmissione successiva incrementa di 1;

Flags e **Fragment Offset** : rimandiamo la discussione alla successiva analisi della frammentazione;

TTL : definisce il numero massimo di router attraverso il quale il datagram IP puo' passare. Ogni router che riceve il pacchetto, decrementa di 1 il TTL. Quando questo e' 0, il router lo scarta e un messaggio di errore ICMP viene mandato al mittente;

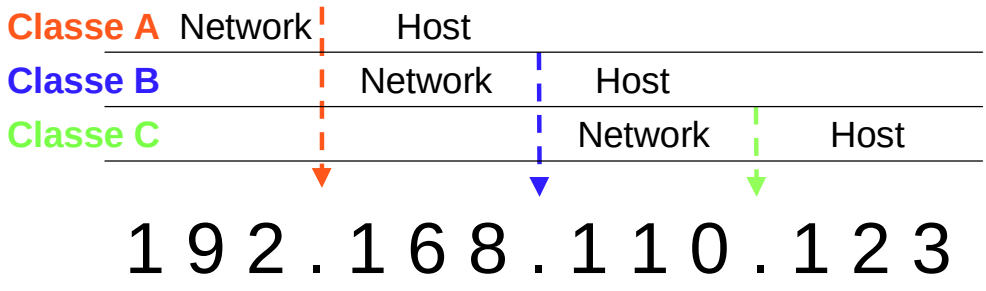
Protocol : identifica quale protocollo a livello di TRANSPORT deve gestire il pacchetto da trasmettere. ICMP: 1; TCP: 6;

Header Checksum : calcolato solo sull'header, non su opzioni e dati. Verifica che i bit non si siano corrotti nella trasmissione;

Source IP Address e **Destination IP Address** : indirizzi IP del mittente e del destinatario del pacchetto.

INDIRIZZI IP

Classe	Network Bits	Host Bits	No. di Host
A	8	24	16M+
B	16	16	65000+
C	24	8	255



© Marco Cremonini 8

Classe	IP Iniziale	IP Finale
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255

INDIRIZZI NON INSTRADABILI:

Le sottoreti 192.168 e 172.16 sono RISERVATE all'uso all'interno di RETI PRIVATE, non sono indirizzi utilizzabili su Internet.

La sottorete 127 e' riservata al LOOPBACK: client e server su stessa macchina (normalmente si usa 127.0.0.1)

© Marco Cremonini 9

SOTTORETE (RFC 950, 1985)

Classe B

16 bit *x* bit *16-x* bit

SUBNET MASK

Es. **140.252.13.32** , classe B, 16 bit per Network. Dei restanti? Come faccio a sapere quali bit indicano la Subnet e quali indicano l'Host?

© Marco Cremonini 10

Problema: la partizione in Network e Host, per le Classi A e B e' poco gestibile (18M+ di host e 65000+ di host rispettivamente). E' molto raro che ci possano essere tanti host connessi ad una stessa rete fisica (senza routing).

Occorre una partizione ulteriore => SUBNET

E' a discrezione dell'amministratore della rete che ha ricevuto la Classe di indirizzi (es. Classe B) decidere la propria politica di divisione in sottoreti settando la SUBNET MASK.

SUBNET MASK : 1 per Network e Subnet; 0 per Host

Classe B

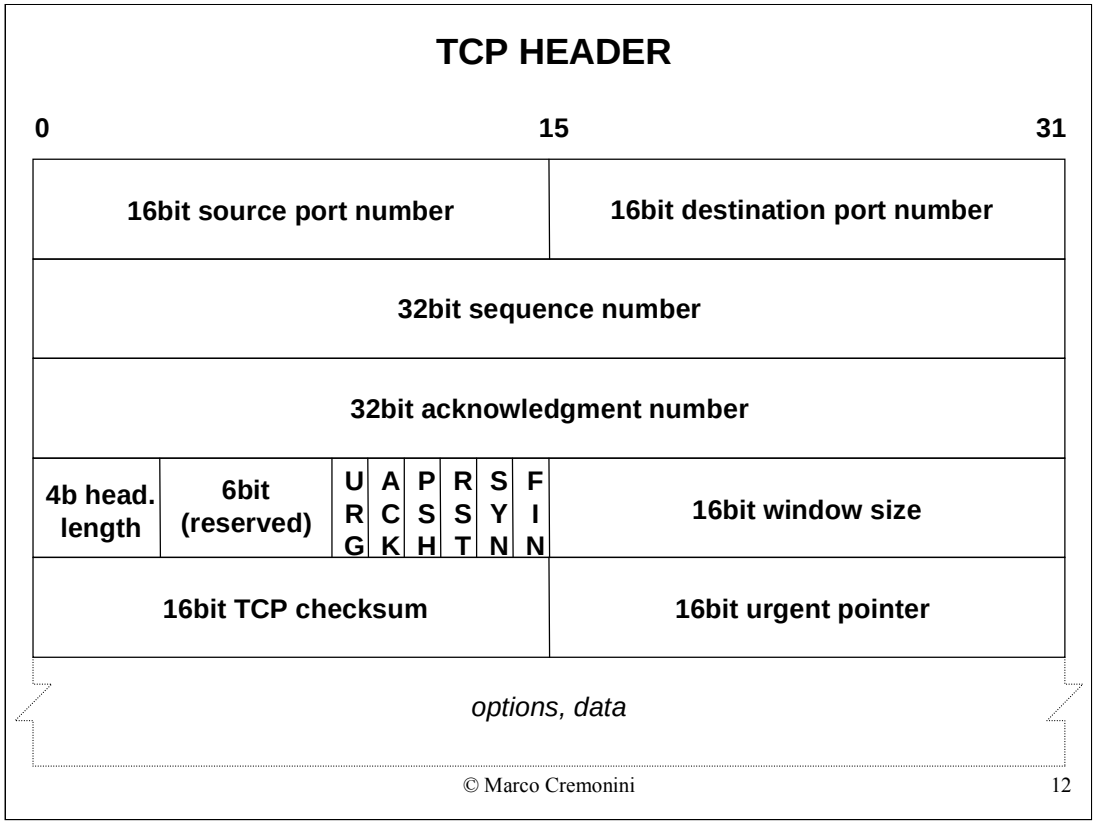
NETWORK		SUBNET	HOST
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0
255	**255**	**255**	**0**
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 0 0 0 0
255	**255**	**255**	**240**
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 0 0 0 0 0 0
255	**255**	**255**	**192**

255.255.255.0 : 255 subnet e 255 host per subnet;

255.255.255.240 : 4095 subnet e 15 host per subnet

255.255.255.192 : 1023 subnet e 63 host per subnet

© Marco Cremonini 11



Source e **Destination Port Number** : Numero della Porta del mittente e del destinatario. Identificano l'applicazione (livello APPLICATION) che sta comunicando attraverso la sessione TCP; i numeri delle due porte insieme agli indirizzi IP identificano in maniera univoca la *connessione*;

Sequence Number : Numero di Sequenza, identifica il primo byte di dati trasmesso con il pacchetto. Quando una connessione viene iniziata, il sequence number viene settato ad un valore RANDOM;

Acknowledgement Number : Numero di Riconoscimento, identifica il SUCCESSIVO sequence number che il ricevente si attende con il successivo pacchetto;

Header Length : come per l'header TCP, il valore normale e' 5;

Flags (URG, ACK, PSH, RST, SYN, FIN) : slide successiva;

Window Size : Dimensione della Finestra, indica il numero di bytes che il ricevente puo' accettare;

Checksum : come per l'header IP;

Urgent Pointer : (non lo trattiamo).

Esempi di Porte

```
ftp              21/tcp File Transfer
ssh              22/tcp Secure SHell
telnet           23/tcp Telnet [112,JBP]
smtp             25/tcp Simple Mail Transfer [102,JBP]
http             80/tcp www www-http World Wide Web HTTP
pop3            110/tcp Post Office Protocol - Version 3
auth            113/tcp Authentication Service
netbios-ssn     137-139/tcp NETBIOS Session Service
imap2           143/tcp Interim Mail Access Protocol v2
login           513/tcp remote login;
unknown         635/tcp unassigned
domain           53/udp  Domain Name Service
domain           53/tcp  Domain Name Service
```

© Marco Cremonini 13

Porte : 0 - 65535, distinte tra TCP e UDP (quindi in realta' le possibili porte sono 65536x2). Identificano il protocollo applicativo (livello APPLICATION) usato tra client e server.

L'associazione NUMERO - APPLICAZIONE (es. 22/tcp - ssh) e' PER CONVENZIONE fissata, nulla vieta di usare una qualsiasi porta per un qualsiasi servizio.

Le porte da 0 a 1024, vengono dette "fidate" (trusted ports) perche' solo l'utente **root** puo' abilitarle (vero prima della nascita dei desktop).

Oggi si puo' dire che nel range 0-1024 sono definite le porte di tutti i servizi tradizionali del mondo UNIX (telnet, ftp, ssh, smtp, pop3, login, etc.). Molti delle applicazioni piu' recenti hanno porte maggiori di 1024.

Lista completa: http://www.isi.edu/in-notes/iana/assignements/port-numbers

TCP FLAGS

SYN : richiesta di stabilire una sessione, sempre il primo pacchetto di una comunicazione TCP;

ACK : conferma del pacchetto precedente, sia esso dati, SYN o FIN;

FIN : indica l'intenzione del mittente di terminare la sessione in maniera concordata;

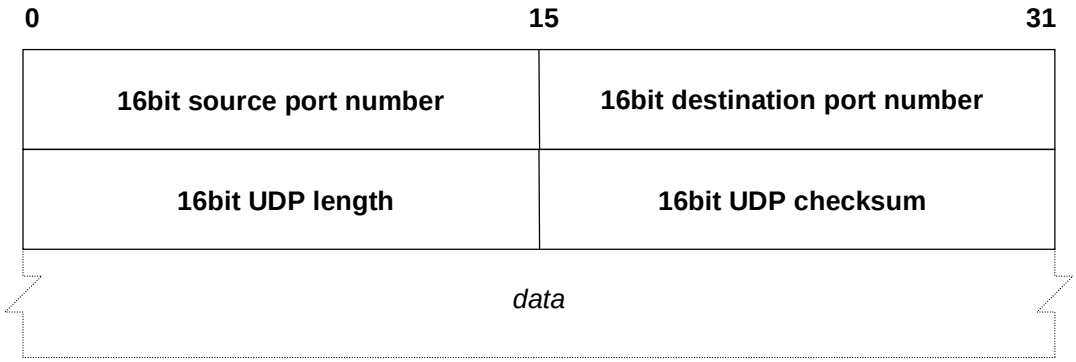
RST : reset della sessione;

PSH : operazione di push, i dati vengono subito inviati al destinatario senza bufferizzarli;

URG : dati urgenti (es. CTRL+C) vengono inviati con precedenza sugli altri;

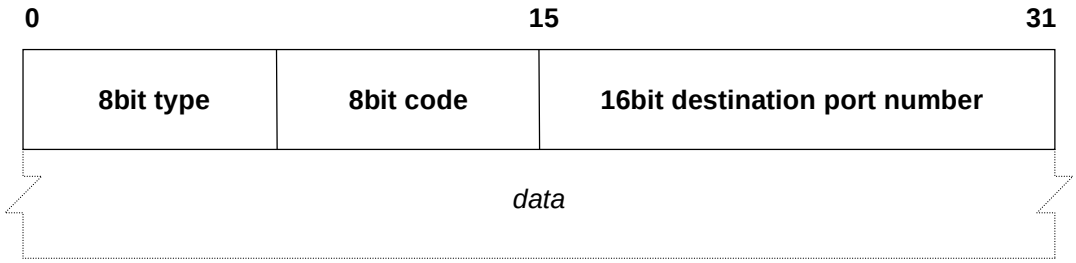
© Marco Cremonini 14

UDP HEADER



© Marco Cremonini 15

ICMP HEADER



© Marco Cremonini 16

Esempi di messaggi ICMP

TIPO	CODICE	
0	0	echo reply (risposta al ping)
3	1	host unreachable
3	13	unreachable - admin prohibited
4	0	source quench
8	0	echo request (ping)
11	0	time exceeded in transit

© Marco Cremonini 17

0 - 0 : risposta al ping;

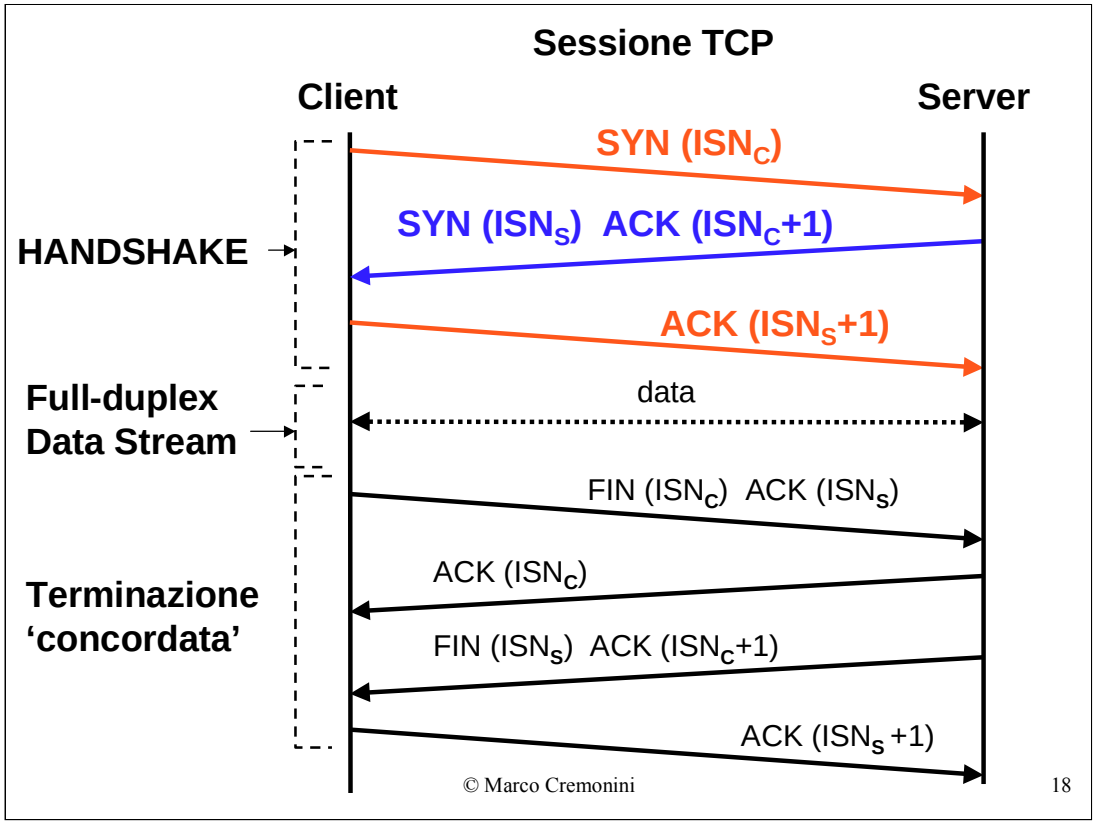
3 - 1 : messaggio di errore quando l'indirizzo IP del destinatario non e' raggiungibile (l'IP non esiste, l'host non e' connesso oppure non riesce a rispondere);

3 - 13 : un router ha un'ACCESS CONTROL LIST configurata per non permettere di instradare quel pacchetto (indirizzo IP vietato, porta non ammessa, etc.)

4 - 0 : messaggio per il controllo del flusso di dati, il destinatario chiede al mittente di ridurre la frequenza di trasmissione;

8 - 0 : ping

11 - 0 : un router verifica che il TTL si e' azzerato e quindi scarta il pacchetto, notificando il mittente con questo messaggio di errore.



HANDSHAKE : Con questo protocollo detto "handshake" si stabilisce la sessione TCP, gli ISN (Initial Sequence Number) vengono settati da client e server;

Full-duplex Data Stream : il flusso di dati (stream, livello Application) si dice full-duplex perche' in una sessione TCP I dati fluiscono da client a server e viceversa all'interno della stessa sessione;

Terminazione : conclusione della connessione. Quella mostrata e' detta "concordata" ("graceful") perche' a seguito di un protocollo con scambio di FIN-ACK che permette alle due parti di prepararsi alla terminazione. Esiste anche il caso di terminazione "non concordata" nella quale una della due parti invia un pacchetto con il flag RST (RESET) attivo che provoca l'abort immediato della sessione.

Analisi del Traffico di Rete

HANDSHAKE - STREAM - TERMINATION *(in questo caso RESET anziche' FIN)*

```
tclient.net.52894 > tserver.com.23: S
                    3900690:3900690(0)
tserver.com.23 > tclient.net.52894: S
1379776:1379776(0) ack 3900691
tclient.net.52894 > tserver.com.23: . ack 1379777
tclient.net.52894 > tserver.com.23: P 1:28(27) ack 1
tserver.com.23 > tclient.net.52894: P 1:14(13) ack 1
tserver.com.23 > tclient.net.52894: P 14:23(9) ack 28
tclient.net.52894 > tserver.com.23: R 28:28(0) ack 1
```

© Marco Cremonini 19

TCPdump : uno dei piu' diffusi analizzatori di traffico di rete

Pacchetti 1-3 : Handshake (notare il Sequence Number degli ACK incrementati di 1);

Pacchetti 4 - 6 : Full-duplex data stream. PER CONVENZIONE di molti analizzatori del traffico di rete (tra cui TCPdump), gli ACK successivi all'handshake prendono una numerazione dei sequence number relativa anziche' assoluta partendo da 1 (i veri sequence number dei pacchetti fisici hanno numerazione assoluta);

Pacchetto 7 : reset della sessione.

Elementi minimi da valutare nell'analisi:

- **L'handshake e' stato completato?**
- **Sono stati trasmessi dati?**
- **Chi ha iniziato e/o terminato la connessione?**

© Marco Cremonini 20

Handshake Completato : se c'e' un handshake completo, significa che il server ha la porta di destinazione APERTA ed accetta quindi connessioni. Questo e' il caso normale per tutti i servizi abilitati ed utilizzati correntemente. Cosa significa pero' se un handshake viene completato verso una porta che non dovrebbe essere attiva o che ritenevamo non esserlo? E' segno di un'intrusione? Un uso illecito delle nostre risorse?

Trasmissione di dati: se c'e' trasmissione di dati significa che client e server stanno comunicando secondo le modalita' di una certa applicazione.

Chi ha iniziato/concluso la sessione? : Riconoscendo quale host ha iniziato la sessione si deduce quale host abbia il controllo della comunicazione.

Specifiche TCP: RFC 793 (Postel, 1981)

Un segmento in arrivo che contenga un RESET viene sempre scartato senza alcuna risposta.

Se una porta e' chiusa

Se il segmento in arrivo NON contiene un RESET allora viene inviato come risposta al mittente un pacchetto con il flag RESET attivo.

Se una porta e' aperta ed e' nello stato di *listen*

Se il segmento in arrivo contiene un ACK allora viene risposto un RESET;

Se il segmento in arrivo contiene un SYN allora viene inviato come risposta al mittente un pacchetto con i flag SYN ACK attivo;

Se nessuno dei precedenti e' vero allora il segmento viene scartato senza risposta.

© Marco Cremonini 21

Stati delle Porte:

CLOSE : il servizio non e' abilitato, nessuna connessione e' possibile;

LISTEN : il servizio e' abilitato ed e' IN ATTESA di connessione;

ESTABLISHED : la connessione e' attiva.

Cosa succede in questa traccia?

```
abc.com.telnet > efg.net.telnet: ack 1379777
abc.com.telnet > hil.org.telnet: ack 1379777
abc.com.imap > efg.net.imap: ack 1379777
abc.com.imap > hil.org.imap: ack 1379777
abc.com.ssh > efg.net.ssh: ack 1379777
abc.com.ssh > hil.org.ssh: ack 1379777
abc.com.telnet > mno.it.telnet: ack 1379777
mno.it.telnet > abc.com.telnet : R
abc.com.telnet > pqr.it.telnet: ack 1379777
pqr.it.telnet > abc.com.telnet : R
```

© Marco Cremonini 22

Elementi da notare:

- Non rispetta l'handshake del protocollo TCP;
- Porta del client < 1024 ed uguale a quella di destinazione;
- Sequence Number sempre uguale;

Come e' possibile? I pacchetti sono stati manipolati ('crafted') volutamente, non sono frutto di una implementazione TCP.

Per quale motivo? Che cosa ricavo?

Se non ottengo risposta significa che il destinatario e' irraggiungibile (inesistente, sconnesso, etc.).

Se ricevo un RESET significa che il destinatario esiste (non so pero' se la porta sia CLOSE o LISTEN).

Ricorda qualcosa?

Spiegazione

Il PING fornisce la stessa informazione.

Perche' usare questo e non il ping? Spesso il ping viene impedito dai firewall (si evita che chiunque da Internet possa scoprire gli host attivi nella rete interna).

Questi pacchetti invece potrebbero essere fatti passare dal firewall, ad esempio perche' scambiati per pacchetti legittimi di un handshake (i terzi dell'handshake). Siamo in presenza di un probabile

ACK SCAN

Scanning: tecniche per acquisire informazioni utili ad un attacco.

© Marco Cremonini

Come viene raccolto questo traffico di rete?

ANALIZZATORE DI TRAFFICO (SNIFFER, SENSORE, PROBE).

Esempio : **TCPdump**

Comunicazione su di una Ethernet

- ogni computer analizza tutti i pacchetti della sottorete;
- se un pacchetto non e' indirizzato a se' stesso, la scheda di rete lo scarta;
- se il pacchetto e' indirizzato a se' stesso viene processato dai livelli dello stack TCP.

© Marco Cremonini 24

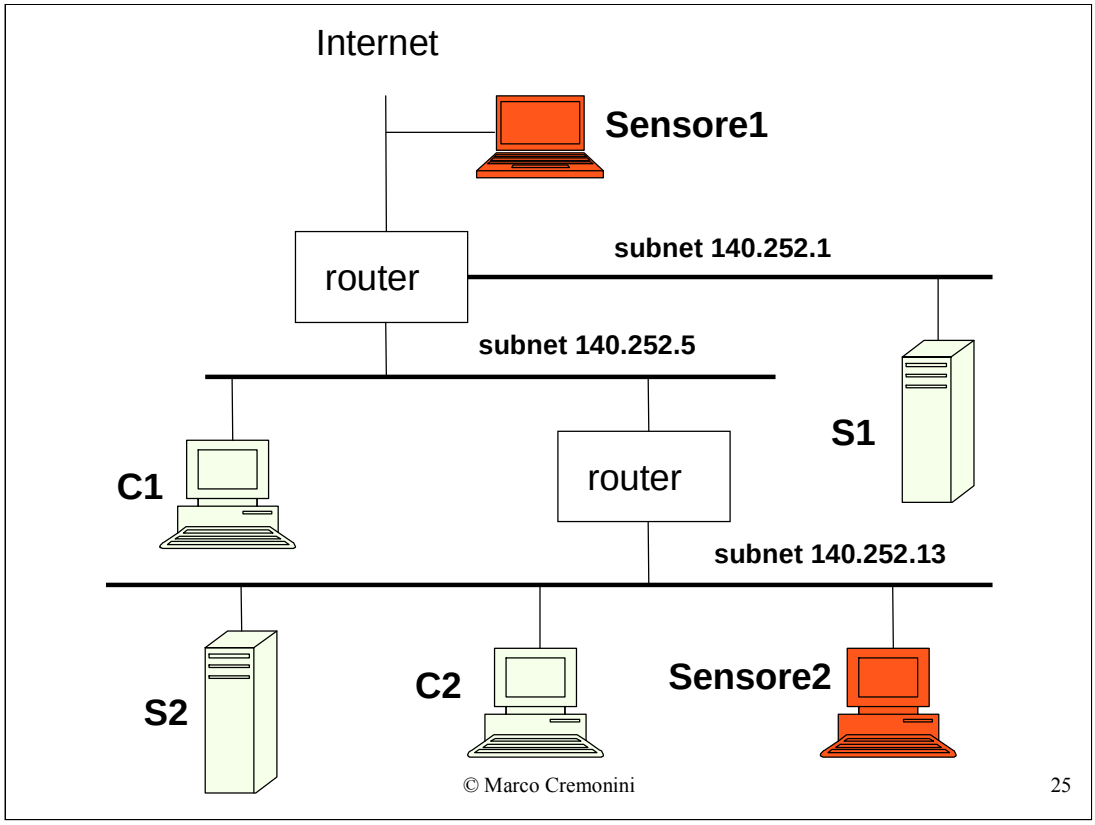
La comunicazione su di un ramo di rete (es. Ethernet) prevede che tutti i pacchetti vengano analizzati da tutti i computer (attraverso le schede di rete) connessi sullo specifico ramo di rete.

Quindi, dal punto di vista fisico, ogni computer in realta' osserva il contenuto dei pacchetti (a livello di LINK) e solo successivamente, se non e' il destinatario, lo scarta.

Per analizzare il traffico di rete, quindi, non occorre nessun hardware specifico ma :

•un funzionamento differente del driver della scheda di rete (si dice che la scheda di rete funziona in **MODALITA' PROMISCUA**);

•un software che gestisca la visualizzazione o il salvataggio dei pacchetti osservati (esempio **TCPdump**).



Quale traffico osservano i due sensori?

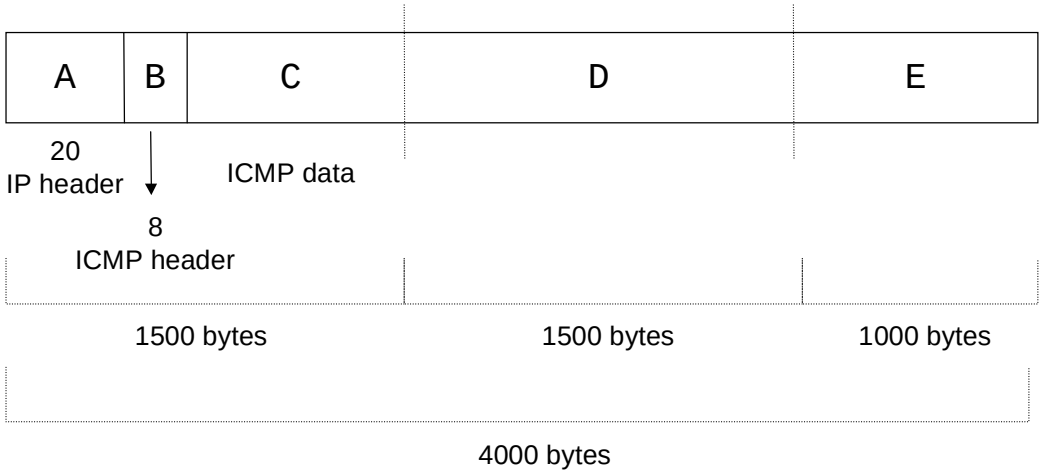
Internet -> Rete Locale
Sensore 1 : tutto
Sensore 2 : solo quello diretto alla subnet 140.252.13;

Locale alla Subnet 140.252.13
Sensore 1 : no
Sensore 2 : si

Tra la subnet 140.252.5 e la subnet 140.252.1
Sensore 1 : no
Sensore 2 : no

FRAMMENTAZIONE

MTU (Maximum Transmission Unit): dimensione massima dei pacchetti ammessi da una rete.
Es. Ethernet MTU = 1500 bytes.



© Marco Cremonini 26

La frammentazione avviene quando un datagram IP deve attraversare una rete che permette solo pachetti di dimensione inferiore al datagram stesso.

In tal caso, il router responsabile dell'instradamento deve FRAMMENTARE il pacchetto in pacchetti piu' ridotti.

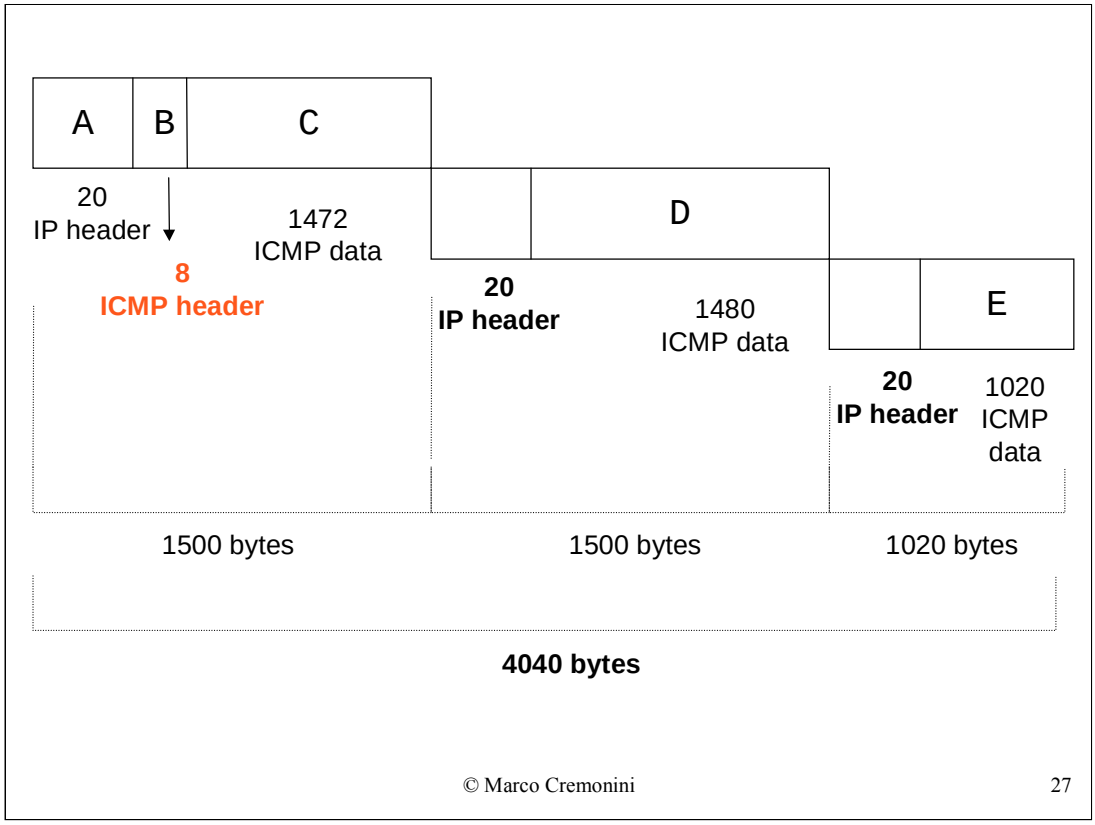
Consideriamo il seguente esempio: un echo request (ping) di 4000 bytes.

20 bytes : IP header;

8 bytes : ICMP header;

I restanti (4000 - 28) = 3972 : ICMP dati

Attraversando una Ethernet (MTU = 1500) deve essere frammentato. Come?

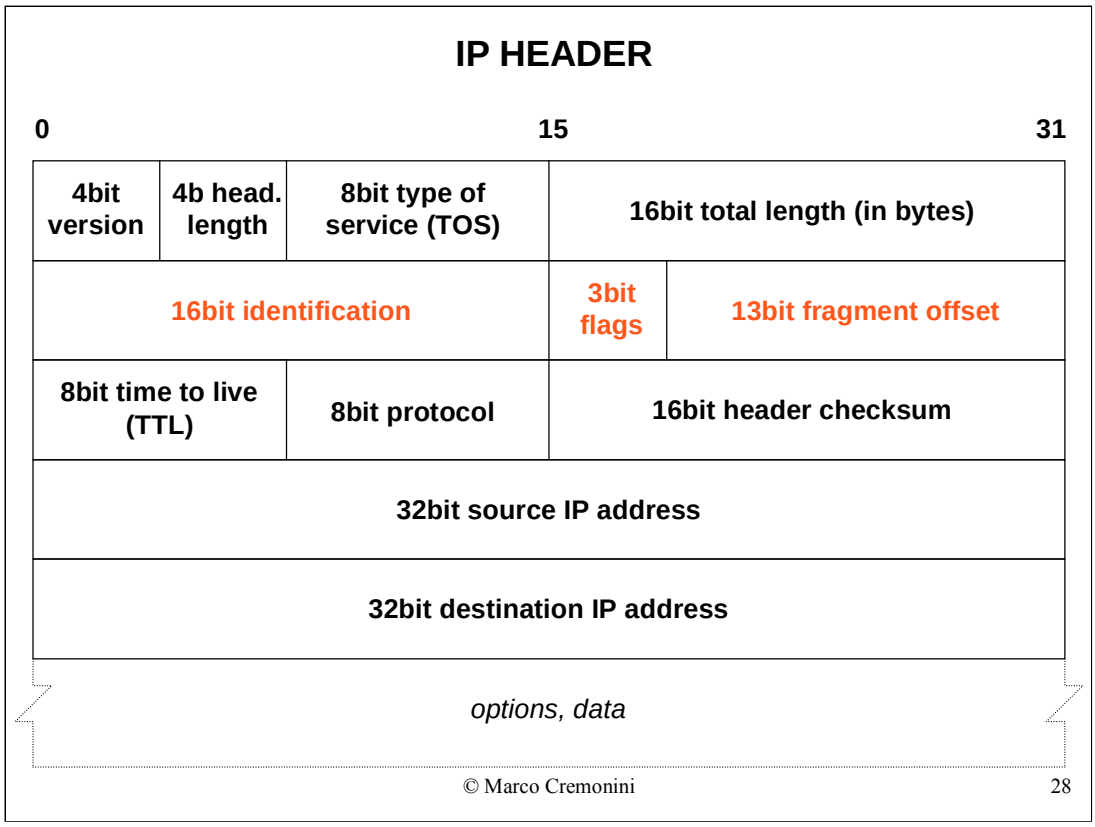


I dati del pacchetto complessivo sono 3972 bytes.

La frammentazione non puo' superare il MTU (1500 bytes) e ogni frammento deve essere un pacchetto INSTRADABILE dai router, quindi deve avere l'IP HEADER (20 bytes).

L'ICMP HEADER serve? No, per il routing non serve.

SOLO IL PRIMO FRAMMENTO MANTIENE L'HEADER DEL PROTOCOLLO DI TRANSPORT (TCP, ICMP), GLI ALTRI HANNO SOLO L'HEADER IP.



Version : attualmente IP V. 4. Esiste anche IPv6 (versione 6), poco diffuso ancora;

Header Length : definisce la lunghezza IN 32-BIT (4 BYTES). Valore Normale = 5 => header IP 20 bytes;

TOS : (non ci interessa, minimizza il ritardo, massimizza il numero di pacchetti inviati, etc.,) molte delle attuali implementazioni del TCP non supportano queste opzioni;

Total Length : lunghezza in bytes dell'intero datagram IP trasmesso (header + opzioni + dati). Valore massimo = 655535 bytes (2**16). Questo valore e' logico non necessariamente fisico, perche' al livello di link esistono limiti piu' stretti derivanti dalla tecnologia di rete (es. Ethernet max 1500 bytes a pacchetto);

Identification : numero identificativo dei datagram IP, ad ogni trasmissione successiva incrementa di 1;

Flags e **Fragment Offset** : slide successiva;

TTL : definisce il numero massimo di router attraverso il quale il datagram IP puo' passare. Ogni router che riceve il pacchetto, decrementa di 1 il TTL. Quando questo e' 0, il router lo scarta e un messaggio di errore ICMP viene mandato al mittente;

Protocol : identifica quale protocollo a livello di TRANSPORT deve gestire il pacchetto da trasmettere. ICMP: 1; TCP: 6;

Header Checksum : calcolato solo sull'header, non su opzioni e dati. Verifica che i bit non si siano corrotti nella trasmissione;

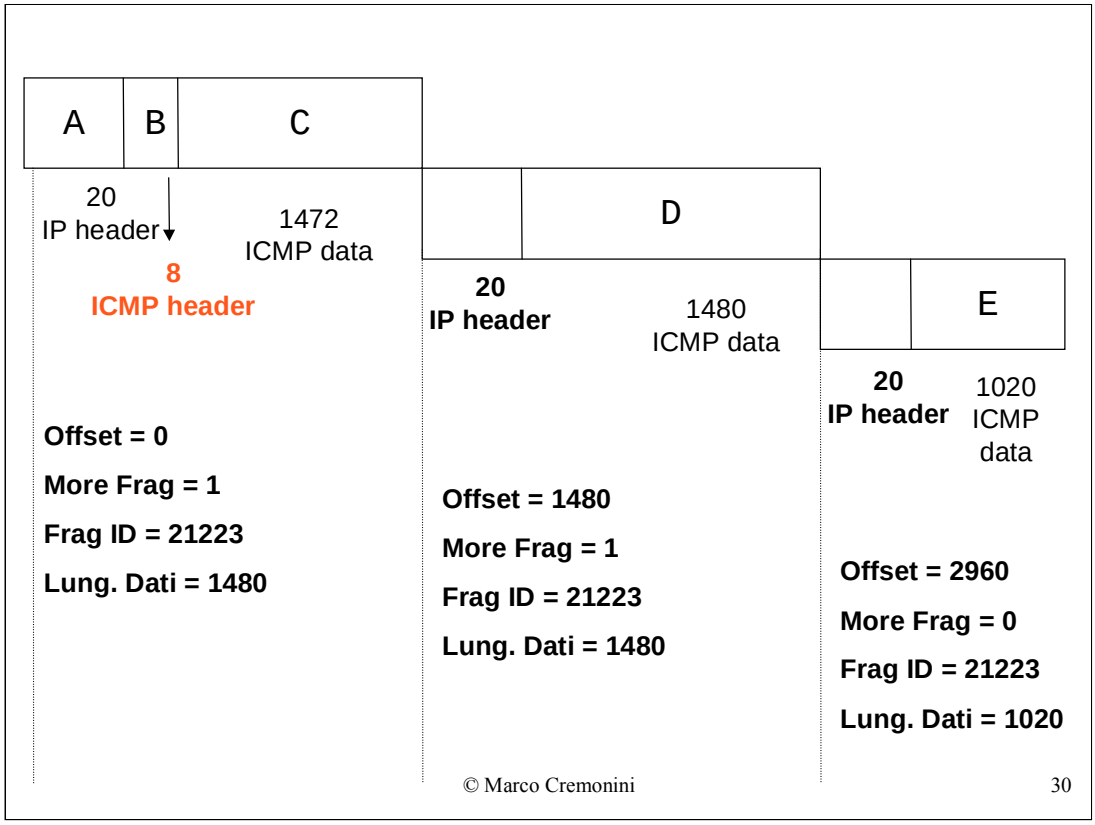
Source IP Address e **Destination IP Address** : indirizzi IP del mittente e del destinatario del pacchetto.

IDENTIFICATION : valore unico per tutti I frammenti di un pacchetto originario che dovra' essere riassemblato a destinazione.

FLAGS : un bit per indicare ***more fragments***, settato a **1** per tutti I frammenti tranne l'ultimo che ha il flag a **0**.
Un secondo bit per indicare ***don't fragment***, cioe' per istruire i router che, se fosse necessaria la frammentazione, il datagram va scartato e non frammentato

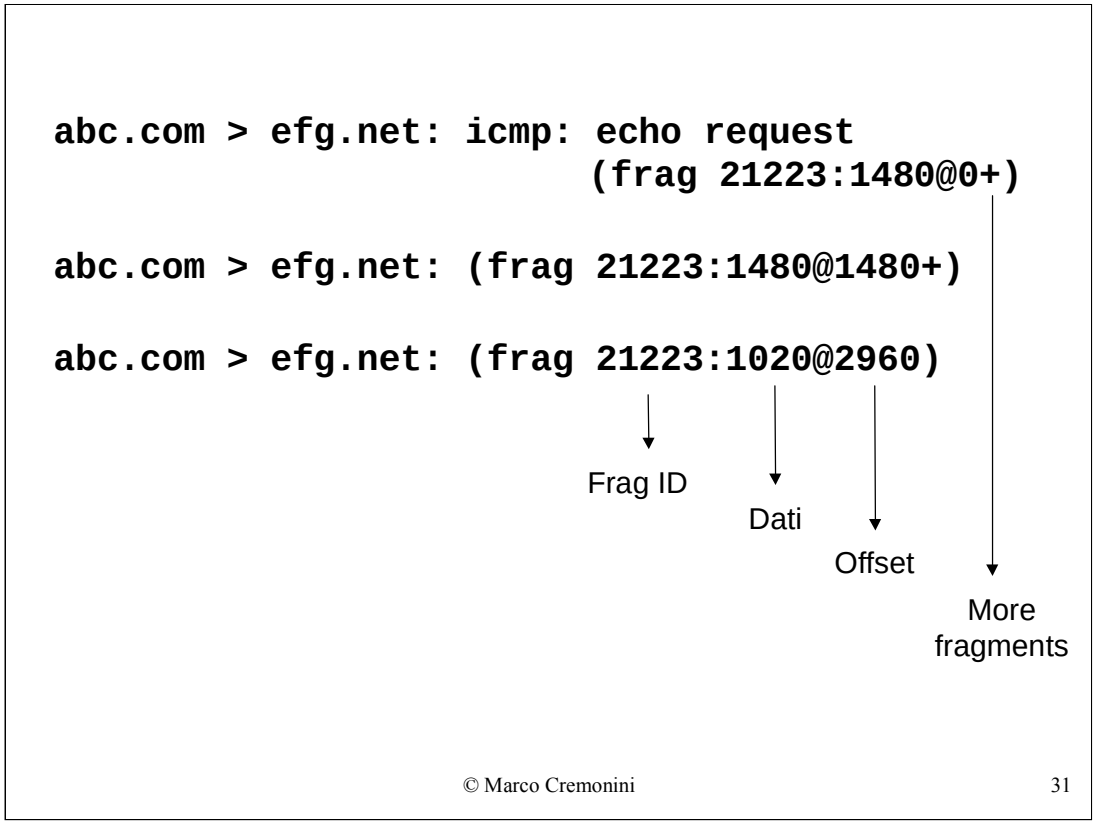
OFFSET : e' l'offset dei dati nei diversi frammenti rispetto I dati del datagram originario.

© Marco Cremonini



Gli offset si calcolano sui dati del datagram originale -> gli IP header aggiunti successivamente ai frammenti non si considerano.

Nei dati del datagram originario (livello NETWORK) e' compreso anche l'Header del livello TRANSPORT (TCP, ICMP).



NOTARE : solo nel primo frammento compare l'indicazione del protocollo (ICMP) e del tipo di messaggio (echo request).

Questo perche' SOLO IL PRIMO FRAMMENTO HA L'HEADER DEL LIVELLO DI TRANSPORT (TCP, ICMP).

Uso del flag **don't fragment** (**DF**)

```
abc.com > efg.net: icmp: echo request (DF)

router.net > abc.com : icmp: efg.net
unreachable - need to frag (mtu 1500) (DF)
```

© Marco Cremonini 32

Se il flag DF fosse stato settato, nell'esempio appena visto (datagram di 4000 bytes), il router responsabile dell'instaradamento sulla Ethernet con MTU = 1500 avrebbe scartato il datagram e risposto al mittente con un messaggio di errore ICMP di tipo host unreachable - need to frag.

Cosa succede in questa traccia?

```
abc.com. > 192.168.133.0: (frag 54050:1480@4440+)
abc.com. > 192.168.133.0: (frag 54050:1480@2960+)
abc.com. > 192.168.133.0: (frag 54050:1480@4440+)
abc.com. > 192.168.133.0: (frag 54050:1480@1480+)
abc.com. > 192.168.133.0: (frag 54050:1480@2960+)
abc.com. > 192.168.133.0: (frag 54050:1480@5920+)
abc.com. > 192.168.133.0: (frag 54050:1480@4440+)
abc.com. > 192.168.133.0: (frag 54050:1480@1480+)
```

© Marco Cremonini 33

Elementi ambigui:

- manca un frammento iniziale (Offset = 0);
- manca un frammento finale (flag More Fragments = 0);
- offset ripetuti;

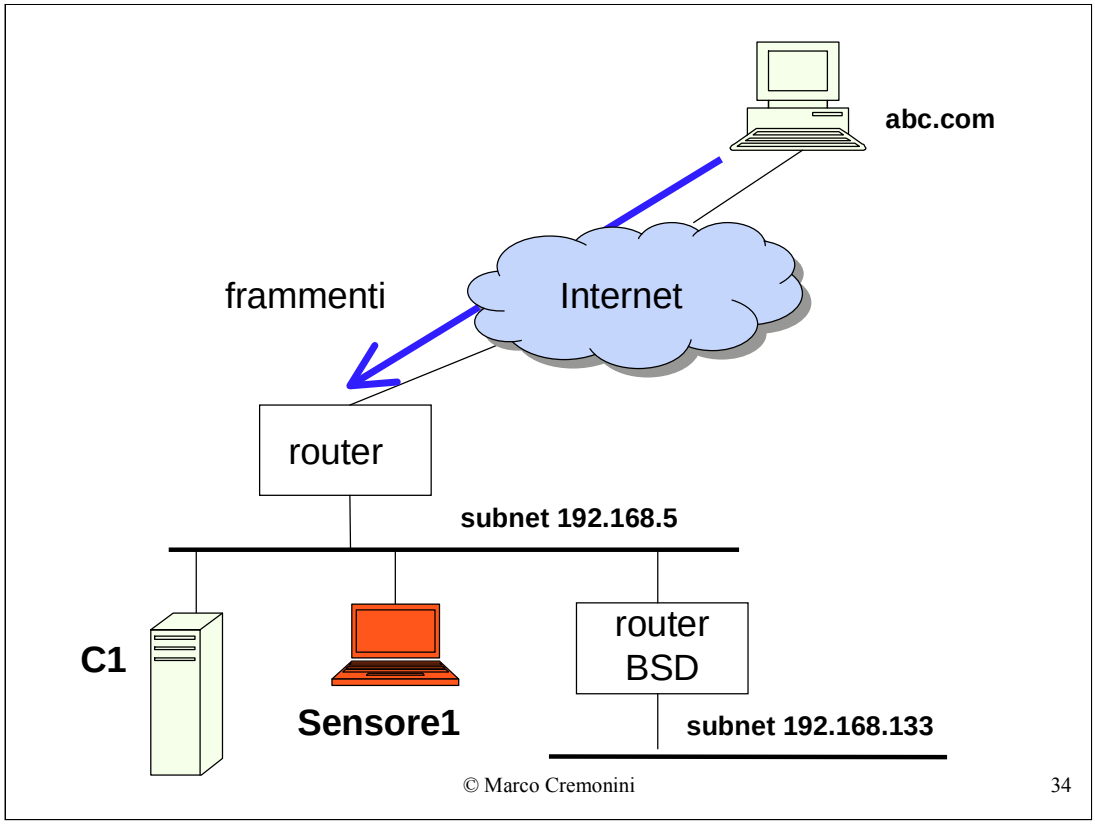
??????? - 1480 ---------- 2960 ----------4440 ---------- 5920 -?????????

1480 bytes 1480 bytes 1480 bytes

- pacchetti non ordinati cronologicamente : non significativo, i pacchetti possono venire instradati in modi differenti (non comune in una stessa sessione).

Come e' possibile?

NOTARE IP ADDRESS DI DESTINAZIONE : 192.168.133.0 -> riservato a reti interne -> il sensore e' posto nella rete interna, cioe' dietro un router o firewall.



Schema grafico dell'esempio che stiamo considerando.

Il sensore che analizza il traffico e che quindi ha riportato la traccia precedente e' collocato dietro il router di frontiera (router connesso ad Internet).

Cosa implica? Implica che eventuali pacchetti rigettati dal router non raggiungono la sottorete interna e quindi non sono intercettati dal sensore.

Quindi? Che ipotesi possiamo fare rispetto la traccia considerata?

Perche' manca il frammento iniziale e quello finale? Perche' ho offset ripetuti?

Spiegazione (prima parte)

Frammento iniziale mancante : il frammento iniziale e' l'unico ad avere l'header del livello di TRANSPORT (TCP, UDP, ICMP). Potrebbe essere stato bloccato dalla politica di sicurezza (tipo di servizio vietato).

Gli altri frammenti, non avendo header TCP/UDP/ICMP superano la verifica.

Frammento finale mancante : nessuna spiegazione, possibile datagram IP manipolato volutamente.

© Marco Cremonini

Spiegazione (seconda parte)

Offset ripetuti : molti frammenti passano attraverso il firewall e raggiungono l'host di destinazione.

Consideriamo meglio l'IP address di destinazione: **192.168.133.0 -> broadcast** per sistemi operativi BSD.

Nel nostro esempio abbiamo un router BSD che:

- riceve i frammenti e li mantiene in cache in attesa di riassemblare il datagram originale;

- frammento iniziale e finale non arrivano impedendo il riassemblaggio;

- molti frammenti con stesso FRAG ID continuano ad arrivare, impedendo il time-out del router;

© Marco Cremonini 36

L'indirizzo X.Y.Z.0 e' l'indirizzo di broadcast per sistemi operativi della famiglia BSD

Analogamente l'indirizzo X.Y.Z.255 e' l'indirizzo di broadcast generico per sistemi operativi Unix.

Un pacchetto inviato all'indirizzo di broadcast X.Y.Z.0 (o X.Y.Z.255 in generale) di un router, se questo ha abilitata la funzione di broadcasting, viene poi reindirizzato a tutti gli host della rete di Classe C X.Y.Z (192.168.133 nel nostro esempio).

Spiegazione (conclusione)

Il router BSD non va in time-out -> overload

- la capacita' di routing del traffico regolare si degrada fino ad annullarsi.

DENIAL OF SERVICE

Quindi:

- attacco portato con successo;
- politica di sicurezza perimetrale non efficace;
- complessita' nel gestire la frammentazione.

`www.cisco.com/warp/public/770/nifrag.shtml`

© Marco Cremonini 37

L'indirizzo X.Y.Z.0 e' l'indirizzo di broadcast per sistemi operativi della famiglia BSD

Analogamente l'indirizzo X.Y.Z.255 e' l'indirizzo di broadcast generico per sistemi operativi Unix.

Un pacchetto inviato all'indirizzo di broadcast X.Y.Z.0 (o X.Y.Z.255 in generale) di un router, se questo ha abilitata la funzione di broadcasting, viene poi reindirizzato a tutti gli host della rete di Classe C X.Y.Z (192.168.133 nel nostro esempio).