5. Traduzione degli indirizzi di rete in indirizzi fisici: ARP

5.1. Introduzione

Due macchine si parlano solo se conoscono l'indirizzo fisico di sottorete

Due applicazioni si parlano solo se conoscono l'indirizzo IP delle macchine su cui eseguono

Gli indirizzi IP ***sono diversi dagli indirizzi fisici di sottorete***

Allora abbiamo un problema: ***come si traduce (si mappa) l'indirizzo IP in indirizzo fisico?***

5.2. Il problema della risoluzione degli indirizzi

Due macchine A e B sulla stessa sottorete fisica; ciascuna con indirizzo IP I_A e I_B, e indirizzo fisico P_A e P_B. I programmi applicativi devono poter usare I_A e I_B, e occorre che il software di rete sappia tradurli negli indirizzi fisici P_A e P_B: questo è il problema della traduzione degli indirizzi di rete in indirizzi fisici.

5.3. Due tipi di indirizzi fisici

Ci sono due tipi di indirizzi fisici

- indirizzi grandi e fissati dall'hardware: es. ethernet
- indirizzi piccoli, che si possono facilmente configurare nell'hardware e quindi sono manipolabili

Il secondo caso è più semplice perché si può adattare l'indirizzo fisico in modo che la risoluzione sia semplice

5.4. Risoluzione mediante mapping diretto

Supponiamo di avere una sottorete i cui indirizzi fisici sono dei piccoli interi, ad esempio minore di 254: possiamo scegliere la parte hostid dell'indirizzo IP uguale all'indirizzo fisico

Risoluzione assolutamente banale, con poche istruzioni di macchina

In generale, se l'indirizzo fisico è determinabile, si può scegliere una funzione *f* che traduce l'indirizzo IP nell'indirizzo fisico: risolvere l'indirizzo IP significa calcolare

$$P_A = f(I_A)$$

Naturalmente *f* deve essere semplice da calcolare, altrimenti il gioco non vale la candela. Se l'indirizzo fisico può essere scelto, si può determinare una funzione *f* facile da calcolare

Se l'indirizzo IP non può essere scelto, si può ricorrere ad una tabella di traduzione che viene acceduta mediante una funzione hash

5.5. Risoluzione mediante binding dinamico

Ethernet è un caso in cui la risoluzione dell'indirizzo IP è difficile perché gli indirizzi ethernet sono più grandi (48 bit) dell'indirizzo IP, e sono fissati nell'hardware della scheda (ed è molto pesante cambiare la configurazione della scheda perché l'indirizzo sia determinato via software). Quando si cambia la scheda (perché guasta) cambia l'indirizzo ethernet

Non vogliamo ricompilare il software di rete, né avere delle grandi tabelle centralizzate di traduzione.

Ma ethernet ha la funzionalità di broadcasting, su cui si basa un protocollo di basso livello di risoluzione degli indirizzi chiamato ***Address Resolution Protocol (ARP)***, descritto nella prossima figura

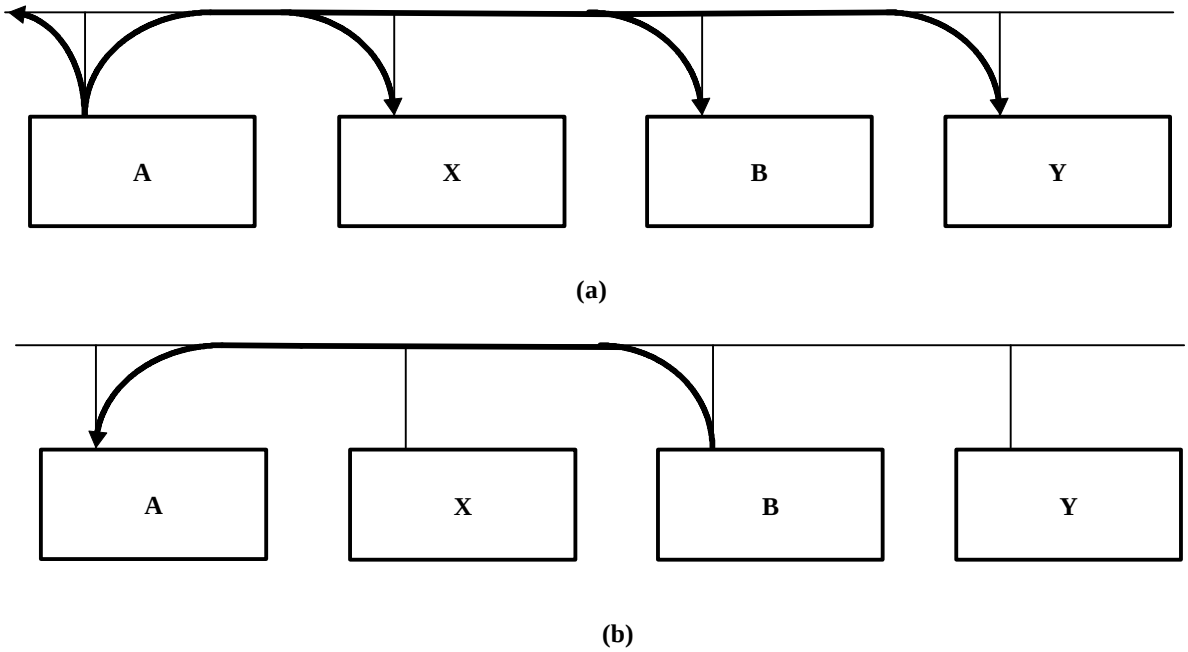


Figura 5-1

Quando A vuole risolvere l'indirizzo I_B, invia in broadcast un pacchetto ARP con il quale chiede all'host di indirizzo I_B di rispondere (notare che risponde senza broadcast) con il suo indirizzo fisico P_B

Tutti ricevono il broadcast ma solo B risponde perché tutti gli host conoscono il proprio indirizzo IP (notare bene!)

A questo punto (b), A conosce la corrispondenza fra indirizzo IP e fisico e può inviare a B il pacchetto IP che desidera

5.6. La cache di risoluzione degli indirizzi

È troppo costoso mandare un pacchetto ARP per ogni pacchetto IP: usare una cache in cui vengono tenute le corrispondenze fra IP e fisico

Anche una piccola cache è molto utile perché le macchine non parlano con tutte le macchine della sottorete

5.7. Raffinamenti di ARP

1. Se A manda un pacchetto a B, probabilmente B dovrà rispondere: inviare nel pacchetto ARP di richiesta anche il legame fra IP e fisico di A, e B metterà la corrispondenza nella sua cache
2. Tutte la macchine della sottorete vedono la richiesta di A, che contiene il legame per A, e quindi la possono mettere in cache
3. Quando una macchina appare sulla rete (fa boot) si annuncia sulla rete con un pacchetto ARP che contiene il legame IP-fisico della macchina (e tutti lo mettono in cache)
4. Le cache devono essere svuotate periodicamente perché
   4.1. Diventano troppo grandi
   4.2. Le macchine possono cambiare il proprio indirizzo fisico (per guasto alla scheda)

5.8. Relazione fra ARP e altri protocolli

ARP sarebbe inutile se le sottoreti conoscessero gli indirizzi IP

ARP impone un nuovo schema di indirizzi al di sopra di quello fisico

> ***ARP è un protocollo di basso livello che nasconde lo schema di indirizzamento fisico e permette di usare uno schema di indirizzamento indipendente dalla tecnologia di sottorete e scelto su criteri di facile gestione della internet***

Internet considera ARP una estensione dei protocolli di accesso fisico al mezzo di comunicazione; potrebbe essere considerato come parte integrante dello strato di rete

5.9. Implementazione di ARP

1. Determinazione degli indirizzi fisici per un pacchetto IP uscente
2. Gestione pacchetti ARP ricevuti

5.9.1. Risoluzione per pacchetti uscenti

Se l'indirizzo destinazione è in cache si risolve; se non lo è si manda una richiesta broadcast, ma:

1. La macchina target può essere down
2. La macchina target può essere molto lenta a rispondere
3. Il pacchetto ARP può essere perso
4. Altre applicazioni possono chiedere di mandare allo stesso indirizzo IP
5. Per un po' di tempo occorre memorizzare il pacchetto uscente, in attesa del binding
6. Occorre cancellare una entry nella cache dopo un certo tempo che vi è stata messa (sostituzione scheda!)

5.9.2. Gestione dei pacchetti ARP in arrivo

Prima cosa: quando arriva un pacchetto ARP si estrae il legame IP-fisico del mittente e si aggiorna la cache

ARP request: la macchina ricevente controlla se lei è il target della richiesta, nel qual caso costruisce il pacchetto di risposta con il proprio legame IP-fisico

ARP reply: risalire dalla risposta ai pacchetti IP in attesa di essere inviati perché privi di risoluzione dell'indirizzo (possono essere più di uno o nessuno (perché?))

5.10. Incapsulamento e identificazione di ARP

Come per tutti i protocolli, i pacchetti di ARP devono viaggiare "portati" da una trama del livello inferiore, in questo caso della sottorete. Abbiamo quindi la seguente situazione

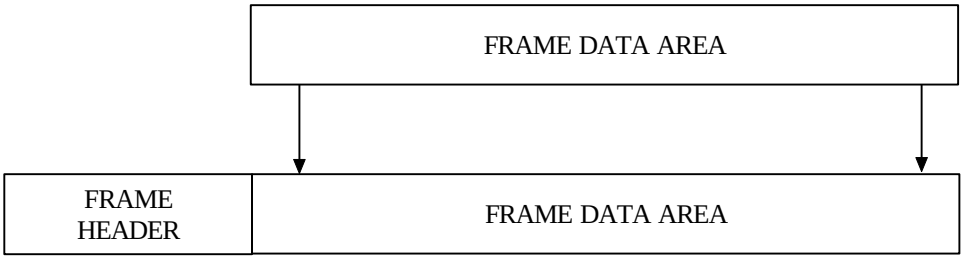


Figura 5-2

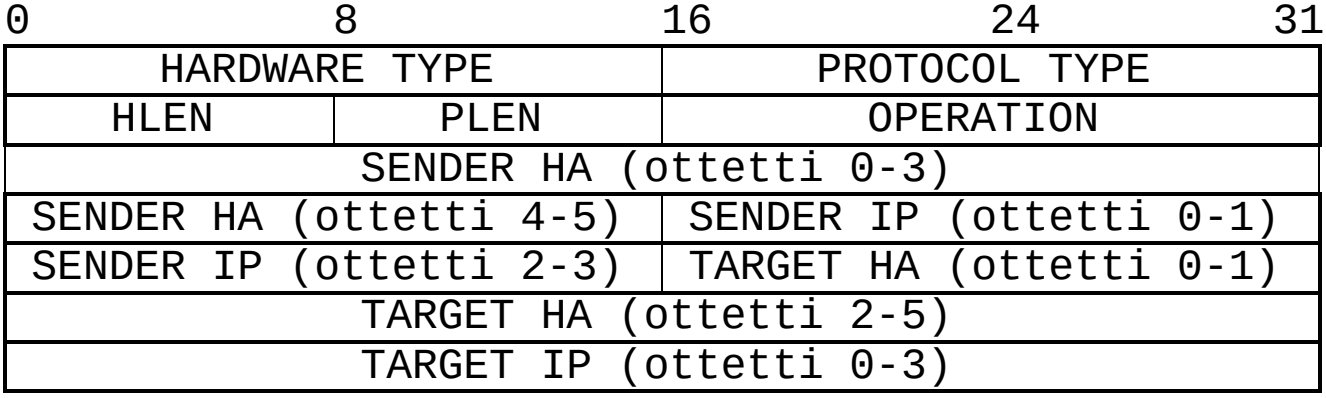
Questo si chiama ***incapsulamento***

Quando il frame viene ricevuto, occorre identificarlo come appartenente ad ARP, perché vi sono tanti protocolli che viaggiano incapsulati nel frame fisico

Come ricordate (!), il frame header di ethernet contiene un campo di 16 bit detto ***frame type***: in questo campo AARP mette il valore 0806_{16} che identifica il protocollo ARP; in questo modo il pacchetto viene passato al software "giusto" che lo sa interpretare

5.11. Formato del protocollo ARP

Non ha un formato fisso, per potersi adattare a diversi frame fisici e diversi formati di indirizzo di network (non solo IP). Qui mostriamo il formato per ethernet e IP



Data la lunghezza variabile, non si ha allineamento a 32 bit. Il frame di ARP è autodescritto come in molti protocolli, cioè contiene campi che servono per interpretare il frame e capirne l'uso

HARDWARE TYPE e PROTOCOL TYPE identificano il mezzo fisico e il protocollo che sta usando ARP; definiti negli standard

OPERATION specifica il tipo di pacchetto ARP (ad es. richiesta=1, risposta=2, richiesta RARP=3, risposta RARP=4 (RARP è una variante di protocollo che usa lo stesso frame))

HLEN e PLEN contengono la lunghezza in ottetti dell'indirizzo fisico e di quello di network (6 e 4 nel nostro caso)

SENDER HA e SENDER IP contengono gli indirizzi hardware e IP del mittente, (ci sono per qualunque OPERATION)

Nei pacchetti di richiesta c'è il TARGET IP nel caso ARP e il TARGET HA nel caso RARP

Nei pacchetti di risposta ci sono tutti e quattro gli indirizzi, ma il mittente e il target sono scambiati; il mittente chi risponde

6. Cenni al protocollo RARP

6.1. Introduzione

Come fa una macchina a sapere il suo indirizzo IP per ognuna delle interfacce che ha installate? Risposta banale: in un file di configurazione! Ma:

- Ci sono anche macchine senza disco
- È un grande costo configurare tutte le macchine di una rete (che possono anche essere molto distanti far loro
- Si usano male gli indirizzi IP quando le macchine in uso sono molte meno di quelle che si possiedono (non sempre sono in sede (portatili), sono spente, etc)
- Quando si deve cambiare qualcosa nella configurazione di rete occorre riconfigurare manualmente tutte le macchine

L'indirizzo IP di una macchina è tenuto in un server della rete al quale la macchina lo chiede prima di entrare in servizio. Ma come fa a comunicare se non ha indirizzo IP?

In realtà non serve comunicare via internet, basta usare la sottorete fisica, il cui indirizzo fisico è nell'hardware ed è unico sulla sottorete: RARP! Anche qui usiamo il broadcast della sottorete

6.2. Reverse Address Resolution Protocol (RARP)

È una variante dell'ARP e usa lo stesso formato di messaggio

L'ethernet frame type field contiene un diverso valore (8035_{16})

Nella richiesta viene specificato l'hardware address del mittente e nel campo SENDER IP si usa l'indirizzo "questa macchina" fatto di tutti 0, inoltre si indica il come TARGET HA il proprio indirizzo fisico. Il server risponde compilando la propria coppia HA-IP e quella richiesta (quindi quella del mittente).

Notare che si può nello stesso modo chiedere il mapping anche per un'altra macchina di cui si conosce l'hardware address, perché il server invia la risposta all'indirizzo hardware che era mittente nella richiesta

Come fa il server a sapere quale è l'indirizzo IP del richiedente? In RARP si basa su tabelle di configurazione, che rendono più semplice la configurazione di una rete (in unico posto).

Altri servizi (ad es. DHCP) permettono di assegnare dinamicamente l'indirizzo IP prendendolo da un intervallo di indirizzi IP configurato

6.3. Temporizzare le transazioni RARP

I pacchetti di RARP si possono perdere per colpa della rete o perché il server è troppo carico

Il richiedente deve quindi ritentare, dopo un certo tempo e per un certo numero di volte

6.4. RARP server primari e di backup

Avere più server è utile per resistere a guasti o a sovraccarichi di un singolo server, ma aggiunge traffico alla rete (molte risposte)

Prima soluzione: ad ogni macchina richiedente viene assegnato un ***server primario***. Gli altri non rispondono alla prima richiesta, ma solo alla eventuale seconda fatta per perdita della prima

Seconda soluzione: Ogni macchina che non è primaria risponde con un ritardo casuale e in questo modo non si affollano tutte assieme a rispondere (diminuisce il carico di rete dovuto all'accesso simultaneo per le risposte)

9. IP: Messaggi di errore e di controllo (ICMP)

9.1. Introduzione

La quarta funzione fondamentale di un servizio di comunicazione di livello di rete è la segnalazione, alla sorgente, di situazioni di errore o di anomalia, in modo che possano essere evitate e corrette

Una funzione correlata è quella di fornire strumenti di controllo e valutazione del funzionamento dell'internet

9.2. Internet Control Message Protocol (ICMP)

Nessun sistema funziona sempre correttamente! Ragioni di fallimento:

- fallimento delle linee di comunicazione e dei processori
- la macchina destinazione e disconnessa temporaneamente o permanentemente (linea commutata)
- router intermedi congestionati (troppo traffico) per cui scartano il traffico

Essendo internet costruita sopra all'hardware di comunicazione, non si hanno informazioni hardware sui malfunzionamenti remoti

ICMP è un protocollo a se stante. È ***obbligatorio*** implementare ICMP assieme a IP. I messaggi di ICMP viaggiano nella parte dati di IP, ma il destinatario è IP stesso. Un utente di IP viene però informato se è all'origine del malfunzionamento o ne può essere interessato

ICMP non è ristretto ai router; anche gli host, con qualche restrizione possono inviare messaggi ICMP

9.3. Segnalazione di errori e correzione di errori

ICMP è un meccanismo di segnalazione di errori: non specifica l'azione da intraprendere a fronte degli errori e non intraprende nessuna azione

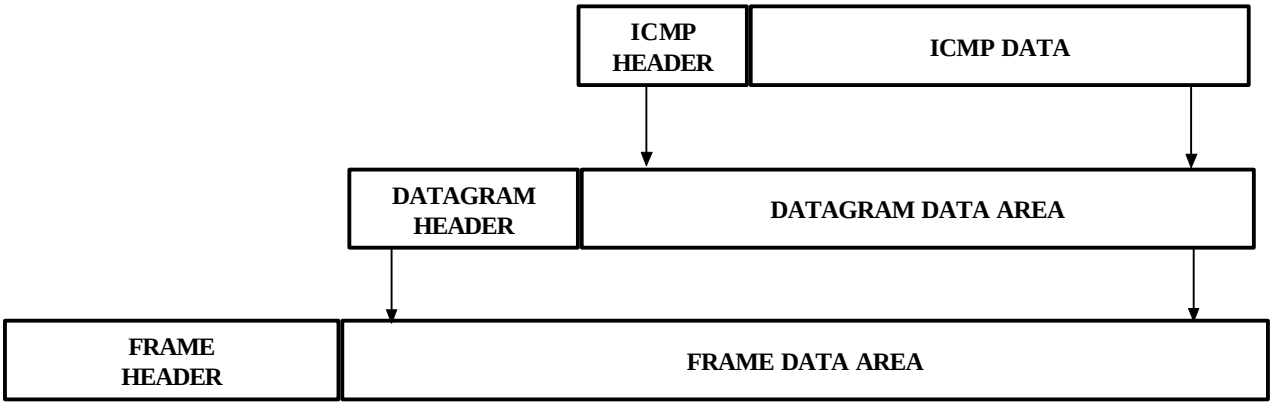
ICMP riporta l'errore alla sorgente del datagramma, e quindi i router intermedi non ne vengono informati (e la strada a ritorno potrebbe essere diversa da quella all'andata!). La sorgente del datagramma a volte non è responsabile dell'errore (ad es. errore di routing) e non è in grado di correggerlo

Perché riportare gli errori alla sorgente?

- il datagramma contiene solo informazioni sulla sorgente e la destinazione
- i router definiscono, e possono cambiare, le proprie tabelle di routing → non esiste un punto centrale in cui ci sia la conoscenza di tutta la internet (a cui riportare l'errore)
- un errore nell'instradamento viene individuato da un router, che non può conoscere la strada seguita dal datagramma
- occorre cooperazione fra gli amministratori degli host e dei router per localizzare e riparare il problema: ***molto difficile!***

9.4. Consegna dei messaggi ICMP

Poiché ICMP è costruito sopra IP, i messaggi ICMP subiscono due incapsulamenti (Fig. 9.1)



Attenzione!!! Anche se i messaggi sono mandati usando IP, ICMP non è considerato un protocollo di livello superiore (brutto!). La ragione è che occorre usare il routing e la frammentazione e riassemblaggio di IP (il campo PROTOCOL vale 1)

9.5. Formato dei messaggi ICMP

Ognuno ha il suo formato, tutti iniziano con tre campi

- TYPE intero 8bit
- CODE 8bit ulteriore identificazione
- CHECKSUM 8bit della sola parte dati

Quelli che riportano errori hanno anche i primi 64bit del datagramma che ha dato errore

Type Field	ICMP Message Type
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect (change route)
8	Echo Request
11	Time Exceeded for Datagram
12	Parameter Problem on a Datagram
13	Timestamp Request
14	Timestamp Reply
15	Information Request (obsolete)
16	Information Reply (obsolete)
17	Address Mask Request
18	Address Mask Reply

9.6. Controllo della raggiungibilità e dello stato di una destinazione

Gli Echo Request e Echo Reply sono fra i più usati. La reply, inviata alla sorgente del datagramma, può opzionalmente contenere una copia del datagramma originale. Viene controllato così che:

- il software IP della macchina sorgente è in grado di instradare verso quella destinazione
- i router intermedi sono in funzione e sanno instradare da sorgente a destinazione
- la macchina destinazione è in funzione (sa rispondere agli interrupt) e sa instradare verso la sorgente
- i router intermedi sono in funzione e sanno instradare da destinazione a sorgente

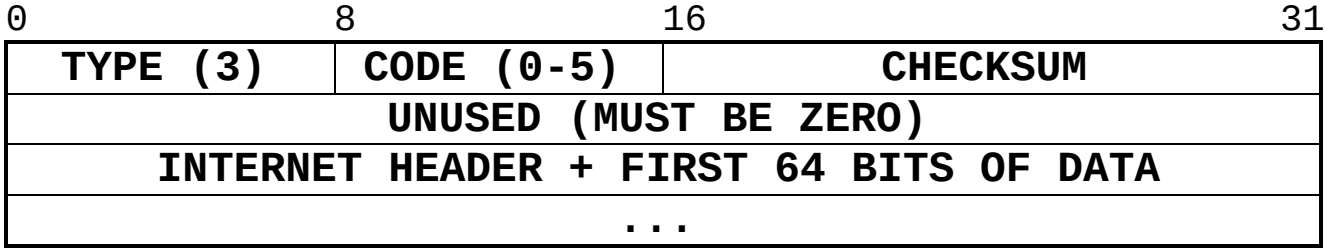
Il programma che usa questa funzione si chiama spesso ***ping***

9.7. Formato Echo Request e Reply

I campi IDENTIFIER, SEQUENCE NUMBER e OPTIONAL DATA permettono al mittente di calcolare statistiche sul ***round trip time*** e tassi di perdita dei pacchetti con pacchetti di lunghezza diversa

9.8. Destinazioni irraggiungibili

Quando un router non può raggiungere una destinazione, invia alla sorgente un messaggio ICMP di destinazione irraggiungibile



Code Value	Meaning
0	Network Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation Needed and DF set
5	Source Route Failed

Un router ***può*** scartare datagrammi quando un errore impedisce di raggiungere la destinazione

Network Unreachable = errore di instradamento

Host Unreachable = fallimento nella consegna, riscontrato dal router (altri possono non essere visibili, vedi es. in ethernet):

- hardware fuori servizio
- indirizzo non esistente

Protocol e port unreachable saranno chiari dopo; gli altri sono autoesplicativi

9.9. Controllo della congestione e del flusso di datagrammi

Connectionless ® i router non possono riservare memoria o risorse di rete per garantire che i datagrammi vengono trattati adeguatamente

I router possono essersi sommersi da traffico che non riescono a smaltire = ***congestione,*** che può arrivare per due ragioni:

- una macchina può generare traffico maggiore di quanto una sottorete può assorbire
- molte macchine generano un traffico aggregato superiore a quanto una sottorete può assorbire

1. ***Singola sorgente:*** una macchina è connessa ad una LAN veloce, e genera traffico verso un host che si trova su una sottorete geografica lenta → il router che si affaccia sulla sottorete lenta riceve più traffico di quanto possa smaltire → congestione
2. ***Molte sorgenti:*** anche se nessuna della macchine genera un singolo flusso di dati superiore alla capacità della sottorete destinazione, il traffico aggregato è superiore a quello che si può smaltire; ad es. il router può avere molte interfacce, tutte della stessa velocità, ma il flusso dei dati e tutto verso una sola linea → congestione

Cosa c'è alla base di questo fenomeno di congestione? Le sottoreti possono avere o meno le seguenti funzionalità:

Riscontro: vi sono elementi di protocollo (messaggi e campi di messaggi) che informano il mittente della corretta (a volta anche della incorretta) ricezione di un messaggio inviato → il mittente ritenta la spedizione se riceve riscontro negativo o non riceve quello positivo entro un certo tempo.
Notare: in assenza di questa funzionalità, messaggi spediti possono essere persi senza che il mittente lo possa scoprire (es. ethernet)

Controllo di flusso: vi sono elementi di protocollo (messaggi e campi di messaggi) che informano il mittente della disponibilità del ricevente a ricevere nuovi messaggi → il mittente spedisce solo se ha l'assicurazione che il ricevente ha la possibilità di trattare il messaggio (buffer di memoria, tempo di CPU). I ricevente può fare ***back-pressure!***
Notare: in assenza di tale funzionalità messaggi potrebbero essere persi non per errori di comunicazione, ma per mancanza di risorse nel ricevente (es. ethernet: anzi questa è praticamente l'unica ragione per cui si perdono messaggi con ethernet). In genere se si ha riscontro si ha anche controllo di flusso (per evitare ritrasmissioni inutili), raramente si trova controllo di flusso senza riscontro.

Da cosa dipende la quantità di dati che si riesce ad inviare su una sottorete?

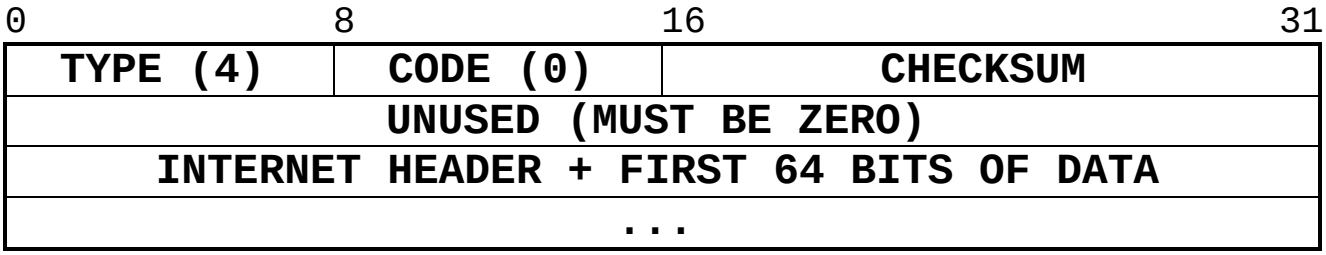
1. Dalla velocità di trasmissione del mezzo fisico: numero di bit al secondo che si riesce a "pompare" dentro al mezzo (es. ethernet = 10Mbit/sec)
2. Dalla velocità di gestione hardware/software del mezzo fisico: i messaggi vanno preparati, prelevati dall'hardware rispondendo alle interruzioni, ... Tutto ciò provoca (può provocare) ritardi e la velocità netta è più bassa di quella del mezzo fisico
3. Protocollo di riscontro: la mancanza di riscontri provoca ritrasmissioni e quindi il traffico realmente smaltito si abbassa
4. Protocollo di controllo di flusso: Se il ricevente è più lento a trattare messaggi del mittente, il mittente viene rallentato, velocità ancora abbassata

Cosa succede se il traffico da inviare è più alto di quello che viene assorbito dalla sottorete? IP accoda i messaggi in uscita. Se la situazione di sbilancio di velocità perdura, finiscono i buffer assegnati a quella sottorete (non si possono assegnare tutti i buffer ad una sottorete! Perché?)

Quando i buffer sono finiti, si può fare back-pressure sulla sottorete che origina il traffico? Se si è in uno schema connection-oriented, si può perché si conosce in anticipo la destinazione del traffico che arriva da una sottorete. In uno schema connectionless (come è IP) non si può fermare una sottorete, perché da quella sottorete giungono pacchetti per molte destinazioni, non solo quella congestionata → quando arriva un pacchetto che deve essere instradato verso la congestionata ***viene scartato!***

Si può però informare la sorgente che il suo pacchetto è stato scartato per congestione: ***source quench (smorzare la sorgente)***

9.10.Messaggio di Source Quench



Normalmente, un messaggio come questo viene inviato alla sorgente ogni volta che si scarta un pacchetto per congestione, oppure quando le code cominciano a essere lunghe, oppure solo verso le sorgenti che hanno tasso di invio alto.

La sorgente rallenta il tasso di invio dei pacchetti, fino a che non riceve più Source Quench. Poiché non esiste un messaggio che chieda di accelerare, la sorgente dopo un po' rialza il tasso di spedizione

Notare che la sorgente smorzata non è necessariamente la causa della congestione, potrebbe essere l'unica a ricevere Source Quench e quindi a rallentare!

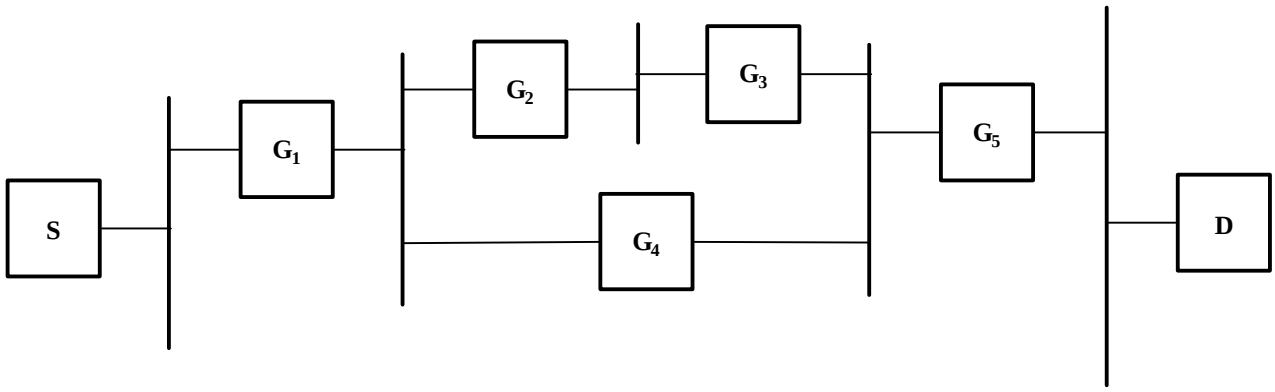
9.11.Richieste di cambiamento di route

I router si scambiano informazioni sull'instradamento per mantenerlo corretto e aggiornato con la corrente topologia dell'internet (modifiche temporanee o permanenti)

> ***Si suppone che i router conoscano le strade corrette; gli host cominciano (a boot time) con informazioni di routing minimali e poi imparano nuove strade dai router***

Nel file di configurazione dell'host in genere c'è solo il default router. Quando un router si accorge che un host sta usando una strada non ottima lo informa con il messaggio ICMP di ***redirect*** (oltre a instradare correttamente il datagramma)

Non bastano per propagare le strade in generale, perché sono limitati ai rapporti fra due macchine direttamente connesse



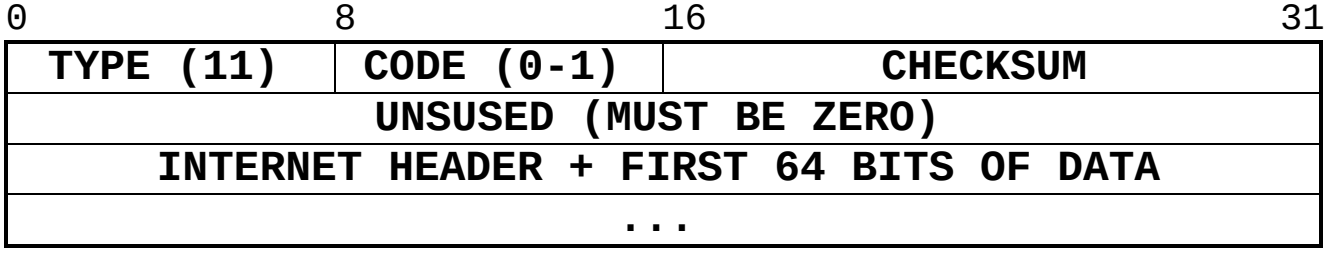
Se G_1 instrada scorrettamente per D via G_2, chi se ne accorge, ad es. G_5, non può avvertire G_1 perché non ne conosce l'indirizzo (non sa nemmeno che la "colpa" è di G_1!). Vedremo i protocolli per propagare le informazione sull'instradamento attraverso l'internet

0	8	16	31
TYPE (5)	**CODE (0-3)**	**CHECKSUM**	
ROUTER INTERNET ADDRESS			
INTERNET HEADER + FIRST 64 BITS OF DATA			
...			

Code Value	**Meaning**
0	**Redirect datagrams for the Net (obsolete)**
1	**Redirect datagrams for the Host**
2	**Redirect datagrams for the Service Type and Net**
3	**Redirect datagrams for the Service Type and Host**

9.12.Individuazione di strade circolari o eccessivamente lunghe

La protezione è data dal time-to-live. Quando un pacchetto viene scartato per time-to-live diventato zero, il router invia un messaggio ICMO di ***time exceeded***. Un host lo invia quando non riesce a riassemblare un pacchetto



Code Value	Meaning
0	Time-to-live count exceeded
1	Fragment reassembly time exceeded

9.13.Segnalazione di altri errori

il messaggio di ***parameter problem*** permette di segnalare errori non coperti dai casi precedenti

9.14.Sincronizzazione dei clock

Due messaggi ICMP permettono di chiedere e ottenere il valore del clock da una macchina al momento della ricezione del pacchetto di richiesta e immediatamente prima della spedizione della risposta

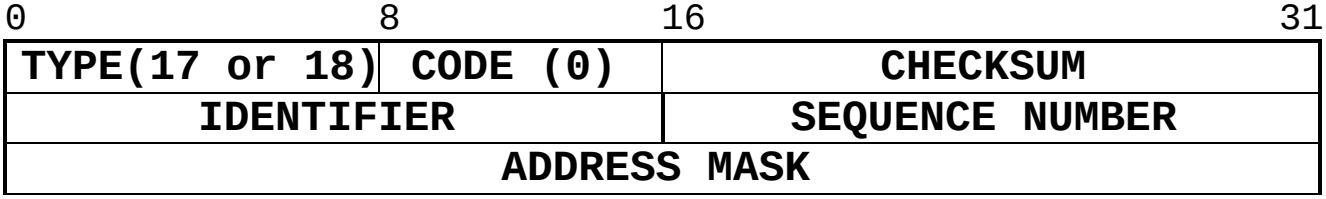
Non serve bene a calcolare il tempo di round-trip

9.15.Richiesta di informazione

Obsoleti, da non usare

9.16.Richiesta di Subnet Mask

Vedremo cosa vuol dire fare ***subnetting***, e quali sono i vantaggi. Un host può aver bisogno di sapere quale è la subnet su una certa rete locale e questa è definita mediate una ***address mask*** di 32bit. Il messaggio di richiesta può essere indirizzato ad un router oppure mandato in broadcast



0	8	16	31
TYPE(17 or 18)	**CODE (0)**	**CHECKSUM**	
IDENTIFIER		**SEQUENCE NUMBER**	
ADDRESS MASK			

9.17. Conclusioni

Abbiamo così visto le quattro funzioni importanti che un servizio di comunicazione di rete deve fornire:

1. un indirizzamento uniforme e globale
2. una funzione consegna di pacchetti che includa frammentazione e riassemblaggio de pacchetti per rendersi indipendenti dalle caratteristiche fisiche delle sottoreti attraversate
3. instradamento fra le sottoreti che costituiscono l'internet, dalla rete sorgente alla rete destinazione: i datagrammi viaggiano attraverso l'internet, da un router ad un altro, fino a che possono essere consegnati direttamente attraverso un sottorete fisica
4. la segnalazione, alla sorgente, di situazioni di errore o di anomalia, in modo che possano essere evitate e corrette