Capitolo 7:

Internetworking TCP/IP

Lo sviluppo delle reti locali ha consentito la nascita di applicazioni e servizi di comunicazione dati su larga scala. Ben presto si è sentita la necessità di estendere la capacità di comunicazione al di fuori dei confini della rete locale anche tra reti diverse e tra calcolatori eterogenei.

Nella prima metà degli anni 70 la DARPA (Defence Advanced Research Project Agency) finanziò degli studi per la progettazione di una rete a commutazione di pacchetto per l'interconnessione di reti di calcolatori. Il lavoro fu portato avanti dalla Stanford University e dalla BBN (Bolt, Beranek and Newman) e portò alla fine degli anni 70 alla definizione di una famiglia di protocolli denominata Internet Protocol (IP) suite. La prima rete basata su IP suite fu la rete ARPAnet, costruita all'interno dello stesso progetto di ricerca, ed ebbe da subito un elevato successo nell'interesse del mondo accademico.

Attualmente si associa alla IP suite il nome TCP/IP anche se in realtà nell'architettura dei protocolli IP rappresenta il protocollo di livello rete e TCP uno dei protocolli di livello trasporto. Dal piccolo numero di utenti di ARPAnet si è passati ad un numero enorme di utenti e alla possibilità di interconnessione mondiale simile a quella della rete telefonica classica.

In Figura 1 è mostrato lo schema dei protocolli della suite IP.

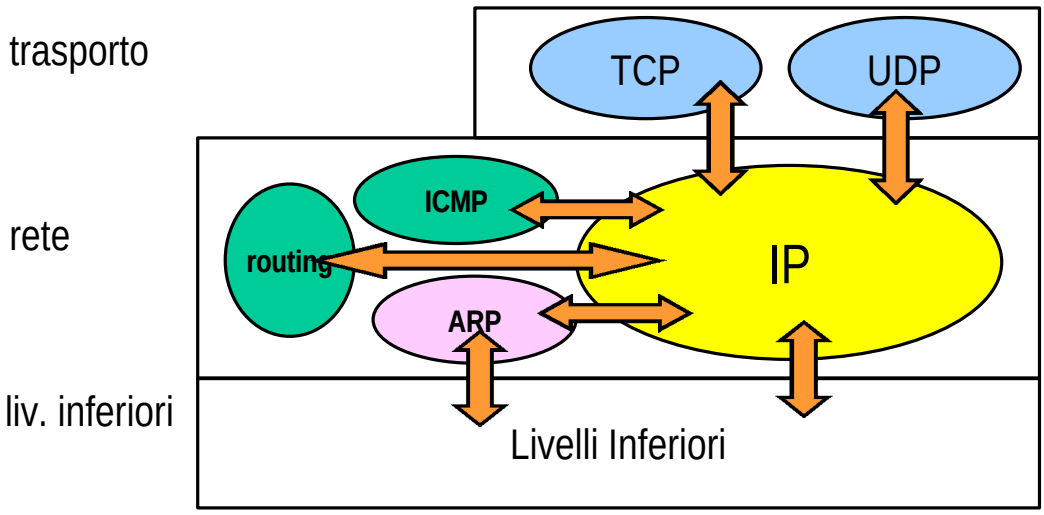


Figura 1: architettura della IP suite

Come si può osservare i livelli inferiori non sono specificati perché l'architettura serve per collegare reti anche eterogenee e quindi caratterizzate da protocolli differenti. La pila dell'IP si appoggia alla pila dei protocolli della rete che deve interconnetere, che per questo viene chiamata sotto-rete. Le sottoreti sono interconnesse da nodi che implementano il livello rete di IP detti *gateway* o *router* nella terminologia IP (Figura 2). Ai livelli inferiori sono richieste alcune funzionalità minime che sono fondamentalmente la capacità di instradare localmente i pacchetti con una qualsiasi modalità, un indirizzamento locale sulla quale basare una corrispondenza con gli indirizzi IP, e la possibilità di effettuare un indirizzamento broadcast.

Il livello di rete è gestito dal protocollo IP, ma,sempre a livello 3 sono presenti altri elementi come il protocollo ARP (Address Resolution Protocol), che serve a mappare gli indirizzi della sotto-rete in indirizzi IP, ICMP (Internet Control Message Protocol), che serve a scambiare semplici messaggi di servizio, e i

protocolli di routing che servono a far scambiare messaggi tra i nodi commutazione per definire il routing da adottare in modo distribuito..

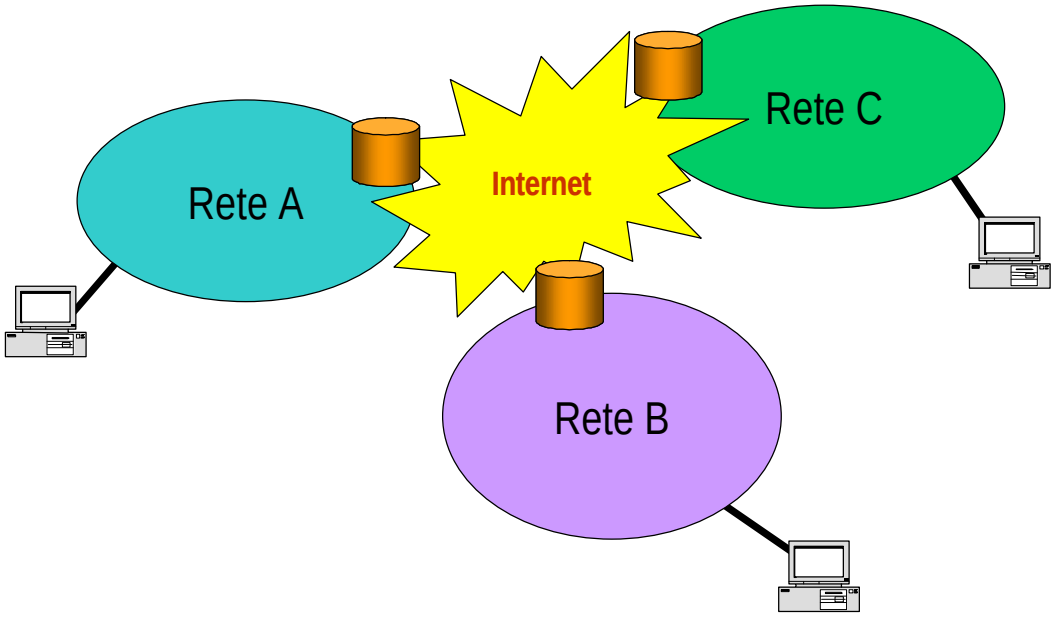


Figura 2: interconnesione di sotto-reti tramite Internet

Il livello di trasporto può far uso del protocollo TCP (Transmission Control Protocol), che effettua un trasporto orientato alla connessione, o del protocollo UDP (User Datagram Protocol) che effettua un trasporto senza connessione.

I protocolli applicativi si appoggiano direttamente sul livello di trasporto.

1. Il protocollo IP

Il protocollo Internet è un protocollo di rete che opera appoggiandosi direttamente ai servizi di trasporto dell'informazione forniti dai livelli sottostanti. Tali livelli inferiori possono essere quelli di una rete locale o quelli di una qualunque altra pila di protocolli in grado di trasportare in qualche modo dei pacchetti.

Dovendo poter utilizzare qualunque tipo di sotto-rete senza limitazioni, per il protocollo IP si è scelta una modalità di trasferimento di tipo datagram senza garanzia di consegna del messaggio. Questa è stata infatti la strada più semplice, avendo come requisito quello di doversi appoggiare sulle reti più disparate che offrono servizi molto diversi tra loro. Si poteva, in linea di principio, anche optare per la soluzione a circuito virtuale, ma in questo caso le funzioni del livello di interconnessione sarebbero diventate molto complesse.

Le funzionalità direttamente legate al colloquio con i livelli inferiori di trasporto (SubNetwork Dependent Convergence Protocol) devono operare la eventuale frammentazione del pacchetto Ip, nel caso la sotto-rete richieda messaggi più corti. I pacchetti frammentati non vengono ricostruiti nodo per nodo, ma solo una volta giunti a destinazione. Infatti, a causa della modalità datagram, frammenti diversi possono in linea di principio seguire strade diverse e quindi possono essere riassemblati solo a destinazione. Il controllo d'errore e il controllo di flusso vengono lasciati al livello superiore.

Il formato dei pacchetti usati da IP nella versione 4 del protocollo (quella usata attualmente), detti trame IP, è mostrato in Figura 3. La trama deve essere un multiplo di 32 bit e i dati appesi devono avere lunghezza multipla di 8 bit.

Il campo versione indica la versione del protocollo IP in uso e vale in questo caso 4.

Il campo Lenght indica la lunghezza dell header che non è fissa a causa della possibilità di introdurre in coda all'heder dei campi Options che servono ad estendere alcune funzionalità e servizi.

Il campo TOS (Type of Service) è stato pensato per indicare la qualità di servizio richiesto. Oggi è inutilizzato ma sono in corso di definizione le modalità con la quale verrà usato tale campo.

Il campo Total lenght indica la lunghezza del datagram, il cui valore massimo è fissato a 65536 ottetti.

Il campo Identification viene usato per identificare i diversi frammenti di un datagram originale (se è necessaria la frammentazione il campo lunghezza viene modificato).

Seguono tre flag. Il primo è il D bit (don't fragment) posto ad 1 quanto il mittente non desidera la frammentazione del pacchetto. In questo caso, se la sottorete non accetta il messaggio senza frammentare, il datagram viene scartato. Il secondo flag è il bit M di "more" che indica la presenza di successivi frammenti. Il campo inizio Fragmento Offset indica la posizione del frammento all'interno del messaggio originale.

Il campo TTL (time to live) è un campo che viene riempito dal mittente e decrementato ad ogni attraversamento di nodo IP, in linea di principio, di una quantità pari al tempo di stimato di percorrenza dell'ultimo link. Quando tale campo raggiunge il valore zero il datagram viene scartato. Questo meccanismo serve per eliminare eventuali messaggi che non trovano la loro destinazione o per messaggi in cui un limitato tempo di consegna è essenziale. A causa del limitato numero di bit del campo TTL, nella pratica il suo uso è molto semplificato e viene legato al numero di collegamenti tra nodi IP, detti *router*, attraversati. ogni router che inoltra il pacchetto decrementa il TTL di 1 unita'.

Il campo protocol indica l'indirizzo del SAP verso il livello superiore, ovvero il livello di trasporto.

L'header checksum è un campo per il controllo d'errore costituito da bit di parità calcolati su tutto l'header.

Il campo options viene utilizzato per indicare un numero variabile di opzioni che costituiscono delle estensioni dell'header. Ciascuna opzione a sua volta è suddivisa fra identità dell'opzione, lunghezza del campo opzione e contenuto dell'opzione. Fra le opzioni definite vi sono, ad esempio, il livello di sicurezza, il source routing, l'identificazione di una connessione, e il time stamp.

I campi source address e destination address contengono gli indirizzi IP di sorgente e destinazione della trama. Gli indirizzi sono costituiti da 32 bit nella versione 4 del protocollo. Le modalità di gestione degli indirizzi in IP necessita di un approfondimento.

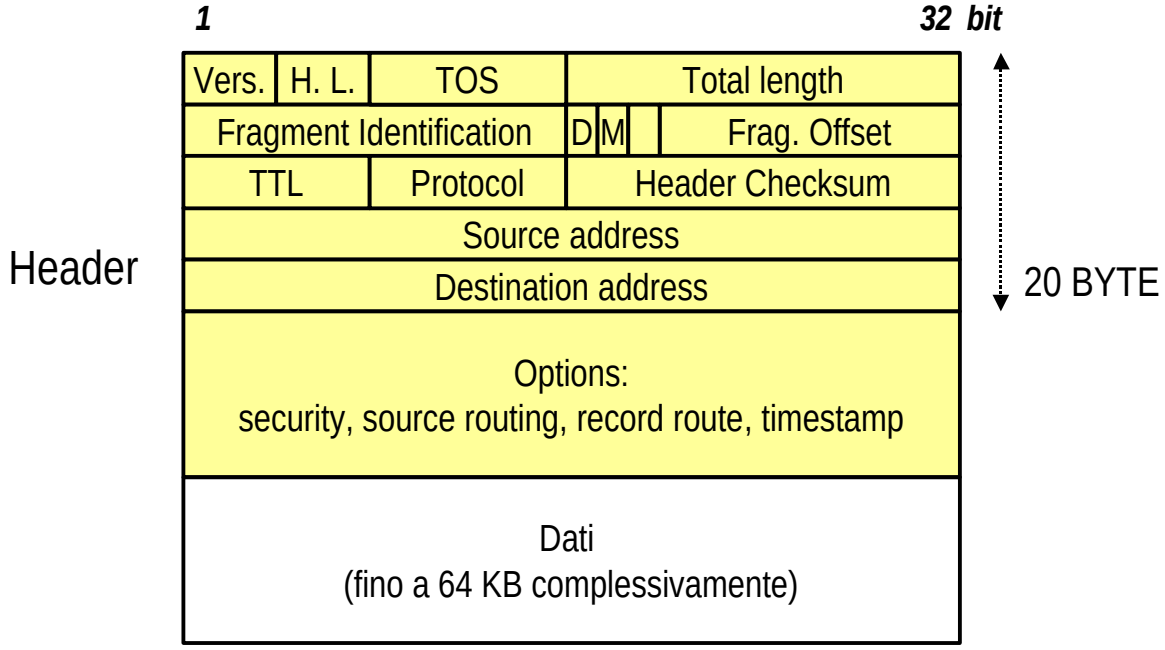


Figura 3: le trame del di IPv4

1.1 L'indirizzamento

L'Internet Protocol introduce un indirizzamento di tipo gerarchico su due livelli, un livello rete e un livello *host*. Il livello rete serve ad identificare le varie sotto-reti che la rete IP deve interconnetere mentre il livello host identifica gli apparati d'utente, chiamati appunto host nella terminologia Internet, all'interno della rete.

Il campo di indirizzo è composto da 32 bit suddivisi in quattro ottetti. Comunemente ciascun ottetto è rappresentato in forma decimale e dunque può assumere valori tra 0 e 255, e i diversi ottetti dell'indirizzo sono separati da punti (ad esempio: 131.175.21.173).

L'indirizzo non identifica un nodo o un end-system della rete, ma una interfaccia verso la rete. Se un nodo ha più interfacce verso la rete, ognuna di essa avrà un differente indirizzo IP. Questa scelta si giustifica con l'associazione di uno dei livelli gerarchici dell'indirizzamento con le sotto-reti da interconnetere. Un nodo con più interfacce verso la rete, sarà in generale collegato a sotto-reti differenti con le diverse interfacce e sarebbe dunque impossibile associare un solo indirizzo al nodo rispettando allo steso tempo il principio della divisione dei livelli gerarchici.

I due livelli gerarchici, rete ed host, non occupano una porzione fissa del campo indirizzo. Nella prima fase di definizione del protocollo vennero introdotte tre tipologie di rete con una diversa divisione tra il campo rete e il campo host e distinte dal valore dei primi bit del primo ottetto come indicato in Figura 4.

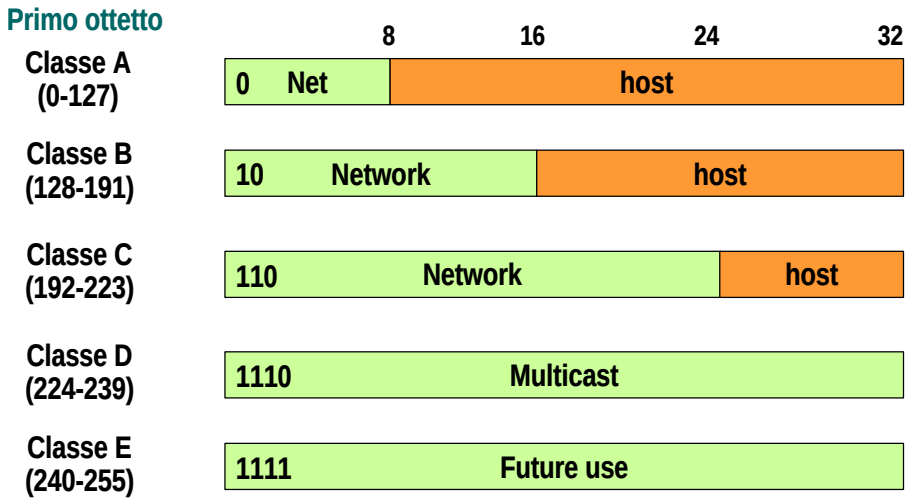


Figura 4: le classi degli indirizzi IP

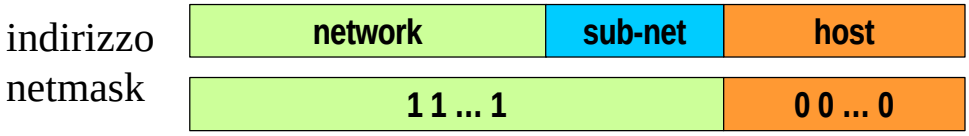
I tre diversi tipi di campi di indirizzamento consentono di avere rispettivamente 128 reti di classe A (primo ottetto 0-127), 16.384 reti di classe B (primo ottetto 128-191), 2.097.152 reti di classe C (primo ottetto 192-223). Esistono inoltre un gruppo di indirizzi dedicato alle applicazioni multicast (classe D) e un gruppo per usi futuri).

Alcuni indirizzi hanno significati speciali. L'indirizzo con il campo host posto a 0 serve ad indicare la sotto-rete il cui indirizzo è contenuto nel campo rete (significato solo indicativo, non usato per i campi source e destination address dell'header IP). Un indirizzo con il campo host di soli 1 assume il significato di indirizzo broadcast della sotto-rete indicata nel campo rete. Un altro indirizzo broadcast molto usato è quello nel quale tutti i 32 bit sono posti ad 1; tale indirizzo indica il broadcast nella stessa sott-rete di chi invia il pacchetto. Quando il campo rete è posto a zero, l'indirizzo indica l'host il cui indirizzo è contenuto nel campo host sulla stessa rete del mittente. Se anche il campo host è posto a zero l'indirizzo indica il mittente stesso del

pacchetto. Infine, l'indirizzo con il primo ottetto pari a 127 e gli altri campi qualsivoglia (normalmente 127.0.0.0) indica il loopback sullo stesso host (usato nei sistemi operativi per testare le funzionalità di rete).

La divisione degli indirizzi IP per gli host nelle tre classi risulta poco efficiente quando occorre flessibilità nella divisione degli indirizzi tra sotto-reti di dimensioni diverse. Per questo è stato successivamente introdotto il concetto di netmask, ovvero un confine variabile tra il campo rete e il campo host definito mediante un maschera di 32 bit composta da una sequenza di 1 in corrispondenza della parte rete e una sequenza di 0 in corrispondenza della parte host. Visto che questa divisione di fatto si sovrappone a quella delle operata dalle classi di solito si dice che la netmask divide il campo rete in due parti, la parte rete e la parte sotto-rete. Rimane ovviamente il requisito che alla sotto-rete debba corrispondere una rete fisica reale in grado di effettuare l'instradamento locale dei pacchetti.

La netmask viene di solito indicata con la stessa simbologia adottata per gli indirizzi IP e associata all'indirizzo a cui si riferisce, oppure viene indicata aggiungendo con un numero dopo l'indirizzo che indica il numero di 1 della maschera (si veda l'esempio in Figura 5).



rappresentazioni equivalenti:

indirizzo:	131.175.21.173
netmask:	255.255.255.0

131.175.21.173/24

Figura 5: la netmask e le sue rapprsentazioni

Gli indirizzi Internet vengono assegnati dalla Internet Assigned Number Authority (IANA) che a sua volta delega varie autorità regionali chiamate Network Information Center (NIC).

1.2 Inoltro delle trame

L'indirizzamento IP consente ai messaggi con di raggiungere la sottorete di destinazione seguendo un cammino che passa da rete a rete attraverso i router che sono in grado di instradare i messaggi IP sulla base di opportune tabelle di instradamento. Il messaggio, una volta giunto in un router che ha una interfaccia nella sotto-rete cui appartiene l'indirizzo di destinazione, viene instradato utilizzando la capacità della sotto-rete di inoltro locale dei pacchetti. Allo scopo viene fatto un mappaggio automatico(descritto nel prossimo paragrafo) tra indirizzo IP e indirizzo della sotto-rete, comunemente indicato come *indirizzo fisico*.

Le tabelle di routing dei router possono essere o statiche o venire aggiornate dinamicamente sulla base di protocolli di routing che, mediante lo scambio di messaggi tra router vicini, consentono una gestione automatica delle tabelle.

Ogni volta che un messaggio giunge in un router viene analizzato l'indirizzo di destinazione e mediante la netmask viene individuata la parte rete. La parte rete dell'indirizzo di destinazione viene confrontata con la parte rete degli indirizzi associati a ciascuna delle interfacce del router. Se vi è una corrispondenza con l'indirizzo di una interfaccia il router provvede al mappaggio dell'indirizzo IP nell'indirizzo fisico e alla consegna del pacchetto alla destinazione. Se invece non vi è corrispondenza, si provvede a consultare le tabelle routing.

Le tabelle di routing contengono un elenco di indirizzi di sotto-reti (parte host posta a 0) e in corrispondenza l'indirizzo di un router indicato come *first-hop* (Figura 6). A ciascun indirizzo di rete è associata anche una netmask. I router first-hop contenuti nella tabella di routing sono router immediantemente vicini, ovvero raggiungibili attraverso una delle sottoreti cui sono collegate le interfacacce del router considerato.

network	netmask	first hop
131.175.21.0	255.255.255.0	131.17.123.254
131.175.16.0	255.255.255.0	131.17.78.254
131.56.0.0	255.255.0.0	131.17.15.254
131.155.0.0	255.255.0.0	131.17.15.254
0.0.0.0	0.0.0.0	131.17.123.254

interface eth0

IP address	131.17.123.1
netmask	255.255.255.0

interface eth1

IP address	131.17.78.1
netmask	255.255.255.0

interface eth2

IP address	131.17.15.12
netmask	255.255.255.0

Figura 6: esempio di tabella di routing e di configurazione delle interfacce

Per scoprire a quale router first-hop deve essere inoltrato il messaggio viene confrontato l'indirizzo di destinazione con gli indirizzi di rete contenuti nella tabella di routing. In particolare viene fatto un AND bit a bit tra indirizzo di destinazione e netmask associata alla riga della tabella e viene confrontato il risultato con l'indirizzo di rete associato. Se il confronto dà esito positivo per più righe della tabella viene selezionata la tabella con la netmask che ha il maggior numero di 1 (si dice comunemente che vale il principio del prefisso più lungo).

Il principio del prefisso più lungo viene comunemente adottato quando nei router della periferia delle rete è conveniente avere una tabella di instradamento corta e quando molte reti di destinazione si raggiungono sempre attraverso lo stesso router di first-hop. In questo caso viene definito una riga con il gateway di defoult cui è associato l'indirizzo di rete 0.0.0.0 e netmask 0.0.0.0. Com'è immediato verificare, questa riga produce un confronto positivo con qualunque indirizzo di destinazione, ma naturalmente ha un prefisso lungo zero. L'instradamento avviene verso il defoult gateway se e soltanto se nessun altra riga dà confronto positivo.

I principi descritti per l'inoltro delle trame IP nei router in reatà sono generali e valgono anche per gli host. Di solito, però, gli host hanno una sola interfaccia e provvedono all'inoltro locale dei pacchetti solo per la sotto-rete associata all'interfaccia. Nella maggior parte dei casi, inoltre, la tabella di routing contine solo la riga del defoult gateway e quindi tutte le trame verso destinazioni fuori dalla sottorete vengono passate ad un solo router.

2. Il meccanismo di ARP

Il meccanismo di ARP (Address Resolution Protocol) è il quello che sovrintende al reperimento degli indirizzi fisici a partire dall'indirizzo Internet, quando la sottorete sottostante sia di tipo IEEE 802, o comunque abbia capacità di tipo broadcast.

Per rendere l'associazione fra Indirizzo Fisico (IF), o indirizzo MAC, e Indirizzo Internet (II) flessibile e non gravare su tabelle di qualche server da configurare e conoscere, si utilizza un protocollo che sfrutta il servizio broadcast messo a disposizione dal livello inferiore. Quando il processo di inoltre delle trama del

livello IP richiede l'inoltro di una trama ad un II appartenente ad una sotto-rete collegata, viene invocato il protocollo ARP che dapprima cerca se l'associazione è presente nella ARP-cache locale e in caso negativo chiede al livello inferiore la trasmissione di una particolare trama broadcast (che raggiunge tutte le destinazioni della sottorete di livello inferiore) che contiene la richiesta di associazione con l'II specificato.

Tutti i nodi della sottorete ricevono la trama di richiesta e il nodo che riconosce il proprio II risponde al mittente segnalando il proprio IF. A questo punto la trama IP, che contiene ovviamente l'II, viene incapsulata nella trama di livello inferiore che utilizza l'IF e viene trasmessa. L'associazione II-IF viene poi immessa nella cache per un più veloce uso futuro.

In linea di principio, essendo la rete di tipo broadcast, anche tutte le altre stazioni che ascoltano lo scambio di messaggi possono approfittare delle associazioni II-IF contenute per aggiornare le loro cache. In pratica l'uso effettivo di questa possibilità dipende dall'implementazione del protocollo nell'host.

In Figura 7 viene indicato il contenuto della trama ARP.

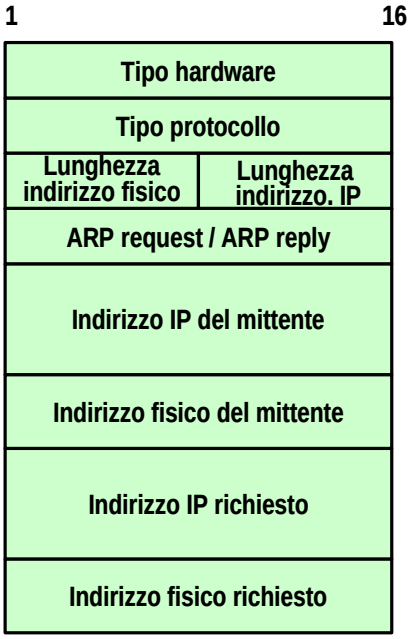


Figura 7: formato dei messagggi ARP

Esiste anche un meccanismo inverso, Reverse ARP o RARP, usato per le workstation che non hanno disco. Queste stazioni non possono essere configurate con gli indirizzi IP, cosicché quando si riaccendono, inviano un messaggio RARP (broadcast) contenente il proprio indirizzo fisico. Il server che sovraintende queste workstation conosce l'associazione e la comunica con un messaggio RARP di risposta.

Il protocollo RARP non è ormai quasi più usato, ma il principio dell'assegnazione dell'indirizzo IP all'host al momento della sua accensione o quando ha necessità di scambiare messaggi in rete è stato conservato. Un protocollo attualmente molto usato che svolge il compito di assegnare dinamicamente gli indirizzi IP agli host è il protocollo DHCP (Dynamic Host Configuration Protocol).

Quando un host ha necessità di avere un indirizzo IP invia un messaggio broadcast di DHCPDISCOVER contente il proprio indirizzo fisico (Figura 8). Tale messaggio può raggiungere uno o più server DHCP in ascolto che rispondono con una messaggio di DHCPOFFER. L'host, una volta selezionato il server, invia un messaggio di DHCPREQUEST, a cui il server risponde con un messaggio di DHCPPACK contente non solo l'indirizzo IP assegnato, ma anche altri importanti parametri come ad esempio la netmask e l'indirizzo del defoult gateway.

Il DHCP è molto usato perché consente di gestire in modo dinamico gli indirizzi IP disponibili. I vantaggi principali solo legati alla possibilità di evitare la configurazione manuale di tutti gli indirizzi in tutti gli host e alla possibilità di gestire un numero di indirizzi minore del numero di host assegnando un indirizzo solo

agli host attivi. Quest'ultimo è il caso dei ASP (internet Access Service Provider) che hanno un limitato numero di indirizzi, ma un numero anche molto elevato di utenti che si connettono alle rete mediante linea telefonica commutata (analogica mediante modem o ISDN).

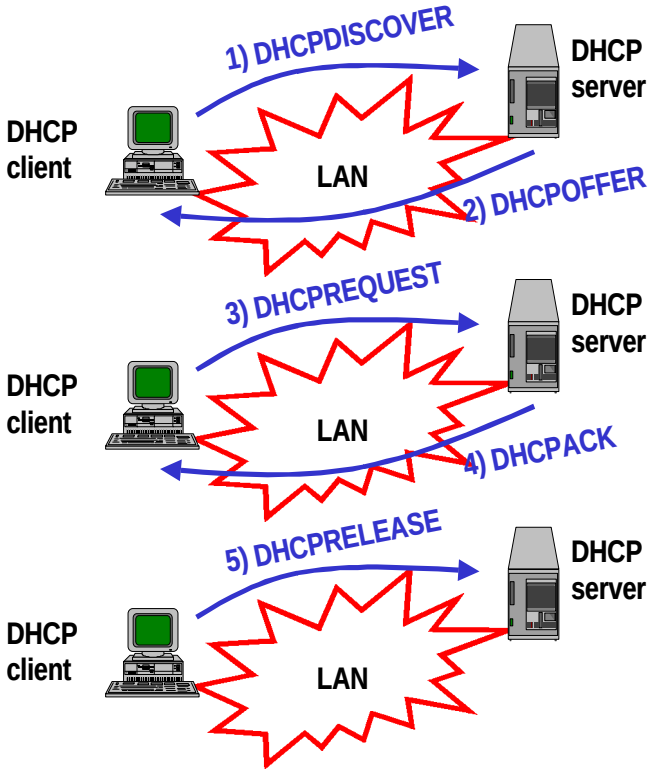


Figura 8: scambio di messaggi per l'assegnazione dell'indirizzo IP con DHCP

3. Il protocollo ICMP

La suite IP fa uso di un protocollo di servizio chiamato Internet Control Message Protocol (ICMP). Tale protocollo è usato per scampi di messaggi di servizio fra host e routers che servono a fornire informazioni sui problemi trovati nell'inoltro dei messaggi. Benché sia logicamente allo stesso livello protocollare di IP, i messaggi ICMP vengono incapsulati nella trama IP come un qualunque altro messaggio.

Fra i messaggi codificati nell ICMP troviamo:

- ✓ *Destination unreachable*, usato per segnalare al mittente che non si è potuto consegnare il messaggio. Le cause possono essere molteplici come ad esempio la caduta di link di rete.
- ✓ *Time exceeded*, indica che il contatore TTL è scaduto prima della consegna.
- ✓ *Parameter Problem*, indica che nell'header è stato rilevato un'errore sintattico o semantico.
- ✓ *Source Quench*, previsto come strumento di controllo di flusso, indica l'esistenza di congestione e la necessità di ridurre il flusso in ingresso (attualmente non è usato ma esistono molte proposte volte a definire meccanismi di notifica esplicita della congestione nella IP suite).
- ✓ *Redirect*, usato quando un router scopre una strada più breve che passa per un altro router.
- ✓ *Echo*, messaggio che forza il ricevente a rispondere con lo stesso messaggio *Echo Reply*, per verificare l'esistenza della via di comunicazione.
- ✓ *Timestamp*, messaggio che serve a misurare i tempi di attraversamento della rete. Questo messaggio porta l'istante di immissione in rete e quando viene ricevuto causa la trasmissione di un *Timestamp*

Reply che ritorna lo stesso Timestamp più un nuovo Timestamp relativo all'istante di trasmissione della risposta.

4. Il livello di trasporto

Il livello di trasporto nelle reti IP è, ovviamente, implementato solo nei sistemi finali e serve a far colloquiare gli applicativi che sono in esecuzione nella macchine remote. Come detto, nelle reti IP i protocolli applicativi usano direttamente i servizi di comunicazione offerti dal livello di trasporto. Quindi, il primo compito che deve svolgere il livello di trasporto è quello di indirizzare i SAP sui sono attestati i diversi processi applicativi in esecuzione su un host, come ad esempio HTTP, FTP, SMTP, ecc (Figura 9).

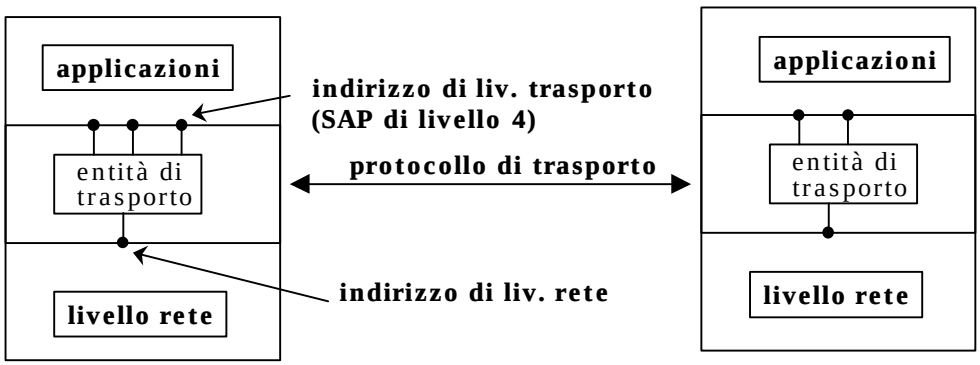


Figura 9: indirizzamento operato dal protocollo di trasporto e dal protocollo di rete

L'indirizzo usato nei protocolli di trasporto della IP suite è chiamato *numero di porta* ed è costituito da 16 bit contenti nell'header. L'insieme di indirizzo IP e numero di porta identifica univocamente un processo in esecuzione su un host e viene spesso indicato con il nome di *socket.*

Il servizio di comunicazione fornito dal livello di trasporto può essere di vari tipi: trasporto affidabile con garanzia di consegna dei messaggi nel corretto ordine, e trasporto non affidabile nel quale viene implementata di fatto la sola funzionalità di indirizzamento. Naturalmente, il servizio realmente fornito all'applicazione dipende anche dal livello rete sottostante.

Nella suite IP sono definiti due tipi di trasporto:

- ✓ TCP (Trasnmission Control Protocol), orientato alla connessione e affidabile
- ✓ UDP (User Datagram Protocol), senza connessione e non affidabile

4.1 Il trasporto UDP

E' il protocollo di trasporto più semplice in grado di usare il servizio di comunicazione e le funzionalità di IP facendo colloquiare processi remoti.

Non aggiunge nulla a IP se non l'indirizzamento delle applicazioni e un blando controllo d'errore sull'header dei messaggi. Quindi è un protocollo che fornisce un servizio di tipo datagram, che non garantisce la consegna e che non esercita nessun controllo sul flusso di dati emesso dall'applicazione.

Il trasporto UDP è usato da tutti quegli applicativi che non necessitano di un trasferimento affidabile e per i quali l'overhead dovuto alla fase di apertura di un servizio di trasporto orientato alla connessione non sarebbe giustificato. Tra questi è possibile ricordare il DNS (Domain Name Service), NFS (Network File System, SNMP (Simple Network Management Protocol), il protocollo di routing RIP (Routing Information Protocol), ecc.

Inoltre, il trasporto UDP è usato dai servizi che non possono tollerare il controllo sul flusso dati tipicamente introdotto dal trasporto a connessione TCP. Tra questi sicuramente di importanza fondamentale nell'immediato futuro sono i servizi di trasporto di flussi stream come voce o video. In quest'ultimo caso, di solito, alle fnzionalità dell'UDP vengono aggiunte le funzionalità di un protocollo aggiuntivo, di fatto classificabile a livello di trasporto, chiamato RTP (Real Time Protocol) che ha come compito principale quello di aggiungere all'header UDP le funzionalità di numerazione dei pacchetti e di time-stamp.

L'header delle trame UDP è mostrato in Figura 10.

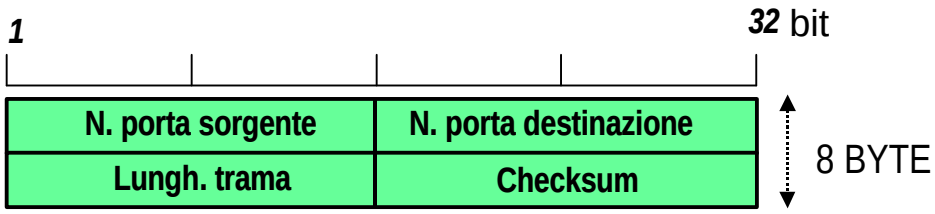


Figura 10: header dell'UDP

Oltre ai campi di numero di porta sorgente e numero di porta destinazione, ciascuno di 16 bit, vi è un campo che indica la lunghezza complessiva della trama e un campo di check-sum calcolato su quello che viene chiamato pseudo-header. Lo pseudo-header è costituito dall'header UDP preceduto dai campi indirizzo IP di sorgente e di destinazione.

4.2 Il trasporto TCP

Il protocollo TCP (Transport Control Protocol) è un protocollo di trasporto pensato per funzionare col protocollo IP ed utilizzato per fornire un servizio di trasporto affidabile di un *flusso di byte* da un applicativo d'utente ad un altro. E' un protocollo full duplex (scambio di dati in entrambe le direzioni), orientato alla connessione, con controllo di flusso e controllo d'errore di tipo GO BACK n.

Il flusso, per venire trasmesso, viene suddiviso dal protocollo in *segmenti* la cui lunghezza massima viene negoziata durante la fase di apertura della connessione (MSS – Maximum Segment Size) e comunque non più lunghi di 64 kbytes dato che il segmento deve poter utilizzare un'unica trama IP.

Dato che l'applicazione scambia flussi di byte con il TCP, è il TCP stesso che decide quando un segmento va trasmesso, anche se l'applicazione ha dei mezzi per forzare la trasmissione di segmenti prima che il TCP intervenga in modo automatico. Anche dal lato ricevente il TCP raccoglie i byte in un buffer che successivamente trasmette al livello superiore. Questa procedura implica che la presa in carico e la consegna dei dati da parte del TCP non garantisce la delimitazione dei dati stessi che, qualora necessaria, deve essere effettuata dal protocollo applicativo superiore.

Il TCP utilizza un unico formato di trame sia per la trasmissione di informazione d'utente che per l'informazione di servizio (apertura e chiusura della connessione, messaggi per il controllo d'errore e e quello di flusso). La trama utilizzata dal TCP è mostrata in Figura 11.

L'header è lungo 20 byte se il campo opzioni non viene utilizzato. Il campo dati può essere vuoto (trame di acknowledgment, connessione, ecc.).

I numeri di porta aggiunti all'indirizzo IP forniscono, come detto, l'identificativo dei SAP di livello Trasporto. In TCP la porta può identificare un preciso tipo di servizio applicativo dal lato server. In particolare, numeri al di sotto di 256 identificano porte dove sono attestati i processi server in ascolto di servizi noti (well-known ports). Per esempio la porta 21 identifica il servizio FTP (File Transfer Protocol), la porta 80 il servizio HTTP (HyperText Transfer Protocol), la porta 23 identifica il servizio TELNET.

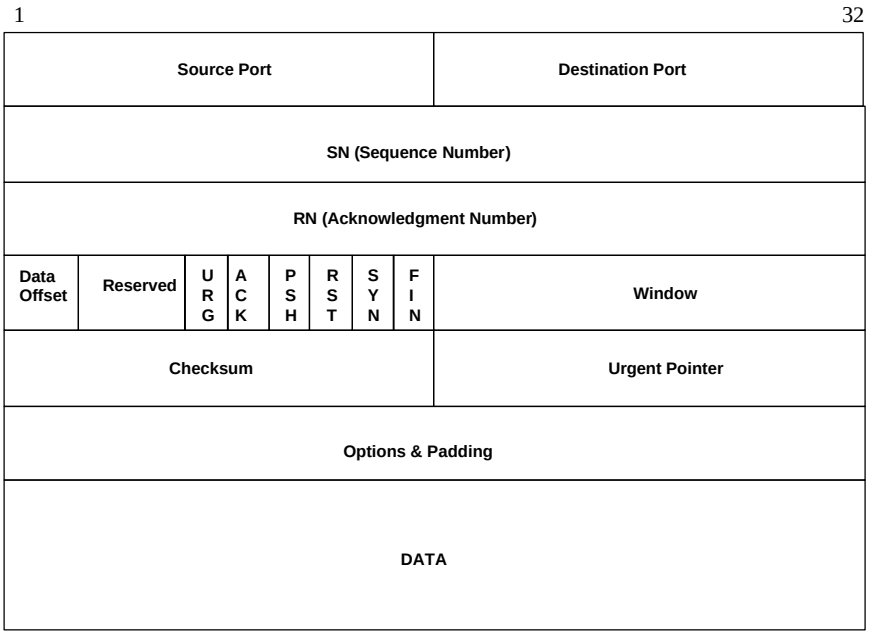


Figura 11: trama TCP

Un numero di porta può essere usato per connessioni multiple, ma l'insieme di socket di sorgente e socket di destinazione identificano univocamente una connessione. Così, ad esempio, un server web può avere moltissime connessioni contemporaneamente attive sulla porta 80, ma riuscire a distinguere le diverse connessioni sulla base dell'indirizzo IP e del numero di porta dei client (Figura 12).

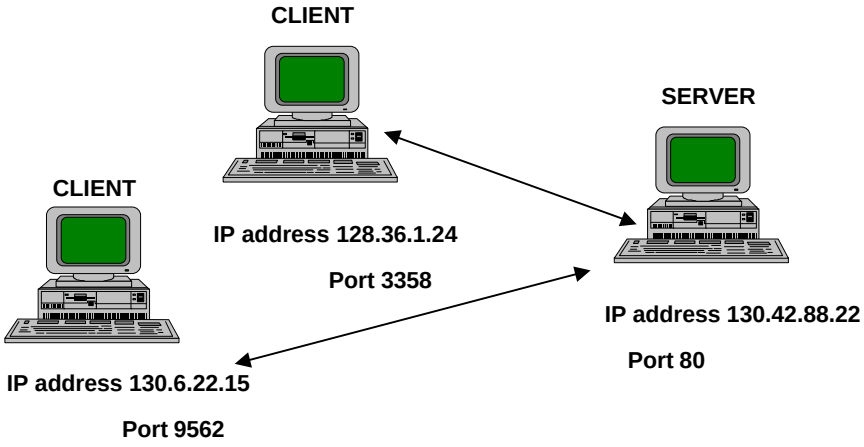


Figura 12: connessioni multiple in TCP

I numeri SN e RN vengono utilizzati per il controllo d'errore e di flusso. Tuttavia l'unità di misura non è il segmento (o pacchetto), come visto nei capitoli precedenti, bensì il byte (il protocollo trasmette flussi di byte). SN indica il numero del primo byte trasmesso mentre RN indica il numero del primo byte da ricevere.

Il campo data offset serve ad indicare la lunghezza dell'header e risulta necessario a causa della presenza del campo opzioni, di lunghezza variabile. Seguono sei bit non utilizzati (reserved) e il campo controllo che contiene 6 bit identificativi di funzioni e di tipo di trama.

Il bit URG è utilizzato per indicare dati urgenti che vengono trasmessi al di fuori del controllo di flusso con un meccanismo che di fatto configura un meccanismo di segnalazione end-to-end tra i processi remoti. Allo scopo viene usato il campo urgent pointer che indica i dati urgenti all'interno del segmento (urgent pointer indica il primo byte dell'informazione urgent).

Il bit ACK è messo a 1 ad indicare che il numero contenuto in RN un valido acknowledgment. Il bit PSH a 1 indica al TCP ricevente di passare immediatamente all'applicazione l'informazione ricevuta. Il bit RST è usato per resettare la connessione mentre i bit SYN e FIN, usati separatamente, indicano trame di inizio e fine connessione rispettivamente.

Il controllo di flusso in TCP non ha finestra fissa ma variabile e la larghezza della finestra deve essere indicata di volta in volta nel campo window della trama usando come unità di misura il MSS.

Il campo checksum viene calcolato su tutta la trama incluso lo pseudoheader ma col campo checksum a zero. Vengono sommate tutte le parole di 16 bit complementate a 1 e la somma viene complementata a 1. In ricezione, gli stessi campi sommati forniscono zero se non vi è stato errore.

Il campo ozioni è usato per fornire caratteristiche non contenute nell'header. La più importante è l'opzione che permette di indicare il MSS durante la fase di instaurazione della connessione. Se questa opzione non è usata il valore di default per il MSS è 536 byte.

4.2.1 Gestione delle connessioni

Nella Figura 13 è esemplificato il meccanismo di apertura di una connessione TCP. Il TCP utilizza il meccanismo di three way handshake, già visto nei capitoli precedenti.

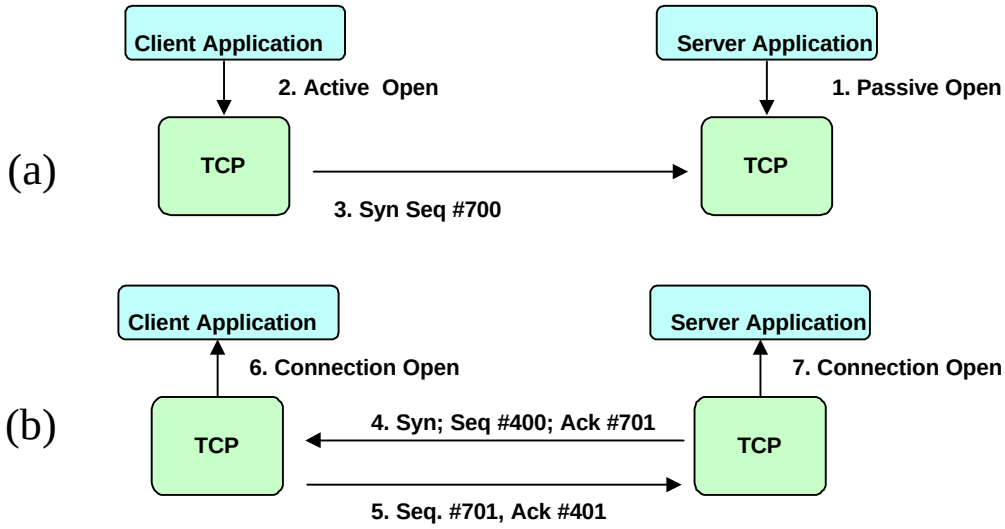


Figura 13: apertura di una connessione TCP

A livello TCP i problemi relativi ad eventuali cadute e riprese di connessione sono molto più critici perchè da un lato i tempi di attraversamento end-to-end sono molto più incerti e dall'altro il TCP non può contare sull'esistenza di un livello superiore per correggere malfunzionamenti. In particolare, molto sentito è il problema dei numeri di sequenza che può portare a malfunzionamenti quando una PDU appartenente a una connessione instaurata successivamente a una caduta ha un numero di sequenza uguale a quello di una PDU della connessione precedente che sia ancora in circolazione. Per questo motivo i numero di sequenza iniziale di una connessione viene scelto in modo casuale o derivandolo dal clock locale dell'host.

Affinché dal lato server di un servizio applicativo vi sia un processo in ascolto su una porta deve avvenire l'attivazione del TCP mediante una primitiva detta di PASSIVE_OPEN. Dal lato client, quando si vuole effettivamente aprire una connessione deve essere passata al TCP locale una primitiva di ACTIVE_OPEN.

La prima trama inviata dal TCP del lato client è una trama di SYN (synchronize) caratterizzata appunto dal bit di SYN posto a 1. Il numero di sequenza contenuto nel campo SN è il numero iniziale scelto dal TCP del

client per la nuova connessione. In risposta il TCP del server invia una trama di SYN-ACK che ha il bit di SYN posto a 1 e il bit di ACK posto a 1. Il campo SN contiene il numero iniziale scelto dal TCP del server per la nuova connessione e il campo RN contiene il riscontro della corretta ricezione della trama precedente (valore di SN rivevuto più 1).

L'apertura della connessione viene completata con l'invio dell'ACK finale contenente il riscontro del TCP del client della corretta ricezione della trama di SYN-ACK del TCP del server.

Anche la chiusura della connessione (Figura 14) è effettuata tramite un meccanismo sicuro che prevede l'invio di una messaggio di FIN (bit di FIN settato a 1) e di un ACK di avvenuta ricezione. Ciò non garantisce la corretta chiusura in ogni caso e il TCP ovvia con dei time out. Se il pacchetto di FIN non è seguito da un ACK, dopo un tempo sufficiente la connessione full duplex viene chiusa. L'altra parte può non accorgersi della chiusura immediatamente ma prima o poi si renderà conto che nessuno risponde e a sua volta chiude.

Normalmente la chiusura avviene in modo indipendente nei due versi, perché anche se una delle due parti ha finito la sua trasmissione, l'altra può ancora avere dati da trasmettere.

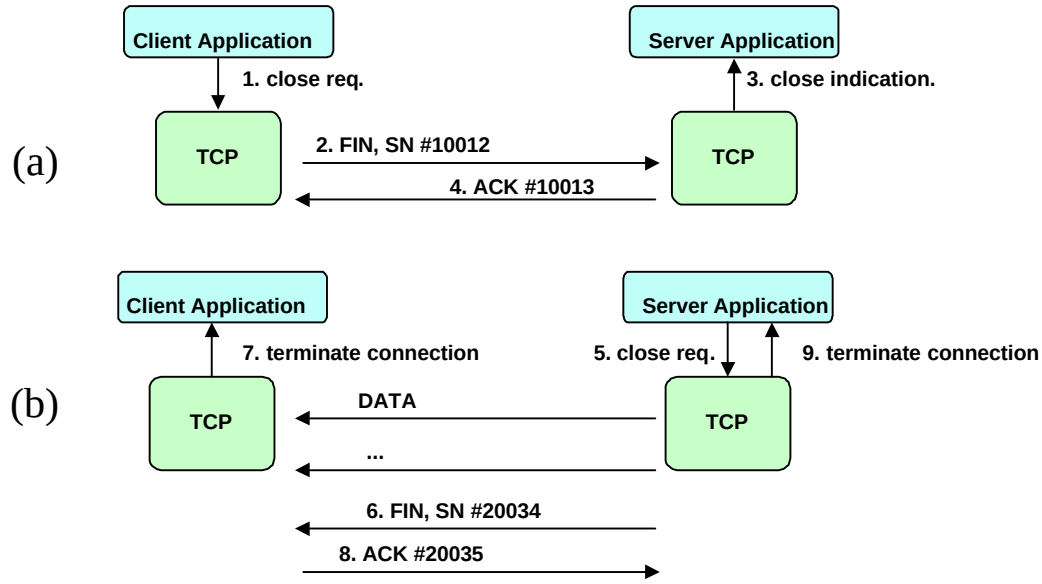


Figura 14: chiusura di una connessione TCP

4.2.2 Il controllo di flusso

Il controllo di flusso adottato nel TCP è basato sulla trasmissione esplicita della finestra di ricezione. L'entità TCP ricevente non trasmette immediatamente i dati che riceve all'utente, ma li immagazzina in un buffer, finché il buffer si riempie o un URG flag viene ricevuto. L'applicazione ricevente assorbe i byte nel buffer di ricezione in base alla sua velocità di elaborazione dell'informazione.

La trasmissione degli ACK da parte del ricevitore è indipendente dal meccanismo di controllo di flusso grazie alla trasmissione esplicita della finestra di ricezione. Il valore contenuto nel campo window indica il numero di MSS ancora ricevibili a partire dal byte indicato in RN.

Nelle prime implementazioni del TCP si era riscontrato un problema, noto come Silly window syndrome, che si riscontrava quando l'applicazione ricevente assorbiva i byte lentamente uno per volta. In questo caso il ricevitore vedeva ogni volta un byte libero nel buffer e con l'invio dell'ACK e del campo window sollecitava la trasmissione di una trama con un solo byte. Naturalmente una quantità di trame con un solo byte portano ad uno spreco di risorse a causa dell'elevatissimo overhead.

Per evitare il problema si sono introdotti meccanismi sia in trasmissione che in ricezione. In ricezione il TCP ricevitore mente indicando un buffer pieno fino a che il buffer non si è svuotato per metà o per una quantità almeno pari a MSS. In trasmissione il TCP trasmettitore tenta di creare segmenti non più piccoli di ½ MSS se non sollecitato con primitive a fare il PUSH dei dati.

4.2.3 Stima del Round Trip Time (RTT)

Il meccanismo di controllo d'errore del TCP è di tipo go-back-n con riscontri solo positivi. Dunque l'inizio di un ciclo di ritrasmissioni avviene sulla base della scadenza di un time-out. Il valore del time-out è un parametro critico per il meccanismo perché il TCP deve essere in grado di funzionare correttamente anche con reti che presentano valori del ritardo di ritorno degli ACK (RTT) anche molto diversi.

Per questo motivo il TCP basa il valore del time-out su un meccanismo di stima del RTT (algoritmo di Karn e Jacobson). Ogni volta che viene ricevuto un ACK valido viene aggiornato il valore stimato di RTT con il nuovo campione:

$$SRTT^{(i)} = (1-a)\, SRTT^{(i-1)} + a\, RTT^{(i)}$$

con a compreso tra 0 ed 1 (tipicamente pari ad 1/8).

Il timeout di ritrasmissione è calcolato sulla base del SRTT, calcolando una stima della deviazione standard di SRTT:

$$DEV = |RTT^{(i)} - SRTT^{(i-1)}|$$

e calcolando un valore mediato (smoothed):

$$SDEV^{(i)} = 3/4\, SDEV^{(i-1)} + 1/4\, DEV$$

Infine, il timeout e' calcolato come

$$TIMEOUT = SRTT + 2\, SDEV$$

All'inizio SRTT viene posto uguale a zero e SDEV = 1.5 s, e quindi il valore del timeout parte a 3 s.

Se scatta un time-out i campioni dei pacchetti ritrasmessi non vengono tenuti in conto a causa delle possibili ambiguità della stima. Si supponga infatti che in un rete RTT passi a 1 s per problemi di traffico mentre il time-out vale 100 ms. Il TCP effettua 10 ritrasmissioni dello stesso pacchetto e quando finalmente riceve un ACK (quello del primo pacchetto arrivato dopo 1 s) stima erroneamente un RTT molto basso.

Dopo ogni ritrasmissione il time-out viene normalmente moltiplicato per un fattore costante (tipicamente 2) fino a che non arriva ad un valore massimo. Dopo un numero massimo di ritrasmissioni la connessione viene resettata.

4.2.4 Il controllo anticongestione

Il meccanismo anticongestione è di tipo a finestra con rete passiva. E', dunque, il TCP stesso a "stimare" se esiste congestione o meno in rete e in tal caso a decidere di rallentare il ritmo di trasmissione.

Il meccanismo anticongestione utilizza una seconda finestra, detta di congestione CW (Congestion Window), oltre a quella di ricezione RW (Receive Window) e i dati che possono essere trasmessi dal trasmettitore devono essere contenuti in entrambe le finestre, ovvero in quella che esprime il vincolo più stringente. Quindi la RW è legata in modo diretto al meccanismo di controllo di flusso, mentre la CW al meccanismo di controllo di congestione.

Il trasmettitore TCP può passare dinamicamente dalla fase di *Slow Start* a quella di *Congestion Avoidance*, e vice versa. La variabile SSTHRESH e' mantenuta al trasmettitore per distinguere le due fasi.

All'inizio, il trasmettitore fa partire la trasmissione in Slow Start inviando un segmento (per esempio 512 byte), e CW è posta a 1 segmento. Quando il trasmettitore riceve l'ACK del segmento, la finestra di

congestione viene incrementata di 1 segmento. In tal modo la finestra di congestione raddoppia ad ogni RTT (incremento esponenziale) se si assume che gli ACK vengano trasmessi immediatamente (Figura 15).

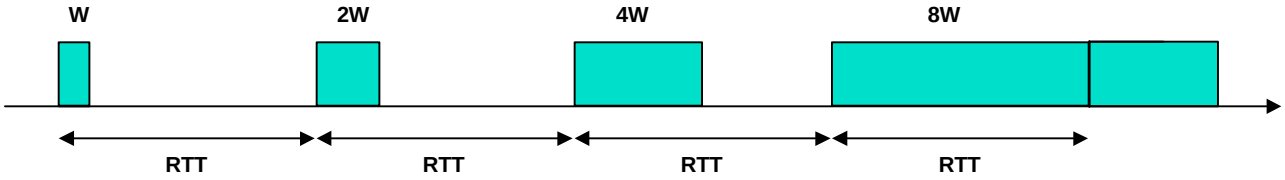


Figura 15: evoluzione della finestra nella fase di Slow Start

Lo slow start continua fino a che la CW diventa grande come SSTHRESH (valore tipico 64 Kbyte) e poi parte la fase di congestion avoidance, durante la quale ad ogni ACK si aumenta CW di 1/CW, dando origine ad una crescita circa lineare di CW.

Se scatta un timeout, il TCP trasmittente reagisce ponendo SSTHRESH uguale alla metà del numero di byte trasmessi e non riscontrati e CW è posta a 1. Ciò si traduce normalmente nel porre:

SSTHRESH = min(CW/2,RW)

Come risultato, CW risulta minore di SSTHRESH e si entra nella fase di Slow Start; il trasmettitore invia il segmento e la sua CW è incrementata di 1 ad ogni ACK. Naturalmente, il trasmettitore trasmette tutti i segmenti a partire da quello per cui il timeout è fallito (politica go-back-N).

In Figura 16 è riportato a titolo esemplificativo una possibile evoluzione della finestra di congestione di una connessione TCP.

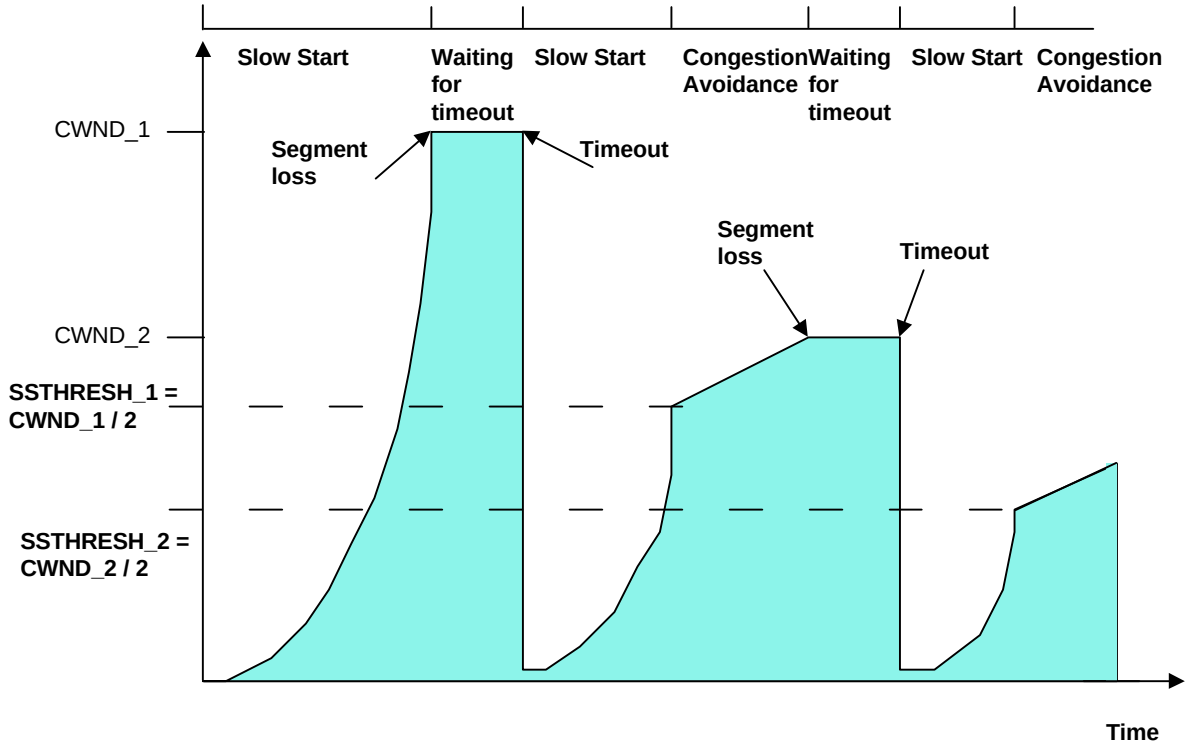


Figura 16: esempio di evoluzione della finestra di congestione di una connessione TCP