RETI DI CALCOLATORI

A.A. 2017/2018

Enrico Martini

COS'E' INTERNET?

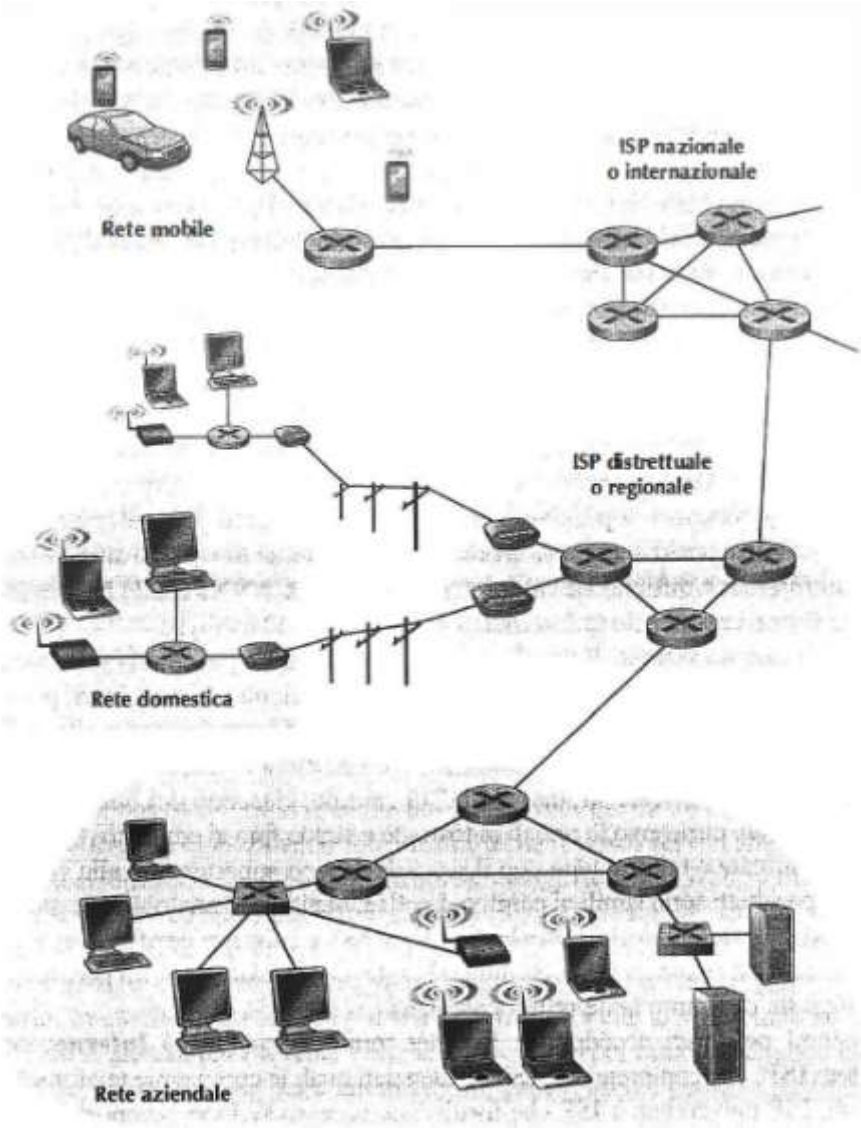
Internet è un insieme di infrastrutture di rete che forniscono servizi ad applicazioni distribuite.

ROUTER: apparecchio che in italiano si tradurrebbe letteralmente con "colui che mette in strada", decide il percorso che l'informazione deve fare.

HOST FINALE: qualsiasi apparecchio che riceve/invia informazioni, ad esempio telefoni, pc, stampanti, server.

INTERNET SERVICE PROVIDER (ISP): operatori che si collegano alla rete e permettono gli scambi di informazione, come Vodafone e TIM.

PROTOCOLLO: Un protocollo definisce il formato e l 'ordine dei messaggi scambiati tra due o più entità in comunicazione, così come le azioni intraprese in fase di trasmissione e/o di ricezione di un messaggio o di un altro evento.



SUDDIVISIONE DELLA RETE

La rete può essere divisa in:

- Rete di accesso
- Nucleo della rete

RETE DI ACCESSO

Rete che connette fisicamente un sistema al suo edge router (router di bordo), che è il primo router sul percorso dal sistema d'origine a un qualsiasi altro sistema di destinazione collocato al di fuori della stessa rete di accesso.

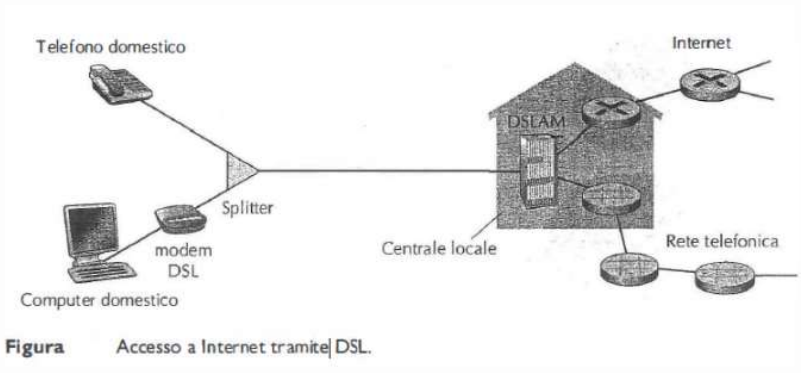


Figura Accesso a Internet tramite DSL.

NUCLEO DELLA RETE

Maglia di commutatori di pacchetti e collegamenti che interconnettono i sistemi periferici di Internet.

COMMUTATORI DI CIRCUITO

Nelle reti a commutazione cli circuito le risorse richieste lungo un percorso per consentire la comunicazione tra sistemi periferici sono riservate per l'intera durata della sessione di comunicazione. Le reti telefoniche sono esempi di reti a commutazione di circuito.

Un circuito all'interno di un collegamento è implementato tramite multiplexing a divisione di frequenza (FDM, frequency-division multiplexing) o multiplexing a divisione di tempo (TDM, time-division multiplexing). Nel FDM si decidono le vare frequenze e si suddividono per chi sta chiedendo il servizio, come ad esempio la radio. Nel TDM si suddivide il tempo in quanti e ogni utente ha una determinata unità di tempo.

COMMUTATORI DI PACCHETTO

Per eseguire i propri compiti le applicazioni distribuite scambiano messaggi. La sorgente suddivide i messaggi lunghi in parti più note come pacchetti. Tra la sorgente e la destinazione, questi pacchetti viaggiano attraverso collegamenti e commutatori di pacchetto. I pacchetti vengono trasmessi su ciascun collegamento a una velocità pari alla velocità totale di trasmissione del collegamento stesso. Non viene perciò fatta nessuna richiesta e il ritardo è dipendente dal numero di apparati che deve attraversare il pacchetto.

Le risorse vengono utilizzate all'occorrenza e divise tra chi ne richiede l'utilizzo. Il sistema aumenta il ritardo aumentando le stazioni in cui si deve passare. Generalmente non si notano rallentamenti in quanto le persone che accedono ad internet lo fanno in maniera discontinua. La commutazione a pacchetto è più efficiente perciò della commutazione a circuito per internet e meno efficiente per quanto riguarda lo streaming audio/video.

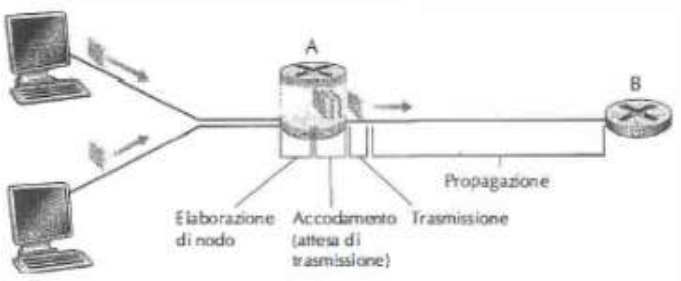
Come analogia, si considerino due ristoranti: uno che richiede la prenotazione e l'altro che non la richiede né l'accetta. Nel primo caso, abbiamo l'incombenza di dover telefonare, ma quando arriviamo, in linea di principio, possiamo immediatamente accedere al nostro tavolo e ordinare la cena. Nel secondo non dobbiamo prenotare, per cui quando arriviamo al ristorante potremmo dover attendere che si liberi un tavolo prima di poterci accomodare.

RITARDI E PERDTE

Le reti di calcolatori limitano necessariamente il **throughput**, cioè la quantità di dati al secondo che può essere trasferita tra due sistemi periferici, introducono ritardi tra questi ultimi e possono addirittura perdere pacchetti.

Esistono 4 tipi di ritardi:

1) Ritardo di elaborazione
2) Ritardo di accodamento
3) Ritardo di trasmissione
4) Ritardo di propagazione



Ritardo di nodo al router A.

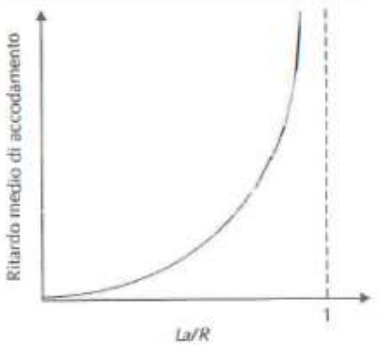
$$d_{node} = d_{process} + d_{queque} + d_{transmission} + d_{propagation}$$

RITARDO DI ELABORAZIONE

Tempo richiesto per esaminare l'intestazione del pacchetto e per determinare dove dirigerlo, comprendendo anche il tempo per controllare eventuali errori a livello di bit occorsi nel pacchetto. Nei router ad alta velocità questi ritardi sono solitamente dell'ordine dei microsecondi o inferiori.

RITARDO DI ACCODAMENTO

Tempo che subisce il pacchetto mentre aspetta di essere trasmesso, dipende dall'intensità del traffico in arrivo. Nella pratica i ritardi di accodamento possono essere dell'ordine dei microsecondi o dei millisecondi.



Ritardo medio di accodamento in funzione dell'intensità di traffico.

RITARDO DI TRASMISSIONE

Tempo richiesto per trasmettere tutti i bit del pacchetto sul collegamento. Sia L la lunghezza del pacchetto, in bit, e R bps la velocità di trasmissione del collegamento dal router A al router B. Il ritardo di trasmissione (transmission delay) risulta essere L/R. Anche i ritardi di trasmissione sono di solito dell'ordine dei microsecondi o dei millisecondi.

RITARDO DI PROPAGAZIONE

Tempo che impiega il pacchetto per arrivare al router B una volta immesso sul collegamento. Il bit viaggia alla velocità di propagazione del collegamento, che dipende dal mezzo fisico. Nelle reti molto estese i ritardi di propagazione sono dell'ordine dei millisecondi.

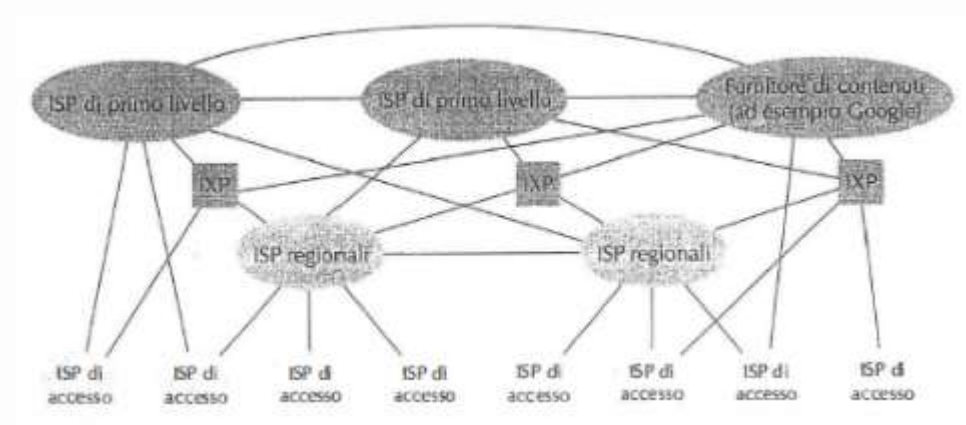
PERDITA DI PACCHETTI

Nel caso la linea sia bloccata dal traffico dati, la coda continua ad aumentare fino ad arrivare al limite fisico del router e un pacchetto può trovare la coda piena. Non essendo possibile memorizzare tale pacchetto, il router lo eliminerà e il pacchetto andrà perduto.

THROUGHPUT

Oltre al ritardo e alla perdita di pacchetti, un'altra misura critica delle prestazioni in una rete di calcolatori è il throughput end-to-end. È la frequenza alla quale i bit sono trasferiti tra mittente e destinatario. Nel trasferimento si può verificare il **collo di bottiglia**, cioè un punto della trasmissione in cui la velocità è minore. In genere il throughput dei server è molto superiore a quello dei client, per evitare rallentamenti.

SUDDIVISIONE DELLA RETE



L' ISP può fornire connettività attraverso una rete cablata o senza fili con svariate tecnologie quali DSL, cavo, FITH, Wi-Fi e cellulare. Si dividono in livelli, a seconda di quanto sono estesi. Per avere una connessione Internet globale gli ISP clienti pagano i loro ISP fornitori. Il costo riflette la quantità di traffico che l'ISP cliente scambia con il fornitore. Per ridurre tali costi, una coppia di ISP vicini e di pari livello gerarchico può fare uso di peering, cioè connettere direttamente le loro reti in modo che tutto il traffico tra di esse passi attraverso una connessione diretta piuttosto che transitare da un intermediario. In questa modalità nessun ISP effettua pagamenti all' altro. Oggigiorno Internet, una rete di reti, è complessa e consiste di dozzine di ISP di primo livello e centinaia di migliaia di ISP di livello inferiore. Gli ISP si distinguono per la copertura geografica: alcuni di essi si estendono per continenti e oceani mentre altri si limitano a ristrette regioni. Gli ISP di livello più basso si collegano a quelli di livello superiore e questi ultimi si interconnettono

tra loro. Gli utenti e i fornitori di contenuto sono clienti degli ISP di livello inferiore, mentre questi ultimi sono a loro volta clienti degli ISP di livello superiore.

ISP LIVELLO 1: provider di livello nazionale/internazionale. Esistono circa una dozzina di ISP di primo livello tra i quali troviamo Level 3 Communications, AT &T, Sprint e NTT. Anche gli ISP di primo livello fanno peering tra di loro a costo zero.

ISP LIVELLO 2: provider più piccoli, si agganciano agli ISP di livello 1, Ma possono anche essere connessi tra loro alla pari.

ISP REGIONALE: in ogni regione può esservi un ISP regionale al quale tutti gli ISP di accesso della regione si connettono.

MODELLO A STRATI

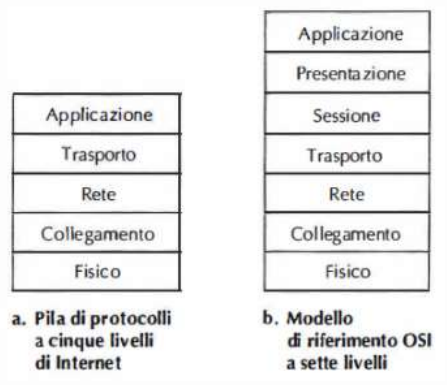
Un' architettura a livelli consente di discutere una parte specifica e ben definita di un sistema articolato e complesso. Questa stessa semplificazione ha un valore considerevole grazie all'introduzione della modularità, che rende molto più facile cambiare l'implementazione del servizio fornito da un determinato livello. Fino a quando il livello fornisce lo stesso servizio allo strato superiore e utilizza gli stessi servizi dello strato inferiore, la parte rimanente del sistema rimane invariata al variare dell'implementazione del livello. Ogni livello fornisce il suo servizio effettuando determinate azioni all'interno del livello stesso e utilizzando i servizi del livello immediatamente inferiore. Un livello di protocolli può essere implementato via software, hardware o con una combinazione dei due. La modularità rende più facile aggiornare la componentistica.

Tra livelli adiacenti si comunica attraverso: REQUEST (richiesta servizio), INDICATION (indicazione evento), RESPONSE (risposta all'indicazione), CONFIRM (conferma richiesta).

Esistono due categorie di comunicazione:

- CONNECTION ORIENTED = scambio di informazioni che garantiscono la consegna sequenziale (fase iniziale il cui scopo serve a verificare la possibilità di scambio);
- CONNECTIONLESS = semplice passaggio d'informazione;

Considerati assieme, i protocolli dei vari livelli sono detti pila di protocolli (protocol stack).



a. Pila di protocolli a cinque livelli di Internet

b. Modello di riferimento OSI a sette livelli

STACK OSI

Divisione in 7 livelli generali. I router ne hanno solamente 3 poiché devono solo indirizzare al punto giusto l'informazione e non interessa il contenuto. Il protocollo di rete è unico perché è quello che rende possibile la connessione end-to-end.

LIVELLO DI APPLICAZIONE

Per lo sviluppatore l'architettura di rete è fissata e fornisce alle applicazioni uno specifico insieme di servizi; il suo compito è progettare **l'architettura dell'applicazione** e stabilire la sua organizzazione sui vari sistemi periferici.

ARCHITETTURA CLIENT-SERVER

Nell'architettura client-server vi è un host sempre attivo, chiamato server, che risponde alle richieste di servizio di molti altri host, detti client.

SERVER → host sempre attivo

IP fisso

Riuniti in **server farm**, che unisce molti host

CLIENT → comunica in qualsiasi momento con i server

IP dinamico

ARCHITETTURA P2P

In un'architettura P2P l'infrastruttura di server in data center è minima o del tutto assente; si sfrutta, invece, la comunicazione diretta tra coppie arbitrarie di host, chiamati peer (ossia pari), collegati in modo intermittente. Ogni partecipante quindi è sia client che server, chiede informazioni e contemporaneamente ne mette a disposizione. Tutto questo però è difficile da gestire, in quanto i peer non sono sempre attivi come i server.

In entrambe le architetture, i processi su due sistemi terminali comunicano scambiandosi messaggi attraverso la rete: il processo mittente crea e invia messaggi nella rete e il processo destinatario li riceve e, quando previsto, invia messaggi di risposta. Nel Web un processo browser avvia il contatto con un processo web server; quindi il primo è il client e il secondo il server.

SOCKET

Un processo invia messaggi nella rete e riceve messaggi dalla rete attraverso un'interfaccia software detta socket. In italiano “presa” , è il punto di accesso per invio/ricezione dei messaggi. Per indirizzare correttamente la socket abbiamo bisogno di:

- Indirizzo IP, che identifica la macchina;
- Numero di porta, che identifica il processo.

Una socket è l'interfaccia tra il livello di applicazione e il livello di trasporto all' interno di un host.

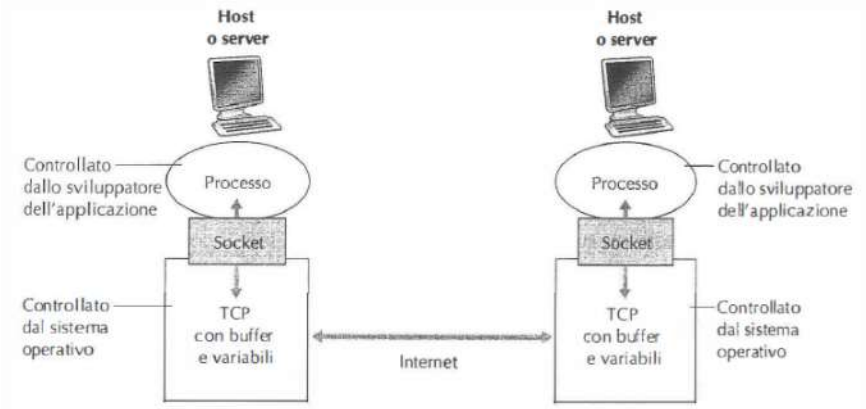


Figura 2.3 Processi, socket e protocollo di trasporto sottostante.

SERVIZI DI TRASPORTO DISPONIBILI

Affidabilità

Se un protocollo fornisce questo tipo di servizio di consegna garantita dei dati, si dice che fornisce un **trasferimento dati affidabile** (*reliable data transfer*). Quando un protocollo a livello di trasporto fornisce tale servizio, il processo mittente può passare i propri dati alla socket e sapere con assoluta certezza che quei dati arriveranno senza errori al processo ricevente. Quando un protocollo a livello di trasporto non fornisce trasferimento dati affidabile, i dati inviati dal processo mittente potrebbero non arrivare mai a quello ricevente. Ciò potrebbe essere accettabile per le **applicazioni che tollerano le perdite** (*losstolerant*).

Throughput

Consiste nella velocità di consegna delle informazioni. Le applicazioni che hanno requisiti di throughput vengono dette **applicazioni sensibili alla banda** (*bandwidth-sensible*). Mentre le applicazioni sensibili alla banda hanno requisiti specifici di throughput, le applicazioni elastiche possono far uso di tanto o di poco throughput a seconda di quanto ce ne sia disponibile. La posta elettronica, il trasferimento di file e il Web sono tutte applicazioni elastiche.

Temporizzazione

Un protocollo a livello di trasporto può anche fornire garanzie di temporizzazione (*timing*) che, come quella per il throughput, possono assumere varie forme. Per esempio, la garanzia potrebbe essere che ogni bit che il mittente invia sulla socket venga ricevuto dalla socket di destinazione non più di 100 millisecondi più tardi. Per le applicazioni non in tempo reale, ritardi inferiori sono sempre preferibili a ritardi più consistenti, ma non si pongono stretti vincoli sui ritardi end-to-end.

Sicurezza

Nell'host mittente, un protocollo di trasporto può cifrare tutti i dati trasmessi dal processo mittente e, nell'host di destinazione, il protocollo di trasporto può decifrare i dati prima di consegnarli al processo ricevente. Un protocollo a livello di trasporto può fornire altri servizi di sicurezza oltre alla riservatezza, compresi l'integrità dei dati e l'autenticazione end-to-end.

SERVIZIO TCP

TCP prevede un servizio orientato alla connessione (*connection oriented*) e il trasporto affidabile dei dati. Esiste inoltre un meccanismo di controllo della congestione, che esegue una "strozzatura" del processo d'invio (client o server) quando il traffico in rete appare eccessivo.

- CONNECTION ORIENTED

  TCP fa in modo che client e server si scambino informazioni di controllo a livello di trasporto prima che i messaggi a livello di applicazione comincino a fluire. Questa procedura, detta di **handshaking**, mette in allerta client e server, preparandoli alla partenza dei pacchetti. Dopo la fase di handshaking, si dice che esiste una **connessione TCP tra le socket** dei due processi. Tale connessione è full-duplex, nel senso che i due processi possono scambiarsi contemporaneamente messaggi sulla connessione.

- TRASPORTO AFFIDABILE

  I processi comunicanti possono contare su TCP per trasportare i dati senza errori e nel giusto ordine.

SERVIZIO UDP

UDP è un protocollo di trasporto **leggero** e senza fronzoli, dotato di un modello di servizio **minimalista**. È **senza connessione**, non necessita quindi di handshaking, fornisce un servizio di trasferimento dati **non affidabile**. Inoltre i messaggi potrebbero giungere a destinazione non in ordine.

Applicazione	Protocollo a livello di applicazione	Protocollo di trasporto sottostante
Posta elettronica	SMTP [RFC 5321]	TCP
Accesso a terminali remoti	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Trasferimento file	FTP [RFC 959]	TCP
Multimedia streaming	HTTP (per esempio: YouTube)	TCP
Telefonia Internet	SIP [RFC 3261], RTP [RFC 3550], o proprietario (per esempio: Skype)	UDP o TCP

WEB E HTTP

Una pagina web (*web page*), detta anche documento, è costituita da oggetti. Un oggetto è semplicemente un file (quale un file HTML, un'immagine JPEG, un applet Java, una clip video e così via) indirizzabile tramite un **URL**. La maggioranza delle pagine web consiste di un file HTML principale e diversi oggetti referenziati da esso.

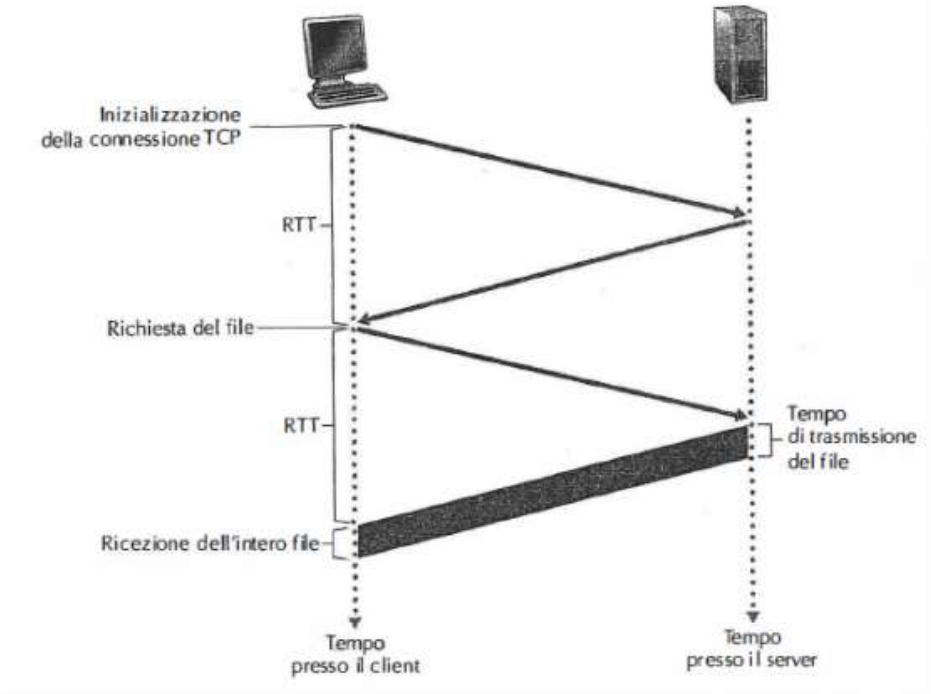
HTTP (*hypertext transfer protocol*), protocollo a livello di applicazione del Web, costituisce il cuore del Web. Questo protocollo è implementato in due programmi, client e server, in

esecuzione su sistemi periferici diversi che comunicano tra loro scambiandosi messaggi HTTP. Un **browser web** (come Internet Explorer o Firefox) implementa il lato client di HTTP. Un **web server**, che implementa il lato server di HITP, ospita oggetti web, indirizzabili tramite URL. HITP utilizza TCP (anziché UDP) come protocollo di trasporto. Dato che i server HTTP non mantengono informazioni sui client, HTTP è classificato come **protocollo senza memoria di stato** (*stateless protocol*).

HTTP CON CONNESSIONI NON PERSISTENTI

Ogni connessione TCP viene chiusa dopo l'invio dell'oggetto da parte del server: vale a dire che ciascuna trasporta soltanto un messaggio di richiesta e un messaggio di risposta.

Bisogna definire il *round-trip-time* (**RTT**), che rappresenta il tempo impiegato da un piccolo pacchetto per viaggiare dal client al server e poi tornare al client. Il RTT include i ritardi di propagazione, di accodamento nei router e nei commutatori intermedi nonché di elaborazione del pacchetto. Il tempo di risposta totale è, approssimativamente, di due RTT, più il tempo di trasmissione da parte del server del file HTML.



Le connessioni non persistenti presentano alcuni limiti: il primo è che per ogni oggetto richiesto occorre stabilire e mantenere una nuova connessione. In secondo luogo, ciascun oggetto subisce un ritardo di consegna di due RTT, uno per stabilire la connessione TCP e uno per richiedere e ricevere un oggetto.

HTTP CON CONNESSIONI PERSISTENTI

Nelle connessioni persistenti il server lascia la connessione TCP aperta dopo l'invio di una risposta, per cui le richieste e le risposte successive tra gli stessi client e server possono essere trasmesse sulla stessa connessione. In generale, il server HTTP chiude la connessione quando essa rimane inattiva per un dato lasso di tempo.

MESSAGGIO DI RICHIESTA HTTP

Esempio:

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool .edu
Connection: close
User-agent: Mozilla/5 .0
Accept -language: fr
```

Consiste di cinque righe, ciascuna seguita da un carattere di ritorno a capo (*carriage return*). La prima riga è detta riga di richiesta (*request line*) e quelle successive righe di intestazione (*header lines*). La riga di richiesta presenta tre campi: il campo metodo, il campo URL e il campo versione di HTTP. Il campo metodo può assumere diversi valori, tra cui **GET**, POST, HEAD, PUT e DELETE. La riga host specifica l'host su cui risiede l'oggetto. Includendo la linea di intestazione *Connection: close*, il browser sta comunicando al server che non si deve occupare di connessioni persistenti, ma vuole che questi chiuda la connessione dopo aver inviato l'oggetto richiesto. La riga di intestazione User-agent: specifica il tipo di browser che sta effettuando la richiesta al server, in questo caso Mozilla/5.0, un browser Firefox. Infine, Accept-language: indica che l'utente preferisce ricevere una versione in francese dell'oggetto se disponibile.

Il metodo **HEAD** è simile a GET, ma tralascia gli oggetti richiesti.

Se il valore del campo metodo è **POST**, allora il corpo contiene ciò che l'utente ha immesso nei campi del form.

Il metodo **PUT**, frequentemente usato assieme agli strumenti di pubblicazione sul Web, consente agli utenti di inviare un oggetto a un percorso specifico (directory) su uno specifico web.

Il metodo **DELETE** consente invece la cancellazione di un oggetto su un server.

MESSAGGIO DI RISPOSTA HTTP

Esempio:

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
(data data data data data . . . )
```

Analizzando in dettaglio questo messaggio di risposta, osserviamo tre sezioni: una riga di stato iniziale, sei righe di intestazione e il corpo. Il server utilizza la riga di intestazione *Connection: close* per comunicare al client che ha intenzione di chiudere la connessione TCP dopo l ' invio del messaggio.

La riga *Date:* indica l'ora e la data di creazione e invio, da parte del server, della risposta HTTP.

La riga *Server*: indica che il messaggio è stato generato da un web server Apache.

La riga *Last-Modified*: indica l'istante e la data il cui oggetto è stato creato o modificato per l 'ultima volta.

La riga di intestazione *Content-Length:* contiene il numero di byte dell' oggetto inviato.

La riga *Content-Type:* indica che l ' oggetto nel corpo è testo HTML.

Il tipo dell'oggetto viene ufficialmente identificato tramite l 'intestazione *Content-Type:* , non tramite l ' estensione del file.

I codici di risposta presenti alla prima riga possono essere:

- 200 → OK
- 301 → Oggetto trasferito permanentemente
- 304 → Oggetto non modificato
- 400 → Messaggio non compreso
- 404 → Pagina non trovata
- 505 → Versione HTTP non supportata

COOKIE

I cookie consentono ai server di tener traccia degli utenti, per fornire contenuti in funzione della loro identità. La tecnologia dei cookie presenta quattro componenti:

1) Una riga di intestazione nel messaggio di risposta HTTP;
2) Una riga di intestazione nel messaggio di richiesta HTTP;
3) Un file mantenuto sul sistema del1'utente e gestito dal browser;
4) Un database sul sito.

Nonostante i cookie semplifichino lo shopping via Internet, sono fonte di controversie, in quanto possono essere considerati una violazione della privacy dell'utente.

WEB CACHING

Una web cache, nota anche come **proxy server**, è un'entità di rete che soddisfa richieste HTTP al posto del web server effettivo. La cache è uno spazio in memoria nel proprio browser riservato per caricare velocemente gli elementi fissi dei siti più frequentati dall'utente. Il proxy è contemporaneamente server e client: quando riceve richieste da un browser e gli invia risposte agisce da server, quando invia richieste e riceve risposte da un server di origine funziona da client. Generalmente un proxy server è acquistato e installato da un ISP. Se esiste una connessione ad alta velocità tra il client e il proxy e se l ' oggetto è nella cache, allora questa sarà in grado di trasportare l ' oggetto rapidamente al client. Inoltre i proxy possono ridurre sostanzialmente il traffico sul collegamento di accesso a

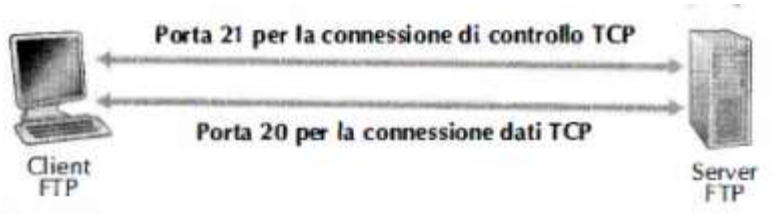
Internet, con il vantaggio di non dover aumentare l'ampiezza di banda frequentemente e ottenere quindi una riduzione dei costi.

Il web caching introduce un nuovo problema: la copia di un oggetto che risiede in cache potrebbe essere scaduta. HTTP presenta un meccanismo che permette alla cache di verificare se i suoi oggetti sono aggiornati. Questo meccanismo è chiamato GET condizionale. Un messaggio di richiesta HTTP viene detto messaggio di GET condizionale se usa il metodo GET e include una riga di intestazione If-modified-since:.

FILE TRANSFERT PROTOCOL (FTP)

In una tipica sessione FTP, l'utente utilizza un host (locale) per trasferire file da o verso un host remoto. L'utente fornisce il nome dell'host remoto, in modo che il processo FTP client nell' host locale stabilisca una connessione TCP con il processo FTP server nell'host remoto e fornisce nome identificativo e password, inviate sulla connessione TCP come parte dei comandi FTP.

Tra le più rilevanti differenze che distinguono questi due protocolli a livello di applicazione, notiamo che, per trasferire i file, FTP utilizza due connessioni TCP parallele, dette **connessione di controllo** (*control connection*) e **connessione dati** (*data connection*). Dato che FTP usa una connessione di controllo separata, si dice che tale protocollo invii le proprie informazioni di controllo **fuori banda**.



la connessione di controllo rimane aperta per l 'intera durata della sessione utente, ma si crea una nuova connessione dati per ogni file trasferito all' interno della sessione.

Codici di invio comuni

- `USER username`: utilizzato per inviare l'identificativo dell'utente al server.
- `PASS password`: usato per inviare la password dell'utente al server.
- `LIST`: utilizzato per chiedere al server di inviare un elenco di tutti i file nella directory remota corrente. Tale elenco viene spedito su una connessione dati (nuova e non persistente) anziché sulla connessione di controllo.
- `RETR filename`: usato per recuperare un file dall'attuale directory dell'host remoto. Forza l'host remoto a inizializzare una connessione dati e a inviare il file richiesto.
- `STOR filename`: utilizzato per memorizzare un file nella directory corrente dell'host remoto.

Codici di ritorno comuni

Oltre al codice "331 Username OK, password required: il nome dell'utente è stato ricevuto, ora occorre inviare la password", esistono anche:

- `125 Data connection already open; transfer starting`: la connessione dati è aperta, può iniziare il trasferimento.
- `425 Can't open data connection`: non è possibile aprire la connessione dati.
- `452 Error writing file`: si è verificato un errore nella scrittura del file.

LA POSTA ELETTRONICA

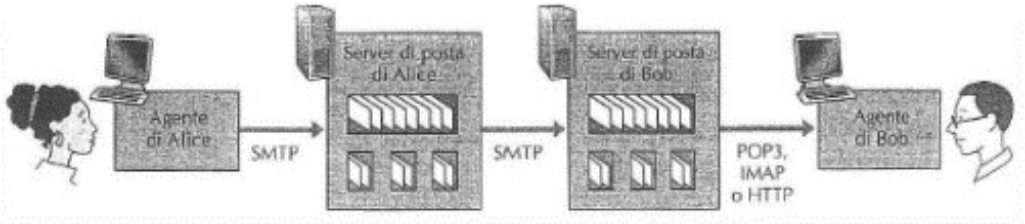
Servizio legato a vari componenti, che uniscono persone con ISP differenti. L'utente invia messaggi al proprio server di posta elettronica, che poi li inoltrerà al server destinatario tramite il protocollo SMTP.

SMTP *(simple mail transfer protocol)*

Vecchio protocollo che trasferisce i messaggi dal mail server del mittente a quello del destinatario. Essendo un protocollo arcaico, tratta il corpo (non solo le intestazioni) di tutti i messaggi di posta come semplice ASCII a 7 bit. Questo è piuttosto penalizzante, in quanto richiede che i dati multimediali binari vengano codificati in ASCII prima di essere inviati e che il messaggio venga nuovamente decodificato in binario dopo il trasporto. Usa il protocollo TCP per trasferire in modo affidabile i messaggi di posta elettronica dal client al server, alla porta 25. Consta in 3 fasi per il trasferimento, usando una connessione persistente:

- Handshaking;
- Trasferimento messaggi;
- Chiusura.

Il trasferimento viene completato introducendo uno speciale protocollo di accesso alla posta, che trasferisce i messaggi dal mail server al PC locale.



POP3 (*post office protocol v.3*)

POP3 è un protocollo di accesso alla posta estremamente semplice, breve e di agevole lettura, ma le sue funzionalità sono piuttosto limitate. Quando la connessione TCP è stabilita, POP3 procede in tre fasi: autorizzazione, transazione e aggiornamento.

Fase di autorizzazione
- Comandi del client:
  - **user:** dichiara il nome dell'utente
  - **pass:** password
- Risposte del server
  - **+OK**
  - **-ERR**

Fase di transazione, client:
- **list:** elenca i numeri dei messaggi
- **retr:** ottiene i messaggi in base al numero
- **dele:** cancella
- **quit**

```
S: +OK POP3 server ready
C: user rob
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

C'è da ricordare inoltre che il protocollo POP3 non mantiene lo stato tra le diverse sessioni.

IMAP (*Internet mail access protocol*)

Anche IMAP è un protocollo di accesso alla posta, ma presenta maggiori potenzialità rispetto a POP3 ed è quindi assai più complesso. I server IMAP conservano informazioni di stato sull'utente da una sessione all' altra: per esempio, i nomi delle cartelle e l'associazione tra i messaggi e le cartelle. Un' altra caratteristica importante di IMAP è la presenza di comandi che permettono agli user agent di ottenere singole parti dei messaggi.

DNS (Domain Name Service)

Esistono due modi per identificare gli host: il nome e l'indirizzo IP. Le persone preferiscono il primo, mentre i router prediligono gli indirizzi IP a lunghezza fissa e strutturati in modo gerarchico. Il compito del DNS è quello di tradurre i nomi degli host nei loro rispettivi indirizzi IP. DNS è contemporaneamente:

- Un database distribuito implementato in una gerarchia di DNS server;
- Un protocollo a livello di applicazione che consente agli host di interrogare il database.

I DNS server sono generalmente macchine UNIX che eseguono un software chiamato BIND. Il protocollo DNS utilizza UDP e la porta 53. Oltre alla traduzione degli hostname in indirizzi IP, DNS mette a disposizione altri importanti servizi.

HOST ALIASING

Un host dal nome complicato può avere uno o più sinonimi (alias). Il DNS può essere invocato da un'applicazione per ottenere l'hostname canonico di un sinonimo, così come l'indirizzo IP dell'host.

MAIL SERVER ALIASING

Un' applicazione di posta può invocare il DNS per ottenere il nome canonico di un sinonimo fornito, così come l 'indirizzo IP dell'host.

LOAD DISTRIBUTION

Il DNS viene anche utilizzato per distribuire il carico tra server replicati, per esempio dei web server. Nel caso di web server replicati, va dunque associato a ogni hostname canonico un insieme di indirizzi IP. Dato che generalmente un client invia il suo messaggio di richiesta HTTP al primo indirizzo IP elencato nell' insieme, la rotazione DNS distribuisce il traffico sui server replicati.

DATABASE DISTRIBUITO E GERARCHICO

Sarebbe impossibile pensare ad un unico Database centralizzato perché ci sarebbero molti problemi, tra cui:

- Un singolo punto di guasto;
- Volume di traffico ingestibile;
- Database troppo distante;
- Manutenzione elevata;

Per trattare il problema della scalabilità, il DNS utilizza un grande numero di server, organizzati in maniera gerarchica e distribuiti nel mondo. Esistono 4 classi di DNS server: i **root server**, i **top-level domain server**, i **server autoritativi** e i **server DNS locali**.

ROOT SERVER

In Internet esistono 13 root server (etichettati da A a M), la maggior parte dei quali sono situati in Nord America. Sebbene abbiamo fatto riferimento a ciascuno dei 13 root server come se si trattasse di un server singolo, ai fini di sicurezza e affidabilità ciascuno di essi è in realtà un cluster di server replicati.

TOP-LEVEL DOMAIN SERVER

Questi server si occupano dei domini di primo livello quali com, org, net, edu e gov, e di tutti i domini di primo livello relativi ai vari paesi, come uk, fr, ca e jp.

SERVER AUTORITATIVI

Ogni organizzazione dotata di host pubblicamente accessibili tramite Internet (quali web server e mail server) deve fornire record DNS pubblicamente accessibili che associno i nomi di tali host a indirizzi IP.

SERVER DNS LOCALI

Ciascun ISP, come un'università, un dipartimento universitario, un'azienda o un ISP residenziale, ha un DNS server locale (*default name server*). Quando un host effettua una richiesta DNS, la query viene inviata al suo server DNS locale, che opera da proxy e inoltra la query in una gerarchia di server DNS.

DNS CACHING

In una concatenazione di richieste, il DNS server che riceve una risposta DNS (contenente per esempio, la traduzione da hostname a indirizzo IP), può mettere in cache le informazioni contenute.

RECORD DNS

I server che implementano i l database distribuito d i DNS memorizzano i cosiddetti record di risorsa (RR, *resource record*), tra cui quelli che forniscono le corrispondenze tra nomi e indirizzi.

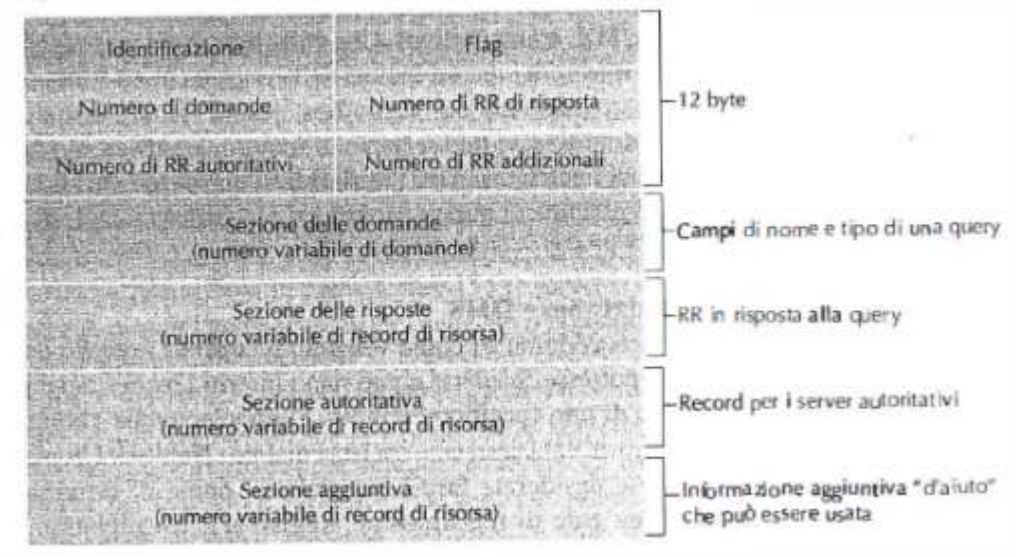
Un record di risorsa contiene i seguenti campi: (Name, Value, Type, TTL)

TTL è il *time to live*, ossia il tempo residuo di vita di un record e determina quando una risorsa vada rimossa dalla cache. Il significato di Name e Value dipende da Type:

- Type = A: name è il nome dell'host, value indirizzo IP;
- Type = NS: name è il dominio, value è il nome dell'host;
- Type = CNAME: name è il nome alias, value è il nome canonico;
- Type = MX: value è il nome del server di posta associato a name.

MESSAGGI DNS

Protocollo DNS: domande (query) e messaggi di risposta, entrambi con lo stesso formato.



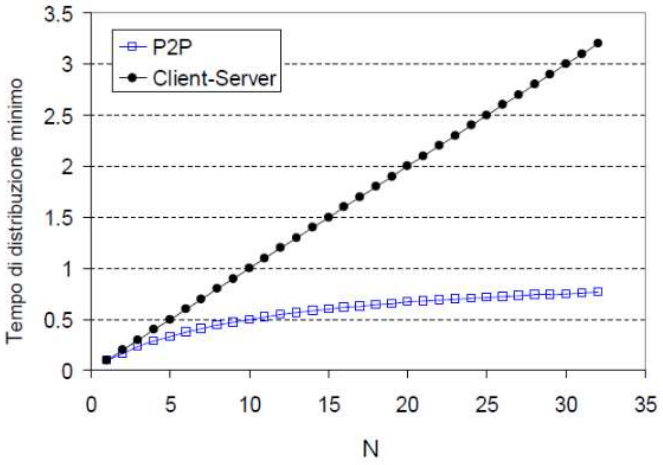
CONDIVISIONE FILE PEER-TO-PEER

Ci sono coppie di host connessi in modo intermittente, chiamati peer, che comunicano direttamente l'uno con l'altro. I peer non appartengono ai fornitori dei servizi, ma sono computer controllati dagli utenti. Viene utilizzata questa modalità di condivisione per:

- Distribuzione di file
- Ricerca informazioni

DISTRIBUZIONE DI FILE P2P

Consideriamo la distribuzione di un file voluminoso. In una distribuzione di file client-server, il server deve inviare una copia del file a ciascun peer, ponendo un enorme fardello sul server e consumandone un'elevata quantità di banda. In una distribuzione di file con P2P ciascun peer può redistribuire agli altri qualsiasi porzione del file abbia ricevuto, aiutando in questo modo il server nel processo di distribuzione.



Il tempo di distribuzione è il tempo richiesto perché tutti gli N peer ottengano una copia del file. Per N sufficientemente grande, il tempo di distribuzione nel caso client-server è generalmente dato da NF/u_s, dove u_s è la banda di upload del collegamento di accesso del server.

Quindi, il tempo di distribuzione aumenta linearmente con il numero N di peer.

BIT TORRENT

BitTorrent è un diffuso protocollo P2P per la distribuzione di file. L'insieme di tutti i peer che partecipano alla distribuzione di un particolare file è chiamato torrent. I peer in un

torrent scaricano **chunk** (parti) del file di uguale dimensione uno dall' altro, con una dimensione tipica di 256 kbyte. Quando un peer entra a far parte di un torrent per la prima volta, non ha chunk del file. Col passare del tempo accumula sempre più pezzi che, mentre scarica, invia agli altri peer. I peer possono entrare e uscire a piacimento dal torrent. Una volta ottenuto l'intero file, il peer può lasciare il torrent (egoisticamente) o (altruisticamente) rimanere collegato.

Indice nei sistemi P2P

Tiene corrispondenza tra le informazioni e la loro posizione negli host. In meccanismi di file-sharing l'indice tiene traccia dinamicamente della posizione dei file. I peer comunicano all'indice ciò che possiedono e consultano l'indice per determinare dove trovare i file. In meccanismi di messaggeria istantanea invece l'indice crea la corrispondenza tra utenti e posizione. Quando l'utente lancia l'applicazione, informa l'indice della sua posizione e può consultare l'indice per determinare l'indirizzo IP dell'utente.

Directory centralizzata

Quando il peer si collega, informa il server centrale con indirizzo IP e contenuto. Il problema sta nel collo di bottiglia per le prestazioni e un unico punto di guasto.

Query flooding

Sistema di trasferimento dati completamente distribuito. Ciascun peer indicizza i file che rende disponibili per la condivisione. Il messaggio di richiesta viene trasmesso sulle connessioni TCP esistenti

Copertura Gerarchica

Combina le migliori caratteristiche di directory centralizzata e query flooding. Ogni peer è leader di un gruppo o è assegnato a un gruppo con un leader. Il leader del gruppo tiene traccia del contenuto di tutti i suoi figli. Un esempio di utilizzo della copertura gerarchica è Skype.

PROGRAMMAZIONE DELLE SOCKET

Una tipica applicazione di rete consiste in una coppia di programmi, detti client e server, che risiedono su sistemi differenti. Quando questi due programmi vengono eseguiti creano un processo client e un processo server che comunicano "scrivendo in" e "leggendo da" delle socket. La maggior parte delle odierne applicazioni di rete implica la comunicazione tra programmi client e server creati da sviluppatori indipendenti, utilizzando le regole dell'RFC. Durante la fase di sviluppo, una delle prime decisioni da prendere è se l'applicazione debba sfruttare TCP o UDP.

SOCKET CON TCP

Prima di iniziare a scambiarsi dati, il programma client deve contattare il programma server specificando l'indirizzo IP e il numero di porta del processo server. Client e server devono effettuare un'operazione di handshake e stabilire una connessione TCP. Un capo della connessione TCP è attaccato alla socket lato client, l'altro a quella lato server. Una

volta stabilita la connessione TCP, quando un lato vuole inviare dati all'altro, deve solo mettere i dati nella connessione TCP attraverso la sua socket.

Con il processo server in esecuzione, il processo client può inizializzare una connessione TCP verso il server. Ciò viene fatto nel programma client creando una socket. TCP avvia un handshake a tre vie e stabilisce una connessione TCP con il server.

SOCKET CON UDP

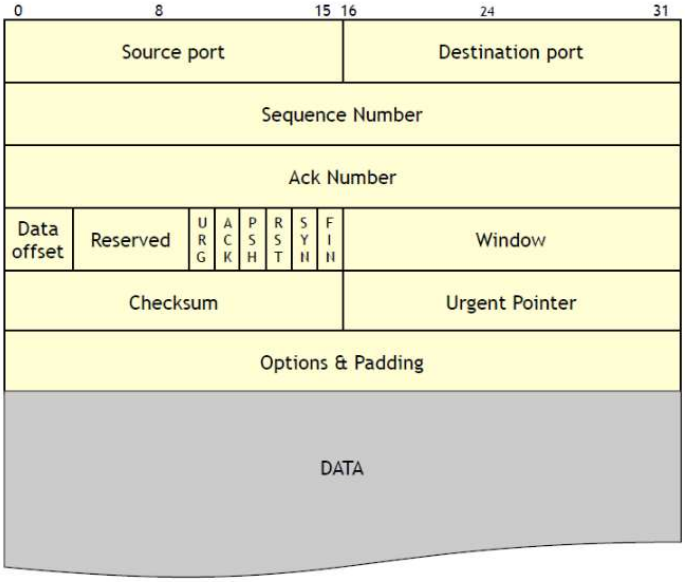
Quando viene creata una socket, le si assegna un identificatore, chiamato numero di porta (*port number*). L'indirizzo di destinazione del pacchetto include l'indirizzo IP dell'host di destinazione e il numero di porta della socket di origine.

LIVELLO DI TRASPORTO

Fornisce un canale di trasporto end-to-end ideale e privo di errori tra due utenti, indipendentemente dalla rete, mettendo a disposizione una comunicazione logica tra processi applicativi di host differenti. Questo avviene spezzando (se necessario) i messaggi applicativi in parti più piccole e aggiungendo a ciascuna di esse un'intestazione di trasporto per creare un segmento.

Internet, e più in generale una rete TCP/IP, mette a disposizione del livello di applicazione due diversi protocolli. Uno è UDP (user datagram protocol), che fornisce alle applicazioni un servizio non affidabile e non orientato alla connessione, l'altro è TCP (transmission contro! protocol), che offre un servizio affidabile e orientato alla connessione.

PROTOCOLLO TCP



- **Source port** – **Destination port** [16 bit]: indirizzi della porta sorgente e della porta destinazione;
- **Sequence Number** [32 bit]: numero di sequenza del primo byte del payload;
- **Acknowledge Number** [32 bit]: numero di sequenza del prossimo byte che si intende ricevere (ha validità se il segmento è un ACK);
- **Offset** [4 bit]: lunghezza dell'header TCP, in multipli di 32 bit;
- **Reserved** [6 bit]: riservato per usi futuri;
- **Window** [16 bit]: ampiezza della finestra di ricezione (comunicato dalla destinazione alla sorgente);
- **Checksum** [16 bit]: risultato di un calcolo che serve per sapere se il segmento corrente contiene errori nel campo dati;
- **Urgent pointer** [16 bit]: indica che il ricevente deve iniziare a leggere il campo dati a partire dal numero di byte specificato. Viene usato se si inviano comandi che danno inizio ad eventi asincroni "urgenti";
- **Options and Padding** [lunghezza variabile]: riempimento (fino a multipli di 32 bit) e campi opzionali come ad esempio durante il setup per comunicare il MSS;
- **Flag** [ogni flag è lunga 1 bit]:
  – **URG**: vale uno se vi sono dati urgenti; in questo caso il campo urgent pointer ha senso

– **ACK**: vale uno se il segmento è un ACK valido; in questo caso l'acknowledge number contiene un numero valido
– **PSH**: vale uno quando il trasmettitore intende usare il comando di PUSH;
– **RST**: reset; resetta la connessione senza un tear down esplicito
– **SYN**: synchronize; usato durante il setup per comunicare i numeri di sequenza iniziale
– **FIN**: usato per la chiusura esplicita di una connessione

Per distinguere a quale applicazione sia destinato il processo, si introduce il concetto di porta, che non è altro che un codice che identifica un applicazione.

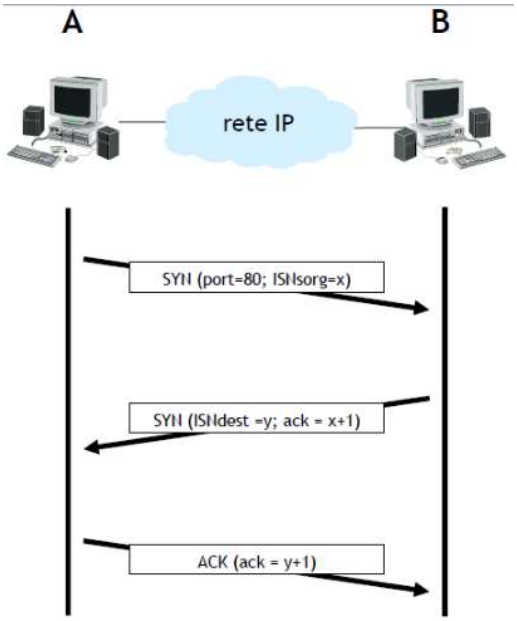
Il numero di porta può essere:

- Statico (identificativi associati a porte largamente utilizzate come posta elettronica);
- Dinamico (identificativi associati dal sistema operativo direttamente all'apertura della connessione).

La porta sorgente e la porta destinazione non sono necessariamente uguali. Le porte servono per il multiplexing e il demultiplexing dei dati da parte del TCP.

INIZIO CONNESSIONE

L'instaurazione della connessione avviene secondo la procedura detta di "three-way handshake".



parametri specificati: numero di porta dell'applicazione cui si intende accedere e Initial Sequence Number (ISNA)

parametri specificati: ISNB e riscontro (ACK) ISNA

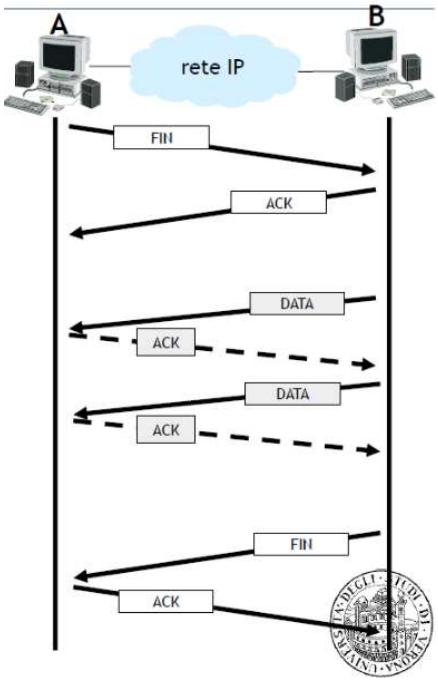
riscontro (ACK) dell'ACK di B

Nota bene:

- Un segmento SYN è un segmento vuoto (è presente solo l'header) in cui il bit SYN è posto a 1;
- Un segmento ACK è un segmento vuoto (è presente solo l'header) in cui il bit ACK è posto a 1;
- Un segmento SYN ACK è un segmento vuoto (è presente solo l'header) in cui i bit SYN e ACK sono posti a 1;
- Un segmento dati è un segmento contenente i dati dell'applicazione in cui i bit SYN e ACK sono posti a 0.

Mss

Tra i vari parametri scambiati all'inizio di una connessione vi può essere anche la MSS, che consente ad una stazione di specificare la dimensione massima (espressa in numero di byte) del campo dati dei segmenti che è disponibile a ricevere (default = 536 byte).



FINE CONNESSIONE

Essendo la connessione bidirezionale, la terminazione deve avvenire in entrambe le direzioni.

La stazione che non ha più dati da trasmettere e decide di chiudere la connessione invia un segmento FIN (segmento con il campo FIN posto a 1 e il campo dati vuoto).

La stazione che riceve il segmento FIN invia un ACK e indica all'applicazione che la comunicazione è stata chiusa nella direzione entrante.

Se questa procedura avviene solo in una direzione (*half close*), nell'altra il trasferimento dati può continuare.

GESTIONE DEGLI ERRORI E DELLE PERDITE

Il TCP garantisce la corretta e ordinata consegna dei segmenti. Le situazioni di errore vengono individuate nel campo checksum. Sia in caso di errore che in caso di perdite, il TCP si occupa della ritrasmissione.

Timeout (RTO)

Tempo entro il quale la sorgente si aspetta di ricevere un riscontro (ACK). Nel caso non arrivi, la sorgente procede alla ritrasmissione. L'RTO non può essere statico perché molto variabile, deve essere calcolato di volta in volta, basandosi sull'**RTT**, cioè l'intervallo di tempo tra l'invio di un segmento e la ricezione del riscontro di quel segmento.

$$SRTT_{attuale} = (\alpha * SRTT_{precedente}) + ((1-\alpha) * RTT_{istantaneo})$$

$RTT_{istantaneo}$ → misura di RTT sull' ultimo segmento
$SRTT_{precedente}$ → stima precedente del valore medio di RTT
$SRTT_{attuale}$ → stima attuale del valore medio di RTT
$\acute{a}$ → coefficiente di peso $(0 < \alpha < 1)$

La sorgente tipicamente attende fino a 2 volte il RTT medio (SRTT) prima di considerare il segmento perso e ritrasmetterlo. In caso di ritrasmissione, il RTO per quel segmento viene ricalcolato in base ad un processo di *exponential backoff,* raddoppiando il tempo di timeout.

CONTROLLO DEL FLUSSO

Azione preventiva finalizzata a limitare l'immissione di dati in rete a seconda della capacità end-to-end di questa. Per incrementare l'efficienza è possibile trasmettere più segmenti consecutivamente senza attendere ogni singolo riscontro.

TCP offre controllo di flusso facendo mantenere al mittente una variabile chiamata finestra di ricezione (*receive window*) che, in sostanza, fornisce al mittente un'indicazione dello spazio libero disponibile nel buffer del destinatario. Alla ricezione dei riscontri dei segmenti iniziali della sequenza, la finestra scorre a destra permettendo al trasmissione di nuovi segmenti (*sliding window*). La dimensione della finestra è fissata in base al valore "Window" (contenuto nell'header TCP) espresso dalle due stazioni durante la fase di connessione e durante lo scambio di dati.

In caso di perdita di pacchetti, avviene ***Go-back-n*** (far tornare la finestra indietro e ritrasmettere tutti i segmenti) oppure ***Selective Repeat*** (ritrasmettere solo il segmento perso). Adesso tutte le implementazioni di TCP utilizzano la ripetizione selettiva per la maggior efficienza in caso di perdite singole e senza sovraccaricare la rete.

CONTROLLO DI CONGESTIONE

Azioni da intraprendere come reazione alla congestione di rete. In caso di congestione della rete, a causa dei buffer limitati degli apparati di rete, alcuni segmenti potrebbero venire persi, sintomo di congestione.

La sorgente dovrebbe essere in grado di reagire diminuendo il tasso di immissione dei nuovi segmenti.

Una soluzione potrebbe essere variare dinamicamente la dimensione della finestra di trasmissione, chiamata **Congestion Window (CWND)**. Due algoritmi per regolare la CWND sono:

- ***Slow Start***: per ciascun riscontro la CWND aumenta di 1 segmento. Perciò si avrà un esponenziale evoluzione della CWND: 1-2-4-8-16-32…
- ***Congestion Avoidance***: per ciascun riscontro ricevuto, la finestra aumento di 1/CWND. L'evoluzione della CWND è perciò lineare.

Le variabili da considerare sono perciò:

- **Congestion Window (CWND**): dimensione della finestra di trasmissione;
- **Receive Window (RECWND):** dimensione massima della finestra di ricezione;
- **Slow Start Threshold (SSTHRESH):** dimensione della finestra raggiunta la quale, invece di seguire l'algoritmo di *Slow Start*, si segue l'algoritmo di *Congestion Avoidance*;
- **RTT**;
- **RTO**;

In caso di errori o perdita di segmenti, la trasmissione si interrompe, si attende lo scadere del timeout RTO, si pone SSTHRESH = CWND / 2 e CWND = 1.

In caso di perdite consecutiva, il timeout per quel segmento viene raddoppiato, CWND =1 e SSTRESH = 2.

Fast Retransmit – Fast Recovery

La perdita di segmenti è causata da:

- Errori di trasmissione (su un singolo segmento);
- Congestione (su più segmenti).

Fast Retransmit e Fast Recovery sono due algoritmi che progettati per le perdite singole. Infatti utilizzandoli il segmento considerato perso viene subito ritrasmesso e la CWND non viene chiusa eccessivamente.

Negli ACK il campo **Ack Number** contiene il successivo numero di sequenza che ci si aspetta arrivi per controllare che i precedenti sono stati inviati correttamente. Se un segmento arriva fuori sequenza, la destinazione invia un ACK indicando il numero di sequenza mancante. La ricezione di un numero sufficientemente alto di ACK duplicati può essere interpretata come forte di indicazione che è avvenuta una perdita.

Se alla sorgente arrivano 3 ACK duplicati:

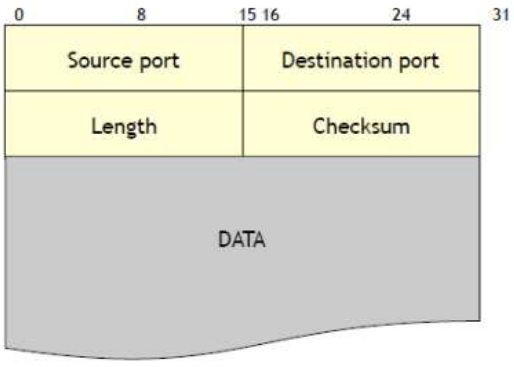
1) SSTHRESH = CWND / 2
2) Fast Retransmit (senza aspettare l'RTO)
3) CWND = SSTHRESH + 3
4) Fast Recovery
5) CWND++ per ogni successivo ACK duplicato

Quando la sorgente riceve l'ACK di conferma del segmento ritrasmesso:

1) CWND = SSTHRESH
2) Trasmissione con Congestion Avoidance

In qualsiasi altro caso, la sorgente attente lo scadere dell'RTO, ritrasmette e riparte in Slow Start.

PROTOCOLLO UDP



- **Source Port e Destination Port [16 bit]**: identificano i processi sorgente e destinazione dei dati;
- **Length [16 bit]**: lunghezza totale (espressa in byte) del datagramma, compreso l'header UDP;
- **Checksum [16 bit]**: campo di controllo che serve per sapere se il datagramma corrente contiene errori nel campo dati.

LIVELLO DI RETE

Il ruolo del livello di rete è trasferire pacchetti da un host a un altro, incapsulando i segmenti in **datagrammi**.

Per fare questo è possibile identificare tre importanti funzioni:

1) INOLTRO (*forwarding*): quando un router riceve un pacchetto, lo deve trasferire sull'appropriato collegamento di uscita;
2) INSTRADAMENTO (*routing*): il livello di rete deve determinare, tramite algoritmi di instradamento (algoritmi di routing), il percorso che i pacchetti devono seguire;
3) INSTAURAZIONE DELLA CONNESSIONE (*connection setup*): alcune architetture di rete richiedono che i router, lungo il percorso scelto tra la sorgente e la destinazione, effettuino l'handshake tra loro per impostare lo stato, prima che i pacchetti inizino a fluire.

Per inoltrare i pacchetti i router estraggono dal campo di intestazione il valore che utilizzano come indice nella tabella di inoltro (***tabella di forwarding*** *o forwarding table*). Il risultato indica a quale interfaccia di collegamento il pacchetto debba essere diretto.

A seconda del protocollo, il valore nell'intestazione del pacchetto può rappresentare l'indirizzo di destinazione o la connessione cui appartiene.

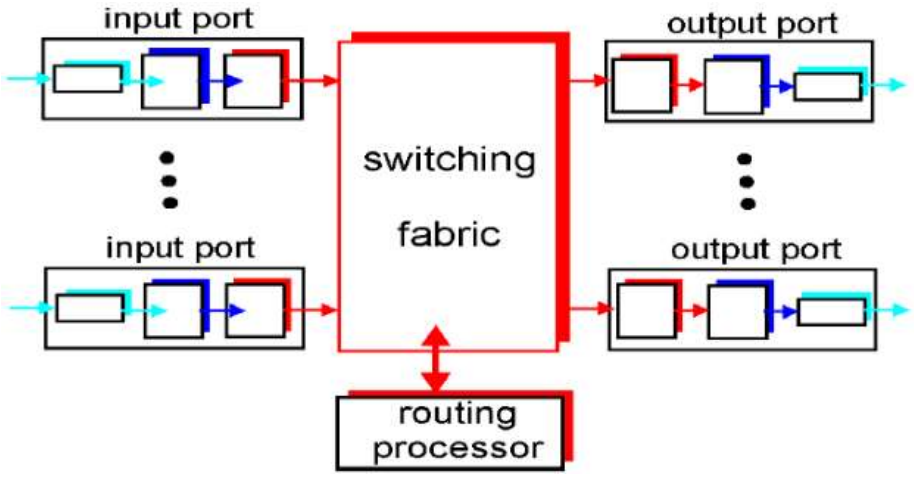
Commutatore di pacchetto

Il termine (*packet switch*) indica un generico dispositivo che si occupa del trasferimento dall' interfaccia in ingresso a quella in uscita, in base al valore del campo nell' intestazione del pacchetto.

- Alcuni commutatori di pacchetti, chiamati commutatori **a livello di collegamento**, stabiliscono l 'inoltro in relazione al valore del campo a livello di collegamento.
- Altri, chiamati **router**, prendono le decisioni di inoltro basandosi sul valore nel campo a livello di rete.

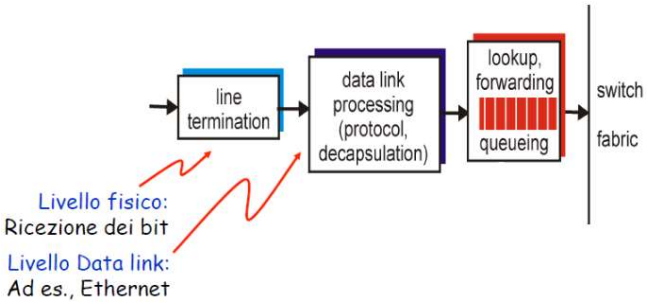
ARCHITETTURA DI UN ROUTER

Le due funzionalità chiave di un router sono l'esecuzione di algoritmi e protocolli di routing e la commutazione (trasferimento) di datagrammi dalla porta di input a quella di output.



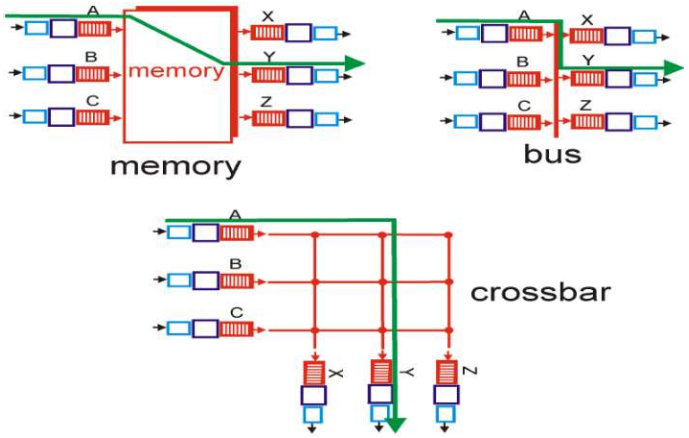
PORTE DI INPUT

Data la destinazione, deve determinare la porta di output mediante le tabelle di forwarding in memoria locale. Se non riesce a completare l'elaborazione dei pacchetti a velocità di linea deve gestire le code (queuing).



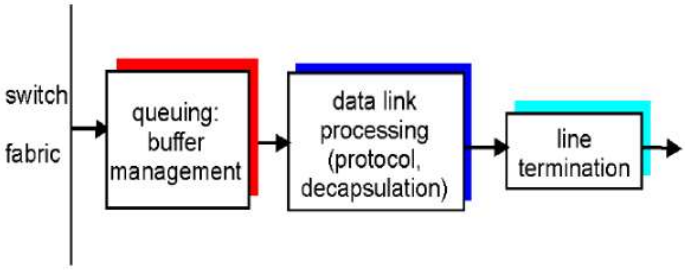
MATRICI DI COMMUTAZIONE

1) Basata su memorie: commutazione controllata dalla CPU, il pacchetto veniva copiato sulla memoria di sistema → limitato dalla velocità della memoria;
2) Via Bus: utilizzo di un bus condiviso, dove la velocità di commutazione è data dalla velocità del bus → contesa per l'utilizzo del bus;
3) Con Rete interconnessa: risolve i limiti dell'architettura via bus;



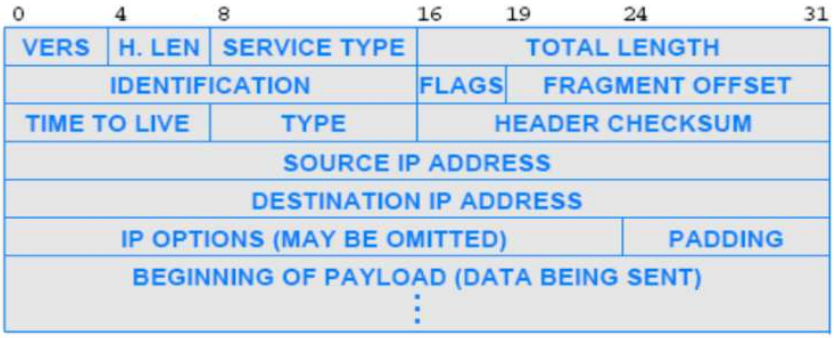
PORTE DI OUTPUT

Il **buffering** è necessario quando la velocità di arrivo dei datagrammi è superiore alla velocità di trasmissione. Lo **scheduling** è utilizzato per determinare l'ordine di trasmissione dei datagrammi in coda nel buffer. È necessario un buffer sulla porta di output poiché la velocità di arrivo è maggiore della velocità di trasmissione, infatti a causa dell'overflow del buffer sono dovute spesso perdite e ritardi.



PROTOCOLLO IP

Nello stack TCP/IP con "datagramma IP" si intende un pacchetto. Ciascun datagramma è formato da un header lungo dai 20 ai 60 bytes, seguito dai dati (**payload**) che possono occupare fino a 64K byte.



- **VERS**: specifica la versione del protocollo;
- **H.LEN** (header length): specifica la dimensione dell'header;
- **SERVICE TYPE**: identificano la classe di servizio del datagramma;
- **TOTAL LENGTH**: numero totale di byte del datagramma intero (header + payload);
- **IDENTIFICATION**: numero utilizzato per ricomporre un datagramma nel caso in cui venga frammentato;
- **FLAGS**: specifica se il datagramma è un frammento o meno, ed eventualmente se è l'ultimo frammento;
- **FRAGMENT OFFSET**: specificano l'offset del frammento rispetto al datagramma originale;
- **TIME TO LIVE**: intero inizializzato alla sorgente, decrementato da ciascun router attraversato dal datagramma. Se raggiunge lo 0, viene scartato e mandato un messaggio di errore alla sorgente;
- **TYPE**: specifica il tipo di dati trasportato nel payload;
- **HEADER CHECKSUM**: checksum dell'header;
- **SOURCE IP ADRESS**: indirizzo della sorgente;
- **DESTINATION IP ADRESS**: indirizzo della destinazione;
- **IP OPTIONS**: campi opzionali con info addizionali;
- **PADDING**: se il campo "ip option" è presente e non è multiplo di 32, vengono aggiunti degli 0.

FRAMMENTAZIONE IP

A seconda della tecnologia hardware, i diversi tratti di rete possono trasportare trame con una lunghezza massima predefinita: *Maximum Transmission Unit* (**MTU**). Un router può comunque connettere reti con MTU differenti.

Quando la dimensione di un datagramma è superiore alla massima MTU della rete verso cui deve essere inviato il router lo divide in "frammenti" e li invia come se fossero datagrammi indipendenti. Nel campo FLAG viene specificato se è un frammento e nel campo FRAGMENT OFFSET viene indicata la posizione in cui collocare il frammento per ricostruire il datagramma. Il router perciò usa MTU e dimensione dell'header per calcolare la dimensione massima dei frammenti che può inviare e il numero di frammenti necessario per dividere il datagramma.

Il router può anche riassembrale i frammenti nel caso in cui provengano da una rete con minor MTU e debbano essere inviati a una rete con maggior MTU, riducendo la quantità di dati da memorizzare nel router e facendo così cambiare percorso ai datagrammi in maniera dinamica. Si inizia a riassemblare i frammenti quando tutti i frammenti sono presenti. IP specifica un tempo massimo per l'arrivo di tutti i frammenti, oltre al quale viene scartato tutto.

INDIRIZZAMENTO NEL LIVELLO DI RETE

Tutti gli host devono utilizzare uno schema di indirizzamento comune. Esso deve essere unico. Non possono essere utilizzati indirizzi MAC perché esistono formati diversi per ciascuna tecnologia. Quando un host invia un pacchetto in internet, deve specificare il proprio indirizzo IP e quello della destinazione.

Per semplificare la gestione e la lettura degli indirizzi, viene utilizzata la "**notazione decimale puntata**", che prevede di dividere 32 bit in 4 sezioni da 8 bit, esprimendo ciascuna sezione in decimale e separandole con un punto. Gli indirizzi IP sono divisi concettualmente in due parti:

- PREFISSO: identifica la rete fisica a cui l'host è connesso;
- SUFFISSO: identidifica l'host specifico all'interno della rete.

I suffissi vengono generati localmente mentre i prefissi vengono assegnati a livello globale in modo che ci sia univocità.

INDIRIZZAMENTO CLASSLESS

Invece di avere un insieme ristretto di lunghezze per i prefissi / suffissi, la scelta della lunghezza viene resa arbitraria. Visto che le decisioni di routing vengono prese analizzando il prefisso, bisogna trovare un modo per identificare dove finisce il prefisso e comincia il suffisso. Invece di aggiungere la dimensione del prefisso esplicitamente, si preferisce usare un'altra tecnica nota come maschera di indirizzo o maschera di subnet, che consiste in un valore a 32 bit in cui sono posti a uno tutti i bit fino a raggiungere la lunghezza del prefisso.

Un router tiene in memoria le reti di destinazione e le corrispondenti maschere. Il router confronta l'indirizzo di destinazione con in prefissi in memoria.

Per facilitarne la gestione da parte degli utenti, si utilizza una notazione più semplice e diretta (*Classless Inter-Domain Routing*, **CIDR**) specificando dopo l'indirizzo con "\m", dove per m si intende l'ultimo bit del prefisso. Dopo aver ricevuto il prefisso CIDR da un ISP, il cliente può assegnare liberamente gli indirizzi di host ai propri utenti.

INDIRIZZI IP SPECIALI

Il protocollo IP definisce un insieme di indirizzi speciali riservati.

Prefix	Suffix	Type Of Address	Purpose
all-0s	all-0s	this computer	used during bootstrap
network	all-0s	network	identifies a network
network	all-1s	directed broadcast	broadcast on specified net
all-1s	all-1s	limited broadcast	broadcast on local net
127/8	any	loopback	testing

1) Indirizzi di rete

Indirizzo che denota il solo prefisso assegnato a una rete, non deve mai comparire come indirizzo di destinazione di un pacchetto.

2) Indirizzi di "Directed Broadcast"

Serve per semplificare l'invio a tutti gli host di una determinata rete. È costruito ponendo a 1 tutti i bit del suffisso.

3) Indirizzi di "Limited Broadcast"

Invio di un messaggio a tutti gli host che appartengono alla stessa rete dell'host mittente. È formato ponendo tutti i bit a 1.

4) Indirizzo "Questo Host"

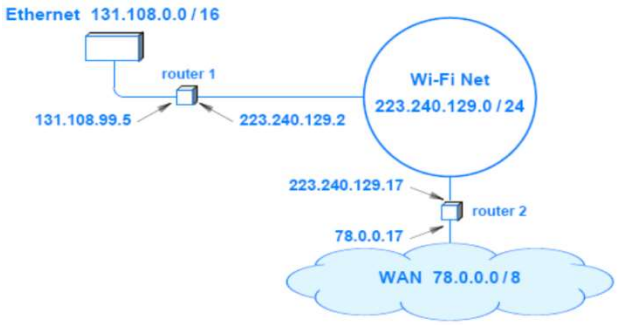
Durante la richiesta dell'indirizzo IP da parte di un dispositivo appena connesso alla rete (startup), l'indirizzo composto da tutti 0 viene chiamato anche "questo host" prima dell'assegnamento finale.

5) Indirizzo di Loopback

Usato per sviluppare applicazioni di rete in fase di test. In fase di sviluppo per testare un applicazione il programmatore invece di usare due host differenti che comunicano, egli utilizza un host unico utilizzando gli indirizzi di loopback per comunicare. Consiste in un qualsiasi suffisso, con il prefisso 127.0.0.0/8 ma, dato che il suffisso è irrilevante, solitamente si usa 127.0.0.1 che è il primo host disponibile.

Principio di indirizzamento IP del router

Ciascun router deve avere due o più indirizzi IP, uno per ciascuna rete a cui è connesso in quanto ogni router ha connessione verso più reti e ogni prefisso corrisponde ad una rete fisica.



INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

Protocollo complementare a quello IP usato principalmente per inviare messaggi di errore alla sorgente in caso di problemi.

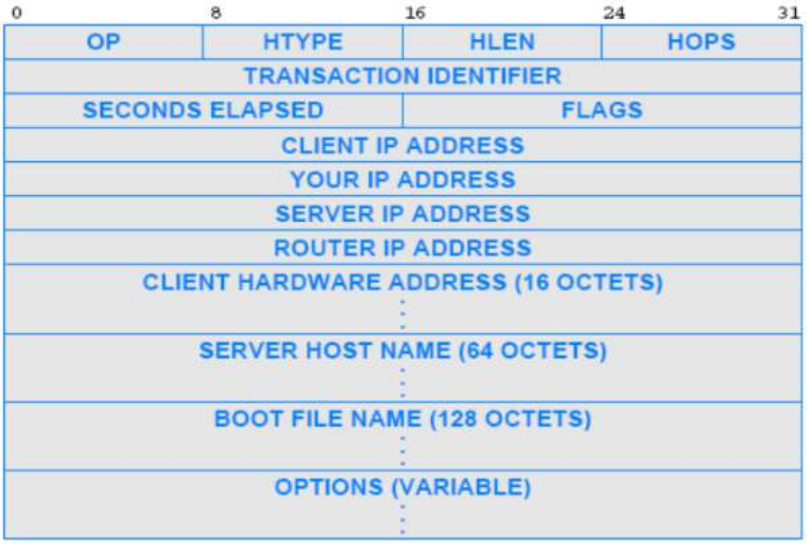
Number	Type	Purpose
0	Echo Reply	Used by the ping program
3	Dest. Unreachable	Datagram could not be delivered
5	Redirect	Host must change a route
8	Echo	Used by the ping program
11	Time Exceeded	TTL expired or fragments timed out
12	Parameter Problem	IP header is incorrect
30	Traceroute	Used by the traceroute program

ICMP può segnalare errori oppure richiedere informazioni, utilizzando IP per trasportare i propri messaggi. Quando un router ha da inviare un messaggio ICMP infatti crea un datagramma IP e mette nel payload il messaggio ICMP. Se un messaggio ICMP di errore causa un errore, non viene inviato nessun altro messaggio di errore per non creare un effetto "valanga".

DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)

Quando un host o un router vengono accesi, c'è bisogno di un inizializzazione. La configurazione degli host si chiama **bootstrapping**, utilizzando il Bootstrap Protocol (BOOTP) per ottenere una serie di parametri con una sola richiesta.

DHCP con un approccio "plug-and-play networking" si connette alla rete e ottiene molte informazioni automaticamente. Quando un host accede infatti invia una richiesta DHCP broadcast. Il serve manda una risposta DHCP offrendo un indirizzo al client, sia esso dinamico oppure statico. Una volta che l'host ha trovato il server DHCP, lo memorizza per utilizzi futuri.



- OP: indica se si tratta di una "Request" o di "Response";
- HTYPE: indica il tipo di hardware della rete;
- HLEN: indica il tipo di lunghezza dell'indirizzo hardware;
- FLAGS: indica se l'host può ricevere messaggi broadcast o risposte dirette;
- HOPS: indica a quanti server bisogna rigirare la richiesta;
- TRANSACTION IDENTIFIER: indica se la risposta si riferisce a una propria richiesta;
- SECONDS ELAPSED: indica quanti secondi sono passati dall'avvio dell'host;

Il resto dei campi viene utilizzato nelle risposte per inviare le informazioni necessarie alla sorgente per potersi configurare.

ALGORITMI DI ROUTING

Si parla di consegna diretta quando un host deve inviare un messaggio ad un altro host connesso alla propria rete, si parla di consegna indiretta quando l'host destinatario appartiene ad un'altra rete, per cui il router si farà carico della consegna.

Il routing è il processo di scoperta del cammino da una sorgente ad ogni destinazione della rete. Un protocollo di routing gestisce una tabella di routing nei router. La **tabella di routing** indica per ogni destinazione, qual è l'output su cui inviare il pacchetto. I nodi perciò fanno scelte locali basandosi su tipologia globale. Ciascun router perciò deve conoscere qualcosa sullo stato globale, grazie alla capacità di riassumere proprie del protocollo di routing.

Per assicurare che tutti i touter mantengano le informazioni su come raggiungere ogni possibile destinazione, ciascun router utilizza un **protocollo di propagazione dei cammini**. Infatti quando si viene a sapere di un cambiamento dei cammini bisogna aggiornare immediatamente la propria tabella di routing. L'algoritmo di routing inoltre per definizione è l'algoritmo che trova il cammino a costo minimo.

Distance Vector Algorithms

Periodicamente ogni nodo invia ai propri vicini il vettore delle distanze. Quando un nodo x riceve il DV da un vicino, aggiorna il proprio DV usando l'equazione di Bellman-Ford:

$$D(x,y) \leftarrow \min_v\{c(x,v) + D(v,y)\} \quad \text{per ciacun nodo } y \in N$$

Ciascuna iterazione locale è causata o dal cambio del costo del collegamento locale o dalla ricezione del DV da un vicino. Ciascun nodo manda il proprio DV solo se cambia. Bisogna però risolvere i vari problemi come il "counting to infinity", stabilendo il costo massimo di un collegamento a 15. Con lo **Split Horizon**, quando un router invia il DV ad un vicino k:

- Omette le destinazione che hanno k come next hop (semplice);
- Imposta la distanza a infinito per le destinazioni che hanno k come next hop (*poisoned reverse*).

I principali problemi consistono in una convergenza lenta in caso di guasto e il fatto che funziona con reti di dimensione limitata.

Routing Information Protocol (RIP)

Protocollo intra-dominio semplice da implementare e gestire, basato su Distance-Vector con i suoi problemi correlati. La metrica è basata sul conteggio degli hop, dove 15 è considerato infinito per limitare il tempo di convergenza. Ciascun router invia i vettori delle distanze a tutti i vicini con UDP (porta 520) ogni 30 secondi o qualora le tabelle di routing cambino per motivi esterni. Le entry hanno un timeout massimo di 3 minuti, nel caso in cui scada allora la distanza viene posta a 15.

Le tabelle di routing di RIP sono gestite da processi di livello applicativo. Vengono mantenuti 3 timer per le operazioni:

- Aggiornamento periodico (30 s): per inviare messaggi di aggiornamento;
- Timer di invalidazione (180 s): se allo scadere del timer un entry non è stata aggiornata, viene considerata non più valida;

- Timer per Garbage Collection (120 s): prima di rimuovere un entry non valida, rimane marcata con distanza = 15.

I messaggi di **input** possono essere "Request" nel caso in cui un router si sia appena avviato e necessiti di una risposta diretta alla richiesta oppure "Response" che consiste in messaggi di aggiornamento o risposta a query e di conseguenza il routing aggiorna le proprie tabelle.

I messaggi di **output** vengono generati quando un router viene avviato, se richiesto dalla procedura di processing di input o dall'aggiornamento regolare.

RIPv2 è una versione successiva e ottimizzata del protocollo RIPv1, con ulteriori campi come il Route Tag e il Next Hop, oltre all'autenticazione che invalida la possibilità di trasmissioni false. Entrambi i modelli sono **interoperabili** in quando RIPv2 risponde alle richieste RIP con risposte RIP, ignorando i campi aggiuntivi.

Le limitazioni dovute alla semplicità sono ad esempio che le destinazioni con metriche superiori a 15 non sono raggiungibili. Un'altra conseguenza sono tabelle di routing sub-ottime e infine, non richiedendo autenticazione, router mal configurati potrebbero creare danni.

Link State Algorithms

Cercano di ottenere una visione globale con un approccio iterativo. La principale differenza con gli algoritmi Distance Vector è che ciascun nodo colleziona prima tutti i link state e successivamente applica l'algoritmo di Dijkstra al grafo. Dopo ciascuna iterazione, l'algoritmo trova una nuova destinazione e il cammino minimo verso tale destinazione. Così facendo, dopo m iterazioni, l'algoritmo ha esplorato i cammini fino a m hop dal nodo i.

L'algoritmo di Dijkstra ha complessità in $O(n^2)$ ma esistono alcune implementazioni più efficienti che arrivano anche ad $O(n\log n)$. La scelta del costo dei collegamenti ha un impatto sul traffico, con metriche statiche, dinamiche e quasi-statiche.

Link State	Distance Vector
❑Le informazioni sulla topologia sono inviate su tutta la rete (flooding)	❑Ciascun router ha una visione limitata della topologia della rete
❑Il miglior cammino viene calcolato da ciascun router localmente	❑Data una destinazione e' possibile individuare il miglior next-hop
❑Il miglior cammino determina il next-hop	❑Il cammino end-to-end e' il risultato della composizione di tutte le scelte di next-hop
❑Funziona solo se la metrica e' condivisa e uniforme	❑Non richiede metriche uniformi tra tutti i router
❑Esempio: OSPF	❑Esempio: RIP

ROUTING REALE

Il routing precedente fornisce una visione idealizzata, dove tutti i router sono identici e la rete è piatta. In realtà è impossibile memorizzare tutte le destinazioni nelle tabelle di routing e gli scambi delle tabelle saturerebbero i collegamenti. Viene a crearsi perciò il routing gerarchico, un aggregazione di routing in regioni chiamate "*autonomous systems*" (**AS**). I router dentro lo stesso AS utilizzano lo stesso protocollo di routing, chiamato "intra-AS". Esistono poi i **gateway**, dei router di bordo che collegano router i differenti AS, con le tabelle di forwarding configurate sia dagli algoritmi di intra-AS che inter-AS. Si verifica perciò, in caso di pacchetti che devono essere trasmessi intra-AS, un fenomeno chiamato "***hot potato running***", dove si invia il pacchetto al gateway più vicino.

Si differenzia tra routing intra-AS e inter-AS per una questione di:

- Scalabilità: riduce la dimensione delle tabelle i il traffico;
- Policy: l'amministratore può controllare come il traffico viene instradato sulla propria rete;
- Prestazioni: cammini minimi su intra-AS, mentre più amministrazione che velocità su inter-AS;

SOLUZIONI PER LA CARENZA DI INDIRIZZI IP

IETF ha definito alcuni range di indirizzi da utilizzare solamente in ambito privato. Ogni volta che un router pubblico riceve un pacchetto destinato ad un indirizzo IP privato, viene segnalato un errore. Le reti con un solo punto di connessione alla Big Internet possono usare l'indirizzamento privato. Per permettere di ricevere i pacchetti all'interno di una rete privata, è stato necessario introdurre il **Network Address Translation (NAT)**, che consiste nell'assegnare un indirizzo pubblico ad router di confine sull'interfaccia verso la rete esterna. NAT traduce l'indirizzo IP dei datagrammi entranti e uscenti sostituendo l'indirizzo sorgente con il proprio indirizzo pubblico e l'indirizzo destinazione di ogni pacchetto entrante con l'indirizzo privato dell'host corretto. Il router NAT mantiene una tabella di record con il mapping tra indirizzo privato sorgente della comunicazione e indirizzo pubblico destinazione della comunicazione. Agisce anche da gateway di livello 4, traducendo sia l'indirizzo IP che la porta (TCP/UDP).

IPv6

Il protocollo IP utilizza 32 bit per l'indirizzo, ma se l'attuale tasso di crescita viene mantenuto tutti i possibili prefissi verranno prima o poi assegnati bloccando così un eventuale ulteriore crescita. Quando IP verrà sostituito dovrà essere ad indirizzamento limitato, con più funzionalità e sufficientemente flessibile.

IPv6 mantiene molte caratteristiche di IPv4 (versione attuale), tra cui il fatto che è **connectionless** e che l'header del datagramma contiene un **numero massimo di hop** che il datagramma può fare prima di essere scartato. La dimensione degli indirizzi è passata da 32 bit a **128 bit**, sufficiente per contenere eventuali crescite future. L'header è completamente differente, con l'implementazione di un "***Extension Header***", header separati a seconda delle informazioni e di lunghezza variabile. Inoltre supporta il traffico Real-Time, creando un cammino sorgente-destinazione e associando i datagrammi a quel cammino, per applicazioni come audio e video che richiedono maggior qualità di servizio. L'**header di base** ha una lunghezza fissa di 40 Byte , anche se contiene meno campi a causa della lunghezza di *source adress* e *destination address*.

- VERS: versione (6);
- TRAFFIC CLASS: classe di traffico, in base alle richieste che deve rispettare;
- PAYLOAD LENGTH: dimensione dei dati dopo l'header;
- HOP LIMIT: corrisponde al campo TIME-TO-LIVE;
- FLOW LABEL: associa un datagramma ad un cammino specifico;
- NEXT HEADER: specifica il tipo di informazione (dati/prossimo header) che segue l'header corrente.

Frammentazione

La frammentazione in IPv6 è simile alla frammentazione IPv4, seppur con qualche differenza.
Non esistono infatti campi predeterminate di base per la frammentazione, per farlo bisogna aggiungere un extention header di tipo "frammentazione". La presenza stessa di un extention header di tipo frammentazione indica che si tratta di un frammento. La parte non frammentabile è formata dall'header di base e dagli extension header che controllano il routing.Con l'utilizzo di header multipli si risparmia spazio, è possibile aggiungere un insieme ampio di funzionalità senza imporre che tutti gli header abbiano un campo predeterminato. Come con CIDR, la divisione tra prefisso e suffisso è arbitraria, con il concetto di **gerarchia multi-livello**. Il livello più alto corrisponde agli ISP, dopodichè ci sono le organizzazioni per poi passare ai siti web e così via. Esistono anche gli indirizzi speciali:

Tipologia	Scopo
unicast	L' indirizzo corrisponde ad un singolo host. Un datagramma inviato a tale indirizzo viene instradato sul cammino minimo
multicast	L' indirizzo corrisponde ad un insieme di host e i membri dell' insieme possono cambiare in qualsiasi momento. Viene consegnata una copia del datagramma a ciascun membro dell' insieme
anycast	L' indirizzo corrisponde ad un insieme di host che condividono un prefisso. Il datagramma viene consegnato ad uno qualsiasi dei memberi dell' insieme (ad es., all' host piu' vicino)

L'indirizzo IPv6 essendo a 128 bit utilizza la notazione esagesimale, raggruppando insiemi di 16 bit e traducendoli in esadecimale, separati da due punti. Le sequenze di zeri si possono comprimere, scrivendo i due punti e facilitando la trascrizione degli indirizzi IPv4 ponendo a 0 i primi 96 bit.

LIVELLO DI COLLEGAMENTO DATI

L'obiettivo principale è fornire al livello di rete di due macchine adiacenti un canale di comunicazione il più possibile affidabile. Per macchine adiacenti si intende macchine fisicamente connesse da un canale di comunicazione.

Servizi offerti al livello di rete:

- Connectionless senza acknowledge: non viene attivata alcuna connessione con invio delle trame senza attendere feedback;
- Connectionless con acknowledge: non viene attivata alcuna connessione e ogni trama viene riscontrata in modo individuale;
- Connection-oriented con acknowledge: attivata una connessione, ogni trama inviata viene riscontrata in modo individuale.

Le principali funzioni svolte dal livello 2 sono:

- Framing: delimitazione delle trame;
- Rilevazione/gestione errori: controlla se la trama contiene errori;
- Controllo di flusso: gestisce la velocità di trasmissione.

FRAMING

Una volta ricevuti i pacchetti dal livello di rete, bisogna delimitare le trame. Bisogna infatti rendere distinguibile una trama dall'altra attraverso l'utilizzo di opportuni codici all'inizio e alla fine della trama stessa.

Character count

Un campo nell'header del frame indica il numero di "caratteri" del frame stesso.

Bit Stuffing

Ogni trama può includere un numero arbitrario di bit, a patto che inizi e finisca con uno speciale pattern di bit, 01111110, chiamato **bit di flag**. Per evitare che la trama al suo interno contenga un pattern uguale, la sorgente se incontra 5 bit "1" aggiunge uno "0". La destinazione poi toglierà quello "0".

RILEVAZIONE ERRORI

Per evitare eventuali errori derivati dalla trasmissione del livello fisico, nell'header di ogni trama è presente un **checksum**, il risultato di un calcolo fatto utilizzando i bit della trama. Sia sorgente che destinazione effettuano questo calcolo e se coincide allora la trama è corretta.

GESTIONE FLUSSO

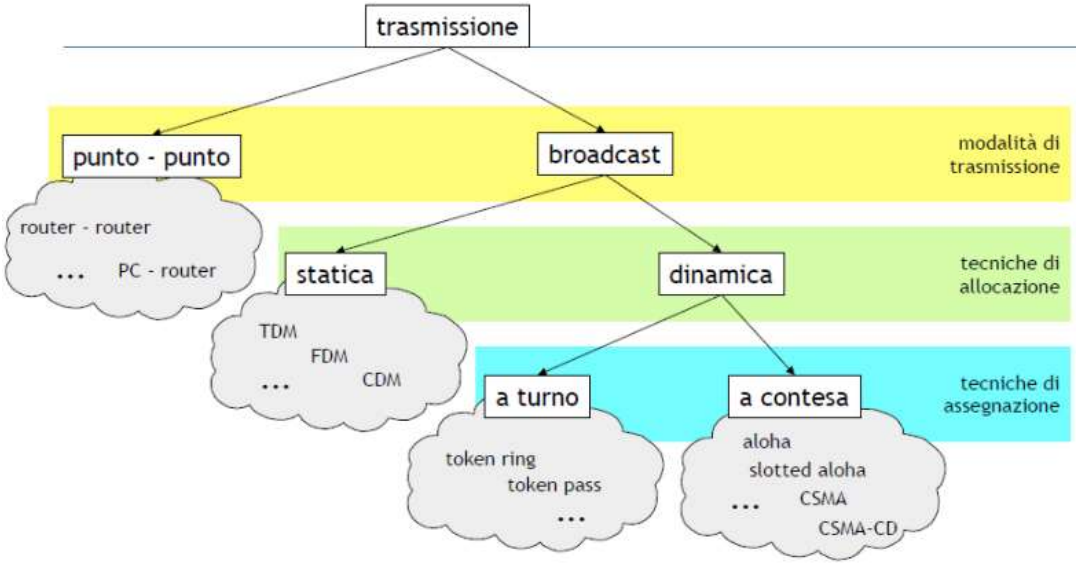
Se la sorgente trasmette trame ad una velocità superiore a quella che la destinazione impiega per accettare l'informazione si crea una congestione del nodo destinazione. Il controllo della velocità di trasmissione è basato perciò su feedback inviati indicando di la quantità di informazione che la destinazione è ancora in grado di gestire e eventualmente di bloccare la trasmissione fino a nuovo comando.

Nelle reti TCP/IP tutto ciò è demandato ai livelli superiori.

IL SOTTO-LIVELLO MAC

Se il mezzo fisico attraverso il quale si trasmettono le trame è condiviso, allora nascono una serie di problematiche come la selezione dell'host che ha diritto di trasmettere e la competizione per la risorsa trasmissiva. Questo sottolivello gestisce le problematiche.

Se due sorgenti trasmettono contemporaneamente infatti vi sarà collisione e l'informazione andrà persa.



ALLOCAZIONE STATICA

Il mezzo trasmissivo viene "partizionato" e ogni partizione viene data alle diverse sorgenti. Esso può avvenire tramite il tempo (*Time Division Multiplexing*) oppure tramite la frequenza (*Frequency Division Multiplexing*). Sono meccanismi semplici ed efficienti in caso di pochi utenti con molto carico costante nel tempo.

ALLOCAZIONE DINAMICA

Il canale viene assegnato di volta in volta a chi ne fa richiesta e può essere utilizzato una volta che questi ha finito di usarlo e lo libera. Il canale può essere assegnato a turno (con sovraccarico gestionale) oppure a contesa (più utilizzato).

ALGORITMI DI ACCESSO MULTIPLO AL MEZZO CONDIVISO CON CONTESA

1) **Pure ALOHA**
   Una sorgente può trasmettere ogni volta che vi sono dati da inviare. La sorgente rileva ascoltando il canale un eventuale collisione e, se succede, aspetta un tempo casuale e ritrasmette la trama. Si definisce il "periodo di vulnerabilità" l'intervallo di tempo in cui può avvenire una collisione che invalida una trasmissione, è pari al doppio del tempo di trama. Il traffico generato (numero di trame per tempo di trama) segue la distribuzione di Poisson. Permette di sfruttare al massimo il 19% degli slot liberi nel caso in cui vengano generate 0.5 trasmissioni per tempo di trama.

2) Slotted ALOHA

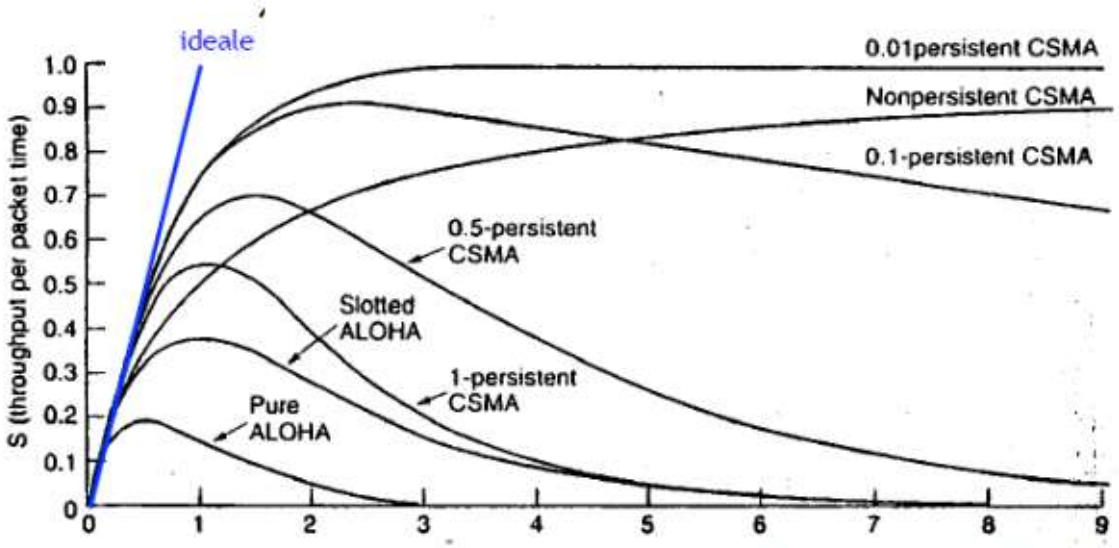
Basato sull'ipotesi del tempo suddiviso ad intervalli discreti (*slotted time*) è un algoritmo basato sul Pure ALOHA, nel quale però la trasmissione di trama può iniziare solo ad intervalli discreti. Per fare ciò è necessario sincronizzare le stazioni. Il periodo di vulnerabilità è perciò il tempo di trama. Permette al massimo di sfruttare il 37% degli slot liberi nel caso in cui viene generata una trasmissione per tempo di trama.

3) CSMA

L'algoritmo *Carrier Sense Multiple Access* comprende la capacità di monitorare lo stato del canale di trasmissione da parte delle stazioni, per verificare se c'è già una trasmissione in corso. Se il canale fosse occupato, si rimanda la trasmissione ad un nuovo istante casuale (non-persistent) oppure si invia nel momento in cui si libera il canale (persistent). In caso di collisione si attende un tempo casuale per ritrasmettere. Il periodo di vulnerabilità è legato al ritardo di propagazione del segnale. Questo algoritmo perciò viene utilizzato in reti il cui ritardo di propagazione è molto inferiore al tempo di trama.

4) CSMA-CD

L'algoritmo CSMA con *Collision Detection* è migliorato in quanto se la stazione che sta trasmettendo rileva la collisione interrrompe immediatamente. Così facendo non si spreca tempo a trasmettere trame già corrotte. Per far sapere che c'è stata una collisione si trasmette una particolare sequenza di jamming.



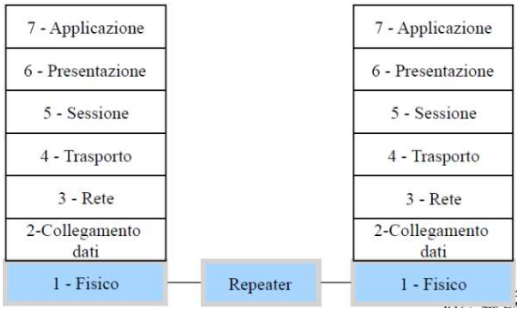
(fonte: A. Tanenbaum, Computer Networks)

LAN ESTESE

Repeater, bridge e switch sono 3 tipi di apparati che estendono le LAN (*Local Area Network*).

Il **dominio di collisione** è la parte della rete per cui, se due stazioni trasmettono dati contemporaneamente, il segnale ricevuto dalle stazioni risulta danneggiato. Il **dominio di broadcast** (segmento data-link) è la parte di rete raggiunta da una trama con indirizzo broadcast. Gli apparati che estendono le LAN possono solo influire sul dominio di collisione.

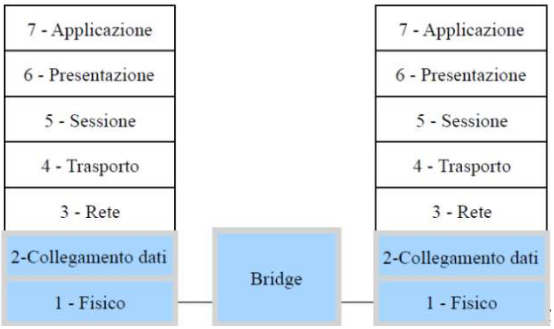
REPEATER E HUB



Interviene a livello fisico replicando le trame in arrivo da un segmento ad un altro, amplificando perciò il segnale. Se un repeater connette più di due segmenti allora si parla di Hub. Il dominio di collisione coincide con il dominio di broadcast, che è il principale problema di queste configurazioni; con i repeater è come se tutte le stazioni condividessero lo stesso mezzo fisico.

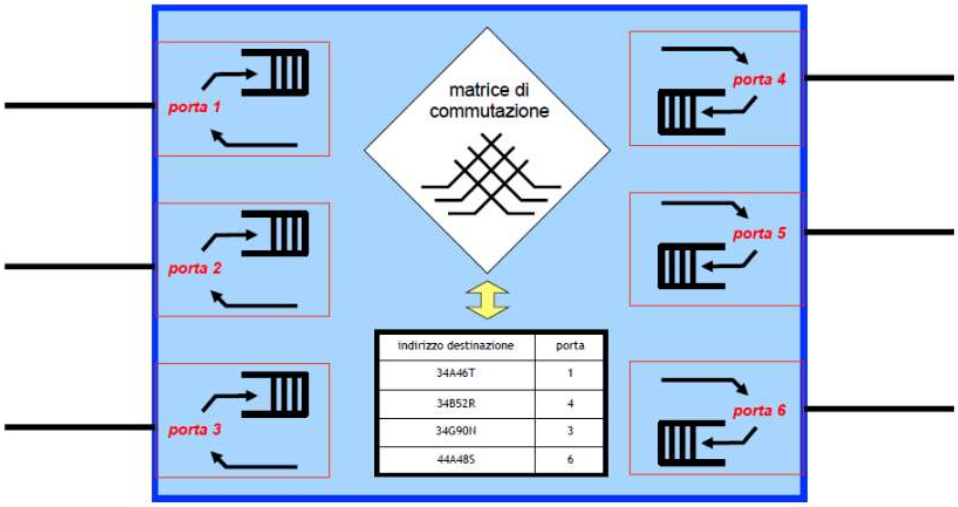
BRIDGE

Apparato dotato di intelligenza, collega due segmenti di rete. Seleziona se ripetere una trama generata da un segmento di rete sull'altro segmento in base ad una tabella che esso mantiene. Così facendo spezza il dominio di collisione.



SWITCH

Lo switch è un bridge multiporta, che mantiene una tabella in cui vi sono associati indirizzi di livello 2 e segmenti di rete di appartenenza. Se ogni porta dello switch è connessa ad un'unica stazione, allora realizza un accesso dedicato per ogni nodo, eliminando le collisioni e aumentando dunque la capacità. Supporta inoltre conversazioni multiple contemporanee.



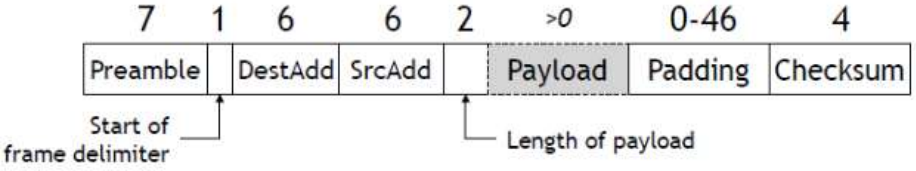
PROTOCOLLI DI LIVELLO 2

Esistono diversi protocolli di livello 2 e ogni hop può avere un protocollo che può essere differente dall'hop successivo. L'elemento unificato è il protocollo di livello 3, che ha visibilità end-to-end.

ETHERNET E STANDARD IEEE 802.3

Utilizzato nelle reti locali (LAN), è una tecnologia economica che si interfaccia direttamente e gestisce il livello fisico. Supporta un carico medio del 30% fino a picchi del 60%. Sotto carico medio il 2-3% dei pacchetti ha una sola collisione, meno dello 0,001% ha più di una collisione.

IEE 802.3 definisce un'intera famiglia di sistemi CSMA/CD con velocità 1-10Mbps, mente Ethernet è solamente 10Mbps. Entrambi implementano un livello MAC di tipo ***CSMA/CD 1-persistent*** e in caso di collisione viene calcolato l'instante in cui ritrasmettere con un algoritmo di ***binary exponential backoff*** (dopo *i* collisioni, il tempo di attesa è casuale nell'intervallo [0,1, …,2^i-1].

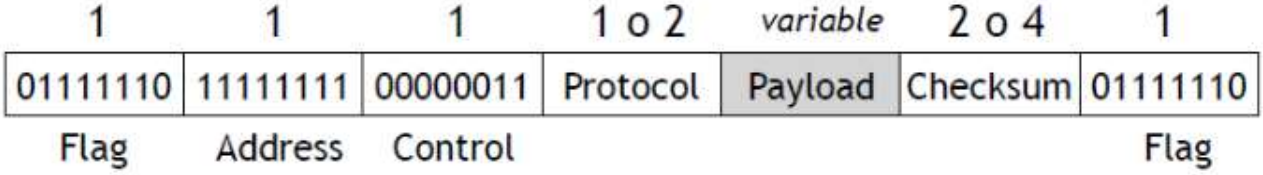


La trama è composta da:

- PREAMBOLO: sequenza di "10101010" per sincronizzare il ricevitore;
- START OF FRAME: flag di inizio della trama "10101011";
- ADDRESSES: indirizzi destinazione e sorgente della trama;
- LENGTH: lunghezza in byte della trama;
- PAYLOAD: informazione trasmessa;
- CHECKSUM: codice per rilevazione errori.

PPP

Protocollo utilizzato sia nell'accesso che nel backbone.



- FLAG: identifica inizio e fine della trama;
- ADDRESS: se ha tutti 1 significa "tutti gli host";
- CONTROL: di default a "00000011";
- PROTOCOL: identifica il livello di frame;
- PAYLOAD: informazione trasmessa;
- CHECKSUM: identificazione dell'errore.

PPP con accesso modem viene utilizzato dalla banda telefonica per inviare i segnali, con limite di 56Kbps. PPP con accesso xDSL (*Digital Subcriber Line*) permette di utilizzare la

banda disponibile del doppino telefonico; si possono distinguere in sistemi simmetrici e asimmetrici.

ARP (*Adress Resolution Protocol*)

Per il forwarding è necessario eseguire una "traduzione", infatti il livello IP deve tradurre l'indirizzo IP del next-hop, che non è altro che un astrazione a livello software, nel corrispondente indirizzo MAC.

Un host può risolvere l'indirizzo di un altro host solo se entrambi sono connessi alla medesima rete fisica. In caso contrario allora si provvederà ad una consegna indiretta, cioè si invia il pacchetto al router che provvederà all'invio verso la rete di destinazione. Lo standard ARP è generale e specifica i diversi messaggi a seconda dei protocolli coinvolti. Esiste infatti un campo che specifica la dimensione dell'indirizzo di livello rete.

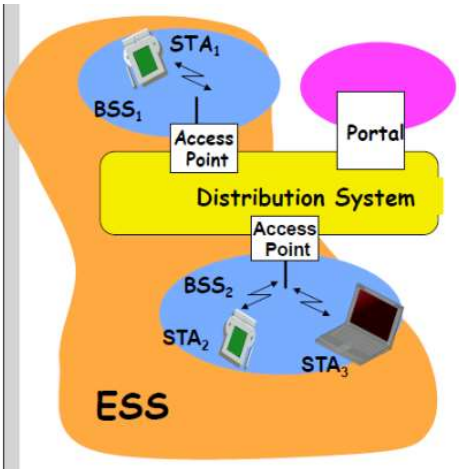
Formato dei messaggi:

- HARDWARE ADRESS TYPE: specifica il tipo di indirizzo hardware;
- PROTOCOL ADDRESS TYPE: specifica il tipo di indirizzo del protocollo;
- HADDR LEN: dimensione in byte dell'indirizzo hardware;
- PADDR LEN: dimensione in byte dell'indirizzo del protocollo;
- OPERATION: specifica se il messaggio è una "Request" oppure "Response";
- SENDER HADDR: indirizzo sorgente hardware;
- SENDER PADDR: indirizzo protocollo sorgente;
- TARGET HADDR: indirizzo destinazione hardware;
- TARGET PADDR: indirizzo protocollo destinazione.

Il messaggio ARP viene considerato come dei dati trasportati dal livello 2, perciò il livello di rete non ne fa il processing. Nell'header della trama esiste un campo "TYPE", per Ethernet 0x806 = ARP. Per ridurre il traffico di rete il software ARP estrae e salva le informazioni delle risposte ARP in modo da poterle utilizzare in futuro. Gestisce la tabella come una cache, rimuovendo le informazioni più vecchie. Prima di inviare una richiesta ARP controlla se esiste in cache tale informazione. Se l'informazione non è presente allora invia in broadcast la richiesta, aspettando la risposta, aggiornando la cache per poi infine inviare la trama.

RETI 802.11

- STA: terminale con capacità di accesso al mezzo wireless;
- BSS: insieme di terminali che usano le stesse frequenze;
- Access Point: stazione integrata sia nella WLAN che nel "Distribution System";
- Portal: bridge verso altre reti;
- Distribution System: rete di interconnessione per formare un'unica rete logica.



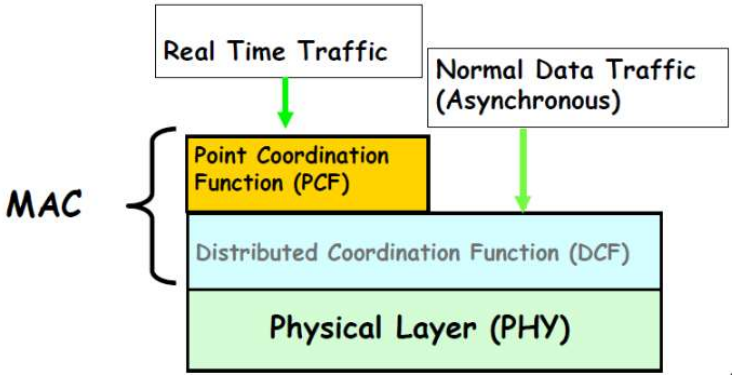
Il Basic Service Set (BSS) è formato da un insieme di terminali con lo stesso protocollo MAC che competono per l'accesso al mezzo condiviso e può essere collegato ad un AP che funziona come bridge. Il protocollo MAC può essere distribuito completamente oppure controllato da un'entità centrale che fa da coordinatore come un AP.
Per Extended Service Set (ESS) si intende due o più BSS interconnesse da un Distribution System.

Principali standard 802.11:

- 802.11a: 5 GHz
- 802.11b: 2.4GHz
- 802.11i: autenticazione e sicurezza

LIVELLO MAC

Non si utilizza lo stesso protocollo CSMA anche nelle reti wireless perché porta a delle collisioni in un mezzo non cablato. Viene utilizzato perciò CSMA-CA.



DCF

Utilizza un algoritmo per la risoluzione delle contese per fornire l'accesso a tutti i tipi di traffico. Il traffico ordinario si appoggia direttamente su DCF.

PCF

Supporta il traffico Real-Time, basato su "polling". Utilizza un algoritmo MAC gestito a livello centralizzato e fornisce un servizio senza contesa del canale. PCF sfrutta le funzionalità DCF per fornire l'accesso agli utenti. Entrambi possono funzionare allo stesso tempo all'interno della stessa BSS fornendo alternativamente periodi con contesa e senza contesa.

TIME SLOT

Il tempo viene suddiviso in intervalli chiamati "slot". Sono molto più piccoli del tempo di trama e variano a seconda della velocità di trasmissione.

IFS (*Inter-Frame Space)*

Intervallo di tempo tra la trasmissione di trame, usato per stabilire livelli di priorità nell'accedere al canale.

Sono stati definiti 4 tipi IFS:

1) **SIFS (*Short IFS*)**

   Usato per separare la trasmissione di trame appartenenti allo stesso "dialogo", corrisponde alla più alta priorità, usato perciò per l'invio di pacchetti corrispondenti ad una risposta immediate (ACK, CTS, risposte a polling).

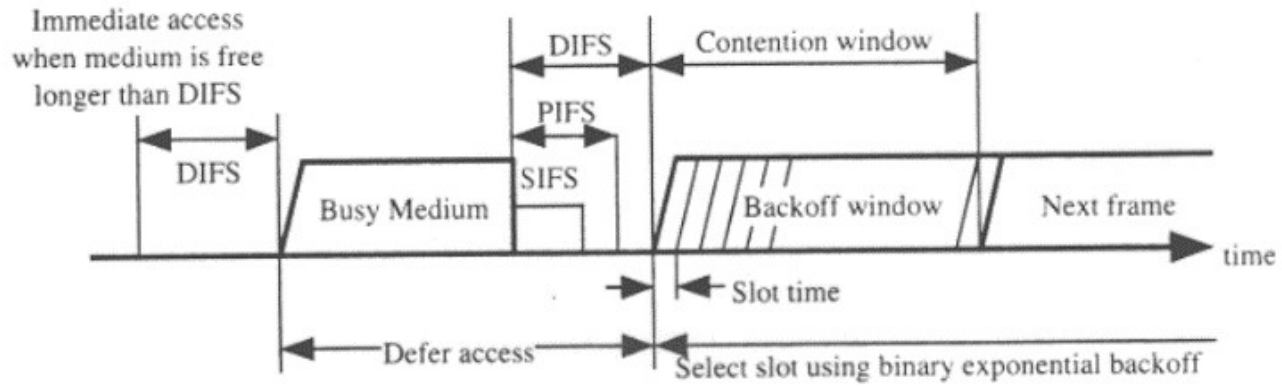
2) **PIFS (*Point Coordination IFS*)**

   Priorità media per servizi real-time che usano PCF. Usato durante il polling dal controlle centrale nello schema PCF.

3) **DIFS (*Distributed IFS*)**

   Priorità più bassa, per il servizio di invio dati asincrono. Usato dalle trame per l'invio asincrono con ritardo minimo nel caso di contesa del canale.

4) **EIFS (*Extended IFS*)**



DCF CON CSMA/CA

Una stazione con dei dati da trasmettere ascolta il canale: se il canale è libero, allora continua ad ascoltare per capire se il mezzo rimane libero per un tempo pari a DIFS. Se il canale è occupato la stazione continua a monitorare il mezzo fino a quando la trasmissione corrente non finisce.

Se il mezzo rimane libero per un intervallo DIFS la stazione estrae un numero casuale di slot uniformemente distribuito all'interno di una Contention Window e fintantochè il canale rimane libero, la stazione decrementa il contatore di backoff man mano che il tempo passa. Se il contatore arriva a 0, allora la stazione trasmette. Se il canale invece torna ad essere occupato, il contatore verrà congelato e si ricomincerà la procedura di attesa.

Il **backoff** è un numero intero che corrisponde al numero di time-slot. CW è il valore della **Contention Window**, che viene aggiornato ad ogni tentativo di trasmissione.

In caso di collisione, viene raddoppiata la CW_{max} e la lunghezza del tempo di backoff viene aumentata esponenzialmente nel caso di ritrasmissioni multiple. Si seleziona un nuovo valore della CW tra 0 e CW_{max} e si attende lo scadere del backoff.

La stazione ricevente manda un ACK immediatamente dopo la ricezione di una trama (SIFS) senza quindi ascoltare il mezzo.

DCF CON RTS/CTS

Il segnale generato dalle stazioni è percepibile solo fino ad una certa distanza, che dipende dalla potenza di emissione del segnale. Quando il segnale è troppo debole, non è possibile ricostruirlo.

In particolari occasioni il segnale emesso da una stazione può essere percepito solo da un sottoinsieme di altre stazioni, creando il cosiddetto terminale nascosto. Per risolvere questo problema la sorgente invia una trama RTS (*Request To Send*) dopo aver percepito il canale libero per un intervallo pari a DIFS. Il ricevente risponde con una trama CTS (*Clear To Send*) dopo un intervallo SIFS. RTS/CTS vengono utilizzati per riservare il canale per la trasmissione dei dati, in modo tale che le eventuali collisioni possano avvenire solo tra i messaggi di controllo. Un'altra eventuale stazione che riceve CTS sa quanto aspettare prima di trasmettere poiché sia su RTS che poi su CTS viene specificata la durata della trasmissione.

NAV (*Network Allocation Vector*)

In 802.11 l'ascolto del canale è sia fisico che virtuale, se una delle due funzionalità indica che il canale è occupato allora si considera il canale occupato. L'ascolto virtuale del canale è fornito dal NAV.

La maggior parte delle trame 802.11 infatti includono il campo di lunghezza della trama. I nodi che percepiscono le trame impostano il NAV al tempo in cui si aspettano che il mezzo sia libero. Se NAV è maggiore di 0, allora il mezzo è considerato occupato.