Capitolo 1

1.0 Introduzione

Il primo argomento da introdurre quando si parla di reti di calcolatori è la differenza tra essi e i sistemi distribuiti.

Un ***sistema distribuito*** è un insieme di computer indipendenti che appare ai propri utenti come un singolo sistema coerente. Un classico esempio di sistema distribuito è il World Wide Web.

In una ***rete di calcolatori*** mancano la coerenza, il modello ed il software tipico di un sistema distribuito. Gli utenti della rete vedono i singoli componenti di essi, un utente difatti per eseguire un programma che si trova su una specifica macchina remota della rete deve collegarsi manualmente ad essa per eseguirlo.

1.1 Applicazione delle rete di calcolatori

Le reti di calcolatori hanno una notevole applicazione nella vita quotidiana, difatti hanno le seguenti applicazioni:

- Applicazioni aziendali: le aziende odierne dispongono di un notevole numero di computer e la necessità di un'azienda è quella di condividere tra essi sia le risorse fisiche (come una stampante) che non (come le informazioni). Una delle esigenze delle grandi aziende è quella di disporre di una ***VPN*** (virtual private network), una rete che permette a più reti di un'azienda sparse sul globo di essere interconnesse e quindi di dare la possibilità ad un dipendente di operare su dati o programmi che si trovano su una macchina distante 15.000 km da esso. In genere nelle rete aziendali si sfrutta l'***architettura client-server***, quindi un database che fornisce i servizi (server) ai vari dipendenti dell'azienda (client). Nello specifico il modello client-server comprende richieste e risposte, un processo client manda un messaggio contente le richieste attraverso la rete al processo server, il quale, una volta ricevuta la richiesta ed elaborata restituisce la risposta. Un ulteriore uso delle reti di calcolatori nell'ambito aziendale è quello delle chiamate ***VoIP***, effettuate sfruttando la rete ed evitando quindi all'azienda spese inutili derivate dagli operatori telefonici. Infine, uno degli ultimi utilizzi, diffusosi molto negli ultimi anni, sono gli ***e-commerce***, quindi la vendita online dei servizi offerti dalle aziende.
- Applicazioni domestiche: negli ultimi anni internet è entrato a far parte della vita quotidiana di ognuno di noi, difatti come disse Bob Metcalfe, l'inventore dei Ethernet, il valore di una rete è proporzionale al quadrato del numero dei suoi utenti, questa ipotesi, nota come ***legge di Metcalfe***, potrebbe spiegare l'immenso successo di internet. I servizi offerti da internet agli utenti riguardano i più svariati campi, come le belle arti, affari, cucina, politica, salute, storia, hobby e molto altro. La maggior parte di essi sono erogati mediante l'architettura client-server, quindi un server che li eroga e migliaia di client che ne usufruiscono, ma esiste un altro popolare modello di accesso alla informazioni che prende il nome di ***peer-to-peer***. In questa forma di comunicazione, gli utenti che costituiscono la rete sono in grado di comunicare tra di loro, senza una divisione di categorie client-server, infatti proprio come dice il nomi "pari-a-pari", ogni utente nella rete è uguale ad un altro. Molti sistemi peer-to-peer non dispongono nemmeno di un database centrale e le informazioni sono divise tra i vari utenti della rete.

Questo tipo di comunicazione viene spesso usata per la condivisione di musica e video (vedi Napster, chiuso perché divenne la più grande violazione di copyright nella storia dell'industria discografica). Altri usi di tale rete sono dati dall'utilizzo di servizi di messaggistica istantanea (Whatsapp), da servizi di condivisione audio e video (YouTube) e da servizi che si trovano a metà strada tra i servizi di messaggistica istantanea e quella di condivisione dati, i social network (Facebook).

- Utenti mobili: la vendita di tali dispositivi (laptop, smartphone) ha superato la vendita dei dei computer fissi, tutto ciò perché ormai quasi tutti gli utenti hanno la necessità di effettuare le operazioni che farebbero da casa/ufficio anche in mobilità, ciò che ha permesso tutto ciò è stato internet. Usi tipici sono l'utilizzo dello smartphone, dotato di GPS, per l'utilizzo delle mappe, o l'utilizzo sempre dello smartphone, se dotato di chip NFC, per effettuare pagamenti senza l'utilizzo di contanti o carte.

1.2 Hardware di rete

Le reti di calcolatori dispongono di due caratteristiche principali: la tecnologia di trasmissione e la scala.

Le tecnologie di trasmissione possiamo dividerle in due tipi:

1. Trasmissione ***punto a punto***: dove i collegamenti sono tra coppie di computer, in cui i pacchetti, per essere trasmessi tra un computer ed un altro, spesso devono attraversare più macchine intermedie per arrivare a destinazione.

2. Trasmissione ***broadcast***: sono dotate di un unico canale di comunicazione, il pacchetto viene inviato da una macchina a tutte le altre macchine, esse alla ricezione esamineranno se è destinato a loro, in tal caso lo elaboreranno, altrimenti verrà semplicemente ignorato. In alcuni sistemi broadcast supportano la trasmissione a un sottoinsieme delle macchine, tale tecnologia viene chiamata ***multicast***.

Infine, la connessione di due o più reti è chiamata ***internetwork***.

La scala di dimensione delle reti possiamo dividerle in:

- Le reti ***PAN*** (personal area network, reti personali): esse permettono ai dispositivi di comunicare nello spazio fisico alla portata di una persona. Un esempio classico è dato da un computer collegato alle sue periferiche (mouse, tastiera, stampante, ecc.). Tali collegamenti, in assenza di una rete wireless, devono essere effettuati tramite cavi. Per ovviare al problema precedente è stato introdotta una rete wireless a corto raggio, il ***Bluetooth.*** Altri tipi di rete Pan sono dati da pacemaker, dispensatori di insulina che possono essere connessi ad un sistema di controllo.

- Le reti ***LAN*** (local area network, reti locali): si tratta di reti che operano all'interno o nelle vicinanze di un singolo edificio come un appartamento, un ufficio o una fabbrica. Generalmente sono usate per connettere dispositivi a risorse comuni (stampanti) e per condividere informazioni. Le reti LAN possono essere anche di tipo wireless e vengono usate in edifici dove sarebbe difficile l'utilizzo di cavi. In questo caso viene usato un dispositivo, chiamato router, che distribuisce i pacchetti collegando in maniera wireless i dispositivi. Lo standard per le reti LAN wireless è ***IEEE 802.11***, noto come ***Wi-Fi***. Nonostante la versatilità delle reti LAN wireless esse

rimangono comunque meno stabili e precise rispetto le reti LAN cablate, avendo una percentuale di errore nella trasmissione dei dati più alta. La topologia di molte LAN cablate è la connessione punto a punto e sfrutta il protocollo ***IEE 802.3***, comunemente chiamato ***Ethernet***. Ogni dispositivo di tale rete capisce il protocollo Ethernet e si connette ad un dispositivo chiamato switch, reti cablate più grandi possono essere composte da più switch collegati tra di loro. Le reti broadcast sia cablate che wireless possono dividersi ulteriormente in:

  - ***Reti statiche***: la connessione viene divisa in intervalli e tramite un algoritmo viene assegnato a ciascuna macchina il proprio intervallo in modo tale che essa accetti comunicazioni solo in quell'intervallo stabilito. Tale allocazione spreca la capacità del canale, perché se la macchina nel suo intervallo non ha niente da trasmettere quell'intervallo sarà sprecato.
  - ***Reti dinamiche***: esse si dividono in reti dinamiche centralizzate, le quali hanno una stazione base che mediante un algoritmo decide quale macchina può trasmettere o ricevere dati, e reti dinamiche non centralizzate le quali decidono autonomamente quando ricevere o trasmettere dati.

- Le reti ***MAN*** (metropolitan area network, rete metropolitana): esse permettono di tenere interconnessa un'intera città. Un esempio di rete metropolitana è la rete TV, in città che non dispongono di una buona connessione via etere le trasmissione televisive vengono effettuate mediante enormi antenne che tramettono il segnale ad intere città. Col passare degli anni sempre meno città hanno avuto bisogno di questa tecnologia quindi questo tipo di trasmissione è stata modificata ed utilizzata per la trasmissione di programmi personalizzati, come notiziari specifici per città, o servizi meteo specifici.
- Le reti ***WAN*** (wide area network, rete geografica): le reti WAN sono rete geograficamente estese, esse spesso coprono intere nazioni o continenti. I computer sparsi nel globo che fanno parte di questa rete sono chiamati host. La sottorete della WAN che connette gli host tra di loro si chiama ***subnet*** ed è formata da due componenti: linee di trasmissione ed elementi di commutazione. Le linee di trasmissione (nel caso di WAN cablate) sono i collegamenti tra i vari host (in rame, fibra ottica, ecc.). Gli elementi di commutazione, generalmente chiamati ***router***, sono dispositivi che una volta ricevuto il segnale lo indirizzano al dispositivo che l'ha richiesto. Un'azienda può però decidere di non crearsi una propria rete ma bensì di noleggiare una già esistente, creando così una ***VPN*** (virtual private network). Noleggiando così da un operatore che prenderà il nome di ***ISP*** (internet service provider). La strategia adottata dalla rete per decidere dove inoltrare il messaggio si chiama ***algoritmo di instradamento*** (o algoritmo di routing), mentre la strategia adottata localmente da ogni router per inoltrare un pacchetto è chiamata ***algoritmo di inoltro*** (o algoritmo di forwarding).

Gli utenti collegati alla rete spesso hanno l'esigenza di comunicare non solo con gli utenti della proprio rete, l'unione quindi di più reti si chiama ***internetwork*** o ***internet***.

1.3 Software di rete

Per diminuire la complessità dell'organizzazione e della gestione di una rete essa è divisa in più livelli e ogni livello offre dei servizi a quello superiore. Il livello più basso è più vicino all'infrastruttura di rete, il livello più in alto è più vicino all'utente finale. Quando il livello *n* di un computer è in comunicazione con il livello *n* di un altro computer la comunicazione viene regolata mediante i ***protocolli***, essi fanno sì che entrambi i livelli utilizzino gli stessi standard di comunicazione e che possano essere in grado di comunicare. L'insieme dei protocolli usati dal sistema viene chiamato ***pila di protocolli***.

In realtà il livello *n* di un computer non comunica direttamente con il livello *n* di un altro, bensì comunica i propri dati al livello sottostante fino a raggiungere il più basso, il livello fisico, attraverso cui si scambiano effettivamente i dati tra i due computer.

L'insieme di livelli e protocolli si chiama ***architettura di rete***.

Alcuni problemi fondamentali nella progettazione delle reti si presentano livello dopo livello.

Uno dei problemi più noti è quello della ricezione di un messaggio sbagliato o parzialmente corretto, in questo caso intervengono gli ***error detection***, ovvero software che riconoscono l'errore e richiedono il messaggio finché non ricevono quello corretto, e gli ***error correction***, software che ricostruiscono il messaggio con i possibili messaggi corretti. Ai bassi livelli si controlla quindi che i pacchetti ricevuti siano corretti, agli alti livelli che il messaggio finale sia corretto.

Altri problemi possono essere come trovare un percorso valido è affidabile su cui trasmettere il messaggio o quelli di utilizzare software aggiornati che siano al passo coi tempi.

Un ottima rete è tale quando si può definire ***scalabile***, ovvero quando all'ingrandimento della rete essa continua a funzionare bene.

Le ***primitive di servizio*** sono delle operazioni base che servono per stabilire una connessione. Un esempio di implementazione di un servizio orientato alla connessione prevede sei primitive:

1. Listen: il server con questa primitiva indica che è pronto ad accettare connessioni.
2. Connect: il client stabilisce una connessione con il server.
3. Accept: il server accetta la richiesta di connessione da parte del client.
4. Receive: il server si mette in attesa quindi del primo messaggio da parte del client.
5. Send: il client invia la richiesta al server, e si imposta su Receive per ricevere la risposta da parte del server.
6. Disconnect: se il client ha finito si disconnette e quindi termina la connessione.

I ***servizi*** sono quindi l'insieme di primitive (operazioni) che un livello offre a quello superiore, ma non dice nulla di come queste operazioni sono implementate. Quindi un servizio è correlato all'interfaccia tra due livelli, quello inferiore è il provider, quello superiore è l'utente.

1.4 Modelli di riferimento

Esistono due importanti architetture di rete, il modelli di riferimento ***OSI*** e il modello ***TCP/IP***. I protocolli associati al modello OSI sono armai in disuso ma sono ancora validi, al contrario, il modello TCP/IP è poco utilizzabile, ma i suoi protocolli sono largamente utilizzati.

Il ***modello di riferimento ISO-OSI*** prende tale nome perché è stato sviluppato dall'international standard organization come primo passo verso la standardizzazione dei protocolli usati nei diversi livelli, mentre OSI (open system interconnection) perché riguarda la connessione di sistemi aperti, ovvero sistemi che sono aperti verso la comunicazione con altri. Tale modello ha sette livelli ed essi sono sette perché:

- quando è richiesta un'astrazione diversa si deve creare un nuovo livello;
- ogni livello deve avere funzioni ben definite;
- ogni livello deve definire standard per protocolli a livello internazionale;
- i confini dei livelli vanno definiti in modo da minimizzare il flusso dei dati attraverso i livelli;
- il numero dei livelli deve essere abbastanza per raggruppare tutti i servizi simili nello stesso livello ma abbastanza piccolo per rendere l'architettura gestibile.

I sette livelli sono i seguenti:

1. ***Livello fisico***: si occupa della trasmissione dei bit sul canale di comunicazione. Le specifiche riguardano interfacce meccaniche o elettriche oltre al mezzo di trasmissione che si trova sotto il livello fisico. Problemi tipici di questo livello sono i tempi di trasmissione del segnale, come avviene l'inizio e la fine della comunicazione o se la comunicazione può avvenire simultaneamente in entrambe le direzioni.
2. ***Livello data link***: si occupa di far sì che la trasmissione che avviene al livello fisica sia priva di errori o che appaia priva di errori al livello rete. Per far ciò i dati trasmessi vengono suddivisi in frame, se il servizio è affidabile, il ricevente conferma la corretta ricezione di ciascun frame restituendo un ***frame di acknowledgment***. Un altro problema avviene se il trasmittente è veloce e finisce per saturare il buffer, per evitare ciò occorre un meccanismo che regoli il traffico. Nelle reti broadcast esiste anche il problema di come controllare l'accesso al canale condiviso, in tali reti esiste un sottoinsieme del livello data link, chiamato medium access control (controllo di accesso al mezzo trasmissivo), che regola il tutto.
3. ***Livello rete***: si occupa della modalità di trasmissione con cui pacchetti sono inoltrati dalla sorgente alla destinazione. Difatti una trasmissione può essere statica e quindi sfruttare una sola rete oppure dinamica e quindi sfruttare più reti per distribuire il carico della trasmissione. I problemi a questo livello possono essere molteplici, la rete può rifiutare il pacchetto perché troppo grosso, problemi di compatibilità dovuti ai protocolli e molti altri. Nelle reti broadcast il problema dell'instradamento è facile, infatti in queste reti il livello rete è ridotto o addirittura assente.
4. ***Livello trasporto***: il compito di questo livello è quello di trasmettere i dati ricevuti dai livelli superiori al livello rete e assicurarsi che arrivino a destinazione. Il tutto va eseguito in modo tale da isolare i livelli superiori da eventuali cambiamenti hardware dei livelli inferiori. Tale

livello definisce inoltre il tipo di servizi da offrire al livello sessione, difatti la trasmissione può essere di diverso tipologie, tra cui:

1. Trasmissione punto a punto, connessione priva di errori di trasmissione poiché i messaggi viaggiano nello stesso ordine usato per la trasmissione.
2. Oppure trasmissioni che trasmettono i vari pacchetti senza la garanzia che i pacchetti arrivino nell'ordine giusto.

5. ***Livello sessione***: permette a utenti su diversi computer di stabile una sessione. Offrendo servizi di:
   - controllo dialogo: tenere traccia di quando è il turno di trasmettere o ricevere.
   - gestione dei token: evitare che le due parti tentino la stessa operazione critica simultaneamente.
   - sincronizzazione: gestione di lunghe trasmissione e permettere di riprendere la trasmissione dal punto in cui si è fermata a causa di crash.
6. ***Livello presentazione***: esso gestisce la sintassi e la semantica dell'informazione trasmessa per consentire comunicazioni tra computer con differenti rappresentazione dei dati.
7. ***Livello applicazione***: comprende l'insieme dei protocolli generalmente richiesti dagli utenti, come ad esempio il protocollo HTTP (hyper text transfer protocol), protocollo usato quando l'utente richiede una pagina web, oppure protocolli usati per il trasferimento dei dati, la posta o le notizie.

Il ***modello di riferimento TCP/IP*** prende il nome dai suoi due protocolli principali e nasce come necessità di collegare più reti tra di loro in modo semplice e che nel caso una delle infrastrutture fosse distrutta la comunicazione continuerebbe ad esistere. Esso infatti prevede sono quattro livelli e sono:

1. ***Livello Link***: a questo livello viene scelta una rete a commutazione di pacchetto, quindi l'informazione viene suddivisa in piccoli pacchetti e inviata mediante diversi reti. In questo livello si descrive come devono avvenire i vari collegamenti.
2. ***Livello internet***: il compito di questo livello è permettere agli utenti di inviare pacchetti su qualsiasi rete e far sì che viaggino in modo indipendente, potendo quindi arrivare al destinatario in ordine sparso, in questo caso sarà compito dei livelli superiore riordinare il tutto. Tale livello definisce un formato ufficiale per i pacchetti e un protocollo chiamato ***IP*** (internet protocol). I problemi principali di questo livello sono dunque l'instradamento dei pacchetti e la gestione dei pacchetti in caso di congestione della rete.
3. ***Livello trasporto***: tale livello, proprio come nel modello OSI, è progettato per consentire la comunicazione tra gli host della rete. A questo livello sono indicati due protocolli:
   1. ***TCP*** (transfer control protocol): è un protocollo affidabile che garantisce una comunicazione prova di errori. Per far ciò i dati trasmessi vengono suddivisi in frame e il ricevente conferma la corretta ricezione di ciascun frame restituendo un ***frame di acknowledgment***.

2. ***UDP*** (user datagram protocol): è un protocollo inaffidabile, non adatto ad applicazioni che vogliono garanzia di ordinamento e di controllo del flusso di TCP ma che preferiscono un protocollo veloce (ad esempio le chiamate VoIp, che non possono rimanere in attesa finché non arriva il pacchetto corretto, altrimenti la chiamata sarebbe piena di buchi).

4. ***Livello applicazione***: il modello TCP/IP non ha i livelli sessione e presentazione, dato che il modello OSI ha spesso dimostrato che questi livelli servivano a poco. Esso inizialmente aveva pochi protocolli come TELNET che gestiva terminali virtuali, FTP per lo scambio dei file e SMTP per la posta.

Criticità del modello e dei protocolli OSI:

- Scarso tempismo: i protocolli del modello OSI furono rilasciati largamente in ritardo, molto tempo dopo quelli del modello TCP/IP, i quali furono già utilizzati a livello globale.
- Tecnologia scadente: i 7 livelli furono concepiti più per politica che per tecnica, il livello sessione ed il livello presentazione sono quasi inutilizzati e i livelli data link e rete sono sovraccarichi. Inoltre la definizione dei servizi e dei protocolli del modello OSI hanno una complessità straordinaria e sono quasi incomprensibili.
- Implementazioni scadenti: data la complessità del modello le prime implementazioni furono enormi, scomode e lente, quindi ben presto il modello OSI fu associato a scarsa qualità, mentre una delle prime applicazioni del modello TCP/IP faceva parte di UNIX.
- Incapacità politica: il modello OSI fu da prima identificato come prodotto della comunità europea e successivamente come prodotto americano, di conseguenza nessuno lo voleva come standard.

Criticità del modello e dei protocolli TCP/IP:

- il modello non distingue bene i concetti di servizio, interfaccia e protocollo, risultando inutile per la progettazione di reti basate su tecnologie nuove.
- il modello è poco generico e inadatto a descrivere protocolli diversi dal TCP/IP, come ad esempio il Bluethoot.
- il livello link non è un vero e proprio livello ma piuttosto un'interfaccia tra il livello rete e link.
- non fa distinzione tra il livello fisico (non lo menziona affatto) ed il livello link.
- infine, anche se i protocolli TCP e IP sono stati progettati e concepiti con attenzione, i restanti protocolli sono stati spesso realizzati da dottorandi e rilasciati quindi gratuitamente diventando così standard nonostante la loro poco utilità.

1.5 Esempi di reti

Esistono molti tipi di rete, piccole e grandi, con diversi obiettivi, scale e tecnologie, e sono le seguenti:

- ***Internet***: è una vasta raccolta di reti diverse che usano determinati protocolli e offrono certi servizi. Esso non ha un progettista e non è controllato da nessuno.
- ***ARPANET***: nasce come commissione della difesa degli USA, durante la guerra fredda, come una rete capace di sopravvivere a una guerra nucleare. Tale commissione deriva dal fatto che fino a quel momento le reti sfruttavano la rete telefonica pubblica, la quale, una volta distrutte alcune infrastrutture superiori avrebbe smesso di funzionare.
- ***NSFNET***: tale rete nasce come risposta all'esigenza del fatto che molte università per comunicare tra di loro usavano ARPANET, poiché per usare tale rete bisogna realizzare un contratto con la difesa molte università non vollero usarla, fu così che la NSF realizzo la NSFNET, una rete capace di collegarsi ad ARPANET senza alcun vincolo.
- ***LAN wireless (802.11)***: tale protocollo nasce dall'esigenza di standardizzare le reti wireless, poiché prima di esso ogni produttore usava il proprio protocollo creando quindi una serie di reti incompatibili tra di loro. Quindi, il comitato IEEE, che precedentemente aveva regolato le reti LAN, introdusse le reti LAN wireless con la numerazione 802.11 (da 802.1 a 802.10 erano già occupate), anche se tale standard è maggiormente noto come ***WiFi***.
- ***RFID***: le etichette RFID a differenza dei dispositivi analizzati finora (computer, cellulari) sono dispositivi estremamente piccoli, più piccoli di un francobollo, utilizzati per tracciare o identificare oggetti, come documenti, animali domestici e molto altro. Queste etichette vengono lette dai lettori RFID che una volta a contatto con le etichette ne estraggono le istruzioni. Le etichette le possiamo dividere in due categorie:
  - RFID passiva: sono etichette che non necessitano di una batteria o una fonte di energia e per funzionare gli basta l'energia fornita sotto forma di onde radia da parte del lettore.
  - RFID attiva: a differenza dell'RFID passiva, per funzionare necessita di una fonte di energia.

1.6 Standardizzazione delle reti

Gli standard definiscono le caratteristiche che deve avere un prodotto per essere interpretabile. Raggiungere l'interoperabilità non è una cosa semplice, ne è testimone lo standard 802.11 che per risolvere tutti i problemi si è creato un gruppo commerciale, chiamato ***WiFi Alliance***.
Gli standard si dividono in due categorie:

- ***de facto*** (di fatto): sono standard affermati sul campo, senza una pianificazione formale, ne è l'esempio il protocollo HTTP.
- ***de jure*** (per legge): sono standard formali, adottati da qualche organismo di standardizzazione autorizzato.

Gli organismi di standardizzazione si dividono a loro volta in organismi adottati dai governi e da organizzazioni volontarie.
Alcune organizzazioni:

- ***ITU (international telecommunication union)***: istituto voluto da molti governi europei per la standardizzazione della telefonia, nel 1947 divenne un'agenzia delle Nazioni Unite. Esso si divide in:
  - ***ITU-T***: si occupa della standardizzazione delle telecomunicazioni.
  - ***ITU-R***: si occupa della standardizzazione delle radiocomunicazioni.
  - ***ITU-D***: si occupa di promuovere lo sviluppo du tecnologie di informazione e comunicazione terrestre.
- ***ISO (international standard organization)***: organizzazione libera fondata da volontari nel 1946. I sui membri includono gli USA, la Gran Bretagna, la Francia, la Germania e molti altri paesi, oltre 150. Esso ha emanato oltre 17.000 standard e oltre 200 comitati di tecnici che si sono occupati di numerosi argomenti: dadi, bulloni, rivestimento dei pali telefonici, reti da pesca e molti altri.
- ***NIST (national institute of standards and technology)***: esso da parte del dipartimento del commercio degli USA ed emette standard obbligatori per gli acquisti di materiali da parte del governo.
- ***IEEE (institute of electrical and electronics engineers)***: esso sviluppa standard nel campo dell'ingegneria elettrica e dei computer. Il comitato IEEE 802 ha standardizzato molti tipi di LAN, come ad esempio:
  - 802.11: Wireless LAN.
  - 802.15: Personal area network (Bluetooth).

Capitolo 2

2.2 Mezzi di trasmissione vincolati

Lo scopo del livello fisico è trasportare un flusso grezzo di bit da una macchina all'altra. Per farlo si usano diversi mezzi, divisi in mezzi vincolati (cavi in rame, fibre ottiche) e mezzi non vincolati (onde radio, laser).
Ecco alcuni esempi di mezzi vincolati:

- ***Supporti magnetici***: è uno dei mezzi più comuni per il trasferimento dei dati, consiste nel scrivere i dati su supporti fisici (DVD, hard disk) e spostarli fisicamente.
- ***Doppino***: rappresenta uno dei mezzi di trasmissione più vecchi ma è ancora largamente usato, esso è composto da due conduttori di rame isolati, spessi circa 1 mm, avvolti uno intorno all'altro in una forma elicoidale, poiché due cavi paralleli formano un'eccellente antenna e creerebbero interferenza. Esistono diverse varietà di doppini, quelli generalmente diffusi sono di categoria 5, ovvero due cavi elettricamente isolati e intrecciati tra di loro, quattro di queste coppie sono raccolte all'interno di una guaina di plastica per proteggerli e mantenerli uniti. In alcuni casi vengono usati anche quelli di categoria 6 e 7, che possono raggiungere i 10 Gbps e dispongono di maggiore isolamento e garantiscono meno distorsioni. Esso è capace di trasmettere segnali per alcuni chilometri, anche se oltre il segnale necessita di ripetitori.
- ***Cavo coassiale***: presentando maggiore schermatura può estendersi a distanze maggiori rispetto al doppino. Esso è composto da un'anima di rame, ricoperta da una guaina isolante, che a sua volta è ricoperta da una calza di conduttori sottili che infine sono avvolti da una guaina protettiva di plastica. Essendo maggiormente schermato permette anche velocità più alte.
- ***Linee elettriche***: le linee elettriche sono state utilizzate in passato per le trasmissione a basso tasso di invio (basso bit rate). L'idea nasce dalla semplicità di collegare i dispositivi alla rete elettrica e basta, senza doverli collegare a nessun'altra presa. Quest'idea è stata poco presa in considerazione dato il fatto che le linee elettriche non sono state concepite per trasmettere dati e per questo presentano cattive prestazioni.
- ***Fibre ottiche***: Un sistema di trasmissione ottico è formato da tre componenti fondamentali: la sorgente luminosa, il mezzo trasmissivo e il rilevatore. Per convenzione, un impulso di luce vale 1 e l'assenza di luce vale 0. Il messo è una fibra di vetro sottilissima che, quando viene colpito dalla luce il rilevatore genera un impulso elettrico. Quindi collegando ad un estremo della fibra una sorgente di luce e un rilevatore all'altro, si crea un sistema di trasmissione unidirezionale che, preso un segnale elettrico, lo converte e lo trasmette sotto forma di un impulso luminoso. Tale mezzo di trasmissione è utile poiché grazie ad un interessante principio della fisica possiamo non disperdere la luce. Tale principio dice che quando un raggio luminoso passa da un materiale a un altro, per esempio dal silicio fuso all'altro, si rifrange (si curva), se durante la curvatura si crea un angolo tale che supera un valore critico rischiando di disperdere il segnale luminoso esso si rifratta indietro non perdendo quindi il segnale luminoso. Tale tecnologia permette di trasmettere dati ad alta velocità, come ***Ftth*** (fiber to the home). I cavi in fibra ottica sono simili ai cavi coassiali, ma non sono avvolti dalla calza conduttrice, e al centro invece di un conduttore di rame c'è un conduttore di vetro. Il segnale luminoso avviene mediante LED o i laser a semiconduttore. I laser a semiconduttore trasmettono più velocemente, su distanze più lunghe ma sono meno duraturi più costosi e hanno una maggiore sensibilità alla temperatura.

2.3 Trasmissioni wireless

Per persone in mobilità che hanno l'esigenza di essere collegati alla rete i doppini, cavi coassiali o fibre non hanno alcuna utilità, per questo si fa riferimento ai seguenti mezzi di trasmissione:

- ***Trasmissioni radio***: le onde radio sono semplici da generare, possono viaggiare per lunghe distanze e attraversano facilmente gli edifici, per questo sono largamente usate. Inoltre hanno la proprietà di essere omnidirezionale, ossia si espandono dalla sorgente in tutte le direzioni, per questo il trasmettitore e il ricevitore non devono essere fisicamente allineati.
- ***Trasmissioni a infrarossi***: sono largamente usate per le comunicazioni a corto raggio, come ad esempio per i telecomandi dei televisori. Sono trasmissioni direzionali, economiche e facili da costruire, ma hanno il grande difetto di non riuscire ad attraversare ostacoli solidi e di non avere una grande portata.
- ***Trasmissioni a onde luminose***: trasmissioni laser permettono di generare una rete LAN con un costo estremamente ridotto e rappresenta un sistema economico. Tuttavia le trasmissioni laser sono difficili da direzionare e sono influenzabili dagli agenti atmosferici.

2.4 Comunicazioni satellitari

Negli anno '50 e '60 la luna veniva utilizzata come satellite naturale e la marina degli Stati Uniti costruì un sistema funzionante che permetteva alle navi di comunicare facendo rimbalzare i segnali su di essa. Successivamente si capì che la costruzione di satelliti artificiali non solo avrebbe permesso di trasmettere segnali ma anche di amplificarli. Nella sua forma più semplice, un satellite è un grande ripetitore di microonde collocato nel cielo, esso contiene diversi transponder e ognuno di essi riceve un segnale in ingresso su una frequenza e lo ritrasmette su un'altra per evitare interferenze. Ben presto, grazie alla legge di Keplero si capì che se il satellite fosse stato installato nella bassa orbita esso scompare facilmente e quindi si necessita di molti per fornire di una copertura continua, quindi quanto più è alto il satellite, più fornisce copertura.
Esistono diversi tipi di satelliti:

- ***Satelliti geostazionari***: Nel 1945 Clarke ideò un satellite che sarebbe apparso fermo nel cielo, stando fermo nella sua orbita se non fosse per il fatto che successivamente lo dichiarò irrealizzabile a causa dell'impossibilità di collocare in orbita satelliti dall'alto consumo di energia, questo però fino al 1962, quando con l'invenzione dei transistor tutto ciò poteva essere realizzato. Da quel momento per evitare il caos totale nel cielo, le allocazioni degli slot orbitali è gestita dall'ITU.
- ***Satelliti su orbite medie***: essi sono collocati ad altitudini più basse e vengono definiti con il nome di ***MEO*** (medium earth orbit). Essi essendo più bassi possono essere raggiunti da trasmettitori meno potenti ma coprono aree meno estese. I satelliti GPS che noi usiamo sono di tipo MEO.
- ***Satelliti su orbite basse***: prendono il nome di ***LEO*** (low earth orbit). Poiché si spostano velocemente per realizzare un sistema completi di LEO è richiesto l'utilizzo di numerosi satelliti. Essi essendo molto vicini alla terra hanno bisogno di meno energia, il ritardo nelle comunicazioni è di pochi millisecondi ed anche il costo di lancio è nettamente inferiore. Alcuni di questi sono usati per la messaggistica, la navigazione, per le chiamate o il trasporto di dati.

2.5 Modulazione digitale e multiplexing

I canali sia cablati che wireless trasportano segnali analogici, la conversione da bit in segnali si chiama ***modulazione digitale***. Esistono due tipologie principali di trasmissioni:

- ***Trasmissione in banda base***: essa consiste nella trasmissione di bit mediante segnali che fanno l'uso di una tensione continua per rappresentare un 1 e l'assenza uno 0, come la fibra ottica che la presenza di luce rappresenta un 1 e l'assenza uno 0. Questa trasmissione per limitare gli errori prevede due strategie:
  - ***Clock recovery***: in segnali che prevedono lunghe trasmissioni di 1 o di 0 è facile incappare in errori, per ovviare ciò si fa uso di un temporizzatore che divide il segnale in porzioni di tempo.
  - ***Segnali bilanciati***: sono segnali che presentano una tensione tanto positiva quanto negativa, avendo una media di zero.
- ***Trasmissione in banda passante***: per spedire informazione su un canale si usa spesso una gamma di frequenze, questo per due motivi principali. Il primo, per vincoli legislativi, il secondo, per evitare interferenze e quindi di avere la possibilità di permettere du coesistere sullo stesso canale diversi segnali.

I canali di trasmissione vengono normalmente condivisi da più segnali, l'utilizzo di un singolo canale per la trasmissione di molti segnali si chiama ***multiplexing***. Esistono varie tipologie di multiplexing:

- ***Multiplexing a divisione di frequenza***: creare un canale tra due uffici che abbia poca o molta banda ha pressappoco lo stesso costo, per questo conviene la creazione di canali con molta banda e suddividere essa tra molti utenti, per far ciò ad ogni utente viene assegnata una determinata gamma di frequenza entro la quale può trasmettere segnali e rimane esclusiva di esso.
- ***Multiplexing a divisione di tempo***: in questo caso gli utenti fanno i turni secondo una politica round-robin e ognuno di loro, periodicamente, prende possesso della banda completa per un tempo limitato.
- ***Multiplexing a divisione di codice***: questa tecnica prevede la trasmissione di segnali a banda stretta sparsi su una banda di frequenza più ampia, permettendo quindi a tutti gli utenti di fare uso dell'intera banda di cui dispone il canale. Per fare questo il tutto viene regolato mediante codice, in cui ogni utente usa un linguaggio diverso dagli altri e ciò impedisce il formarsi di interferenze.

2.6 La rete telefonica pubblica commutata

Nel 1876 quando nacque il telefono l'apparecchiatura veniva venduta a coppie, uno veniva installato al chiamante ed un altro al chiamato, le spese dell'installazione dei cavi era totalmente a carico del cliente ed esso se voleva comunicare con n utenti doveva creare n collegamenti con i cavi. Quest'idea fu presto abbandonata perché fu chiaro che non poteva funzionare e furono aperti i primi uffici di commutazione. Quando un cliente voleva effettuare una chiamata girava una manovella che faceva suonare una campanella nell'ufficio di commutazione per richiamare l'attenzione di un dipendente il quale manualmente creava il collegamento tra chi voleva chiamare e chi doveva ricevere la chiamata. Il sistema telefonico possiamo quindi costituirlo in tre componenti principali:

1. ultimo miglio/collegamenti locali: doppini analogici che arrivano nelle abitazioni.
2. trunk/connessioni interurbane: fibre ottiche digitali che collegano le centrali di commutazioni.
3. centrali di commutazioni: che spostano le chiamate da una linea all'altra.

Per esaminare bene il funzionamento del sistema telefonico dobbiamo partire dal primo componente di esso, l'ultimo miglio, esso prende questo nome perché rappresenta la parte finale del collegamento anche se esso negli ultimi anni, sopratutto con la fibra, ha superato di gran lunga il miglio. Nel corso degli anni l'ultimo miglio è stato rappresentato a partire dai modem telefonici fino alla fibra ottica, vediamoli nel dettaglio:

- ***Modem telefonici***: come abbiamo visto per spedire i bit sull'ultimo miglio bisogna convertirli in segnali analogici, tale conversione è ottenuta grazie ai modem. Dal punto di vista logico un modem si colloca tra il computer (digitale) e il sistema telefonico (analogico). Questa tecnologia è ormai molto datata ed in disuso, basti pensare che con gli ultimi standard il V.90 ed il V.92 si raggiungevano velocità di 56 kbps.
- ***Linee DSL***: ben presto i 56 kbps offerti dalle reti telefoniche furono ritenuti non abbastanza, basti pensare che le reti televisive raggiungevano velocità di 10 Mbps, furono così avanzate diverse proposte che furono raccolte tutte sotto il nome di ***xDSL***, tra cui la più popolare divenne l'***ADSL*** (asymmetric DSL, questo nome perché offriva velocità in upload minori rispetto a quelle di download, poiché la maggior parte degli utenti preleva più informazioni di quante ne trasmette). In questo tipo di tecnologia più è alta la velocità minore è la copertura, più è bassa la velocità più è alta la copertura, allo stesso tempo più si è lontani dalla centrale più le prestazioni diminuiscono.
- ***FTTH*** (fiber to the home): l'ultimo miglio in rame limita le prestazioni che possiamo ottenere, per questo è preferibile che l'ultimo miglio sia realizzato in fibra ottica. Nel caso migliore avremo l'FTTH, ovvero l'ultimo miglio completamente realizzato in fibra fino al nostro modem in casa, nel caso peggiore avremo l'FTTC, ovvero l'ultimo miglio è realizzato in fibra fino alla cabina e poi dalla cabina al modem di casa dal doppino in rame.

Con il termine ***trunk*** si indicano i segmenti principali di una rete telefonica, quelli destinati a trasportare la maggior parte del traffico e a collegare tra loro i principali centri di commutazione. Essi hanno una maggior velocità rispetto all'ultimo miglio e trasportano unicamente informazioni in formato digitale, quindi non necessità di un apparato di conversione. Essi trasportano migliaia di informazioni simultaneamente e ciò serve a sopperire l'elevato costo di realizzazione di questi collegamenti. La capacità di trasmettere migliaia di informazione simultaneamente prendi il nome di multiplexing TDM e FDM, vediamoli nel dettaglio:

- ***FDM***: le tecniche FDM sono state usate per molti anni per la trasmissione principalmente di canali voci, successivamente adattati per trasmettere dati. Queste tecniche vengono ancora oggi utilizzate per la trasmissione di informazioni che sfruttano collegamenti in rame o microonde. Queste apparecchiature tuttavia essendo analogiche non possono essere gestite da un computer.
- ***TDM***: questa tecnica con il tempo si è confermata essere molto affidabile se non fosse per il fatto che l'ITU non riuscì a creare uno standard internazionale facendo sì che nel mondo venissero utilizzare una serie di modalità incompatibili tra di loro. Generalmente questa tecnica fa riferimento alla portante ***T1***, essa è composta da 24 canali vocali uniti in multiplexing, ognuno dei quali, a turno, inserisce 8 bit nel flusso in uscita. Con il passare degli anni, per creare collegamenti sempre più performanti si è deciso di:
  - ***T2***: collegare 4 canali T1, ognuno dei quali garantiva 1.544 Mbps, per formare un unico canale che garantisse velocità di 6,312 Mbps (non 6,176 perché c'è l'aggiunta di bit di controllo).
  - ***T3***: collegare 7 canali T2, ognuno dei quali garantiva 6,312 Mbps, per formare un unico canale che garantisse velocità di 44,736 Mbps.
  - ***T4***: collegare 6 canali T3, ognuno dei quali garantiva 44,736 Mbps, per formare un unico canale che garantisse velocità di 274,176 Mbps.

Finora abbiamo analizzato il sistema telefonico dal punto di vista dell'impianto esterno (ultimo miglio e i trunk), andiamo ora ad analizzare l'impianto interno, quello composto quindi dai commutatori. Attualmente le reti usano due tecniche di commutazione:

- ***Commutazione di circuito***: tecnica su cui si basa il sistema telefonico tradizionale, tale tecnica, concettualmente, quando una persona avvia una telefonata, l'apparecchiatura di commutazione cerca di creare un percorso fisico completo tra il chiamante ed il chiamato. Agli inizi della telefonia era l'operatore manualmente a creare questo percorso oggi ci sono dispositivi di commutazione automatica.
- ***Commutazione di pacchetto***: con questa tecnica la comunicazione tra i due utenti avviene mediante l'invio di pacchetti, questo fa si che si hanno notevoli benefici, tra cui, non avere la necessità di un percorso fisico dedicato, poiché ogni pacchetto segue una strada diversa, seguendo la disponibilità della banda dei percorsi disponibili, inoltre il pacchetto appena generato viene inviato, senza la necessità di creare prima un percorso. Di seguito vediamo le differenze principali tra le due tecniche.

Caratteristica	Commutazione di circuito	Commutazione di pacchetto
Impostazione della chiamata	Richiesta	Non richiesta
Percorso fisico dedicato	Sì	No
Ogni pacchetto segue la stessa strada	Sì	No
I pacchetti arrivano in ordine	Sì	No
Il guasto dello switch è fatale	Sì	No
Banda disponibile	Fissa	Dinamica
Istante di possibile congestione	All'impostazione	A ogni pacchetto
Potenziale banda sprecata	Sì	No
Trasmissione store-and-forward	No	Sì
Tariffazione	Al minuto	Al pacchetto

Figura 2.44 Confronto tra reti a commutazione di circuito e a commutazione di pacchetto.

2.7 Il sistema telefonico mobile

La tecnologia nel mondo del sistema telefonico mobile ha avuto diverse generazione:

- ***Prima generazione (1G)***: Nasce nel 1946 a St. Louis e utilizzava un singolo trasmettitore di grandi dimensioni collocato in cima ad un alto edificio e aveva un un unico canale utilizzato si per trasmettere che per ricevere, il quale veniva attivato premendo un pulsante, da lì il nome di ***push-to-talk***. Negli anni '60 venne installato l'***IMTS*** che si basava sullo stesso principio del suo predecessore solo che questa volta il trasmettitore era più grande e situato su una collina ed aveva un canale in entrata ed uno in uscita. Tutto questo cambiò con ***AMPS*** (advanced mobile phone system), il quale divideva tutti i sistemi telefonici mobili in celle, ed ogni cella copriva un'aria dai 10 ai 20 km. Ogni cella utilizzava un insieme di frequenze, diverse dalle celle adiacenti, ma uguali a quelle usate da celle non adiacenti, questo gli permetteva di gestire più chiamate all'interno della stessa cella, sfruttando diverse frequenze.
- ***Seconda generazione (2G)***: A differenza della prima generazione che era completamente analogica la seconda generazione era completamente digitale e proprio come nella prima non ci fu uno standard anche nella seconda non fu creato e per questo nacquero diverse tecnologie, tra le quali si affermo la tecnologia ***GSM*** (global system for mobile communications), creata dal gruppo francese Groupe Specialé Mobile, da cui prende l'acronimo, negli anni '80. La nuova generazione sfrutta la stessa tecnologia delle celle di quella precedente, migliorandola, fornendo maggiore sicurezza nella cifrature delle chiamate e la possibilità di dividere le frequenze in intervalli di tempo, in modo da usare le stesse frequenze per più chiamate. La comunicazione avviene mediante un terminale, che contiene un chip rimovibile, chiamato ***SIM*** card, il quale contiene dati segreti che permettono al terminale di identificarsi nella rete e cifrare le telefonate. Il terminale comunica quindi con la centrale mediante un collegamento wireless che prende il nome di ***air interface***.
- ***Terza generazione (3G)***: La terza generazione, completamente digitale, si concentra nella trasmissione di voce digitale e dati.
- ***Quarta generazione (4G)***: E' basata unicamente su la tecnologia a pacchetti. Le chiamate, ormai unicamente ***VoIP***, sono separate dal traffico dati e godono di un approccio multiplexing. L'***EPC*** (Evolved Jacket Core), ovvero il meccanismo che regola la pacchettizzazione, gestisce il tutto affinché si abbia semplice un'alta qualità delle chiamate vocali.
- ***Quinta generazione (5G)***: Latenza più basse e banda più alta.

Capitolo 3

3.0 Introduzione

Nel livello data link discuteremo degli algoritmi usai per ottenere una comunicazione affidabile ed efficiente tra unità d'informazione chiamate frame fra due macchine adiacenti, ovvero due macchine collegate da un canale di comunicazione che agisce concettualmente come un cavo.

3.1 Progettazione del livello data link

Il livello data link fa uso del servizio messo a disposizione dal livello fisico per inviare e ricevere bit su canali di comunicazione. Esso offre molte funzioni, tra cui:

- un'interfaccia di servizio ben definita per il livello rete
- gestire gli errori di trasmissione
- regolare il flusso dati, attraverso un buffer, per evitare che dispositivi riceventi lenti vengano inondati di dati da dispositivi trasmittenti veloci

Per garantire queste funzioni esso prende i pacchetti e li incapsula in frame. Ogni frame è composto da un header, una sezione che contiene il pacchetto da inviare ed una sequenza di chiusura. Tale livello offre diversi servizi di controllo degli errori, tra cui:

- ***servizio senza conferma senza connessione*** (unacknowledged): in questo servizio la macchina sorgente invia i frame alla macchina destinazione ed essa non deve inviare nessuna conferma dell'avvenuta ricezione.
- ***servizio con conferma senza connessione*** (acknowledged): in questo servizio la macchina sorgente invia i frame alla macchina destinazione ed essa invia una conferma dell'avvenuta ricezione.
- ***servizio con conferma orientato alla connessione***: il più sofisticato dei tre e in questo servizio la macchina sorgente invia i frame alla macchina destinazione ed essa invia una conferma dell'avvenuta ricezione ed inoltre tra le due macchine si stabilisce una connessione prima di iniziare a trasmettere i dati ed ogni frame viene numerato ed inviato nell'ordine corretto.

Il tipico approccio di questo livello è di calcolare il ***checksum*** di ogni frame, ovvero il valore del frame, ed inserirlo nel frame, alla ricezione del frame il check sum viene ricalcolato e se corrisponde vuol dire che il frame è stato ricevuto correttamente. La suddivisione in frame viene detta framing, ed è compito del livello data link facilitare l'identificazione di un nuovo frame in una comunicazione, per farlo usa quattro tecniche:

- ***conteggio dei byte***: prima dell'invio si inserisce nell'header del frame il numero di byte da cui è formato, così che il ricevente sa quando il frame termina.
- ***flag byte con byte stuffing***: questa tecnica consiste nell'inserire byte speciali all'inizio e alla fine di ogni frame. Tuttavia può capitare che il byte speciale sia contenuto anche nel frame, in questo modo sarà impossibile capire l'inizio o la fine frame, per ovviare ciò prima dell'invio di ogni frame viene inserito un byte di escape, tale tecnica si chiama ***byte stuffing***.

- ***flag bit con bit stuffing***: questa tecnica è identica alla precedente con la sola differenza che accetta in input frame di diverse dimensioni e non solo frame composti da byte.

- ***violazioni della codifica del livello fisico***: come abbiamo visto spesso il livello fisico per conservare l'integrità dei dati inserisce ridondanza, quindi nell'invio di 4 bit di dati spesso si usano pacchetti da 5 bit, tenendo così 16 delle 32 combinazioni libere, questa tecnica usa proprio quelle combinazioni libere per inserire l'inizio e la fine del frame.

3.2 Rilevazione e correzione degli errori

Come abbiamo al livello rete alcuni canali, come le fibre ottiche, hanno bassi tassi di errore, altri, come doppini o connessioni wireless, ne hanno molti di più. Per ovviare a questi errori il livello data link usa due tecniche:

- ***codice a correzione d'errore***: strategia che permette di inserire informazioni che permettono al destinatario di dedurre se i dati ricevuti sono corretti.
- ***codice a rilevazione d'errore***: strategia che permette di inserire informazioni che permettono al destinatario di dedurre se i dati ricevuti sono corretti, ma non il motivo per cui non sono corretti.

I codici a correzione d'errore vengono invece usati nelle reti a bassa qualità, come quelle wireless, dove il tasso di errore è altro, ecco quattro codici:

1. ***Codice di Hamming***: dati due frame, 10001001 e 10110001, si effettua lo XOR e si ottiene 00111000, il numeri di bit diversi ottenuti facendo lo XOR si chiama distanza di Hamming. L'obbiettivo è quindi di trovare la distanza di Hamming minima.
2. ***Codice convoluzionale***: in questo codice un sistema di codifica elabora i bit in input e restituisce una sequenza di bit in output. Non vengono fissati a priori né lunghezza né delimitatori del messaggio ma l'output dipende dal messaggio in input e da quelli precedenti. Questa tecnica è largamente usata nelle comunicazioni GSM, satellitare e 802.11.
3. ***Codice di Reed-Solomon***: esso è simile al codice di Hamming solo che sfruttando tecnologie più sofisticate non lavora sui singoli bit ma su gruppi di bit.
4. ***Codice a controllo di parità a bassa densità (LDPC)***: codice inventato nel 1962 da Gallager ma utilizzato solo nel 1995 a causa della sua complessità computazione poiché un bit di output è formato solo da una frazione di bit di input.

I codici a rilevazione d'errore vengono invece usati nelle reti ad alta qualità, come fibra ottica, dove il tasso di errore è molto più basso, ecco tre codici:

- ***Bit di parità***: questa tecnica fa sì che il numero di bit 1 all'interno della parola sia dispari, nel caso in cui sia pari esso aggiunge un bit 1 alla fine della parola. (Esempio: parola 1011010 diventa 10110101)
- ***Checksum***: esso è solitamente posizionato alla fine del messaggio come complemento a uno del risultato della somma dei dati, in modo che sommando l'intera parola ed il checksum il risultato sia zero nel caso di assenza di errori.
- ***Controllo di ridondanza ciclica (CRC)***: tra i tre codici è sicuramente quello più affidabile e potente noto anche come ***codifica polinomiale***. Le codifiche polinomiali si basano sul fatto di trattare le sequenze di bit come dei polinomi a coefficienti che possono assumere solo i valori 0 e 1.

3.3 Protocolli data link elementari

Partiamo dal presupposto che nei livelli fisico, data link e rete siano presenti dei processi indipendenti che si scambiano messaggi.l I processi del livello fisico ed alcuni del livello data link sono gestiti da hardware specifico (scheda di rete) mentre i restanti processi del livello data link e del livello rete vengono eseguiti dalla CPU. Adesso immaginiamo che una macchina A vuole scambiare messaggi con una macchina B usando un servizio affidabile e orientato alla connessione, per fare ciò si necessita di protocolli, vediamone alcuni elementari:

- ***Protocollo simplex utopistico***: è il protocollo più semplice possibile in quanto non si preoccupa che qualcosa possa andare male, infatti parte dal presupposto che i dati vengono trasmessi in una sola direzione, il mittente ed il destinatario sono sempre pronti, il tempo di elaborazione dei dati può essere ignorato, la comunicazione presenta buffer infiniti ed inoltre durante la comunicazione non si perde o danneggia alcun frame.
- ***Protocollo simplex stop-and-wait (canale privo di errori)***: tale protocollo previene che il mittente sommerga il destinatario con dati trasmessi a velocità maggiore di quanto quest'ultimo sia in grado di elaborare. Tuttavia anch'esso parte dal presupposto utopistico che il canale di comunicazione sia privo di errore e che la comunicazione avvenga in una sola direzione. Una soluzione semplice ma efficace è che il mittente aspetti un feedback (dummy) da parte del destinatario una volta che quest'ultimo ha ricevuto ed elaborato i dati. Protocolli basati su questa gestione vengono chiamati protocolli stop-and-wait.
- ***Protocollo simplex stop-and-wait (canale soggetto ad errori)***: in una situazione normale, ovvero in cui il canale di comunicazione commette degli errori, i frame potrebbero essere danneggiati o persi, un modo per adattare il protocollo stop-and-wait a questa situazione è quella dell'aggiunta di un timer. Il mittente invia il frame, il frame si danneggia, il destinatario si accorge che il frame è danneggiato e non invia l'acknowledgment, scatta quindi il timer ed il mittente rinvia il frame, questo fino a quando non riceve la conferma da parte del destinatario. Il tutto funziona fino a che l'acknowledgment non si perde, nel caso in cui esso si perda il mittente rispedisce lo stesso frame ed il destinatario ne riceve una copia. Per evitare ciò basta inserire un numero di sequenza nell'header del frame quando viene inviato, così che il destinatario in presenza di due frame con lo stesso numero ne cestina uno.

3.4 Protocolli a finestra scorrevole

I protocolli prima citati consideravano una comunicazione in cui i frami venivano trasmessi in una sola direzione (simplex), mentre nelle comunicazioni moderne i frame vengono inviati in entrambe le direzioni (full-duplex). In questo modo le informazioni che vengono da A e da B sono mescolate ai bit di acknowledgment, tuttavia c'è un modo per evitare l'invio del solo acknowledgment, ovvero inserirlo nell'header del frame successivo, questa tecnica prende il nome di ***piggybacking*** (trasportare in groppa). Questa tecnica evita l'invio di un frame "inutile" ma nel caso in cui il frame successivo impiegasse molto tempo per essere trasmesso? In questo caso l'acknowledgment viene inviato da solo. I successivi protocolli sono quindi di tipo full-duplex e appartengono alla classe chiamata a ***finestra scorrevole.*** Finora abbiamo considerato che il tempo di trasmissione del frame successivo e del rispettivo acknowledgment nell'header sia trascurabile, ma non è sempre così. Nei contesti in cui ciò non è trascurabile la sorgente invia un pacchetto *w* di frame (dove w è maggiore di 1), con un numero consistente di frame inviati ci troviamo nella situazione in cui nel frattempo che il mittente elabora ed invia il nuovo pacchetto di frame i acknowledgment del pacchetto precedente sono già arrivati e quindi la trasmissione è continua e senza rallentamenti. Cosa succede però se nei pacchetti di frame uno di quei frame si danneggia o perde? In questi casi si usano due tecniche:

- ***Go-back-n***: il destinatario rifiuta tutti i pacchetti inviati dopo il frame danneggiato e aspetta che il mittente li rinvii tutti a partire dal frame danneggiato, esso funziona bene se gli errori sono rari, altrimenti il costo del rinvio di tutti i pacchetti sarebbe altissimo e non praticabile.
- ***Selective repeat***: il frame in errore viene scartato, i frame corretti vengono salvati nel buffer ed il destinatario richiede al mittente solo il frame danneggiato.

Capitolo 4

4.0 Introduzione

I collegamenti di rete sono suddivisi in due categorie: connessioni punto a punto e connessione broadcast.
Nel capitolo due si è parlato delle connessioni punto a punto, in questo si parlerà delle connessioni broadcast e dei loro protocolli. I canali broadcast vengono chiamati anche canali ad accesso multiplo o canali ad accesso casuale. I protocolli per la gestione di un canale broadcast appartengono ad un sottolivello del livello data link, chiamato ***MAC***.

4.1 Problema dell'allocazione del canale

Un modo semplice ed efficace per allocare un singolo canale come una linea in una rete telefonica tra molteplici utenti è quello di utilizzare una strategia di multiplexing, come la FDM. In questa tecnica la banda del canale viene divisa in N utenti ed ognuno di esso ha quindi delle frequenze prestabilite, così facendo si evitano interferenze. Questo metodo andrebbe bene se il canale viene usato da un numero costante di utenti, se il numero degli utenti cambia costantemente ci troveremo con una parte importante della banda del canale occupata ma non usata. Una volta capito che l'***allocazione statica*** per i canali dinamici non è una buona idea vediamo le premesse che bisogna fare per gestire ***canali dinamici***:

- ***Traffico indipendente***: ogni utente collegato è indipendente, ha un proprio dispositivo per collegarsi alla rete;
- ***Canale singolo***: Tutte le comunicazioni avvengono su un unico canale;
- ***Collisioni osservabili***: Due frame trasmessi simultaneamente creano una collisione, ed il risultato sarà una comunicazione incomprensibile;
- ***Tempo continuo o diviso in intervalli***: Nel caso in cui la comunicazione non si gestisca a tempo continuo si può dividere il tempo in intervalli discreti e le trasmissioni dei frame iniziano con l'inizio dello slot.
- ***Rilevamento di portante o non rilevamento di portante***: Nel caso di rilevamento della portante le stazioni possono indicare se il canale è in uso evitando collisioni, in assenza di portante ciò non può essere stabilito e si saprà solo ad invio dei frame avvenuto.

4.2 Protocolli ad accesso multiplo

Per allocare un canale ad accesso multiplo esistono molti algoritmi:
- ***ALOHA***: Il tutto inizia alle Hawaii nei primi anno '70 quando l'università delle Hawaii cercava di connettere gli utenti delle varie isole, così utilizzarono una radio a corto raggio, tutti i dispositivi trasmettevano sulla stessa frequenza al computer centrale il quali li ritrasmetteva in broadcast.
- ***ALOHA puro***: l'idea parte dal presupposto di far inviare i dati ai vari utenti ogni volta che vogliono, ciò ovviamente causa collisioni e danneggiamenti dei frami.
- ***ALOHA slotted***: gli utenti non possono inviare quando vogliono ma solo negli slot che gli sono stati dedicati.

I protocolli in cui le stazioni si mettono in ascolto di una portante (di una trasmissione) e si comportano di conseguenza si chiamano protocolli con ***rilevamento della portante***, e sono i seguenti:

- ***CSMA 1-persistente***: quando una stazione ha da trasmettere dei dati ascolta la portante, se è libera trasmette, se è occupata attente che si liberi.
- ***CSMA non persistente***: quando una stazione ha da trasmettere dei dati ascolta la portante, se è libera trasmette, se è occupata attente che si liberi, in questo caso non effettua il controllo di continuo ma attente un intervallo di tempo prima di ripetere il controllo.
- ***CSMA p-persistente***: si applica ai canali divisi in intervalli temporali, quando è il suo turno controllo se la portante è libera e nel caso trasmette.

Una volta che il trasmettitore ottiene il canale con certezza non si ha il problema delle collisioni, in questo caso si usano i seguenti ***protocolli senza collisioni***:

- ***Protocollo a mappa di bit***: Il tempo viene diviso in slot e la stazione 0 invierà un frame nello slot 0, la stazione 1 invierà un frame nello slot 1 e così via, indipendentemente da ciò che fa una stazione, se trasmette o meno quella successiva aspetterà comunque il suo turno.
- ***Protocolli token passing***: ogni stazione trasmette un frame a turno in un ordine predefinito.

Per quanto riguarda le ***LAN wireless*** potremmo avere un approccio simile e usare quindi il protocollo CSMA, peccato che questo non sia il più adatto, poiché non importa solo che la portante sia occupata ma parlando di dispositivi wireless possono creare interferenze anche tra di loro stando vicini, per questo useremo il protocollo ***MACA*** (multiple access with collision avoidance), tale protocollo, ideato nel 1990, esorta il trasmittente ad inviare un ***frame RTS*** (request to send) che contiene la lunghezza del frame che dovrà inviare, i dispositivi adiacenti che lo capteranno resteranno in silenzio per il tempo utili ad inviare il frame.

4.3 Ethernet

Bob Metcalfe, laureato al M.I.T. venne a conoscenza del lavoro di Abramson (inventare protocollo ALOHA) che decise di passare l'estate alle Hawaii, lì vide che furono inventati i precursori di quelli che noi oggi chiamiamo personal computer ma allo stesso tempo le macchine erano scollegate tra loro. Progettò quindi la prima ***LAN*** usando un singolo cavo coassiale lungo e spesso e la chiamo Ethernet, da quel momento furono inventate numerose versioni di Ethernet:

- ***Ethernet commutata***: inizialmente si basava su ***hub***, dove ogni computer era collegato a questo dispositivo che ne gestiva le comunicazioni, esso presentava numerosi vantaggi come la facile e continua aggiunta di stazioni con la possibilità di riutilizzare i cablaggi esistenti, e semplicità nella manutenzione e gestione dei guasti. Tale sistema fu rimpiazzato dagli ***switch***, tale sistema aveva gli stessi vantaggi degli hub ma portando due nuovi vantaggi: evitava le collisioni e più frame potevano essere inviati simultaneamente. Le collisioni vengono evitate poiché lo switch memorizza ogni IP su quale porta trasmette (nel momento in cui una macchina connessa invia un pacchetto lo switch segna in una sua tabella interna che quella macchina invia attraverso quella porta) e perché lo switch ha un buffer interno, quindi quando due frame devono usare la stessa porta per essere trasmessi mentre uno viene trasmesso l'altro viene momentaneamente memorizzato nel buffer.
- ***Fast ethernet***: in fibra raggiunge i 100 Mbps in full duplex con segmenti di massimo 2 km.
- ***Gigabit ethernet***: in fibra raggiunge i 1000 Mbps in full duplex con segmenti di massimo 5 km.
- ***10-gigabit ethernet***: in fibra raggiunge i 10 Gbps in full duplex con segmenti di massimo 40 km.

4.4 LAN wireless

Il principale standard per LAN wireless è 802.11. Sono usate generalmente per collegare dispositivi ad internet senza l'ausilio di cavi ma anche per connettere e far scambiare dati a più dispositivi vicini senza usare internet.
Le reti 802:11 possono essere quindi usate in due modalità:

- ***Modalità infrastruttura***: ogni client è associato ad un AP (access point) che a sua volta è collegato alla rete.
- ***Modalità ad hoc***: i client sono interconnessi tra di loro e scambiano informazioni senza l'ausilio di alcun strumento.

Tutte le tecniche 802.11 utilizzano apparati radio a corto raggio per trasmettere nelle bande di frequenze dei 2,4 GHz o dei 5 GHz poiché queste frequenze sono libere e non necessitano di nessuna licenza per essere usate. Questo fa sì che vengano usate anche dai produttori di telefoni cordless, forni a microonde. Difatti la 5 GHz risulta essere la frequenza più libera e per questo la migliore.
Il protocollo usato è generalmente il protocollo CSMA/CA (CSMA with collision avoidance).
La sicurezza nelle LAN wireless è data dallo schema WPA2, il quale parlando con un server di autenticazione che ha un database di utenti e password determina se la stazione ha il permesso di accesso alla rete. La privacy è data dall'algoritmo di cifratura ***AES*** (standard governativo americano approvato nel 2002) integrato nello schema ***WPA2***.

4.5 Wireless a banda larga

WiMAX combina aspetti sia di 802.11 sia di 3G, rendendolo molto simile alla tecnologia 4G. Come lo standard 802.11 si occupa unicamente di collegare dei dispositivi ad internet. L'architettura è composta da stazioni base che si connettono direttamente al provider, in modalità wireless, che a sua volta si collega a internet. La maggior parte delle installazione WiMAX utilizzano uno spettro di licenza tra i 2,5 ed i 3,5 GHz anche se lo standard è progettato per essere flessibile e poter operare trai 2 e gli 11 GHz.

4.6 Bluetooth

Nel 94 Ericsson insieme ad IBM, Intel, Nokia e Toshiba formarono la SIG con l'intento di sviluppare nuove tecnologie wireless. Il primo progetto fu proprio quello di sviluppare uno standard wireless a basso costo, a bassa potenza e portata ridotta, tale progetto prese il nome di Bluetooth. I protocolli di questo standard permettono ad apparecchiature dotate di questa tecnologia di individuarsi a vicenda e connettersi (effettuando il pairing), per poi trasferire in sicurezza i dati. Gli utilizzi di questo standard sono molteplici, utilizzare due telefoni come walkie talkie, usare auricolari wireless o connettere periferiche (tastiere, mouse o stampanti) a computer senza l'uso di alcun cavo.
Questo standard esclude qualsiasi modello (ISO-OSI, TCP/IP) poiché esso corrisponde più o meno al livello fisico e data link del modello ISO-OSI.
Al livello fisico troviamo un sistema che sfrutta le onde radio di 2,4 GHZ a bassa potenza, avendo una portata di circa 10 metri.
Al livello data link troviamo i protocolli:

- ***L2CAP***, che divide i messaggi in frame e fornisce stabilità;
- ***RFcomm*** emula le porte seriali standard che si trovano nei PC per connettere i dispositivi tra loro.

4.7 RFID

I tag RFID sono dispositivi piccoli ed economici, con un unico identificatore RPC a 96 bit e una piccola quantità di memoria che può essere letta e scritta da un lettore RFID. La memoria può essere utilizzata per registrare lo storico della posizione di un oggetto, per identificare un oggetto, per contenere piccole informazioni e molto altro. I tag possono essere divisi in due categorie: quelli di classe 1 che sono sprovvisti di energia e la ottengono mediante le onde radio trasmesse da un lettore RFID e quelli più avanzati che hanno anche una fonte di energia interna.

4.8 Commutazione a livello data link

Molte aziende hanno la necessità di unire più LAN tra di loro, questo è possibile mediante dispositivi chiamati ***bridge*** (switch). In questo paragrafo vedremo come unire più LAN fisiche mediante bridge e come gestire una LAN fisica come diverse LAN logiche, chiamate ***VLAN*** (virtual LAN).
I motivi per cui è utile unire più LAN mediante switch, tra cui:

- con il tempo un'azienda si espande creando nuovi edifici e nuove LAN, gli switch servono per unire le nuove LAN alle LAN già esistenti;
- un'azienda può avere edifici geograficamente distribuiti;
- per suddividere il carico di lavoro di una sola LAN su più LAN.

I bridge per sapere com'è configurata la rete usano l'algoritmo ***backward learning*** esso crea una tabella hash in cui non appena un frame viene inviato da una macchina su una determinata porta la tabella sarà aggiornata inserendo che quella macchina trasmette e riceve su quella determinata porta.
I dati vecchi di qualche minuto nella tabella hash vengono eliminati, poiché se una macchina viene scollegata e collegata su un'altra porta la tabella deve essere aggiornata per non creare errori.

Differenze tra repeater, hub, bridge, switch, router e gateway

Questi dispositivi seppur svolgono ruoli simili all'interno di una rete hanno numerose seppur differenti caratteristiche e si trovano in livelli differenti.

Livello applicazione	Gateway applicazione
Livello trasporto	Gateway trasporto
Livello rete	Router
Livello data link	Bridge, switch
Livello fisico	Repeater, Hub

Repeater: Situati nel livello fisico sono dispositivi analogici che gestiscono segnali sui cavi a cui sono collegati, il loro scopo è prendere un segnale su un cavo, ripulirlo dal rumore, amplificarlo e trasmetterlo su un altro cavo.
Hub: Situati nel livello fisico essi hanno più dispositivi collegati ad essi. Se due dispositivi comunicano contemporaneamente con l'hub i frame collidono (per evitare si usa l'algoritmo CSMA/CD). Tutti i dispositivi collegati all'hub devono operare alla stessa velocità.
Bridge: Situati nel livello data link vengono usati per collegare più LAN tra di loro. A differenza degli hub ogni porta è isolata e se la porta ha una linea full duplex punto a punto non c'è bisogno nemmeno dell'algoritmo CSMA per evitare le collisioni. I bridge quindi offrono prestazioni migliori rispetto agli hub, l'isolamento delle porte garantisce anche che ogni dispositivo possa viaggiare a velocità diverse. Essi hanno anche un buffer interno.

Switch: Sono una versione moderna dei bridge. Differiscono dai bridge poiché nell'epoca moderna le istallazioni usano collegamenti punto a punto, e i singoli computer devono essere collegati direttamente allo switch da qui nasce l'esigenza di avere più porte e nasce quindi lo switch.
Router: Utilizzati per la trasmissione di pacchetti verso la rete esterna mediante gli IP.
Gateway trasporto: Connettono due computer che usano protocolli di trasporto differenti.
Gateway applicazione: Comprendono il formato e il contenuto dei dati ricevuti e possono trasdurre i dati da un formato ad un altro.

Virtual LAN

Dato l'avanzamento delle tecnologie oggi è possibile configurare *n* LAN virtuali piuttosto che *n* LAN fisiche.
Esse nascono da quattro esigenze:

1. L'importanza di sapere quale utente accede a cosa e quindi dividere la LAN fisica in più LAN logiche permette di dividere le classi di utenti e dare ad ognuno di loro l'accesso solo alle risorse di cui ne hanno l'autorizzazione;
2. Suddividere il carico della rete, dando ad ogni LAN logica una certa quantità di banda evitando che qualche settore dell'azienda intasi totalmente la LAN lasciando gli altri senza rete;
3. Gestione del traffico broadcast. Quando uno switch non conosce la posizione della macchina a cui deve trasmettere un dato invia un messaggio broadcast chiedendo a quale macchina corrisponde quell'indirizzo IP. Suddividere la LAN permette che quel messaggio broadcast viene inviato solo alla LAN logica di riferimento e non a tutta la LAN fisica intasando quindi inutilmente tutta la LAN;
4. In caso di guasti di rete e più facile contenere i danni.

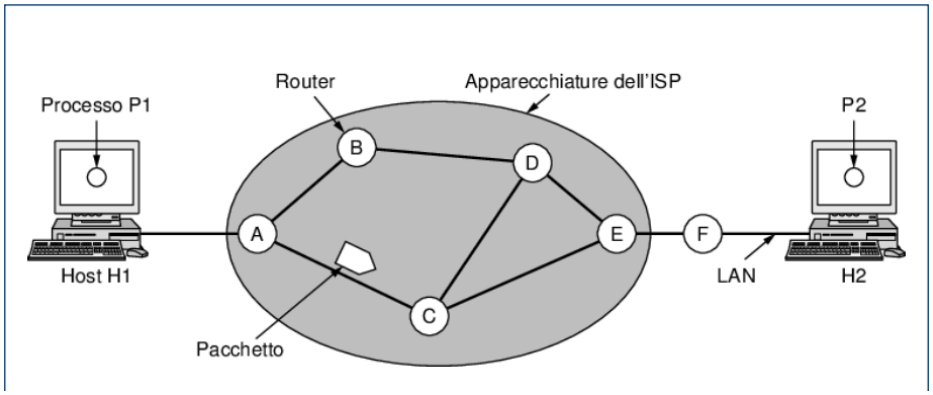
In risposta a queste esigenze il comitato IEEE 802 ha realizzato lo standard ***IEEE 802.1Q*** introducendo quindi il concetto di ***VLAN***. Questo standard inserisce nel frame un nuovo campo dove vengono specificate le informazioni della macchina destinataria situata sulla stessa VLAN.
Le VLAN si basano su appositi switch (VLAN-aware), dove l'amministratore di rete decide quante VLAN dovranno esserci, quali computer collegare ad ognuna e quale nome dare ad ogni VLAN.

Capitolo 5

5.0 Introduzione

Il livello di rete si occupa del trasporto dei pacchetti dall'origine della trasmissione alla destinazione finale. Lo scopo di questo livello è data la conoscenza della topologia della rete (l'insieme dei router e dei collegamenti che si frappongono tra il trasmittente ed il destinatario) trovare il percorso migliore per i pacchetti.

5.1 Problematiche nella progettazione del livello rete



Come mostra l'immagine un pacchetto inviato dall'host H1 per raggiunger l'host H2 deve attraversare numerosi router, per far ciò e restare integro si usa la tecnica di store-and-forward, il pacchetto viene quindi memorizzato di router in router ed inviato al router successivo per verificare il checksum.
I servizi che offre il livello rete al livello trasporto devono rispettare le seguenti caratteristiche:

- non devono dipendere dalla tecnologia dei router;
- al livello trasporto devono essere nascosti dettagli come quanti e di che topologia sono i router;
- gli indirizzi di rete al livello trasporto devono essere uniformi.

Date queste direttive i progettisti del livello rete sono liberi di definire i servizi da offrire al livello trasporto. I progettisti del livello rete si dividono in due categorie:

1. Una fazione afferma che il lavoro dei router e quello di spostare un pacchetto da un router all'altro e niente più. Affermando che la rete, seppur progettata bene, non è affidabile ed è compito degli host gestire eventuali errori di trasmissione;
2. L'altra fazione afferma il contrario, sostenendo che la rete dovrebbe offrire un servizio affidabile ed orientato alla connessione.

I servizi offerti quindi possono essere di due tipi:

1. ***Servizi senza connessione (datagram)***: in questo caso i pacchetti sono instradati indipendente l'uno dall'altro senza definire prima il percorso che dovranno effettuare.

Di conseguenza ogni router sa che per inoltrare un pacchetto ad un determinato host a quale router deve inoltrare il pacchetto. Ad esempio pacchetto che va da H1 ad H2 parte da A e deve arrivare ad F.
A sa che per arrivare ad F deve inoltrare a C.
C sa che per arrivare ad F deve inoltrare a E.
E sa che per arrivare ad F deve inoltrare a F.

2. ***Servizi orientati alla connessione (a circuito virtuale)***: l'idea è quella di non dover creare ogni volta una strada per inviare un pacchetto da H1 a H2 ma una volta stabilita la

connessione tra i due host si definisce un percorso (circuito virtuale) che viene salvato nella tabella dei router.

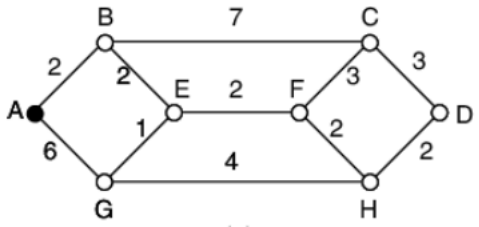
Problema	Rete datagram	Rete a circuito virtuale
Impostazione del circuito	Non necessaria	Necessaria
Indirizzamento	Ogni pacchetto contiene gli indirizzi completi di sorgente e destinazione	Ogni pacchetto contiene un breve identificatore di circuito virtuale
Informazioni di stato	I router non mantengono informazioni di stato sulle connessioni	Ogni circuito virtuale richiede spazio nella tabella del router per la connessione
Routing	Ogni pacchetto è inoltrato in maniera indipendente	Il percorso viene scelto durante l'impostazione del circuito e tutti i pacchetti lo seguono
Effetto del malfunzionamento di un router	Nessuno, tranne i pacchetti persi perché in transito durante il malfunzionamento	Tutti i circuiti virtuali che passano dal router vengono chiusi
Qualità di servizio	Difficile	Facile se abbastanza risorse possono essere allocate in anticipo a ogni circuito virtuale
Controllo della congestione	Difficile	Facile se abbastanza risorse possono essere allocate in anticipo a ogni circuito virtuale

5.2 Algoritmi di routing

L'algoritmo di routing è quella parte del software del livello di rete che si occupa di scegliere lungo quale linea di uscita inoltrati i pacchetti in arrivo.
Gli algoritmi di routing si dividono in due categorie:

1. ***Algoritmi non adattivi (routing statico)***: tale algoritmo non tiene conto del traffico o della congestione di un determinato circuito ma organizza il percorso di trasmissione di un pacchetto al momento dell'avvio della rete.
2. ***Algoritmi adattivi (routing dinamico)***: prima di decidere il percorso da dover far fare ad un pacchetto tiene conto di possibili modifiche effettuate alla rete e al traffico.

Un principio alla base degli algoritmi di routing è il ***principio di ottimalità***, esso afferma che se il router J si trova sul percorso ottimale che collega I a K allora anche il percorso ottimale da J a K segue la stessa sequenza di router.
Algoritmo di cammino minimo: tale algoritmo definisce come cammino minimo e quindi ottimale il percorso in cui ci sono meno salti.



Prendendo in considerazione questo algoritmo il percorso ABC E ABE sono uguali sebbene ABC è un percorso decisamente più lungo, ecco il limite più grande di questo algoritmo.

Flooding: gli algoritmi di routing devono effettuare scelte in base alle conoscenze locali e non in base all'intera rete, difatti una semplice tecnica locale è il flooding, essa consiste nell'inviare ogni

pacchetto in arrivo su tutte le linee in uscita tranne quella da cui proviene. Questa tecnica seppur semplice ed efficace ha l'enorme problema che così facendo si generano enormi quantità di pacchetti duplicati, per evitare ciò ad ogni pacchetto nell'header può essere inserito un counter, ad ogni salgo il counter diminuisce e all'arrivare a 0 o ha raggiunto la destinazione o viene cestinato. Per evitare una seconda trasmissione inoltre nell'header devono essere contenuti i router da cui il pacchetto è stato inoltrato. Come abbiamo potuto notare nonostante le contromisure il flooding rimane comunque una tecnica che genera numerosi pacchetti duplicati ma a sua volta ha diversi punti a favore:

- è robusto, quindi anche se un gran numero di router saltasse il pacchetto riuscirebbe comunque ad arrivare a destinazione;
- è uno spreco se la destinazione del pacchetto è unico ma è molto efficace nelle reti broadcast.

Le reti di calcolatori utilizzano generalmente algoritmi di routing dinamici, più complessi ma più efficienti del flooding, i due principali sono:

1. Algoritmo di routing basato sul ***vettore delle distanze***: esso fa in modo che ogni router conservi una tabella che contiene la distanza ed il percorso per raggiungere ogni router della rete. Queste tabelle si aggiornano scambiando informazioni con i router vicini. Alla fine dell'esecuzione dell'algoritmo ogni router conosce il collegamento migliore per raggiungere qualsiasi altro router.
2. Algoritmo di routing basato sullo ***stato dei collegamenti***: questo algoritmo può essere riassunto in cinque fasi:
    1. scoprire i router vicini ed i relativi indirizzi di rete
    2. misurare la distanza e la metrica di costo di ogni vicino
    3. costruire un pacchetto che contiene le informazioni precedentemente raccolte
    4. inviare tale pacchetto ai router vicini e ricevere da loro i pacchetti
    5. attraverso le informazioni raccolte elaborare il percorso più breve per raggiungere gli altri router

Routing gerarchico: all'aumentare della dimensione della rete aumenta anche la dimensione delle tabelle di routing. Questa conseguenza non solo consuma la memoria del router ma aumenta anche il tempo che la CPU impiega per analizzare i dati. Per evitare questi problemi si utilizza il routing gerarchico, la rete viene divisa in regioni ed i router conoscono le informazioni solo dei router contenuti nella loro regione. Per raggiungere un altra regione un solo router conosce la strada per raggiungere un solo router di un altra regione.

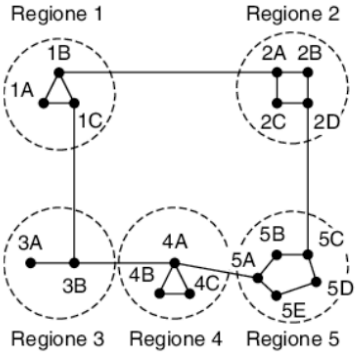


Tabella completa per 1A

Dest.	Linea	Salti
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Routing broadcast: in alcune applicazioni gli host hanno bisogno di inviare a molti o a tutti gli altri host. La trasmissione di un pacchetto a tutte le destinazioni è chiamata broadcasting. Ci sono due metodi principali per effettuarla e sono:

- ***Multidestination routing***: ogni pacchetto inviato contiene una lista delle destinazioni da raggiungere, il router che lo riceve controlla questa lista e vede se a sua volta deve inoltrarlo.
- ***Reverse path forwarding***: quando un host riceve un pacchetto broadcast verifica se il pacchetto, inviato da quello specifico host, ha seguito la solita, e quindi migliore, linea per raggiungerlo, se la risposta è affermativa lo tiene se è negativa considera che il pacchetto sia un duplicato e lo elimina.

Routing multicast: avviene quando un host ha la necessita di inviare informazioni a più host della reta ma non ad abbastanza host per sostenere i costi di una comunicazione broadcast, ad esempio inviare a 1000 host in una rete formata da milioni di host. Per la trasmissione dei pacchetti tuttavia usa la stessa strategia del broadcasting, inviando il pacchetto solo alle regioni di interesse.

Routing anycast: nell'anycast il pacchetto viene inviato al router più vicino. Questa funzione è utile nel caso in cui l'informazione non ci interessa da quale host arriva ma che arrivi nel minor tempo possibile, come richiedere l'ora del giorno o la distribuzione di contenuti. Questa tecnica è usata da internet come parte del servizio DNS.

Routing per host mobili: A differenza degli host visti finora qui gli host non hanno una posizione fissa ma sono in continuo movimento. Dato il grande numero di host mobili che abbiamo oggi è impossibile calcolare ogni volta il percorso e trovare dove si trova l'host mobile nel momento in cui gli dobbiamo recapitare un pacchetto. Per evitare ciò sarà proprio l'host mobile a comunicare la posizione ad un host della LAN in cui si trova in quel momento che prenderà il nome di ***home agent***.

Routing ad hoc: Prima abbiamo visto cosa accade quando gli host sono mobili ma i router sono fissi, nel caso in cui entrambi i dispositivi sono mobili avremo che ogni host è sua router che host. Queste reti vengono usate in un area di copertura 802.11 (reti wireless), in scenari come campi di battaglia o soccorritori durante una catastrofe ambientale.

N.B.
Unicast: trasmette ad un solo host.
Broadcast: trasmette a tutti gli host.
Multicast: trasmette ad un gruppo di host.
Anycast: trasmette all'host più vicino.

5.3 Algoritmi per il controllo della congestione

Quando in una rete sono presenti troppi pacchetti, si verificano ritardi e perdite di pacchetti che causano un calo di prestazioni, questa situazione si chiama **congestione**. Un modo efficace per evitarla è quello di controllare il flusso di dati immesso nella rete dal livello di trasporto, e nel caso serva di limitarlo. Questo implica che se la rete non è ben progettata presto si verificherà un collasso da congestione.
Un esempio di causa di congestione è quando le sorgenti ritrasmettono pacchetti fortemente in ritardo, causando così una serie di duplicati che intasano la rete, una soluzione potrebbe essere quello di aumentare la memoria di buffer della rete ma come scoprì Nagle nell'87 tutto ciò è inutile, anzi provocherebbe l'effetto contrario da quello voluto perché immagazzinando tutti i pacchetti alla fine si finirà per spedire anche i pacchetti fortemente in ritardo che ormai non servono più. Il controllo di flusso in generale server per gestire le casistiche più comuni della congestione, ovvero:

- su una rete di 100 Gbps un supercomputer invia con una capacità di 1 Gbps ad un personal computer, questo non intasa la rete ma il personal computer non è in grado di gestire tutti quei dati ricevuti;
- su una rete di 1 Mbps 1000 computer trasmettono con una capacità di 100 kbps, in questo caso è la rete a non supportare tutti quei dati contemporaneamente.

Tra le soluzione troviamo quattro tecniche:

1. **Traffic-aware routing**: In questa strategia vengono usati gli schemi di routing e vengono creati in base al traffico di dati che pesa su ogni collegamento, cercando quindi di evitare tratti troppo carichi di trasmissioni e dividendo il traffico su tutta la rete. Tuttavia questo metodo ha un limite, prendiamo il caso di due reti diverse collegate da due linee, inizialmente la prima linea è quella più usata quindi l'algoritmo di routing dirà che dal prossimo invio bisognerà usare la seconda, portando così ad essere la seconda linea quella più usate e quindi ci troveremo al punto di inizio, dove a quel punto l'algoritmo di routing dirà che bisognerà tornare ad usare la prima linea e rendendo così la prima linea quella più usata. Per risolvere questo problema bisognerà o dividere equamente il traffico su entrambe le linee o rallentare l'immissione del traffico sulla rete.
2. **Controllo di ammissione**: tale tecnica si basa su un'idea semplice, nessun circuito virtuale viene impostato se la rete non è in grado di supportarlo, esso infatti crede sia meglio evitare di far collegare l'utente alla rete piuttosto che farlo collegare e creare congestione ed un malfunzionamento generale.
3. **Limitazione del traffico**: tale tecnica prevede nel far gestire alle macchine quanto traffico immettere in rete, lasciando a loro il compito di valutare quanto carico la rete possa supportare, però nel caso la rete si trovi vicino ad un punto di congestione gli utenti verranno avvisati ed invitati ad immettere meno dati in rete.
4. **Load shedding**: quando nessuna delle tecniche precedenti basta allora i router passano all'artiglieria pesante eliminando il carico, scartando quindi una quantità di pacchetti tale che eviti la congestione.

5.4 Qualità del servizio

Le tecniche descritte nei paragrafi precedenti sono state progettate per ridurre la congestione ma non bastano, poiché i clienti della rete vogliono garanzie di qualità, come una capacità minima della rete ed un ritardo massimo dell'invio dei pacchetti. Una soluzione semplice per fornire una buona qualità del servizio è chiamata **overprovisioning**, ovvero costruire una rete capace di gestire qualunque traffico venga immesso. Il problema di questa soluzione sono i costi, quindi prima di costruire una rete per un servizio bisogna sapere:

- di che cosa hanno bisogno le applicazioni della rete;
- come regolare il traffico immesso in rete;
- come prenotare le risorse de router;
- capire se la rete può gestire il traffico.

5.5 Internetworking

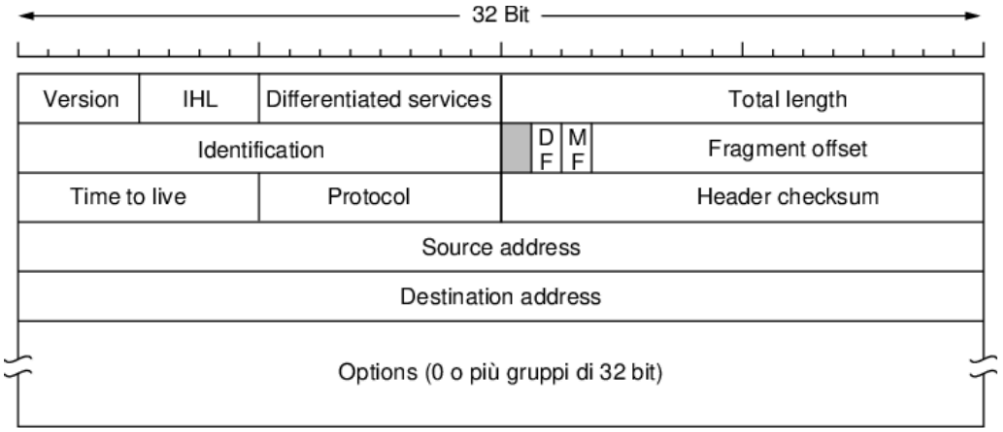
La connessione tra due reti diverse (LAN, PAN, WAN) forma una **internetwork**, o semplicemente **internet**. La comunicazione tra due reti diverse però non è così semplice poiché le reti differiscono tra loro per diversi aspetti, e come si fa a connettere due reti che hanno velocità di banda diverse o che una supporta il multicast e l'altra no? Queste differenze possono essere superate in due diversi modi, o costruiamo dispositivi che traducono o convertano i pacchetti da un tipo di rete all'altra o aggiungiamo un livello sopra gli altri comune per tutte le reti. La seconda soluzione fu adottata da Cerf e Kahn nel 74' con l'introduzione dei protocolli **TCP** e **IP**. Il protocollo **IP** nello specifico fornisce un formato universale per i pacchetti, riconosciuto da tutti i router e adatto a ogni rete.
Tuttavia ancora oggi non viene usato un unico protocollo, sebbene vecchi protocolli come IPX e Appletalk sono ormai in disuso esistono oggi protocolli diversi molto usati, come **IPv4** e **IPv6**. Router in grado di gestire diversi protocolli sono definiti router **multiprotocollo**.
Sebbene gestire la comunicazione tra reti diversi può essere complesso esiste un caso in cui ciò è molto più semplice, ovvero il caso in cui il trasmittente ed il ricevente usano il medesimo modello rete ma sono collegate tra loro da un modello di rete diversa.
Tale problema viene risolto grazie ad una tecnica chiamata **tunneling**, prendiamo l'esempio in cui due reti reti A e C usano reti IPv6 e sono collegate tra di loro da una rete B che usa una rete IPv4, in questo caso il tunneling consiste nel fatto che la rete A genera un pacchetto IPv6 e lo incapsula in un pacchetto IPv4, lo trasmette alla rete B la quale lo trasmette alla rete C, la rete C quindi ricevuto il pacchetto in formato IPv4 estrae da esso il pacchetto IPv6.
Collegare reti diverse tra di loro rappresenta una sfida maggiore, ad esempio se una rete usa un link state routing, e quindi necessita di conoscere l'intera topologia di rete, e l'altra un distance vector routing, che non necessita invece di conoscere la topologia di rete, trovare la strada migliore per indirizzare i pacchetti risulta molto difficile. Sopratutto se le due reti hanno due operatori diversi ed un operatore per ragioni di marketing non vuole far sapere i suoi percorsi nella rete. A risolvere tale problema interviene l'introduzione dell'**algoritmo di routing a due** livelli:

- **Protocollo di routing intradominio** (interior gateway protocol) usato all'interno della rete;
- **protocollo di routing interdominio** (exterior gateway protocol) usato per comunicare tra le reti, difatti le reti possono usare un proprio protocollo di routing intradominio ma devono usare lo stesso protocollo di routing interdominio che in internet viene chiamato **BGP**.

Ogni rete viene gestita in modo indipendente e viene definita **AS** (autonomous system), un tipico esempio di AS è la rete di un ISP (internet service provider).

5.6 Il livello di rete in Internet

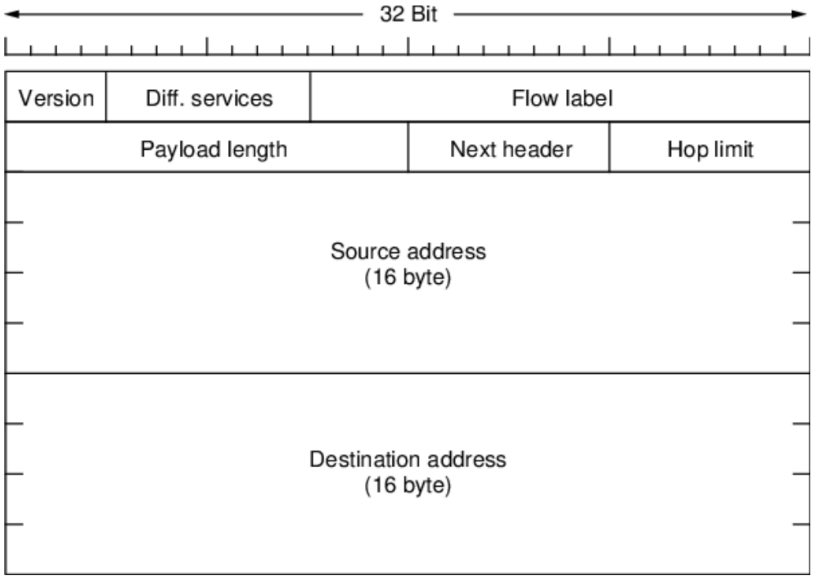
Nel livello di rete Internet viene visto come un insieme di reti o di AS interconnesse. Non esiste una vera struttura esistono però dorsali principali a banda larga e router veloci, chiamate reti di livello 1, alle quali sono collegati gli ISP, ai quali sono connesse le reti regionali, LAN di molte università o aziende. Il collante che unisce tutte questi reti è il protocollo **IP**.
Un **protocollo IPv4** è costituito da una parte di intestazione (20 byte) e da un corpo di lunghezza opzionale.
I bit vengono letti da sinistra verso destra e dall'alto verso il basso usando l'**ordine big endian**, difatti macchine che usano little endian (ad esempio macchine Intel) devono effettuare una traduzione sia in ricezione che in trasmissione. A posteriori little endian sarebbe stata la scelta migliore, ma quando IP fu progettato nessuno immaginava che sarebbe diventato dominante.



Version: tiene traccia del protocollo usato;
IHL: poiché la dimensione del pacchetto non è costante c'è il campo IHL che indica la lunghezza dell'intestazione
Differentiated services: indica il tipo di servizi che richiede quel pacchetto, ad esempio la voce digitalizzata richiede più velocità di trasmissione a dispetto della precisione di consegna, trasmissione dati invece è il contrario;
Total lenght: dimensione totale del pacchetto tra intestazione e dati;
Identification: server all'host che lo riceve per capire a quale datagramma appartiene;
Bit inutilizzato: essendone già pochi non si conosce il motivo di questo bit inutilizzato;
DM (don't fragment): indica che il pacchetto non deve essere frammentato;
FM: indica che il pacchetto è un frammento, tutti i frammenti, tranne l'ultimo, hanno questo bit settato a 1;
Fragment offset: indica la posizione del frammento nel datagramma;
Time to live: è un contatore che può indicare il tempo di vita del pacchetto;
Protocol: Il protocollo di trasporto usato (TCP, UDP);
Header checksum: il checksum dell'intestazione;
Source address e destination address: indica gli indirizzi dell'interfaccia di rete di partenza e di arrivo;
Options: campo riservato a versione successive;
In origine il campo options comprendeva notizie riguardo la sicurezza, il percorso che deve compiere il datagramma, l'elenco dei router del percorso, ogni router aggiunge il suo IP al campo e ogni router doveva aggiungere l'indirizzo e l'ora in cui è passato il datagramma.

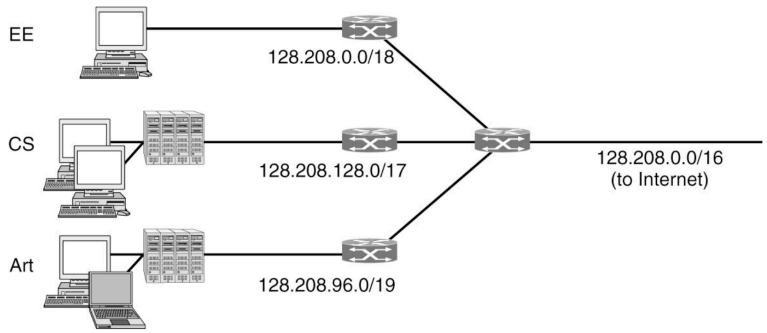
IPv6 risolve quattro aspetti critici di IPv4, e fornisce quindi:

- indirizzi più lunghi, 128 bit al posto dei 32 bit di IPv4, fornendo quindi indirizzi IP pressoché illimitati
- offre un intestazione semplificata, solo 7 campi al posto dei 13 di IPv4;
- supporta meglio le opzioni, i campi che una volta erano obbligatori ora sono opzionali, poiché non usati così spesso;
- migliorata la sicurezza, cosa che negli IPv4 è stata introdotta solo dopo la sua creazione.



Di sopra sono riportati i campi obbligatori dell'intestazione di IPv6 e sono:
Version: indicando la versione dell'indirizzo IP (6 per IPv6 e 4 per IPv4)
Differentiated services: indica il tipo di servizi che richiede quel pacchetto, ad esempio la voce digitalizzata richiede più velocità di trasmissione a dispetto della precisione di consegna, trasmissione dati invece è il contrario;
Flow label: consente di marchiare un gruppo di pacchetti, che, avendo gli stessi requisiti, devono essere trattati allo stesso modo;
Payload lenght: indica la dimensione del pacchetto esclusa la dimensione dell'intestazione (40 byte)
Next header: indica, se usata, quali delle 6 intestazioni opzionali è presente dopo l'intestazione di base;
Hop limit: indica il tempo di vita del pacchetto, quindi quanti hop (salti) deve eseguire prima di morire, quindi è un contatore che ad ogni salto viene decrementato, a differenza di Time to live di IPv4 che era un tempo espresso in secondo e che non veniva mai usato;
Source address e destination address: indica gli indirizzi dell'interfaccia di rete di partenza e di arrivo;
Le intestazioni estese di IPv6 inoltre possono contenere informazioni sui router su cui passerà il pacchetto, la lista dei router da visitare, gestione dei frammenti del datagramma, informazioni relative al contenuto cifrato e molto altre.

Gli indirizzi IP sono gerarchici, questo significa che una rete corrisponde a un blocco contiguo di indirizzi IP, tale blocco è chiamato prefisso. Gli indirizzi di rete sono scritti in **notazione decimale puntata**. In questo formato ognuno dei 4 byte è rappresentato in forma decimale con un numero che vari tra 0 e 255 (255.255.255.255). Nel caso di un indirizzo come 128.208.0.0/24 il nostro prefisso contiene 2^8 indirizzi IP e 24 sono i bit dedicati alla rete. Per evitare i conflitti i numeri di reti sono affidati ad un ente no profit chiamato ICANN.



In questo caso all'università sono affidati /16 blocchi, quindi possono collegarsi 65.536 host di cui /17 (32768) al dipartimento degli informatici, /18 (16384) al dipartimento degli ingegneri elettronici e /19 (8192) al dipartimento di arte.
Gli indirizzi IP tuttavia sono pochi, un ISP potrebbe avere un indirizzo /16 che gestisce 65.526 host ma se supera quel numero sorgono problemi. Per evitare tale problema si possono usare tre strategie:

- Un modo per evitare tale problema è assegnare alle macchina **IP dinamici**, quindi quando la macchina smette di essere collegata alla rete l'indirizzo IP gli viene revocato ed assegnato ad un'altra macchina;
- L'utilizzo dello standard **IPv6** che adotta indirizzi a 128 bit rispetto ai 32 bit di IPv4;
- Usare come soluzione il **NAT (network address translation)**, esso consiste nell'assegnare a ciascuna azienda o casa un unico indirizzo IP pubblico per comunicare in rete e localmente assegnare una serie di indirizzi privati a ciascuna macchina collegata, la quale per comunicare in rete userà l'indirizzo pubblico. Qui sorge il problema di non sapere poi quale macchina sta effettivamente comunicando in rete e a quale macchina devo comunicare il risultato, qui interviene l'uso delle **porte**. Prendendo l'esempio di connessione del tipo TCP, quando un processo vuole stabilire una connessione TCP con un processo remoto si lega ad una porta TCP inutilizzata della macchina chiamata **porta sorgente** ed indica a quale **porta di destinazione** devono arrivare i pacchetti.

Oltre a IP ci sono altri protocolli di controllo molto utilizzati in Internet e sono:

- **ICMP (internet control message protocol)**: protocollo usato per monitorare il corretto funzionamento di una rete, nel caso qualcosa non funzionasse i tipici messaggi di errore sono i seguenti:

Tipo di messaggio	Descrizione
DESTINATION UNREACHABLE	Non è stato possibile consegnare il pacchetto
TIME EXCEEDED	Il campo Time to live ha raggiunto il valore 0
PARAMETER PROBLEM	Campo dell'intestazione non valido
SOURCE QUENCH	Pacchetto choke
REDIRECT	Insegna a un router la geografia
ECHO e ECHO REPLY	Controlla se una macchina è funzionante
TIMESTAMP REQUEST/REPLY	Come Echo, ma con un timestamp

- **ARP (address resolution protocol)**: ogni macchina dispone di uno o più indirizzi IP, tuttavia non possono essere usati poiché le macchine non comprendono gli indirizzi IP, nel caso di macchine che usano una scheda di rete, ciascuna di loro ha un indirizzo Ethernet di 48 bit univoco che usano per comunicare con altre macchine, senza saper nulla di IP. Quindi come si associa un indirizzo IP ad un indirizzo Ethernet? A questo ci pensa il protocollo ARP, il gestore della rete deve solo assegnare ad ogni macchina un indirizzo IP e stabilire le maschere di sottorete, a tutto il resto ci pensa ARP.
- **DHCP (dynamic host configuration protocol)**: ARP, come altri protocolli, partono dal presupposto che gli host sono configurati con alcune informazioni base, come il proprio indirizzo IP, questo è possibile farlo manualmente ma sarebbe complicato, ad aiutarci invece c'è il DHCP che effettua tutto in maniera automatica.

Finora abbiamo parlato di invio di pacchetti di datagrammi, utilizzando quindi una tecnologia che non nasce orientata alle connessione ma che bensì è stata adattata. Una tecnologia che nasce orientata alle connessione è chiamata **MPLS (multiprotcol label switching)** essa consiste nell'aggiungere un etichetta all'inizio di ogni pacchetto, usando quella per conoscere a quale router inoltrare il pacchetto piuttosto che usare l'indirizzo di destinazione.

Routing intradominio: internet è gestita da diverse reti indipendenti, ed ognuna, all'interna della propria rete può adottare un proprio algoritmo di routing. I protocolli più usati sono OSPF (usato dalle reti aziendali) e IS-IS (usato dagli ISP).

OSPF ha diverse caratteristiche:

1. è di dominio pubblico, da li la "O" che sta per open;
2. capace di supportare diverse metriche come la distanza fisica o i ritardi dei pacchetti;
3. doveva essere dinamico, capace di adattarsi velocemente alla topologia di rete;
4. capace di adattarsi in base al tipo di traffico che trasmetteva;
5. capace di dividere il carico su tutta la rete, evitando congestioni;
6. capace di supportare sistemi gerarchici, essendo internet in continua espansione;
7. doveva essere un protocollo sicuro.

Esso ha tipicamente 4 tipi di messaggi:

1. **Hello**: invia messaggi in modalità multicast per conoscere i router vicini ad esso;
2. **Link state update**: comunica ai vicini lo stato del dispositivo;
3. **Database description**: comunica lo stato dei collegamenti posseduti in quel momento;
4. **Link state request**: tra la coppia dei router in comunicazione si controlla chi dei due ha dati più recenti, trasmettendo quindi le informazioni più recenti alla rete.

Routing interdominio: per la comunicazione tra reti diverse, quindi di AS diversi si usano protocolli di routing standard, in grado di mantenere tale gestione del routing sopra a politiche economiche o statali, generalmente si usa il protocollo **BGP (border gateway protocol)**, esso ad esempio è in grado di garantire che:

- non ci sia traffico commerciale su reti per la ricerca;
- mai porrei l'Iraq su un percorso che inizia dal Pentagono;
- il traffico generato o in arrivo a Apple non dovrebbe passare attraverso Google.

Mobile IP: Molti utenti usano Internet attraverso computer portatili, spostandosi quindi geograficamente volendo però rimanere connessi a internet. Questo rende la gestione degli IP complicata, perché bisogna far usare ad un host mobile il proprio indirizzo IP sulla LAN di appartenenza ovunque si trovi, per fare ciò ci viene in auto l'**home agent**, il quale assegna temporaneamente alla macchina un **IP temporaneo**, dirottando il traffico in arrivo sull'IP della macchina all'IP temporanea effettuando un tunneling.

Capitolo 6

6.0 Introduzione

Il livello di trasporto si basa sul livello di rete e fornisce come servizio quello di trasportare dati da una macchina sorgente ad una destinataria, il tutto fornendo affidabilità ed indipendenza dalla reti fisiche usate.

6.1.1 Servizi forniti ai livelli superiori

Come detto in precedenza l'obiettivo del livello trasporto è quello di fornire un servizio di trasmissione efficace, affidabile ed efficiente in termini di costi.

Il servizio di trasporto, come quello di rete, si divide in:

- **servizio orientato alle connessione**: costituito dalle tre fasi: impostazione, trasferimento dati e rilascio;
- **servizio non orientato alla connessione**: anche se non largamente usato poiché è inefficiente crea una connessione per inviare un solo pacchetto e poi distruggerla;

Il codice del livello di trasporto è eseguito interamente sulle macchine dei client, mentre quello del livello di rete è eseguito interamente sui router. Tale livello difatti nasce per rendere il servizio di trasporto più affidabile rispetto al servizio offerto dal livello di rete. Grazie ad esso, infatti, i programmatori possono scrivere codice usando primitive standard, scrivendo così codice funzionante su reti diverse, come ad esempio:

Primitiva	Pacchetto inviato	Significato
LISTEN	(nessuno)	Si blocca fino a quando un processo cerca di connettersi
CONNECT	CONNECTION REQ.	Tenta di stabilire una connessione
SEND	DATA	Invia informazioni
RECEIVE	(nessuno)	Si blocca fino all'arrivo di un pacchetto DATA
DISCONNECT	DISCONNECTION REQ.	Questo lato desidera rilasciare la connessione

6.1.3 Le socket di Berkeley

Le primitive della socket di Berkeley sono un altro gruppo di primitive, utilizzate per TCP, per sistemi UNIX.

Primitiva	Significato
SOCKET	Crea un nuovo punto finale do comunicazione
BIND	Associa un indirizzo locale a una socket
LISTEN	Annuncia la capacità di accettare connessioni; fornisce la dimensione della coda
ACCEPT	Blocca il chiamante fino all'arrivo di un tentativo di connessione
CONNECT	Tenta in modo attivo di stabilire una connessione
SEND	Invia alcuni dati sulla connessione
RECEIVE	Riceve alcuni dati sulla connessione
CLOSE	Rilancia la connessione

Dal punto di vista del server esse sono:

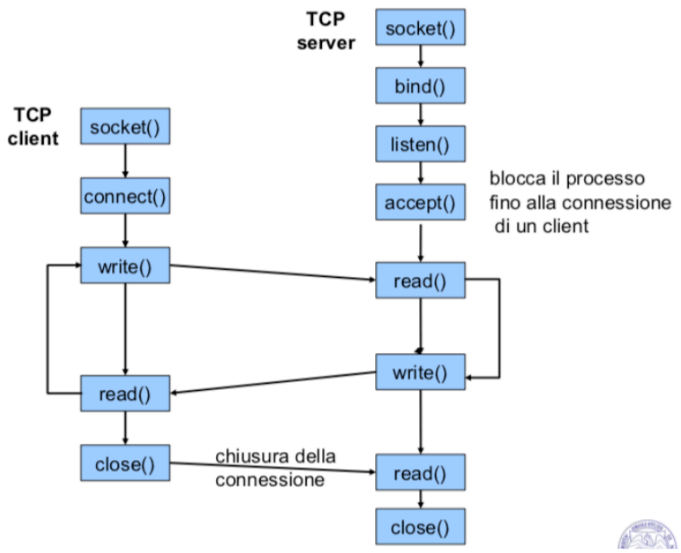
- **SOCKET**: crea una connessione, volendo come parametri il formato dell'indirizzo da usare (IPv4, IPv6), il tipo di servizio da usare (TCP, UDP) e il protocollo, se avviene correttamente crea un descrittore proprio come la chiamata OPEN su file;
- **BIND**: associa l'indirizzo di rete alla socket, non si lascia effettuare questo processo alla socket poiché alcuni servizi per anni hanno usato lo stesso indirizzo (IP-porta) e vogliono continuare ad usare quello;
- **LISTEN**: alloca spazio per accettare le richieste dai client, nel caso si volessero collegare più client contemporaneamente;
- **ACCEPT**: blocca la socket in attesa di connessioni, una volta ricevuta la richiesta di connessione il server può decidere di generare un nuovo processo o un nuovo thread per gestire la connessione e lasciare quindi il server in ascolto sulla socket originale, anche la socket restituisce un descrittore proprio come si fa con i file;

Dal punto di vista del client esse sono:

- **SOCKET**: anche il client dovrà creare un descrittore della socket, ma non servirà assegnare un indirizzo ad esso con la BIND;
- **CONNECT**: avvia la connessione con il server

Entrambi usano invece:

- **SEND** e **RECEIVE**: per trasmettere o ricevere dati;
- **CLOSE**: le risorse vengono rilasciate quando entrambi i lati hanno effettuato una close;

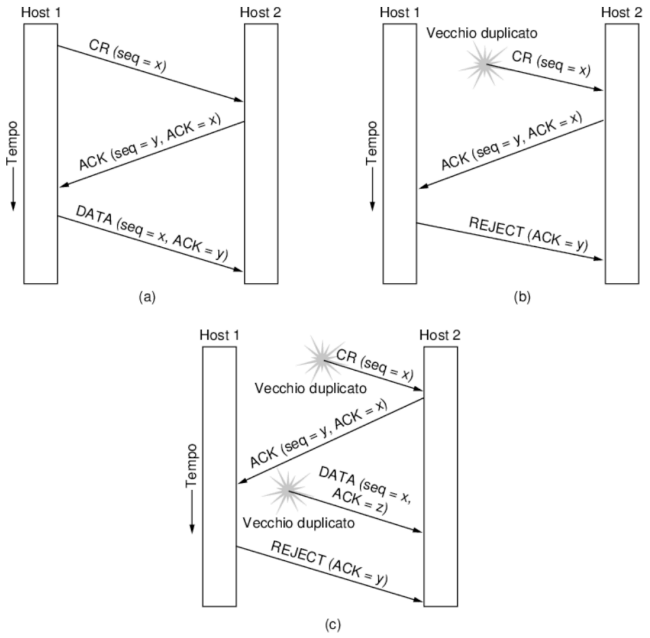


6.2.1 Indirizzamento

Quando un processo intende creare una connessione verso un altro processo remoto deve specificare a quale intende connettersi, per fare ciò generalmente al livello di trasporto si usano indirizzi di trasporto su cui i processi di rete restano in ascolto, questi indirizzi sono chiamati **porte**, gli omonimi indirizzi di rete invece si chiamano indirizzi IP. Quindi al livello di trasporto abbiamo le porte ed al livello di rete abbiamo gli indirizzi IP.

6.2.2 Stabilire una connessione

Una delle tecniche più usate per effettuare una connessione è chiamata **handshake a tre vie** (three way handshake).



Essa in genere prevede tre scenari:

- Nel primo scenario, quello ideale, HOST 1 richiede una connessione (CR), HOST 2 accetta la connessione inviando un segmento di ACK, a questo punto stabilita la connessione HOST 1 invia i dati ad HOST2;
- Nel secondo scenario, ovvero quando avvengono duplicati, l'HOST 2 riceve una richiesta di connessione da parte di HOST 1 senza che lui lo sappia, quindi invia un segmento di ACK all'HOST 1 ma quest'ultimo non volendo effettuare una connessione lo rifiuta ed interrompe la connessione;
- Nel terzo scenario, quello peggiore, ovvero quando in rete circolano sia duplicati di una richiesta di connessione che di un segmento di ACK, a questo punto gli HOST risolvono tale problema andando a vedere se la richiesta di connessione e l'ACK corrispondono ed hanno lo stesso numero di sequenza;

6.2.3 Rilascio della connessione

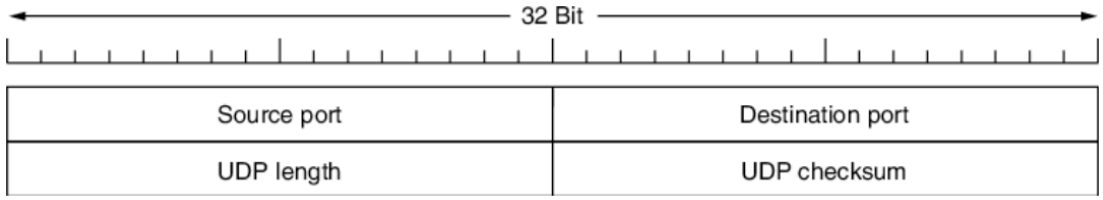
La chiusura di una connessione può sembrare semplice, ma non basta che entrambi gli host comunichino la volontà di interrompere la connessione, poiché tale volontà, proprio come i dati, può essere persa o arrivare in ritardo. Per ovviare a questo problema si usa la stessa tecnica del three-way-handshake dove il tutto viene sincronizzato con un ACK.

6.4 Protocolli di trasporto di Internet: UDP

Nel livello di trasporto Internet possiede due protocolli principali: un protocollo non orientato alla connessione, UDP, e uno orientato alla connessione, TCP. UDP non fa praticamente altro che spedire pacchetti tra le applicazioni mentre TCP fa praticamente tutto: crea connessioni e aumenta l'affidabilità con le ritrasmissioni, implementa anche il controllo di flusso e il controllo di congestione, tutto per conto delle applicazioni che lo usano.

6.4.1 UDP

UDP, acronimo di user datagram protocol, offre la possibilità alle applicazioni di inviare datagrammi senza stabilire una connessione. Difatti i suoi datagrammi hanno la seguente configurazione:



I primi 16 bit sono occupati dalle due porte, quella del mittente e quella del destinatario, i successivi 8 bit contengono la dimensione del datagramma ed i dati da inviare ed infine gli ultimi 8 bit contengono in maniera opzionale il checksum per aumentare l'affidabilità.

6.4.2 Remote procedure call

In un certo senso l'invio di un messaggio ad un host remoto e l'attesa della risposto corrisponde in qualche modo all'esecuzione di una chiamata a una funzione. Questa osservazione ha portato a gestire le interazioni richiesta/risposta sulle reti come chiamate a procedure. Un esempio può essere immaginiamo un funzione get_IP_address (host_name) che funziona tramite l'invio di un pacchetto UDP ad un server DNS, il quale ricevuto il nome dell'host restituisce il relativo indirizzo IP, facendo il tutto senza che il programmatore sappia nulla. Questa tecnica prende il nome di **RPC**, acronimo di **Remote Procedure Call**.

6.4.3 Protocolli di trasporto real-time

Un'altra dove UDP è molto diffuso riguarda le applicazioni multimediali real-time. **RTP**, acronimo di **real time transport protocol**, nasce dall'esigenza di creare un protocollo generico per le applicazioni real time, dato il loro esponenziale aumento negli ultimi anni, vedi app per streaming video/audio, app per fare videoconferenza e molte altre.
Tale protocollo si base su due parti: la prima è l'invio di pacchetti UDP ad uno o più destinatari, il secondo riguarda la sincronizzazione di essi poiché come ben sappiamo in UDP non c'è la certezza che i tutti i pacchetti vengano spediti correttamente. Il formato RTP ha numerose caratteristiche che aiutano il destinatario ad elaborare correttamente i dati ricevuti, come un campo chiamato **sequence number**, il quale è un contatore dei datagrammi inviati dal trasmittente, il ricevente quindi in caso si ritrovi in una situazione in cui mancano datagrammi può decidere come agire, ad esempio nel caso di una chiamata è meglio trasmettere i datagrammi successi piuttosto che aspettare i datagrammi persi.

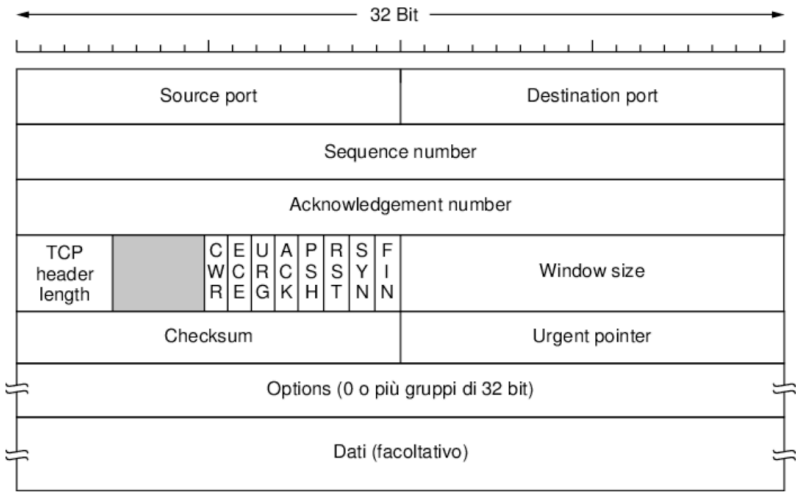
6.5 Protocolli di trasporto di Internet: TCP

UDP è un protocollo semplice con alcuni utilizzi molto importanti, come le interazioni client-server e le trasmissioni multimediali, ma per la maggior parte delle applicazioni Internet è necessaria una consegna affidabile, pertanto è richiesto un altro protocollo, chiamato TCP.

6.5.1 TCP

TCP, acronimo di **transmission control protocol**, è stato progettato appositamente per fornire un flusso di byte affidabile end-to-end su una internetwork inaffidabile. Una comunicazione TCP si ottiene tra due punti chiamati **socket**, identificati da un indirizzo IP ed un numero di porta. I numeri di porta minori di 1024 sono riversati ai servizi standard, come la porta 22 per il servizio SSH (login remoto), la porta 80 per il servizio HTTP o la 20 e 21 per il servizio FTP (trasferimento di file), dalla porta 1024 alla 65.000 circa possono essere registrate presso un servizio chiamato **IANA**. Tutte le connessioni TCP sono di tipo full-duplex ovvero il traffico procede in entrambe le direzioni.

6.5.4 Intestazione del segmento TCP



Ogni segmento inizia con un'intestazione di 20 byte che contiene i seguenti campi:

- **source port/destination port**: i primi due campi sono occupati dalle due porte, quella del mittente e quella del destinatario
- **sequence number**: è un contatore dei pacchetti inviati dal trasmittente, il ricevente così può controllare se gli sono arrivati tutti i pacchetti;
- **acknowledgement number**: svolge la sua solita funzione;
- **TCP header lenght**: indica quanti gruppi di 32 bit sono contenuti nell'intestazione;
- **4 bit inutilizzati**: solo 2 dei 6 bit inutilizzati originali sono stati utilizzati in 30 anni, questo ci fa capire quanto TCP sia stato sviluppato bene;
- **8 flag di un bit**: usati per segnalare la congestione;
- **windows size**: quanti byte possono essere inviati a partire da quello che ha ricevuto l'acknowledgement;
- **checksum**: esattamente come in UDP viene usato per migliorare l'affidabilità;
- **options**: campi usati per aggiungere funzionalità;

6.5.5 Instaurazione della connessione tcp

Una connessione TCP avviene mediante il metodo three-way-handshake.

Nello specifico il client genera una richiesta CONNECT e crea un record per la connessione impostandolo nello stato SYN SENT ed invia un SYN ed un ACK al server, il server invece inizialmente si trova nello stato LISTEN in attesa i connessione, quando riceve il SYN e l'ACK da parte del client passa allo stato SYN RCVD ed invia un SYN ed un ACK al client, nel momento in cui anche il SYN e l'ACK del server raggiungono il client la connessione passa allo stato ESTABLISHED ed è possibile inviare e ricevere dati.

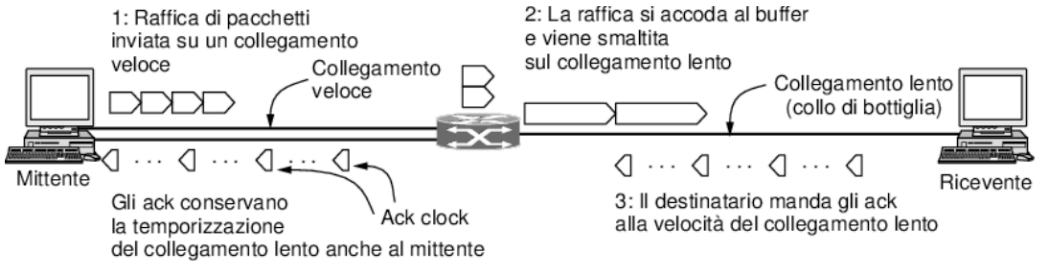
6.5.9 Gestione dei timer di TCP

TCP utilizza numerosi timer, tra cui il più importante è l'**RTO**, acronimo di **retrasmission timeout**, timer che se scade prima dell'arrivo dell'acknowledgement fa procedere ad una nuova trasmissione del pacchetto.

6.5.10 Controllo della congestione di TCP

Il livello di rete rileva la congestione quando le code dei router aumentano e risolve scartando dei pacchetti, al contrario il livello di trasporto rilevata una possibile congestione cerca di gestirla magari limitando il traffico sulla rete. Per farlo, TCP, usa una **finestra di congestione** la cui dimensione è pari al numero di pacchetti che la rete può gestire in quel momento. Tale finestra però è in aggiunta alla finestra di controllo di flusso originale e si sceglie di usare quella che tra le due ha il valore più basso, per evitare congestione. Esempi di controllo di congestione sono i seguenti:

1. situazione in cui il mittente si trova su un collegamento veloce e trasmette in maniera veloce, il ricevente, al contrario, si ritrova su un collegamento lento, per regolare il flusso in modo che non si crei congestione si utilizza una temporizzazione che prende il nome di **ack clock**, la quale indica che una volta che il ricevente riceve un pacchetto invia un ack al mittente, il quale ricevendone diversi per i pacchetti inviati capirà la velocità di ricezione del ricevente e si adeguerà a quella velocità per inviare i suoi pacchetti (ricevo un ack al secondo dovrò inviare un pacchetto al secondo).



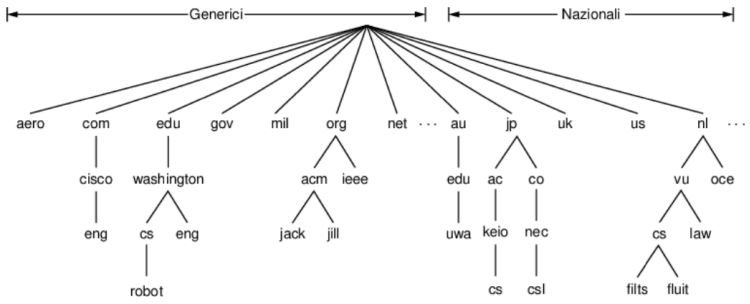
2. la finestra di congestione si imposta inizialmente ad 1 pacchetto, man mano che il mittente riceve in maniera corretta gli ack da parte del ricevente per i pacchetti inviati la finestra di congestione aumenta di un pacchetto;

Jacobson unì queste due tecniche ideando un algoritmo che prende il nome di **slow start**, ma non è lento per niente, ha una crescita esponenziale, difatti, imposta la finestra di congestione iniziale ad un massimo di quattro pacchetti, partendo dall'esempio che all'inizio invierà un segmento che contiene un pacchetto, se riceve in maniera corretta l'ack da parte del ricevente invierà due segmenti con un pacchetto ciascuno, se riceve in maniera corretta gli ack da parte del ricevente invierà quattro segmenti con un pacchetto ciascuno, se riceve in maniera corretta gli ack da parte del ricevente invierà otto segmenti con un pacchetto ciascuno, e così via, raddoppiando ad ogni passo il numero di pacchetti inviato fino al raggiungimento del massimo

Capitolo 7

7.1 DNS - domain name system

Anche se i programmi possono raggiungere le risorse in rete tramite i loro indirizzi di rete, ad esempio mediante il loro indirizzo IP, per la gente comune è difficile ricordare per ogni servizio il suo indirizzo IP, senza contare il fatto che se il server che eroga un servizio venisse spostato su un'altra macchina con un IP diverso l'azienda che fornisce tale servizio dovrebbe informare tutti del cambiamento. Per risolvere tale problema ad ogni servizio viene fornito un nome, ma poiché la rete funziona secondo indirizzi sono necessari meccanismi per convertirei i nomi in indirizzi di rete. Ai tempi di ARPANET tale meccanismo consisteva in un file host.txt il quale conteneva per ogni host il relativo indirizzo, con l'espansione di internet tale file iniziava a presentare problemi, il primo era che il file stava diventando troppo grande, il secondo era rappresentato dal fatto che tale file veniva aggiornato una sola volta al giorno, di notte, creando innumerevoli conflitti tra i nomi degli host. Per risolvere tale problema nel 1983 fu inventato il **DNS**. Tale sistema si basa uno schema di denominazione gerarchico, ed il suo funzionamento è al quanto semplice da capire, per associare un nome ad un indirizzo IP un programma invoca una procedura chiamata resolver, la quale ricevuto come parametro il nome dell'host, invia un pacchetto UDP al server DNS locale, che cerca il nome all'interno di un database e restituisce l'indirizzo IP corretto al resolver.
I nomi, e quindi i domini, vengono assegnati agli host da un ente chiamato **ICANN**, acronimo di **Internet corporation for assigned names and numbers**, ente no-profit ideato nel 1998 per evitare problematiche economiche. Come detto inizialmente i domini sono gerarchici, difatti in alto alla gerarchia troviamo i domini di primo livello, divisi in generici e nazionali.



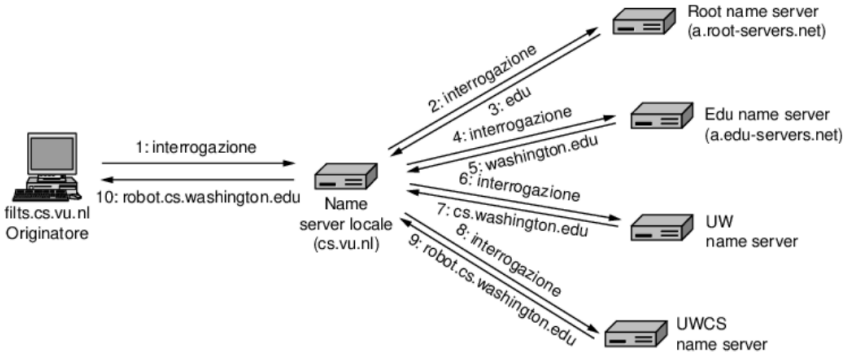
Ogni dominio di primo livello riguarda un ramo dell'Internet, come:
- **com**: uso commerciale
- **edu**: educazione e ricerca
- **gov**: pubblica amministrazione, governo
- **it-us**: indica domini nazionali italiani e americani in questo caso

Un esempio di dominio che riguarda la divisione ingegneristica di Cisco potrebbe essere **eng.cisco.com**.
All'interno del database DNS i domini sono rappresentati da record formati da cinque campi, e sono i seguenti:
- **Domain_name**: indica il dominio, quindi il nome dell'host a cui si riferisce il record.
- **Time_to_live**: Indica il tempo per cui quelle informazioni resteranno valide, per servizi estremamente stabili tale valore è 86.400 secondi, ovvero un giorno.
- **Class**: indica se tali informazioni riguardano Internet, in questo caso assume il valore di IN, o altri campi.
- **Type**: specifica che tipo di record, e presenta vari valori come **A** se si riferisce ad un indirizzo IPv4, **CNAME** se si riferisce ad un nome del dominio.
- **Value**: fornisce informazioni riguardo il record, ad esempio se è un indirizzo IPv4 nel campo value ci verrà specificato che si tratta di un indirizzo a 32 bit.

Esempio di record è: cs.mit.edu 86400 IN CNAME csail.mit.edu

Tale record è riferito al dipartimento di informatica del MIT, ed il record indica:
- cs.mit.edu: il nome dell'host;
- 86400:tale record resterà valido per un giorno
- IN: riguarda un servizio Internet
- CNAME: il record si riferisce ad un nome di dominio
- csail.mit.edu: indica il nome canonico del dominio

In teoria un solo server DNS potrebbe contenere l'intero database DNS ma ciò non sarebbe sostenibile date le numerose richieste, senza contare il fatto che in caso di malfunzionamento del server l'intera rete Internet resterebbe paralizzata. Per evitare complicazioni quindi la rete è divisa in zone ed ad ogni zona sono associati uno o più name server.

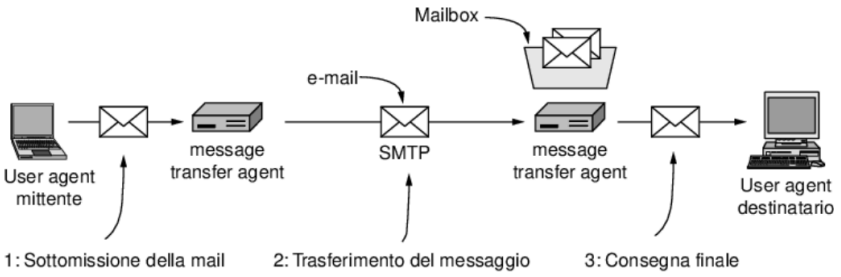


Nell'immagine sovrastante vediamo come un host di nome filts esegue una richiesta ad un name server locale riguardo la risoluzione di un dominio chiamato cs.vu.nl. Il name server quindi inizierà ad interrogare i vari name server ed in 10 passi avrà la soluzione. In questo scenario ci sono tre questioni tecniche di cui parlare:

1. il primo punto riguarda il fatto che l'interrogazione avviene mediante due meccanismi di risoluzione:
    1. l'host invia la richiesta di risoluzione al name server locale, il quale non restituirà risposte parziali ma continuerà ad interrogare name server finché non conoscerà la risposta alla domanda effettuata dall'host filts, tale meccanismo è chiamato **interrogazione ricorsiva**;
    2. tutta via il root name server non continuerà l'interrogazione ricorsiva fino a trovare la risposta ma invierà una risposta parziale e passerà all'interrogazione successiva, tale meccanismo prende il nome di **interrogazione iterativa**;
2. il secondo punto riguarda il **caching**, tutte le risposte, comprese quelle parziali, vengono salvate nella cache, in modo che se un altro host pone la stessa domanda già si conosce la risposta;
3. Il terzo punto riguarda il protocollo usato per le interrogazioni, UDP. Se tale protocollo fallisce l'invio della risposta più volte, il name server proverà ad interrogare un altro name server provocando rallentamenti nella risoluzione del dominio.

7.2 Posta elettronica

La posta elettronica è disponibile da oltre 40 anni, essendo più veloce ed economica della posta tradizionale divenne subito popolare. I protocolli e-mail si sono evoluti nel corso degli anni, agli inizi, infatti, erano semplici protocolli di trasferimento per file testuali con la differenza che la prima riga di ogni messaggio conteneva l'indirizzo del destinatario. L'architettura del sistema è abbastanza semplice, ci sono due **user agent** rappresentati dalla applicazioni che consentono di leggere ed inviare le mail agli utenti, due **message transfer agent** che spostano i messaggi dalla sorgente al destinatario ed il tutto facendo uso di **SMTP**, acronimo di **simple mail transfer protocol**, protocollo che invia le e-mail con un approccio orientato alla connessione e notifica lo stato della consegna o i relativi errori.



Il formato dei messaggi della posta elettronica è descritto come RFC 5322, il quale ha diversi campi nell'intestazione.

Intestazione	Significato
To:	Gli indirizzi di posta elettronica dei destinatari primari
Cc:	Gli indirizzi di posta elettronica dei destinatari secondari
Bcc:	Gli indirizzi di posta elettronica per le copie per conoscenza nascoste
From:	La persona che ha creato il messaggio
Sender:	L'indirizzo di posta elettronica del mittente vero e proprio
Received:	Riga aggiunta da ogni transfer agent lungo il percorso
Return-path:	Può essere utilizzato per identificare un percorso di ritorno al mittente

L'invio finale del messaggio avviene tramite un protocollo chiamato IMAP (Internet message access protocol), protocollo che offre il suo servizio tramite un server in ascolto sulla porta 143. Tale protocollo è un aggiornamento del protocollo POP3, ed a differenza di quest'ultimo offre più funzionalità e sicurezza. Esso infatti instaura una connessione di trasporto sicura effettuando un login sul server, una volta effettuato l'accesso offre diversi servizi come trovare un messaggio, fare una copia di un messaggio o eliminarlo.

7.3 Il World Wide Web

Il Web, appellativo con cui viene definito il **World Wide Web**, è un infrastruttura che consente l'accesso a documenti sparsi su milioni di macchina in Internet. Il Web nacque nel 1989 al CERN di Ginevra, idea del fisico **Tim Berners-Lee**, con l'intento di collegare i ricercatori di tutto il mondo per condividere gli studi. Il primo prototipo fu presentato 18 mesi dopo, nel 91 ad una conferenza chiamata Hypertext in Texas, la dimostrazione pubblica fu talmente apprezza che **Marc Andreessen** dell'Università dell'Illinois decise di sviluppare il primo browser grafico, che preso il nome di **Mosaic** e fu reso disponibile al pubblico nel febbraio del 1993. Ad oggi il Web è per gli utenti comuni una vasta raccolta mondiale di **pagine Web** accessibili mediante un programma chiamato **browser**. Il browser permetteva di risolvere e rispondere alle tre domande che si ponevano i pionieri di internet per raggiungere una pagina Web, ovvero come si chiamasse la pagina, dov'era situata e come si poteva accedere ad essa. Le soluzione a queste domande furono l'**URL** (uniform resource locator) per identificare univocamente ciascuna pagina Web, mentre il **DNS** si occupava di rispondere alle esigenze delle altre due domande.
Da un punto di vista strutturale il web è composto da due figure principali, il **client** che richiede il servizio ed il **server** che offre il servizio richiesto dal client.
Il client per recuperare una pagina web svolge i seguenti passaggi:

1. il browser determina l'URL selezionato dal client;
2. successivamente richiede al DNS l'indirizzo IP del server che corrisponde a quell'URL;
3. il DNS restituisce l'IP richiesto dal browser;
4. il browser esegue una connessione TCP alla porta 80 sull'indirizzo IP indicato dal DNS;
5. invia una richiesta HTTP al server richiedendo la home della pagina web;
6. il server in ascolto sull'indirizzo IP risponde, sotto forma di risposta HTTP, restituendo la risorsa richiesta dal client;
7. a questo punto il browser mostra la risorsa richiesta al client;
8. se non ci sono ulteriori richiesti da parte del client per quel server specifico la connessione TCP viene rilasciata;

Il server invece per restituire la risorsa richiesta dal client esegue i seguenti passaggi:

1. accetta la connessione TCP dal client;
2. ottiene il percorso della pagina richiesta dal client;
3. ottiene il file relativo alla pagina web che si trova sul disco;
4. invia il contenuto del file al client;
5. se non ci sono ulteriori richiesti da parte del client per il server la connessione TCP viene rilasciata;

Navigare nel web quindi come abbiamo visto consiste nel recuperare pagine web che si trovano su macchine remote, e poiché nel Web non esiste il concetto di sessione, ovvero il server una volta servito il client dimentica di aver mai visto quel particolare client, fu introdotto un meccanismo che consentisse ai server di tenere traccia delle scelte da parte dei client, tale meccanismo prese il nome di **cookie**.

Come indicato in precedenza la comunicazione lato web consiste in richieste **HTTP**. Tale protocollo, acronimo di **hypertext transfer protocol**, è semplice e si basa sul modello di domande e risposta (request-response), normalmente implemento da TCP. Tali richieste non sono eseguite unicamente da browser Web ma ad esempio un software antivirus potrebbe usare HTTP per scaricare gli ultimi aggiornamenti o sviluppatori potrebbero impiegare HTTP per recuperare file di progetti. La connessione come detto avviene mediante il protocollo TCP sulla porta 80 della macchina server, agli inizi, con HTTP 1.0 dopo aver stabilito la connessione veniva eseguita una sola richiesta e successivamente la connessione veniva interrotta. Tale pratica, fu subito un limite quando i servizi, e le pagine Web spesso presentavano collegamenti ad altri servizi o pagine Web, fu così che con HTTP 1.1 si integrò il supporto a connessione persistenti, quindi una volta eseguita la connessione era possibile effettuare più richieste da parte del client ed effettuare più risposte da parte del server.

HTTP lasciò inoltre possibilità di sviluppi futuri attraverso il supporto a varie operazione chiamate **metodi**, di seguito ne vediamo alcuni:

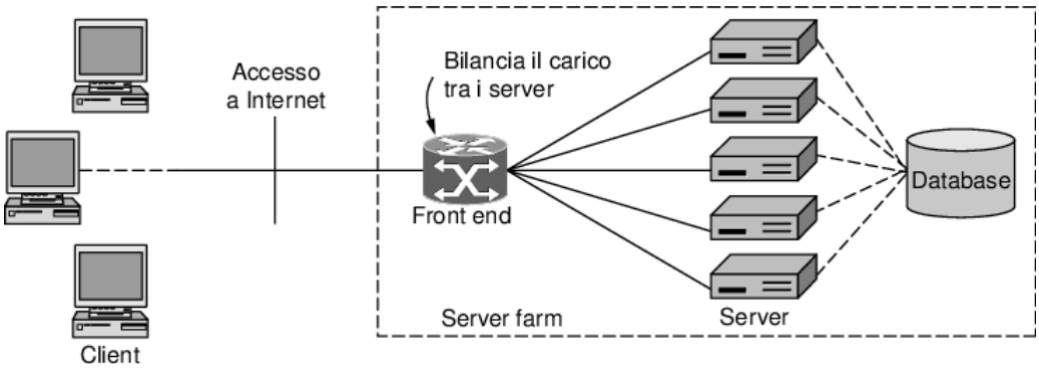
Metodo	Descrizione
GET	Legge una pagina Web
HEAD	Legge l'intestazione di una pagina Web
POST	Aggiunge a una pagina Web
PUT	Memorizza una pagina Web
DELETE	Rimuove la pagina Web
TRACE	Visualizza la richiesta in ingresso
CONNECT	Si collega tramite un proxy
OPTIONS	Richiede le opzioni di una pagina

Ogni richiesta veniva inoltre risposta con una riga di stato o di errore in caso di malfunzionamento, di seguito ne vediamo alcune:

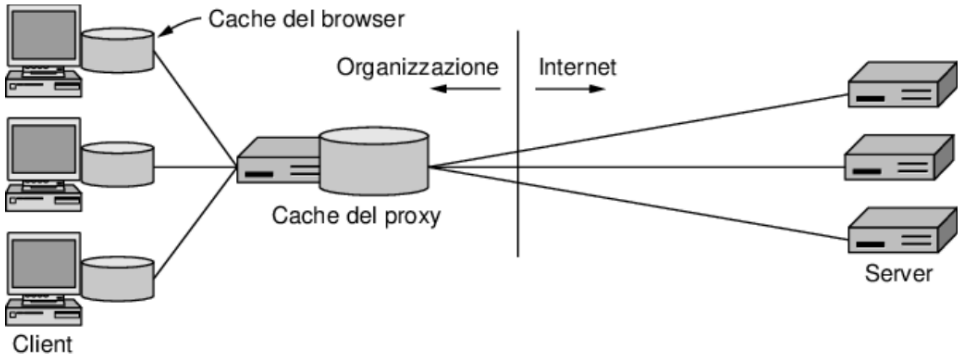
Codice	Significato	Esempi
1xx	Informazione	100 = il server accetta di soddisfare la richiesta del client
2xx	Successo	200 = richiesta eseguita con successo; 204 = nessun contenuto presente
3xx	Reindirizzamento	301 = pagina spostata; 304 = pagina nella cache ancora valida
4xx	Errore del client	403 = pagina vietata; 404 = pagina non trovata
5xx	Errore del server	500 = errore interno del server; 503 = riprovare più tardi

Per velocizzare la disponibilità delle pagine Web per l'utente sono state introdotto numerose tecniche, come:

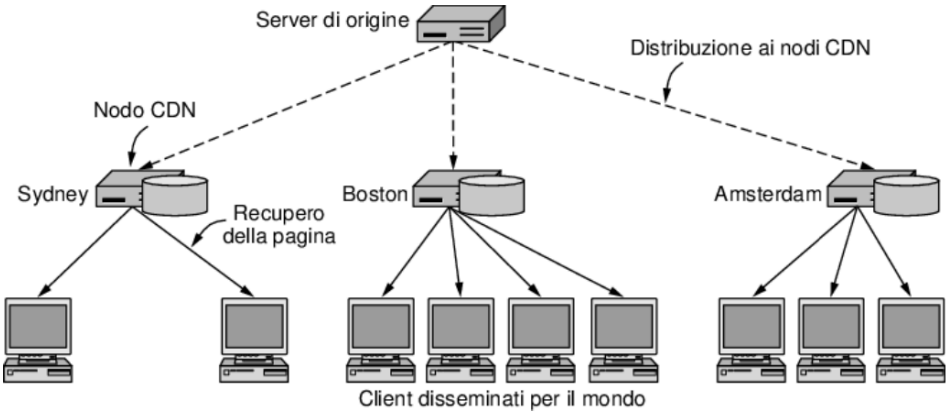
1. è l'utilizzo del **caching**, esso consiste nel salvare le pagine Web in una memoria cache in modo tale che se l'utente richiede la stessa pagina Web più volte essa è subito disponibile.
2. Per rispondere alle odierne numerose richieste di pagine web è stato presto introdotto il modello **server farm**, ovvero un gruppo di server capace di smistarsi le richieste rivolte ad una risorsa per evitare il sovraccarico.



3. Un'altra tecnica per ovviare a lunghe attese per ricevere una pagina Web è il concetto di **proxy**, esso è usato per fare condivisione di una cache tra utenti.



4. Le **CDN** (*content delivery network*) sovvertono l'idea della cache Web tradizionale. Invece di avere dei client che cercano una copia della pagina richiesta in una cache vicina, nelle CDN è il provider a mettere una copia della pagina in un insieme di nodi in differenti posizioni e a dirigere il client in modo che usi un nodo a lui vicino come server



Reti peer-to-peer

Non tutti possono permettersi una CDN di 1000 nodi sparsi in tutti il mondo, a fronte di ciò esiste un'alternativa semplice da usare in grado di distribuire grandi quantità di contenuti: una rete **P2P** (**peer-to-peer**). Tali reti nascono nel 1990 e la loro prima applicazione fu un crimine di massa, 50 milioni di utenti di Napster si scambiavano musica coperta da copyright senza il permesso dei proprietari. L'idea alla base è che molti computer si uniscano e mettano le loro risorse in comune per formare un sistema di distribuzione di contenuti. Le reti P2P, grazie alla loro configurazione hanno un incredibile capacità di distribuire contenuti, questo grazie al fatto che se una rete P2P è composta da N utenti, ciascuno di essi dispone di una banda larga da 1 Mbps, allora quella rete potrà condividere alla velocità di N Mbps.
Uno dei protocolli più diffusi è **BitTorrent**, sviluppato da Cohen nel 2001 per permette a dei peer di condividere dei file rapidamente e facilmente. Per fare ciò il creatore si è posto tre domande:

1. come fa un peer a trovare altri peer che abbiano il contenuto che lui intende scaricare;
2. come si replica il contenuto tra i peer per far sì di ottenere alta velocità di trasmissione;
3. come incoraggiare i peer a caricare contenuti per gli altri;

Il primo problema venne risolto creando una descrizione del contenuto di ogni peer chiamato **torrent**, esso è un descrittore che specifica il contenuto del file e ne facilità quindi la ricerca. Tale file contiene due informazioni chiavi, il **tracker**, ovvero il server che conduce i peer al contenuto del torrent, ed il chunk, ovvero un elenco dei pezzi che insieme costituiscono la risorsa richiesta dal peer.
Il secondo problema è risolto dal fatto che un peer una volta collegato prende il nome di **seeder** e otterrà i chunk che gli mancano dagli altri seeder ma allo stesso tempo condividerà i chunk che già possiede e che servono agli altri.
Il terzo problema è stato risolto premiando i peer che si offrivano di essere anche seeder fornendogli velocità maggiori.

Capitolo 8

8.1 Crittografia

La parola crittografia deriva dal greco e significa parola segreta. Introduciamo ora alcuni termini tecnici della crittografia:

- **testo in chiaro**: il messaggio da cifrare;
- **testo cifrato**: il messaggio in chiaro cifrato;
- **criptoanalisi**: l'arte di forzare gli algoritmi di cifratura;
- **crittologia**: l'arte di inventare nuovi algoritmi;
- **decriptare**: l'arte di forzare un algoritmo e trovare la soluzione;
- **decifrare**: l'arte legittima di leggere un messaggio cifrato;

Tutti gli algoritmo rispondono al **principio di Kerckhoff**, il quale afferma che tutti gli algoritmi devono essere pubblici e solo le chiavi devono essere segrete.

Ci sono inoltre due principi crittografici alla base dello studio dei sistemi crittografici e sono:

1. **ridondanza**: tale principio afferma che tutti i messaggi cifrati devono contenere ridondanza, ovvero contenere una parte d'informazione non necessaria alla comprensione del testo. Facciamo un esempio di un'azienda che memorizza gli ordini dei suoi clienti cifrati nel seguente modo: 16 byte per il nome del cliente e 3 byte per i dati dell'ordine (1byte per la quantità e 2 byte per il numero del prodotto). Se un dipendente licenziato per vendicarsi ruba i dati riguardanti i nomi dei clienti e successivamente riempie i 3 byte a caso è così in grado di inviare innumerevoli ordini fasulli. Per l'algoritmo di cifratura il messaggio cifrato è corretto, poiché rispetta le caratteristiche e quindi l'azienda si troverà con migliaia di ordini fasulli. Per risolvere tale problema il principio di ridondanza dice che invece di usare solo 3 byte per i dati bisogna usarne 12, 3 per i dati effettivi ed i restanti 9 impostati a 0, così facendo se il dipendente non conosce la disposizione di tali 0 non può forzare l'algoritmo.
2. **attualità**: tale principio afferma che è necessario avere un sistema che verifichi l'attualità del messaggio ricevuto, ovvero che sia stato trasmesso di recente. Questo per evitare che messaggi intercettati possano essere inviati di continuo sulla linea e quindi bloccare il sistema.

Vediamo ora alcuni dei sistemi crittografici più conosciuti, i cifrari:

- **cifrari a sostituzione**: uno dei più famosi cifrari a sostituzione è il **cifrario di Cesare**, attribuito a Giulio Cesare. Tale cifrario sostituisce ad empio la lettera A con la lettera D, la lettera B con la lettera E, la lettera C con la lettera F e così via.
- **cifrari a trasposizione**: uno dei più famosi cifrari a trasposizione è il **cifrario a trasposizione colonnare**. Tale cifrario indica che dobbiamo innanzitutto trovare una chiave, la quale non contenga duplicati di caratteri. Successivamente bisogna identificare le colonne, la colonna numero 1 sarà quella corrispondente alla lettera della chiave più vicina all'inizio dell'alfabeto, nel caso di chiave MEGABUCK la prima colonna sarà la colonna corrispondete alla lettera A, la seconda colonna alle lettera V, la terza colonna alla lettera C e così via.

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Testo in chiaro

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Testo cifrato

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

- **cifrario a blocco monouso**: Per prima cosa si prende una chiave formata da una stringa di bit generati a caso. Poi, si converte il testo in chiaro in un'altra stringa di bit, per esempio usando la rappresentazione ASCII dei i caratteri. Infine, si calcola lo XOR (*OR esclusivo*) delle due stringhe, bit per bit, tale tecnica è nota come blocco monouso. Nell'esempio, il messaggio 1 "I love you" viene convertito in ASCII a 7 bit. A quel punto il blocco monouso 1 viene scelto e usato per l'operazione di XOR con il messaggio per produrre il testo cifrato.

```
Messaggio 1:  1001001 0100000 1101100 1101111 1110110 1100101 0100000 1111001 1101111 1110101 0101110
Blocco 1:     1010010 1001011 1110010 1010101 1010010 1100011 0001011 0101010 1010111 1100110 0101011
Testo cifrato: 0011011 1101011 0011110 0111010 0100100 0000110 0101011 1010011 0111000 0010011 0000101
Blocco 2:     1011110 0000111 1101000 1010011 1010111 0100110 1000111 0111010 1001110 1110110 1110110
Testo cifrato 2: 1000101 1101100 1110110 1101001 1110011 0100000 1101100 1101001 1110110 1100101 1110011
```

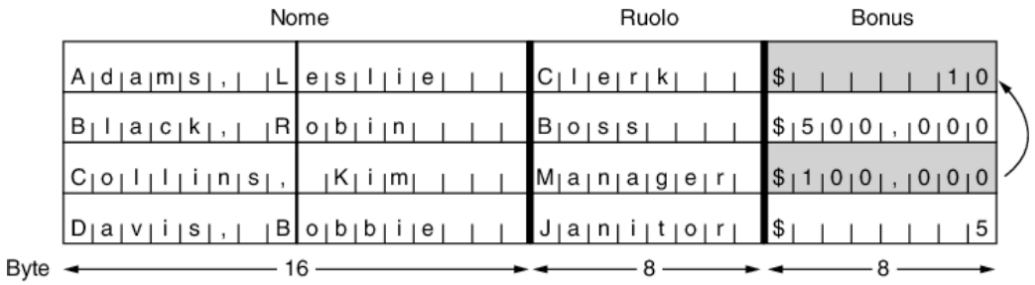
8.2 Algoritmi a chiave simmetrica

Gli algoritmi a chiave simmetrica prendono tale appellativo poiché usano la stessa chiave per cifrare e decifrare i messaggi.
Vediamo ora alcuni algoritmi:

- **DES** (**data encryption standard**): tale algoritmo è stato sviluppato nel 1977 dall'IBM per il governo degli Stati Uniti, esso cifra il testo in chiaro in blocchi di 64 bit che generano 64 bit di testo cifrato. L'algoritmo usa una chiave a 59 bit ed esegue 19 stadi distinti, ad esempio il primo consiste in una trasposizione dei 64 bit del testo in chiaro, l'ultimo stadio è esattamente l'opposto del primo, il penultimo consiste nello scambiare i 32 bit a sinistra con i 32 bit a destra ed i restanti 16 stadi sono funzionalmente identici ma parametrizzati da diverse funzioni della chiave.
- **Triplo DES**: Fin dal 1979 IBM si rese conto che la chiave del DES era troppo corta e quindi inventò il triplo DES. Esso era un sistema che usava 2 chiavi e 3 stadi, nel primo stadio il testo in chiaro veniva cifrato con DES usando la prima chiave, nel secondo stadio il testo cifrato nel primo stadio veniva decifrato usando il DES e la seconda chiave ed infine nel terzo stadio il testo ottenuto dal secondo stadio veniva nuovamente cifrato usando il DES e la prima chiave.
- **AES** (**advanced encryption standard**): nel 1997 fu la fine anche del triplo DES e furono messe le basi di un nuovo standard che avrebbe preso il nome di AES. Tale standard fu partorito da un concorso pubblico, il quale aveva le seguenti regole:
  1. l'algoritmo doveva essere un cifrario simmetrico a blocchi;
  2. la struttura doveva essere interamente pubblica;
  3. doveva supportare chiavi di 128, 192 e 256 bit;
  4. doveva essere possibile effettuare implementazioni software ed hardware;
  5. l'algoritmo doveva essere pubblico o rilasciato senza vincoli di alcun tipo;

  Tale concorso fu vinto dall'algoritmo **Rijndael**, tale algoritmo prevedeva sostituzioni e permutazione in iterazione, tante iterazioni quanti erano i bit della chiave, 10 iterazione per chiavi a 128 bit con blocchi da 128 bit, fino a 14 iterazioni per la chiave e il blocco più lunghi.

Tuttavia anche AES, nonostante la maggiore sicurezza, aveva lo stesso problema di tutti gli algoritmo citati in precedenza, prendiamo come esempio un testo in chiaro e suddividiamo in blocchi di 8 byte che poi vengono cifrati usando la stessa chiave, otterremo blocchi cifrati ma presenteranno un grande problema, prendendo l'esempio in basso a Leslie basterà copiare il contenuto del dodicesimo blocco nel terzo blocco per ottenere lo stesso bonus di Kim. La soluzione fu quindi di unire i blocchi in maniera irregolare in modo da evitare tali problemi.



8.3 Algoritmi a chiave pubblica

Ottenere un modo sicuro per trasmettere le chiave è sempre stato un problema così nel 1976 due studenti dell'Università di Stanford idearono metodo in cui la chiave di cifratura e decifrazione erano differenti ed anche gli algoritmi di cifratura e decifrazione erano differenti. Quindi la chiave e l'algoritmo di cifratura potevano essere resi pubblici ed un utente poteva usarli per inviare il suo messaggio cifrato alla persona che custodiva la chiave e l'algoritmo di decifrazione.
Un ottimo metodo è stato scoperto da un gruppo del M.I.T. (Rivest et al., 1978), ed è noto con le iniziali **RSA** dei suoi tre ideatori (Rivest, Shamir e Adleman).
RSA si basa su alcuni principi di teoria dei numeri. Faremo riepilogo sull'uso di questo metodo, mentre per i dettagli consigliamo di consultare l'articolo originale.
1. Scegliamo due numeri primi, p e q (tipicamente di 1024 bit).
2. Calcoliamo $n = p \times q$ e $z = (p - 1) \times (q - 1)$.
3. Scegliamo un numero relativamente primo rispetto a z, detto d. 4. Troviamo e tale che $e \times d = 1 \mod z$.
Avendo calcolato questi parametri in anticipo, possiamo cominciare la cifratura. Dividiamo in blocchi il testo in chiaro (visto come una stringa di bit), in modo che ogni messaggio in chiaro, P, cada nell'intervallo $0 < P < n$. Per fare questo, raggruppiamo il testo in chiaro in blocchi di k bit, dove k è il più grande intero per cui vale $2^k < n$. Per cifrare il messaggio P, calcoliamo $C = P^e \pmod n$. Per decifrare C, calcoliamo $P = C^d \pmod n$. Possiamo dimostrare che per ogni P nell'intervallo specificato, le funzioni di cifratura e decifratura sono una l'inversa dell'altra. Per cifrare abbiamo bisogno di e e di n; per decifrare invece ci servono d e n. Quindi la chiave pubblica consiste nella coppia (e, n), mentre la chiave privata consiste in (d, n).
La sicurezza del metodo è basata sulla difficoltà di scomporre in fattori primi i numeri molto grandi. Se il criptoanalista riuscisse a fattorizzare il numero (noto pubblicamente) n, allora riuscirebbe anche a trovare p e q e da questi anche z. Con la conoscenza di z, si possono trovare e e d tramite l'algoritmo di Euclide. Fortunatamente, i matematici si occupano della fattorizzazione dei grandi numeri da più di 300 anni e dalla conoscenza acquisita si sa che questo è un problema veramente difficile.

Secondo Rivest e colleghi, fattorizzare un numero di 500 cifre usando la forza bruta richiede 10^{25} anni, anche assumendo di usare il miglior algoritmo noto e un computer che elabora le istruzioni in 1 ms.
La possibilità di connettere qualunque computer che si trova in qualunque posto a qualunque altro computer in una qualsiasi località, può essere vista allo stesso tempo come una comodità e come una fonte di problemi. Per proteggere i dati in transito fu introdotto il concetto di firewall. I **firewall** (letteralmente, *muro di fuoco*) sono una versione moderna del vecchio rimedio medievale per le emergenze di sicurezza: scavare un profondo fossato attorno al proprio castello. Questa struttura obbligava chiunque volesse entrare o uscire dal castello a passare per un singolo ponte levatoio, dove la polizia poteva facilmente eseguire le sue ispezioni. Il firewall agisce come un **filtro per i pacchetti** (packet filter). Ispeziona ogni singolo pacchetto in entrata e in uscita. I pacchetti che rispecchiano i criteri descritti dalle regole formulate dall'amministratore di rete vengono inoltrati normalmente; quelli che falliscono il test vengono buttati via senza troppe cerimonie.
Una rete costituita da computer aziendali e linee telefoniche affittate ed ad uso esclusivo prende il nome di rete privata. Tuttavia il noleggio di una rete privata ha costi altissimi, difatti dopo l'introduzione delle reti pubbliche la maggior parte delle aziende ha migrato i suoi dati verso di esse. Tale scenario ha dato inizio ad una serie di problemi, come quelli per la sicurezza, la soluzione fu l'introduzione delle **VPN** (**virtual private network**), esse si sovrapponevano alle reti pubbliche fornendo alle aziende tutte le proprietà e la sicurezza delle reti private. **PGP** (pretty good privacy) usa un cifrario a blocchi per i dati chiamato **IDEA** (international data encryption algorithm), con una chiave a 128 bit. IDEA è stato sviluppato in Svizzera, quando DES era considerato compromesso e AES non era ancora stato inventato. Concettualmente, IDEA è simile a DES e AES: lavora mescolando i bit in una serie di iterazioni.