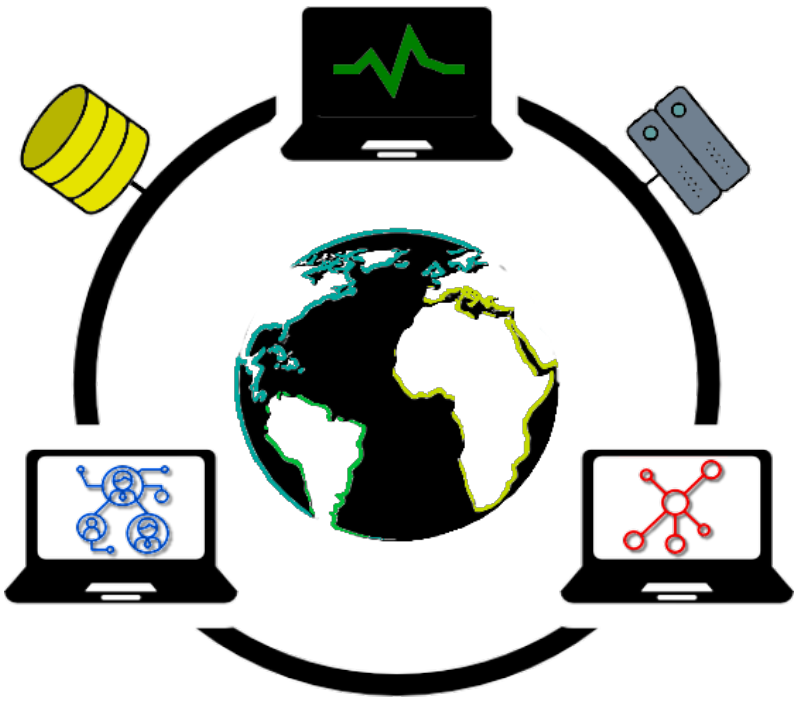
Appunti ragionati di

Reti di Calcolatori



Università Telematica Internazionale
UNINETTUNO

Ottobre 2018

Studente: Lorenzo L. Ancora
Matricola: 1953HHHINGINFO
Docente: Prof. Emanuel Weitschek

METADATI

Titolo: Appunti ragionati di Reti di Calcolatori
Autore: Lorenzo Lucio Ancora
Release: 07/10/18
Versione: 1.1.2
Download: http://ht.ly/TjOA30m4v5V

STATISTICHE

pagine: 124
paragrafi: 1044
parole: 27706
caratteri: 176738

Indice generale:

Materiale di supporto:

Visitare i link nella Prefazione per scaricare la versione più recente del materiale.

Intenzionalmente bianca.

Prefazione

Questo compendio riunisce una parte dei miei appunti, scritti per il corso di Reti di Calcolatori dell'Università Telematica Internazionale UNINETTUNO. Il testo segue fedelmente le videolezioni del docente *Mario Baldi* del Politecnico di Torino.

Gli appunti, da semplici file Markdown, sono stati rivisti e riformattati tramite programmi Open Source per consentire il ripasso veloce da parte di terzi pur mantenendo una quantità ragionevole di approfondimenti. Questa trattazione non sostituisce il libro di testo (lettura raccomandata: "Fondamenti di Reti di Calcolatori" di *Tanenbaum* e *Wetherall*, a cura di *S. Gaito* e *D. Maggiorini*) ed è stata pensata come supporto per coloro che hanno già seguito il corso universitario.

Il compendio è suddiviso in capitoli: trattando una materia naturalmente ricca ed eterogenea, ogni sezione può ribadire i concetti delle sezioni precedenti secondo diversi punti di vista e fornire spunti per collegamenti interdisciplinari. Il tutor o il docente suggeriranno l'ordine di studio e la metodologia da seguire.

Il fine di queste pagine è quello di risvegliare la curiosità dello studente verso le infrastrutture informatiche di uso quotidiano che, grazie al livello di affidabilità raggiunto, diamo tutti per scontate ma che in realtà sono frutto di decenni di perfezionamento da parte delle menti più eccelse.

Lorenzo Ancora

Scansiona il codice QR con il tuo smartphone, clicca sulle icone o digita l'URL nel web browser per ottenere la versione più aggiornata del compendio!

http://ht.ly/TjOA30m4v5V

Legenda

Apparenza	Significato
Testo in grassetto	Terminologia chiave rilevante rispetto al contesto. Termini da tenere a mente.
Testo in corsivo	Dettaglio che merita attenzione. Frasi da tenere in considerazione quando ci si smarrisce.
Testo comune Inserto contestuale Testo comune	Ammodernamenti, suggerimenti e considerazioni personali. Paragrafi che possono essere ignorati da chi non ha bisogno di approfondire.
Testo comune Inserto letterale Testo comune	Citazione e/o formula. Paragrafi che devono essere letti preferibilmente in modo letterale ed appuntati se necessario.
• Testo commento: ▪ Testo annidato commento annidato; • Testo;	Lista gerarchica. Gli elementi più annidati dipendono dall'elemento più esterno che li precede. Il commento rientrato si riferisce sempre all'elemento di lista che lo precede.
1. Testo commento; 2. Testo;	Lista gerarchica ordinale. Il commento rientrato si riferisce sempre all'elemento di lista che lo precede. Il numero di lista ha valore semantico rispetto al contesto.
`Codice sorgente`	Listato di codice sorgente o pseudo-codice.

Questa legenda *vale per tutti i capitoli* degli appunti.

1 • Introduzione alle Reti di Calcolatori

In una rete di calcolatori si condividono informazioni e risorse, che rimangono disponibili anche in caso di malfunzionamenti. Questo riduce il costo delle applicazioni perché diventa possibile unire la potenza di calcolo di più calcolatori, fare il bootstrap remoto, condividere software etc.

Per realizzare una rete servono protocolli standard, che stabiliscono il formato dei dati e la sequenza temporale di operazioni per scambiarli: se sono standard, grazie alla competizione del mercato aperto, si riducono i costi.

Gli standard delle reti sono stabiliti da:

- **ISO** (*International Organization for Standardization*):
  organizzazione internazionale legata al mondo delle reti di calcolatori;
- **IEEE** (*Institute of Electricals and Electronics Engineers*):
  organizzazione professionale legata all'elettronica;
- **ITU-T** (*International Telecommunication Unit - settore Telecomunicazioni*):
  organizzazione legata alle reti pubbliche.

Le reti private di calcolatori, oggetto del nostro studio, come tutte le altre reti riguardano l'ICT (IT+C, *Information and Communication Technology*).

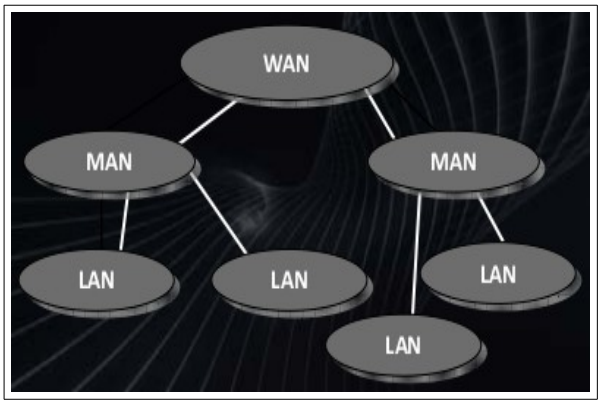
Gli ordini di grandezza per le reti sono:

Network on chip	0.01m	Chip
Massive Parallel	0.1 m	Scheda
Multi-Processor	1 m	Sistema
Cluster	10 m	Stanza
Local network	100 m	Edificio

Extended local network	1 km	Comprensorio
Metro network	10 km	Città
Reti georgrafiche	100 km	Nazione
Interconnessione di reti geografiche	1000 km	Continente
Interconnessione di reti geografiche	10.000 km	Pianeta

Tutte le reti, avendo caratteristiche differenti, richiedono accortezze diverse e noi ci occupiamo in particolare di:

- Standard ISO:
  - **LAN** (*Local Area Network*):
    raggiungono velocità maggiori di 100 Mb/s (Megabit al secondo) e dimensioni massime di qualche chilometro.
- Standard ITU-T:
  - **MAN** (*Metropolitan Area Network*):
    raggiungono velocità maggiori di 2 Mb/s ed hanno estensione cittadina;
  - **WAN** (*Wide Area Network*):
    raggiungono velocità molto elevate in base al costo ed hanno estensione globale.



Un canale di comunicazione può essere fisico o logico, in ogni caso suddiviso in più categorie d'uso:

- **punto-punto**:
  semplice canale mono-direzionale o bi-direzionale;
- multi-punto (obsolete):
  nodo *master* che controlla innumerevoli nodi;

- **broadcast** (moderne):
  innumerevoli nodi identificati da indirizzi, con ciascun nodo che trasmette verso tutti gli altri nodi.

La topografia di queste reti può essere:

- A stella:
  hub centrale con innumerevoli nodi ad esso collegati;
- Bus:
  senza *hub* ma con un bus fisico, condiviso da tutti i nodi ad esso collegati;
- Ad anello:
  reti *bus* circolari, ove ciascun calcolatore fa inoltro al successivo fino al raggiungimento del destinatario in uno dei due sensi, con elevata fault-tollerance;
- A maglia parziale:
  rete generica con elevata ridondanza;
- Satellitare:
  rete *a stella* sul pianeta terra che riceve trasmissioni broadcast dallo spazio.

La **multiplazione** è una tecnica che consente di condividere un canale tramite una suddivisione basata sul tempo, sulla frequenza o su codici. Nel caso della frequenza, se il canale è ottico (fibra), si parla di lunghezza d'onda tramite il fenomeno del *color shifting o meglio wavelength modulation*.

Grazie alla multiplazione, più segnali possono usare lo stesso canale perché due device opposti, il *multiplexer* ed il *demultiplexer* prevengono i conflitti, facendo in modo che ciascun calcolatore possa ricevere solo le informazioni a lui dirette. In particolare, il demultiplexer distingue i pacchetti grazie alla loro

intestazione, che permette l'uso statistico[1] del canale di comunicazione.

Esiste anche la tecnica TDM (*Time-Division Multiplexing*), che distingue i pacchetti in base alla loro posizione nel tempo ma richiede la perfetta sincronia tra ricevitore e trasmettitore e la trasmissione continua. Nonostante non permetta l'uso statistico, viene utilizzata sulle reti telefoniche classiche[2] che, proprio per questo, vengono tariffate a tempo e non a pacchetto.

La *commutazione* viene fatta da IS (*Intermediate Systems*) che, in base alle loro qualità, prendono il nome di *nodo*, *switch*[3] o *router*[4]. Al contrario dei comuni hub di rete, il loro compito è distinguere i pacchetti e smistarli solo verso i legittimi destinatari. Gli switch sono multiplexer che fanno *packet switching*: sono a tutti gli effetti calcolatori che smistano i pacchetti in base al loro header, facendo quindi *routing* verso canali diversi.

Nelle reti ottiche si usa il *circuit switching*, che può essere basato sul tempo o sulla frequenza. In questo caso il multiplexer ha vincoli di tempo strettissimi o, nel caso si basi sulla frequenza, necessita di tecnologie avanzate.

Oggigiorno la rete Internet è una WAN che utilizza sia comunicazioni con segnali ottici che a pacchetto ma, un giorno, si useranno solo comunicazioni a pacchetto.

1 Le informazioni vengono mandate solo se disponibili e necessarie per la sessione, con notevole risparmio di dati.
2 Siccome il costo del monitoraggio può essere elevato, molte aziende recenti hanno adottato la *tariffazione flat* (fissa) e richiedono semplicemente un canone mensile.
3 Ovvero "commutatore", non si traduce.
4 Ovvero "instradatore", non si traduce.

2 • Standard ISO/OSI

L'architettura di rete risolve il problema della comunicazione tramite strategia *divide et impera*. Ciascuna architettura protocollare è a livelli e, nell'architettura ISO/OSI, il livello N+1 si basa sui servizi del sottolivello N: ogni livello risolve un sotto-problema del divide et impera. I livelli adiacenti forniscono i propri servizi attraverso un'interfaccia intermedia, usata dal livello superiore tramite primitive apposite.

OSI sta per **Open System Interconnection**, in opposizione con le architettura proprietarie come **SNA** e **DECNET**[5], rispettivamente di IBM[6] e DEC[7]. ISO/OSI fa da modello sia per - l'ormai onnipresente - TCP/IP ("Transport Control Protocol over Internet Protocol", parti dell'**Internet Protocol Suite**) che per le altre architetture moderne, perché definisce la terminologia ed il metodo per creare le architetture[8].

In OSI due processi comunicano tramite due pile di protocolli, a cui si accede dal livello applicativo e fondate sul livello fisico. Ciascun **End Point** ha una pila così composta:

1. **Application**:
   livello applicativo, aggiunge il proprio header e dipende interamente dal processo che produce informazioni;
2. **Presentation**:
   livello di presentazione, aggiunge il proprio header;
3. **Session**:
   livello di sessione, aggiunge il proprio header;

5 Queste architetture proprietarie non sono ancora estinte perché i sistemi obsoleti perdurano, ma saranno sostituite da OSI e TCP/IP, nate in ambiente UNIX e divenute standard de facto.

6 “International Business Machines Corporation”®.

7 “Digital Equipment Corporation”®, pubblicizzata semplicemente come “Digital”®.

8 L'architettura ISO/OSI non viene quasi mai implementata completamente a causa della sua complessità: in pratica viene utilizzata come strumento di categorizzazione e confronto.

4. **Transport**:
   livello di trasporto, aggiunge il proprio header;
5. **Network**:
   livello di rete, aggiunge il proprio header;
6. **Data Link**:
   livello di collegamento dati, aggiunge il proprio header ed un trailer per segnare la fine della trama;
7. **Physical**[9]:
   livello di collegamento fisico, trasferisce i singoli bit della trama.

I dati passano per tutti i livelli, dall'applicazione trasmittente al livello fisico del ricevente, fino a giungere all'applicazione del ricevente, ove le applicazioni sono processi indipendenti. Due ES (**End Systems**) comunicano in genere tramite uno o più IS (**Intermediate Systems**) che fanno **routing** e trasportano i dati dal mittente al ricevente secondo la via più conveniente. Gli IS hanno perlomeno i livelli Data Link e Fisico e quelli a pacchetto anche il livello Network, ma possono fare a meno degli altri livelli superiori.

Il livello Network degli ES sceglie gli IS ideali tra i limitrofi, per ridurre il tempo di trasmissione scegliendo la via più veloce; i livelli che si basano su di esso lavorano sui pacchetti (frame o insiemi di frame), che nella terminologia OSI si chiamano **PDU** (**Protocol Data Unit**). In particolare il livello Network ha il fine di consegnare i pacchetti facendo "**forwarding**" (inoltro) e consentire così il **routing multi-nodo**, previa identificazione degli **EP** (**End Point**s) tramite gli indirizzi di rete.

Il livello Physical determina i mezzi fisici, i collegamenti e le caratteristiche di trasmettitore e ricevitore, che in OSI possono essere di produttori differenti.

9 Il livello fisico viene approfondito nel capitolo 3.

Il livello Data Link tratta gruppi di bit (detti "frames"), attuando così la **frame delineation**, che distingue i frame nel flusso di bit tramite **headers** e **trailers**. Questo livello implementa anche il **MAC** (**Media Access Control**) e gestisce gli errori di trasmissione rilevandoli, correggendoli e regolando il flusso sulle reti lente per evitare errori di congestione.

Il livello Transport opera solo End-to-End ed ignora l'esistenza degli IS. Il suo fine è poter trasferire messaggi (sequenze di bytes prodotte dalle applicazioni) senza doversi curare del concetto di pacchetto. Così facendo può gestire gli errori semantici, di flusso e congestione[10].
Il fine principale di questo livello è l'organizzazione dei pacchetti in un flusso coerente.

Il livello Session opera a livello di transazione, nascondendo gli errori di connettività e gestendoli in modo trasparente.

Il livello Presentation gestisce l'informazione e non il dato: adatta il formato dei dati traducendone la sintassi in modo che il destinatario li possa comprendere e trasformare in informazioni utili. Due funzioni fondamentali di questo livello sono la **cifratura di sicurezza** e la **negoziazione**.

In OSI si dice che le **PE** (**Protocol Entity**) realizzano le funzionalità di un livello tramite quelle del livello inferiore, indicate come "N-1 Entity". Queste entità comunicano tramite **SAP** (**Service Access Points**), atti a determinare le PE che forniscono i servizi richiesti usando gli indirizzi. I SAP possono essere su sistemi diversi, perché dispongono di indirizzi unici.

Nel modello a strati la comunicazione avviene per livelli. Ogni livello comunica con il livello adiacente attraverso il SAP, in particolare il livello N fornisce servizi al livello superiore (N + 1) usando i servizi del livello inferiore (N - 1) e le proprie funzioni.

Ciascuna PDU trasferita passa attraverso tutte le PE di ciascun livello. Appena la PDU entra in un SAP viene considerata una **SDU**

10 Su alcuni sistemi questi errori vengono gestiti direttamente nei livelli inferiori, per questioni di performance.

(**Service Data Unit**) e, solo dopo che la PE del SAP ha aggiunto il proprio header **PCI** (**Protocol Control Information**), torna ad essere considerata una PDU e può essere consegnata al successivo SAP. Questo processo si chiama **imbustamento**.

I servizi che non si basano sulle connessioni (**connectionless**) usano **datagrammi**, informazioni prive di stato sia negli ES che negli IS e per questo logicamente indipendenti. I datagrammi sono trasferiti in modo **best-effort**, con la possibilità che vengano persi e l'applicazione debba gestire l'errore ad alto livello.

Questi protocolli semplificati hanno un basso overhead e supportano le comunicazioni multi-punto ma purtroppo non prevedono la conferma di recapito dei pacchetti inviati! Negli IS i datagrammi non sono soggetti a controlli di flusso e congestione e quindi devono essere utilizzati con parsimonia.

I servizi che si basano sulle connessioni (**connection-oriented**) usano sessioni stabili e preparate. Richiedono un protocollo di segnalazione dei guasti e che IS ed ES mantengano lo stato della comunicazione. Con un certo overhead questi protocolli garantiscono che i dati siano consegnati in ordine, senza duplicati e correttamente tramite lo stream della connessione.

3 • Il livello fisico

L'informazione viene trasferita tramite un collegamento che funziona grazie ad un conduttore che ne determina la portata massima e la quantità di dati trasferibili in un lasso di tempo:

- Guida d'onda:
  segnali ottici su cavo con anima diamagnetica (**fibra**);
- Elettricità:
  variazione di voltaggio su un cavo con anima elettromagnetica (**rame**, ...);
- Onde:
  onde elettromagnetiche diffuse nell'**etere** (Wi-Fi).

Per trasferire più bit in contemporanea servono più coppie trasmettitore/ricevitore, una per ciascun bit. Per le velocità di trasmissione elevate, la trasmissione è generalmente seriale e quindi sempre consecutiva nel tempo.

Esistono due tipi di trasmissioni:

- Numerica:
  il segnale assume **valori discreti**;
- Analogica:
  il segnale assume valori non predeterminabili.

La **line coding** (**codifica di linea**) determina la corrispondenza tra valori o transizioni e valore semantico di ciascun bit, che può essere attivo (1) o disattivo (0)[11]. Nelle reti **Ethernet** si possono usare la **codifica Manchester**, basata su transizioni, la codifica **NRZ** (No-Return to Zero), la **NRZI** (No-Return to Zero Inverted) ed addirittura la complessa **MLT-N** (Multi-Level Transmit a *N* livelli).

11 Le lezioni successive dimostrano che nelle codifiche di linea complesse possono esistere molti valori semantici, in particolare ne esiste uno che significa "nulla".

Queste codifiche determinano quindi la forma d'onda e stabiliscono il bit rate (inversamente proporzionale alla durata del bit time, che è la durata di un singolo bit): affinché sia possibile avere un bit time, ricevitore e trasmettitore devono usare l'uno il clock dell'altro, sincronizzandosi a vicenda. Elaborando a campione il segnale, *il ricevitore determina autonomamente il bit time*, perché lo può dedurre proprio dalle transizioni d'onda.

Lo **spettro del segnale** è la rappresentazione di un segnale nel dominio delle frequenze rispetto all'intensità. Siccome il segnale varia sempre in modo costante e predicibile, possiamo affermare che è composto da infinite sinusoidi (l'onda quadra ne ha letteralmente infinite) e si può sempre disegnare su un piano.

La **banda del canale** è l'intervallo di sinusoidi che si possono propagare nel mezzo trasmissivo e si restringe con l'allungarsi del mezzo trasmissivo. Se la banda è abbastanza larga, tutte le frequenze vengono preservate; se la banda è troppo stretta, alcune frequenze vengono attenuate e il segnale si distorce e diventa privo di senso!

Lo spettro deve essere compatibile con la banda: **il numero di transizioni cresce con la banda richiesta**, quindi un segnale molto variabile (con tante informazioni) richiede la banda più larga. Per questo un bit rate elevato si ottiene usando codifiche che permettono di ottenere un segnale con transizioni frequenti in un piccolo spettro.

I canali più lunghi attenuano il segnale, che perde in ampiezza e diventa indistinguibile, rendendo impossibile la sincronizzazione. Aumentare la potenza in uscita dal trasmettitore è antieconomico perché crea calore e riduce la **dinamicità del segnale (ovvero la frequenza di campionamento massima)**: conviene sempre ottimizzare i canali trasmissivi.

I collegamenti in rame utilizzano la **variazione di potenziale** ma creano **correnti indotte**, che si sommano alla corrente del

trasmettitore, causando quindi interferenze elettromagnetiche che corrompono i dati[12]. Il problema si risolve con dei cavi isolati o attorcigliati in modo singolare, ma comunque standardizzati:

- **Coassiale**:
  dotati di nucleo di rame, isolato da uno speciale isolatore dielettrico, a sua volta coperto da due schermi elettromagnetici, rispettivamente a foglio ed intrecciato. Sono scomodi da posare e particolarmente delicati, inadatti all'uso d'ufficio;
- **Twisted pair** (doppino attorcigliato):
  tipici delle reti di calcolatori. Le correnti indotte delle due spire ruotano nello spazio in senso opposto e tendono ad annullarsi. Esistono due standard:
  - **UTP** (**Unshielded Twisted Pair**):
    4 coppie di cavi attorcigliati, a loro volta attorcigliati. Esistono diverse categorie di cavo UTP, numerate per qualità di realizzazione crescente;
  - **STP** (**Shielded Twisted Pair**):
    cavo UTP schermato come un cavo coassiale senza materiale dielettrico. Costoso, scomodo da posare e molto meno usato dell'UTP. STP ha qualità trasmissiva eccelsa, velocità elevata ed ovviamente largo spettro.

I connettori UTP ed STP sono normali **RJ-45**, ma gli STP richiedono l'uso di plug e jack con contatti di schermo (la schermatura del cavo).

Gli **amplificatori** sono IS del livello fisico che aumentano l'ampiezza del segnale, trattandolo sempre come se fosse analogico e senza interpretarlo. Proprio perché non sono in grado di interpretare il segnale, amplificano anche le interferenze in

12 Ovviamente il problema non si pone con i mezzi trasmissivi dielettrici, come le fibre ottiche.

entrata e lasciano il segnale deformato ma perlomeno sincronizzabile.

I **ripetitori** risolvono l'handicap degli amplificatori sui segnali digitali: essendo a conoscenza delle codifiche di linea possono rigenerare il segnale trasmettendo in uscita ogni bit in entrata. Questi sono IS intelligenti che annullano tutte le distorsioni e le attenuazioni. Con un numero adeguato di ripetitori ben posati a breve (ma costante) distanza uno dall'altro, il segnale arriva dovunque con notevole affidabilità.

I semplici ripetitori però hanno a loro volta un handicap: propagano anche gli errori di trasmissione. Conviene quindi investire direttamente in IS di livello superiore (2 o 3 ISO/OSI), capaci di gestire subito gli errori e far arrivare al destinatario una comunicazione sempre integra.

4 • Controllo dell'errore ed ARQ

Gli errori di trasmissione derivano da interferenze, attenuazioni etc., e sono tipici dei canali wireless. Quando l'elettronica degli ES fallisce o le PDU vengono scartate dagli IS a causa della congestione, questo conta come errore di rete. Gli errori possono essere gestiti su due livelli differenti:

- A livello Transport:

  ove sono definiti i protocolli come TCP ed UDP;

- A livello Data Link:

  ideale quando il **BER** (**Bit Error Rate**, tasso di errore) è elevato, cosa insolita nei canali moderni, ad eccezione del wireless. **Il wireless, infatti, richiede il controllo End-to-End**. I canali in rame ormai hanno un BER trascurabile.

L'error checking richiede l'aggiunta delle **PCI** (**Protocol Control Informations**) ai pacchetti, con overhead limitato, perché comunque troppe PCI vanificherebbero l'error checking. Tipiche PCI sono quelle dei **bit di parità** (basso overhead, a seconda della parità/disparità rileva soltanto un numero di errori pari o dispari) e del **codice di ripetizione** (overhead del 100%, rileva al massimo 8 errori e non a coppie[13]).

Tecniche più sofisticate generano un codice unico per un gruppo di dati e lo trasmettono assieme ad essi. Nella pratica il ricevente ricalcola il codice usando il medesimo algoritmo (che quindi deve essere ben noto) e lo confronta con il codice ricevuto: se i codici differiscono, il pacchetto viene scartato e si richiede la ritrasmissione. Algoritmi comuni per la generazione di questi codici sono il **CRC**, l'**hashing**, il **checksum** e il controllo riga-colonna.

13 Si intende che gli errori che si presentano nella stessa posizione sia nel PCI che nel pacchetto passano inosservati, perché in quel caso il codice di errore corrisponde alle informazioni: sia le informazioni che il PCI sono stati corrotti in modo identico.

I **FEC** (**Forward Error Correction**) sono algoritmi che non solo trovano gli errori, ma ne correggono alcuni al prezzo di un grosso overhead. Per questo le tecniche FEC vengono usate di rado.

La tecnica **ARQ** (**Automatic Retransmission Request**) è invece più utilizzata, perché risolve il problema alla radice chiedendo al mittente di ritrasmettere le PDU corrotte e addirittura quelle mancanti. ARQ utilizza PCI che non solo hanno il codice di error checking, ma sono anche numerate in modo predicibile; inoltre utilizza anche alcune PDU che non possono essere considerate SDU, come ad esempio i pacchetti ACK.

La prima tecnica ARQ è "stop-and-wait", ove il mittente aspetta l'ACK dal ricevente dopo ciascun invio, fino ad un eventuale timeout o ad una notifica NACK (Not ACK, solo se il ricevente aspettava una nuova PDU e non l'ha ricevuta).

Se il timer è troppo corto si rischiano ritrasmissioni inutili (che, se va tutto bene, causano solo lo scarto della PDU duplicata!), se è troppo lungo si spreca tempo. Le PDU duplicate si verificano anche quando gli ACK vanno persi ed in questo caso se la PDU non è numerata si rischia che venga accettata e l'informazione risulti duplicata!

Per evitare questo fenomeno le PDU, gli ACK ed i NACK devono essere sempre numerati[14]: usando la tecnica "stop-and-wait" tale numerazione richiede un solo bit. **ARQ si può utilizzare solo con i protocolli orientati alla connessione** e richiede la preventiva negoziazione e numerosi messaggi di controllo.

Possiamo affermare che la quantità di informazioni trasferite con successo equivale a:

Throughtput$_{\text{Stop-and-Wait}}$ = Lunghezza PDU / **Round-Trip Time**

Round-Trip Time$_{\textbf{canale}}$ = (Lunghezza PDU / Capacità canale) + (Lunghezza canale / a*c) + Tempo di elaborazione +

14 Le ACK hanno sempre la stessa numerazione della PDU di cui sono la reazione e quindi il numero della prossima PDU attesa sarà "(N+1) % 2".

`(Lunghezza ACK / Capacità canale) + (Tempo propagazione / a*c)`

...per questo Stop-and-Wait ha efficienza inversamente proporzionale alla velocità ed alla lunghezza del canale!

Su Internet il RTT (Round-Trip Time) di un secondo o più è eccessivo: con Stop-and-Wait l'uso di canali migliori è inutile, perché comunque il mittente aspetta la risposta prima di trasmettere ancora.

L'RTT è determinato fondamentalmente dalla lunghezza del canale e noi oggi utilizziamo canali capienti e lunghi chiamati "Long Fat Pipe". Possiamo affermare che:

$\textbf{Round-Trip Time}_{\text{approssimato}}$ = K * (Lunghezza PDU / Lunghezza canale)

...quindi conviene sempre aumentare il numero di PDU inviate prima dell'attesa, perché il Throughtput si riduce con l'allungarsi del canale e pertanto inviare PDU grosse crea problemi tecnici.

L'**ARQ a finestra** (**Window-based ARQ**, WARQ) consente al trasmettitore di inviare molti PDU di seguito in base alla dimensione prestabilita della "finestra", per poi attendere la serie di ACK.

Spesso gli ACK giungono prima che si inneschi l'attesa, permettendo la trasmissione continua se la finestra è abbastanza grossa. I PDU sono adesso numerati con più bit e la finestra deve essere più grande in base a quanto è grande il RTT.

Bisogna tenere conto del fatto che più è grande la finestra dell'ARQ e più memoria deve avere il ricevitore[15]. Nelle big fat pipes la memoria richiesta è molta e quindi WARQ non si usa tra nodi adiacenti (a livello ISO/OSI 2). Tipicamente ARQ viene utilizzato a livello ISO/OSI 4, tra ES con più sessioni, per aumentarne l'efficienza.

15 In gergo questa è la "memoria del canale".

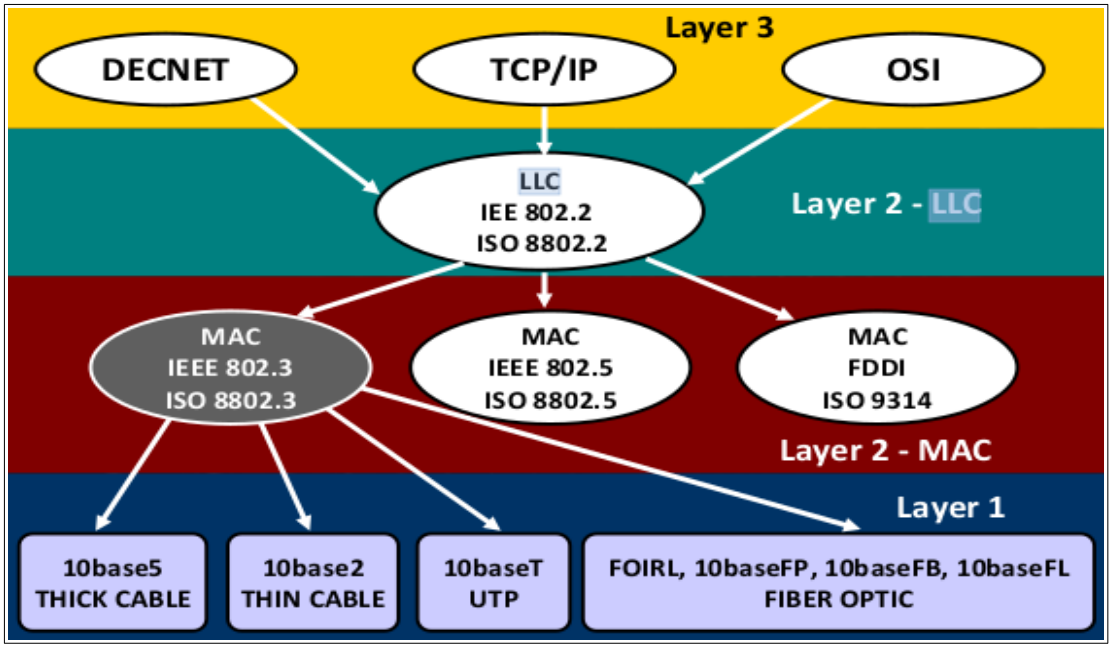
La tecnica della "**sliding window**" permette di variare dinamicamente la dimensione della finestra in base a quanti ACK arrivano. In particolare ne esistono due varianti:

- "**go-back N**":
  quando un PDU manca, si ritrasmette a partire dall'inizio della finestra man mano che questa "slitta" sul flusso di PDU;
- "**selective repeat**":
  si specificano i PDU mancanti negli ACK e l'intera finestra si ritrasmette solo se scade il timer.

5 • Medium Access Control

IEEE 802 definisce la LAN come un sistema di comunicazione che consente la ricetrasmissione in una zona limitata in estensione, con canali ad alta velocità e basso tasso di errore. Per questo i protocolli LAN gestiscono trasferimenti a breve distanza, di brevissima durata ("burst" di dati) e con pochi errori potenziali. Le reti LAN sono fatte da canali condivisi dai device che li usano a turno.

Il cablaggio è stabilito da diversi standard IEEE 802 (ISO 8802) che definiscono reti multi-vendor con collegamenti multi-vendor, secondo un modello protocollare che definisce solo i livelli Data Link e Physical. Il protocollo **LLC** (**Logical Link Control** di ISO 8802.2) soprassiede a MAC ed è presente unicamente nel livello Data Link; al contrario, MAC è presente sia nel livello Data Link che in quello Physical: esso deve interfacciarsi con protocolli puramente Physical come FDDI (necessario per CSMA/CD), 802.3, 802.5 (ambedue usati nel **token passing**) e molti altri. Il MAC stabilisce il ricevente tramite **indirizzi MAC** e può trasmettere usando diversi algoritmi, di cui però **l'unico ancora in uso è il 802.3** (token passing).



DECNET, TCP/IP ed OSI utilizzano tutti i servizi di LLC che utilizza a sua volta diversi standard MAC, anch'essi parte del livello 2 "Data Link" e compatibili con diversi tipi di cablaggio.

802.3 ed **Ethernet 2.0** sono strutturati per coesistere[16]: tutti e due i protocolli, infatti, sono pensati per reti LAN collegate con topologia a bus e trasmissione non deterministica.

Gli indirizzi MAC sono lunghi 6 B (48 b) ed i primi 2 B dichiarano il tipo di indirizzo, che può essere esclusivamente:

- Universale:
  vd. prima cifra del primo byte → bit disattivo.
- Locale:
  vd. prima cifra del primo byte → bit attivo.

...ed in contemporanea, esclusivamente:

- Di gruppo:
  vd. seconda cifra del primo byte → bit disattivo;
- Individuale:
  vd. seconda cifra del primo byte → bit attivo.

Gli indirizzi universali unicast (quindi individuali) individuano univocamente la scheda di rete, perché i primi 3 B del MAC universale unicast sono **OUI** (**Organization Unique Identifier**), concessi da IEEE al costruttore del trasmettitore.

L'**indirizzo di broadcast** "FF FF FF FF FF FF" ha tutti i bit attivi e tutte le stazioni della LAN lo ascoltano.

Gli indirizzi multicast fanno capo ad un gruppo di interfacce di rete ed hanno la seconda cifra esadecimale dispari (X*X* XX XX XX XX XX).

16 Non è un caso: IEEE 802.3 è il successore del protocollo proprietario Ethernet 2.0, quest'ultimo ideato da Digital, Intel e Xerox.

CSMA/CD (**Carrier Sense Multiple Access with Collision Detection**) è l'algoritmo che decide chi può trasmettere su un collegamento. L'algoritmo "tasta la portante del cavo" per capire se qualche altra stazione sta generando un segnale[17] e, quando ne individua uno, si limita ad attendere per poi trasmettere quando esso scompare.

A causa di vincoli fisici irrisolvibili, i segnali non arrivano istantaneamente e l'algoritmo potrebbe non rilevare la portante: se il trasmettitore trasmette quando c'è un segnale interferente (inevitabilmente lo fa!) allora si genera una **collisione** di rete.

Per tamponare il problema delle collisioni la stazione deve ascoltare anche mentre trasmette ed assicurarsi in ogni istante che il proprio segnale sia intelligibile: se non lo è allora c'è una collisione in corso ed il trasmettitore non può far altro che generare un particolare tipo di segnale contenente "spazzatura", per far capire alle altre stazioni che c'è una collisione in corso. A questo punto il trasmettitore attende "alla cieca" un pò di tempo per far propagare il segnale spazzatura e poi riprova a trasmettere nella speranza che le altre stazioni abbiano rilevato i dati corrotti ed innescato a loro volta l'algoritmo di pausa.

In particolare il segnale spazzatura è detto **jamming sequence** (letteralmente sequenza di disturbo) e la sua esistenza, assieme a quella delle collisioni, rende il tempo di trasmissione non-deterministico!

Comunque, quando le stazioni sono molto lontane (il diametro della rete è ampio) **c'è un ritardo nel rilevamento delle collisioni che impone un limite all'area che una rete LAN può occupare**.

Il diametro massimo della rete è uguale a:

$$2 * (\text{Diametro rete} / \text{Velocità segnale}) \leq (\text{Dim. pacchetto} / \text{Velocità di trasmissione})$$

17 Il carrier sense può individuare solo quei segnali che potrebbero fare collisione nel medesimo segmento di rete del trasmettitore, ovvero nel raggio delle collisioni potenziali (trattato nelle lezioni successive).

Da questa formula sappiamo che una LAN non può superare i 5,7 km di diametro.

Al crescere delle velocità di trasmissione il diametro massimo della rete si riduce, per questo da 100 Gb/s in poi servono particolari accortezze per superare il limite teorico del diametro.

Quando MAC richiede la ritrasmissione essa è velocissima; quando si esce dalla LAN e MAC non è più disponibile le prestazioni calano. Una nozione generalmente valida è che la prestazione della rete è ottimale se il RTT è minore del tempo necessario a trasmettere il pacchetto più piccolo in assoluto, perché **non si possono rilevare le collisioni se la trasmissione non è ancora in corso**.

Per garantire la sincronia, ogni frame deve essere preceduto da un **preambolo** (che sia un segnale portante privo di dati o altro dipende dall'algoritmo) e dallo **SFD** (**Starting Frame Delimiter**) da cui inizia il pacchetto. Per consentire il rilevamento delle collisioni ciascun pacchetto deve essere lungo almeno 64 B, di cui almeno 46 B sono il payload.

Proprio per questo in ciascun frame è presente un intero che indica la lunghezza del payload, nel caso esso sia più piccolo di 46 B e debba essere troncato prima di passarlo ai livelli ISO superiori. Il frame inizia con 12 B atti a contenere i MAC degli EP e termina con la **FCS** (**Frame Check Sequence**), contenuta nel MAC trailer.

Alla FCS segue un silenzio di 4.7 microsecondi (in ricezione) o 9.6 microsecondi (in trasmissione), che anticipa la prossima trama: questo silenzio è detto **IPG (Inter-Packet Gap) o IFS (Inter-Frame Space)**.

Per evitare che un singolo trasmettitore monopolizzi il canale, lo standard sancisce anche che le singole trame non possano essere più lunghe di 1518 B[18].

18 Questi limiti saranno ridiscussi nei capitoli successivi e il loro significato varia in base alla velocità.

Le schede hanno una lista di indirizzi multicast abilitati alla ricezione (su cui ascoltare) ed in alcuni casi possono ascoltare su tutti i MAC multicast o su tutto il traffico di rete sia unicast che multicast, con una modalità di funzionamento per il debugging detta "**modalità promiscua**".

6 • Ethernet + IEEE 802.3: LLC, PS e dimensionamento della rete

Lo strato **L**ogical **L**ink **C**ontrol serve perché Ethernet 2.0 ed IEEE 802.3 hanno trame uguali in tutto tranne che nel terzo campo di 2 B, che in Ethernet 2.0 contiene l'**ether type**, ovvero il livello OSI destinatario del PDU, mentre in IEEE 802.3 è la lunghezza del payload. Per questo Ethernet 2.0 (che è proprietario) richiede tanti moduli diversi in base al significato del pacchetto dettato dall'ether type, mentre IEEE 802.3 richiede lo strato LLC, che deduce in un secondo momento il tipo di frame.

Tutti i protocolli hanno ether type maggiore di 1500 B e la lunghezza di un frame non può eccedere tale estensione: per questo il terzo campo può essere utilizzato per distinguere in tempo reale i due protocolli e quindi utilizzarli contemporaneamente sulla stessa rete LAN.

Se il frame è IEEE 802.3, il payload è sempre di tipo LLC e contiene tre campi prima delle informazioni utili:

1. **DSAP** (**Destination SAP**):
   contenente un valore standard IEEE;
2. **SSAP** (**Source SAP**):
   contenente un valore standard IEEE;
3. **Control**:
   usato da LLC per indicare il tipo di frame. Se vale "3" allora il frame è stato inviato tramite un protocollo best-effort e quindi è inaffidabile.

Le informazioni del frame LLC sono ovviamente seguite dalla FCS (Frame Check Sequence).

DSAP e SSAP contengono valori standard IEEE per i protocolli LAN più comuni ma i protocolli standard "de facto" devono utilizzare particolari stratagemmi: i protocolli più utilizzati oggi, ovvero TCP/IP ed IP, devono usare l'estensione standard **LLC SNAP**

(**SubNetwork Access Protocol**). SNAP si attiva impostando DSAP e SSAP al valore speciale "FE" ed aggiungendo 5 B dopo il campo Control, composti da un codice unico OUI e da 2 B di codice identificativo protocollare. Questa tecnica rende possibile l'**imbustamento** di tutti i protocolli di livello superiore dentro LLC.

Quando il produttore del protocollo non dispone di un proprio OUI o il protocollo non è proprietario (come IP e TCP/IP), si deve seguire una procedura ancora più particolare: si imposta l'OUI dello SNAP a 0 ed i 2 B di identificativo protocollare ad un valore dell'ether type di Ethernet 2.0, che per IP è "0800". Elaborata l'estensione LLC SNAP il pacchetto può essere processato dai livelli superiori.

Per evitare le complicazioni imposte da SNAP, le schede di rete moderne generano direttamente pacchetti Ethernet 2.0 pur dichiarandosi compatibili con IEEE 802.3 (proprio perché Ethernet ingloba il protocol type nell'intestazione MAC sotto il nome di *ether type*).

IEEE 802.3 prevede tanti standard fisici, a differenza di Ethernet 2.0 che è meno flessibile e ne consente solo uno: per questo le schede si possono spacciare per IEEE 802.3 quando supportano questi standard per il collegamento fisico.

3 standard IEEE 802.3 per il rame:

- **10BASE5**:
  cavo coassiale spesso. Trasmissione a 10 Mb/s in banda base, max. 500 m. Costoso e difficile da posare;
- **10BASE2**:
  cavo coassiale sottile. Max. 500 m;
- **10BASE-T**:
  doppino telefonico standard. Max. 100 m. Economico e semplice da posare, di grande successo.

Per superare le limitazioni in lunghezza si utilizzano dei ripetitori, semplici IS di livello 1 capaci di ritrasmettere i frame così come arrivano.

4 standard IEEE 802.3 per la fibra:

- FOIRL:
  collega vicendevolmente i ripetitori;
- 10BASE-FL:
  collega ripetitori e stazioni;
- 10BASE-FB:
  collega con fault-tollerance;
- 10BASE-FP:
  topologia a stella ottica con centro stella passivo, capace di riflettere e replicare il segnale senza richiedere alimentazione ma limitato al semplice broadcasting dei segnali in entrata.

10BASE-T usa un semplice e robusto cavo UTP, utile per cablare gli uffici o le case in alternativa al wireless, ed adopera il comune RJ-45 per jack e plug ma si utilizza solo per connessioni punto-punto con topologia a stella ed hub centrale (ripetitore multi-porta).
Il risultato è una rete con topologia equivalente a quella bus, perché solo una stazione per volta può trasmettere e vengono utilizzate solo 2 coppie del doppino a 4 coppie.

ISO/IEC 11801 + TIA/EIA 568A stabiliscono 90 m di cablaggio più 10 m di patch cords, collegati ad un armadio con ripetitore: 10BASE-T è compatibile con questi standard perché usa la codifica Manchester, ma ovviamente impone di non superare il tempo massimo di propagazione.

Per stabilire il diametro ideale della rete con lo standard 10BASE-T, non servono calcoli ma semplici accortezze:

1. Non facendo tratte più lunghe di 100 m;
2. Non usando mai più di 4 ripetitori in cascata: ciascun pacchetto deve passare al massimo per 4 ripetitori;
3. Se si utilizzano spezzoni in fibra ottica, non usare più di 2 ripetitori in cascata (anche se ci sono altri spezzoni in rame).

In una LAN le collisioni riducono l'efficienza del 40%. Dobbiamo quindi creare piccoli domini di collisione con poche stazioni, per ridurre al minimo le trasmissioni contemporanee. Basse complessità sono sempre la chiave del successo, anche in ambito economico.

7 • Bridging

Il bridging è l'uso dei bridge nella LAN. I bridge consentono di collegare più LAN, spezzando così i grossi domini di collisione in domini di collisioni più piccoli: ogni LAN collegata al bridge può trasmettere senza curarsi delle altre.

I bridge sono prodotti standard IEEE 802.1D che definiscono BLAN (Bridged LAN) e consentono quindi l'inoltro selettivo dei frame e quindi l'isolamento del traffico. Questi vengono chiamati anche **bridge trasparenti**, perché le stazioni LAN ne ignorano l'esistenza.

I bridge fanno **store & forward** tramite CSMA/CD: la Collision Detection viene rispettata[19] grazie allo storaging ed al conseguente inoltro ritardato dei frame.

Un bridge che connette più di due domini di collisione si chiama **switch** ed ha molte più porte di rete. Più la LAN cresce e più i bridge divengono inadeguati, anche perché gli switch possono sostituire in modo trasparente i bridge.

La pratica di aggiungere switch si chiama "**segment switching**" e, a stadi avanzati, viene chiamata "**station switching**" perché si arriva al punto di collegare ciascun server ad uno switch separato. L'aggiunta di switch è benefica perché riduce al minimo il numero di collisioni nella LAN ed è conveniente, oltre che semplice.

Ogni switch fa l'**inoltro selettivo** ("selective forwarding") tramite un **filtering database**[20] gestito dinamicamente. Il DB serve per associare ogni MAC alla sua porta di inoltro: **lo switch impara da ogni pacchetto** e se è incerto fa broadcasting su tutte le

19 La collision detection non viene rispettata dai bridge che utilizzano la tecnica del "cut-trough" quando non rilevano collisioni. Questi bridge non sono standard e creano più problemi, mentre l'aumento di performance dell'invio immediato è limitato a pochi casi.

20 Taluni lo chiamano forwarding database.

porte. Lo switch scarta[21] i frame non validi e quelli che hanno come destinazione la porta da cui sono entrati.

Se una stazione si sposta (e quindi la topologia di rete cambia) i pacchetti vengono inoltrati a vuoto finché lo switch non fa scadere l'associazione obsoleta usando la tecnica di **aging**: ogni entry del DB invecchia fino a scadere e quando viene riconfermata o modificata, ringiovanisce.

Gli switch imparano automaticamente dal campo mittente di ogni frame e quindi tanto più si trasmette e tanto più sono aggiornati dalla **fase di learning**.

Gli switch fanno **flooding** quando:

- Ricevono un pacchetto da mittente noto e destinatario non ancora noto;
- Ricevono un pacchetto multicast:
  perché i bridge non possono memorizzare indirizzi multicast;
- Ricevono un pacchetto broadcast:
  perché i pacchetti broadcast sono fatti apposta e devono essere ricevuti da tutta la LAN. Questo tipo di traffico occupa tutta la rete e può essere problematico;

Quando avviene il flooding non esiste separazione del traffico e la trama occupa banda sull'intera rete.

Grazie al learning automatico gli switch sono velocissimi e poco costosi: li chiamano "switch" proprio per la loro abilità di inoltrare molte trame da un segmento all'altro in poco tempo e per il fatto che sono logicamente superiori ai semplici ripetitori (che trattano bit e non frames).

Anche chi non ha competenze informatiche può usare gli switch senza difficoltà: quando il filtering DB è vuoto non serve

21 Solo se non fa cut-trough.

configurazione manuale, perché lo switch fa flooding su tutte le LAN e poi impara dall'indirizzo sorgente dalla trama appena inoltrata; quando giunge una risposta, impara anche da essa e se possibile evita di fare flooding, perché chi riceve è già noto.

Quando due stazioni limitrofe comunicano nello stesso dominio di collisione, il bridge fa isolamento di rete e non inoltra i pacchetti sulle altre LAN diminuendo la congestione in tutta la rete aziendale.

8 • Il protocollo Spanning Tree

Il bridging trasparente richiede l'uso del protocollo Spanning Tree perché, quando la LAN ha percorsi chiusi, si creano delle **broadcast storms** che bloccano i SI.

Queste tempeste di pacchetti vengono generate - inevitabilmente - appena un device si connette alla LAN ed invia un pacchetto broadcast, ma solo se quest'ultima funziona tramite un circolo chiuso di SI (bridge, switch, ...): una stessa trama viene replicata all'infinito nel tempo e satura i canali condivisi, fino a che i SI terminano le proprie risorse di calcolo e bloccano la rete[22]. L'unica soluzione è evitare le broadcast storms (che sono per definizione ingestibili) "tagliando" i collegamenti che generano percorsi chiusi!
Qui nasce un paradosso, dato che i percorsi chiusi servono per garantire l'efficienza della rete fornendo la ridondanza quando un collegamento diviene inutilizzabile. Per questo dobbiamo operare dei tagli software automatici e dinamici e non dei tagli hardware permanenti.

Il **protocollo Spanning Tree** chiude dinamicamente le porte dei SI per trasformare la topografia di rete circolare in un albero, secondo lo standard IEEE 802.1D richiesto dai SI. Questo protocollo è essenziale, perché l'errore umano è altamente probabile durante la configurazione dei collegamenti di rete negli switch e deve essere "**Plug & Play**", ovvero economico ed auto-configurante.

I passi dell'algoritmo Spanning Tree possono essere sintetizzati in:

1. Elegge[23] tra tutti i bridge disponibili la radice dell'albero, ovvero il **root bridge**;
2. Elegge la **root port** di ciascun bridge dell'albero, che sarà l'unica porta utilizzata per comunicare con il *root bridge*;

22 Dopo una broadcast storm, tutti i SI devono essere riavviati affinché la LAN torni operativa.
23 Ogni bridge della LAN deve accettare autonomamente il root bridge e propagare la scelta denunciando di non essere il root bridge. La diffusione è naturale e richiede un tempo indeterminato in base alla dimensione della rete.

3. Seleziona le **designated ports** di ciascun bridge: queste saranno le uniche porte abilitate all'inoltro dei pacchetti nella LAN.

Ogni bridge ha una **bridge priority** che lo designa o al contrario lo scarta nel processo di elezione della radice ideale dell'albero; ogni porta di ogni bridge ha associato un **port cost**, che determina il costo economico della trasmissione su quel tratto di rete (ovviamente si cerca di ridurre i costi al minimo).

Tutti i valori configurabili hanno defaults ragionevoli (predefiniti nel SI) e se tutti i nodi della rete non sono stati configurati (e quindi hanno la stessa priorità) allora Spanning Tree utilizza l'indirizzo MAC per distinguerli e scegliere la root in base al *port cost*, valore che verrà poi utilizzato anche nella scelta delle *designated ports*.

Ogni porta può essere in stato "forwarding" o "blocking": se sono bloccate allora scartano il traffico in entrata ed in uscita, mentre se sono abilitate all'inoltro fanno normalmente routing.

Le porte non-root e non-designated non vengono utilizzate e quindi rimangono *blocked* finché non avviene un guasto di rete o comunque un evento che determina un cambiamento topologico della LAN.

Spanning Tree utilizza le **BPDU** (**Bridge Protocol Data Unit**) per configurare la rete tramite delle speciali **C-BPDU** (**Configuration BPDU**), inviate su un indirizzo multicast predefinito. Le BPDU vengono ancora utilizzate dopo la creazione della rete, quando essa cambia topologia o ha problemi. Per questi cambiamenti improvvisi vengono utilizzate le ancor più particolari **TCN-BPDU** (**Topology Change Notification BPDU**).

Quando la rete viene attivata ogni bridge mente dichiarando di essere il root bridge ed invia il proprio **root identifier**, che contiene sia l'indirizzo MAC che la priorità. In seguito ciascun bridge capisce se deve essere effettivamente la radice o no a seconda delle C-BPDU ricevute e si dichiara non-root

semplicemente smettendo di inviare il proprio root identifier. Quando tutti i bridge hanno in memoria lo stesso root identifier la rete ha eletto a tutti gli effetti il root bridge.

Ogni C-BPDU contiene il costo del percorso fatto a partire dalla radice: ogni bridge in totale autonomia sceglie la propria root port in base al costo più basso. Così facendo, la root port di ciascun nodo utilizza sempre il percorso più breve per inviare frame al root bridge. Dopo aver eletto la root port, il bridge smette di inviare C-BPDU su quest'ultima e le propaga solo sulle designated ports ovvero verso le foglie dell'albero.

Dopo aver completato la configurazione iniziale, tutti i bridge della rete inoltreranno il traffico dalle foglie verso la radice tramite la root port. Il root bridge, che a questo punto è cosciente di essere la radice, al contrario invierà il traffico selettivamente verso i sotto-alberi.

Le designated ports vengono di volta in volta scelte in base al costo delle C-BPDU: due nodi sullo stesso segmento[24] si confrontano in base alla C-BPDU reciprocamente ricevute ed uno di loro si mette in stato di blocco su tutte le porte (eccetto la root port) non appena comprende che la sua porta non deve essere designated perché può creare loops ed è più costosa di quella del vicino.

Questi nodi si dicono non-designated e si riattivano automaticamente quando il **Link Integrity Test**[25] fallisce o non riceve periodicamente C-BPDU dal vicino. Appena il SI smette di essere non-designated, invia dalla root port una nuova TCN-BPDU e il root bridge ne prende atto: esso reagisce inviando sui sotto-alberi nuove C-BPDU con il **topology change bit** attivo, che serviranno a reinizializzare la configurazione della rete per adattarla al cambiamento topologico.

I nodi che ricevono questi pacchetti speciali sulla root port, rispondono inviando C-BPDU con il **topology change**

24 Quindi con la stessa profondità e lo stesso sotto-albero.

25 Questo è un test fisico del collegamento che deve essere esplicitamente supportato dal collegamento ed in generale dalla rete.

acknowledgement bit attivo e subito dopo svuotano il proprio filtering DB. Con tutti i DB vuoti la configurazione di rete ricomincia, favorita dal fatto che il root è stato di volta in volta riconfermato e quindi è improbabile che cambi.

Il protocollo Spanning Tree richiede che i timer all'interno di ciascun bridge abbiano la giusta velocità: se sono troppo veloci si genera immediatamente una broadcast storm, mentre se sono lenti questa eventualità è scongiurata. Questo implica che le LAN sono sempre lente nel reagire ai cambiamenti topologici.

Spanning Tree è sconveniente nei costosi collegamenti geografici (WAN) basati su bit fat pipes perché, oltre a non reagire abbastanza velocemente ai guasti, alcuni collegamenti rimangono inutilizzati ed altri possono divenire sovraccarichi ed usurarsi velocemente.

A causa della rapida usura, su Internet i bridge sono stati soppiantati dai **router** nei collegamenti più costosi perché, potendo quest'ultimi gestire più alberi diversi, non lasciano collegamenti inutilizzati e non causano il sovraccarico di alcuni nodi di rete.

9 • Il protocollo Fast Ethernet ed affini

Nell'aggiornare gli standard è preferibile evitare di modificare il formato dei frame IEEE 802.3 o Ethernet 2.0, oltre a mantenere bassi i costi rispetto all'aumento prestazionale. Per aumentare la velocità di trasmissione bisogna accorciare le trame, ma questo riduce il diametro massimo della LAN.

Gigabit Ethernet risolve il problema aumentando il tempo minimo necessario alla trasmissione di tutte le trame. Altra soluzione è fare a meno di CSMA/CD, come nel caso di 10G Ethernet, che però impone piccole modifiche al livello MAC e l'uso dello station switching: si perde l'abilità di condividere un canale con altre stazioni ma, in teoria, si può ancora utilizzare il canale in full-duplex tramite due doppini dedicati.

Il full-duplex richiede che la Collision Detection venga disabilitata, azione che ha i bonus aggiunti di rimuovere il vincolo MAC (non quello fisico!) sul diametro della LAN e raddoppiare il throughput a costo zero. Questo, sempre a patto di utilizzare connessioni dirette.

Questi standard sono amichevoli verso il mercato aperto perché, essendo modulari, consentono a diversi produttori di specializzarsi nella costruzione dei singoli moduli e quindi creano competizione: la rete è totalmente multi-vendor e le schede supportano moduli inseribili per essere compatibili con diversi standard di collegamento.
Ciò consente ai produttori l'aggiornamento economico delle schede ed un basso **time-to-market**[26]. Per questo sono sempre più frequenti le interfacce multi-standard, capaci di negoziare il protocollo e scalare gradualmente la velocità di trasmissione finché non si trova lo standard più efficiente supportato da tutti e due gli EP.
Tale processo è detto **auto-negoziazione** ed è disponibile solo per i canali in rame, ad eccezione della negoziazione della modalità di trasmissione (half-duplex/full-duplex) che è

26 Tempo necessario affinché un nuovo standard venga adottato dai produttori hardware e questi vendano hardware compatibile con esso.

disponibile anche su fibra ottica: i SI e gli EP provano di volta in volta modalità e velocità inferiori fino a che non riescono a comunicare.

IEEE 802.3u, pubblicizzato come **Fast Ethernet**, non modifica il livello MAC ma aggiunge molti sottolivelli (moduli) che vengono riconciliati da un livello superiore, su cui si basa CSMA/CD.

Ogni modulo si occupa di uno standard differente:

- **100BASE-T4** → 4 coppie in uso, di cui 2 commutate a turno in RX/TX e le restanti stabili in RX/TX. Poco usato;
- 100BASE-T2 → "hack" per le reti in cui gli amministratori hanno i cavi e quindi due coppie non sono disponibili. Usa una modulazione molto complessa per ovviare alla mancanza.

100BASE-T4 usa la codifica di linea 8B6T, ovvero 8 simboli binari codificati in 6 simboli ternari: tre simboli ternari hanno più ridondanza di quattro simboli binari a causa del maggior numero di combinazioni potenziali.

La ridondanza permette di usare per l'IPG un simbolo apposito e di evitare l'uso di terne che manterrebbero il segnale costante, aumentando così il numero di transizioni in un piccolo spettro. Questa codifica aumenta il numero di simboli trasmessi a 25 Mbaud (25 Mega bauds) per ogni coppia a 100 Mb/s: questa velocità viene raggiunta su un UTCP cat. 3 su 100 m, notevole per un tipo di cavo di qualità così bassa!

100BASE-T2 usa MAC in modalità half-duplex ma permette l'uso contemporaneo di una stessa coppia per ascoltare e trasmettere, al costo non trascurabile di aggiungere un trasformatore ibrido all'interfaccia e di usare la complessa codifica di linea **PAM5** (**5-level pulse amplitude modulation**), che è multi-livello e molto complessa. Il risparmio sul cablaggio si paga con schede più costose.

Gli standard ancora in uso sono i 100BASE-X, che hanno molti moduli sotto il solito strato di riconciliazione con CSMA/CD. Il **Physical Coding Sublayer** rende trasparente il mezzo (fibra/rame) usando la codifica di linea 4B5B, che aumenta il consumo di banda di 25 Mb/s ma genera ridondanza, che ancora permette l'IPG con simbolo apposito evitando la desincronizzazione del clock proprio perché la scheda non si disattiva. 100BASE-X richiede collegamenti STP o UTP cat. 5, per far fronte all'occupazione spettrale più larga.

La versione 100BASE-TX usa sia la codifica di linea NRZI che la MLT-3, rispettivamente per permettere schede con supporto misto ottico-rame e per ridurre lo spettro richiesto su rame.

Una rete 100BASE-TX, ovviamente basata su ripetitori, ha domini di collisione di max. 205 metri. Per unire le LAN devono essere utilizzati obbligatoriamente bridge switch che però, a causa di CSMA/CD, limitano la rete a spezzoni di 300 m anche quando si utilizza la fibra ottica.

10 • Gigabit Ethernet, fibra ottica e 10/40/100G Ethernet

IEEE 802.3A/B/Z sono gli standard Ethernet ad alta velocità. Le collisioni si devono propagare al mittente nel tempo di trasmissione della trama più piccola, quindi l'aumento di 10 volte del bitrate porterebbe ad un diametro di rete di max. 50 metri e quindi totalmente inaccettabile. Per ovviare si aumenta artificialmente la dimensione della trama minima con la **tecnica del Carrier Extension**: si accoda ai dati l'**extension bit**, per portare la lunghezza della trama a 4158 b ed ottenendo così lo stesso diametro massimo di Fast Ethernet.

Per aumentare la **collision window** si possono aggiungere fino a 3584 b, che però comportano un grande overhead per le trame più piccole e solo per poter tenere occupato il canale. Per ovviare Gigabit Ethernet implementa la **burst mode**, che permette la trasmissione consecutiva (ovviamente continua) di più trame fino ad un max. di 77.000 b, con una **burst window** di 65 Kbit[27].

Non essendoci più l'IPG, si usano dei bit di riempimento (i "**fill bit**"), realizzati con una speciale codifica di linea per renderli equivalenti all'IPG/IFG. Questa è l'alternativa performante al classico Carrier Extension, ma si può utilizzare solo quando la stazione deve trasmettere molte trame.

La Burst Mode disabilita necessariamente CSMA/CD quando ci sono ripetitori ed in full-duplex il protocollo carrier extension e la Burst Mode sono superflue perché non ci sono collisioni da rilevare.

Gli standard Ethernet ad altissima velocità sono:

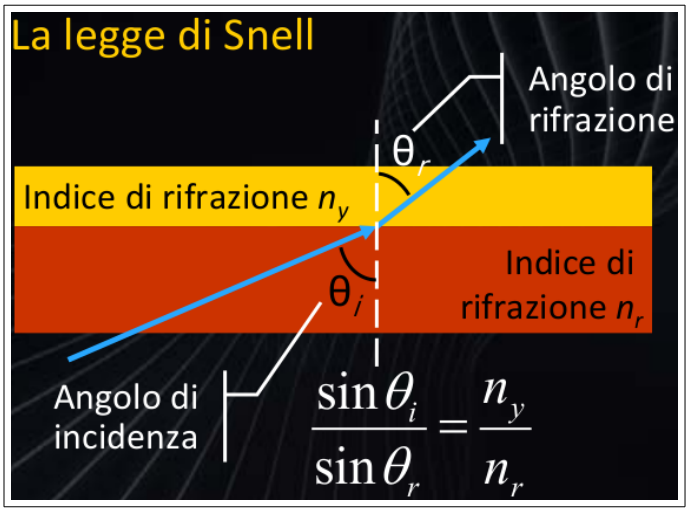
- **1000BASE-T** → cavo in rame, collegamento full-duplex su UTP cat. 5 dotato di 4 coppie. Questo full-duplex è reale/fisico e permette trasmissioni fino a 100 m, con RX e TX paralleli su tutte e quattro le coppie in contemporanea.

27 In seguito il canale deve essere obbligatoriamente liberato.

Richiede PAM5, con 8 b mappati su 4 simboli base-5[28] ed un rendimento di 125 Mbaud/coppia. La ridondanza si usa per ridurre le forti interferenze scegliendo oculatamente i simboli di ogni coppia interferente.

- **1000BASE-X** → codifica 8B10B creata dalla *Fiber Channel*®. Crea ridondanza, che viene poi utilizzata per evitare di creare interferenze. Si basa sul livello fisico "1000BASE-CX", che prevede collegamenti a breve distanza con connettori speciali ed usa solo 2 coppie, una in TX e l'altra in RX. Esistono standard 1000BASE-X per la fibra ottica, che però richiedono connettori speciali dotati di chiavi per la corretta inserzione:
  - 1000BASE-SX → onde corte (implica una frequenza ottica elevata) con tratte di max. 550 metri;
  - 1000BASE-LX → onde lunghe (implica una frequenza ottica ridotta) con tratte da 550 metri a 5 km;

Nozioni sulle fibre ottiche



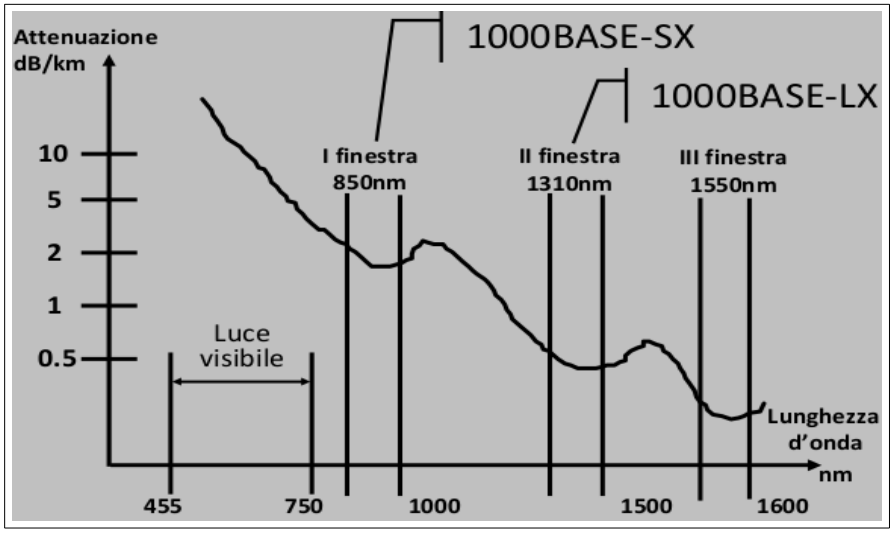
Legge di Snell: due conduttori dielettrici con diversi indici di rifrazione differenti. Il segnale incidente sul cavo composto da questi materiali non viene deviato e rimbalza.

28 In gergo vengono chiamati "quinary symbols".

Le fibre ottiche si basano su due materiali, uno dei quali riflette la luce invisibile, che viaggia in quello con alta rifrazione. Così il nucleo (detto "**core**") della fibra ha alta rifrazione e fa da guida d'onda e l'isolante (il "**clodding**") evita la dispersione del segnale ottico. Se la fibra è pura il segnale viaggia più a lungo.

Le fibre multi-modali hanno un core relativamente grande (50 micron) e quelle mono-modali uno piccolo (10 micron). Le fibre ottiche sono costituite da fibra silicea ("silica") o, nel caso di quelle grosse a lunga distanza, plastica. I trasmettitori possono essere luci LEDs o pregiati lasers, ma la trasmissione è "stile morse" (ON/OFF/ON...).

Diverse frequenze hanno ciascuna la propria attenuazione fisica, per questo non si utilizza la luce visibile e si scelgono invece quelle frequenze con attenuazione minima. Esistono soltanto tre "finestre di trasmissione ideale", di cui due richiedono l'uso del laser.



Il **Wavelength Division Multiplexing** permette la trasmissione simultanea di più segnali su una stessa fibra, a patto di usare più frequenze differenti. La versione "dense" permette di trasmettere migliaia di canali nella stessa finestra, mentre la "coarse" poche decine in modo più economico.

Per approfondire:

- 1000BASE-SX:
  multi-modale, max. 220/500 metri;
- 1000BASE-LX:
  - multi-modale, max. 550 metri;
  - mono-modale, max. 5 km;
- Non-standard:
  - mono-modale su 2° finestra di trasmissione, max. 10 km;
  - mono-modale su 3° finestra di trasmissione, max. 100 km

10G Ethernet non usa ripetitori o CSMA/CD, quindi neanche la burst mode ed la carrier sense. Gli standard 10BASE-W si basano su SONET/SDH e supportano le MAN/WAN esistenti a 9.6 Gb/s, purché utilizzino solo fibra ottica e possano utilizzare più finestre in combinazione a DWDM[29].

29 Non ancora trattati.

11 • Reti Wireless (IEEE 802.11)

Le reti senza fili sono inaffidabili: ostacoli, veicoli e persone in transito richiedono accortezze da parte delle trasmittenti, quali la riduzione della velocità di trasmissione e la resilienza alla variazione della copertura di rete. Per questo le reti wireless richiedono **ARQ** e **FEC**[30], protocolli per la correzione e la ritrasmissione che garantiscono la consegna dei pacchetti.

Le reti wireless sono in genere usate come estensione delle reti cablate preesistenti, ad esempio per gli smartphones, i laptop e gli ospiti dell'ufficio/casa. Le reti wireless sono vulnerabili allo **sniffing** e quindi è fondamentale sia la cifratura che l'autenticazione, per evitare che le comunicazioni possano essere compromesse o lette dalle stazioni malevoli.

802.11 fa parte degli standard MAN/WAN (dipartimento 802 di IEEE) ed è anche chiamato standard per le **Wireless LAN** (**WLAN**) o "**Wi-Fi**"[31]. Questo standard opera su bande non regolate, come 2,4 GHz e quindi prevede sempre la condivisione della banda con le reti limitrofe, con perdita prestazionale; altri sotto-standard usano gli infrarossi, 5 GHz, o una combinazione di 2,4 e 5 GHz.

802.11 nasce nel **1997** e si sviluppa fino al 2003, creando vari standard ancora oggi supportati, tra cui il moderno **802.11g.** Le schede multi-standard, per la loro flessibilità di impiego, sono quelle preferite dai produttori.

Gli standard fondamentali per le trasmissioni wireless sono:

- **802.11e**:
  permette di privilegiare certi tipi di traffico in base a delle classi di Qualità del Servizio (QoS);

30 In aggiunta a vari adattamenti dei protocolli di livello trasporto.

31 Questa dicitura è un marchio commerciale obsoleto, risalente al periodo in cui esistevano pochi standard. I device lo usano per il suo valore storico e per indicare una certa retro-compatibilità.

- **802.11i**:
  sicurezza **WPA**;
- **802.11F**:
  permette la mobilità delle stazioni tra diverse reti wireless tramite l'Inter-Access Point Protocol (IAAP).

Al variare delle caratteristiche del canale, il ricevitore ed il trasmettitore riducono intenzionalmente la velocità, per poi rialzarla quando il canale torna ad avere una certa affidabilità: per questo le applicazioni critiche sono incompatibili con il wireless, specie se da esse dipende la vita umana.

Frequenze diverse richiedono differenti codifiche e tecniche di trasmissione, come:

- **Frequency Hopping Spread Spectrum**:
  le tecniche di Frequency Hopping variano dinamicamente la frequenza in modo rapido e pseudo-casuale, così che le portanti del segnale si sovrappongano di rado quando più reti si intersecano;
- **Direct Sequence Spread Spectrum**:
  tecnica avanzata utilizzatissima nelle reti moderne, resistente agli errori e disponibile anche in una versione "high rate" per le reti ad alta velocità;
- **Orthogonal Frequency Division Multiplexing**[32]:
  tecnica che consente trasmissioni a 54 Mbps e che sta prendendo piede.

L'elemento base di una WLAN è il **Basic Service Set** (**BSS**), ovvero una topologia di rete formata da almeno due terminali[33]. Un BSS più completo è quello con infrastruttura, che richiede almeno due stazioni ed un AP.

32 Questa tecnica si utilizza solo sulle reti a 5 GHz, ma è retro-compatibile con 2,4 GHz.

33 Questo tipo di collegamento si dice Ad-Hoc e non richiede mai un Access Point (AP). Si utilizza in situazioni di bisogno per brevi collegamenti tra due o più stazioni che non hanno un Access Point a loro disposizione.

L'**Extended Service Set** (**ESS**) è l'unione di due o più BSS che usa un bridge di lunghezza arbitraria, formando una sola grossa rete a partire da due reti potenzialmente isolate: i BSS si possono estendere con sovrapposizione o tramite una catena di AP appositamente configurati per fare da bridge.

Le stazioni cambiano BSS da sole in base alla forza del segnale, a patto che gli AP coinvolti siano al corrente del loro spostamento tramite un protocollo IEEE. Lo standard consente di sovrapporre totalmente i BSS per creare ridondanza[34], in quando le stazioni utilizzano sempre e solo un singolo AP dell'ESS per volta.

Tutte le stazioni si devono sempre preventivamente autenticare con l'AP, che riceve e valuta la loro richiesta di autenticazione in base alla propria configurazione, per poi confermare l'identità:

- Autenticazione aperta:
  detta **Open Systems Authentication**, ove l'AP non vuole credenziali e si limita a distinguere la stazione dalle altre;
- Autenticazione con password:
  detta **Shared Key Authentication**, ove l'AP richiede una chiave di rete conosciuta a priori. Questa chiave è segreta ed è stata condivisa in precedenza dall'amministratore di rete, usando mezzi sicuri esterni alla rete stessa. L'AP ammette nella rete soltanto terminali che forniscono tale chiave di rete e rifiuta gli altri.

La privacy della trasmissione è garantita da:

- **WEP** (Wired-Equivalent Privacy):
  debole crittografia basata su pre-shared key. Ormai inutile;
- **WPA** (Wi-Fi Protected Access):
  robusta crittografia dello standard 802.11i.

34 La ridondanza tra l'altro consente all'amministratore di rete di distribuire il traffico quando la rete è oberata.

L'associazione è il meccanismo fondamentale del **roaming**[35], perché tutti gli AP del BSS ne vengono al corrente, anche in caso si tratti di ambiente ESS. La connessione ad una rete wireless avviene per passi:

1. **Channel scanning**:
   la stazione cerca di rilevare o sniffare (attivamente o passivamente) gli altri client e gli AP attivi;
2. Autenticazione:
   effettuata verso un AP del BSS;
3. Associazione:
   la stazione adotta i **parametri MAC** della rete e i criteri di trasmissione fisica indispensabili per la comunicazione armoniosa.

Il MAC (controllo di accesso al mezzo) decide chi e quando può trasmettere nella rete wireless[36], usando:

- **DCF** (Distributed Coordination Function) che usa:
  - **Carrier Sense Multiple Access**:
    variante in cui si cerca di evitare le collisioni aspettando e non si tenta affatto il rilevamento (impossibile nell'etere);
  - **RTS/CTS**:
    tecnica di richiesta/permesso, in cui la stazione chiede il permesso di trasmettere all’altra stazione e viceversa;
- **PCF** (Point Control Function).

35 Supporto alla comunicazione in mobilità tramite diversi AP.
36 Nelle reti wireless bisogna sempre inviare un ACK per ogni trama ricevuta.

12 • Internet ed IPv4

La primordiale Internet, **ARPANET**, nasce negli anni 70, ad opera di:

- **DARPA**:
  il Dipartimento della Difesa degli U.S.A.;
- **BBN**:
  collaboratore privato del DARPA;
- **Università di Stanford**.

ARPANET venne progettata sul concetto di **packet switching**, per poter scambiare informazioni tra i calcolatori eterogenei dei ricercatori. Per far questo, ARPANET si basa sulla **Internet Protocol Suite**, composta in particolare da **IP** (**Internet Protocol**) e **TCP** (**Transport Control Protocol**).

ARPANET venne però estesa con nuove reti: nacque così **Internet**, la rete composta da reti, che venne poi estesa da protocolli come **UDP** (**User Datagram Protocol**), **NFS** (**Network File System**) ed **ARP** (**Address Resolution Protocol**).

TCP/IP è un'architettura di dominio pubblico ed è divenuta uno *standard de facto* slegato da costruttori specifici e basato sulle **RFC** (**Request For Comment**): chiunque può proporre una RFC su Internet e la comunità si accorda per modificarla e rivederla, adottarla[37] o ignorarla.

Internet è basato sull'**imbustamento**[38] dei pacchetti che vengono analizzati e decomposti alla ricezione. Grazie a questa solida base, Internet ignora i livelli ISO 1 e 2: IP deve funzionare a prescindere dalla scheda di rete e dal livello fisico. IP sa come imbustare i propri pacchetti in Ethernet, IEEE 802.11, PPP ed in

37 Esistono ormai migliaia di RFC e solo alcune sono diventate parte dello standard di Internet, che quindi è solo ispirato ma non vincolato ad OSI.
38 Alias "incapsulamento" / "encapsulation".

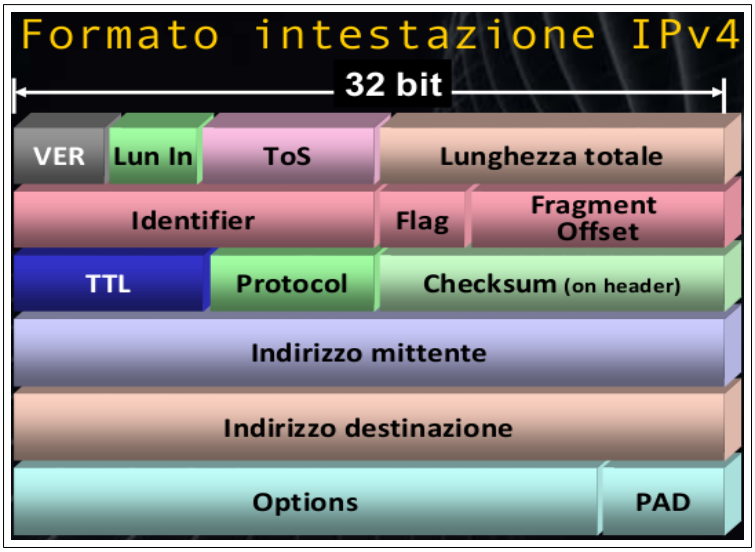
tantissimi altri standard di livello fisico, perfino se sono già obsoleti da tempo.

La rete IP è gerarchica ed usa protocolli e tecnologie eterogenee, quindi gli IS sono progettati per consentire il trasferimento di pacchetti IP tra reti eterogenee.
Per questo IP è al livello Network e si specializza in **ICMP** (**Internet Control Message Protocol**), **ARP**, **IGMP** (**Internet Group Management Protocol**) ed in vari protocolli di routing: tutti questi protocolli hanno in comune la caratteristica principale di IP, ovvero l'essere connectionless e basata su datagrammi (pacchetti indipendenti), consegnati con politica best-effort, senza preventiva negoziazione ed in modo inaffidabile. L'unico uso affidabile di IP avviene tramite i protocolli di livello superiore, come TCP.

I pacchetti IP, essendo datagrammi, possono seguire ciascuno un percorso differente ed arrivare in disordine, oltre a poter congestionare la rete o perdersi nel collegamento. Tuttavia, proprio questa inaffidabilità lo rende ideale: è così semplice da risultare robusto e di facile adozione, il che ne ha facilitato enormemente la diffusione. In particolare il traffico dati è "bursty" (impulsivo) e quindi adatto ad IP, a differenza di quello telefonico.

IP specifica:

- Formato dei pacchetti;
- **Frammentazione** e riassemblaggio;
- Formato degli indirizzi (**addressing**) ed assegnazione degli stessi;
- Instradamento (**routing**);



- **Classi di servizio**[39] del traffico.

Ciascun pacchetto IPv4 è formato da innumerevoli campi, quindi i tecnici rappresentano i pacchetti dei livelli 3 e superiori in gruppi di 32 b disposti "*a lasagna*". Un tipico pacchetto IP è composto da almeno 40 B[40] [41].

La *checksum* IP consente di rilevare errori nell'header, per scartare i pacchetti durante il routing.
Il campo *Protocol* consente l'imbustamento di protocolli superiori ed il campo **TTL** determina il tempo di vita massimo del pacchetto: queste sono funzionalità fondamentali, perché ogni router *sbusta* il pacchetto e decrementa il TTL di 1 e, se tale conteggio arriva a 0, il pacchetto viene subito scartato da chi lo riceve per prevenire cicli infiniti nella rete.
Ogni *Opzione* IP è composta da 2 B che ne determinano la lunghezza e poi dai dati opzionali di lunghezza variabile: in questo modo le stazioni possono ignorare a piacere le estensioni al protocollo IP senza dover scartare l'intero pacchetto.
Si dice **MTU** (**Maximum Transmission Unit**) la dimensione massima dei pacchetti in bytes che il livello 2 può trasportare. Vale sia l'MTU dei IS che quello degli ES: se l'MTU della rete non basta, i pacchetti più grossi devono essere frammentati da un router. I frammenti devono essere poi riordinati e riassemblati dagli ES[42], usando un timer apposito, dei campi identificativi ed ovviamente l'identità del mittente.

I frammenti hanno anche dei flag, che indicano se il pacchetto può essere frammentato e qual'è l'ultimo pacchetto di una serie di frammenti, ciascuno dei quali è identificato da uno speciale campo **offset** (multiplo di 80 B), tutto per consentire sia la frammentazione che il riassemblaggio.

39 Le classi trovano uso in tecnologie come QoS ed aiutano a decongestionare la rete e consentire una minor latenza percepita.
40 L'intestazione IP è composta da almeno 40 B di campi allineati a 4 B. Questo allineamento dipende dalla CPU.
41 L'intestazione IP contiene campi opzionali, quindi la sua lunghezza totale è memorizzata nel campo Lun In. I campi opzionali hanno un padding, riempito man mano che i campi opzionali scompaiono o diminuiscono in lunghezza.
42 In rari casi tale compito è assolto direttamente dai router.

13 • Gli indirizzi IP

Le reti IP consentono comunicazioni mediate da routers[43], IS di livello 3 che gestiscono gli host di rete inoltrandone i pacchetti.

IP ha due concetti fondamentali:

- La rete fisica;
- La sottorete logica:
  in gergo "**logical subnet**".

La rete logica è detta **LIS** (**Logical IP Subnet**) perché i routers hanno tante interfacce e per ciascuna di esse un indirizzo IP, composto da un prefisso - che designa le interfacce appartenenti alla stessa rete - e dall'identificativo dell'host. A ciascuna rete logica si deve obbligatoriamente associare una rete fisica[44] e tutti gli host devono condividere il suo prefisso, che identifica in modo univoco su Internet quella rete e la sua controparte logica.

Quindi ogni indirizzo IP è suddiviso in:

- Network part:
  il **prefisso di rete**;
- Host part:
  l'**identificativo della stazione**.

Questo dà alla rete la capacità di crescere tramite l'aggiunta di sottoreti: i router devono sapere solo dove si trovano le reti, a differenza degli switch che devono conoscere i singoli host. In tutte le reti IP i prefissi sono assegnati in modo gerarchico, mentre nelle reti Ethernet gli indirizzi sono puramente arbitrari. Proprio per questo IP ha un addressing inefficiente, in cui solo il 25% degli indirizzi sono effettivamente utilizzabili ed il resto è usato per garantire la scalabilità della rete.

43 In gergo si indicano anche con il termine obsoleto "gateway", ma questo termine dovrebbe essere utilizzato solo per gli IS di livello 4 o superiore.

44 In IPv6 le reti fisiche sono dette semplicemente "link" e la distinzione tra rete fisica e logica è ancor più labile.

Ogni IPv4 è espresso in **dotted notation** decimale ed è lungo ben 32 b, con ogni campo di 8 b con valore da 0 a 255.

13.1 • Classi di IP

La **classe** è il metodo originale usato per distinguere la network part dalla host part. Esistono 3 classi di IP con prefisso ben noto:

- **Class A** (0):
  prefisso di 1 B.
  16 M di host, il resto in prefissi;
- **Class B** (10):
  prefisso di 2 B.
  64 K di host, 16 M di prefissi;
- **Class C** (110):
  prefisso di 3 B.
  255 host, 2 M di prefissi.

...a cui si aggiungono altre classi ben note di IP speciali[45], come ad esempio:

- Class D (1110):
  indirizzi multicast;
- Class E (11110):
  indirizzi anycast.

Il valore del primo byte (in parentesi) viene usato dalle interfacce per conoscere la classe dell'IP, alla maniera dei prefissi telefonici.

Una host part con valore 255 (tutti i bit attivati) indica una comunicazione **directed broadcast** verso la rete del prefisso, che verrà ignorata dalle stazioni non interessate; una host part di 0 (tutti i bit disattivati) indica la LIS del prefisso che, pur potendo essere assegnato alle interfacce, viene comunemente usato solo

45 Le classi dalla D (compresa) in poi si usano per servizi particolari ed i loro IP non si assegnano alle interfacce. Questi sono indirizzi particolari che rappresentano più di una stazione.

nella documentazione tecnica. In ogni caso una host part di *n* bit ha 2^n-2 IP utilizzabili[46].

L'indirizzo con tutti i bit attivati (**255.255.255.255**) si dice di **limited broadcast**: i router propagano il pacchetto solo nella rete fisica di origine e non lo inoltrano in altre reti, per questo viene usato da protocolli particolari quando il destinatario di una comunicazione è nella LAN ma non è ben noto; l'indirizzo con tutti i bit disattivati (**0.0.0.0**) indica l'host di origine e viene usato dalle stazioni che non possiedono ancora un proprio indirizzo IP o non hanno una memoria permanente per ricordarselo.

I pacchetti inviati agli **indirizzi di loopback** (**127**...*) non vengono passati ai livelli inferiori, ma vengono ricevuti dal mittente stesso, per fare test di rete senza far uscire il traffico dalla stazione.

13.2 • Netmask IP

Il **classfull addressing** visto in precedenza spreca molti IP, perché la sua **granularità di assegnazione** non è abbastanza fine, e richiede la gestione di un ente centrale globale, tutt'oggi impensabile. Per superare il limite delle classi sono state create le **netmask**: le organizzazioni possiedono un prefisso classfull[47] e poi usano la netmask per allungarlo, estendendo la network part dell'indirizzo a discapito della host part. Le netmask quindi si usano nel **classless addressing** e si associano a ciascun IP di ciascuna interfaccia della LIS. Ciascuna netmask è una sequenza di bit lunga quanto la sequenza di bit che compone l'IP a cui è associata: ogni bit attivo della netmask designa il prefisso, che termina quando iniziano i bit inattivi.

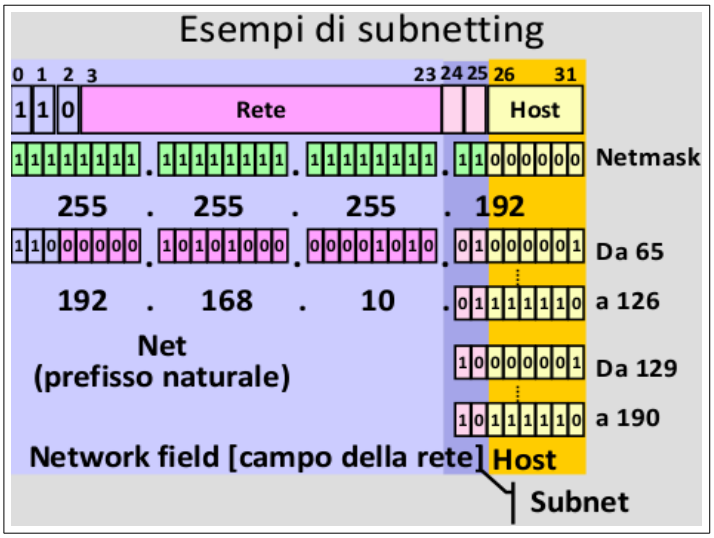
46 L'usanza di utilizzare il primo IP nella documentazione è tanto radicata che alcuni vecchi router bloccano questi pacchetti. Le nuove reti dovrebbero poter utilizzare l'IP X.X.X.0 come qualsiasi altro IP, senza questo handicap e quindi la formula più corretta è 2n-1.

47 Nel gergo dell'addressing si chiama "prefisso naturale".

Si dicono "netmask naturali" quelle netmask associate univocamente ad una classe standard di IP. Ad esempio, le netmask dei prefissi naturali (da tenere a mente) sono:

- Class A:
  255.0.0.0;
- Class B:
  255.255.0.0;
- Class C:

  255.255.255.0,
  usata per i privati.



Queste netmask sono utili perché la classe dell'IP viene ignorata nel classless addressing, a favore della netmask esplicita.

Le netmask sono utili perché permettono il subnetting (prefisso più lungo di quello naturale) o il supernetting[48] (prefisso più corto di quello naturale). L'estensione del prefisso naturale creata per il subnetting si chiama **subnet**[49], mentre il suo prefisso naturale si chiama **net**.

48 Solo i router usano il supernetting, per ridurre la quantità di informazione da elaborare.

49 Quanto la netmask non è multipla di 8, conviene rappresentarla in binario per capire quanti host per subnet può contenere.

14 • Routing IP

IP è progettato per l'architettura gerarchica di Internet, ove una rete logica è composta da stazioni che condividono il medesimo prefisso ed equivale alla rete fisica. I pacchetti IP possono essere inviati dal mittente fino al destinatario direttamente incapsulandoli in una trama di livello 2 (Data Link) o indirettamente su una diversa rete, inviandoli al **default gateway** preconfigurato, ovvero il router che farà l'inoltro scartando l'imbustamento del livello 2 che sarà poi ripristinato dagli altri IS.

Ciascun host deduce la posizione logica del destinatario tramite il confronto dei prefissi IP e, se questi non è della sua stessa rete, invia il pacchetto al default gateway: gli host con il medesimo prefisso scelgono automaticamente di non far uso del default gateway.

Per convenzione, le interfacce dei router hanno l'indirizzo host più alto (max-1, max-2, ...) o più basso (min,min+1,...) disponibile nella subnet ed ovviamente posseggono lo stesso prefisso della rete che gestiscono. Nella rete quindi l'assegnazione degli indirizzi IP è fondamentale per tutte le stazioni: ogni host con indirizzo IP classless deve avere la medesima netmask del resto della LIS. L'eccezione a questa regola è la tecnica del **variable subnetting**, in cui due reti fisiche collegate possono avere netmask che differiscono in lunghezza.

14.1 • Prefix matching

Gli host usano la tecnica del **prefix matching** per capire se appartengono alla stessa LIS: l'indirizzo del mittente e l'indirizzo del destinatario vengono trattati come sequenze di 32 b e confrontati. Nella pratica, il calcolatore fa l'**AND bit-a-bit** (bitwise AND in un singolo colpo di clock) con la bitmask binaria ed ottiene così il prefisso dell'IP con la parte host azzerata. Questo valore speciale viene confrontato con lo stesso tipo di valore preso

dall'IP di destinazione mediante lo stesso identico calcolo. Il calcolatore deduce dal confronto se il destinatario appartiene alla sua stessa LIS: se i due valori corrispondono non serve coinvolgere il default gateway!

I router applicano la tecnica del prefix matching su ogni interfaccia, per capire su quale di esse debba essere inoltrato il pacchetto in entrata. Se i prefissi corrispondono allora il router (che acconsentirà sempre al massimo ad un solo match positivo) può reincapsulare il pacchetto in una trama Ethernet ed inoltrarlo.

La **routing table** (tabella di instradamento) del router ha una struttura particolare, ove ogni riga contiene almeno:

- Destinazione:

  una coppia **<IP della rete, netmask>**. Le netmask possono differire in lunghezza[50] a partire dal medesimo prefisso naturale;

- **Next hop**:

  IP del prossimo router a cui inoltrare i pacchetti per una certa destinazione. Ha lo stesso prefisso di una delle interfacce del router e quindi vi è collegato direttamente.

Se non c'è matching il pacchetto viene scartato e non viene inviato in broadcast, a differenza di come accade nel bridge e nello switch. Se più di una riga fa match, il router predilige la destinazione con il prefisso più lungo e quindi sempre quella con la netmask con meno host.

Alcune peculiarità delle routing tables da tenere ben a mente sono:

- Si dice **default route** una entry speciale che ha sia l'IP che la netmask azzerati esplicitamente, in modo che i pacchetti che non matchano altre righe vengano gestiti da essa ed inviati al suo next hop invece che essere scartati.

50 Le netmask superiori a 30 b ammettono un solo host e quindi non hanno senso ai fini del routing.

- Se ci sono linee con route più generiche (netmask brevi) seguite in ordine da route più specifiche, il router usa in ordine le route più generiche come default route in base al contesto e solo in extremis ricorre alla default route del primo punto.

L'addressing e le prestazioni del routing sono tematiche collegate: è importante avere una corrispondenza tra reti fisiche e LIS, usare netmask mirate e tabelle di routing ottimamente progettate.

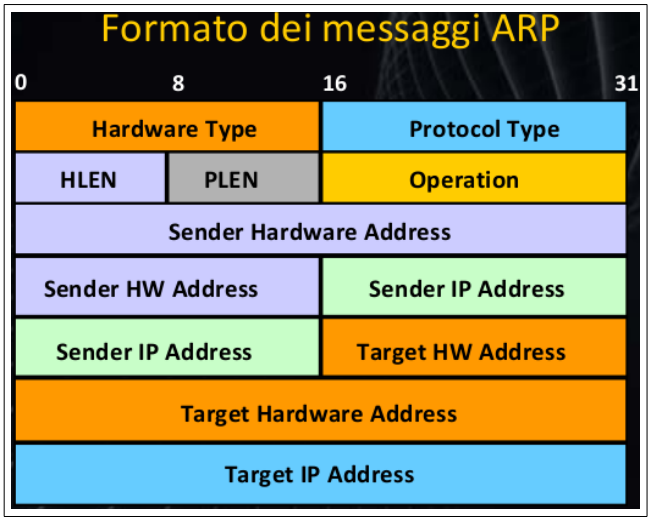
15 • ARP ed ICMP

15.1 • ARP

ARP (**Address Resolution Protocol**) è un protocollo di servizio che si pone al livello 3 e che aiuta il livello Transport a decidere la destinazione. ARP è di tipo "solicitation" con broadcasting, quindi invia richieste a tutti gli host della LIS. Esso viene imbustato in trame di livello 2 e consente di trovare e memorizzare in una speciale cache la corrispondenza tra indirizzi MAC ed IPv4[51], creando quindi l'**ARP cache**.

Quando una stazione deve inviare un pacchetto genera una **ARP Request** solo se l'indirizzo MAC non è nella ARP cache, così da poter poi inviare pacchetti IP ad un indirizzo noto. La richiesta ARP contiene l'IP del mittente e del destinatario[52], ma ha anche uno spazio vuoto che il destinatario potrà riempire con il proprio indirizzo MAC. Siccome la richiesta ARP è broadcast, tutti gli host (router compresi) la ricevono e rispondono solo se l'IP del destinatario nel pacchetto coincide con il proprio; chi non risponde, specialmente se è un router, sfrutta il pacchetto ricevuto per popolare in anticipo la propria ARP cache con il nuovo MAC, in previsione di future comunicazioni bidirezionali.

La ARP cache ha validità limitata e scade quando non vengono visti più pacchetti dal MAC/IP in scadenza e può anche mancare del tutto, perché ARP suggerisce ma non impone il caching. I pacchetti ARP devono essere processati dalla CPU di chi li riceve e quindi può causare un rallentamento



51 IPv6 non usa ARP ma il Neighbor Discovery Protocol (NDP), che non richiede broadcasting.
52 Indicato anche come "target IP".

di tutte le stazioni collegate alla rete, ma questo non è un problema perché al giorno d'oggi - con l'avvento di Internet - gran parte delle stazioni si limitano a contattare direttamente solo il router gateway.

HLEN e PLEN sono la lunghezza degli indirizzi di livello 2 e 3, mentre i campi apparentemente duplicati sono in realtà singoli campi di lunghezza variabile.

15.2 • Proxy ARP

L'**ARP Proxy** è una tecnica per permettere l'uso di uno stesso prefisso su due reti fisiche diverse, nel caso in cui la configurazione di rete sia errata[53] [54]. Viene usata dai router che ricevono molte ARP Request di consecutive: il router reagisce attivando il proprio algoritmo di ARP Proxy, che inizia a fare l'inoltro di ARP e delle sue risposte sulle reti fisiche limitrofe. IPv6 utilizza una versione modificata e perfezionata di ICMP, che include le funzionalità di ARP senza la necessità di proxy appositi.

15.3 • Reverse ARP

Talvolta si conosce l'indirizzo di livello 2 ma non quello di livello 3, rendendo impossibile la comunicazione IPv4. In questo caso viene usato il protocollo **RARP**[55] (Reverse ARP) per recuperare l'IP del destinatario a partire dal suo MAC. Ricorre ad ARP[56], ad esempio, chi non conosce il proprio IP (come le stazioni diskless) e non può affidarsi al DHCP. Chi usa RARP genera una richiesta MAC verso l'IP di broadcast della subnet, specificando come MAC del mittente e del destinatario il proprio indirizzo: chi riceve questa richiesta particolare, include nella risposta sia il proprio IP che l'IP del mittente.

53 Queste cattive configurazioni si creano quando una rete cresce nel corso del tempo e si verifica un degrado prestazionale (dovuto ad Ethernet) tale da imporre la suddivisione della rete con un nuovo router, riducendo così il numero di stazioni che competono per la banda.

54 Talvolta queste cattive configurazioni originano da un cattivo addressing dell'amministratore di rete, che sbaglia la netmask di due reti fisiche limitrofe, ad esempio quando si dimentica di specificare la netmask di una rete ed il SO ricorre a quella di default per la classe (tipicamente /24), costringendo il router all'inoltro verso le destinazioni più specifiche.

55 RARP usa Ethernet con l'Ethertype 0x8035.

56 ARP usa tipicamente Ethernet con l'Ethertype 0x0806.

15.4 • ICMP

ICMP[57] (**Internet Control Message Protocol**) si usa per verificare il funzionamento del livello 3. Esso notifica una condizione ma non specifica mai una reazione ad essa, nè impone la risposta. Per questo ICMP è sempre opzionale e può essere ignorato dagli host di rete, anche se il suo uso permette di:

- Notificare le anomalie;
- Scoprire la netmask della rete;
- Scegliere il default gateway migliore per la stazione.

ICMP è un protocollo particolare, perché i suoi datagrammi contengono un tipo, un sottotipo, il checksum e molte altre informazioni, ma ogni tipo di messaggio ICMP ha una struttura diversa e pacchetti specializzati con tante varianti, oltre ad includere parte dell'header del pacchetto che ha causato l'anomalia, ove possibile.

Le richieste ICMP *echo*[58] sono numerate sequenzialmente e le risposte si possono quindi associare a ciascuna richiesta, per fare debugging di rete; le risposte/notifiche ICMP *destination unreachable* e le loro varianti sono generate dai router che falliscono l'inoltro; ICMP *redirect* viene generato dai default gateway che rilevano che il destinatario è sulla stessa LIS e quindi il mittente può fare connessione diretta; ICMP *time exceeded* si usa per il traceroute ed altre finezze.

57 ICMP ha IP Protocol pari a 0x01.
58 Il tipo "echo" spesso ignorato dagli host e scartato dagli IS per migliorare la sicurezza della rete.

16 • IP Transport layer

TCP ed UDP sono protocolli di livello Transport, che si pongono agli antipodi: con UDP le app mandano sequenze di datagrammi[59] contenenti ciascuno un messaggio indipendente, usando il criterio inaffidabile best-effort; con TCP le app mandano sequenze di pacchetti che compongono messaggi, usando una connessione affidabile.

16.1.1 • Multiplexing e demultiplexing

Il **multiplexing/demultiplexing** consente a più app di utilizzare il livello Transport che fornisce i meccanismi per distinguere i pacchetti di ciascuna app in esecuzione, sia in trasmissione sia in ricezione, usando il concetto di **porta di rete** ignorato dai livelli inferiori. Per questo ogni pacchetto IP ha sia una **source port** che una **destination port**, entrambe specificate all'inizio della sua intestazione come campi di 2 B.

Le porte da 0 a 1023 sono dette **porte statiche** e sono fatte per processi in esecuzione permanente (tipicamente i processi server) che necessitano di utilizzare sempre lo stesso numero di porta, ben noto anche alla controparte client; le porte da 1023 in poi sono dette **porte dinamiche**, perché sono in genere usate dai processi utente ed in generale da tutti quei processi che non necessitano di un numero di porta stabile tra una runtime e l'altra.

Le RFC specificano tuple <porta, protocollo> per tutti i servizi standard o comunemente utilizzati, detti "well known services"[60].

La IANA ha deciso tutte le porte inferiori a 1024. Dal 2001 sono gestite da ICANN, che fa le veci di IANA.

59 Dal punto di vista del livello applicativo possono essere inviati uno per volta.

60 I server standard hanno un file /etc/services che permette alle applicazioni di risalire alle porte/protocolli da usare per il nome di un servizio e viceversa. Su GNU/Linux ci sono API apposite.

Ciascuna comunicazione è identificata in modo univoco da:

- IP sorgente;
- IP destinazione;
- Protocollo di livello 4;
- Porta del mittente;
- Porta del destinatario.

...o come ennupla:

<SIP, DIP, Proto, SPort, DPort>

Ad esempio una telefonata VoIP è fatta da due app su due host, applicazioni che usano numeri di porta specifici in modo esclusivo su ciascun EP.

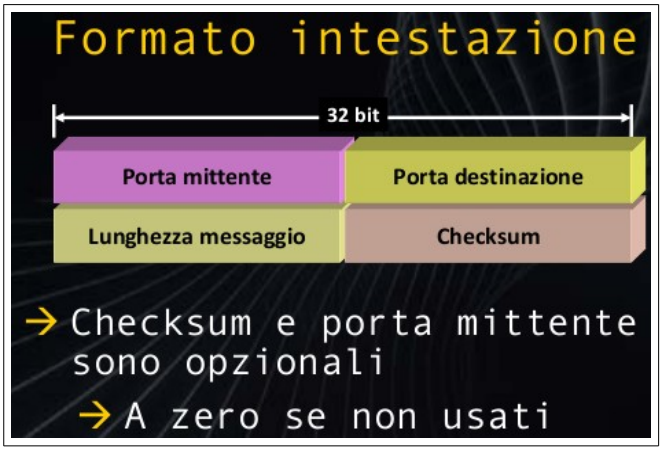
16.2 • Protocolli datagram

I protocolli basati sui datagrammi sono connectionless, non fanno uso di negoziazione iniziale[61] ed usano messaggi indipendenti che sono privi di numerazione ed ordine. I messaggi nei datagrammi possono andare persi o giungere in disordine, eventualità che l'app deve gestire ad alto livello[62].

61 La negoziazione iniziale è necessaria solo quando si deve intraprendere una connessione.

62 IP ed UDP sembrano fornire lo stesso servizio di UDP, ma quest'ultimo ha il vantaggio di fare multiplexing e quindi di poter essere usato da più app su diverse porte.

16.2.1 • User Datagram Protocol



In particolare, oltre al semplice multiplexing, UDP fornisce funzionalità opzionali come la checksum (con scarto automatico a basso livello dei pacchetti corrotti) ma è comunque **privo di controllo della congestione**: può saturare i SI ma non si ferma mai durante l'invio, seguendo rigorosamente la politica best-effort[63].

UDP conviene quando il ritardo della connessione è inaccettabile o l'overhead di apertura/chiusura delle stesse è eccessivo per il sistema. UDP conviene anche perché non mantiene lo stato (risparmi memoria) ed ha un'intestazione piccola rispetto a quella del TCP. In generale, tutte le app che non tollerano il controllo di congestione usano UDP.

La porta del mittente può essere omessa quando non ci si aspetta una risposta e perfino il checksum è opzionale.

Ad esempio, i protocolli affidabili come **NFS** (Network Filesystem, condivide in rete dischi tra SO Linux) usano l'inaffidabile UDP, perché sono utilizzati solo su percorsi brevi con collegamenti fisici affidabili: sarebbe comunque troppo lento accedere a dischi lontani e quindi ci si specializza in reti affidabili gestendo i pochi errori ad alto livello.

63 L'UDP non è "altruista" ed approfitta delle app che usano TCP, le quali si bloccano quando la rete è congestionata...

Altri utilizzi di UDP sono i compiti remoti periodici, le telefonate etc. ove la perdita di qualche dato non causa danni irreversibili o non compromette troppo la qualità del servizio, ma è comunque importante ridurre al minimo la latenza di rete. Altro caso è il SNMP (Simple Network Management Protocol) che usa UDP per le sue qualità best-effort: SNMP contatta apparati limitrofi in caso di problemi ed innesca reazioni specifiche nelle applicazioni.

UDP viene infine usato quando bisogna inviare un solo messaggio di pochi bytes, come nel caso dei sensori di temperatura e durante lo streaming di contenuti multimediali.

Le app multimediali, infatti, hanno causato un grosso aumento del traffico UDP su Internet, che da tempo viene gestito in modo particolare dagli ISP.
In generale UDP conviene alle app dal funzionamento legato al tempo e quindi "poco elastiche". Ad esempio il trasferimento di file privilegia sempre TCP, mentre lo streaming video l'UDP.

16.3 • Transport Control Protocol

TCP crea **connessioni full-duplex** (quindi bidirezionali) e richiede alla app l'invio di messaggi che vengono riordinati arbitrariamente dal protocollo in flussi coerenti di bytes. Per questo le app TCP devono ricevere in continuazione byte-per-byte in un **ciclo iterativo indeterminato, senza prefissare la quantità di dati ricevuti per ogni lettura dal socket**. Tuttavia l'ordine di ricezione dei dati è garantito, così come la loro integrità ed il corretto ed affidabile trasferimento end-to-end.

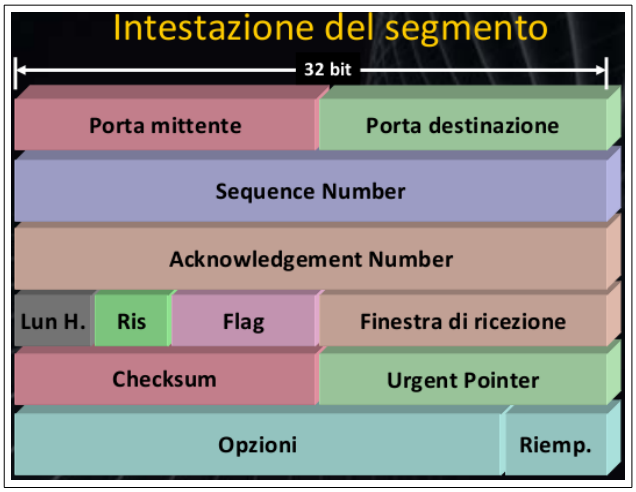
Protocolli affidabili come FTP, SMTP, HTTP, etc. richiedono TCP per la sua affidabilità e per le seguenti caratteristiche fondamentali:

- **Gestione delle connessioni**;
- **Multiplexing**:
  alla pari di UDP;

- **Controllo di flusso**:
  il ricevente può richiedere di rallentare l'invio;
- **Controllo di congestione** di rete:
  non può sovraccaricare gli IS;
- **Uso di ARQ**:
  garantisce l'integrità dei dati tramite la ritrasmissione automatica;
- **Segmentazione** del flusso di dati:
  causa la discrepanza tra invii e ricezioni, ma garantisce la consegna in condizioni avverse.

È bene ricordare che TCP ha anche un grosso overhead e fa spesso ritrasmissioni inutili.

17 • TCP in dettaglio



I pacchetti TCP vengono chiamati ***segmenti****.*

- Il *sequence number* viene usato per coordinare la trasmissione e numera il primo byte della sequenza trasmessa;

- L'***ack****nowledgement number* contiene il numero di sequenza che il mittente si aspetta.

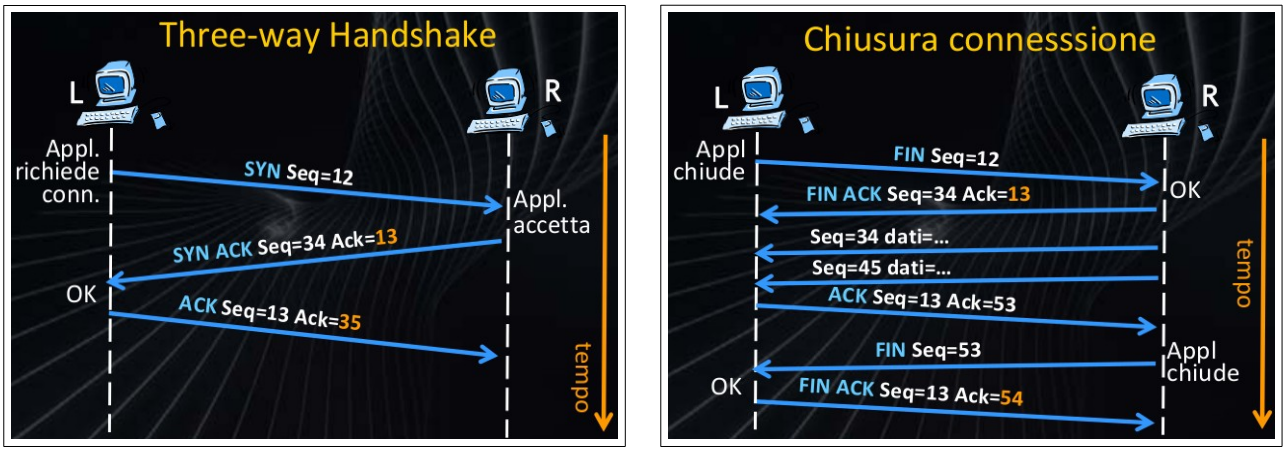
17.1 • Dettagli sui campi

Siccome la comunicazione TCP è basata su ACK, alcune implementazioni allegano ai dati il valore di *acknowledgement number*, evitando così la doppia trasmissione tramite **ACK piggybacking**.

- Il flag *ACK* ha valore diverso da 0 se il mittente ritiene che l'ACK trasmesso con i dati della sua porta debba essere preso in considerazione.
- Il flag *PSH* (detto *push flag*) indica che i dati della trasmissione/ricezione vanno passati direttamente ai livelli superiori, che li gestiranno in special modo.

- Il flag *URG* è attivo quando *Urgent Pointer* contiene un valore valido e quindi deve essere preso in considerazione: questo campo indica che alcuni dati - ma non tutti - sono urgenti e devono essere passati ai livelli superiori (con un meccanismo diverso dal *push*) e si usa in casi particolari, per questo viene attivato da un flag separato.

Il padding delle opzioni è utile perché queste possono essere omesse.



Handshakes di inizio e fine connessione.

Il flag **SYN** viene attivato quando si rende necessario sincronizzare i numeri di sequenza, tipicamente all'apertura della connessione; il mittente attiva il flag **FIN** per avvertire che è pronto a chiudere il proprio lato in uscita; il flag **RST** si usa per resettare (chiudere) la connessione tempestivamente sia in entrata che in uscita su tutti e due gli EP.
RST viene usato quando la rete è disturbata e la trasmissione deve essere riavviata per poter tornare sincrona o quando, come nel caso di HTTP, l'utente può interrompere anticipatamente la comunicazione ed RST può essere usato per risparmiare banda.

17.2 • Gestione degli errori

TCP usa **go-back-N**[64] e quindi ogni errore causa la ritrasmissione di un gruppo di bytes, perché in genere su TCP viene sempre

64 Per evitare ritrasmissioni inutili, alcune implementazioni usano la ritrasmissione selettiva.

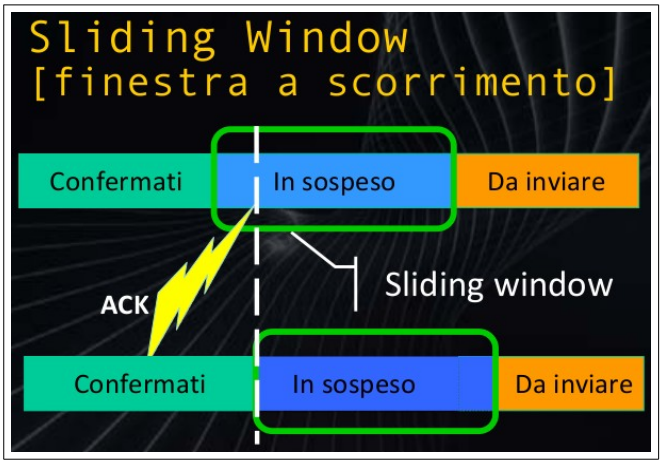
perso più di un pacchetto. La ritrasmissione avviene quando il timer del mittente scade prima della ricezione del relativo ACK, ma in compenso il timer del segmento "outstanding" (inviato ma non confermato dal destinatario) viene raddoppiato ad ogni ritrasmissione. Il giusto tempismo dei timer è fondamentale, quindi il trasmettitore cerca di stimare continuamente il RTT[65] della rete.

TCP può ridurre il numero di ACK inviando "ACK cumulativi" dei precedenti, ovvero ACK che hanno un numero successivo al semplice incremento e quindi indicano la ricezione di un gruppo di pacchetti. Questo si implementa con la tecnica del **delayed acknowledgement**, che ritarda l'invio dell'ACK il più possibile, in modo da accumulare dati e poi confermarli in blocco. Quando l'ACK accumula tanti dati si usa la tecnica del **fast retransmit**, ove si inviano più ACK uguali e consecutivi per indicare che un vecchio pacchetto è stato perso: il mittente userà go-back-N per ovviare e quindi verranno inviati più dati ridondanti.

17.3 • Controllo di flusso

TCP usa una **sliding window** che è composta dai bytes in sospeso. I dati da inviare saranno raggiunti dalla finestra ed inviati, mentre quelli confermati dall'altro EP vengono via via scartati. La finestra, di volta in volta, prende più bytes possibili e li invia per poi bloccarsi quando è piena di bytes in sospeso. Se la finestra è abbastanza grossa la trasmissione è continua, ma non può essere troppo grossa per evitare di esaurire la memoria del canale, del processo o di saturare la rete...

65 Ci sono molti studi in merito, nonché molte varianti dell'algoritmo di stima.



Il ricevitore ha un buffer ove tiene e riordina i bytes ricevuti prima di elaborarli: se è pieno è costretto a scartare i pacchetti. Il campo TCP *receiver window* informa di volta in volta il trasmettitore della dimensione di questo buffer. Il trasmettitore si assicura di ridurre la finestra in modo che sia sempre più piccola del buffer del ricevitore.

17.4 • Controllo di congestione

La congestione della rete con go-back-N innesca un ciclo di rinvii che aggrava la congestione stessa[66].
Per prevenirlo il trasmettitore riduce la finestra in base al numero di ACK duplicati ricevuti ed al numero di time-out, per poi ingrandirla gradualmente quando la congestione si riduce. L'algoritmo di ingrandimento chiamato **slow start**, segue prima un andamento esponenziale e poi lineare, mentre l'algoritmo di riduzione la divide o la resetta.

> Se le entità di trasporto di molte macchine inviano troppo velocemente troppi pacchetti nella rete, questa si congestionerà con decadimento delle prestazioni, in quanto i pacchetti vengono ritardati e persi. Controllare la congestione per evitare questo problema è una responsabilità combinata dei livelli di rete e di trasporto. La congestione avviene nei router, quindi viene individuata dal livello di rete. Tuttavia la congestione è in ultima analisi causata dal traffico inviato in rete dal livello di trasporto. L'unico modo efficace per

66 Nel 1986 questo ciclo vizioso ha bloccato Internet ed è stato quindi aggiunto il controllo di congestione.

controllare la congestione è che i protocolli di trasporto spediscano i pacchetti in rete più lentamente. [...][67]

67 Cit. da *Fondamenti di Reti di Calcolatori*, cap. 6.3, §1.

18 • Domain Name System

Un indirizzo IP può essere associato - per comodità dell'utente finale - a più nomi di host, mentre ciascun nome di host può nascondere più indirizzi IP per fare load balancing.
I **CDN** (**Content Delivery Network**) fanno load balancing inviando i contenuti all'utente sempre dal server ad esso più vicino.

Il file UNIX */etc/hosts* conserva la corrispondenza locale tra IP e nomi di host, ma offre un meccanismo difficile da replicare e che quindi non scala bene su reti di grosse dimensioni. DNS risolve il problema specializzandosi nei grossi network e costruendo un database distribuito con struttura gerarchica. Tale gerarchia è slegata dalla gerarchia di routing di ciascuna rete e può essere utilizzata e modificata indipendentemente per accomodare ogni esigenza.

Un nome DNS completo è detto **FQDN** (**Fully Qualified Domain Name**) e possiede un numero variabile di livelli gerarchici separati dal carattere punto, tipicamente con valore semantico. Questo tipo di hostname ha una versione internazionalizzata che consente l'uso di due caratteri ASCII per rappresentare i caratteri composti o peculiari di molte lingue.

18.1 • Domini di primo livello

I **TLD** (**Top Level Domains**) sono i domini principali, posti all'estrema destra di ciascun FQDN. Essi hanno tre varianti:

- **ccTLD** (country code TLD):
  geograficamente localizzati in una specifica nazione e registrabili da chiunque in base alle leggi territoriali;
- **gTLD** (generic TLD), che si dividono in due categorie:
  - liberi:
    non vincolati a nessuna nazione specifica e

registrabili da chiunque in base alla finalità. Ad esempio: com, net, org, info, etc.;

- sponsorizzati:

  vincolati alle organizzazioni che ne hanno chiesto la creazione e mantengono attivi i loro server. Ad esempio: gov, edu, etc.;

18.2 • Domini inferiori

I livelli inferiori seguono il TLD verso sinistra e sono distinti dal punto. Lo scopo di questi domini è rappresentare l'azienda o l'organizzazione che li ha registrati o, come nel caso di bt.co, co.uk, ac.uk, etc., caratterizzare il TLD con un ulteriore livello di organizzazione.

Tutti i domini principali sono gestiti da IANA, che decide quali sono i TLD. IANA delega poi le operazioni sui domini (come la registrazione di un nuovo dominio) a vari gestori di TLD che gestiscono i DNS per ciascun dominio. Ogni dominio di secondo livello ha un server che conosce i nomi degli host ed i loro indirizzi, oltre che i DNS dei livelli inferiori, detti sottodomini.

DNS consente questa gestione tramite configurazione manuale: esiste un **root name server** gestito da IANA e replicato su più macchine[68], a cui si può accedere tramite più domini per fare load balancing e che conosce gli indirizzi dei TLD DNS servers. Questi server radice sono ben noti e sono gestiti direttamente da IANA, che si assicura fisicamente la loro reperibilità.

L'ultimo livello DNS è quello che conosce i nomi delle stazioni a cui fa capo il FQDN.

68 Vi si può accedere tramite i sottodomini di root-server.net, che sono hard-coded in molte applicazioni, tipicamente come a.root-server.net ... m.root-server.net.

Un calcolatore può ospitare[69] più server differenti e più calcolatori ridondanti si sincronizzano: i server secondari si sincronizzano con i primari e forniscono le stesse identiche informazioni.

18.3 • Risoluzione degli hostname

Ciascuna stazione deve avere l'indirizzo di uno o più server DNS che, grazie all'imbustamento di TCP/IP, possono essere ovunque e tramite la ridondanza si possono dislocare per il mondo in modo da ridurre la latenza nella risposta alle query degli utenti. Le stazioni possono avere anche un **dominio di default**, usato dal SO per completare in automatico i nomi di dominio composti solo dal nome della stazione.

La **risoluzione dei nomi** di dominio è un'operazione complessa che restituisce molte informazioni utili a chi la richiede. La risoluzione può avvenire in due modi:

- **Ricorsivamente**:

  funziona "a staffetta":

  1. il server DNS contattato dal client, se non dispone delle informazioni richieste, contatta il root name server;
  2. il root name server trova l'indirizzo della gerarchia DNS più consona alla richiesta originale;
  3. il root name server contatta il server più in alto nella gerarchia del TLD;
  4. il server del TLD contatta il server del livello inferiore, che contatta quello del livello ancora più in basso e così via...;
  5. se nessun server aveva in memoria l'informazione l'ultimo server della gerarchia è quello che risponderà con l'IP della stazione;

69 Il service provider che consente la registrazione dei nomi può anche offrire il DNS hosting ai propri clienti, perché il DNS dovrebbe sempre essere on-line e quindi delegare la sua gestione e la sua manutenzione in genere è conveniente, in special modo per i privati. A fronte di un canone il provider tiene attivo il server DNS e consente la sua configurazione remota grazie al fatto che la gerarchia DNS non è vincolata dal routing e quindi il DNS server può essere esterno alla rete del cliente.

6. l'informazione viene riflessa di server in server, memorizzandola, fino al server radice del TLD;
7. il server radice del TLD risponde al root name server con i dati della query;
8. il root name server risponde al DNS del client con i dati della query;
9. il DNS del client risponde al client fornendo i dati della query. Tale risposta è autorevole solo se non viene da una cache.

- **Iterativamente**:
  funziona tramite richieste dirette:

1. il server DNS contattato dal client, se non dispone delle informazioni richieste, contatta il root name server;
2. il root name server trova l'indirizzo della gerarchia DNS più consona alla richiesta originale;
3. il root name server non contatta il root server della gerarchia, ma invia l'indirizzo di quest'ultimo nella risposta al DNS del client;
4. il DNS del client contatta il DNS radice del TLD, che a sua volta fornisce un nuovo indirizzo da contattare, che verrà interrogato e fornirà un nuovo indirizzo etc.;
5. se l'indirizzo da contattare non era in nessuna cache (alla pari dei dati della query) allora tutti i server della gerarchia del FQDN sono stati contattati dal DNS del client e rimane solo il DNS che conosce l'IP della stazione;
6. il DNS che conosce l'IP della stazione viene contattato dal DNS del client e risponde con i dati della query;

7. il DNS del client risponde al client fornendo i dati della query. Tale risposta è autorevole solo se non viene da una cache.

Si preferisce sempre la modalità ricorsiva perché consente a più server di fare caching e riduce enormemente gli sforzi del DNS del client, oltre che la latenza. Le risposte **non autoritative** sono fornite dalle cache e quindi possono essere obsolete e devono essere accompagnate dall'IP del DNS ufficiale per quel FQDN, da contattare per ottenere una risposta autorevole ed aggiornata.

I DNS forniscono su richiesta del client anche **record MX** (indirizzo del server di posta per il dominio), **record CNAME** (alias del nome canonico dell'host), **record NS** (server DNS autoritativo per il dominio), etc.

DNS consente anche la **risoluzione inversa** fornendo il nome canonico a partire da un IP. Questo meccanismo è reso possibile dal tipo di record **PTR** ("pointer"), che viene consultato quando la query richiede la risoluzione di un nome fittizio nella forma:

x.y.z.y.IN-ADDR.ARPA

IN-ADDR.ARPA (“*INernet-ADDRess*”) è gestito da IANA ed è un DNS speciale della gerarchia DNS “ARPA” (del tutto simile ad es. “COM”) che si ramifica in base ai campi dell’IPv4 (*x.y.z.y*). Il PTR record memorizzato in un qualsiasi server della gerarchia DNS contiene una corrispondenza IP-nome che si basa su questa gerarchia. Se l’IP è 216.58.198.35, “35” sarà l’ultima ramificazione, “198” la penultima e così via fino a raggiungere “in-addr”, ove solo “35” punta al nome di dominio.

I messaggi DNS sono molto articolati ed hanno un campo *parametri* pieno di flags utili. A tutti i campi DNS è associato un **lifetime**, che determina - alla sua scadenza - lo scarto dalle cache. Il DNS ha anche estensioni come DDNS (Dynamic DNS, per permettere la configurazione automatica) e DNSSEC (DNS SECurity, per rendere sicura la configurazione automatica).

19 • Protocolli di livello applicativo

I protocolli applicativi possono appoggiarsi sui protocolli del livello Transport, come TCP ed UDP, e fornire quindi anche le funzionalità inferiori (simulando i livelli Session e Presentation) o essere puramente applicativi appoggiandosi sui livelli di Session e Presentation (come fa NFS).

19.1 • Paradigmi

Le app TCP/IP sono basate sul paradigma client-server, con il processo server in esecuzione permanente ed il client che lo contatta tramite IP/hostname e porta (tipicamente statica e ben nota). FTP, WWW, e-mail, etc. usano questo paradigma, mentre altri protocolli si affidano al paradigma **P2P** (**Peer-to-Peer**), in cui gli host non hanno un ruolo predefinito e ciascuno di essi può essere contattato o contattare: nel modello P2P tutti i processi sono client-server e sono detti genericamente **peer**. Ogni peer può conoscere gli altri chiedendo i loro indirizzi ad un peer specializzato, detto **super peer** e noto a priori. P2P è un modello recente usato da app famose come eMule, Skype ed in generale dalle piattaforme di file sharing.

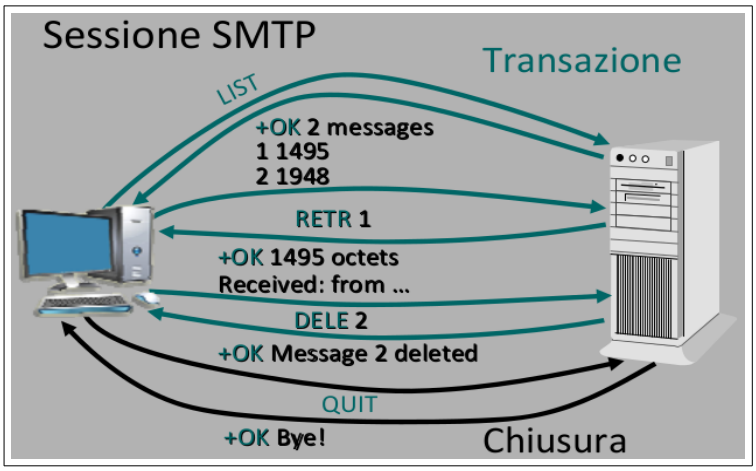
Quasi tutti i protocolli applicativi sono testuali ed inefficienti rispetto a quelli dei livelli inferiori in termini di spazio: il vantaggio è nella ricerca di guasti tramite sniffing ed in uno sviluppo più rapido del protocollo stesso.

19.2 • Posta elettronica

La **posta elettronica** è basata sull'architettura client-server, ove il server si dice **mail server** o **post office**. L'utente usa un client di posta per trasferire il messaggio da e verso il server: normalmente un server svolge tutte e due i compiti di assistenza all'utente, ma si possono avere server separati. La posta rimane nel server fino a che l'utente non la recupera per poi cancellarla.

Il protocollo per la consegna ed il trasferimento da server a server è l'**SMTP**. (**Simple Mail Transfer Protocol**), mentre i protocolli per il recupero possono essere HTTP (le **webmail**), **POP** (**Post Office Protocol**), **IMAP** (**Internet Message Access Protocol**), etc. e sono usati esclusivamente dall'utente.

Con SMTP[70], un mail server ben noto a priori e preconfigurato a mano, detto **outgoing mail server**, gestisce la posta in uscita.
SMTP è basato sul modello client-server ed usa un **pattern command-response**, ove ogni comando riceve uno **status code** in risposta, seguito da un messaggio intelligibile preconfigurato a piacere dall'amministratore del server.
Il client SMTP usa comandi di 4 lettere, spesso volutamente sgrammaticate (come "HELO") e se riceve uno stato che inizia con 2 ("2XX") è andato tutto bene. Un tipica sessione SMTP



funziona così:

Questa è la **fase di presentazione** o "**handshaking**" del protocollo SMTP. Segue il comando "DATA" e poi il server richiede l'immissione dei dati dell'email, fornendo tra l'altro istruzioni "amichevoli" per farlo da un terminale... Tipicamente il client legge il messaggio dall'utente, lo scrive e lo termina su una riga contenente solo il carattere punto ed un new line per simulare la pressione di ENTER. Il server lo accetta con "250 Messaggio accettato" (o qualcosa di simile, varia), il client invia "QUIT", il server saluta e poi la connessione si chiude. **Telnet**, che

70 SMTP: 25/TCP, text-based.

permette di fare connessioni TCP verso porte arbitrarie di macchine remote, consente di simulare il client per fini diagnostici.

I server si tutelano dallo SPAM non accettando i messaggi da fonti non preventivate e verso destinazioni non ben note, in modo da ridurre al minimo le probabilità che insorgano problemi legali per colpa di server ignoti e client maliziosi: il codice di errore 571 indica che il server gestisce messaggi solo per uno specifico dominio di posta e non fa inoltro (tipico errore "...we do not relay!").

I server di posta accettano sempre i messaggi per gli account del loro dominio o locali, perché SMTP è usato anche tra server. Nessuno vieta ai client di fingersi altri (salvo la legge, che però ha molti bug...), quindi i server potrebbero anche filtrare certi range di indirizzi IP ed imporre dei meccanismi di autenticazione. Comunque tra server SMTP non ci si autentica ed il destinatario e il suo server di posta vengono individuati grazie al DNS.

Tutti i messaggi di posta vengono fatti da righe di caratteri ASCII di lunghezza eventualmente limitata, quindi le immagini vengono codificate in sequenze di caratteri ASCII, con aumento di peso (es. base64) ma senza perdita di qualità. Ogni messaggio ha headers e corpo, secondo lo standard **MIME** (**Multipurpose Internet Mail Extensions**) che prevede degli header addizionali e permette il riconoscimento degli oggetti nel messaggio. Il testo del corpo può contenere anche HTML, quindi collegamenti e vari media.

Il download delle email può avvenire tramite HTTP, usando un webserver ed una webmail: in tal caso i messaggi rimangono sempre sul server, ma in genere non li si può consultare offline. Altri protocolli, come POP[71], sono più adatti ad utenti con un singolo PC e consentono la consultazione offline. POP usa status codes semplici, preceduti da "+" per indicare il successo, "-" per indicare il fallimento ed in caso vari messaggi di aiuto (alla pari di SMTP).

71 POP3: 110/TCP, text-based.

POP è sempre autenticato ed ha due fasi: autenticazione e fruizione, che consente il listing con "LIST", il download con "RETR" e la cancellazione[72] con "DELE", seguiti infine dal classico "QUIT".

IMAP[73] è simile a POP, ma unisce il meglio delle webmail e di POP. Con questo protocollo i messaggi rimangono sul server, organizzati in cartelle remote gerarchiche, di volta in volta sincronizzate con tutte le stazioni dell'utente, che può quindi consultare i messaggi già scaricati anche offline.

72 Attenzione: non è detto che i server cancellino davvero i messaggi che scompaiono dalla lista...
73 IMAP: 143/TCP, text-based.

20 • World Wide Web

Il web si chiama così perché è composto da una ragnatela di pagine web su scala mondiale, collegata tramite link comodi da usare e che rendono semplice il reperimento delle risorse anche agli incompetenti.

Le pagine web sono scritte in HTML ed identificate da **URI** (**Universal Resource Identifiers**). Ogni pagina web è multimediale e composta da vari oggetti strutturati tramite l'**HTML** (**Hyper-Text Markup Language**), che rende alcuni oggetti delle pagine "sensibili" o attivabili per compiere azioni predeterminate: il browser web scarica quindi il testo, la struttura ed i riferimenti agli oggetti esterni, per poi farne il rendering ed attendere l'input dell'utente.

Ogni risorsa web è identificata da URI o **URL** (**Universal Resource Locator**[74]) perché le risorse devono sempre essere localizzate per essere scaricate.

Ogni URI è formato in ordine da:

1. **Schema**:
   determina protocollo e formato dell'intero URI;
2. **Hostname**:
   server da contattare;
3. **Porta di rete del server**:
   se omessa, il navigatore usa la porta 80 (HTTP), 443 (HTTPS) o comunque la porta più consona per il protocollo dello schema;
4. **Path**:
   percorso all'interno del server, usata per distinguere la risorsa tipicamente - ma non necessariamente - localizzata nel filesystem remoto;

74 *Resource Locator*: nel corso si considera equivalente al Resource Identifier, per semplicità.

5. **Oggetto**:
   nome della risorsa[75], eventualmente il nome di un file;
6. **Ancora**:
   permette di ottenere più URL diversi per una risorsa, che puntano[76] (se la risorsa è markup) ad una posizione (ancora) omonima nella risorsa stessa.

Il client ha valori di default per visualizzare tutti gli elementi della pagina HTML.

Fare riferimento al W3C per le specifiche aggiornate.

I browser web possono visualizzare oggetti di vario tipo (immagini, pubblicità[77], video, suoni...) e possono usare vari protocolli, come SIP (**S**ession **I**nitiation **P**rotocol, inizia telefonate o chats), FTP[78], etc., oltre a poter essere esteso tramite plugins o estensioni. Il browser piace perché è intuitivo, colorato dalle pagine web e molto versatile.

20.1 • HTTP

HTTP[79] è un protocollo versatile e text-based, che segue strettamente il modello client-server e sfrutta al meglio TCP. HTTP segue il pattern request-response, che vede il client inviare la richiesta dopo aver aperto una nuova connessione TCP ed il server fornire la risorsa richiesta sulla medesima connessione. HTTP è **stateless**, perché il server non tiene traccia delle interazioni dell'utente, per cui servono altri meccanismi lato client per mantenere lo stato.

Ciascuna richiesta HTTP è formata da una **linea di richiesta o di stato**, vari campi di intestazione (ciascuno su una linea

75 Il server invia pagine di default quando in un link non è specificata la risorsa da visualizzare.
76 I navigatori web sfruttano il nome d'ancora passato nell'URL per spostare il viewport in un punto preciso della pagina web, scrollando quando necessario.
77 Il web è reso popolare dalla possibilità di inserire annunci pubblicitari, che ripagano il loro costo di mantenimento.
78 FTPS (FTP over SSL) è FTP crittografato; SFTP (Secure FTP) è FTP tramite SSH (Shell Sicura, crittografato). Il classico FTP in chiaro **non** è adatto all'uso su Internet perché è vulnerabile allo sniffing.
79 HTTP: 80/TCP (statica), 8080/TCP (utente/cache/proxy), .../TCP > 1024.

indipendente) con il formato "nome:valore" indipendente dalla posizione[80], una linea lasciata volontariamente vuota che contiene CRLF[81] e poi il corpo che contiene la risorsa voluta dal client. La request line contiene il metodo HTTP[82], seguito dall'URI e dalla versione del protocollo[83] in uso.

HTTP/1.1 consente di usare una singola **connessione persistente** per scaricare più risorse e diminuire così l'overhead, oltre ad altre finezze. L'URI della request line contiene solo elementi non deducibili, quindi tralascia l'host, che viene specificato su una riga separata in modo da consentire a ciascun server di ospitare più siti differenti e far quindi capo a più hostname DNS distinti.

Fare riferimento al W3C per le specifiche aggiornate.

La status line delle risposte inviate dal server è formata dalla versione di HTTP, seguita dallo status code (200 per il successo, etc.), che può a sua volta essere seguito da una breve spiegazione sulla stessa linea. Il server ha cura di includere un header "**Content-Type**" che specifica il MIME da usare per capire il tipo della risorsa a prescindere dall'oggetto visibile nell'URL: il server può sempre restituire un contenuto arbitrario, a prescindere da ciò che l'utente si aspetta dall'oggetto mostrato nell'URL, se presente.

In generale gli stati sono così classificati:

- 2XX:

  “successo, ecco la risorsa”;

- 3XX:

  “la risorsa è altrove, segui questa redirezione solo per adesso/per sempre...”;

80 I protocolli inferiori usano la posizione dei dati per risparmiare spazio, mentre HTTP funziona indipendentemente dalla posizione (purché non sia ancora iniziato il payload della richiesta) ma spreca molta più banda.
81 ASCII Carriage Return + ASCII Line Feed.
82 Metodi HTTP standard: GET, POST, PUT, HEAD, ...
83 Formato standard: protocollo/versione, tipicamente HTTP/1.1.

- 4XX:
  errore nelle richiesta del client/navigatore o richiesta di autenticazione (“ti sei dimenticato che servono le credenziali?”);
- 5XX:
  “errore interno del server, riprova più tardi”.

HTTP prevede anche l'autenticazione tramite il codice 401. Il server include un header "**WWW-Authenticate**", cui fa seguire un **codice challenge**. Il client memorizza il challenge e risponde con una richiesta GET, dotata di header "**Authorization**" che ha il valore delle credenziali immesse dall'utente, eventualmente **crittografate** con il codice challenge già in memoria.
Da questo momento, se l'autenticazione ha successo, il client invia sempre le credenziali con ogni richiesta a quel particolare server. **HTTPS** implementa forme di sicurezza ed autenticazione superiori e più adeguate ai tempi...

I **cookie** servono per compensare il fatto che HTTP è stateless e consentire quindi al server di identificare un client servito in precedenza, così da poter fare interazioni **statefull** senza il bisogno di autenticazione. Questo meccanismo (che richiede la massima attenzione per essere messo in sicurezza) è progettato per facilitare lo shopping on-line e ricordare le preferenze dell'utente. Per motivi di privacy, i moderni browser consentono di limitare e/o disattivare i cookie in base al dominio.

Il browser web fa caching delle risorse, la cui copia viene fatta scadere dopo un controllo tramite richieste HEAD o tramite l'intestazione "If-modified-since", il cui valore è la data dell'oggetto in cache, a cui può seguire risposta con stato "304 Not Modified" o direttamente con la risorsa più aggiornata, che rinfrescherà la cache. I proxy web fanno caching per più utenti, migliorando la performance della navigazione, alla pari dei CDN.

21 • Assegnazione degli indirizzi ed indirizzi privati

Idealmente ogni stazione dovrebbe avere un IP unico assegnato da IANA. Questo è impossibile per ragioni logistiche, quindi IANA delega l'assegnazione a dei registratori ("**registers**") come RIPE NCC, ARIN, LACNIC, APNIC ed AfriNIC. Ciascuno di essi rappresenta un continente e delega a sua volta l'assegnazione agli ISP. Nelle aziende ci sono poi dipartimenti informatici e centri appositi che regolamentano l'uso degli indirizzi per evitarne l'abuso.

Gli **indirizzi privati** non sono unici ma evitano di dover fare richieste per ottenere nuovi IP: si usano solo in ambiti ristretti per collegare i dispositivi, creando la rete locale. Siccome i router usano sempre il percorso più breve, gli indirizzi privati causano problemi se usati in Internet proprio perché più stazioni in contemporanea li possono utilizzare senza preavviso:

- 10.0.0.0/8 → prefisso di classe A;
- Da 172.16.0.0/16 fino a 172.31.0.0/16 → 16 prefissi di classe B;
- Da 192.168.0.0/24 fino a 192.168.255.0/25 → 256 prefissi di classe C.

Tutti i prefissi qui sopra sono pre-autorizzati dalla IANA perché non sono pensati per essere unici. Al contrario dare indirizzi qualsiasi alle stazioni LAN è inopportuno, perché se la stazione cerca di contattare l'host di Internet con lo stesso IP, questi non riceverà i pacchetti perché la sua distanza in hop è ovviamente maggiore.

Una rete Intranet è una comune rete aziendale o universitaria basata su TCP/IP, che contiene host con IP privati ed alcuni host con IP pubblici. La parte pubblica dell'Intranet collega la sua parte privata ad Internet ed è più piccola di quest'ultima: un host della intranet privata può comunicare con Internet ma non può ricevere risposta, perché ci sarà quasi sempre un altro host con lo stesso IP privato ma più vicino al server Internet contattato. Per ovviare e consentire collegamenti esterni alle stazioni con

indirizzi privati (cosa che ha prolungato di 20 anni la "vita" di IPv4!) si attuano diversi stratagemmi:

- **Proxy**
  usare un PC della rete come intermediario;
- **NAT**
  scambiare temporaneamente IP con un IP pubblico;

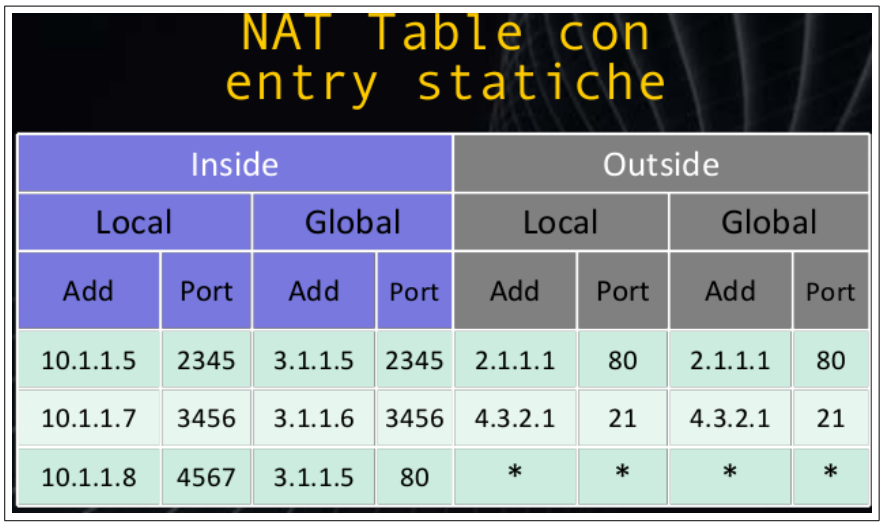
Il proxy è specializzato in una specifica applicazione/protocollo e viene chiamato anche "**proxy applicativo**". Il destinatario comunica solo con il proxy tramite un protocollo noto a priori (es. HTTP), che rimbalza la comunicazione alla stazione in modo non trasparente ed altamente specializzato: per usare un proxy, la stazione (e spesso la singola applicazione) deve essere sempre configurata esplicitamente.

NAT (**Network Address Translation**) risolve il problema del cambio di IP modificando i pacchetti che passano per il gateway Internet, tipicamente configurato come default gateway delle stazioni Intranet. Il router NAT sostituirà dinamicamente l'IP privato con il proprio IP pubblico in ciascun pacchetto. Se il router ha più IP pubblici (si dice "**pool di IP** pubblici"), ne sceglie di volta in volta uno a caso e lo sostituisce a quello privato. Il pacchetto arriva al server Internet con l'IP pubblico assegnato alla rete di partenza e non con quello privato della stazione Intranet; la risposta arriva al router, che si ricorda della sostituzione e la inverte, imponendo l'IP privato ed inoltrando il pacchetto in LAN. L'effetto, dal punto di vista del destinatario, è quello di un proxy; dal punto di vista del mittente, è quello della comunicazione diretta! NAT è quindi trasparente e funziona a prescindere dall'applicazione che invia e riceve i dati.

NAT si usa anche quando ci sono spazi di indirizzamento privati sovrapposti (IP privati uguali in due intranet con lo stesso prefisso, situazione tipicamente causata dalle fusioni aziendali) o quando ci si trova in una Extranet (insieme di reti private collegate). NAT è quindi utile per far comunicare enti distinti o fare **address expansion**: più indirizzi locali/privati vengono

soppiantati da pochi indirizzi globali/pubblici. Questo in particolare è il fenomeno che ha salvato Internet.

NAT è complesso e si basa sulla **NAT Mapping Table**, che associa stazioni esterne a stazioni interne. Per ciascuna stazione interna/esterna la tabella memorizza l'indirizzo locale e quello globale, oltre alla porta di comunicazione (es. 8080) associata a ciascun IP. Questo permette tra l'altro l'uso di indirizzi fittizi tra reti locali, risolvendo il problema delle Extranet.



La NAT Mapping Table è enorme e complessa, quindi la sua gestione è critica a livello prestazionale, dato che deve essere enumerata per ogni pacchetto in transito.

Quando il router NAT si trova in difficoltà (es. per corrispondenze ambigue), spesso altera la NAT Mapping Table sostituendo la porta del mittente con un valore casuale: questa **PAT** (**Port Address Translation**, detta anche **Address Overload**) previene pro-attivamente le ambiguità sostituendo il numero di porta del mittente in uscita, in modo da poter poi discernere i pacchetti in entrata (le risposte da Internet).

NAT può anche essere statico se la LAN contiene dei server, inserendo manualmente righe nella tabella e contrassegnandole come inalterabili: in tal caso la porta esterna della riga è piena di caratteri jolly e quella interna è fissa.

22 • Configurazione delle stazioni (DHCP e RARP)

La configurazione di ogni stazione della rete deve contenere almeno le seguenti informazioni:

- **Default gateway**:
  la stazione ne usa sempre e solo uno per volta (con fallback al successivo quando la risoluzione ARP fallisce);
- DNS server:
  un DNS principale ed altri DNS come fallback (Linux ne supporta 3 in totale);
- Indirizzo IP:
  indispensabile;
- Netmask:
  indispensabile.

...ed opzionalmente:

- **Nome**:
  le stazioni non hanno bisogno di conoscere il proprio, ma possono raggiungere le altre tramite il loro nome;
- **Dominio di default**:
  usato per completare automaticamente i nomi che non sono FQDN;
- *Server WINS* (Windows Internet Naming Service)[84]
  per scoprire i servizi Windows.

Tutti i SO devono fornire gli strumenti per configurare questi parametri di rete, eventualmente tramite una GUI[85].

Questo tipo di configurazione statica è però inadatta alle stazioni mobili e quindi il SO permette di delegare la configurazione al

84 WINS è per i nomi NetBIOS quello che il DNS è per i nomi di dominio, ma i mapping WINS sono aggiornati dinamicamente e le query dipendono da TCP/IP. Si sconsiglia fortemente l'abilitazione di NetBIOS.

85 La GUI le nomina in modo diverso, quindi bisogna documentarsi di volta in volta.

complesso[86] protocollo **DHCP** (**Dynamic Host Configuration Protocol**).

22.1 • DHCP

DHCP funziona tramite un server DHCP contenete un DB di configurazioni: il client invia delle richieste al server e questi offre una configurazione rispondendo alle richieste in broadcast[87].

A causa dell'imbustamento IP, anche il client deve inviare i messaggi in broadcast[88] IP e MAC, usando però lo speciale IP sorgente 0.0.0.0 e l'IP di destinazione broadcast 255.255.255.255.

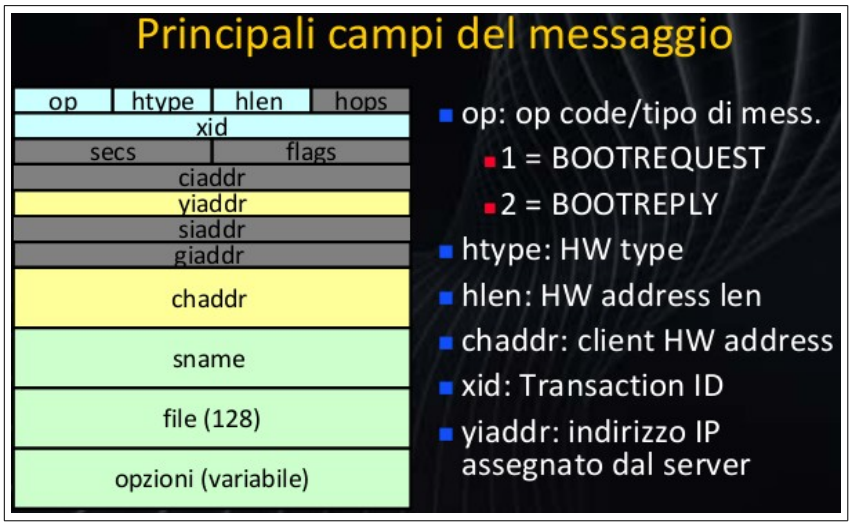
Il server DHCP può associare gli IP alle stazioni con le tecniche *esclusive* di:

- **associazione dinamica**
  prende un qualsiasi IP inutilizzato dal pool e lo riassegna liberamente senza tenere conto dell'identità o della storia della stazione client;
- **allocazione automatica**
  identica a quella dinamica, ma cerca di dare lo stesso indirizzo allo stesso MAC, identificando così nel tempo la stazione;
- **allocazione manuale**
  imposizione di un IP ad un certo MAC secondo la decisione esplicita (IP per IP, MAC per MAC!) dell'amministratore di rete;
- **configurazione manuale**
  i client usano configurazioni arbitrarie, imponendosi.

86 Relativamente complesso, perché l'alternativa a DHCP è data dal vecchio metodo della configurazione RARP+ICMP, che comporta richieste multiple ed indipendenti (gateway discovery, netmask discovery, IP discovery).

87 Broadcast UDP sulla porta 57.

88 Lo fa proprio perché non conosce l'interlocutore ed anche perché non è detto che ci sia un solo server DHCP nella LAN.



Ciascun messaggio DHCP estende il vecchio protocollo **BOOTP** e ne eredita la **terminologia di rete**[89]; inoltre contiene il tipo, la lunghezza ed i byte dell'indirizzo hardware (es. su Ethernet il MAC address), il codice unico della transazione DHCP e l'IP da assegnare.

Seguono campi opzionali in **stile code-length-value** che permette di ignorare facilmente i singoli campi ed estendere il protocollo nel tempo. Un campo opzionale può contenere la netmask, il DNS, il default gateway etc... ma il campo opzionale più importante è quello con codice 53, che specifica il tipo di messaggio DHCP, come ad esempio "DHCPdiscover" per scoprire i server DHCP nella LAN o "DHCPoffer" per offrire alla stazione client una configurazione.

La risposta del server è in broadcast, alla pari della richiesta: il client deve rispondere subito con "DHCPrequest" per accettare (se lo desidera) la configurazione di rete proposta, di fatto prenotandola. Ogni configurazione, però, ha vita limitata e deve essere rinnovata prima che il valore di "**lease**" scada, anche se - nonostante gli sforzi del client per non far scadere la configurazione - il server può offrire in ogni momento un nuovo IP in risposta. Quando le stazioni si riavviano la loro configurazione non scade esplicitamente e, a seconda del SO, hanno la possibilità di rinnovare la configurazione che avevano memorizzato su disco quando erano accese.

89 Ad esempio DHCP contiene il campo "op", che può essere "bootp request" o "bootp reply".

22.2 • Dettagli sulla configurazione automatica

Se non c’è un server DHCP disponibile, le stazioni usano indirizzi riservati come 169.254.0.0/16 o IPv6 link-local. Sarà poi il SO a generare da se la host part, usando il MAC dell'interfaccia come **seme dell'algoritmo** ed ARP per verificare che l'IP scelto non sia duplicato nella LAN. Lo "studio del dentista", privo di server DHCP e con poche stazioni, usa la configurazione automatica che non richiede competenze.

22.3 • DHCP Relay

DHCP Relay è una funzionalità dei moderni router che rende possibile l'inoltro dei pacchetti DHCP da una rete fisica all'altra: in questo modo anche le reti prive di DHCP servers possono essere configurate senza aggiungere nuovi server DHCP locali. Con Relay la richiesta DHCP viene arricchita con l'IP del DHCP Relay router, così il server DHCP ricevente può discriminare le stazioni in base alla LIS e fornire una risposta diversa per preservare il pool di IP della LIS in cui si trova. DHCP Relay è trasparente per il client, ma i SO più vecchi devono aggiornare da sé i record PTR degli eventuali DDNS, mentre i SO più nuovi lasciano che sia il server DHCP remoto ad aggiornare il DDNS server con i PTR corretti.

23 • Algoritmi di routing

Il **routing** consiste nel determinare il percorso che i pacchetti seguiranno nella rete, mentre il **forwarding** opera una decisione di routing per far avanzare i pacchetti nella rete.

Il *routing proattivo*[90] cerca di predeterminare il percorso che il pacchetto deve seguire per arrivare a destinazione ancor prima che questi sia arrivato al router, in modo indipendente dal traffico e con la necessità di determinare in anticipo le stazioni raggiungibili[91].

Il *routing al volo* crea la routing table in base ai singoli pacchetti ed alle informazioni del routing proattivo o alle segnalazioni di rete. Si può utilizzare tramite diverse tecniche[92]:

- **Routing by network address**
  si basa sull'IP indicato nel pacchetto in transito;
- **Label swapping**
  usa un'etichetta speciale in ogni pacchetto, che viene via via modificata dai router durante il transito;
- **Source routing**
  usa esclusivamente la configurazione della stazione;
- *Flooding*
  la tecnica di base.

Ciascun router, una volta deciso il **Next Hop**, sposta (in gergo "fa **switching**") il pacchetto fisicamente dalla porta di entrata alla nuova porta di uscita.

Il **Fixed Directory Routing** o **routing statico** è il routing proattivo simulato dall'amministratore di rete tramite configurazione manuale. Con questo tipo di configurazione

90 Questo termine è utilizzato dal docente ma nella pratica (purtroppo) questa distinzione viene trascurata.
91 Normalmente il routing proattivo viene chiamato solo routing.
92 I router utilizzano sempre una combinazione di queste tecniche!

l'admin può facilmente commettere errori e la rete non ha tolleranza ai guasti, salvo che questi abbia cura di configurare delle **route di backup** e dei router per il *load balancing*. Non è comunque un tipo di configurazione adatta alle grosse reti perché può creare un pericoloso **loop di routing**, capace di bloccare la rete in caso di guasti. Proprio a causa di questi loop il routing statico si usa solo nelle zone periferiche della rete, ove i router non hanno ridondanza.

Il **routing centralizzato** si basa su un **RCC** (**Routing Control Center**), centro di calcolo che fa routing proattivo per tutta la rete. Questo facilita la ricerca di guasti ed il debugging ma causa molto **traffico di controllo** attorno al RCC che a sua volta causa molto **traffico di routing**. Inoltre l'RCC è un collo di bottiglia ed un **single point of failure** per l'intera rete, quindi si adatta bene solo alle reti che cambiano di rado topologia.

Con il **routing isolato** ogni nodo fa routing proattivo in modo indipendente e non crea traffico di routing sfruttando il **backward learning** di 802.11 in modo eccellente.

Il **routing distribuito** si attua come il **routing isolato** ma i router decidono in modo coerente il percorso dei pacchetti scambiando in tempo reale informazioni con gli altri router. Questo tipo di routing è migliore di quello isolato, ma necessita di **algoritmi di learning** come l'**algoritmo Distance Vector**: ogni router genera una lista delle destinazioni da lui raggiungibili e la scambia con altre liste in entrata da altri router sulla rete, creando una lista detta "**distance vector**" che contiene tutte le destinazioni raggiungibili e la loro distanza posizionale in hop. Ogni router, per poter usare il distance vector, deve memorizzare la lista di *tutte* le possibili destinazioni provocando un notevole impatto prestazionale direttamente proporzionale alle dimensioni della rete.

I distance vector vengono uniti e filtrati da chi li riceve, estrapolando i percorsi più economici o - se ci sono dubbi - un percorso a caso qualora il costo sia equivalente. I router inviano questa lista solo quando essa varia in seguito ad un cambiamento

topologico ed in tal caso gli altri router devono scartare il lavoro di fusione e filtraggio già svolto e ricostruire il distance vector da zero[93].

Quando la topologia è instabile questo algoritmo può creare dei **black hole di rete**, dei **loop di rete**, conteggi infiniti etc.
Il problema si risolve solo parzialmente, con modifiche complesse e poco efficaci dell'algoritmo originale: l'unica soluzione efficace è far conoscere ai router la topologia della rete come grafo completo!

L'**algoritmo Link State** risolve i problemi del Distance Vector cercando di costruire un grafo completo della rete. Link State fa inviare ai router la topologia *fisica* della rete che li circonda creando così una **mappa distribuita** dell'intera LIS. Questo richiede l'implementazione del **selective flooding** per evitare bug e congestione di rete. Alla fine ogni nodo ha la stessa mappa della rete degli altri e calcola la route migliore con l'**algoritmo Shortest Path First** di *Dijkastra*. Questo algoritmo evita i loop, se e solo se tutti i router hanno la stessa mappa della rete, altrimenti causa il blocco dell'intera rete. L'algoritmo Link State è quindi molto complesso ed ha un ritardo iniziale elevato, ma quando può essere utilizzato è ottimale in termini di complessità:

```
Numero link * log(Numero nodi)
```

...quindi è veloce a calcolare i percorsi, è scalabile e richiede poca potenza di calcolo per creare i link states che sono piccoli e rimangono immutabili. La ricerca di guasti è facilitata dal fatto che tutti i nodi condividono lo stesso stato. Il downside è che Link State è difficile da configurare e da implementare[94].

93 Il Distance Vector limita la velocità del routing a quella del router più lento ed ha complessità quadratica, quindi in realtà viene usato solo su reti semplici e piccole.

94 La prima versione dell'algoritmo ha richiesto ben 5 anni per essere ultimata!

24 • Architettura e protocolli di routing su Internet

24.1 • Protocolli di routing

I protocolli di routing fanno routing proattivo, ponendosi come base di supporto del livello 3, ma potendo essere imbustati anche nel livello 4; tipicamente vengono utilizzati nei router e si ispirano ad un algoritmo di routing ben noto.

L'algoritmo di Dijkastra necessita di una metrica ma non ne specifica l'unità, quindi il protocollo di routing può usare diversi parametri (come il numero di hop) per ottenerne una e codificarla arbitrariamente nei pacchetti.

Un **dominio di routing** è un percorso o una rete connessa alle altre che utilizza specifico protocollo di routing in modo uniforme, affinché ciascun router che appartiene a più domini possa utilizzare in contemporanea più protocolli di routing. Siccome i protocolli di rete utilizzano metriche differenti, l'amministratore deve configurare con cura le capacità minime per ogni protocollo e la politica per l'annunciazione. Il router deve calcolare la tabella di routing discriminando, tramite la configurazione dell'amministratore di rete, il protocollo a cui affidarsi caso per caso.

24.2 • Autonomous Systems

Un **Autonomous System** (**AS**, sistema autonomo) è una porzione di rete composta da sottoreti topologicamente vicine e gestite tipicamente da un singolo ISP: essendoci più domini di routing, le sottoreti devono essere progettate e configurate con coerenza da un singolo fornitore di servizi.

Gli AS sono collegati da nodi che non inoltrano tutti i dati ma, al contrario, agiscono da filtro: il flusso di dati è strettamente controllato anche per ragioni commerciali e di sicurezza. Le scelte del routing tra AS (detto "**routing esterno**") devono essere

concordate tra gli ISP, mentre quelle del routing interno appartengono al singolo ISP in totale autonomia. Per ridurre il traffico tra gli AS spesso gli annunci vengono aggregati e spesso compattati rimuovendo le informazioni superflue per i router geograficamente lontani.

Gli AS sono resi indipendenti tramite l'uso di:

- **Gateways Interni**
  sono router che utilizzano il protocollo **IGP** (**Internal Gateway Protocol**), posti dentro la rete;
- speciali **Exterior/Border Gateways** (detti anche "**Boundary Router**")
  sono router che utilizzano il protocollo **EGP** (**Exterior Gateway Protocol**) e che vengono posti nelle zone limitrofe tra gli AS di ISP differenti.

Ogni AS è identificato globalmente da un **numero** di IANA lungo 4 B e privatamente da numeri della singola rete ISP. I numeri vengono usati per determinare il percorso degli annunci tra collegamenti con EGP.

Ciascun gestore di AS può configurare solo i propri AS e quindi sta accorto ad inoltrare gli annunci solo su determinati collegamenti e non su altri, così i router vedono solo determinati percorsi possibili per i dati.

Il routing EGP è quindi basato su politiche configurabili che riflettono gli accordi commerciali tra gli ISP degli AS comunicanti: due AS collegati non sempre sono autorizzati a comunicare. I router ai bordi degli AS usano il **supernetting** per aggregare gli annunci, celando le parti superflue delle subnet agli AS esterni che saranno "felici" di elaborare meno annunci...

La rete Internet è sostanzialmente fatta da ISP, ciascuno con almeno un AS. Gli AS di ogni ISP sono interconnessi tra di loro e con gli ISP più piccoli usando Border Gateways. I provider con "**tier**" minore sono più grossi di quelli con "tier" maggiore. I

provider dello stesso livello/tier usano collegamenti di **private peering** per scambiarsi dati per vie più brevi del normale a favore dei clienti; i provider più grandi forniscono dietro pagamento la loro connettività a quelli più piccoli, tramite collegamenti **client-provider** o "di transito", che però sono vincolati da un contratto (e quindi da accordi commerciali) a comunicare solo verso destinazioni concordate (es. America ← Europa ma non America ← Africa).

Gli ISP che fanno private peering spesso creano o affittano link privati in fibra, per non dover utilizzare centri di smistamento MAP e **IXP** o comunque infrastrutture neutrali di enti terzi che sono spesso semplici stanze comuni con router condivisi.

> *Semplicemente, un IXP è una stanza piena di router, almeno uno per ogni ISP. Una rete LAN, all'interno della stanza, connette tutti i router in modo che i pacchetti possano essere inoltrati dalla dorsale di un ISP a quella di un altro. Gli IXP possono essere grandi centri gestiti da terze parti. Uno dei più grandi IXP è l'Amsterdam Internet Exchange, a cui sono connessi centinaia di ISP che si scambiano centinaia di gigabit di traffico ogni secondo.*[95]

95 Cit. da *Fondamenti di Reti di Calcolatori*, cap. 1.5.1, §13.

25 • Protocolli di routing e servizi di consegna speciali

I protocolli IGP si utilizzano solo per il routing intradominio, ovvero all'interno dell'AS; quelli BGP si usano per il routing interdominio, ovvero tra AS diversi.

IGP distribuisce informazioni sulla topologia della rete e cerca la route "migliore", qualsiasi cosa voglia dire: il protocollo raccoglie informazioni e calcola il percorso migliore per ogni destinazione possibile. EGP, al contrario, distribuisce dei "costi amministrativi" senza cercare la strada "migliore" ma solo quella preferita dagli admin per gli accordi commerciali in essere.

Tra gli algoritmi di routing basati sul Distance Vector figurano:

- per IGP:
  - **RIP**[96] (**Routing Information Protocol**)
    può calcolare distance vectors di al massimo 15 hop ed ogni 30 secondi li deve distribuire. Ha scarsa efficienza ed un tempo di convergenza di 3 minuti: la rete RIP impiega ben 3 minuti per cambiare topologia e nel frattempo è inutilizzabile;
  - **IGRP**[97] (**Interior Gateway Routing Protocol**)
    usa metriche complesse (banda, ritardo, carico di rete, affidabilità ed MTU) e scala meglio di RIP sulle grandi reti, perché la scelta del percorso "migliore" viene fatta in modo più oculato ed il protocollo fornisce valori di default accettabili. IGRP consente inoltre di fare **multi-path routing per distribuire il carico** su tutti i percorsi in ordine di costo;
  - **E-IGRP** (**Enhanced-IGRP**)
    versione potenziata di IGRP.
- basati su Link State:

96 Protocollo sviluppato per altre architetture e poi adattato ad Internet con la vecchissima RFC 1058 (del 1988!). Fu implementato in UNIX e Linux, perché questi sistemi furono concepiti per apprendere altri gateway oltre a quello di default ed in caso fare da router.

97 Protocollo proprietario ed utilizzabile solo su apparati di Cisco Systems. Spesso è l'unica alternativa a RIP.

- **OSPF**[98] (**Open Shortest Path First**)

  divide la rete in aree gerarchiche (la cui radice è l'"area 0" o "**area di backbone**") e tra di esse trasmette solo le informazioni indispensabili. Necessita di più tipi di router, come l'**Internal Router** e l'**Area Border Router**, quest'ultimo posto al limite dell'area 0 per poter riassumere gli annunci delle aree inferiori, in modo che ogni router abbia a che fare solo con le informazioni pertinenti all'area che gestisce;

- **Integrated IS-IS**[99]

  consente il routing standardizzato (ISO/IEC 10589:2002) e gerarchico secondo il modello OSI. È lo *standard de facto* per la *backbone* degli ISP e funziona in modo simile ad OSPF grazie all'algoritmo di Dijkstra;

- per EGP:

  - **BGP**[100] (**Border Gateway Protocol**)

    basato su **Path Vector**, lista composta dalla sequenza di AS dal router fino alla destinazione. Viene usato per imporre le politiche amministrative;

  - **IDRP**[101] (**Inter Domain Protocol**).

Il routing statico è l'alternativa migliore quando i protocolli EGP non sono soddisfacenti.

98 Protocollo standardizzato nella RFC 1247 del 1991, con ritardo rispetto agli altri e con una configurazione ostica. Non è molto usato.

99 Protocollo basato sull'omonimo protocollo "IS-IS" di OSI. Deriva da un adattamento da OSI ad IP, creato appositamente per non dipendere da protocolli proprietari. Scala così bene che tutt'oggi i grossi ISP lo utilizzano!

100 Protocollo standardizzato con la RFC 1654 del 1994 ed ormai giunto alla quarta versione.

101 Protocollo evolutosi dalla versione OSI di BGP e poi riadattato per IP, in modo inverso a come si è fatto per IS-IS. IDRP è così superiore a BGP che si riteneva diventasse parte di IPv6: BGP però è stato aggiornato ed IDRP ha trovato scarsa adozione...

25.1 • Content Delivery Networks

I **CDN** (**Content Delivery Networks**) sono utilizzatissimi per distribuire velocemente contenuti pesanti. I CDN sono fatti da cache o da speciali **Replica server** diffusi per il mondo e l'utente userà sempre il server a lui più vicino per scaricare il file, ottenendo una copia esatta del materiale a prescindere dal server replica. Questa scelta di routing è fatta tramite DNS (che tipicamente è già scelto in base alla vicinanza al client!) inserendo indirizzi specifici in ciascun DNS. Si possono anche riscrivere gli URL nelle pagine web per indirizzare il navigatore dall'utente al server Replica geograficamente più vicino.

Altra alternativa è l'uso di **Anycast** tramite indirizzi che si riferiscono a gruppi di router su Internet, che a loro volta sceglieranno ovviamente il percorso "migliore"... che in questo caso è il più vicino.

Ultima soluzione è il **multicasting IP** che usa indirizzi che identificano gruppi di stazioni (IP di classe D, IP che iniziano con 224... o 239...): il pacchetto arriva a tutti gli host del gruppo, ovunque siano nella rete Internet, delegando la consegna al livello MAC proprio perché a ciascun IP multicast corrisponde un MAC identico ai suoi 32 b meno significativi. I router inoltrano a tutte le stazioni e dopo solo le schede appositamente configurate riceveranno il pacchetto, altrimenti lo scarteranno a basso livello. I router usano **IGMP** (**Internet Group Management Protocol**) per "scoprire" il gruppo multicast: o il router lo richiede alle stazioni, o le stazioni si annunciano per iscriversi al gruppo multicast. I router diffondono poi le informazioni IGMP agli altri router e così via. Il multicast IP non è molto supportato e spesso viene disattivato sui router perché gli admin sono diffidenti e ritengono che esso violi alcune prassi comuni...

26 • Panoramica sulla sicurezza delle informazioni

La privacy è un diritto fondamentale: vogliamo che chi intercetta i pacchetti non sia in grado di comprendere ed alterare la comunicazione e che gli EP siano capaci di identificarsi in modo affidabile tramite autenticazione. Alla fine l'utente che ha iniziato la comunicazione non può ripudiare i dati inviati, rimanendo quindi sempre identificabile.

Questi problemi vengono risolti con tecniche di criptazione e protocolli crittografici: la versione codificata del dato non rivela l'informazione a chi non possiede i parametri segreti dell'algoritmo usato per la codifica, anche se questi e le sue tecniche sono pubblici e noti a tutti. Ai dati in chiaro si applica un algoritmo che usa anche un parametro segreto come INPUT, detto "**chiave**" e composto - preferibilmente - da una sequenza di bit robusta e varia, generando così la versione criptata del dato.

Diversi algoritmi, a parità di lunghezza della chiave, hanno diversa robustezza! Quando si usa una sola chiave per decriptare, questa è detta "chiave segreta" o "chiave condivisa" e si parla di **crittografia simmetrica**, che richiede la condivisione della chiave attraverso altri canali verso entrambe le parti: con la crittografia simmetrica i due enti devono avere già una relazione, essendosi in precedenza scambiati la chiave segreta.
La conseguenza è che quando la chiave condivisa/segreta viene rubata o compromessa, la comunicazione diviene impossibile perché insicura.

La **crittografia asimmetrica** risolve il problema facendo usare alle due parti lo stesso algoritmo (ad esempio RSA) ma due chiavi differenti. Viene detta anche **crittografia a chiave pubblica**, perché la chiave per criptare non si può utilizzare per decriptare ed in genere viene pubblicata, mentre quella privata viene tenuta segreta e può decriptare e generare nuove chiavi per criptare.

Le **smart card** contengono e proteggono una chiave privata e le operazioni per utilizzarne le funzionalità, grazie alla tutela di un processore isolato e schermato.

Le chiavi per criptare possono essere pubblicate su uno speciale repository che le associa a dei dati identificativi per ritrovarla o identificare meglio il proprietario. Le chiavi per decriptare, al contrario, devono essere custodite gelosamente!

Più le chiavi sono lunghe e più è difficile utilizzarle in termini di potenza di calcolo, quindi la crittografia simmetrica è più leggera. Nelle comunicazioni si utilizza la crittografia asimmetrica per scambiarsi una leggera chiave simmetrica e si utilizza quest'ultima per criptare e decriptare: se questa chiave viene violata la comunicazione non è totalmente compromessa e può continuare, perché pochi istanti dopo verrà di nuovo cambiata in modo automatico e periodico. Le aziende offrono premi in denaro per chi riesce a superare l'algoritmo di crittografia, in modo da poter perfezionare l'algoritmo stesso senza danneggiare nessuno...

Siccome le chiavi sono collegate, se la decrittazione ha successo si ha la garanzia che il mittente sia corretto e che il messaggio sia inalterato perché qualsiasi manomissione causa corruzione a catena facile da individuare.

La **firma digitale** si basa sul **message digest**, una sintesi del documento, che muta ad ogni minima variazione dello stesso. I **digest crittografici**, a partire da un messaggio e da una chiave pubblica, creano un **hash** unico ed utilizzabile nelle firme elettroniche. Il tal caso, a differenza della firma a penna, quella elettronica è falsificabile. Tipici hash crittografici per il digest sono **MD5** e **SHA1**.

Affinché la firma del messaggio sia affidabile la chiave pubblica del repository deve essere autenticata per evitare che chi verifica la firma utilizzi la chiave "di un omonimo". Il problema si risolve con i **certificati a chiave pubblica**, detti anche "**certificati digitali**" e formati da una chiave, dall'identità del suo possessore

e da una firma su entrambe che le rende intoccabili. Il certificato digitale garantisce l'identità della firma digitale con la sua chiave e rende il documento firmato non ripudiabile dal mittente. Per questo i certificati devono essere consegnati da una Certification Authority (**CA**, autorità per la certificazione) al possessore "di persona" o comunque solo dopo averne verificato l'identità "offline" tramite un documento di identità o una risorsa identificativa adatta allo scopo del certificato.

Il certificato, una volta firmato, è sostanzialmente incorruttibile e contiene anche una chiave pubblica affidabile per comunicare con il proprietario. La firma viene fatta personalmente dalla CA ed ha in genere validità legale.

La **Public Key Infrastructure** è composta da una gerarchia di CA, in modo da facilitare il rilascio dei certificati digitali: chi deve verificare la chiave però deve ottenere i certificati di tutti gli elementi della catena di certificazione fino alla radice, detta "**Root CA**". Le Root CA firmano i propri certificati che quindi non sono di per se verificabili senza averne già una copia considerata sicura. Il processo si conclude quando il certificato della root CA viene consegnato in modo sicuro da tutte le CA e lo si può verificare: per questo le CA generano coppie di chiavi e vi allegano il certificato della propria root CA facendo da garanti. L'ultima parola è però data ai programmatori: i browser ed i SO contengono, in speciali archivi, i certificati di tutte le root CA che i programmatori ritengono affidabili. Alla fine la sicurezza dell'infrastruttura di certificazione dipende dagli sviluppatori e di quanto ci fidiamo di loro (e la fiducia è accordata usando i loro software)!

27 • Sicurezza di rete (IPsec, SSL, ...)

27.1 • Panoramica

Lo **snooping** consiste nello spiare il traffico di rete che scorre sui cavi, nei router[102] o nell'etere.

La **perturbazione del servizio** consiste nell'interrompere il routing mandando messaggi di routing fasulli o facendo **DNS cache poisoning** (alterazione malevola della cache DNS in remoto).

I **DoS** (**Denial of Service**, case-sensitive) sono attacchi elementari che negano il servizio di rete inondando un host o una rete con molto traffico spazzatura.

Gli attaccanti usano **exploit** (bug o comunque debolezze abusate) per bloccare o violare un nodo di rete, in modo da compromettere l'intera rete tramite bugs, tipicamente presenti nelle funzionalità meno utilizzate e quindi meno testate dei software. Ad esempio l'attaccante può mandare frammenti IP sovrapposti (che contengono la stessa parte del pacchetto originale) per bloccare il SO, violando quindi lo stack di rete.

L'**address spoofing** consiste nell'appropriarsi di un identificatore di una stazione di rete per acquisirne l'identità e ricevere il suo traffico o inviare traffico a suo nome.

Virus, **trojan horse** e **worms** possono essere veicolo di **backdoors** che lasciano un accesso secondario aperto per l'attaccante nel SO infettato, che può essere utilizzato per unire l'host infetto ad una **botnet** o spiare le azioni dell'utente.

Spesso gli attaccanti riescono semplicemente ad indovinare o rubare (tramite **phishing**/truffa) le credenziali.

102 I router hanno un SO e si possono infettare e dirottare.

Sono anche comuni le semplici frodi via e-mail, quindi è bene educare l'utenza.

Molte tecniche di difesa si basano sulla crittografia, sull'identificazione e sull'autenticazione tramite protocolli crittografici che ci servono per stabilire gli algoritmi, cambiare i certificati, etc.

I **firewall** ("muri tagliafuoco") sono apparati o applicazioni che filtrano il traffico e bloccano gli attacchi di rete, i programmi malevoli (una variante del firewall è l'antivirus) ed alcune truffe comuni. Su ciascun host si possono anche realizzare meccanismi di verifica e firma del codice[103] per limitare i programmi che possono essere caricati in memoria ed eseguiti. Nella pratica un firewall è un filtro di pacchetti che si basa sui campi dei protocolli, funzionando normalmente come una whitelist e scartando i pacchetti imprevisti. In base al grado di specializzazione possono essere **stateless**, a livello flusso, **statefull** o **applicativi**.

Gli **IDS** (**Intrusion Detection Systems**) sono sistemi che, piazzati in vari punti strategici della rete, rilevano un gran numero di attacchi senza però bloccarli[104].

27.2 • IPsec

IPsec (IP security) fornisce tutto il necessario per autenticazione e crittografia, usando un paio di chiavi di sessione per ogni direzione: la prima per autenticare e la seconda per criptare, tutte cambiate periodicamente durante la comunicazione bidirezionale. IPsec cambia le chiavi usando lo schema **IKE** (**Internet Key Exchange**), che genera un **framework** dedito unicamente alla negoziazione sicura. IKE è un framework basato su **ISAKMP**[105] e su altri protocolli specializzati nello scambio sicuro di chiavi crittografiche.

103 La Trusted Computing Platform ne è un esempio.
104 I malware (ed in genere i comportamenti anomali delle applicazioni) sono anche rilevabili da IDS e firewall avanzati.
105 IKE fornisce molti protocolli fra cui ISAKMP.

Le chiavi possono essere basate sul segreto condiviso (quando ci sono amministratori in comune) o sulle chiavi asimmetriche, tramite il celebre **algoritmo di Diffie-Hellman**[106] o usando la crittografia a chiave pubblica.

La **Transmission Mode Encapsulation** è la cifratura delle informazioni di livello trasporto usando **pacchetti ESP** (**Encapsulation Security Payload**) per imbustare i pacchetti IP. L'intestazione ESP autentica il payload e lo cifra fino alla coda ESP della busta, ma volendo ESP si può utilizzare anche per autenticare sia il payload che la coda stessa!

L'**Authentication Header** è un'alternativa[107] ad ESP che autentica anche l'intestazione IP rendendo il pacchetto totalmente inalterabile senza però criptarlo. Lo svantaggio rispetto ad ESP, però, è che il **MAC** (**Message Authentication Code**) di Authentication Header è incompatibile con il routing IP e quindi non deve includere il TTL, il ToS etc. per poter essere inoltrato su Internet.

Questi protocolli di imbustamento sono incompatibili con il NAT che scarta questo tipo di pacchetti non appena arrivano. La soluzione è l'**IPsec Tunneling** che risolve il problema imbustando ulteriormente i pacchetti autenticati in una VPN: questa è la **Tunnel Mode Encapsulation**, compatibile con il NAT e capace di nascondere allo sniffer sia la destinazione che il mittente del pacchetto, a patto che la rete contenga un device apposito che la supporti.

106 Questo algoritmo permette di negoziare una chiave senza inviarla!
107 In realtà lo si può utilizzare assieme ad ESP, imbustandolo dopo l'header di autenticazione.

27.3 • SSL

SSL (**Secure Socket Layer**) si preoccupa di cifrare ed autenticare TCP ed UDP, facendo da **wrapper** sui socket. La sua versione standard si chiama **TLS** (**Transport Layer Security**) perché SSL è nata su Internet ed è solo uno *standard de facto*. I server che usano protocolli sicuri[108] in genere usano porte alternative, ma TLS al contrario consente il riutilizzo della stessa porta usata per il protocollo in chiaro, tramite negoziazione **STARTTLS**: server e client negoziano se utilizzare la modalità sicura oppure no; nel caso di SSL la comunicazione sicura è data per scontata e quindi richiede una porta dedicata.

108 Es. HTTP+SSL = HTTPS su porta 443.

28 • IPv6 e socket

IP versione 6 ci serve solo per avere uno spazio di indirizzi più grande: altri motivi sono futili perché IPv4 è stato via via aggiornato ed ha quasi tutte le features di IPv6. Tutte le caratteristiche di IPv6 sono ormai state portate ad IPv4, come IPsec, multicast, PnP e l'auto-configurazione.

Gli indirizzi IPv4 sono lunghi 32 b e la loro cardinalità è di 4 miliardi, fortemente ridotti dall'assegnazione gerarchica che ne riduce la quantità utilizzabile per le reti fisiche. L'**addressing efficiency** di IPv4 è data da:

> H = $\log_{10}$(numero di indirizzi usati) / (max. numero di bit dell'indirizzo)

Si è verificato che H varia tra 0.22 e 0.26: l'efficienza di IPv4 è pertanto solo del 22%!

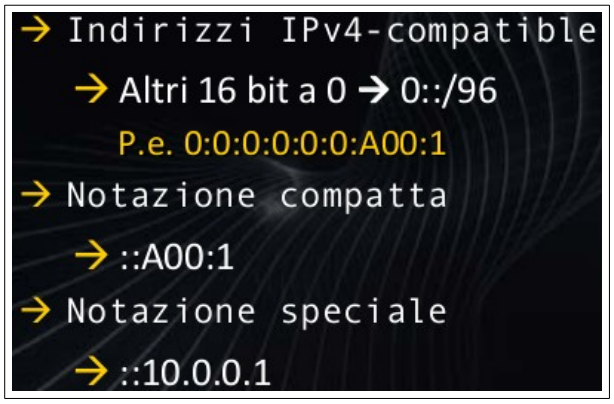
In origine IPv6 cercava di avere un milione di miliardo di stazioni, ma gli sviluppatori sono stati intimoriti dall'esaurimento degli IPv4 ed hanno deciso di utilizzare molti più bit del necessario: per garantire alle generazioni future il servizio Internet si è scelto di usare indirizzi IPv6 di 128 b (16 B), arrivando a ben 655.570.792.368.866.943.898.599 (655 miliardi di miliardi) possibili indirizzi IPv6 per m^2 di superficie terrestre!

Scriviamo gli indirizzi IPv6 in esadecimale per avere una rappresentazione compatta e separiamo gli 8 numeri esadecimali che li compongono con "`:`". Purtroppo sono gruppi di 2 B difficili da scrivere e da ricordare!
Per questo in IPv6 le operazioni di configurazione manuale sono ridotte al minimo...

Per comodità gli 0 iniziali ed i gruppi di 0 possono essere omessi, usando "`::`" per rappresentare un gruppo omesso.

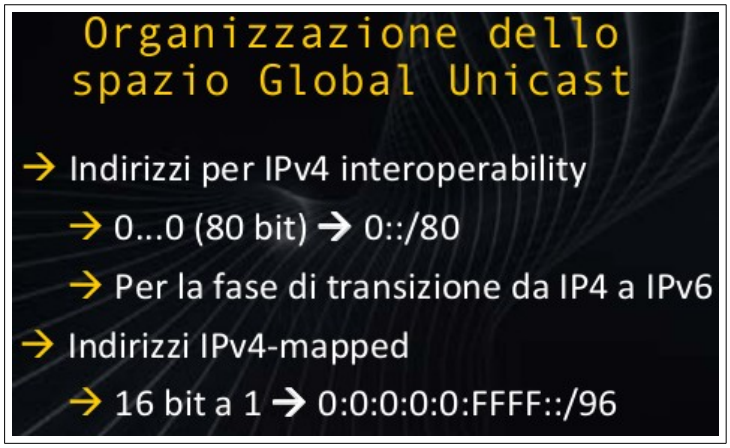
Gli indirizzi **IPv6 multicast** iniziano con "FF".

Gli indirizzi **IPv6 site-local** ed **IPv6 link-local** (ovvero quelli privati) iniziano per "FE80::/64" (link-local) o "FEC::/10" (site-local, equivalenti alla classe 10.0.0.0).



Gli **indirizzi IPv6 di compatibilità IPv4** iniziano con "0::/80" e quindi hanno 80 b disattivi.

Gli **indirizzi IPv6 mapped** hanno 64 b disattivi ed i restanti includono l'indirizzo IPv4; se i restanti bit sono disattivi e terminano con 4 B a 0 si dicono **indirizzi IPv6 compatible** e



supportano la comoda notazione speciale "::XX.XX.XX.XX" che include l'IPv4 in chiaro.

Gli **indirizzi IPv6 global unicast** vengono assegnati gradualmente e IANA tiene da parte molti spazi di indirizzi per cambiare metodo di assegnazione in futuro. La prassi proposta è stata quella di usare il metodo **Aggregatable Global Unicast**, che usa indirizzi "2..." ed aggrega più indirizzi in uno, per evitare che i router distanti debbano gestire troppi IP (aggregazione sotto

prefissi comuni a favore del routing). Questo si fa in IPv6 assegnando i prefissi non più in base alla geografia, ma in base al tier degli ISP, che nel tempo hanno già assunto una struttura gerarchica ma indipendente dal territorio.

IPv6 divide, come IPv4, la rete in reti e sottoreti collegate da routers. Tuttavia tutte le stazioni delle sottoreti IPv6 hanno in comune ciascuna un prefisso, che è l'unico indirizzo trattato dai router. Le LIS di IPv4 sono state sostituite dai **subnetworks** (prefisso in comune per tutte le stazioni) e le reti fisiche da **links** che devono corrispondere al subnetwork. Gli host **off link**[109] devono differire in prefisso e devono utilizzare un router. La coppia < IPv4, netmask > è sostituita da un prefisso `IPv6/N`, ove `N` è la lunghezza in bit del prefisso. Non ci sono classi di indirizzi IPv6 e quindi non si può dedurre la lunghezza del prefisso, che deve sempre essere esplicitato.

Gli IPv6 globali usano un prefisso di 64 b ed un indirizzo di 64 b, fatto per poter includere anche l'indirizzo Ethernet in chiaro della stazione. Il prefisso è composto[110] da 16 b di **Top Level Authority ID**, 32 b di **Next Level Authority** e 16 b di **Subnet Level Authority ID**. Così da ciascun IPv6 si può risalire alla gerarchia di assegnazione ma, quando un'azienda cambia ISP, si deve utilizzare la rinumerazione automatica IPv6 che riassegna all'intera azienda nuovi IP.

109 Le destinazioni *off link* sono quelle collegate indirettamente (tipicamente esterne alla LAN) e possono essere raggiunte solo usando il default gateway come nodo intermedio.

110 I criteri di assegnazione degli indirizzi IPv6 potrebbero cambiare in futuro: il lettore è invitato a segnalare eventuali imprecisioni, sviste e/o dati obsoleti.

IPv6 supporta sia la configurazione automatica senza DHCP (stateless) che quella con DHCP (statefull).

In IPv6, IP è stato pesantemente modificato alla pari di ICMP: ARP è adesso parte di ICMP, così come lo è IGMP. Sono stati aggiornati DNS (vd. record AAAA), RIP, OSPF, BGP, IDRP e perfino TCP ed UDP. TCP ed UDP mantengono lo stesso funzionamento ma cambiano l'interfaccia socket ed in generale l'implementazione.

28.1 • Socket

La tecnologia socket permette di accedere a TCP/IP dall'interno dei programmi ed è nata in UNIX per consentire l'accesso ai file usando appositi descrittori di file. Il socket è il punto di accesso alla rete ed è associato ad una connessione TCP o ad una sessione UDP. Con `listen()` si può attendere una connessione sulla LIS e con `connect()` la si può iniziare, specificando porta ed indirizzo di destinazione.

29 • IPv6 - approfondimento

I pacchetti IPv6 hanno meno campi di quelli IPv4, pur pesando il doppio (ben 40 B). Tutti i campi IPv6 sono di lunghezza fissa ed il pacchetto ha ethertype differente, per garantire che IPv4 possa coesistere con IPv6 sullo stesso host.

Il campo priority equivale al ToS di IPv4; il campo **flow label** risolve il problema dell'associazione dei pacchetti; *hop limit* equivale al TTL di IPv4.
IPv6 non comprende il *checksum*, che viene invece affidato direttamente al livello 2. Il campo della frammentazione è presente solo nel primo pacchetto di una serie di pacchetti IPv6, invece di essere ripetuto.

I **campi di estensione di IPv6** sono più facili da elaborare rispetto a quelli opzionali di IPv4 ed appaiono dopo il campo next header, come ad esempio l'estensione di routing, quella di autenticazione, l'**Encrypted Security Payload**, etc. Tutti gli header di estensione iniziano con il campo next header che permette di concatenarli o ignorarli, ed in caso di dati variabili, hanno anche un campo che contiene la lunghezza. L'ordine degli header IPv6 è un fattore prestazionale determinante e devono essere ben ordinati.

In IPv6, ARP viene sostituito da ICMP, che permette di scoprire l'indirizzo di livello 2 di tutti i vicini senza fare broadcast, ma utilizzando il multicast: con il Neighbour **Discovery** multicast poche stazioni elaborano l'ICMP discovery e solo se sottoscritte a quel **Solicited Node Multicast Address**.
A differenza di quanto accade con ARP, i nodi possono scartare i discovery che non li riguardano a basso livello.

Il multicast IPv6 è basato su trame MAC con indirizzo MAC multicast inglobato in un indirizzo IPv6 multicast. Per risolvere un indirizzo IPv6 la stazione invia un pacchetto **ICMP Neighbour Solicitation** al **Solicited Node Multicast Address** ricavato dall'indirizzo IPv6 da risolvere: questo messaggio viene ricevuto

dal gruppo di stazioni sottoscritto e solo la stazione con l'indirizzo IPv6 specificato risponde con una notifica che verrà memorizzata nella **host cache** (equivalente alla ARP cache) del mittente.

La transizione verso IPv6 deve essere graduale, facile e senza interruzioni del servizio: **IPv5** è stato dichiarato inadeguato perché era eccessivamente simile ad IPv4 ed avrebbe richiesto troppe transizioni future (con relative interruzioni del servizio). Lo stack di IPv6 è separato da quello di IPv4 per semplificare lo sviluppo e permette di trattarli come protocolli separati. Grazie a quest'accortezza dei progettisti possiamo imbustare pacchetti IPv6 in IPv4 e viceversa!

Tramite "isole" di **stazioni dual-stack** (collegate con tunnel IPv6→IPv4) la transizione avrà inizio e non causerà interruzioni, per poi espandere il trend ad isole di rete più grandi. Alla fine ci saranno più stazioni IPv6 che IPv4 ed Internet sarà convertita ad IPv6. I pochi host IPv4 rimanenti dovranno usare device per la traduzione IPv6→IPv4. Avremo interi ISP con connettività nativa IPv6 ed i clienti passeranno pian piano ad IPv6. Nel **giorno del giudizio** (**IPv6 doomsday**) avremo solo tunnel IPv4→IPv6 e tutti gli ISP offriranno solo IPv6. Siamo pronti alla transizione dal 1996, anno in cui i protocolli di IPv6 sono stati ufficialmente specificati. Quando gli indirizzi IPv4 finiranno[111] del tutto il passaggio sarà obbligato: i prefissi di classe A sono già terminati nel 2011...

111 Vd. statistiche su www.vyncke.org.

30 • Mobilità nelle reti IP

Il movimento delle stazioni dovrebbe essere trasparente rispetto ad IP e gestito a livello 2 senza essere notato dai livelli superiori. Tuttavia il prefisso IP corrisponde ad una LIS e quindi ad una rete fisica: la stazione dovrà obbligatoriamente cambiare IP!

Il cambio di indirizzo può essere problematico a causa della dipendenza di TCP e di altri protocolli dall'IP per identificare i pacchetti della sessione. Questo può causare il troncamento delle connessioni o comunque lo scadere delle sessioni al cambio di IP. Ulteriore svantaggio è che il cambio di IP è incompatibile con i meccanismi di sicurezza basati sulle IP whitelist.

La RFC 3344 del 2002 definisce le parti dello stack necessarie alla mobilità delle stazioni IP, anche qualora non sia già supportata. RFC 3344, oltre a prevedere sistemi anti-spoofing, è trasparente al livello applicativo e di trasporto. Qualunque sia l'indirizzo della stazione e la sua posizione nel mondo la mobilità IP le permette di comunicare senza interruzioni, a patto che non cambi rete fisica più di una volta al secondo[112].

Ogni stazione mobile ha un indirizzo permanente detto dell'**home network** chiamato **home address**. Quando la stazione si muove in un **foreign network** deve acquisire un nuovo indirizzo locale detto **care-of address** (indirizzo "presso di").
La stazione usa l'home address per l'invio e la ricezione di pacchetti (come sorgente e destinazione), ma questi pacchetti vengono imbustati in un altro pacchetto facendo tunneling: i pacchetti vengono mandati all'home address ma consegnati al care-of address!

L'**home agent** è un device che si occupa di fare questo imbustamento a discapito della velocità del routing. Il care-of address può essere **co-located**, ovvero aggiunto alla stazione con DHCP o permanente ma con tunneling a carico dell'host e possibilità di avere un **foreign agent** che gestisce il tunneling

112 Questo rende la mobilità IP incompatibile con i veicoli, che attraversano più reti ad alta velocità.

direttamente nella LAN. La stazione si presenta solo con il care-of address ed in tal caso l'indirizzo aggiunto può essere condiviso da più stazioni. In ogni caso le stazioni devono comunicare il care-of address e registrarsi con il foreign agent che notificherà a sua volta l'home agent. La fase di registrazione deve sempre essere autenticata e si basa su ICMP, che permette alla stazione di capire se la rete supporta la mobilità e dove si trova (se è la home oppure no).

L'alternativa a mobile IP è **Proxy Mobile IPv6** (detto **PMIPv6** o semplicemente **PMIP**), che nasce per supportare IPv6 ma nella pratica supporta anche IPv4. Ha il pregio di non dipendere dall'host ed usa un device di rete che traccia gli host a prescindere dal loro stack supportando tutti i protocolli standard. Il device tracciante fa tunneling e gestisce da se tutte le notifiche usando complessi protocolli speciali. La stazione si dice **Mobile Node** (**MN**) e deve essere monitorata da un **Local Mobility Anchor** (**LMA**) nell'home network. Un **MAG** (**Mobile Access Gateway**) crea un tunnel con il LMA, ma MN e MAG devono trovarsi nel foreign network ed il LMA è equivalente all'Home Agent Mobile di IPv6.

Altra alternativa è **LISP** (**Location/Identifier Separation Protocol**), che separa i ruoli dell'IP (localizzazione + identificazione[113]) in due identificatori arbitrari.
LISP non è stato pensato per la mobilità ma le aziende lo usano anche per questo: si usa quando "il protocollo non dice al router come localizzare la stazione", come accade nei data center cloud con forte virtualizzazione. LISP è utile anche quando l'azienda è servita da più di un singolo ISP. LISP era in origine basato su BGP ma poi ha usato DNS ed altri protocolli. In ogni caso - a prescindere dall'host - basta che il router lo supporti e che l'identificatore "**locator**" cambi con lo spostamento.

> [...] In breve, ogni organizzazione locale che vuole permettere ai propri utenti di fare roaming, deve mettere a disposizione un supporto chiamato **home agent** (*agente domestico*). Quando un host mobile compare in un sito remoto ottiene un nuovo indirizzo IP temporaneo (chiamato anche *care of address*) dal sito remoto. Quindi, il dispositivo

113 **HIP** è un'alternativa a LISP che usa identificatori crittografici sicuri.

mobile indica all'home agent dove si trova in quel momento, dandogli l'indirizzo temporaneo. Quando un pacchetto per l'host mobile arriva all'indirizzo di casa mentre è da qualche altra parte, l'home agent prende il pacchetto e lo invia via tunnel all'host mobile facendo riferimento al suo indirizzo temporaneo. L'host mobile può inviare pacchetti di risposta direttamente a chi vuole comunicare con esso, ma usando l'indirizzo della rete di appartenenza come indirizzo sorgente. [...][114]

114 Cit. da *Fondamenti di Reti di calcolatori*, cap. 5.5.8, §2.

Fine del corso
di
Reti di Calcolatori

—

UniNettuno – 2018

Appendici

Networking con GNUstep Objective-C

Il seguente esempio di codice utilizza il framework GNUstep per inviare informazioni formattate tra sistemi GNU/Linux® e viene compilato direttamente in codice C (std. GNU). È richiesta la conoscenza avanzata di C, Objective-C, Automake e dello stack TCP/IP. L'esempio funzionerà sui sistemi GNU/Linux® e - con modifiche minori in base alla versione delle API - su Mac OS X di Apple®.
Ho progettato il codice affinché possa funzionare senza modifiche sui sistemi GNU/Linux® Debian[115] e possa essere integrato comodamente in un oggetto.

- `GNUmakefile:`

```
GNUSTEP_MAKEFILES=$(shell gnustep-config --variable=GNUSTEP_MAKEFILES)
include $(GNUSTEP_MAKEFILES)/common.make
TOOL_NAME = SocketExample
SocketExample_OBJC_FILES = VersionSocket.m
SocketExample_OBJCFLAGS = -DTOOL_NAME=\"$(TOOL_NAME)\" -DTARGET_OS=\"$
(GNUSTEP_TARGET)\"
include $(GNUSTEP_MAKEFILES)/tool.make
```

- `VersionSocket.m:`

```
#import <Foundation/Foundation.h>

int main(void) {
// Garbage Collection automatica
CREATE_AUTORELEASE_POOL(autoreleasePool);
NSLog(@"Init.");
NSPortMessage* netMessage = nil; // Messaggio
NSSocketPort* netInputPort = nil; // Porta in ricezione
NSSocketPort* netOutputPort = nil; // Porta in trasmissione

// Allocazione del messaggio
ASSIGN(netMessage, [NSPortMessage alloc]);
// Allocazione ed istanziazione della porta in ricezione
// Lo sviluppatore può estendere l'esempio per ascoltare la risposta del
peer su questa porta (se presente), integrando un NSRunLoop in base alle API
fornite della specifica versione del framework
ASSIGN(netInputPort, [NSSocketPort portWithNumber:4000 onHost:[NSHost
localHost] forceAddress:nil listener:YES]);
// Allocazione ed istanziazione della porta in trasmissione
// Se volessimo contattare un host remoto richiedendo la risoluzione DNS:
//ASSIGN(netOutputPort, [NSSocketPort portWithNumber:80 onHost:[NSHost
```

115 Debian GNU/Linux 9 (www.debian.org).

```
hostWithName:@"google.it"] forceAddress:nil listener:NO]);
// Se volessimo contattare la macchina corrente (mettere in ascolto ad es.
netcat sulle porte 7777 o usare direttamente tcpdump)
ASSIGN(netOutputPort, [NSSocketPort portWithNumber:7777 onHost:[NSHost
localHost] forceAddress:nil listener:NO]);
// Vogliamo inviare il nome della nostra applicazione (es. “PK”) in formato
ASCII
// Istanziazione di un blocco di dati contenente “PK”
NSData* outDataAppName = [@"PK" dataUsingEncoding:NSASCIIStringEncoding];
// Vogliamo anche inviare separatamente (ma nella stessa sessione) la
versione della nostra applicazione (es. “02”) in formato ASCII
// Istanziazione di un blocco di dati contenente “02”
NSData* outDataAppVersion = [@"02" dataUsingEncoding:NSASCIIStringEncoding];
// Istanziazione del messaggio di rete usando gli oggetti precedentemente
creati
// Il messaggio è composto da un Array di dimensione costante (NSArray)
contenenti i valori da inviare separati da alcuni bytes di padding ma senza
bytes in coda (il framework inserisce già dei metadati in testa!)
netMessage = [ netMessage initWithSendPort:netOutputPort
receivePort:netInputPort components:[NSArray arrayWithObjects:
outDataAppName, outDataAppVersion, nil] ];
// Impostiamo l’identificativo del messaggio di rete a “3” (sfruttando il
casting implicito garantito da intero ad NSInteger) per distinguere il
messaggio da oggetti eterogenei
[netMessage setMsgid:3];
NSLog(@"Sending with %@...", netMessage);
// Invio del messaggio sul socket precedentemente creato entro 3 secondi
(invio asincrono su TCP in modalità bloccante al massimo per 3 secondi)
BOOL sent = [netMessage sendBeforeDate:[NSDate
dateWithTimeIntervalSinceNow:3.0]];
NSLog(@"Sent: %i.", sent);
// Rilascio di tutti gli oggetti allocati
[autoreleasePool drain];
// Uscita con successo tramite funzione ANSI C
exit(EXIT_SUCCESS);
}
```

Fine del codice di esempio.

Integrazione di un kernel sperimentale aarch64

Segue uno script completo che scrissi per ovviare automaticamente al mancato riconoscimento delle interfacce wireless nei SoC delle board Raspberry Pi® 3 B: il kernel sperimentale aarch64 4.18 di GNU/Linux Debian usa codice nativo ARMv8 (64 bit) ma non è compatibile con i blob di codice proprietari della Raspberry Pi Foundation®. Il modo più elementare per ovviare al problema è scaricare manualmente il file DTB più aggiornato ed integrarlo nella directory del firmware, in modo che in seguito all'aggiornamento del kernel (tipicamente con riavvio automatico, essendo board usate per l'IoT) il SO possa rilevare le interfacce Wi-Fi®, Bluetooth® ed eventuali chip di espansione della connettività integrati tramite i socket GPIO e riconnettersi automaticamente tramite DHCP, riducendo al minimo il downtime.
Sono richieste alcune nozioni elementari di Bash scripting ed ovviamente la lettura preliminare del **man**uale di *sha256sum*, *test* e *wget*.
Questo script[116] [117] scarica il DTB da Internet tramite HTTPS, lo installa e ne verifica l'integrità usando l'hashing SHA256.

```
#!/bin/bash

# DTB UPDATER 0.2 for Raspberry Pi 3 (aarch64)
# Copyright (C) 2018 Lorenzo Ancora
# DTB UPDATER is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
# DTB UPDATER is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.

#
# EXAMPLE: update_DT_RPi3.sh
"https://repo.example.com/dtb/e0e02ecb1c455a7c8db3a53f9b408d6b/raw/16ec1953f3453
4a941bb90b2f6f42f30888157fc/bcm2710-rpi-3-b.dtb" "bcm2710-rpi-3-b.dtb"
# RETURN CODES: 0 (SUCCESS), 1 (FAILURE), 2 (FAILURE)
```

116 Questo script è compatibile con Bash (Bourne Again Shell) e rispetta gli standard GNU e POSIX.

117 Lo script è progettato per essere distribuito nei pacchetti .deb del kernel ed è quindi obbligatoriamente sotto licenza GPL versione 3. Eseguire lo script **solo** su board Raspberry Pi 3 B o su macchine virtuali ripristinabili.

```
#

DTBURL=$1                       # Full URL of the DTB file.
DTBDST="/boot/firmware/"        # Firmware directory of the Raspberry Pi 3.
DTBFIL=$2                       # Destination filename under DTBDST.
CKNAME="sha256.txt"             # File which contains the OUTPUT of "sha256sum
--tag <DTBFILE>", placed in the execution directory of this script.

function exitFailure () {
        echo "FAILURE: $1."
        if [ $2 -eq 1 ]; then
                echo "WARNING: your Wi-Fi interfaces may not be recognized."
                echo "WARNING: make sure that the latest" $DTBFIL "is in"
$DTBDST "before rebooting."
        fi
        exit $2
}

if [ ! -f $DTBDST$DTBFIL ]; then
        echo "Installing DTB file for Raspberry Pi 3 from" $DTBURL "..."
        wget --quiet --timeout=7 --random-wait --tries=3 --retry-connrefused
$DTBURL --output-document=$DTBDST$DTBFIL
        if [ -f $DTBDST$DTBFIL ]; then
                echo "DTB STATUS: UPDATED."
        else exitFailure "DOWNLOAD FAILED" 1
        fi
        cp $CKNAME $DTBDST$CKNAME
        pushd "$DTBDST" > /dev/null
        sha256sum --status --strict --check $CKNAME
        ckRes=$?
        popd > /dev/null
        if [ $ckRes -eq 0 ]; then
                echo "DTB INTEGRITY: CONFIRMED."
                rm -f $DTBDST$CKNAME
        else exitFailure "NEW DTB FILE IS CORRUPTED" 1
        fi
        exit 0 # SUCCESS
else exitFailure "DTB ALREADY INSTALLED (PLEASE BACKUP AND DELETE $DTBFIL FROM
$DTBDST)" 2
fi

### EXAMPLE OUTPUT: ###
# Installing DTB file for Raspberry Pi 3 from
https://gist.github.com/chschlue/e0e02ecb1c455a7c8db3a53f9b408d6b/raw/16ec1953f3
4534a941bb90b2f6f42f30888157fc/bcm2710-rpi-3-b.dtb ...
# DTB STATUS: UPDATED.
# DTB INTEGRITY: CONFIRMED.
###       EOF       ###
```

Fine del codice di esempio.