Internet, Reti e Sicurezza
Digital Ed.

cleopinoli

Contents

Introduzione

Cos'è Internet?

Fondamentalmente è una rete che collega tra loro più unità di calcolo sparse geograficamente. Si tratta di una rete a commutazione di pacchetto (packet-switch network, PSN), in cui più terminali (o End System, o Host, insomma qualsiasi tipo di computer che si trova ad un capo estremo della rete, un laptop, uno smartphone, un sensore...) condividono lo stesso cammino di rete o una parte di esso. Torneremo su commutazione di circuito e di pacchetto a breve.

Normalmente, i terminali non sono collegati tra loro in modo diretto, ma tramite dispositivi di commutazione e instradamento (router) che prelevano le informazioni in dei pacchetti e le inoltrano sui link di uscita. Il percorso compiuto dai pacchetti attraverso la rete prende il nome di Route, o Path. Letteralmente *percorso*.

L'accesso dei terminali ad Internet avviene attraverso gli ISP (Internet Service Provider), aziende di telecomunicazioni come AT&T o Vodafone, che costuiscono ciascuna una rete di router a cui poter collegare gli access point come la ADSL domestica (che di per sé costituisce una rete, in questo caso domestica, in cui tutti gli Host comunicano tra loro (internamente) e con il resto di internet (esternamente) attraverso quello che comunemente chiamiamo Modem o Gateway).

TCP/IP: insieme di protocolli (spesso si parla di stack TCP/IP, perché sono posti uno sopra l'altro) che gestiscono invio e ricezione di pacchetti in rete.

Intranet: è un tipo di rete privata, strutturata similmente alla pubblica Internet.

Protocollo: insieme di regole formalmente descritte al fine di favorire la comunicazione tra una o più entità (è un termine generico che si può applicare anche ad altri contesti, ovviamente in questo caso parliamo di Host o di componenti in grado di comunicare in rete).

App distribuite: in Internet si lavora tramite app distribuite, i.e. due o più processi che vengono eseguiti in parallelo su macchine diverse e che interagiscono tra loro attraverso Internet.

Host: termine che si riferisce a dei sistemi periferici che ospitano programmi applicativi come ad esempio web browser; Gli host si suddividono ulteriormente in Client e Server, rispettivamente host che richiedono un serivizio e quelli che lo forniscono. Un singolo host può fungere da client e da server (connessione Peer-to-Peer).

DSL (Digital Subscriber Line): accesso residenziale a banda larga.

Stack dei layer di Internet: TCP/IP VS modello ISO/OSI

TCP/IP		ISO/OSI		
Layer	n°	Layer	n°	Esempi di Protocolli
Applicazione	5	Applicazione Presentazione Sessione	7 6 5	HTTP(S), POP, SMTP, FTP, SSH, DHCP...
Trasporto	4	Trasporto	4	TCP, UDP
Rete	3	Rete	3	IPv4, IPv6, ICMP
Datalink	2	Datalink	2	MAC
Fisico	1	Fisico	1	Ethernet, cavo coassiale

Sono due rappresentazioni della stessa stack di classi di protocolli, semplicemente l'ISO[1] ha separato l'application layer in 3 layers (Application, Presentation, Session), che è una rappresentazione un po' più dettagliata, ma noi faremo riferimento principalmente al modello TCP/IP. (N.B.: alle volte, nel modello TCP/IP, i layer Datalink e Fisico[2] vengono trattati indistintamente, come fossero un layer unico.)

Tipi di ritardo e altri concetti

Tipi di ritardo (accenno, ci torneremo)[3]

i. di Elaborazione: si crea nel router per esaminare il pacchetto;

ii. di Accodamento: si crea in coda in uscita, un buffer;

[1] International Standardization Organization, l'agenzia che si occupa di distribuire standard di varie tecnologie e produzioni.

[2] il layer fisico/physical alle volte viene detto anche Network Access Layer.

[3] in verità quasi tutto questo capitolo è un accenno a cose che vedremo più avanti in maggiore dettaglio, ndr.

iii. di Trasmissione ($= \frac{L}{R}$): dipende dalla dimensione del pacchetto (L);

iv. di Propagazione: prettamente fisico, dipende dalla distanza tra router A e router B.

La somma di questi ritardi costituisce il nodal delay, il ritardo accumulato ad ogni nodo del percorso.

Network core vs Network edge: fondamentalmente il network edge sono i sistemi periferici (host come laptops o smartphones), mentre il network core è la struttura che li collega (routers etc.).

LAN (Local Area Network): usata per collegare end systems ad un edge router. La tecnologia d'accesso usata di solito in queste reti è Ethernet.

Standard IEEE che regolamenta il Wi-Fi[4]: 802.11.

Qualche nota sui Physical Media (Mezzi di trasmissione fisici, i.e. cavi eccetera):

- Twisted Pair Copper wire (il doppino intrecciato in rame):
  - UTP (Unshielded Twisted Pair, doppino non schermato): usato comunemente nelle LAN. Un cavo UTP di categoria 6A può trasmettere ad una velocità fino a 10Gbps;
  - La ragione per cui i fili nel doppino sono intrecciati a due a due a quella maniera ha a che fare con la cancellazione dei disturbi del segnale;
- Cavo coassiale: quello a sezione circolare dell'antenna TV. Può essere usato in contemporanea da più end systems; alcune categorie trasmettono a velocità nell'ordine di diversi Gbps;
- Fibra ottica: molto veloce, molto costosa, va dai 51.8M ai 39.8 Gbps;

Il tempo di trasmissione è dato da $\frac{L}{R}$, dove L = dimensione del pacchetto in bit e R = transmission rate sul link.

Store-and-forward: il router deve ricevere tutto il pacchetto prima di inoltrarlo dove opportuno; così facendo, il ritardo totale diventa $= \frac{2L}{R}$ (per ciascun nodo, credo).

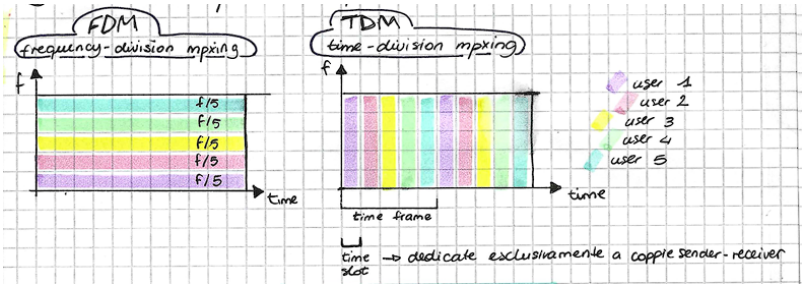
"Modem" sta per Modulatore-Demodulatore: sebbene spesso ci riferiamo allo "scatolotto" che collega la nostra LAN domestica ad Internet con il termine Modem (quei dispositivi come il TIM Hub, insomma, a volte li si chiama anche Router, che non è troppo sbagliato), in realtà il modem è un modulo incluso in quello scatolotto.

Commutazione di Circuito vs Commutazione di Pacchetto:

[4]Secondo la Wi-Fi Alliance, il termine "Wi-Fi" non è mai stato la contrazione di "Wireless Fidelity, sebbene l'IEEE affermi diversamente.

- Comm. di circuito: le risorse richieste da un end system per comunicare sono riservate ad esso per tutta la durata della connessione; è il caso della telefonia, se qualcuno sta usando il telefono di casa in una stanza, un secondo telefono collegato alla stessa linea non può avviare un'altra chiamata);
- Comm. di pacchetto: le risorse non sono riservate, ma più end systems usano il canale più la banda viene limitata a ciascuno;

La commutazione di circuito può avere due tipi di multiplexing: Frequency-division e Time-division:



In Fig. 1 è riassunta la struttura di Internet e le sue evoluzioni con le varie addizioni di elementi come i Point of Presence & IXP.

Intensità di traffico: $\frac{L \cdot a}{R}$, L e R li abbiamo già menzionati, a = average rate of packets per second; un'intensità di traffico > 1 equivale a dire che la # di bit che arriva in coda supera la # di bit che esce dalla coda (converrete con me che è una cosa da evitare, significa che la roba che entra supera la roba che esce, c'è congestione).

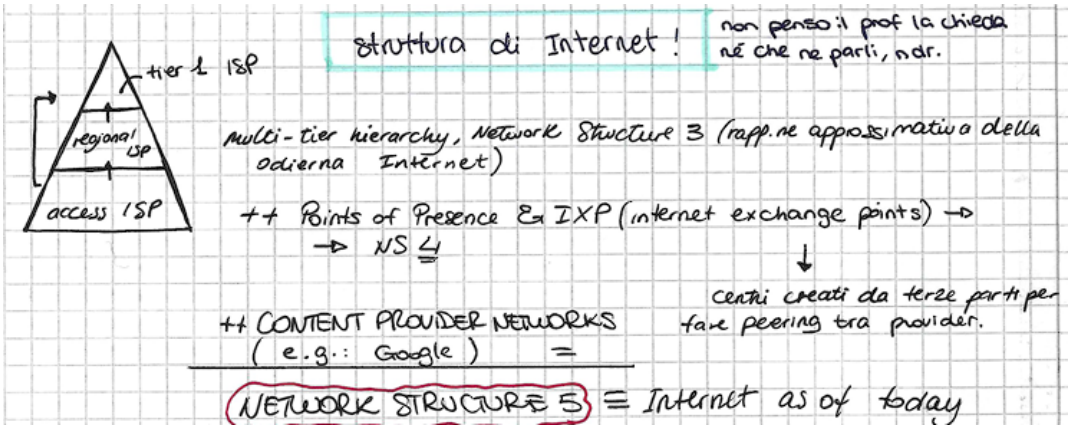
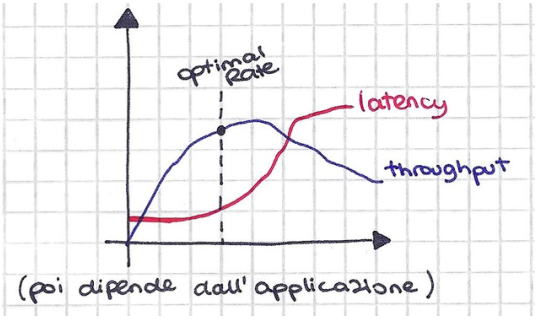


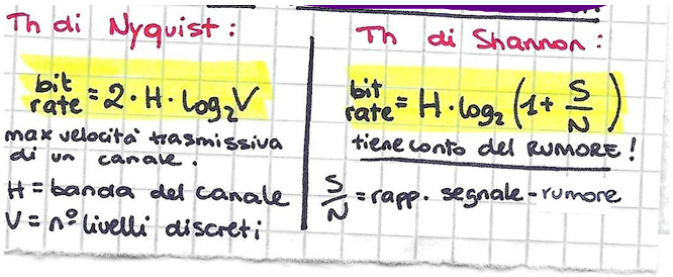
Figure 1: Sì non credo vi verrà richiesta, ma nel dubbio è una cosa del genere.

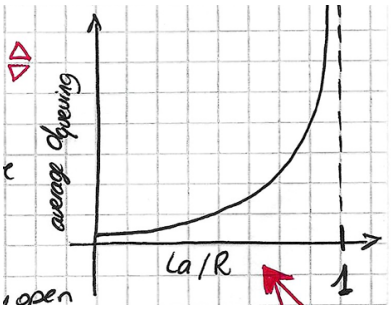
Latenza: intervallo di tempo tra quando spedisco un input e quando è disponibile l'output.



Throughput: "vera velocità" di rete, vera capacità di un canale trasmissivo. Non si può considerare costante, si parla solo di throughput medio end-to-end.

Bandwidth: intervallo di frequenza che un sistema può garantire per trasmettere. Bitrate: dipende dalla bandwidth (teorema di Nyquist-Shannon) (garantito nominalmente).





Intensità di traffico: $\frac{L \cdot a}{R}$, L e R li abbiamo già menzionati, a = average rate of packets per second; un'intensità di traffico > 1 equivale a dire che la # di bit che arriva in coda supera la # di bit che esce dalla coda (converrete con me che è una cosa da evitare, significa che la roba che entra supera la roba che esce, c'è congestione).

Note sugli attacchi DoS (Denial of Service):

- vulnerability attack: usa un messaggio "maligno" per compromettere app o host;
- bandwidth flooding: invio smisurato di pacchetti al target allo scopo di congestionare l'access link e renderlo inutilizzabile;
- connection flooding: richiesta elevata di connessioni TCP fully open (che rimangono sempre attive) al punto tale che l'host vittima non riesce più ad accettarne altre genuine. Spesso questo attacco è reso possibile grazie a Distributed DoS, i.e. l'attacker controlla più macchine "zombie" che dirotta a sua discrezione all'attacco della vittima.

Packet sniffing (che è quello che fa Wireshark): i bad guys si posizionano tra due host e "origliano" il traffico senza manometterlo allo scopo di estrarre informazioni sensibili;
IP spoofing: "rubare" l'identità di qualcun altro mediante il suo indirizzo IP;
Traceroute: manda tot pacchetti ai router compresi tra host e destinazione, rilevandoli tutti

```
tracert www.rai.it (212.162.68.64)
```

Ogni protocollo appartiene ad un qualche layer della stack e può essere hardware, software o misto.

PDU (Protocol Data Unit): l'unità di messaggio, che a seconda del layer in cui si trova prende un nome diverso (e.g.: frame, datagramma). La PDU è sempre formata da un header e da un payload, dove header contiene informazioni per la trasmissione e il payload è il contenuto del messaggio;
Incapsulamento: ciascun layer riceve il messaggio dal layer sopra, aggiunge un suo header, passa il nuovo messaggio al layer sotto. Nel layer sotto succede di nuovo: il layer riceve questo (header + payload) e lo considera come un payload', aggiunge il suo header e passa header' + payload' ancora sotto;

SAP Service Access Point: servizi messi a disposizione dalle interfacce che si trovano tra due layer comunicanti della stack.

Per fornire accesso a Internet agli utenti, gli ISP distribuiscono lungo il territorio dei PoP, Points of Presence.

Indirizzi della forma 192.168.1.### non sono indirizzi pubblici, ma interni alla nostra rete.

Fiber To The...

- FTTH: ...Home
- FTTN: ...Node
- FTTC: ...Cabinet
- FTTS: ...Street (=FTTC)
- FTTB: ...Building

Internet ha un generale approccio al servizio detto "best effort" (i.e.: "che Dio ce la mandi buona"): nulla è di fatto garantito al 100%, ma tutti fanno del proprio meglio perché funzioni.

Legge di Metcalfe (che allego qui ma non vi serve): "L'utilità e il valore di una rete sono proporzionali al quadrato del nr di utenti." Dato il nr di utenti, il massimo numero di connessioni possibili è $= n^2 - n$. (Si tratta di una legge di dubbia correttezza e largamente contestata, oltre al fatto che non è mai stato possibile testarla con dati reali).

LO! Story Time: La prima parola ad essere trasmessa in una versione preistorica di Internet è stata: 'LO'. Transitava dal UCLA al SRI e doveva finire per essere la parola "LOGIN", lettera dopo lettera, ma per qualche ragione l'host del SRI crashò ricevendo la G. L'ingegnere Leonard Kleinrock, che era lì a lavorare al progetto ARPANET, scrisse in una celebre intervista che quel tentativo fallito di trasmettere la parola in rete ebbe comunque qualcosa di *profetico:* "Lo", come in "Lo and behold!" (che più o meno significa *"Ecco, preparatevi!"*), come a presagire la svolta enorme che sarebbe arrivata in futuro dopo quell'esperimento - con la nascita di Internet.

Application Layer

(Layer 5 nello stack TCP/IP)

Introduzione

Ci sono 2 principali architetture in cui gli host si dividono:

- Client-Server (e.g.: web server);
- Peer-to-Peer;

Network application: coppia di processi che si scambiano messaggi.

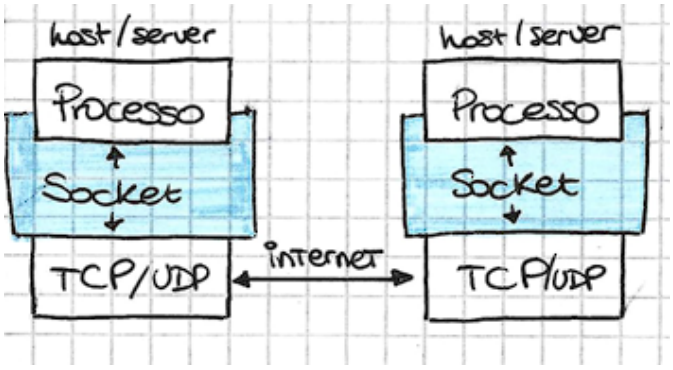


Figure 2: Network application. Le socket (o API) sono interfacce software tra l'application layer e il transport layer di ciascun host. Ne esistono tra tutte le coppie di layer adiacenti.

Per identificare il processo ricevente serve specificare: l'indirizzo IP del Host, e un numero di porta che è dedicato al servizio (in maniera permanente o temporanea), tipo 100.100.100.1:80; ci sono diversi numeri di porta noti per essere riservati a certi servizi, tipo 80 è una porta riservata ad HTTP, 443 ad HTTPS, oppure 110 a POP3, 20 e 21 a FTP (dati e controllo rispettivamente),

67 e 68 a DHCP (requests e replies rispettivamente), 25 al mail server SMTP, 520 a RIP, e così via.[5]
La nostra app, a livello sottostante (transport layer), potrà usare principalmente due protocolli:

- TCP (Transmission Control Protocol)
  - servizio orientato alla connessione (è richiesto che le parti comunicanti si identifichino e completino un handshake per iniziare a comunicare);
  - trasferimento dati affidabile (gestione della perdita di pacchetti);
  - HTTP, SMTP, FTP, HTTP 1.1, HTTPS, Telnet etc. si appoggiano tutti su TCP per il trasporto
- UDP (User Datagram Protocol)
  - minimale, leggero, trasmette il minimo indispensabile delle informazioni necessarie per consegnare un pacchetto da punto A a punto B;
  - connectionless: niente handshake, niente connessione preliminare tra A e B. Il messaggio viene sparato da A e se arriva o meno non è dato saperlo.
  - per via della sua leggerezza, UDP alle volte viene utilizzato per lo streaming live e on-demand (e.g.: Twitch), servizi per cui non è necessario garantire che tutti i dati arrivino a destinazione (al limite si perde qualche frame, ma il contenuto si capisce ancora abbastanza bene).

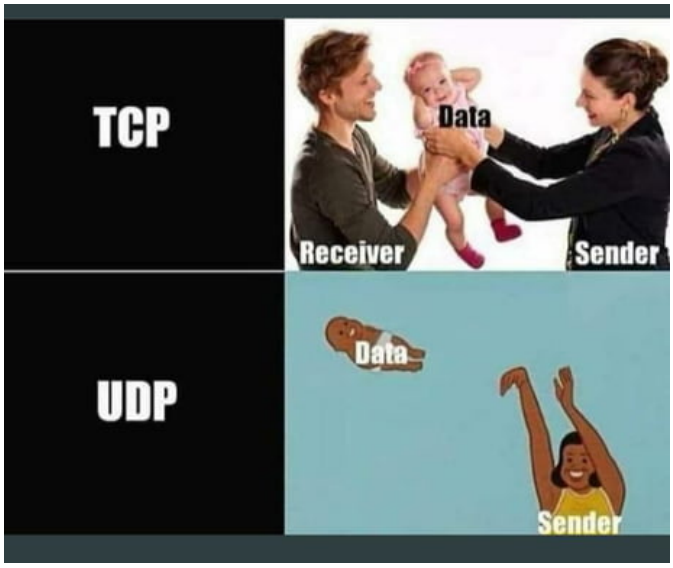


Figure 3: Non so a chi attribuire questa creazione, ma è uno dei miei esempi umoristici preferiti riguardo TCP vs UDP. Kudos al creatore originale.

[5] I numeri di porta sono documentati alla pagina: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

Nessuno dei due implementa sistemi di crittografia, a meno che non si usi TCP con TLS (Transport Layer Security), ma di base TCP non fornisce questo servizio.

HTTP

Hyper Text Transfer Protocol, default port: 80

Protocollo a livello di applicazione Web: una pagina Web è un insieme di oggetti (e.g.: file HTML[6], immagine JPEG, file JavaScript, file CSS...). Tipicamente alla base delle pagine Web c'è un file HTML di base, che fa riferimento agli altri oggetti della pagina attraverso i relativi URL[7].

I Web Browser (e.g.: Google Chrome) implementano il lato Client di HTTP;
È un protocollo stateless: se un client manda due richieste identiche nell'arco di qualche secondo, HTTP non ne ha alcuna memoria, risponderà 2 volte con la stessa risposta.
Connessioni TCP:

- persistente: tutte le richieste/risposte avvengono nella stessa sessione TCP;
- non persistente: ogni richiesta/risposta avviene in una sessione TCP nuova a se stante (ogni volta ne viene avviata una nuova).

HTTP usa una connessione persistente di default, ma può usarle entrambe.

RTT - Round Trip Time: tempo impiegato da un pacchetto per andare da client a server e poi tornare indietro.

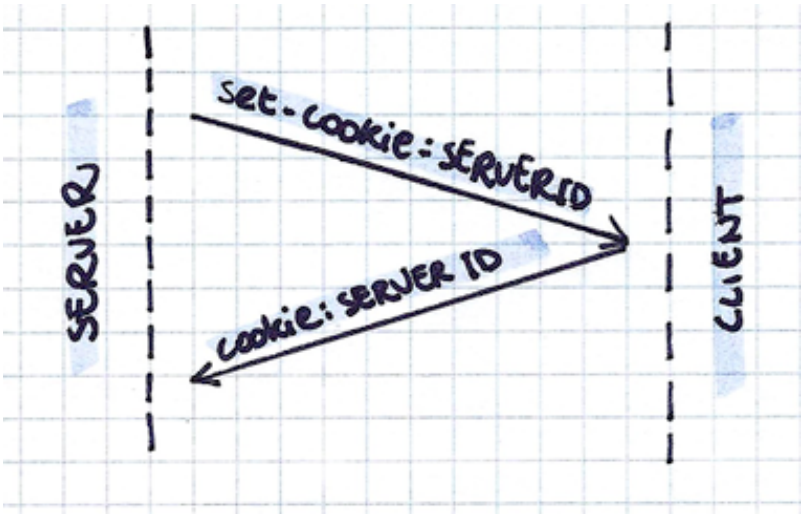
Messaggi HTTP	
Richiesta	Risposta
get[1] /dir/page.html[2] HTTP/1.1[3] [1]Request line - metodo [2] URL [3] versione HTTP	HTTP/1.1 200 OK[4] [4]stato
Header: host: www.(...) connection: close \textit{(non persistente)} user_agent: Mozilla/5.0 (tipo browser) ... *altre istruzioni HTTP*	Header: connection: close date:[current date] server: Apache/2.2.3 last-modified: [date] Content length: 6821 [byte] Content type: text/html *RIGA VUOTA, fine Header* Contenuto: data data data...

[6]Hyper Text Markup Language
[7]Uniform Resource Locator

Qual è la differenza tra HTTP 1.0 e HTTP 1.1? La più importante forse è la persistenza della connessione (abbiamo visto cosa significa poco fa), oltre ad altre cose come, ad esempio, l'introduzione di meccaniche di caching.

Cookies

(argomento interno ad HTTP) I cookies sono un tipo di informazione che viene memorizzata lato client. Può servire in un secondo momento al server per ristabilire una connessione.



- ogni browser/sistema operativo archivia i cookies in un posto a sua discrezione, non esiste una convenzione universale su dove salvare i file di cookies;
- inizialmente, tutti i cookies erano semplici file di testo, ora la maggior parte è in formato SQLite (piccoli database!);
- non tutti i siti web usano i cookies;
- (tecnicamente,) i cookies sono facoltativi.
- i cookies possono essere disabilitati;
- i cookies sono una minaccia per la privacy in quanto profilano gli utenti! Raccogliendo informazioni sulle loro abitudini, comportamenti etc. Vengono usati in quella che si dice Pubblicità comportamentale: analisi dell'attività in rete degli utenti al fine di permettere alle aziente di presentare a ciascuno pubblicità mirate ai loro gusti.[8]

[8] https://youronlinechoices.com/it/

Web Caching

(o Proxy Server)

È un'entità della rete in grado di rispondere ad alcune richieste HTTP al posto dei server a cui sono indirizzate. Funziona circa così:

1. Browser invia richiesta HTTP al proxy per (ad esempio) una pagina web;

2a Se il Proxy contiene una copia in locale della pagina richiesta → il Proxy risponde al client;

2b Se il Proxy non contiene una copia della pagina → il proxy richiede la pagina al server che dovrebbe averla, una volta che la ha ricevuta la recapita al client browser e ne salva una copia in locale.

"Conditional GET", richiesta dal proxy per il server: se l'oggetto richiesto è stato modificato di recente, mandamene la copia aggiornata, altrimenti manderò al client quella che ho io.

Perché si fa questo lavoro di web cache?

- per ridurre i tempi di attesa[9];
- ridurre il traffico complessivo nel Web.

L'intensità del traffico può essere misurata come:

$$T = X \cdot \frac{Y}{Z}$$

dove:
X = richieste/secondo;
Y = Megabit/richiesta;
Z = Megabit/secondo.

e-Mail

Componenti chiave del sistema di mailing in Internet:

- User Agents: lo strumento con cui si leggono/scrivono le mail, come la app di Gmail per Android o un browser web che accede al sito https://mail.google.com/mail/;
- Mail Servers: il "core" dell'infrastruttura di e-mailing. È il mail server che contiene le caselle degli utenti, i messaggi viaggiano dal MS del mittente alla mailbox nel MS del destinatario/i;

[9] normalmente l'intensità di traffico è molto più alta al di fuori di una LAN (in uscita verso Internet) che non al suo interno, da cui le potenziali latenze elevate.

- SMTP (Simple Mail Transfer Protocol): il principale protocollo a livello applicativo per la posta elettronica.

Principali comandi SMTP: HELO (hello), MAIL FROM, RCPT TO, DATA, QUIT.

SMPT è un "push protocol", si può usare solo per *inviare* messaggi di posta elettronica; per richiedere dei messaggi da leggere, invece, occorre utilizzare un altro tipo di protocollo.

HTTP e IMAP (Internet Mail Access Protocol) permettono di fare cose come gestire cartelle, eliminare messaggi etc. Nello specifico, HTTP serve giusto a supporto di Web app come Gmail, IMAP in sé è sufficiente se si vuole richiedere un messaggio di posta da un server e leggerlo da un'interfaccia anche spartana.

POP3

POP3 è un protocollo di posta elettronica (Post Office Protocol). Come IMAP, POP lavora in ASCII su 2 porte con TCP (110, 995). Comandi client per autenticazione sono semplicemente `user` e `pass`, il server risponde con `+ok` o `-err`.
Transazioni:

- `list`: elenca i messaggi (numero e dimensione);
- `retr` $< n >$: retrive msg con numero n;
- `dele`: delete msg;
- `quit`: esci

DNS

Domain Name System, default port: 53

Si tratta di un database distribuito e di un protocollo a livello applicazione per inviare queries ai Server DNS, si occupa di tradurre dli hostname (e.g.: *www.facebook.com*) nei corrispondenti indirizzi IP (e.g.: *69.63.176.13*).

DNS offre servizi quali Host aliasing (tradurre un hostname in un altro hostname, spesso meno user-friendly

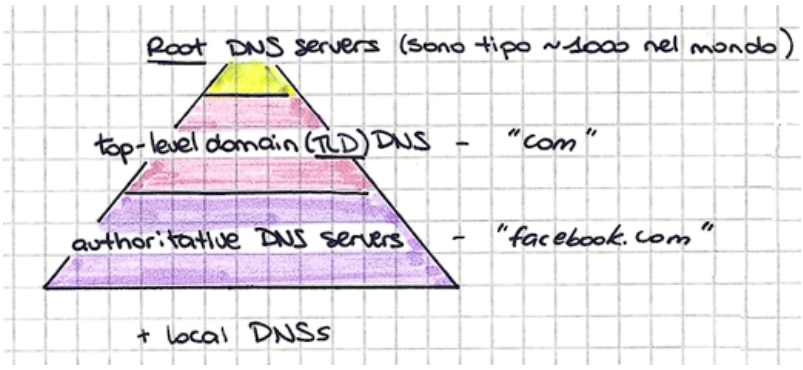


Figure 4: Gerarchia a tre livelli dei DNS.

di quello ricevuto), mail server alias-
ing, load distribution.

DNS Caching: stesso meccanismo del Web caching, ma applicato al DNS.

Resource Records (RR): informazioni immagazzinate nei database DNS. Sono triple della forma

```
(Name, Value, Type, TTL)
```

Dove TTL sta per Time-To-Live, tempo di vita del record nel database (scaduto il quale questo viene eliminato).

Type	Name	Value
A	hostname	IP Address
NS	domain	hostname del DNS che sa ottenere l'IP degli host nel dominio
CNAME	host	valore canonico per l'host che ha sinonimo *nome*
MX	host	valore canonico di un mail server per host sinonimo di *nome*

Table 1: Alcuni esempi di Resource Record. I valori dei campi "Name" e "Value" dipendono dal valore in "Type".

Telnet

Uno dei primi protocolli in assoluto per TCP/IP, basato su architettura client-server, molto semplice ma nella sua semplicità è deprecato, ora al suo posto si usa SSH (Secure SHell). Fondamentalmente Telnet permette di fare collegamenti ad altri terminali attraverso la rete (alla porta 23).

Come funziona: il client manda richieste al server Telnet, e quello risponde. Ogni tasto che viene battuto sulla tastiera del lato client, ogni singolo carattere, viene spedito sulla rete e ricevuto dal server. Il server, ricevuto un carattere, lo rimanda in risposta come una "echo". Quindi non stiamo parlando di digitare una parola o una frase e poi premere "invio" per inviarla al server, l'invio avviene lettera per lettera.
Server a fare accesso remoto ad altre macchine. Perché non si usa più e piuttosto si usa SSH? Perché Telnet trasmette in cleartext, ovvero: ogni carattere viene trasmesso in chiaro, senza essere cifrato in nessuna maniera. Attraversa la rete esattamente come viene scritto. Chiunque riesca a posizionarsi tra client e server potrebbe origliare ciò che viene detto e impossessarsi delle informazioni.

Virtual Machines

(cenni).

Una Macchina Virtuale (VM), essenzialmente, è un software che, montato e avviato, è in grado di simulare un computer a sé stante istanziato all'interno del nostro computer. Virtualmente, è come se ritagliassimo qualche pezzetto hardware dal nostro computer e usassimo questi pezzetti per creare un computerino fatto dai ritagli del nostro computer. Ok, questo, ma a livello software. Senza segare letteralmente la CPU a metà.

Per fare ciò, presa un'immagine di un sistema operativo, dovremo usare un Hypervisor, cioè un programma che virtualizza degli host usando l'hardware della nostra macchina. Se avete abbastanza risorse hardware, potreste anche avviare più di una macchina virtuale contemporaneamente! L'hypervisor che ci è stato consigliato si chiama **VirtualBox** di Oracle[10], in alternativa ce ne sono altre come VMWare Workstation. Perlomeno nelle versioni base, entrambi quelli che ho menzionato sono gratuiti.

FTP

File Transfer Protocol. Protocollo per il trasferimento di files.

- di tipo client-server;
- usa connessioni TCP su porte 20 e 21, entrambe aperte dal client:
  - porta 20 per i dati;
  - porta 21 per i comandi (controllo);
- permette operazioni di upload e download;
- spesso oggi anziché usare FTP si preferisce scambiarsi file usando HTTP, perché come Telnet, anche in FTP tutte le informazioni transitano in chiaro, in formato ASCII.

Qualche comando FTP:

- USER: username;
- PASS: password;
- LIST: elenca i contenuti della cartella;
- RETR: retrieve, richiedi un file;
- STORE: carica un file.

Come HTTP, FTP ha dei codici di stato (tipo il "404 Not Found"). Alcuni codici sono:

- 331 : username ok, password required;

[10] https://www.virtualbox.org/wiki/Downloads

- 125 : data connection already open; transfer starting;
- 425 : can't open data connection;
- 452 : error writing file;

Nel 99% dei casi, FTP si usa da interfaccia a linea di comando (CLI): per avviare una connessione ftp, si usa il comando (è possibile autenticarsi anche come anonymous)

```
ftp <indirizzo>
```

La questione PASV/PORT che forse vi interessa: in Windows 10, usare FTP può dare problemi a causa della questione della *"modalità passiva/modalità port" (PASV/PORT)*. PASV e PORT sono entrambi comandi per la connessione dati (una delle due porte usate da FTP, no? Dati e comandi, ecco, quella dei dati); la connessione dati FTP, a volte, viene ostacolata dai Firewall, quindi se in principio bastava usare il comando PORT, successivamente all'avvento di NAT e Firewall si è reso necessario aggiungere PASV, che non è altro che una modalità trasferimento dati compatibile con Firewall.

tFTP

Ovvero, trivial FTP. A differenza di FTP, tFTP:

- usa UDP (alla porta 69);
- ha pochissime funzioni (da cui il termine *"trivial", banale*;
- non conosce il concetto di Directory (cartella);
- non usa autenticazione;
- ha quindi un utilizzo molto limitato;
- ha 2 modalità di trasferimento: ASCII (NETASCII) e binario (OCTET);

alcuni comandi tFTP:

- **RR**: read request;
- **WR**: write request;
- **DATA**: dati;
- **ACK**: acknowledged;
- **ERR**: errore;

I pacchetti UDP inviati da tFTP sono a lunghezza fissa $= 512$ Bytes. La trasmissione si considera conclusa quando viene ricevuto un pacchetto di dimensioni < 512 B.

SNMP

Simple Network Management Protocol

Normalmente, se presente, la porta standard usata è 161 (UDP). Anche SNMP viene spesso bloccato dal firewall - com'è giusto che sia, visto che per design questo protocollo dovrebbe arrivare soltanto dall'interno.

Usato per gestione di reti, monitoraggio e configurazione di dispositivi di rete.

Consiste in un protocollo molto semplice che fa:

```
GET - GETNEXT - GETBULK - SET - TRAP
```

Fa uso di Access Control List (ACL), regole che indicano cosa può e cosa non può fare un dispositivo);

Si possono istanziare manager o agent ($\approx$ server, cioè un'entità che ha il suo database di MIB, Management Info Base) - ogni oggetto MIB è identificato da una serie di numeri in un formato tipo:

```
1.3.6.1.2.1.1.3 = SysUptime
```

E' un modo macchinoso di immagazzinare informazioni riguardo il device e la rete. Tutti quei numeretti sono come un sistema di coordinate GPS, l'interno di questi record somiglia vagamente a un JSON.

DHCP

(argomento interno all'app layer)

Dynamic Host Configuration Protocol - serve a configurare gli host.
Problemi sorti nella decisione di "come deve essere fatto un sistema di idnirizzamento:

- duplicazione degli indirizzi IP: che facciamo se qualcuno si assegna un indirizzo IP che già appartiene ad un altro Host?[11]
- riassegnazione degli indirizzi IP: ad esempio, come riassegniamo gli indirizzi quando spostiamo un ufficio da una sede ad una nuova sede fisica?
- spreco di indirizzi IP inutilizzati;
- interfaccia che renda possibile vedere tutti questi indirizzi;[12]

Usando TCP, si possono usare 2 protocolli che si occupano di configurazione host (assegnano indirizzi IP in maniera automatica):

- BOOTP : funziona con i thin client (host senza disco), usa UDP;
- DHCP : come BOOTP, ma ulteriormente sviluppato. Introduce il parametro di "lease" : parametro di tempo, durante il quale l'host può usare l'indirizzo IP che DHCP gli ha assegnato (tipo un token di noleggio).

DHCP configura:

- indirizzo IP;
- maschera di sottorete;
- indirizzi Gateway e DNS (ma non sempre);
- altri parametri.

L'addressing può avvenire in 3 modalità:

[11] Tra l'altro, e mi permetto di aggiungere "ovviamente", gli indirizzi IPv4. Per ovviare a questo problema, spesso si ricorre al NAT, Network Address Translation, che si fa anche per risparmiare sull'uso degli indirizzi IP pubblici.
[12] se avete curiosità, fate un salto al sito showmyip.com/

- assegnazione manuale (o reservation): legata all'indirizzo MAC;
- assegnazione automatica: il tempo di lease normalmente è $= \infty$;
- assegnazione dinamica: il tempo di lease è molto breve $\approx 1h$;

L'addressing avviene in $4(+1)$ fasi:

1. discover;
2. offer;
3. request;
4. ack;
5. release (non uso più l'indirizzo, i.e. ho spento il PC)*;

*È buona pratica che non si interrompa la connessione scollegando il cavo o mettendo il PC in standby quando si lascia un qualche luogo: così facendo, il DHCP non riesce a venire a sapere che la connessione è stata interrotta (si spreca l'indirizzo).

Note di Lab: Il protocollo NETBEUI o NETBIOS (di Microsoft, nei sistemi Linux esiste il protocollo SAMBA), un vecchio protocollo di rete che funziona solo in locale. Il DNS di NETBEUI si chiama WINS (Windows Internet Name Service).

Banalmente, per essere raggiungibile, il DHCP Server deve essere configurato con indirizzo Statico!

DHCP Relay: funzione implementata nei router per cui le richieste DHCP dagli host vengono inoltrate ad un DHCP Server limitrofo. Se non c'è un DHCP Server a disposizione, un sistema Windows ha IP 169.254.0.0/16.

Transport Layer

(Layer 4 nello stack TCP/IP)

A livello di trasport, Internet utilizza due protocolli[13]

- TCP;
- UDP;

Sappiamo già che TCP è connection-oriented, con meccanismi di controllo della congestione e di flusso e di sicurezza etc., laddove invece UDP si limita a spedire il pacchetto dove gli viene indicato e Dio provvede di quello che succede al pacchetto.

Un pacchetto dati a livello di trasporto prende il nome di segmento: in alcuni documenti e RFC[14], i segmenti vengono chiamati anche "datagram", ma con quel termine c'è un piccolo problema di ambiguità: infatti, con il termine "datagram" ci si riferisce anche ai pacchetti dati che raggiungono il livello di rete. Tutte le mie source utilizzano il termine "segmento", quindi magari usiamo quello e basta.

In genere, un segmento a livello di trasporto contiene tre elementi:

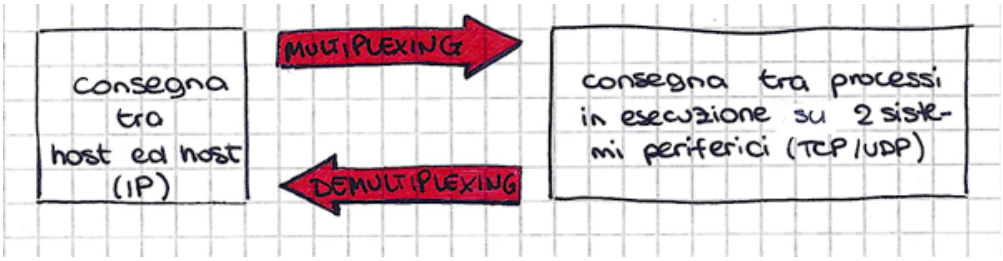
1. nr. porta di origine / nr. porta di destinazione (campi lunghi 16 bits ciascuno, per un totale di $16 + 16 = 32$);
2. altri campi di intestazione[15] (tipo e lunghezza variano tra TCP e UDP);
3. messaggio (il contenuto).

Multiplexing a livello di trasporto: raduna diversi dati da diverse socket e li incapsula in un unico pacco di dati da spedire in rete (è importante che ogni socket abbia un numero di porta per completare questa operazione in avanti e indietro). In genere multiplexing è una qualche

[13] nota: sebbene TCP e UDP siano i due protocolli più diffuso per il transport layer, non sono gli unici due protocolli esistenti per il trasporto in assoluto!

[14] non li avevo menzionati finora? RFC sta per Request For Comment, sono documenti che stabiliscono praticamente degli standard di questo e quel protocollo o meccanismo, li pubblicano svariati enti come la IETF.

[15] useremo i termini "intestazione" e "header" intercambiabilmente.



operazione che prende n input ed ha 1 output;

Demultiplexing a livello di trasporto: esamina determinati campi del messaggio ricevuto e decide a quale socket del ricevente consegnarlo. Qui stiamo parlando di ricevere da un unico input e di distribuire il contenuto ricevuto a n possibili output;

Sia TCP che UDP fanno MPXing e DMPXing.

Nota abbastanza importante sui numeri di porta:

- *le porte numerate da* $0 \div 1023$ *vengono dette "porte note" perché sono riservate a servizi quali HTTP. Non si possono usare arbitrariamente per altre cose;*
- *le porte comprese tra* $1024 \div 49552$ *si chiamano "indirizzi effimeri", assegnati randomicamente al momento dell'apertura della porta;*
- *le porte comprese tra* $49553 \div 65535$ *sono "porte private".*

UDP

User Datagram Protocol

- Protocollo connectionless - niente handshake a inizio connessione. Questo riduce notevolmente la congestione e rende così UDP per certi versi più veloce di TCP;
- "senza fronzoli": l'intestazione di un pacchetto UDP è lunga solo 8 Byte (contro quella di TCP che è lunga ben 20)

Header (intestazione) segmento UDP: [1]: usata per Demultiplexing;

(2 Bytes)	*(2 Bytes)*
Porta di origine	Porta di destinazione[1]
Lunghezza msg[2]	checksum[3]

[2]: indica dove finisce il messaggio;
[3]: per verificare errori nel messaggio: se i complementi a 1 sono andati a buon fine (e quindi

anche l'invio/ricezione del pacchetto), il risultato del checksum deve essere uguale a 11111111 11111111 (16 volte 1).

UDP non fa niente che non sia Multiplexing e Demultiplexing. Per la maggior parte dei protocolli dell'application layer (tipo HTTP) viene utilizzato TCP, perché TCP fornisce molti servizi legati alla stabilità e affidabilità della connessione. UDP viene usato perlopiù dove è richiesta non tanto correttezza di tutti i pacchetti ma piuttosto una alta responsiveness (quindi tempi d'attesa brevissimi) e ci si può permettere una certa percentuale di packet loss senza influire sul risultato visibile in maniera critica - quindi in servizi come streaming video o VoIP (voice over IP, ossia Skype, Discord etc., ma viste le performance di internet si sta iniziando ad usare TCP anche per questi qui).

RDT - Principi di Trasferimento Dati Affidabile

Piccola digressione in cui parleremo, in maniera un po' teorica, del sistema di trasferimento dati affidabile (Reliable Data Transfer, RDT): una serie di meccanismi da adottare per prevenire la perdita di informazione a seconda dei vari problemi che ci possono essere (ritardi, timeout, errori etc.). Questo perché TCP, per quanto premuroso possa essere come protocollo, si appoggia a protocolli a livello sottostante che non sono in grado di garantire davvero una comunicazione perfetta: qualcosa può andare male, e servono strategie per mitigare questo "male".

RDT 1.0 - Canale Affidabile

L'approccio naïve, ovvero quello basato sull'assunzione che il canale sottostante sia perfettamente affidabile. Quindi non fa nulla di contromisure.

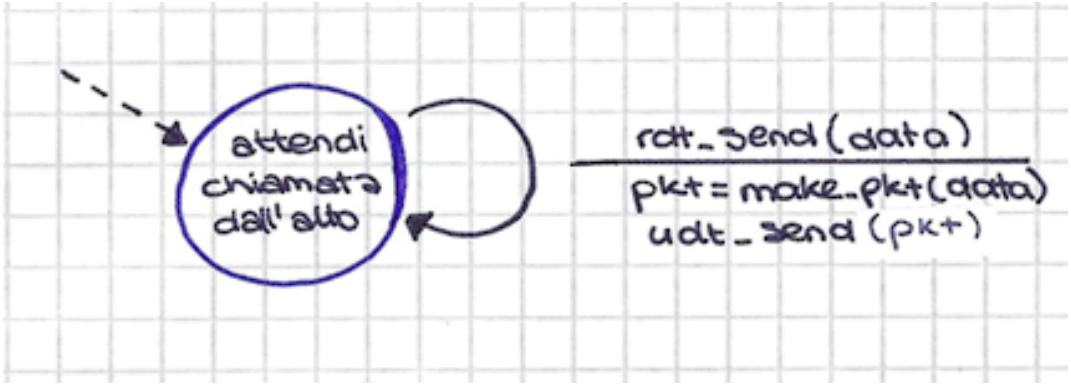


Figure 5: RDT 1.0, lato mittente. Spieghino veloce di cosa succede qui: c'è uno stato solo, in cui il mittente attende la chiamata dall'alto (ovvero dal layer sopra); quando riceve il comando "send(data)", risponde con 2 azioni - crea il pacchetto con i dati e le info necessarie, e lo invia.

Figure 6: RDT 1.0, lato ricevente. Quando viene ricevuta una chiamata "receive(data)", estrae i dati dal pacchetto (extract) e li consegna al layer sopra (deliver). Come dicevamo, nessuno dei due lati fa nient'altro.

RDT 2.0 - Rilevamento Errori

Supponiamo ora che ci possano essere errori. Dobbiamo quindi fare in modo che il lato ricevente dia un feedback (ACK, NAK). Chiamiamo questo approccio error detection-feedback-retransmission.[16]

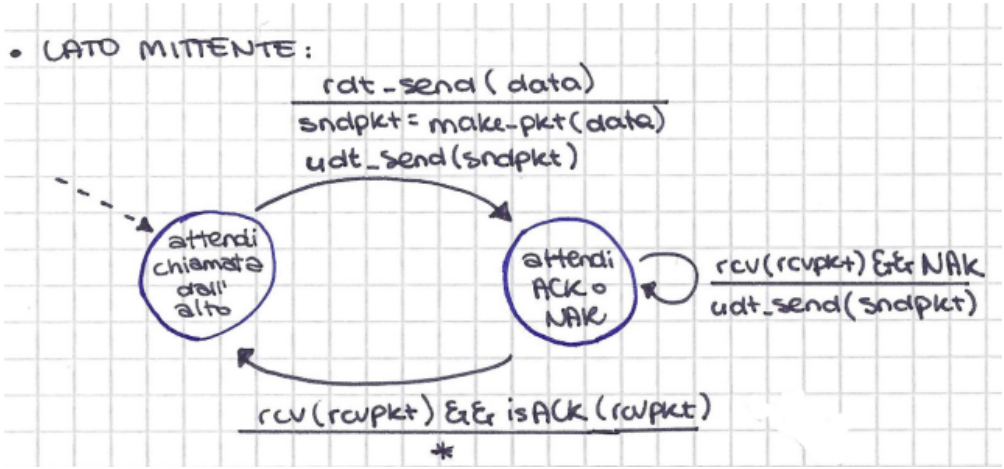


Figure 7: RDT 2.0, lato mittente. Questo ha 2 stati, uno di attesa chiamata e uno di attesa feedback: RDT riceve la chiamata "send", quindi impacchetta, spedisce il messaggio e si mette in attesa di feedback. Se arriva un NAK (ovvero errore rilevato), allora rispedisce lo stesso pacchetto di nuovo; altrimenti (ACK), torna in attesa che il layer sopra gli passi un nuovo pacchetto da inviare.

[16] La famiglia di protocolli che fanno questo tipo di cosa si chiama ARQ, Automatic Repeat reQuest (o Query).

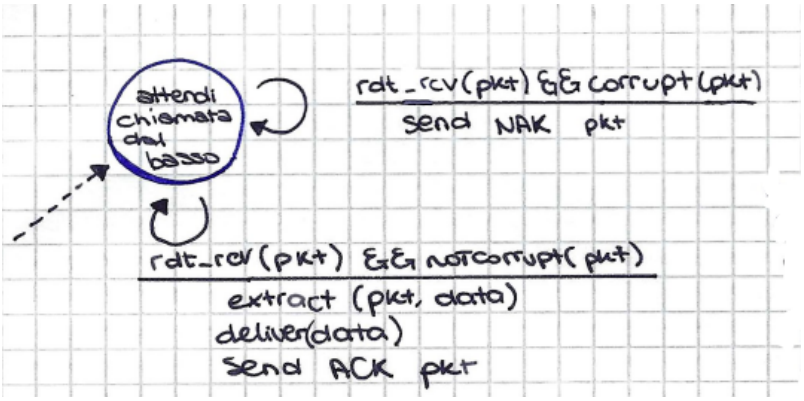


Figure 8: RDT 2.0, lato ricevente. Qui, all'arrivo di un messaggio dal layer sottostante, possono succedere 2 cose: il pacchetto non va bene (corrupt) → invia NAK in risposta; il pacchetto va bene → risponde ACK al mittente e consegna il pacchetto al layer sopra.

Semplice semplice. Ora però sorge un altro problema ancora: e se fosse il pacchetto contenente "ACK/NAK" ad essere corrotto? → soluzione: aggiungere ad ogni pacchetto un nuovo campo di informazione: il Sequence Number. Tipicamente vengono usati numeri naturali crescenti, ma è sufficiente anche alternare messaggi con SN 0 e SN 1 perché la magia funzioni.

RDT 2.1 - Sequence Numbers

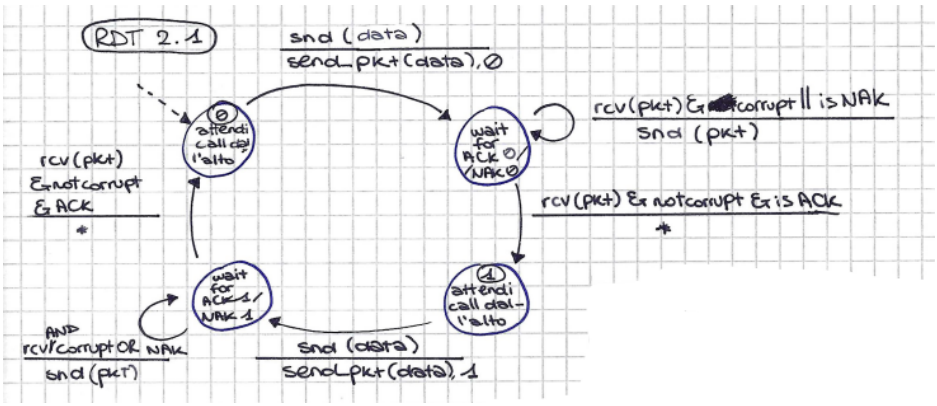


Figure 9: RDT 2.1, lato mittente: questo qui fondamentalmente dice: se il pacchetto di feedback arriva corrotto o contentente un NAK, in base allo stato dell'automa in cui mi trovo o rispedisco il pacchetto con SN $= 0$ oppure quello con SN $= 1$; in questa maniera, se il destinatario aveva capito il messaggio con SN 0 e il mittente gli manda un pacchetto con lo stesso SN, il destinatario si rende conto che è una ripetizione e non una nuova trasmissione. (E questo credo sia tutto ciò che cambia lato destinatario, spero sia la ragione per cui ho omesso l'automa del lato destinatario interamente)

RDT 2.2 e 3.0 - ACK duplicati e Time-out

In breve, RDT 2.2 introduce al posto di ACK e NAK, soltanto l'uso di ACK ma duplicati, nel senso: invece di mandare un messaggio del tipo "non ho capito questo" dice "l'ultimo messaggio che ho capito bene è quello" (identificando *questo e quello* con i sequence number dei messaggi in questione.);
RDT 3.0 introduce il meccanismo di Time-out: è possibile che un pacchetto si smarrisca per strada, quindi si fa uso di una sorta di countdown per ogni pacchetto - un timer, se vogliamo - per stabilire se e quando smettere di aspettare un riscontro.
Credo e spero che non valga la pena annettere gli automi che illustrano il funzionamento anche di questi due, perché sono sempre più densi di stati e comportamento che trovo più facile riassumere a parole. È una scelta che invecchierà male? Lo scopriremo (ma non credo).

E questi sono protocolli di tipo "Stop-and-Wait" che approcciano il problema della gestione degli errori tramite attesa di feedback o time-out; ora vedremo il GO-BACK-N, che è una famiglia di protocolli che ha a che fare con il "pipelining" - ovvero, invece di fermarsi ad aspettare feedback ogni volta che si manda un singolo pacchetto, se ne mandano n uno dietro l'altro. È indubbiamente più efficiente, ma quanti pacchetti posso mandare in sicurezza senza che un errore a un certo punto generi fatalmente *chaos & confusion*?

Go-Back-N

Per rispondere alla domanda di cui sopra, si fa uso di quella che viene chiamata "finestra scorrevole"[17]: la larghezza di questa finestra cambia dinamicamente ma non a caso, la dimensione della finestra viene adattata dal controllo di flow e congestione!

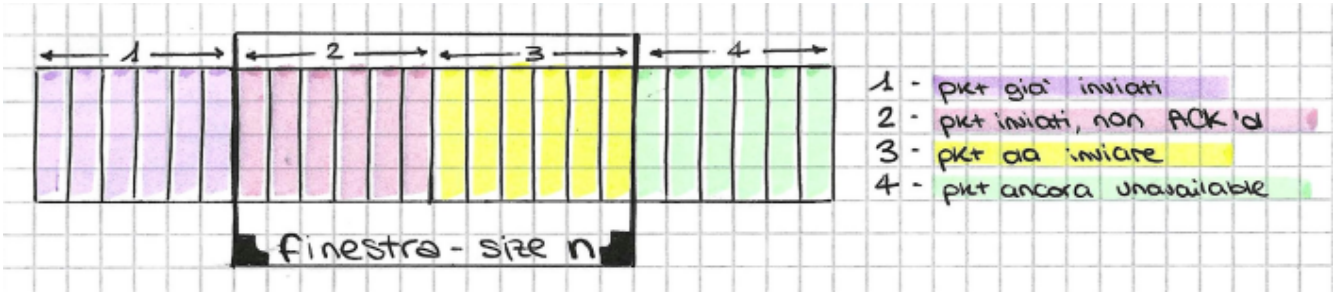


Figure 10: Quanto mi piace usare 'sti colori. Finestra scorrevole, con annessa adorabile legenda dei colori. L'ultima parola a destra nel punto 4 è "unavailable", ndr.

Riscontro cumulativo: mandare un ACK per il pacchetto numero N implica che tutti i messaggi fino ad N sono stati ricevuti correttamente $\approx$ "fino a N li ho capiti tutti". In caso di time-out, il mittente rispedisce tutti i pacchetti della zona 2 (in rosa[18]), cioè quelli inviati ma non confermati da ACK.

[17] mi pare che in inglese si parli di congestion window.
[18] pastello! :D

Selective Repeat (SR): "non ritrasmettere tutto quanto, ma solo le parti che non ho capito" (starà poi al ricevente metterli nel giusto ordine quando vengono ritrasmessi e ricevuti).

TCP

Transmission Control Protocol

> With TCP, two hosts are a company, and three are a crowd.
>
> *Una qualche edizione del Kurose-Ross*

- connection-oriented: prevede che si faccia un handshake (lit. "stretta di mano") prima di iniziare lo scambio di messaggi vero e proprio;
- sempre point-to-point: connette un solo host con un solo host, fine.

Fantastici parametri TCP e a quali sigle trovarli:

- MSS, Maximum Segment Size: limite di dati che possono essere messi in un segmento TCP. Questo parametro dipende dal (vedi punto successivo)
- MTU, Maximum Transmission Unit: dimensione massima dei dati che possono essere gestiti al livello datalink - ad esempio, Ethernet ha una MTU di 1500 Bytes;
- Flag TCP! Da ricordare. Sono bit (quindi valori di lunghezza 1 che possono essere o $= 0$ o $= 1$). Vediamole:
  - RST: sta per "ReSeT", si usa in caso di gravi errori;
  - PSH: (PUSH, credo) indica al ricevente di passare subito questo segmento al layer soprastante, senza elaborare niente;
  - URG: manca il contenuto del segmento come URGente;
  - FIN e SYN: si usano rispettivamente per indicare una chiusura e un'apertura della connessione;
  - ISN: Initial Sequence Number: numero generato in modo pseudorandomico all'avvio di una connessione TCP. È compreso tra i valori $0 \div (2^{32} - 1)$, da quel numero in poi si conteranno i sequence number (tipicamente in modo crescente, andando di successori);
  - MSL: Max Segment Lifetime, il tempo durante il cui il segmento resterà in vita nella rete. Scaduto questo tempo, il pacchetto viene soppresso;
  - CWR e ECE, le vedremo più avanti.

Queste flag appaiono nel header TCP nel seguente ordine: CWR - ECE - URG - ACK - PSH - RST - SYN - FIN;

Fattore di Time-out: ci interessa che il Time-out sia maggiore del RTT (round-trip time), naturalmente, altrimenti il pacchetto non ha nemmeno modo di arrivare a destinazione perché muore prima. In TCP, il RTT viene preso ad ogni ACK. Non può essere stabilito a priori, al limite si può stimare.

estimatedRTT:

$$eRTT = (1 - \alpha) \cdot eRTT + \alpha \cdot sampleRTT$$

α è un fattore costante, di solito $= 0,125$, per fare una media pesata di quei due parametri; $sampleRTT$ è il RTT misurato *(sampled)* per ogni andata e ritorno.

$$devRTT = (1 - \beta) \cdot devRTT + \beta \cdot (sampleRTT - eRTT)$$

con $\beta = 0,25$, questa è la deviazione standard RTT. Alla fine, il RTT calcolato avrà valori che convergono in modo abbastanza stabile, e la deviazione standard valori abbastanza bassi.

Notine di dubbia utilità di lab:

- TSHARK non è altro che Wireshark ma con una interfaccia command line anziché interfaccia grafica;
- Wireshark ha un tool per l'analisi RTT:

```
tcp.analysis.ack_rtt
```

RTO: Retransmission Time-Out: tempo entro cui la sorgente si aspetta di ricevere un riscontro. Non può essere un valore statico predefinito, dipende da moltissimi fattori. Si calcola dinamicamente (perché basato sul RTT, che si calcola dinamicamente), di solito il risultato è compreso tra $200ms < RTO < 60 + sec$:

$$RTO = eRTT + 4 \cdot devRTT$$

RcvWindow: Buffer che serve ad evitare problemi di flusso dei dati. Memorizza i byte ricevuti per poi passarli all'app layer. Quando questo buffer è pieno, il destinatario manda byte di controllo al sender per dire che non può ricevere nient'altro al momento. Appena si liberano n bytes, il destinatario[19] invia un messaggio al mittente per dire "Ok, ora ho spazio, mi aspetto messaggi a partire da $<$(ultimo messaggio ricevuto $+1 >$ e ho n byte di spazio."

[19] probabilmente per abitudine tenderò ad abbreviare mittente e destinatario con snd e rcv qualche volta (sender, receiver), sapevatelo

SYN-SYNACK-ACK, il Three-Way Handshake in TCP

È un concetto abbastanza importante. Fondamentalmente questo è ciò che va fatto prima di avviare ogni conversazione tra host TCP. È come quando al telefono ci si risponde "Pronto, chi è?" "Sono Tizio, tu sei Caio?" "Ciao Tizio, sì io sono Caio". Una fase di autenticazione preliminare, insomma. Fatto questo, si può iniziare la conversazione vera e propria.

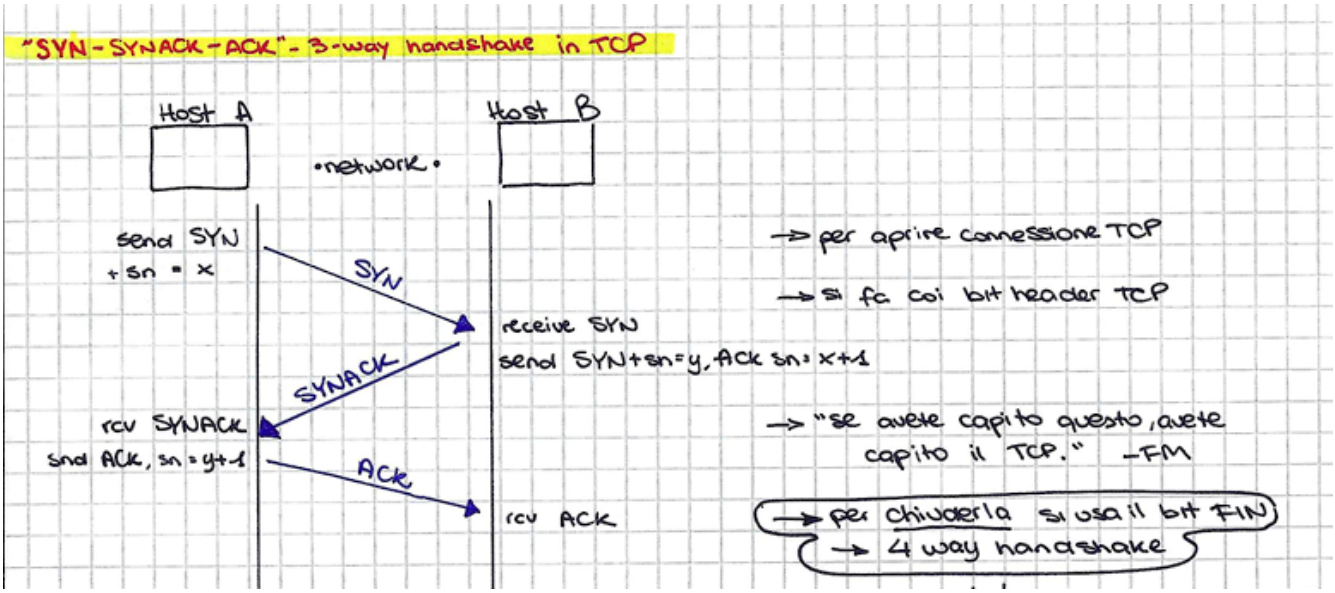


Figure 11: Three-way handshake TCP.

Finita la conversazione, per continuare la metafora della telefonata, vogliamo segnalare che abbiamo finito di parlare, quindi dire una cosa come Va bene, ora ti saluto, ciao!", a cui l'interlocutore risponde OK, ciao!" e poi si riattacca. Paradossalmente questo si fa in 4 messaggi, più di quanti non ce ne vogliano per iniziare la conversazione - ma queste sono le meraviglie di TCP. Comunque, 4-way handshake di chiusura in Fig. 12:
Controllo della congestione, nota: immettere in rete dei "pacchetti di controllo" a scopo di controllo della congestione è fortemente sconsigliato, paradossalmente congestiona esso stesso la rete.

Il discorso su prestazioni e scenari relativi è discusso in maniera molto esaustiva sia nel libro che nelle slides, you're on your own with this one >:)

Comunque ci sono due approcci principali al congestion control: End-to-End e Network-assisted. Per quanto riguarda TCP e il congestion control, occorre parlare di finestra di congestione.
congWindow: finestra di congestione. Quantità di dati riscontrati da un host durante la connessione. Vincolo: TCP non può inviare dati ad un rate maggiore della ampiezza della finestra di congestione, e questa finestra, come ha senso che sia, cambia ampiezza dinamicamente a seconda del livello corrente di congestione. Per cambiare il ritmo di invio in funzione della congestione si usano diversi approcci (io qui li ho chiamati Algoritmi, non so se sia accurato come termine,

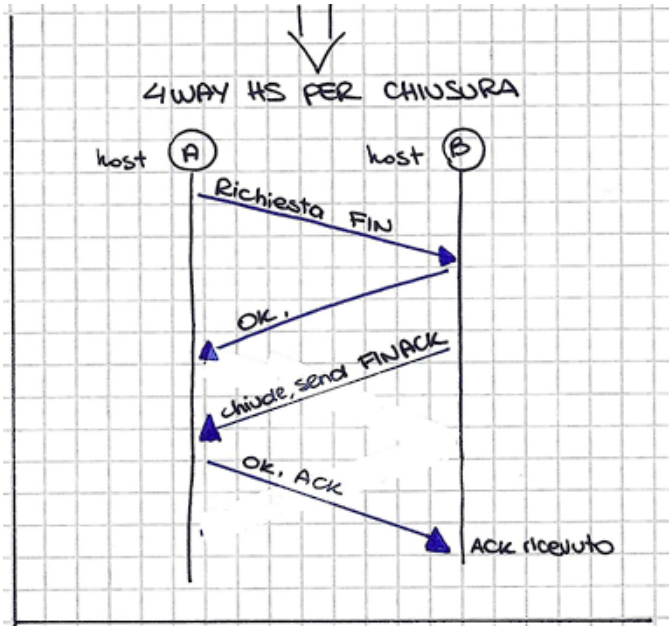


Figure 12: Handshake di chiusura TCP.

ma penso stessi facendo riferimento ai TCP Tahoe e Reno):

- AIMD: incremento additivo, decremento moltiplicativo;
- Slow Start: pre ogni ACK ricevuto, raddoppio la dim. della finestra;
- Fast Recovery: questo non lo vedremo, rip

Threshold (soglia): limite tra decremento moltiplicativo e incremento additivo: se il valore corrente è $< Thr \rightarrow$ si adotta slow start; se il valore corrente è $> Thr \rightarrow$ si adotta congestion avoidance (AIMD). Nel modulo di tutorato si vedono meglio questi discorsi con TCP Reno e compangia. Comunque, l'idea è che l'algoritmo vada a convergere e stabilizzarsi su un certo range di rata di trasmissione, anche se nella pratica poi non va così (vedasi TCP CUBIC, non ho nessuna memoria di cosa sia questa roba).

Note di Lab:

Nmap, network scanner! 2 tecniche principali:

- PORTSCAN: attività promiscua (losca, sospetta), da usare solo previa autorizzazione se non vogliamo cacciarci in qualche guaio, perché il port scanning di solito è una tecnica utilizzata per scoprire porte aperte in altri dispositivi, al fine di sfruttarne le vulnerabilità e provare qualche attacco. Port scanning aiuta un malintenzionato a trovare porte aperte e a capire se queste sono in ascolto o stanno trasmettendo dati. Inoltre, può rivelare misure di sicurezza (come un firewall) sono in uso in qualche rete aziendale[20];

[20] fonte Qui

- PING SWEEP: port monitoring, analisi di sicurezza, information gathering, attacchi, beh suona abbastanza simile al port scanning. Il ping sweeping consiste in mandare messaggi di ping a più indirizzi IP per trovare degli host "vivi" (insomma accesi) nella rete, da cui poi si procede col port scanning sugli host vivi.[21]

nmap utilizza delle "fingerprints" (impronte digitali) per raccogliere informazioni su di un target. Eventualmente è possibile utilizzarlo con interfaccia grafica (che si chiama Zenmap), ma *il prof, con un velo di sarcasmo, ci fa sapere che "Zenmap è per quegli utenti che usano il Mac".* Nmap ha incorporato uno script engine in LUA, ovvero una collezione di script che si possono eseguire per trovare vulnerabilità e affini. Basta eseguire il comando

```
nmap --script-updatedb
```

per aggiornare questa fantastica lista di script curata dagli eroi invisibili della rete.
ncat invece è un tool open source a riga di comando utile a collegarsi ad un altro host da remoto, pretty cool. Tutte le info che abbiamo visto di ncat si trovano nelle slides del prof - a questo indirizzo, nella tabella "didattica" $>$ Internet, Reti e Sicurezza $>$ Esercitazioni e Laboratorio Wireshark al nome "nmap.pdf".

[21] Se volete divertirvi a leggere/guardare, ho trovato questo

Network Layer

(Layer 3 nello stack TCP/IP)

Per iniziare, il Network Layer può essere diviso in due macro-componenti che interagiscono fra di loro: Data Plane e Control Plane (dove Plane significa piano, strato). A questo layer dello stack, la PDU si chiama Datagramma.

Nel Data Plane vengono trattate tutte le funzioni "pre-router", incluso il tradizionale IP Forwarding che vedremo a breve.

Forwarding VS Routing:

- Forwarding: azione svolta dal router, consiste nel trasferire un pacchetto da una input link interface alla opportuna output link interface;

- Routing: azione svolta a livello di rete, comprende TUTTO il processo per determinare il percorso (path) che il pacchetto percorre da mittente al destinatario.

N.B.: i protocolli di routing $\neq$ protocolli routabili ($\approx$ instradabili[22]). Also, in internet non esistono (e non possono esistere) due indirizzi uguali corrispondenti a due diversi Host.

IP

Internet Protocol

Definisce il modo in cui i pacchetti sono instradati.

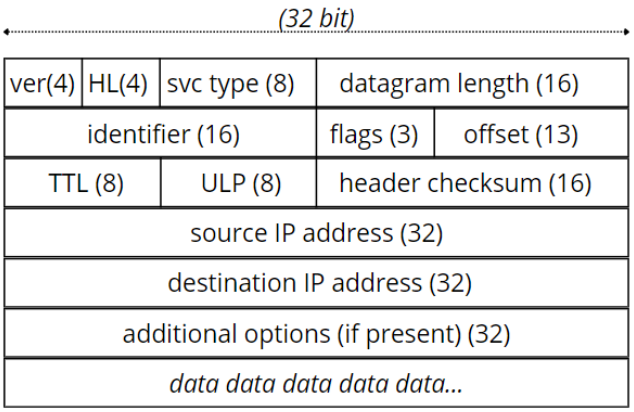


Figure 13: Struttura di un header IPv4. Tra parentesi sono indicati i bit di lunghezza di ciascun campo.

Datagramma IPv4

In Figura 13 è raffigurata la struttura di un header (intestazione) a livello di rete, nello specifico un header IP, versione 4 (esiste anche IPv6, lo vedremo brevemente più avanti). Le varie sigle stanno per:

- ver: *versione;*
- HL: *header length*, lunghezza header;
- svc type: *service type;*
- TTL: *Time-To-Live;*
- ULP: *Upper Layer Protocol,* ovvero che protocollo aspettarsi al layer immediatamente superiore (i.e., il layer di trasporto, quindi solitamente TCP o UDP);

N.B.: il time-to-live, sebbene esprima un concetto di "tempo", non è un valore da intendersi come un timer di ore e minuti, una cosa tipo *hh:mm:ss*: è una unità di tempo *logica*, ovvero un contatore a n che viene decrementato di 1 ogni volta che un pacchetto arriva ad un router[23]. Se il TTL arriva a $= 0$, allora il pacchetto viene dichiarato morto: questo per evitare che i pacchetti entrino per errore in circolo all'infinito tra i vari router della rete.

IPv6, come datagramma, è molto simile a IPv4 ma con qualche differenza, la più rilevante credo sia nelle dimensioni dell'indirizzo - non più 32 ma 128 bit - ed altre differenze nei contenuti.[24]

[22]protocollo utilizzato per mandare dati attraverso più reti, selezionando i formati dati opportuni in base alle reti in questione. Tipo IP. IP non decide nulla a livello di percorsi attraverso la rete (non è un protocollo di routing, ma routabile).

[23]Per capirci, invece di contare quanto tempo impiegate per arrivare dalla porta del bagno a quella della cucina, contate i passi che fate. I passi, coi piedi, i passi fisici.

[24]Da qualche parte che non so dirvi al momento, esistono tools e metodi per convertire un indirizzo IPv4 in IPv6 e viceversa. Penso basti andarseli a cercare sul web.

Interfaccia di rete: punto di connessione tra host e router (e.g.: una scheda Bluetooth, Ethernet, etc.). Ciascuna interfaccia di rete può avere più IP Address associati.

Evoluzione degli Indirizzamenti

Ovvero, piccolo excursus storico.

- 1981 - Indirizzamento a 2 livelli classful: semplice da comprendere e implementare, questo tipo di addressing architecture si basa sulla divisione in 5 classi di indirizzi, basate sui primi 4 bit dell'indirizzo IPv4. In questa architettura 2-layer classful, la struttura dell'indirizzo IP era:

  ```
  (network_ID).(host_ID)
  ```

  la prima metà dell'indirizzo identificava la rete, la seconda metà l'host. Quindi la classe (identificata nei primi 4 bit a sinistra) è associata alla rete!

classe	start address (decimale e binario)
classe A	0.0.0.0 00000000.00000000. (...)
classe B	128.0.0.0 1000000.00000000. (...)
classe C	192.0.0.0 11000000.00000000. (...)
classe D *(multicast)*	224.0.0.0 11100000.00000000. (...)
classe E *(reserved)*	240.0.0.0 11110000.00000000. (...)

  Questo sistema 2-layer classful è deprecato, non più in uso, ma talvolta viene chiesto all'esame, quindi il mio consiglio spassionato è: esercitatevi a convertire i numeri dal decimale al binario o memorizzate i numeri che delimitano le classi (0, 128, 192, 224 e 240. Gli intervalli, naturalmente, sono $0 \div 127$, $128 \div 191$, $192 \div 223$, $224 \div 239$ e $240 \div 255$.);

- 1984 - Indirizzamento a 3 livelli classful: la struttura dell'indirizzo non è più:

  ```
  (network_ID).(host_ID)
  ```

  ma cambia in:

```
(network_ID).(subnet_ID).(host_ID)
```

Il subnetting è un argomento che vedremo tra un attimo;

- 1993 - CIDR (Classless Inter-Domaining Routing): viene eliminata la divisione in classi; indirizzi vengono gestiti in modo efficiente per fare routing a questa maniera

  ```
  IP = < prefisso, suffisso >
  ```

  dove il prefisso indica la rete e il suffisso indica l'host connesso alla rete. È una forma di subnetting, la dimensione di questi due campi varia arbitrariamente! Per risalire a dove inizia uno e dove finisce l'altro, per distinguerli tra di loro occorre usare un'altra stringa di bit chiamata maschera di rete (netmask): la netmask è una stringa di 32 bit disposti in una sequenza di x bit "1" seguiti da y volte "0" (ovviamente, $x + y$ deve tornare 32), tipo:

  ```
  11111111.11111111.11111111.00000000
  ```

  questo è il caso di una netmask /24 (cosiddetta *barra 24,* ovvero, 24 bit '1' seguiti da 8 bit '0'). Facendo l'operazione di AND logico tra la netmask e l'indirizzo CIDR, possiamo separare la parte di rete dalla parte di host (rispettivamente nel prefisso e suffisso).

Indirizzi IP da ricordare (particolari, riservati a questo scopo unico):

- 0.0.0.0 : indirizzo di avvio stack TCP/IP;
- 127.0.0.1 : indirizzo di loopback a localhost. Permette di comunicare con la propria stessa macchina come se fosse un altro host in rete (per fare una metafora personale, è come parlare alla propria immagine riflessa, il messaggio è indirizzato a me stess* e *rimbalza* sulla superficie dello specchio tornando a me). Perché dovrei voler comunicare con me stesso? Boh, per fare test;
- $< net\_ID >$ seguito da tutti 1 : indirizzo di Broadcast. Manda pacchetti a TUTTA la rete contrassegnata dall'indirizzo $< net\_ID >$;
- $< net\_ID >$ seguito da tutti 0 : indirizzo di rete (o sottorete), non identifica un host.
- 255.255.255.255 (in binario, sono 32 volte 1) : broadcast locale.

Subnetting

Ovvero, dividere logicamente (non fisicamente) la rete in sotto-reti da TOT indirizzi ciascuna usando maschere di rete.

(prefisso di rete)		
net_ID	subnet_ID	host_ID

Quanto è lungo il prefisso di rete? Dipende: in caso di subnetting statico, ha lunghezza fissa; nel caso di subnetting dinamico (Variable Length Subnet Mask, VLSM), ha lunghezza variabile.[25]

Esempio: risaliamo alla rete, dato l'indirizzo

```
193.205.92.150/25
```

Per risalire alla rete, facciamo un AND logico con la netmask a barra 25 (cioè con 25 volte 1 seguìti da 7 volte 0):

```
11000001.11001101.01011100.10010110
```

(∧)

```
11111111.11111111.11111111.10000000 =

11000001.11001101.01011100.10000000
```

che, tradotto da binario a decimale, sarebbe

```
193.205.92.128
```

Ci verrà anche richiesto il procedimento opposto, allo scopo di progettare reti e sottoreti: partendo da un indirizzo e una barra, determinare quante e quali sottoreti assegnare (e per ciascuna, quanti host si possono assegnare).

Alle volte è buona pratica assegnare la barra di sottorete un numero più alta del minimo indispensabile (cioè raddoppiare gli host disponibili, che è ciò che succede se si alloca un altro bit agli host): ad esempio se abbiamo 62 host già assegnati, è meglio non andare per una netmask a /24 (che offre 64 indirizzi, solo 2 in più di quelli che abbiamo già), altrimenti appena ne vogliamo aggiungere altri 3 quella sottorete non ci basterà più! E tocca riconfigurare tutto da capo. Quindi meglio una /23. Accontentatevi di una /24 solo nel caso in cui l'esercizio dice "sappiamo già che non intendono apportare espansioni alla rete in futuro".

[25] Potete usare questo tool online per verificare la correttezza delle vostre subnet quando fate esercizi di subnetting :)

Datalink e LAN

LAN

Local Area Network
LAN, Definizione del IEEE: Sistema di comunicazione che permette ad apparecchiature indipendenti di comunicare tra loro entro un'area delimitata, utilizzando un canale fisico ad elevata velocità e con basso tasso d'errore.

- tipicamente, non sono trasmissioni di dati continue, ma a "burst", cioè a intervalli, a raffiche discontinue;
- tutte le macchine della LAN condividono lo stesso canale fisico di comunicazione;
- è una rete...
  - economica;
  - facile da modificare;
  - di facile manutenzione;
  - capace di sopportare grossi carichi di dati;
  - duratura nel tempo (anni, se ben progettata e configurata).

Aspetti meno piacevoli e delicati della LAN:

- tutti gli host collegati devono essere identificati;
- bisogna stabilire delle regole per permettere agli host di comunicare tra di loro;
- flessibilità - compatibilità tra host di natura diversa (pensate ad un dispositivo mobile, un Desktop Computer, un sensore);
- modularità;
- espandibilità;
- gestibilità (vedere SNMP, protocollo Applicativo visto nel capitolo Application Layer);

Cosa ci serve in una LAN?

- beh, per iniziare, uno o più Host;
- software di rete (non necessariamente associato al sistema operativo);
- le NIC(Network Interface Card, che possono essere cablate o wireless). Le schede di rete, fondamentalmente;
- delle API (Application Programming Interface), per interfacciarsi con il software;
- dei Network Hub, o Concentratori (e.g.: switch);
- cablaggio strutturato (cavi, antenne, etc.).

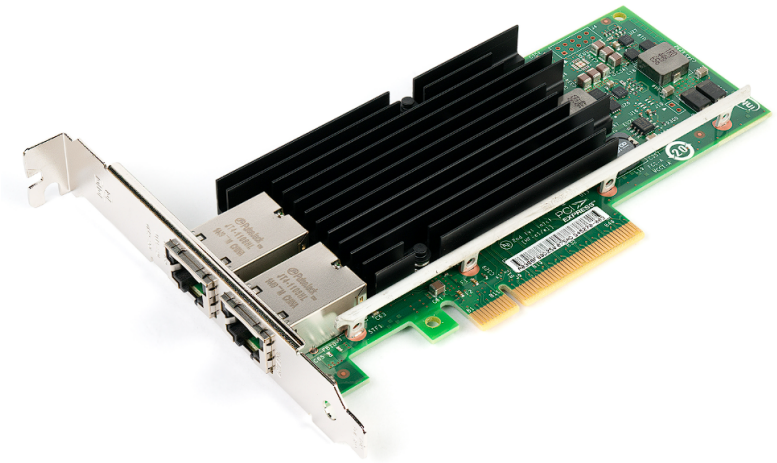


Figure 14: Una scheda di rete (NIC).

Le specifiche della LAN sono definite nello standard EIA/TIA 568 e ISO/EIC 11801: nello specifico, ISO/EIC 11801 definisce le specifiche del cablaggio standard. In genere, la sigla che volete ricordare è EIA/TIA 568.
IEEE 802.11 (anche questo va ricordato perché è uno standard importante) è il working group di IEEE che si occupa di gestire lo standard Wireless LAN; IEEE 802.3 invece si occupa dello standard Ethernet.

LAN di solito è divisa in 2 macro categorie: Hardware (topologia, cavi, antenne tutto ciò che la rende funzionante sia per via cablata che wireless) e Software (tutti i protocolli e i software applicativi).

Datalink

Ovvero, il layer 2 dello stack TCP/IP.
A questo layer si riceve il messaggio dal layer di rete (quindi molto probabilmente si tratta di un messaggio IP), si struttura il messaggio al solito e lo si passa al layer sottostante (layer fisico, 1 nello stack). Con "struttura il messaggio" si intende questo: il messaggio IP viene suddiviso in frame delle dimensioni richieste dal layer sottostante, e gli viene aggiunta una FCS (Frame Check Sequence).

Il layer Datalink si divide in:

- LLC : Logical Link Control;

- MAC : Medium Access Control. Quest'ultimo è quello che si occupa di reperire informazioni su che tipo di mezzo fisico c'è al layer 1; funge anche da "arbitro" nella gestione del canale, perché nel momento in cui in rete si comunica con un canale unico diviso in broadcast sorge naturalmente il problema: come regolamentiamo l'accesso a questo canale? Occorre inventarsi un sistema che:
  - Trovi gli indirizzi di tutti gli host connessi alla rete;
  - Trovi, per ciascuno di questi indirizzi, il corrispondente indirizzo MAC (ora ci arriviamo, al MAC).

La PDU del Datalink Layer: il Frame

Un frame MAC contiene le seguenti informazioni:

- DSAP/SSAP: destination/source SAP (Service Access Point), sono i campi principali del frame, e sono indirizzi univoci a livello mondiale;
- payload: il corpo del messaggio da trasmettere;
- FCS: Frame Check Sequence, è un CRC[26] su 32 bit, un sistema di controllo integrità tipo checksum.

Frame Ethernet: come di consueto, c'è una intestazione a cui segue il payload.
Nell'header c'è quello che si chiama un preambolo: una sequenza di 7 byte tutti composti da 010101010101... (lungo $7 \cdot 8 = 56$ bit), seguito da un 8^o byte che, a differenza dei primi 7, è 01010111 (quindi termina con '11'): il preambolo serve ad annunciare nel broadcast "OK, questo è il segnale che adesso parlo io, quello che segue è il mio messaggio". Seguono gli indirizzi del destinatario e del mittente, poi 2 byte che specificano la lunghezza del messaggio, poi il payload e il CRC.

Indirizzo MAC

Lungo 48 bit (6 byte) formattati in 6 coppie esadecimali:

```
08:00:2b:3c:07:9a
```

- i primi 3 byte (nell'esempio, 08:00:2b) identificano il Vendor Code (detto anche OUI, Organization Unique Identifier). Standardizzati dal IEEE, sono codici associati ai produttori di schede di rete (e.g. Cisco);
- gli ultimi 3 byte (nell'esempio, 3c:07:9a) sono una numerazione progressiva decisa dai produttori. Questi identificano la scheda di rete $\approx$ *"questa è la scheda Wireless n° 101 prodotta da Cisco"*(è unica, come un numero di serie);

[26] Controllo a Ridondanza Ciclica

La scheda di rete può essere divisa in 2 componenti:

- hardware (interfaccia di rete) (immagino qui mi riferissi alle cose come la porta Ethernet o l'ingresso per il cavo coassiale, le periferiche insomma);
- CPU + memoria - queste lavorano indipendentemente dal resto del PC, non elaborano dati da inviare al processore!

L'indirizzo fisico deve essere unico nella LAN! In Internet ci possono essere duplicati, ma in una rete locale non è ammesso. Normalmente, gli indirizzi fisici sono statici, tuttavia alle volte possono essere riassegnati.

L'indirizzo fisico può rappresentare:

- UNICAST : un singolo host;
- MULTICAST : un gruppo di host;
- BROADCAST : tutte le stazioni (analogamente all'indirizzo IPv4, il MAC broadcast è della forma ff:ff:ff:ff:ff:ff, il più alto indirizzo possibile come 255.255.255.255). Nota: nel caso di un indirizzo broadcast, di norma il frame viene sempre analizzato (immagino per motivi di sicurezza);

Gli indirizzi di gruppo servono principalmente per fare neighbor discovery(raccogliere info su chi altro è connesso alla rete). Vengono usati secondo due modalità di impiego:

- solicitation: la stazione richiede un servizio e manda un messaggio multicast con l'indirizzo del servizio; le stazioni che offrono quel servizio, risponderanno ($\approx$ *"mi serve questo, chi ce l'ha?"*);
- discovery: le stazioni che offrono un servizio inviano, a cadenza regolare, un messaggio multicast per informare del servizio offerto ($\approx$ *"ho questo, a chi serve?"*)

Comandi per neighbor discovery:

```
    Linux terminal : ip neigh show
Windows PowerShell : Get-NetNeighbor
```

Esiste un protocollo di Network Discovery (chiamato appunto NDP), ma è configurato per IPv6.
Importante da ricordare: all'interno della LAN si comunica con indirizzi MAC, non IP!
Protocolli ARP e RARP: stanno, rispettivamente, per Address Resolution Protocol e Reverse Address Resolution Protocol; ARP traduce l'indirizzo IP in indirizzo MAC, RARP passa da MAC a IP.

IP e la Frammentazione dei Pacchetti

Nell'intestazione IP, ricordiamo, sono presenti alcuni campi come:

- PROTOCOL (8 bit) : è l'informazione chiave che permette la lettura del payload - ovviamente, è quella che specifica che protocollo usare;
- HEADER CHECKSUM : parity check, serve a controllare l'integrità del messaggio. Si prendono 16 bit dell'header e si fa il completamento a 1 della somma di tutti i 16 bit dell'header. Se tutto è corretto, il risultato è composto da tutti bit a 1
- il 2° blocco di 32 bit dell'header IP è riservato alla frammentazione: identificativo, flags, fragment offset. La frammentazione è quel procedimento in cui un frame a livello datalink viene, appunto, frammentato in blocchi delle dimensioni richieste dal mezzo fisico di trasmissione sottostante (specificato nel MTU, Max Transmission Unit). Il riassemblaggio del frame viene effettuato solo una volta che questo ha raggiunto la destinazione, nonostante lungo il tragitto potrebbe passare per dei canali che ammettono delle MTU più grandi. Campi di frammentazione:
  - primi 16 bit (di 32): identificativo;
  - 3 bit seguenti: flags. Queste sono:
    1. Bit riservato;
    2. Bit che, se posto a 1, significa "non frammentabile" (errore ICMP);
    3. Bit che, se posto a 0, significa "questo frammento è l'ultimo (o l'unico) del datagramma);
  - 13 bit finali : offset di frammentazione.

Problematiche:

- maggiore overhead (impiego di risorse non necessarie) di trasmissione;
- con la frammentazione è facile orchestrare attacchi DoS, mandando tantissimi pacchetti che costringono l'Host vittima ad impiegare molte risorse;

Questa funzionalità, propria di IPv4, in IPv6 non è presente.

Note di Lab: esistono modi per calcolare la MTU più piccola possibile, dato un certo percorso di rete! Si può provare ad inserire in Wireshark il comando:

```
ping -f -l 193.205.92.2 // o qualche IP address, immagino
```

Fun fact: c'è qualche personaggio, appassionato di videogames e smacchinamenti di rete, che da qualche parte in qualche impostazione che non so, va a cambiare manualmente la dimensione di frammentazione dei PDU Ethernet da 1500 a 1473 (numero stranamente specifico, mah), questo perché sostengono che in questo modo si ottiene una velocità più elevata di navigazione, e questo aiuta a ridurre il ping in ms quando si gioca online. A detta del prof cambia poco e niente. (Beh,

1500 − 1473 non è che sia chissà che miglioramento, in effetti)

Dal momento che ogni router attraversato dal pacchetto ne modifica il Time-To-Live (che è un campo nel header IP), ogni volta va anche modificato il checksum (altrimenti se un campo cambia i conti non tornano più nella verifica. Che sbatta.)

Note di Lab: per vedere su Wireshark la frammentazione, bisogna assicurarsi che nelle Preferenze il campo "Reassemble..." non sia selezionato. (Questo campo si trova sotto Protocols > IPv4.)

Routing - Instradamento

Ovvero, tecniche ed algoritmi per fare in modo che i pacchetti arrivino a destinazione. ($\neq$ inoltro)
N.B.: il router opera al layer 3 (IP).

Tabella di instradamento: è una specie di "database" memorizzato in un router o host; contiene le metriche dei costi, in termini di tempo, per poter valutare quale rete conviene attraversare da un host all'altro (come un GPS seleziona il percorso più breve, date location di partenza e destinazione).

Distinzione tra Routing e Forwarding:

- Routing: insieme di regole per popolare queste tabelle;
- Forwarding: regole con le quali il pacchetto viene inviato a determinate porte di uscita del router (tipicamente questo viene fatto consultando le tabelle già popolate).

Per stampare la tabella di routing, da terminale Windows PowerShell c'è il comando:

```
> route print
```

Nelle tabelle di instradamento, per ogni sottorete, è elencato:

- il relativo Network ID;
- l'indirizzo del router di inoltro;

Detta nel gergo dei database, i Record (i valori inseriti, i dati) nelle tabelle sono composti da:

- indirizzo della rete di destinazione;
- maschera di rete;
- l'interfaccia su cui inoltrare;

- l'indirizzo del next hop (prossimo router da ragggiungere).

Ad ogni percorso è associata una metrica, ovvero un costo che può essere un'unità di tempo approssimato, oppure un numero di hop, dipende dal protocollo utilizzato.

Per il routing si usano 2 procedimenti:

- Diretto: se l'host mittente è nella stessa rete del destinatario, inoltro direttamente al destinatario;
- Indiretto: gli host sono in due reti diverse, quindi dovrò inoltrare il messaggio ad uno o più router intermedi (quindi ci sarà almeno un next hop);

Domanda, però: come faccio a sapere se due host sono o meno nella stessa rete?
Tramite gli indirizzi IP degli host, risalgo agli indirizzi delle reti a cui i due rispettivamente appartengono: se i valori estratti corrispondono, allora si può procedere con instradamento diretto.

Per il routing diretto si usa il MAC: coinvolge i layer 1 e 2 della stack;
Per il routing indiretto si usa l'indirizzo IP del router, quindi i layer coinvolti della stack sono 1, 2 e anche 3 (IP).

La tabella di routing è *sempre* presente in tutte le macchine che operano con IP, che siano Host o Router. Di solito la tabella contiene sempre almeno il next hop migliore, assieme ad altre cose. Il next hop è configurato in una sola direzione, tant'è che i percorsi di andata e ritorno sono asimmetrici di solito - la via del ritorno potrebbe prendere altre strade per arrivare al punto di partenza.

Tipi di rotte:

- statiche : configurabili dall'amministratore di rete;
- dinamiche : da reperire tramite protocollo di routing;
- dirette : legate alle interfacce del router.

Nota: si consideri la Figura 15: quegli indicatori ".1" e ".2" stanno a specificare l'interfaccia di rete. Normalmente i router ne hanno più di una, quindi è buona abitudine specificare a quale di queste si fa riferimento ogni volta.

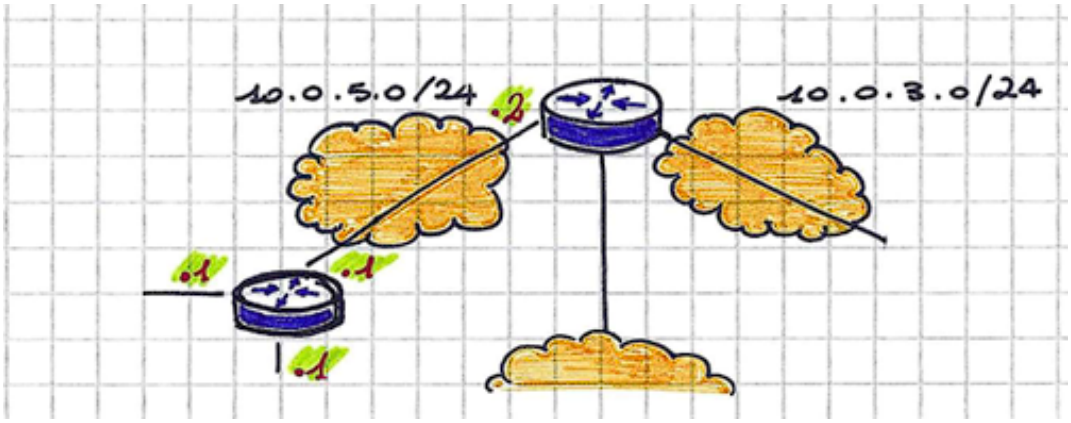


Figure 15: Dettaglio di un grafico di reti. Disegnato a mano da Yours Truly.

La tabella di routing ha un aspetto del genere (stavolta fatta a mano perché è molto più carino spiegare ciascun campo così) (vedasi Fig. 16):

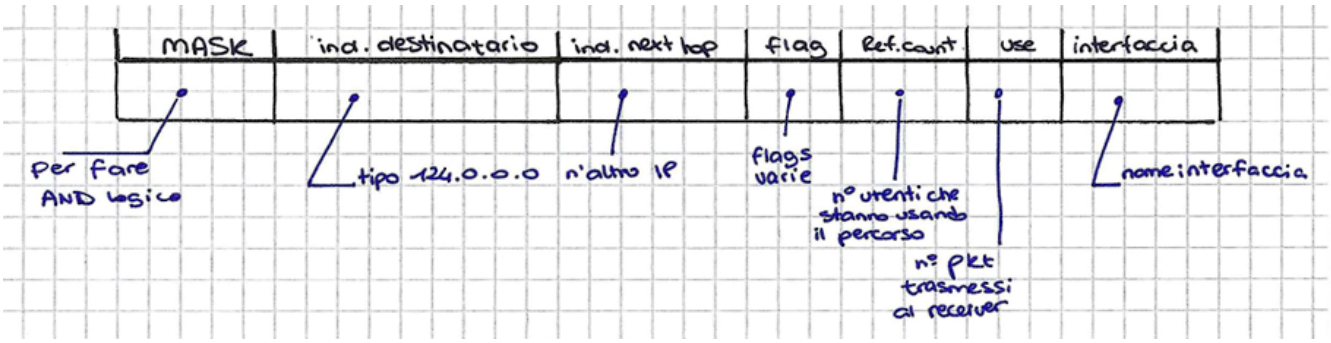


Figure 16: Dettaglio di un record di una tabella di routing. Di nuovo, cortesia di Yours Truly.

ICMP

Internet Control Message Protocol
incluso nello stack TCP/IP, è richiesto per qualsiasi implementazione standard di IP.

Fondamentalmente viene usato da IP per inviare messaggi di errore (per farlo, ICMP usa a sua volta IP. Meta! :D).

Header ICMP (32 bit)

type (8 bit)	code (8 bit)	ICMP header checksum (16 bit)

"type" specifica il formato, è un valore numerico associato a qualche informazione (e.g.: '0' = echo reply; '4' = source quench; '5' = redirect); "code" specifica il tipo di errore; il checksum è,

di nuovo, il complemento a 1 di informazioni dell'header (IP, credo).

I messaggi sono di 2 tipi, fondamentalmente: segnalazione errori e richiesta informazioni.

Quando si usa il comando "ping", quello che succede è che viene creato un pacchetto ICMP di tipo "8", la risposta che arriva è un pacchetto ICMP di tipo "1".

Un altro tipo di messaggio era quello del "timestamp", serviva ad ottenere data e ora esatta segnata da certi host: ora questo servizio non si usa più perché tutti gli host usano il protocollo NTP (Network Time Protocol), che sincronizza gli orologi di tutti gli host al mondo con un orologio di riferimento.

Come funziona il Traceroute:

- il primo pacchetto è un messaggio ICMP con un TTL settato a 1: arriva al router immediatamente confinante, al che il TTL viene decrementato a 0 che innesca un messaggio ICMP di errore per TTL giunto a termine;
- il pacchetto seguente è di nuovo un messaggio ICMP con TTL settato a $1 + 1 = 2$: arriva al router immediatamente dopo, e si ripete la cosa del messaggio di errore;
- il processo viene ripetuto iterativamente con TTL incrementali così, in modo da ottenere info da tutti i router nel percorso. Fatto il traceroute :D

Protocolli di Routing AS (Autonomous Systems)[27]:

- Protocolli inter-AS (esterni): BGP;
- Protocolli intra-AS (interni):
  - distance vector:
    - RIP, RIP2;
    - IGRP, EIGRP;
  - link state:
    - OSPF, OSPF2;
    - IS-IS;

È importante che si sappia bene la differenza tra protocolli distance vector e protocolli link state, perché hanno degli scopi diversi!! I distance vector servono a trovare il percorso migliore tra router dato il numero di hop, i link state servono a scoprire com'è fatta la rete! Con i link state scopriamo tutte le rotte disponibili, con i distance vector scegliamo quella ottimale. Uno è puramente esplorativo, uno concerne l'ottimizzazione.

[27] ASBR (Autonomous System Border Router), router che tra le altre cose deve avere un'istanza sia dei router interni alla rete che di quelli esterni.

Protocolli Distance Vector

Vengono utilizzati per trovare il percorso migliore nella rete, considerando come unità di misura il numero di hop tra un router e l'altro.

Fondamentalmente, ogni x secondi, il router invia ai suoi vicini la propria routing table.

Algoritmo RIP

Routing Information Protocol

Differenza tra RIP e RIP 2: RIP era classful, RIP2 invece usa maschera (quindi è classless).

1. Consideriamo un router che al momento ha una tabella di routing così fatta:

Routing Table

rete raggiungibile	n. Hop necessari	router di origine:
Rete 1	7	A
Rete 2	2	C
Rete 6	8	F
Rete 8	4	E
Rete 9	4	F

2. Il router riceve un messaggo da un certo Router C contenente la tabella di routing di C:

Routing Table di C

rete raggiungibile	n. Hop necessari
Rete 2	4
Rete 3	8
Rete 6	4
Rete 8	3
Rete 9	5

3. Il router prende queste informazioni, aumenta subito di 1 tutti gli hop dal Router C (l'hop che stiamo incrementando è il passo che serve per andare da questo router al router C);

4. La tabella ricevuta e incrementata viene ora comparata con quella che il router ha già:

   - se la prima ha meno hop per raggiungere qualche rete, aggiorno quel campo con quello di C;
   - se alcuni campi non erano presenti, vengono aggiunti (ad esempio, attraverso il router C ora possiamo raggiungere Rete 3);

- se il numero di Hop è lo stesso, è indifferente se aggiorniamo il campo della nostra tabella o no;
- se il numero di Hop è maggiore di quello che ho in tabella, non aggiorno la tabella con la nuova informazione;

Routing Table aggiornata

rete raggiungibile	n. Hop necessari	router di origine:
Rete 1	7	A
Rete 2	5	C
Rete 3	9 (nuovo)	C
Rete 6	5 (sostituito 8F)	C
Rete 8	4 (uguale)	C (oppure E)
Rete 9	4 (non aggiornato, $6 > 4$)	F

Table 2: Tabella di routing aggiornata. Nota: il valore di Rete 2 era 2 e proveniente da C, ma è stato aggiornato perché ora da C si arriva a Rete 2 con 5 hop.

5. e questo è RIP.

Algoritmi di Instradamento

O quella che il prof chiamò: Breve Carrellata di Algoritmi di Instradamento.

Il compito del livello di rete è di trasportare i pacchetti da un indirizzo di origine a un indirizzo di destinazione, ma non spetta a protocolli come IP occuparsi di come questo avviene fatto nella rete - di questo si occupano i Router! Di quale percorso far fare ai dati.

Riguardo il Forwarding diretto: all'interno dello stesso mezzo fisico possono esserci più reti a livello logico (pensate al subnetting con netmask): è compito del router occuparsi dell'instradamento anche lì tra quelle reti. In casi del genere è preferibile utilizzare una sola mega-rete che corrisponde alla topologia fisica. In casi quali? Immagino pensassi a casi in cui si fa spesso comunicazione interna, piuttosto che attraverso internet, non lo so sinceramente.

Protocollo Routabile (instradabile): protocollo che può essere utilizzato per applicare algoritmi di routing. Routing e Forwarding, utilizzati insieme, sono necessari per l'operatività di una rete. Informazione fondamentale a tale scopo è la tabella di routing.

Su tutte le macchine in cui è operativa la stack TCP/IP, siano essi End System o Router etc., è presente una tabella di routing ed esiste almeno un protocollo di routing.

Ciascun host è sempre collegato ad un default router, detto anche default gateway, detto anche first-hop router, detto anche router di primo rilancio, che comunica con la rete esterna.

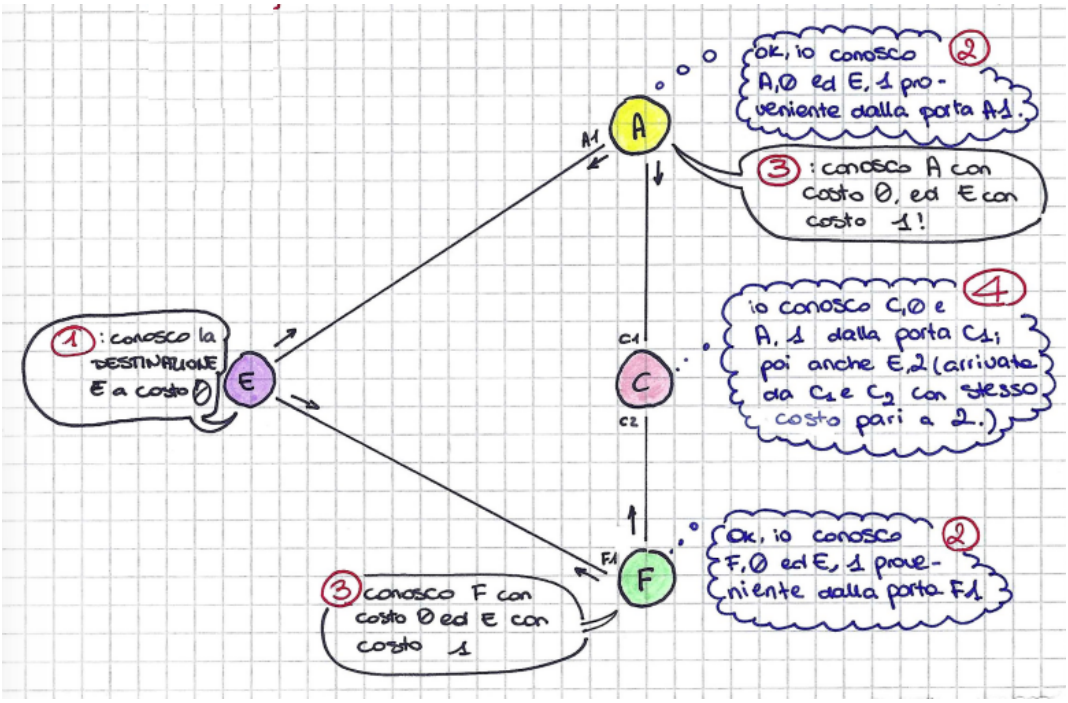
Now onto our carrellata:

- algoritmo di routing: trovare il miglior percorso da punto A a punto B ($\approx$ con il minor costo, data una certa misura tipo il nr Hop);
- Teoria dei Grafi (concetto visto nel corso di Algoritmi, tra le altre cose):
  - Grafo: un Grafo $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ è un insieme di $\mathcal{N}$ nodi ed $\mathcal{E}$ archi (dove E sta per Edges);
  - Ad ogni arco che va dal nodo x al nodo y (arco (x, y)) è associato un costo $c(x, y)$;
  - se il grafo non è orientato, allora vale sempre che gli archi sono uguali sia che vengono attraversati da x verso y che viceversa. $c(x, y) = c(y, x)$;
  - y è adiacente (o vicino) ad x se esiste un arco che li collega: $(x, y) \in \mathcal{E}$;
  - il costo complessivo di un percorso è la somma dei costi degli archi che lo compongono:

$$c(x_1, x_n) = c(x_1, x_2) + c(x_2, x_3) + ... + c(x_{n-1}, x_n)$$

- 2 categorie di algoritmi di instradamento:
  - statici: basati su tabelle manuali, percorsi che cambiano raramente;
  - dinamici: la topologia della rete, i percorsi e i costi possono cambiare per cui bisogna adeguarsi spesso ai cambiamenti (questo viene fatto in base a un certo timer);
- gli algoritmi si possono suddividere anche in queste due categorie:
  - sensibili al carico: a seconda del carico della rete, cambierà la metrica;
  - insensibili al carico: sono quelli utilizzati per la maggior parte (RIP, OSPF, BGP...)
- ci sono altre due categorie che si possono usare come criterio:
  - globali ($\approx$ link state): utilizzano tutti gli algoritmi a disposizione per conoscere l'intera mappa della rete. Per fare ciò, fanno quello che si dice "flooding" della rete;
  - decentralizzati ($\approx$ distance vector): non c'è una conoscenza *globale* della rete, solo dei nodi immediatamente vicini (più eventualmente il default gateway, ogni router ha conoscenza di se stesso!).

Principio Distance Vector



Proprietà di:

- distance vector:
  - \+ semplici da implementare;
  - \+ supportati da sistemi con scarse capacità di computazione e memoria;
  - \- problema del loop infinito (? anche se, stando alle mie note, questo è un problema che può essere risolto con il metodo "split horizon"[28]);
  - \- convergenza lenta (dipendente dal numero di nodi);
  - \- non possono usare molti hop (RIP ne ha massimo 15);
- link state:
  - \+ mappano la rete completamente;
  - \+ non sono suscettibili a errori;
  - \- consumano moltissima banda (flooding);
  - \- non facilissimi da configurare;
  - \- richiedono capacità elevate in generale;
  - $\sim$ si usano in topologie dense di router.

[28] vedere qui :)

La rete dell'università usa il protocollo link state OSPF.

I pacchetti che il router invia per comunicare con gli altri le proprie informazioni si chiamano "hello packet".

Algoritmi Link State

Protocolli SSSP (Single Source Shortest Path):

- Algoritmo di Dijkstra;
- Algoritmo di Bellman-Ford;[29]

Alg. di Dijkstra

L'algoritmo di Dijkstra si vede nel corso di Algoritmi, ragion per cui non lo approfondiremo troppo. Si consideri la rete in Fig. 17, con i nodi che sono Router e gli archi che rappresentano il costo di ciascun collegamento:

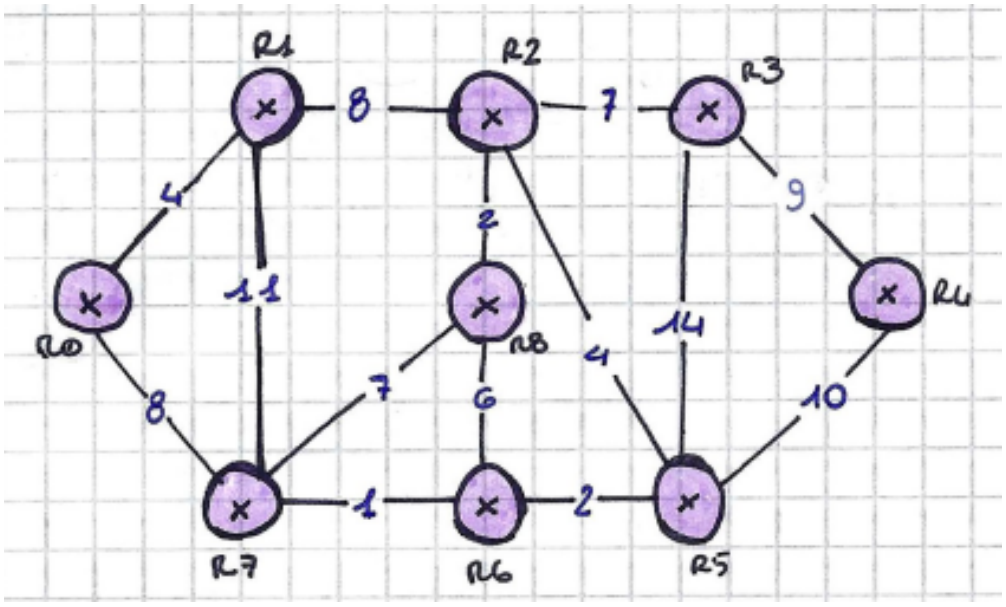


Figure 17: Enter Caption

1. Partiamo da R_0: i nodi che posso raggiungere e che scopro immediatamente sono R_1 con $c = 4$ e R_7 con $c = 8$;
2. da R_1, R_7 posso scoprire R_2, R_8 e R_6;

[29] differiscono essenzialmente per la metrica, se ho capito bene: il Bellman-Ford gestisce anche costi negativi, Dijkstra solo positivi.

3. qual è il percorso meno costoso (che conosco ora) da R_0 a $\{R_2, R_8, R_6\}$?
   (a) $(R_0 \Rightarrow R_2)$: $R_0 \to R_1 \to R_2$ con costo $4 + 8 = 12$;
   (b) $(R_0 \Rightarrow R_8)$: $R_0 \to R_1 \to R_2 \to R_8$ a costo $4 + 8 + 2 = 14$;
   (c) $(R_0 \Rightarrow R_6)$: $R_0 \to R_7 \to R_6$ a costo $8 + 1 = 9$;
4. da R_2, R_8, R_6 posso scoprire R_3 e R_5;
5. qual è il percorso meno costoso (che conosco ora) da R_0 a $\{R_3, R_5\}$?
   (a) $(R_0 \Rightarrow R_3)$: $R_0 \to R_1 \to R_2 \to R_3$ a costo $4 + 8 + 7 = 19$;
   (b) $(R_0 \Rightarrow R_5)$: $R_0 \to R_2 \to R_6 \to R_5$ a costo $8 + 1 + 2 = 11$;
6. infine, $R_0 \Rightarrow R_4$ si raggiunge via $R_0 \to R_7 \to R_6 \to R_5 \to R_4$ a costo $8 + 1 + 2 + 10 = 21$.
7. Ci sono più percorsi possibili da R_0 ai vari router, ma questi che ho riportato sono i meno costosi!! Questo è ciò che ci interessa determinare. Tipo $(R_0 \Rightarrow R_8)$ passando per $R_0 \to R_1 \to R_7$, ma il costo sarebbe $4 + 11 + 7 = 22$;

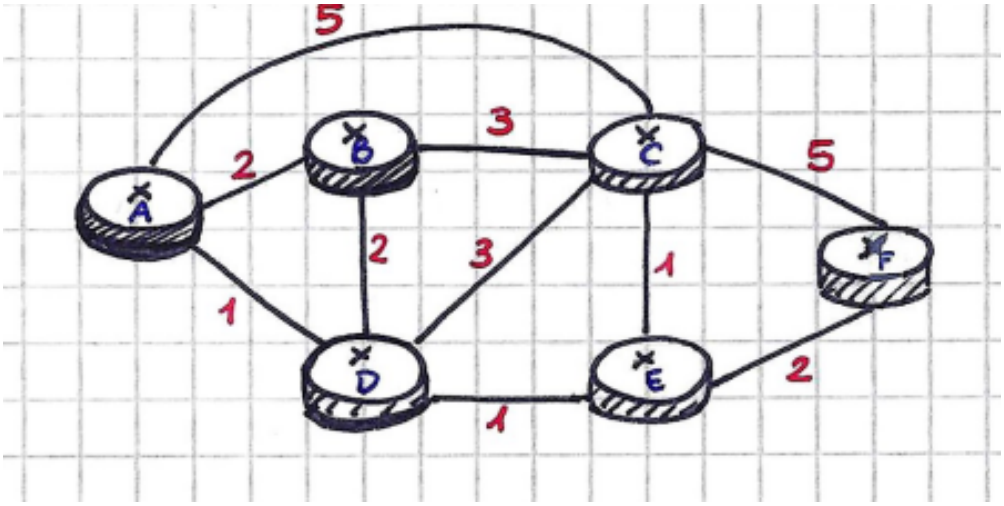


Figure 18: Altro esercizio da fare con Dijkstra.

Al termine dell'esercizio proposto in Fig. 18, sperando che io non mi sia sbagliata, dovreste avere un risultato del genere:

Altre Cose su Data Link ed Ethernet

In LAN, nel momento in cui un host trasmette nella rete, esso diventa proprietario di TUTTO il mezzo trasmissivo! (nel senso che, fintanto che è lui a parlare, è lui a monopolizzare il canale). Questo era un problema di sicurezza "non indifferente" agli albori della tecnologia LAN, il fatto che tutti i messaggi fossero broadcast, a cui si è in qualche modo ovviato con CSMA-CD, che vedremo in un attimo.

n. Hop da A	Router attraversati	B	C	D	E	F
0	A	2,A	5,A	1,A	∞	∞
1	AD	2,A	4,D	-	2,D	∞
2	ADE	2,A	3,E	-	-	4,E
3	ADEB	-	3,E	-	-	4,E
4	ADEBC	-	-	-	-	4,E
5	ADEBCF	-	-	-	-	-

Table 3: Percorsi di minor costo da A alle varie direzioni.

destinazione	next hop (passo 1)	costo
F	D	4
C	D	3
B	(B stesso)	2
3	D	2
D	(D stesso)	1

Table 4: Tabella di Routing interna ad A.

Con il termine Ethernet ci si riferisce sia alla tecnologia per connettere dispositivi in LAN (cablaggio e porte, immagino), sia ad una componente logica, algoritmica se vogliamo, utilizzata per gestire le trasmissioni di dati su questa rete. In altre parole, sentiamo parlare sia di "Cavo Ethernet" (tecnologia fisica) che di "Frame Ethernet" (tecnologia logica, come un Pacchetto IP).

Memo degli standard e di chi li ha pubblicati (cercate di ricordarveli questi, sono importanti):

- IEEE 802 : definisce diversi standard LAN, tra cui quelli più rilevanti sono:
  - 802.3 : Ethernet;
  - 802.11 : WLAN (Wireless LAN) *& Mesh (Wi-Fi certification)* (qualsiasi cosa significhi ciò);
  - gli altri punti di IEEE 802 sono stati dismessi, come ad esempio 802.5 che definiva la topologia Token Ring (ampiamente in disuso oggi).
- EIA/TIA 568 : definisce gli standard per il cablaggio strutturato (e.g. come vanno disposti i cavi colorati all'interno della spina RJ-45 ad un estremo e all'altro del cavo Ethernet, se si vogliono creare cavi Straight-Through o Crossover, questo lo vedremo nel prossimo capitolo);

Di nuovo, giusto per ripetermi: a livello Data Link, la PDU prende il nome di Trama o Frame (quindi per esempio si parla di "Frame" Ethernet).

La struttura del Frame Data Link dipende dalla tecnologia fisica sottostante, pertanto non c'è uno standard unico ben preciso sulla struttura di un Frame. In linea generica, però, le sue dimensioni sono comprese tra $64 \div 1518$ Bytes, e tra le informazioni trasmesse ci sono:

- un trailer all'inizio, che contiene indirizzi di destinatario e mittente;

- il payload;
- una forma di rilevazione errori come CRC, controllo a ridondanza ciclica, o controllo di parità i somme di controllo;

I collegamenti possono essere fondamentalmente di 2 tipi:

- Broadcast : più stazioni connesse con lo stesso mezzo trasmissivo;
- Point to Point : cablato o wireless, collega una stazione ad una stazione e basta. (in casi come questo viene utilizzato il cavo Cross-Over, e più in generale ogniqualvolta si intende collegare tra loro due dispositivi dello stesso tipo, Router a Router, Host a Host...)

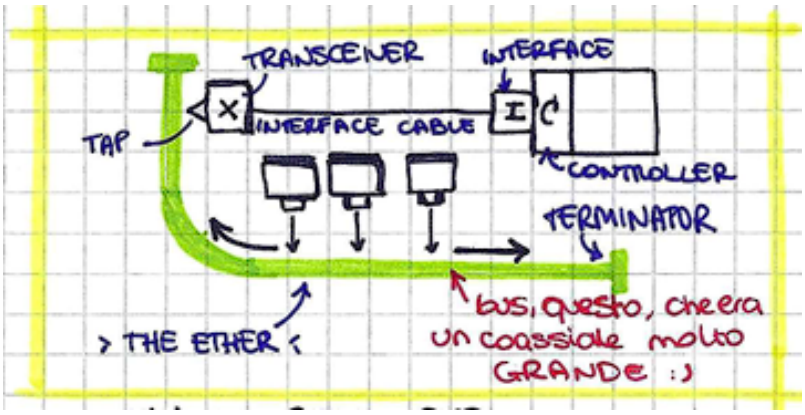
In caso di reti ad accesso multiplo (LAN, reti via satellite...), sono richiesti protocollo ad accesso multiplo per dirigere il traffico senza causare collisioni. Questi possono operare per:

- suddivisione del canale in TDM/FDM (come dicevamo nell'introduzione, dividere la banda comune in slot di tempo o di frequenza) ($\approx$ Channel Partitioning Protocols);
- accesso casuale (RAP, Random Access Protocols), come lo Slotted ALOHA;
- a rotazione (taking-turn protocols), in cui gli host parlano a turno secondo diversi criteri come Polling o Token-Passing.

Tecniche di allocazione del canale trasmissivo:

- statica: il mezzo viene partizionato tipo TDM/FDM;
- dinamica: tutte le frequenze del mezzo trasmissivo vengono assegnate di volta in volta ai vari host (nel caso dei protocolli turn-based, naturalmente, occorre definire un algoritmo che gestisca questa assegnazione dinamica).

Sketch originale (riprodotto da me, ofc) della tecnologia Ethernet, ideata nel 1976 da Metcalfe:



- c'è un solo bus condiviso (in origine, un cavo coassiale);
- è non-deterministico - non c'è nessuna regola che stabilisce quando parla chi (quindi ci si chiede: quando e dove si trasmette sul bus?)
- ad oggi, Ethernet è la tecnologia cablata più diffusa, si è passati da una topologia a Bus come quella nello sketch ad una topologia a stella (switched Ethernet[30])

Topologia a Bus:

- interruzione del bus in qualsiasi punto → interruzione di tutta la rete;
- aggiungere un end system alla rete comporta un grosso intervento, con interruzione del servizio nell'intero sistema;
- un frame immesso nel bus si propaga ogni volta in entrambe le direzioni del bus.

Aloha: protocollo per topologie a Bus (indovinate in che paese è stato ideato). Consente trasmissione in modo casuale, quindi spesso i frame mandati in rete allo stesso momento andavano in collisione! Collisioni che potevano essere parziali o totali, a seconda della % di timeslot sovrapposti tra i messaggi in conflitto. È un protocollo facile da realizzare, ma naturalmente presenta dei problemi di efficienza.

CSMA

Carrier Sense Multiple Access, dove Carrier Sense è un modo per dire, più o meno: "prima di parlare, ascolta." (e mentre trasmetti, continua ad ascoltare)

- CSMA-CA: Collision Avoidance;
- CSMA-CD: Collision Detection.

[30] vedremo poi che si è passati all'utilizzo del cosiddetto Ethernet Hub, che comunque contiene una forma di Bus al suo interno

Ma! Neppure l'utilizzo dei protocolli CSMA-CA/CD scongiura del tutto le collisioni.
Supponete che per caso inizino a parlare 2 End Systems in contemporanea: collidono una volta, si fermano entrambi e lasciano parlare l'altro, tipo la scena cliché in cui due persone iniziano a dire entrambe "no scusa, vai prima tu" e così facendo parlano assieme di nuovo e avanti così più e più volte.
Come si scongiura questa situazione di stallo infinito deadlock? Si tenta di dire, al momento della collisione, agli host di aspettare un tempo T generato randomicamente: questa istruzione di aspettare viene data attraverso un segnale speciale ("jamming") che ha una certa frequenza; Se, disgraziatamente, i tempi randomici scelti dalle parti coinvolte sono di nuovo uguali e c'è di nuovo interferenza, si raddoppia il range di tempi T di attesa da scegliere a caso! Nel senso: se alla prima collisione avrei aspettato un tempo nell'insieme I$\mathcal{T} = \{0, 1\}$, la seconda volta sceglierò tra $\mathcal{T}' = \{0, 1, 2, 3\}$: più elementi ci sono nell'insieme $\mathcal{T}$, minore è la probabilità che 2 End System peschino lo stesso t. Questo meccanismo si chiama "Exponential Backoff".

Andiamo avanti. Dominio:

- di Broadcast: parti di rete in cui il messaggio broadcast riesce a raggiungere tutti gli host: nel caso di una LAN, il dominio di BC e limitato dal Router (Gateway).
- di Collisione: parti della rete dove c'è la possibilità che si verifichino collisioni.

Tempo per rilevare una collisione:

$$a = \frac{\text{lunghezza\_collegamento}}{\text{lunghezza\_pkt}}$$

6 - Livello Fisico e Sicurezza

Specifiche fisiche Ethernet

- 10BaseT, dove 10 è la banda base, T sta per "Twisted Pair".
  - trasmissione 10mbps in banda base;
  - sia coassiale che UTP (la cronologia dei cambiamenti nella tecnologia ha fatto: coassiale thick > coassiale thin > doppino CAT3);
  - lunghezza max cavo: 100 metri;
  - connettore RJ45, affidabile ed economico, facile da implementare, al contrario del BNC che unisce i coassiali.
- 100BaseT, differisce da 10BaseT in:
  - la velocità di trasmissione;
  - ha 3 tipi possibili di connettore:
    - 100Base-T4, doppino 4 coppie;
    - 100Base-TX, doppino 2 coppie;
    - 100Base-FX, fibra ottica.
  - comunque è retrocompatibile con 10BaseT;
- Gigabit Ethernet (802.3Z): fino 1Gbps e non retrocompatibile.
- Questi standard Ethernet sono tutti a tecnologia CSMA-CD

Topologia ad Anello:

- semplifica coordinamento accessi;
- più facile identificare guasti;

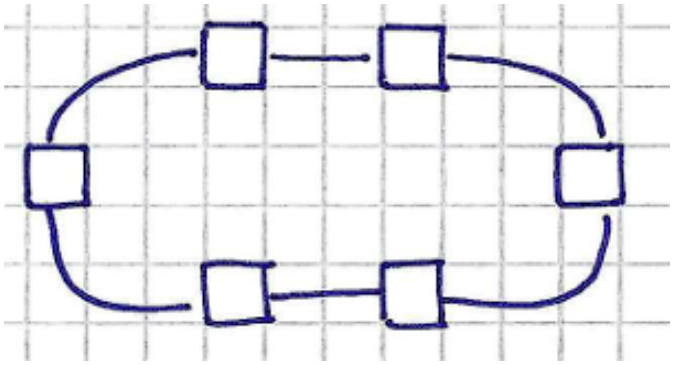


Figure 19: Topologia ad anello, supersemplificata.

- se si guasta un solo elemento, la rete è persa (come i circuiti in serie contrapposti a quelli in parallelo, non so se avete presente il discorso fondamentale dei circuiti elettrici)

Tecnologia "Token Ring": dove il Token è un tipo particolare di frame che dà il permesso agli host per parlare. In Ethernet non c'è modo di stabilire chi può parlare quando. Sul piano Logico, le Token Ring sono disposte ad anello (vedasi Fig. 19), ma Fisicamente conviene disporle come stella a doppio anello e utilizzare un MAU, Media/Multistation Access Unit.
Topologia FDDI: Fiber Distributed Data Interface (in disuso), sempre di tipo Token Ring, sempre cablata ad anello o stella, viaggiava a velocità elevate. Usava il doppio anello, due anelli indipendenti (rete autoriparante!).

Livello fisico, reti wireless

Utilizzano (standard IEEE 802.11[31]):

- radiofrequenze;
- infrarossi.

Devono avere un access point collegato alla rete cablata;
L'arbitraggio del canale trasmissivo si può fare attraverso vari algoritmi (non c'è uno standard preciso);
Velocità di trasmissione: fino a 300Mbps;
Alcune problematiche del caso:

- propagazione onde radio;
- occupazione delle frequenze;
- inaffidabilità;

[31] 802.11a : standard per la rete 5 GHz, 802.11b/g : 2.4GHz (802.11b ha max vel. 11Mbps, 802.11g ha max vel. 54Mbps

- potenza ridotta;
- sicurezza.

Più l'host è distante dall'access point, meno sarà la velocità di navigazione (max 50mt circa).

Tipi di interferenze:

- strutturali: in edifici di cemento armato, la propagazione del segnale è difficile;
- su distanze elevate (e.g. ponti radio): edifici, vegetazione, nebbia e fenomeni atmosferici possono disturbare il segnale, le antenne vanno posizionate in un punto adeguato;
- dovute ad altre sorgenti: Bluetooth, interfoni per neonati, molti apparecchi utilizzano la frequenza $f = 2.4GHz$ (con la 5 GHz si verificano meno interferenze).

Modalità di collegamento:

- Peer to Peer (P2P): senza passare per l'access point[32]. Dipende dalla scheda di rete del dispositivo;
- Client - Access Point;
- Multiple Access Point & Roaming. mobilità: l'utente cambia access point mentre si sposta, senza percepire interruzioni di servizio;
- Bridging con antenna direzionale

Area di servizio: l'ambito in cui interviene l'access point.

Sicurezza - Wireless

Livelli di sicurezza Wireless:

- open system: detto anche "con chiave di autenticazione condivisa", non è esattamente la stessa cosa che dire "possono entrare tutti". Vuol dire che è aperta, sì, ma non sono necessariamente la stessa cosa;
- WEP: Wired Equivalent Privacy;
- WPA/WPA2: WiFi Protected Access. Queste offrono:
  - cifratura dei dati (standard IEEE 802.1X);
  - integrità dei dati (garanzia che i dati ricevuti sono uguali a quelli spediti, nessuno li ha manomessi lungo il tragitto);

[32]Penso che il Peer to Peer in internet sia diverso dal Peer to Peer che si intende qui. Qui penso si intenda una connessione fisica come quella che potremmo fare tra due PC sulla stessa scrivania, magari con tecnologia Bluetooth

  - protezione da attacchi Replay[33]

WPA:

- Personal: privata;
- Enterprise: ha bisogno di un Server di Autenticazione (e.g.: Radius, TACACS) che contiene cose come le credenziali degli utenti;

Wireless Sensor Network

Reti di sensori collegate a controllori come Arduino. I sensori sono dispositivi piccoli, economici, limitati in capacità di elaborazione e trasmissione. Fondamentalmente rilevano delle misurazioni: di luminosità, di temperatura, di umidità, etc.
Caratteristiche delle reti di sensori:

- composte da molti sensori che monitorano fenomeni in continuazione;
- reti Ad-Hoc, disegnate su misura per ciascuna situazione;
- dialogano tra loro tramite antenna;
- convergono tutti con i propri dati nel Sink (server centrale);
- ZigBee: standard (802.15.4) RL-WPAN, Wireless Personal Area Network; opera su frequenza 2.4 GHz a basso consumo.

Propagazione delle onde:
Potenza di radiazione: la normativa tecnica ETS 300-328 impone di NON irradiare con una potenza EIRP[34] $>$ 100mW ($\approx$ 20dBm).

Propagazione:

- Direzionale: guadagno > 1;
- Omnidirezionale: guadagno $= 1$.

Ogni antenna ha il suo diagramma di radiazione per aiutare a capire come orientarla al meglio (sketch in Fig. 20).

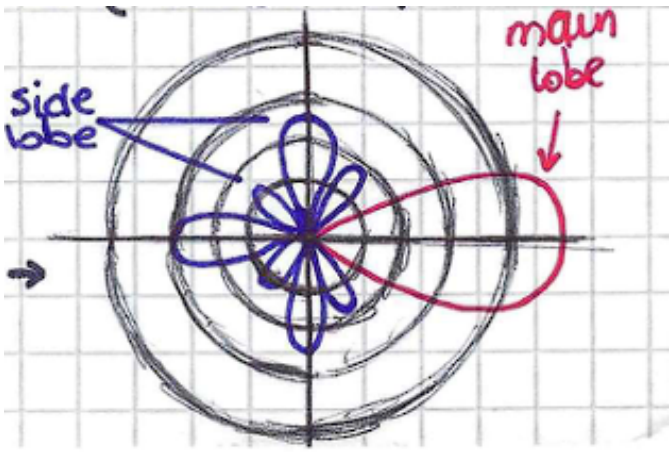


Figure 20: Diagramma di radiazione supersemplificato

Grado di protezione IP(Ingress Protection), indicatore del grado di protezione di un dispositivo da agenti esterni. Formato da 2 cifre:

[33] attacco Replay: catturare pacchetti scambiati tra client e server e replicarli facendoli partire dall'indirizzo dell'attaccante allo scopo di risalire a chiavi di autenticazione.
[34] Effective Isotropic Radiated Power

I. (Valore $0 \div 6$): protezione da oggetti solidi (dove 0 corrisponde a "nessuna protezione" e 6 corrisponde a "protezione totale da polveri");

II. (Valore $0 \div 8$): permeabilità dell'acqua (dove 0 corrisponde a "nessuna protezione" e 8 corrisponde a "resistente a immersione continua")

Altre specifiche dello Standard 802.3

Ovvero, altro cablaggio.

Coassiale e Doppino Intrecciato

- 10Base5: definisce il cosiddetto "coassiale thick" (lunghezza fino a 500m)
  - "thicknet": consisteva in un unico bus coassiale, ciascuna macchina per collegarsi al bus aveva bisogno di un Transceiver (costoso, ndr). Sulle schede di rete, per collegarsi al Transceiver, c'era una interfaccia AUI come quella in Fig. 21
- 10Base2: ovvero, il "coassiale thin" (lunghezza fino a 185m). Nella "thinnet" non c'era più bisogno del Transceiver, in compenso però non si potevano usare prolunghe. L'interfaccia usata si chiama comunemente "T" (Fig. 22)

Doppino telefonico:

- non ci si collega più ad un bus comune ma a degli Switch;
- connettori BNC sostituiti da connettori RJ45;
- max distanza 100m.

La disposizione dei fili all'interno del connettore RJ45 determina che tipo di collegamento si sta creando (schema T568-A, T568-B, in Fig. 23):

- se si realizza un cavo con connettori uguali a entrambe le estremità (quindi A-A o B-B), il cavo sarà di tipo straight-through;
- se si configurano connettori diversi (A-B, B-A), il cavo esce di tipo cross-over;

Doppini:

- UTP: doppino non-schermato (Unshielded Twisted Pair);

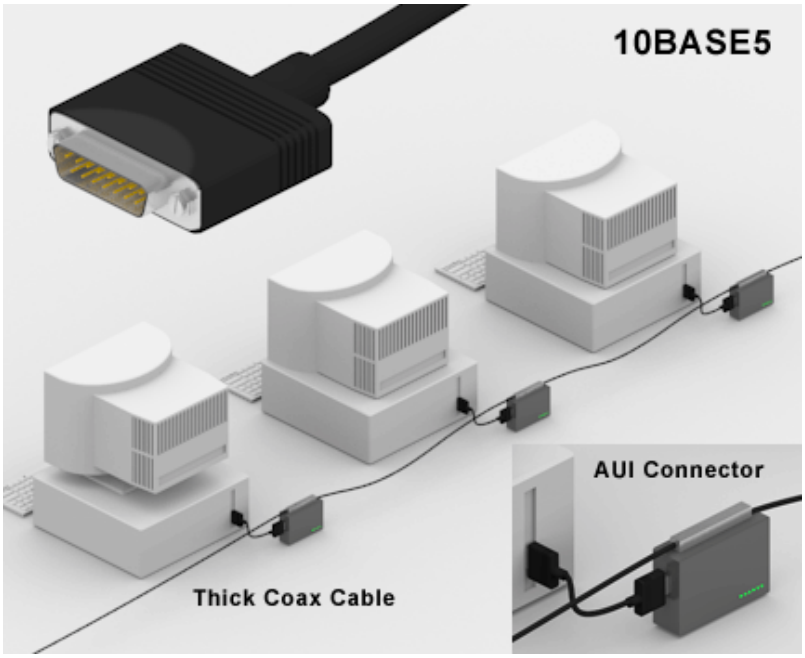


Figure 21: Rappresentazione di rete Thicknet. Fonte: ComputerLanguage.com

- FTP: doppino schermato con foglio di alluminio + collegamento a massa (Foiled Twisted Pair);
- STP: doppino con schermo locale (Shielded Twisted Pair);
- S-UTP / S-FTP: doppino con schermo locale e foglio di alluminio.

Figure 22: Connettore T per reti Thinnet. All'estremità "lunga" viene collegato il terminale, a quelle corte rispettivamente l'input e il resto del ring (o un Terminatore, tipo un tappo a impedenza 50Ω di tipo BNC come le altre estremità dei coassiali)

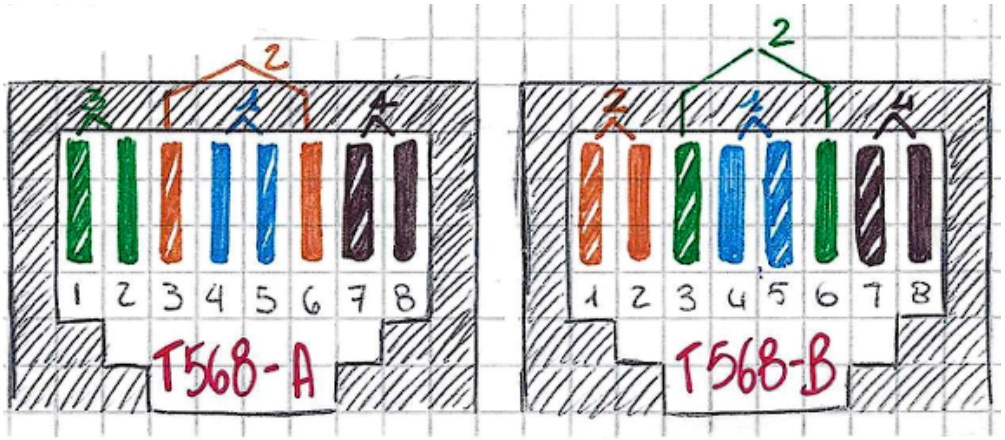


Figure 23: Schema delle binature T568-A e T568-B.

Categorie:

CAT 1 : solo per segnali vocali (telefonia);

CAT 2 : max vel. 4Mbps;

CAT 3 : max vel. 10Mbps;

CAT 4 : max vel. 16Mbps;

CAT 5 : max vel. 100Mbps con banda passante 100MHz;

CAT 5e : max vel. fino a 1Gbps;

CAT 6 : max vel. fino a 10Gbps, banda passante fino a 250MHz;

CAT 7 : non riconosciuto da EIA/TIA.

N.B.: queste note sono state redatte inizialmente nel 2021, quindi non sono da escludere nuovi sviluppi di standard del doppino. Per preservare l'intreccio dei cavi interni, le norme di installazione vogliono che non si eserciti una forza superiore a $11,3$ Kg (insomma, non tirare i cavi così forte).

Fibra Ottica

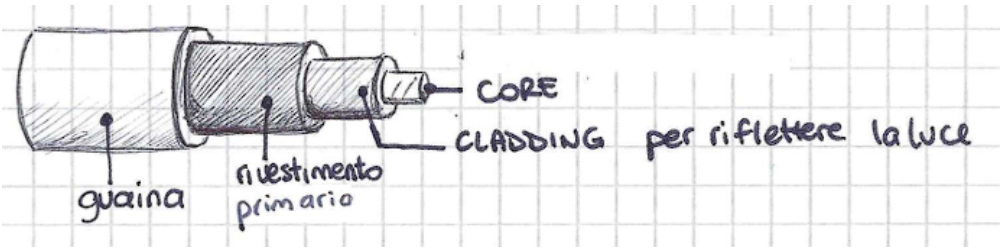


Figure 24: Schema della struttura interna di un cavo di fibra ottica.

- la fibra ottica, al contrario del coassiale e del doppino (che hanno bisogno di tutte quelle schermature) è immune alle interferenze;
- può essere:
  - monomodale;
  - multimodale (trasmette luce su più frequenze);
- tipo di connettore: ST (Straight Tip), o più spesso SC (Standard Connector).

Estensione di una LAN

Perché estendere una LAN?

- unificare LAN costruite in momenti diversi;
- unificare LAN situate in diversi edifici;
- distribuire carichi di traffico elevati;
- aumentare la distanza copribile;
- aumentare l'affidabilità;
- aumentare la sicurezza.

Ci sono naturalmente delle limitazioni dovute a protocolli di accesso e al decadimento del segnale - non si può espandere all'infinito, insomma.
Come estendere una LAN?

- con dei Repeater (Layer 1 device): come un buffer di segnale;
- con dei Bridge (Layer 2 device): leggono l'intestazione dei frame e inoltrano in base al contenuto che leggono. Non propagano collisioni. Sono adattivi, ovvero si configurano a mano a mano che ricevono informazioni sulla rete (credo facciano similmente agli switch). È possibile collegare con un bridge anche reti con tecnologie diverse, ad esempio Ethernet con Token Ring.

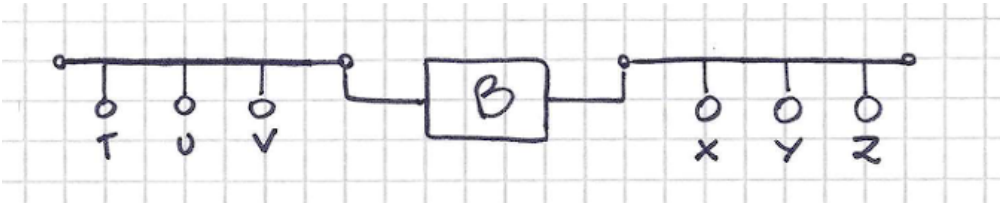


Figure 25: Bridge B collega due LAN di host TUV e XYZ.

- Transparent Bridge: se X deve comunicare con Z nell'esempio in Fig. 25, la prima volta il frame viene inoltrato lungo TUTTA la rete, ma al ritorno il bridge avrà già appreso che X e Z sono nella stessa porzione di rete, quindi eviterà di inoltrarlo alla parte con TUV. (Backward Learning)
- Source Node Bridge: i bridge non mantengono queste informazioni di routing, ma sono capaci di selezionare il percorso ottimale tra due punti.

Ciclo di Bridge: dovuto a flooding, alcuni frame rimangono in circolo a lungo senza raggiungere una destinazione. Soluzione: Distributed Spanning Tree Protocol (STP), serve per far comunicare i bridge, in breve.

Fault Tolerance

Si costruisce una rete con doppi concentratori (Fig. 26), in modo tale che se uno dovesse smettere di funzionare, l'altro terrebbe comunque in normale funzione il servizio. Aggiungere dispositivi "di scorta" prende il nome di Ridondanza.
Gli switch possono essere stackati (ad esempio, se ne può creare uno a 96 porte impilandone 2 da 48). Gli switch si possono gestire con CLI o GUI, accessibili via cavo o remote shell.
La tecnologia VLAN (Virtual LAN) serve a dividere virtualmente un solo switch in più LAN.
Reti Commutate (con Switch) $\approx$ come un mega bridge, in un certo senso. Lavorano in 3 modalità:

- Cut-Through: lo switch legge il MAC, inoltra dove deve, fine.

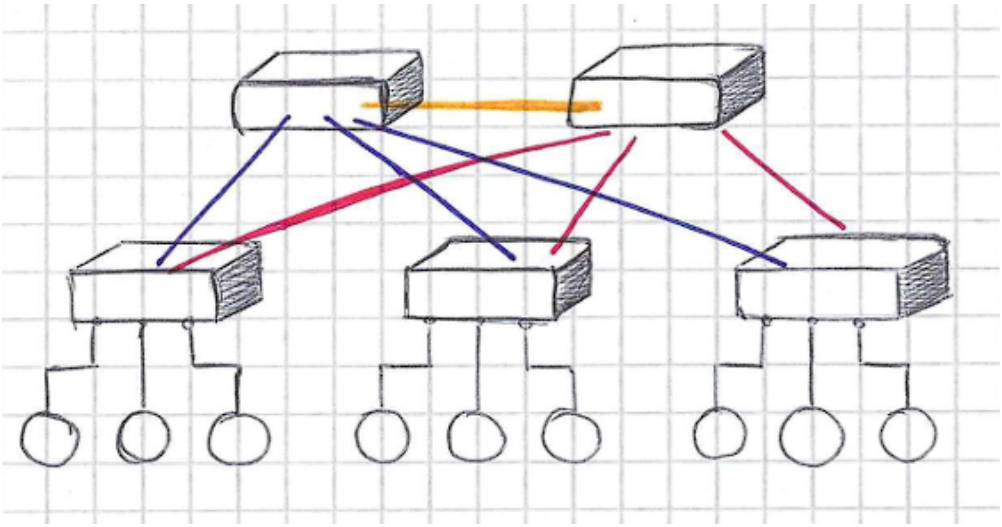


Figure 26: Centro Stella Ridondato, meccanismo di fault tolerance

- Store & Forward: Legge il MAC, immagazzina tutto il messaggio e ne controlla i CRC, inoltra se tutto ok;
- Fragment Free: una via di mezzo, legge solo i primi 64 Byte del frame e fa controlli solo su quel prefisso. Se tutto ok, inoltra. (tant'è che, a differenza del header IP, nello header MAC viene prima il campo Destinazione poi Sorgente)

Indice Alfabetico dei Contenuti