INTERNET RETI & SICUREZZA

tentativo di magnum opus a cura di Cleo R.
2021. SOURCE: KUROSE-ROSS 8ed., appunti vari, ricerche, etc.

CAPITOLO I: Computer Networks and the Internet

- **Cosa è Internet?** --> concetto trasversale!
  - una RETE che collega tra loro più unità di calcolo sparse geograficamente;
  - una Rete a commutazione di pacchetto (definizione misera ma hey, lo è!)

Normalmente, i terminali non sono collegati tra loro in modo diretto, ma tramite dispositivi di COMMUTAZIONE (ROUTER) che prelevano le informazioni in pacchetti e le reindirizzano sui link di uscita;

Route/path: itinerario compiuto dal pacchetto attraverso la rete;

Commutazione di pacchetto (PBN: Packet based network): più terminali condividono lo stesso cammino o una parte;

L'accesso dei terminali ad internet avviene attraverso gli ISP (Internet Service Provider); ogni ISP costituisce una RETE DI ROUTER;

top Italian ISPs:
i) Vodafone;
ii) Fastweb;
iii) Telecom Italia;
iv) EOLO; (2013)

TCP/IP: protocolli che gestiscono invii e ricezioni;

Intranet: Reti private strutturate similmente alla pubblica internet;

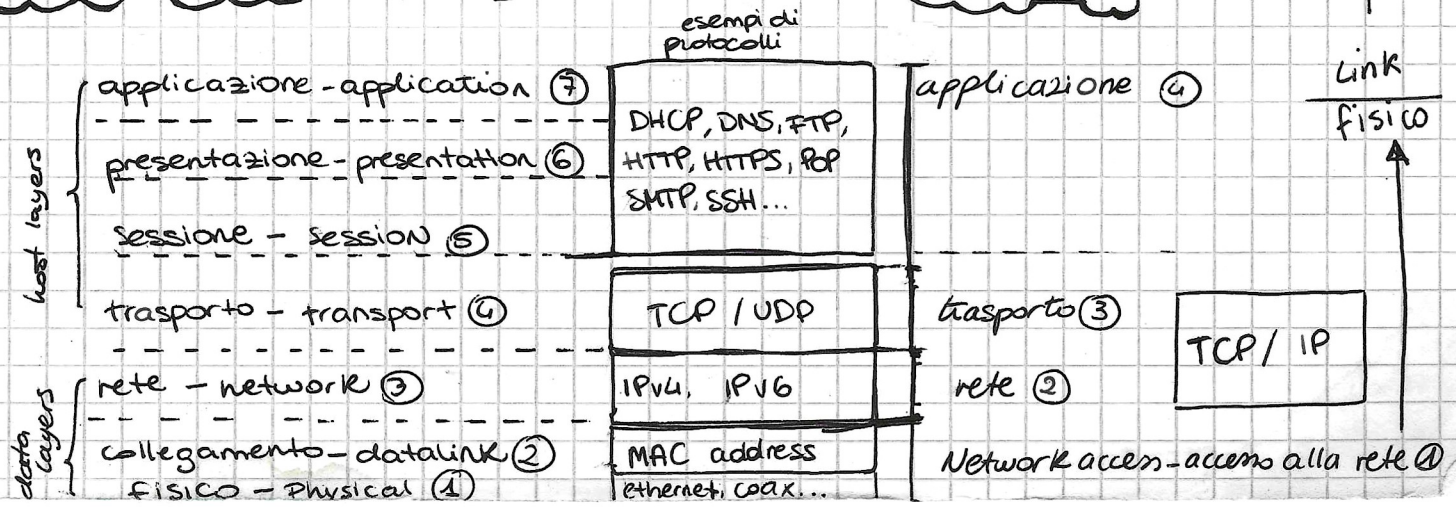
Protocollo: insieme di regole formalmente descritte al fine di favorire la comunicazione tra una o più entità;
Due entità che intendono comunicare tra loro, devono adottare lo stesso protocollo.

→ Transmission Control Protocol: connection-oriented service [RFC793]
- Reliability; - Flow Control; - Congestion Control;

IMPORTANTE

me parlava fossero 5...

ISO-OSI model (VS) TCP-IP



Che poi permettetemi di fare un heads-up: ISO/OSI prende il nome da:

- **ISO**, che è l'organizzazione degli standard internazionale;
- **OSI**, il nome del modello (Open Systems Interconnection);

Host: sistemi periferici che ospitano programmi applicativi (browser etc).
- **client**: host che richiede dei servizi
- **server**: host che li fornisce

app distribuite
in internet si lavora tramite APP DISTRIBUITE: due o più processi eseguiti in parallelo su diverse macchine attraverso Internet

DSL: digital subscriber line - accesso residenziale a banda larga

Tipi di Ritardo: i) di elaborazione; ii) di accodamento; iii) di trasmissione (L/R) iiii) di Propagazione

$\sum R$ = NODAL Delay

Network core vs Network edge: basicamente, il NE sono i sistemi periferici (e.g. laptop), il NC è tutta la struttura nel mezzo (router etc.)

LAN (Local Area Network): usata per collegare un end system all'edge router.
E.G.: Ethernet (tecnologia d'accesso più usata)

Standard IEE per WiFi: **802.11** → fun fact: WiFi non è mai stato ufficialmente ~~[illegible]~~ contrazione di Wireless Fidelity - nonostante l'IEEE poi lo abbia affermato.

perlomeno non secondo la Wifi Alliance

Physical Media

- **twisted pair copper wire** (doppino intrecciato in rame)
  UTP unshielded twisted pair - usato comunemente in LAN
  un cavo UTP categoria 6a può trasportare fino a 10 Gbps in velocità.
  - fun fact: il doppino è intrecciato così perché rende molto più "semplice" la cancellazione dei disturbi.
- **cavo coassiale**: quello dell'antenna, precisamente.
  può essere usato in condivisione da più ESs.
- **fibra ottica**: fast & pricy boi che va dai 51.8 Mai 39.8 Gbps di link speed.

$$\text{TEMPO DI TRASMISSIONE} = \frac{L}{R}$$

L: dimensione pacchetto in bit
R: rateo di trasmissione sul link

MODEM: è una specie di acronimo che sta per MOdulatore-DEModulatore.

Store-and-forward: il router deve ricevere tutto il pacchetto prima di inoltrarlo dove spetta. → Ritardo totale $2L/R$ per pacchetto.

telefonia!

COMMUTAZIONE di CIRCUITO (vedi dopo) VS **COMMUTAZIONE di PACCHETTO**

Le risorse richieste dagli ES per comunicare sono RISERVATE a questi per tutta la durata della connessione. Come l'obbligo di prenotare al ristorante.

Le risorse non sono riservate. I messaggi usano le risorse on-demand e potrebbero esserci queuing delays.

heads-up: ISO/OSI prende il nome da:

egli standard internazionale;
pen Systems Interconnection);

ano programmi applicativi (browser etc).

de dei servizi
ornisce

app distribuite

in Internet si lavora tramite APP DISTRIBUITE: due o più processi eseguiti in parallelo su diverse macchine attraverso Internet

- accesso residenziale a banda larga

orazione; ii) di accodamento; iii) di trasmissione (L/R)
opagazione

$\sum R$ = NODAL Delay

asicamente, il NE sono i sistemi periferici (e.g. laptop),
l mezzo (router etc.)

sata per collegare un end system all'edge router.
(tecnologia d'accesso più usata)

02.11 → fun fact: WiFi non e' mai stato ufficialmente ~~[illegible]~~ contrazione di Wireless Fidelity - nonostante l'IEEE poi lo abbia affermato.

perlomeno non secondo la Wifi Alliance

ppino intrecciato in rame)
pair - usato comunemente in LAN
può trasportare fino a

→ fun fact: il doppino e' intrecciato cosi' perche' rende molto più "semplice" la cancellazione dei disturbi.

antenna, precisamente.
ondivisione da più ESs.

che va dai 51.8Mai 39.8Gbps

imensione pacchetto
in bit

ateo di trasmissione
sul link

MODEM: e' una specie di acronimo che sta per MOdulatore-DEModulatore!

deve ricevere tutto il pacchetto prima di
→ Ritardo totale $2L/R$ per pacchetto.

COMMUTAZIONE di PACCHETTO

Le risorse NON sono riservate.
I messaggi usano le risorse on-demand e potrebbero esserci queuing delays.

FOCUS SUI RITARDI:

i) si crea nel router per esaminare il pacchetto;

ii) si crea in coda in uscita in un BUFFER;

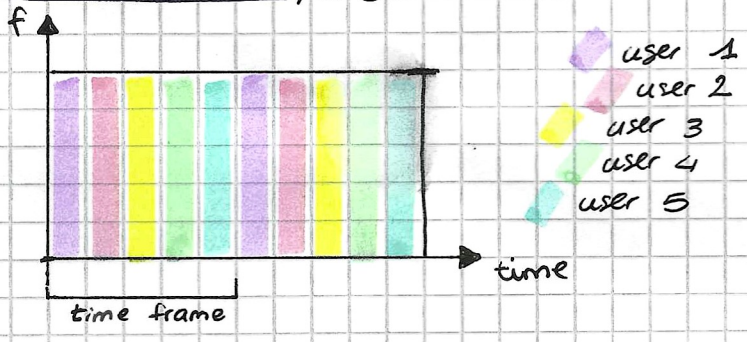
iii) il pacchetto dev'essere trasmesso sottoforma di 0 e 1, dipende dalla sua dimensione;

iiii) questo e' pettamente FISICO, dipende dalla DISTANZA tra Router A e Router B;

(1) la comm. di circuito può avere due tipi di multiplexing:

FDM
frequency-division mpxing

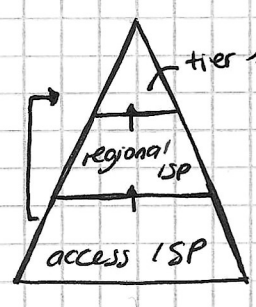
TDM
time-division mpxing



time slot → dedicate esclusivamente a coppie sender-receiver

struttura di Internet!

non penso il prof la chieda né che ne parli, n.d.r.



multi-tier hierarchy, Network Structure 3 (rapp.ne approssimativa della odierna Internet)

++ Points of Presence & IXP (internet exchange points) → → NS 4

↓

centri creati da terze parti per fare peering tra provider.

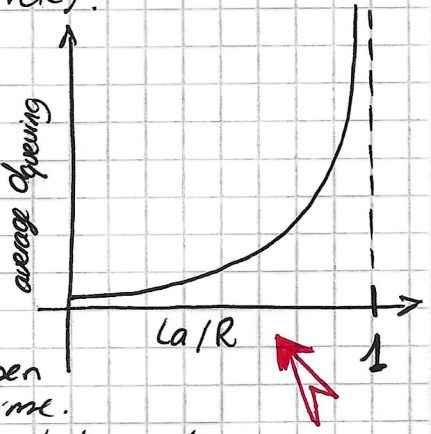
++ CONTENT PROVIDER NETWORKS (e.g.: Google) =

NETWORK STRUCTURE 5 ≡ Internet as of today

Intensità di traffico: La/R con L: dimensione in bits
a: average rate of packets/sec
R: transmission rate

Se l'intensità di traffico è > 1, allora la # di bit che arriva in coda supera la # di bit che esce dalla coda (che è molto male, converrete).

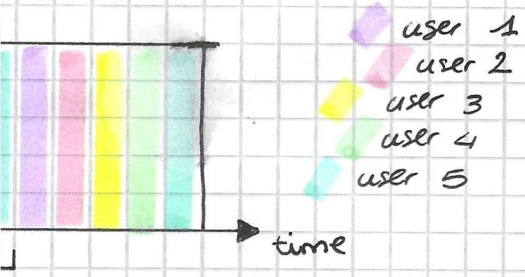
DoS attacks!  (DoS = denial of service)



- vulnerability attack: usa un messaggio "maligno" per compromettere app. o host interi;
- bandwidth flooding: invio spropositato di pacchetti al target da intasare l'access link, quindi inutilizzabile;
- connection flooding: richiesta enorme di connessioni TCP fully open al punto che l'host vittima non può più accettarne altre legittime.
  ↳ spesso possibile tramite DDoS (distributed DoS), l'attaccante controlla più macchine "zombie" che impiega per attaccare l'host vittima.

tiplexing:

on mpxing)



dicate esclusivamente a coppie sender-receiver

Internet! non penso il prof la chieda né che ne parli, n dr.

rk Structure 3 (rapp.ne approssimativa della

* IXP (internet exchange points) →

↓

centri creati da terze parti per fare peering tra provider.

WORKS
=

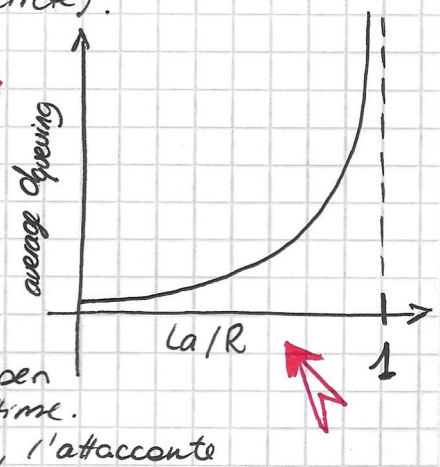
= 5) ≡ Internet as of today

: dimensione in bits
average rate of packets/sec
transmission rate
di bit che arriva in coda supera la #
molto male, converrete).

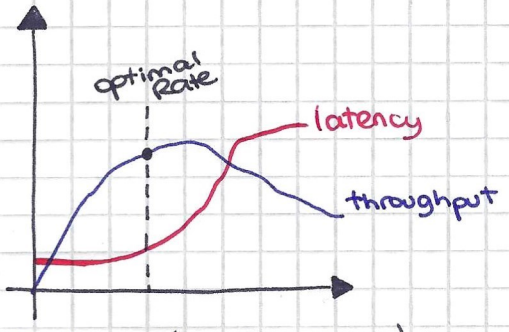
ervice)

' per compromettere

cchetti al target
bile;
nessioni TCP fully open
cettarne altre legittime.
(distributed DoS), l'attaccante
e' che impiega per attaccare l'host vittima.



Latenza → intervallo di tempo tra quando spedisco un input e quando e' disponibile l'output



(poi dipende dall'applicazione)

TRACEROUTE manda tot pacchetti ai router compresi tra host e destinazione rilevandoli tutti.
> tracert www.rai.it (212.162.68.64)

un protocollo APPARTIENE ad un layer, e può essere Hardware, Software o misto.

PDU: protocol data unit: il messaggio, basically, che a seconda del layer che attraversa cambia nome (E.g.: datagramma, frame...)
La PDU e' sempre formata da (header) + (payload). vedremo.

Incapsulamento: ricevo il messaggio dal layer sopra, ci aggiungo il mio header e lo passo sotto. E sotto succede la stessa cosa. In generale:
$PDU_n = HeaderPDU_n + (PDU_{n+1})$
spero sia chiaro

Tra 2 layer nel mezzo ci sono interfacce.

SAP service access point: servizi messi a disposizione dalle interfacce

Bandwidth: intervallo di frequenza che un sistema può garantire per trasmettere
↳ BITRATE: dipende da bandwidth (teorema di Nyquist - Shannon) (garantito NOMINALMENTE)
THROUGHPUT: VERA VELOCITA' della rete, vera capacita' di un canale trasmissivo
↳ non può essere considerato COSTANTE, si parla di throughput MEDIO end-to-end.

Th di Nyquist:

$\text{bit rate} = 2 \cdot H \cdot \log_2 V$

max velocità trasmissiva di un canale.
H = banda del canale
V = n° livelli discreti

Th di Shannon:

$\text{bit rate} = H \cdot \log_2\left(1 + \frac{S}{N}\right)$

tiene conto del RUMORE!

$\frac{S}{N}$ = rapp. segnale-rumore

Wireshark stesso è un packet sniffer :)

- packet sniffing: i bad guys si posizionano tra due host (anche cablati) e "origliano" il traffico senza alterarlo, per poi estrarre info sensibili
- IP spoofing: "rubare" l'identità di qualcun altro tramite indirizzo IP.

Legge di Metcalfe: (che non credo vi serva ma...)

nella connessione P2P (peer to peer) un host può essere client o anche server.

«L'utilità e il valore di una rete sono proporzionali al quadrato del n° utenti.»

Dato n il n° utenti, il numero MAX di connessioni possibili è: $n_{max} = n^2 - n$.

~ legge quest'ultima largamente contestata, tant'è che se cercate "Metcalfe's law" su Google la prima domanda che Google suggerisce tra i risultati è:

- why is M.'s law important?
- why is M's law WRONG?

'non è stato possibile manco testarla con dati reali.

Fun fact 2: la prima parola ad essere trasmessa ~~se~~ in un very rudimental Internet, nel 1969, è stata: LO. Transitava dal UCLA al SRI e doveva essere in realtà «LOGIN», ma per qualche ragione l'host di SRI andò in crash nel ricevere la G.

L'ingegnere L. Kleinrock, che era lì a lavorare al progetto ARPANET (precursore di Internet, ndr), scrive in una celebre intervista che quel tentativo mezzo fallito ha un non-so-che di profetico: "LO!", come in "Lo and behold!" (letteralmente "ecco, preparatevi!" o qualcosa di simile), come a presagio della svolta enorme che si profilava nel futuro da quel piccolo esperimento - la nascita della gigantesca Internet.

siglette e cosette viste a lezione:

PDF pag. 91

FIBER to THE...

- fttb - ...home
- fttn - ...node
- fttc - ...cabinet
- ftts - ...street ≡ Fttc
- fttb - ...building

Forwarding: muovere il pacchetto attraverso i vari router

protocolli — INSTRADABILI (appletalk) / NON INSTRADABILI

Router: dispositivo che:
i) collega tra loro 2+ reti;
ii) alle volte cambiando protocollo;

Internet ha un generale approccio al servizio detto **best effort** - o come dice il prof, «che Dio ce la mandi buona». Nulla è di fatto garantito al 100%, ma tutti fanno del proprio perché funzioni.

Reti
- CIRCUIT SWITCH
  - TDM
  - FDM
- PACKET SWITCH
  - DATA-GRAM
  - VIRTUAL C. SWITCH

autonomous systems - parla IGP Internal Gateway Protocol - collezione di prefissi di routing IP gestiti da 1+ network operator. → per parlare tra loro usano EGP. Nel "backbone" si parla con BGP.
↳ collega vari AS su dei punti di contatto

NAMEX di NAPOLI ↑

Gli ISP forniscono dei **POP** Point of Presence agli utenti.

"192.168.1.###" non sono indirizzi IP pubblici, ma interni alla nostra rete.

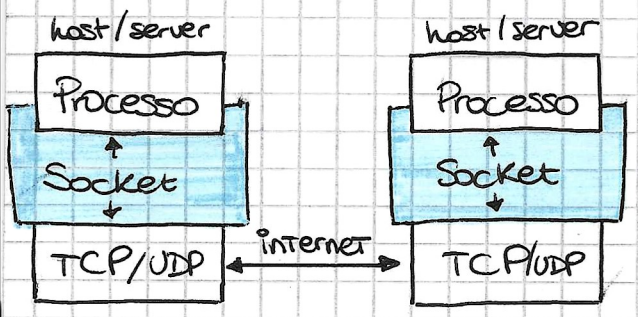
CAPITOLO 2: Application Layer

"il Layer imbrattato di sangue" (raga questo è Kill Bill, coraggio)

2 principali architetture:
- CLIENT-SERVER ~ e.g.: web server
- PEER-TO-PEER (P2P)

Network application - coppia di PROCESSI che si scambiano MESSAGGI;

: importante! è di processi che parliamo :



=> Socket (o API): interfaccia software tra l'application layer e il transport layer di un host.

Per identificare il processo ricevente serve specificare:
- L'INDIRIZZO DELL'HOST (IP);
- UN IDENTIFICATORE DEL PROCESSO RICEVENTE (NUMERI DI PORTA);

Protocolli a livello di TRASPORTO che la nostra app può usare, principalmente:

TCP — TRANSMISSION CONTROL PROTOCOL
- servizio orientato alla connessione (handshake and such);*
- trasf.to dati affidabile;

*SVC switched virtual circuit ~ proprio quello, comm. di pacchetto

UDP — USER DATAGRAM PROTOCOL
- SENZA FRONZOLI (minimale, leggero)
- CONNECTIONLESS (no handshakes, nulla)

NESSUNA garanzia che il messaggio arrivi :D beautiful.

- NO CRITTOGRAFIA! -

...a meno che non si usi anche TLS con TCP!
↳ Transport Layer Security

PROTOCOLLI Application Layer:
- HTTP - web | multimedia;
- SMTP - mail;
- FTP - file transfer;
- HTTP 1.1 - web;
- Telnet - accesso remoto;
- etc. etc.

} usano tutti TCP

checklist

PORTA	SERVIZIO
80	web server (default http)
25	SMTP mail server
20	connessione dati TCP - FTP
21	connessione CTRL TCP
110	POP3
53	DNS protocol
69	tFTP
68	DHCP replies
67	DHCP requests
123	NTP
520	RIP
	more at iana.org

HTTP - hyper text transfer protocol

default port: 80

- protocollo a livello applicazione;
- WEB!;
  - ↳ una pagina web è un insieme di oggetti (e.g.: file HTML[1], 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 il rispettivo URL[2].
- i browser web implementano il lato client di HTTP;
- protocollo STATELESS: se un client manda due richieste identiche nell'arco di qualche secondo, HTTP non ne ha alcuna memoria, e risponderà 2 volte con la stessa risposta;

[1] HyperText Markup Language
[2] Uniform Resource Locator

(TCP) Connessione:
- → persistente: tutte le richieste/risposte avvengono nella STESSA sessione TCP;
- → non persistente: ogni richiesta/risposta avviene in una sessione TCP a sé (ogni volta ne avvio una nuova);

> quale usa HTTP? persistente, di default. Ma può usare entrambe.

***

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

Messaggi HTTP

Richiesta

get[1] /dir/page.html[2] HTTP/1.1[3]

[4]
- host: www. ... (↵)
- connection: close (↵)[5]
- user_agent: Mozilla/5.0 (browser firefox)[6]
- : altre istruzioni HTTP

1: Request line - metodo
2: URL
3: versione http
4: HEADER
5: conn. non persistente

Risposta

http/1.1 200 Ok ← STATO

- connection: close
- date: mon 1 Jan 2000
- server: Apache/2.2.3
- Last-modified: (date)
- Content length: 6821 (byte)

(Intestazione)

Content type: text/html
* RIGA VUOTA *
((Corpo del messaggio blablablabla))

Web Caching

(o proxy server)

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

[BROWSER]: invia richiesta HTTP al proxy per, say, una pagina web;

[PROXY]: a) contiene una copia in locale della pagina richiesta → Risponde a browser;

o

b) non la contiene → la va a chiedere al server che dovrebbe averla, la riceve e oltre a recapitarla al browser ne salva una COPIA al suo interno;

Si usano per 1) Ridurre i tempi di attesa;*
2) Ridurre il traffico nel Web complessivamente.

*normalmente l'intensità di traffico è molto più alta al di fuori di una LAN (public Internet -> LAN) che non al suo interno

$$X \frac{\text{richieste}}{\text{secondo}} \cdot Y \frac{\text{Mbit}}{\text{richiesta}} / \frac{}{} \frac{\text{Mbit}}{\text{secondo}} = \text{fun fact esce un valore puro (e.g.: 0.15)}$$

Conditional GET: "server dammi l'oggetto solo se l'oggetto è stato modificato. altrimenti invio al client la copia che ho in cache."

e-Mail:

Componenti chiave del sistema di mail Internet:

1) user agents;
2) mail servers;
3) the SMTP (simple mail transfer protocol)

fun fact: qui avevo annotato un misterioso "129" a matita, senza aggiungere nessuna nota di contesto. Tuttora non so cosa mi rappresentasse, ma ora so che alla porta 129 c'è di solito UDP per Command & Conquer: Generals. :)

1: Lo USER AGENT è lo strumento che permette di leggere email, comporre eccetera (e.g.: Outlook);

2: Il MAIL SERVER è "il core dell'infrastruttura di e-mailing": la mia mailbox è contenuta in un MS e i messaggi vanno da lì ai MS contenenti le mailbox dei destinatari;

3: il Protocollo SMTP è il p. principale a livello App per la posta elettronica

Comandi SMTP che forse vorrete sapere:

HELO (hello), MAIL FROM, RCPT TO, DATA, QUIT.

SMTP è un push protocol: questo implica che per RICHIEDERE, non inviare, un messaggio di posta elettronica, c'è da usare qualcos'altro:

- HTTP;
- IMAP (internet mail access protocol);] permettono di: gestire cartelle, eliminare messaggi, etc

DNS

domain name System

PORTA 53 _dns:

"www.facebook.com" è detto hostname
=
69.63.176.13 è un indirizzo IP

Si tratta di:

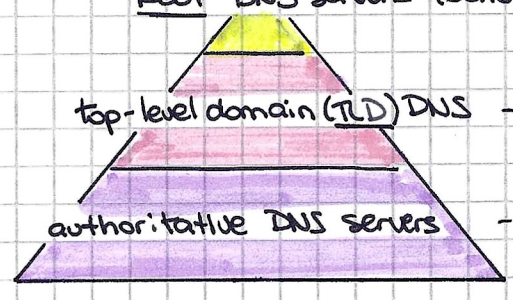
1) database distribuito;
2) protocollo app-layer per inviare query ai server DNS;

Si occupa di:

- tradurre gli hostname nei corrispondenti indirizzi IP

Servizi offerti da DNS: host aliasing - mail server aliasing - load distribution

Ci sono 3 livelli nella gerarchia di DNS:



IMPORTANTE!
Io in questo capitolo, avendo fatto principalmente riferimento al Kurose-Ross 8° Ed., NON ho menzionato alcuni protocolli applicativi quali: FTP, tFTP P2P, POP3 e forse pure l'IMAP. Ma il prof li chiederà!! (A.A. 2021-2022)

DNS caching: think web caching, ma applicata al sistema DNS.

Resource Records (RR): well, informazioni immagazzinate in questi database DNS.

(Name, Value, Type, TTL)
↳ time-to-live (in cache)

sono tutti nelle slides. Forse ho degli appunti. se sì, li integrerò. :) CP

[name e value dipendono da type]
e.g.: TYPE = A → NAME = nome host
VALUE = indirizzo IP

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 con sinonimo *name*
MX (not mexico)	host	valore c. di un MAIL server per host sinonimo di *name*

Allora dunque, ecco l'integrazione di quei protocolli / argomenti che sono nelle slides ma non nel Kurose Ross 8ª edizione! :)

- prima di tutto, nel caso lo avessi omesso:

qual'è la differenza tra HTTP 1.0 e 1.1?

↳ Ha delle cose in più, fondamentalmente. La più importante è forse la PERSISTENZA della connessione.

Poi introduce anche delle "meccaniche di caching" (che ha a che fare con le informazioni cached, non col suono dei $$$).

Ed altro, sicuramente, ma la cosa noteworthy credo sia fondamentalmente quella delle **connessioni persistenti**.

Telnet

- uno dei primi protocolli per TCP/IP;
- client - server;
- ora al suo posto spesso si usa SSH (Secure SHell); *

- serve a fare **collegamenti ad altri terminali** attraverso la rete (Port 23);

COME FUNZIONA: la dinamica client–server si svolge con il lato client che manda richieste al server telnet, e quello risponde.
Nello specifico, quando io client batto un tasto sulla mia tastiera, un carattere, questo viene spedito sulla rete e ricevuto dal server, che rimanda in risposta la "echo" del carattere. LOGIN REMOTO.

* **E QUESTO PERCHÉ!** Non è esattamente il massimo della sicurezza! E questo, di nuovo, perché, tutto sulla rete gira sotto forma di caratteri ASCII, nudo, leggibile, dati sensibili inclusi.

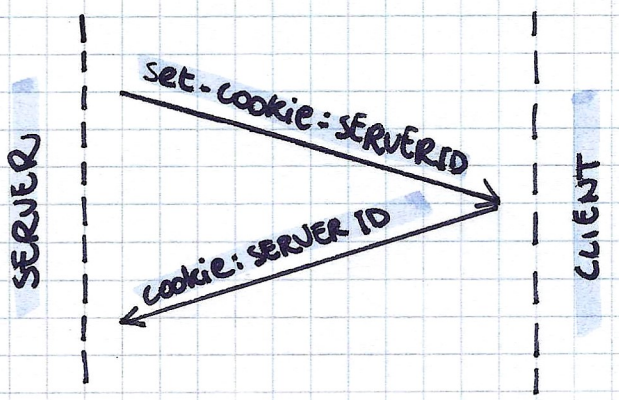
Brevi cenni su **VIRTUAL MACHINES**

Una MACCHINA VIRTUALE, essenzialmente, è un FILE DI UN COMPUTER.
Non nel senso che è un file dentro ad un computer (cioè sì, per forza, i file devono pur stare da qualche parte :), ma nel senso che è un file che RAPPRESENTA, RACCHIUDE, il sistema di un computer a sé stante.
Viene visto come fosse **un altro computer all'interno del nostro computer.**

•) Per fare ciò, il computer crea quello che si chiama un **Hypervisor**, ossia un programma che virtualizza degli host usando il mio hardware.
L'Hypervisor che il prof consiglia si chiama **VirtualBox** di Oracle ma io sono "gnorri" e uso VMware Workstation essendomi aggiudicata una licenza fortuitamente a tipo €1,50 anziché $200, si trovano. :)
Una vale l'altra, badate, vi servirà per il laboratorio!
Badate x2: VMware WS è gratis, la mia licenza è per l'edizione Pro!

Cookies (argomento interno ad HTTP)

- È un tipo di informazione che viene memorizzata lato client; Servirà in un secondo momento al server per ristabilire la connessione.



- ogni browser/sistema operativo archivia i cookies in un posto suo, non hanno un Path universale naturalmente;
- inizialmente, tutti i cookies erano semplici files di testo, ora la maggior parte è in formato SQLite (piccoli database);
- non tutti i siti web usano i cookies;
- i cookies sono facoltativi;
- i cookies possono essere disabilitati;
- i cookies sono un pericolo per la PRIVACY → profilano gli utenti! ↳ ovvero raccolgono informazioni sulle abitudini, comportamenti...

"Pubblicità COMPORTAMENTALE": consiste nel- l'analizzare l'attività in rete degli utenti per consentire alle aziende di fornire a ciascuno pubblicità mirate, ad hoc. → youronlinechoice.com

FTP

File Transfer Protocol

Protocollo, indovina un po', per il trasferimento di files!

- client-server;
- connessione TCP su PORTE 20 e 21:
  - 20 per i dati;
  - 21 per i comandi (controllo); → connessioni entrambe aperte dal CLIENT;
- operazioni di upload e di download;
- spesso oggi anziché usare FTP si scambiano file usando HTTP
  - questo perché, anche in FTP, TUTTO TRANSITA IN ASCII! pure username e pwd :)

Qualche comando FTP:

- USER - username;
- PASS - password;
- LIST - vedere i contenti della cartella;
- RETR - richiedere file;
- STORE - aggiungere file (upload);

Come HTTP, FTP ha dei codici di stato (tipo 404 = "Not Found")

- alcuni codici di stato FTP:
  - 331 = "username OK, password required"
  - 125 = "data connection already open; transfer starting"
  - 425 = "can't open data connection"
  - 452 = "error writing file"

da terminal: `ftp <indirizzo>` (FTP si usa a linea di comando nel 99% dei casi)

Nota: ci si può autenticare anche come anonymous

La questione PASV/PORT

In Windows 10, usare FTP può dare problemi a causa della questione modalità passiva/modalità port (PASV / PORT).

PASV e PORT sono entrambi comandi per la connessione dati →

→ sì perché uno degli aspetti da ricordare di FTP è che USA 2 PORTE perché ISTANZIA 2 CONNESSIONI TCP:
- una per i comandi (controllo);
- una per i dati.

quest'ultima, a volte, viene ostacolata dai FIREWALL. Quindi, se in principio bastava usare il comando PORT, successivamente, con l'avvento di NAT e FIREWALL, si è reso necessario aggiungere PASV, che non è altro che una modalità trasferimento dati compatibile con firewall.

tFTP — Trivial FTP

A differenza di FTP, tFTP:
- usa UDP (porta 69);

ha pochissime funzioni (da cui triviale):
- non conosce il concetto di DIRECTORY;
- non usa autenticazione;
- ha un utilizzo molto limitato

HA 2 MODALITÀ DI TRASFERIMENTO: ASCII (NETASCII) / binario (OCTET)

alcuni comandi TFTP:

- RR - Read Request;
- WR = Write Request;
- DATA = Dati;
- ACK = acknowledged;
- ERR = errore;

I pacchetti UDP inviati sono a lunghezza fissa = 512 Bytes;

La trasmissione si considera FINITA quando viene ricevuto un pacchetto < 512 Bytes.

qualche POP3 - cosa

- intanto POP3 e' un protocollo di posta elettronica e sta per Post Office Protocol;
- come IMAP, lavora in ASCII su 2 porte, TCP

COMANDI CLIENT - AUTENTICAZIONE:

- user
- pass

Risposte server:

+Ok
-ERR

- TRANSAZIONE:

- list: elenco messaggi (numero e dimensione msg)
- retr: retrieve msg in base al numero
- dele: elimina msg
- quit: esci

SNMP

Simple Network Management Protocol

// Normalmente, se c'è, la porta standard è UDP 161

// anche questo spesso bloccato da firewall a ragione, poi, deve arrivare dall' interno soltanto!

- usato per gestione reti -> monitorare e configurare disp.vi di Rete;
- consiste in un protocollo molto SEMPLICE che fa:
  GET - GETNEXT - GETBULK - SET - TRAP
  ↳ segnala un evento ad altre stazioni
  { agent (≈ server) = quello che ha il suo database di MIB: Management Info Base
  { manager
- usa le ACL - Access Control List (regole che indicano cosa può e non può fare un dispositivo);
- ogni oggetto MIB e' identificato da una serie di numeri, scritti tipo: (e' tipo una struttura nested)
  1.3.6.1.2.1.1.3 = sysUpTime. È macchinoso e scomodo, ma si usa.

non sono lacrime è un probabilmente misto di pioggia e energy drink per piangere che IRS si supera

CAPITOLO 3: livello di Trasporto

Oh, dunque. Cominciamo dall'angolo delle certezze.

Internet, a livello di trasporto, utilizza 2 protocolli, indovina un po': **UDP e TCP.**
E sappiamo pure che TCP è connection-oriented, con controlli di congestione e di flusso e di sicurezza eccetera, laddove UDP invece invia quel che c'è da inviare e Dio provvede.

Un pacchetto dati, a livello di trasporto, prende il nome di **SEGMENTO**
Ora, in alcuni documenti e RFC, i segmenti UDP vengono chiamati "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 Network Layer. In ogni caso, sia il Kurose-Ross che il prof stesso si riferiscono ai segmenti TCP-UDP con "SEGMENTI". Sooo.. :)

Genericamente, un segmento a livello di trasporto contiene 3 elementi:

i) # porta di origine / # porta di destinazione [16+16 = 32 BIT]

ii) altri campi di intestazione [variano a seconda di TCP-UDP]

iii) messaggio

PDU!
Sia UDP che TCP usano 4 PDU:
- Protocol
- Data
- Unit

unità di informazione o pacchetto scambiate tra host.

consegna tra host ed host (IP)	MULTIPLEXING → ← DEMULTIPLEXING	consegna tra processi in esecuzione su 2 sistemi periferici (TCP/UDP)

MULTIPLEXING a livello di trasporto: Raduna diversi dati da diverse Socket e li incapsula in un unico pacco di dati da spedire in Rete.
→ OGNI SOCKET DELL'HOST DEVE AVERE UN # PORTA.

DEMULTIPLEXING a livello di trasporto: esamina determinati campi del MSG ricevuto e decide a quale SOCKET del ricevente consegnarlo.

incapsulamento!
Cosa abbastanza importante: la cosa dell'incapsulamento. In pratica è un fenomeno che "colpisce" la gerarchia di Internet come segue:

Quando i dati arrivano dal layer N al layer N-1, il protocollo N-1 aggiunge un suo segmento di informazioni e manda il tutto al layer sottostante.

Sia TCP che UDP fanno MPX/DMPXing.

ogni SOCKET ha un ID che determina quale è il livello superiore a cui deve passare i dati.

lab
il comando shell "netstat" mostra tutte le socket attualmente in uso.

ora: il K-R a questo punto parla di UDP, ma il prof è partito da TCP.
Non lo so hombre. Iniziamo da UDP, che non c'è troppo da dire.

UDP user datagram protocol

certified protocollo senza fronzoli

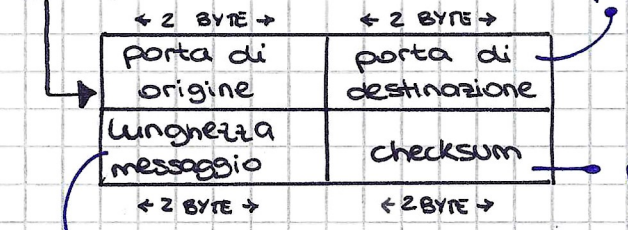
- connectionless → NON FA HANDSHAKE!
  - > per certi versi, a causa di ciò, UDP è "più veloce" di TCP, perché riduce notevolmente la CONGESTIONE;
- SENZA FRONZOLI: il pacchetto UDP ha intestazione della dimensione di 8 byte (contro TCP che ne ha 20!)

NOTINA SULLE PORTE:

- le porte numerate da 0÷1023 vengono dette PORTE NOTE, perché riservate a specifici utilizzi quali HTTP;
- le porte compresetra $1024 < (x) < 49552$ sono si dicono indirizzi EFFIMERI, generati randomicamente al momento della apertura della porta;
- le porte compresetra $49553 < (x) < 65535$ sono porte PRIVATE;

segmento UDP

header + messaggio



- il checksum, se i complementi a 1 sono andati a buonfine (e quindi anche l'invio/ricezione del pacchetto), dà 1111 1111 1111 1111

that's preddy much all :)

UDP non fa praticamente nulla oltre a MPX - DMPXing. #stayFronzoless
Per la maggior parte degli utilizzi nel application Layer (tipo HTTP) viene usato TCP perché fornisce molti ~~[redacted]~~ servizi legati alla stabilità ed affidabilità della connessione. UDP viene usato prevalentemente dove è richiesta RESPONSIVENESS (quindi tempi d'attesa brevissimi) e ci si può permettere una certa percentuale di packet loss senza tragiche conseguenze. TIPO DOVE? Tipo nello streaming video. O nel VoIP.
Anche se anche lì iniziano a usare TCP. #ioStoConUDP #UDPmatters

PRINCIPI di TRASFERIMENTO DATI AFFIDABILE - o RDT

Brevemente facciamo un excursus, con l'ausilio dei miei automini preferiti, nel discorso del trasferimento dati affidabile - una sorta di sistema, di protocollo che si rende necessario adottare considerando che in molti casi i protocolli affidabili si appoggiano su quelli sottostanti che però sono inaffidabili (si pensi a TCP e IP).
Quindi, in parole unpo' [redacted] povere, 4 modi or less di metterci 'na pezza.

→

RDT 1.0: la rosea premessa di un canale perfettamente affidabile;

- LATO MITTENTE

rdt_send(data)
pkt = make_pkt(data)
udt_send(pkt)

meaning

Rdt RICEVE LA CHIAMATA "send" PER INVIARE DATI, PERCIÒ:
- CREA UN PACCHETTO CON DATI + TINFO NECESSARIE HEADER etc.;
- INVIA IL pkt CREATO;

- LATO DESTINATARIO (O FORSE MEGLIO RICEVENTE)

rdt_rcv(data)
extract(pkt, data)
deliver(data)

meaning

Rdt RICEVE LA CHIAMATA 'ricevi" CON DEI DATI IMPACCHETTATI IN INGRESSO, PERCIÒ:
- ESTRAE I DATI DAL pkt;
- LI CONSEGNA AL PIANO DI SOPRA;

Rdt 1.0 e' as simple as that.
Non ha bisogno di nient'altro perché assume che il canale di comunicazione sottostante sia PERFETTAMENTE affidabile, no need di controllare nulla.

ora consideriamo la possibilità che ci siano ERRORI:

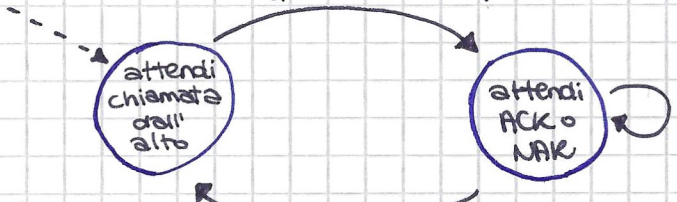
RDT 2.0: rilevamento errori

Dobbiamo fare in modo che il ricevente dia un feedback
error detection-feedback-retransmission

ACK NAK

- LATO MITTENTE:

rdt_send(data)
sndpkt = make_pkt(data)
udt_send(sndpkt)



rcv(rcvpkt) && NAK
udt_send(sndpkt)

rcv(rcvpkt) && isACK(rcvpkt)
*

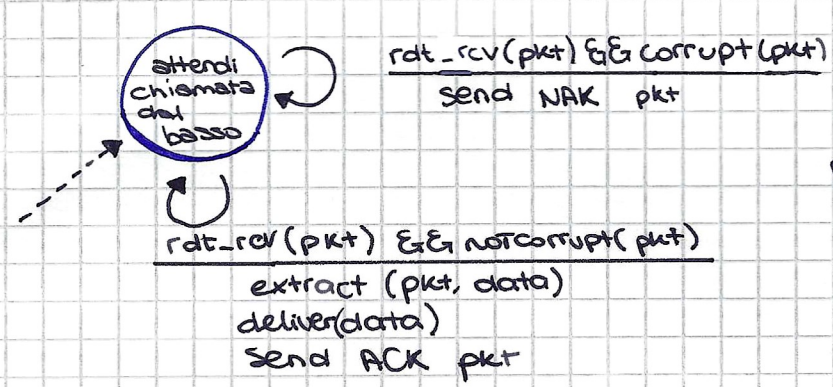
meaning

Rdt RICEVE LA CHIAMATA "send", IMPACCHETTA, SPEDISCE E SI METTE IN ATTESA DI FEEDBACK. SE ARRIVA UN NAK (errore rilevato), RISPEDISCE IL PKT, ALTRIMENTI* TORNA IN ATTESA DI UN NUOVO pkt DA INVIARE.
* ACK

protocolli che fanno questo tipo di lavoro si chiamano arq:
automatic
repeater
request

RDT 2.0 LATO RECEIVER

questi protocolli vengono chiamati Stop-and-wait
stop: STOP.
and: AND.
wait: WAIT.

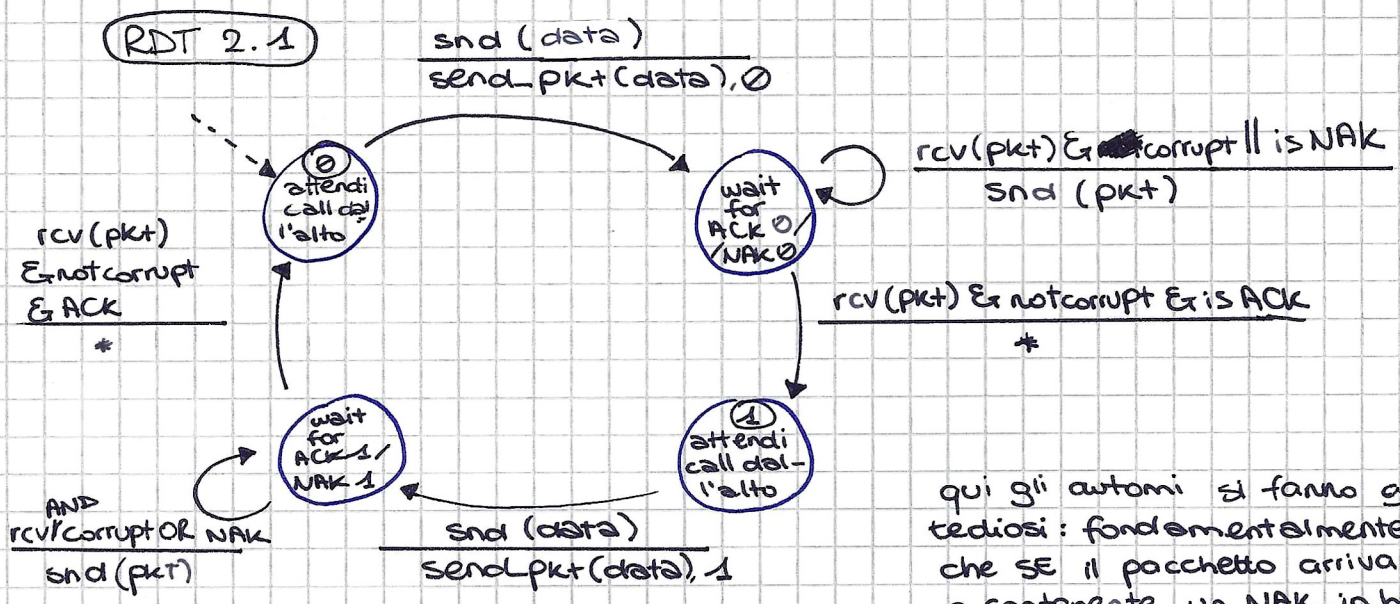


Beh qui 2 cose possono succedere, fondamentalmente:
- RICEVE IL PKT E NON È BUONO:
  - INVIA NAK IN RISPOSTA;
- RICEVE IL PKT ED È BUONO:
  - INVIA ACK IN RISPOSTA;
  - CONSEGNA PKT AL PIANO DI SOPRA;

simpel as that.

> e se il ACK\NAK pkt fosse corrotto? :V
> SOLUZIONE: aggiungere al pacchetto un nuovo campo: un sequence number. Basta anche solo alternare messaggi con SN "0" e "1".

RDT 2.1



qui gli automi si fanno grossi e tediosi: fondamentalmente vuol dire che SE il pacchetto arriva corrotto o contenente un NAK, in base a dove mi trovo rispedirò il pacchetto 0 o 1 anche se non ho capito se il ricevente ha capito o meno. Di modo tale che SE lui aveva capito il pacchetto con #0 e io gliela rispedisco con 0, lui si accorge che è una ripetizione :)

TLDR: in RDT 2.2, in pratica, invece di usare ACK / NAK usa ACK duplicati. Cioè invece di "non ho capito questo" dice "l'ultimo che ho capito è quello".
Il 3.0 introduce il timeout: è possibile che un pacchetto si smarrisca per strada, quindi introduce una sorta di COUNTDOWN timer per stabilire se e quando SMETTERE di aspettare un riscontro.

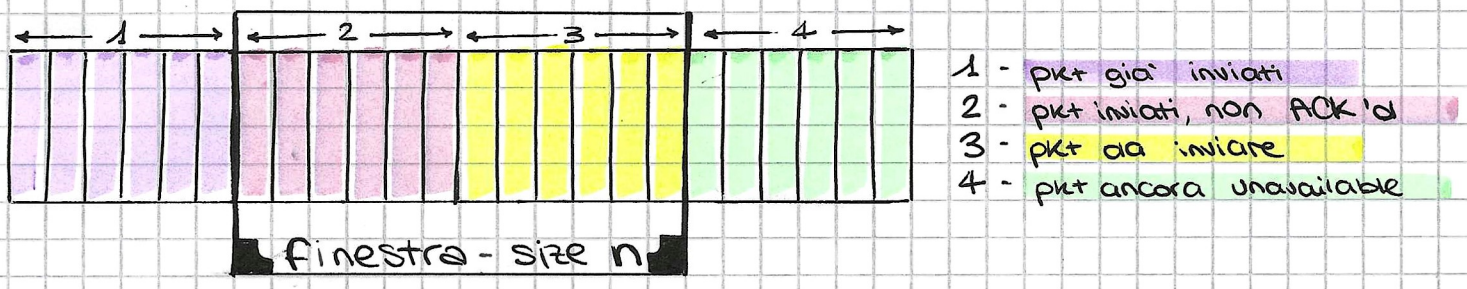
actually sono tutti stop and wait questi qui.

protocolli (PIPELINING - GO_BACK_N): perché mandare 1 pkt per volta? → time inefficient

e quanti pacchetti posso mandare safely senza fare casini?

→ finestra scorrevole (#n)

↳ non può essere grande a piacere! gestita da flow & congestion control.



1 - pkt già inviati
2 - pkt inviati, non ACK'd
3 - pkt da inviare
4 - pkt ancora unavailable

Riscontro Cumulativo: un riscontro per pacchetto con SN "n" implica "ACK per TUTTI i pacchetti precedenti = "fino a qui ho capito tutto".

Timeout: in caso di timeout, il mittente rispedisce TUTTI i pacchetti di tipo 2, inviati ma non confermati da ACK.

SR: Selective Repeat: "non ritrasmettere TUTTO QUANTO, ma solo le parti che non ho capito."
Dopo spetterà al receiver risistemarli nell'ordine giusto.

Ricevere un pkt fuori ordine è come non riceverlo.

TCP transmission control protocol — no ribbon for him >.>

- CONNECTION-ORIENTED: prevede che si faccia un HANDSHAKE prima di iniziare a scambiarsi info vere e proprie.
- SEMPRE POINT-TO-POINT: connette un host a un altro, fine.

MSS: max segment size - limite di dati che possono essere schiaffati in un SEGMENTO TCP.
↳ dipende dal MTU, max transmission unit che è la dimensione max dei dati trasmissibile nel link layer - E.g. Ethernet ha MTU 1500 bytes.

With TCP, two hosts are a company, and 3 are a crowd. KR page 227

flags TCP (6): sono bit, boole - ci sono pure CWR e ECE ma vedremo poi

- RST: reset, si usa in caso di "grave errore";
  NB l'ordine è CWR-ECE-URG-ACK-PSH-RST-SYN-FIN
- PSH: "il ricevente deve passare SUBITO questo segmento al layer sopra";
- URG: marca il contenuto del segmento come URGENTE;
- FIN e SYN come RST si usano per chiudere/aprire - connessioni;

altre variabili istanziate:

ISN - initial sequence number: NUMERO generato in modo PSEUDOCASUALE all'avvio di una connessione TCP, compreso tra $0 \div 2^{32}-1$ *, da cui partire.

legato a MSL

MSL - max segment lifetime per cui il segmento resta in vita in rete.

fattore di TIMEOUT - ci interessa che il TIMEOUT sia maggiore del RTT, trivialmente.

└ in TCP, il RTT viene preso ad ogni ACK.
Non può essere stabilito a priori - Si può STIMARE, quello sì lol

Lab: TSHARK non è altro che Wireshark ma a command line interface.

$$estimatedRTT = (1-\alpha)\,estimatedRTT + \alpha \cdot sampleRTT$$

estimatedRTT: Basically è una media pesata

$(1-\alpha)$: val. costante, normalmente $=0,125$

$sampleRTT$: misurato per ogni andata e ritorno

$$devRTT = (1-\beta)\,devRTT + \beta\,(sampleRTT - estimRTT)$$

devRTT: variazione RTT

$\beta = 0.25$

Alla fine, il RTT calcolato avrà valori convergenti in modo abbastanza stabile, e la deviazione standard valori abbastanza bassi.

lab

su Wireshark c'è un tool per il RTT_
tcp.analysis.ack_rtt

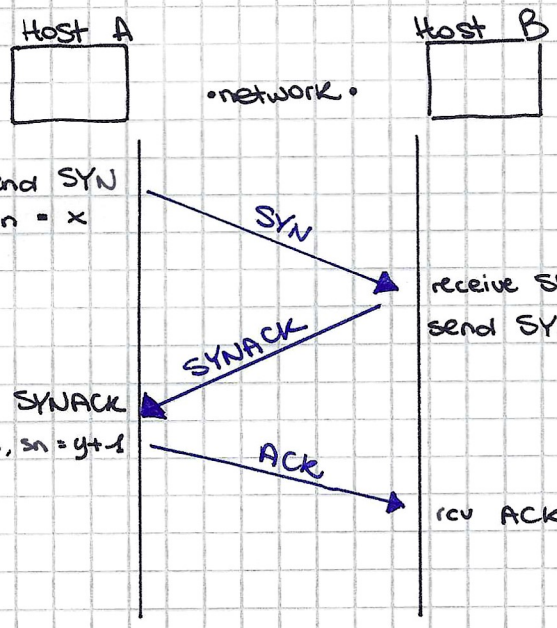
RTO ReTransmission Time Out:
= tempo entro cui la sorgente si aspetta di ricevere un riscontro.
NON può essere un valore statico predefinito, dipende da moltissimi fattori. Si calcola DINAMICAMENTE, basato sul RTT, di solito compreso tra $200ms < RTO < 60s$ +

- deve essere > RTTestimato, opportunamente, non troppo

$$RTO = estimatedRTT + 4 * devRTT$$

RcvWindow Buffer che serve ad evitare problemi al flusso dati.
Memorizza i byte ricevuti per poi passarli all'App. Layer.
Quando questo buffer è pieno, il RCV manda byte di controllo al sender, per dire che non può riceverne altri, sostanzialmente.
→ si liberano TOT byte, rcv manda un msg al sender per dire Ok ho spazio, mi aspetto [redacted] byte a partire da (ultimo ricevuto +1) e ho TOT spazio. → snd manda TOT byte.

"SYN - SYNACK - ACK" - 3-way handshake in TCP



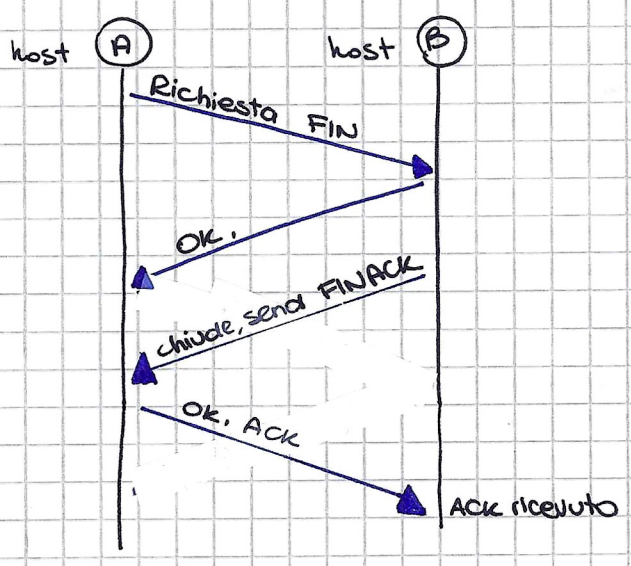
→ per aprire connessione TCP

→ si fa coi bit header TCP

→ "se avete capito questo, avete capito il TCP." -FM

→ per chiuderla si usa il bit FIN
→ 4 way handshake

4WAY HS PER CHIUSURA



CONTROLLO CONGESTIONE

Nb: mettere in Rete dei "pacchetti di controllo", a scopo di controllo congestione, e' fortemente sconsigliato - congestiona inutilmente la rete.

Il discorso sulle prestazioni e sugli scenari me lo risparmio, e' et nelle slides et nel libro in maniera moolto analoga.

2 approcci al C.Control:
- End-to-End
- Network-assisted

TCP & congestion control:

congWin - finestra di congestione
quantità di dati riscontrati da un host durante la connessione.
Vincolo: TCP non può inviare dati ad un rate > della finestra di congestione.
la finestra di congestione cambia dimensioni dinamicamente, in base appunto al livello di congestione.
ALGORITMI per cambiare il ritmo di invio in funzione della congestione:

i) **AIMD**: incremento additivo, decremento moltiplicativo;
ii) **SLOW START**: per ogni ACK ricevuto, RADDOPPIO la window;
iii) **FAST RECOVERY**: non lo vedremo questo :D

TCP TAHOE/RENO, non utilizzati

threshold limite tra decremento molt.vo ed incr. additivo:
valore < TH → si adotta SLOW START
valore > TH → si adotta CONGESTION AVOIDANCE (AIMD)
In linea teorica l'algoritmo si dovrebbe STABILIZZARE e convergere su un certo range, all'atto pratico non funge così. VEDERE TCP CUBIC

lab

Nmap - network scanner - 2 tecniche:

- PORTSCAN
- PING SWEEP

↓ usato per

port monitoring, analisi di sicurezza, info gathering, attacchi

→ attività potenzialmente losca/equivocabile: da usare solo nelle reti in cui siamo autorizzati!

Nmap utilizza delle "fingerprints" per raccogliere informazioni sul target.

ping sweep: fare un ping per vedere se l'host è acceso.

Eventualmente è possibile utilizzarlo con GUI, 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".

KEKW - nmap ha incorporato uno SCRIPT ENGINE in LUA, una serie di script per trovare vulnerabilità e affini. Any time, just type `nmap --script -updatedb` per fare l'update del DB degli script disponibili.

Mi pare poi che nmap si possa usare anche all'interno di WireShark, o forse sto confondendo la UI con quella di ZenMap...

"Modalità promiscuous": qualcuno sta usando un packet sniffer. → how do you detect it?

ncat è un tool open source a riga di comando utile sostanzialmente a collegarsi ad un altro host in remoto. Negli appunti ho scritto che è tutto nelle slides, ma io queste slides non le sto trovando lol. Anyway it's very cool.

Anche GNS3 è immenso, ma lo vedremo poi :)

→ trovate!
Sono all'indirizzo web:
https://computerscience.unicam.it/marcantoni/
tabella "didattica" → "INTERNET, RETI e SICUREZZA" sotto a Slide Lezioni.
Si chiamano "nmap.pdf".
Oppure chiedile a me, che tanto le ho.

CAPITOLO 4: Network Layer

Further info:
- Kurose-Ross page ≈ 314 / 775 (.pdf), namely 3003
- slides: chapter 04.1 (→ 04.8)
- lezioni: siamo a quella di Nov. 03, 2021

Ho perso la parte in cui il prof fa questa distinzione, forse pure perché NON la ha fatta, ma il libro dice: il NETWORK LAYER può essere diviso in DUE COMPONENTI INTERACTING:
- il data plane;
- il control plane;

} questa cosa per qualche motivo la ricordo dal corso di Cloud Computing, nor.

Nel data plane vengono trattate tutte le funzioni "pre-router", incluso il tradizionale IP forwarding che vedremo soon enough.

FORWARDING vs ROUTING
- forwarding: azione svolta dal router, che consiste nel trasferire un pacchetto da un'input link interface all'opportuna output L. interface;
- routing: azione svolta network-wide che comprende tutto il processo per determinare il PATH da percorrere per i pacchetti da mittente a destinatario.

N.B.: protocolli di routing ≠ PROTOCOLLI ROUTABLE! (o instradabili)

Su Internet NON ESISTONO (e non possono esistere) indirizzi uguali.

IP internet protocol

- Definisce i MODI in cui instradare i pacchetti.

IPv4 datagram [NB: la PDU a livello di network si chiama DATAGRAMMA, come si accennava qualche paragrafo fa parlando di UDP e ambiguità.]

Non ricordo minimamente dove siano, NDR, ma esistono tool e metodi per convertire IPv4 in IPv6 e vice versa. Google it :^)

Formato datagramma IPv4:

32 BITS

VERS	HL	SUCTYPE	datagram length
16bit identifier		Flags	13bit offset
TTL	upper layer protocol	header checksum	
32-bit SOURCE IP address			
32-bit DESTINATION IP address			
opzioni eventuali			
Dati			

HL = header length SUCTYPE = service type
TTL = time-to-live
NB: il time-to-live non e' un indicatore di "hh:mm:ss", non e' tempo inteso così, e' un CONTATORE a N che viene decrementato di 1 ogni volta che un pacchetto arriva in un router, per evitare situazioni di loop nella rete - se il TTL = 0, allora il pacchetto viene dichiarato morto, F in chat.

- IPv6, come datagram, e' molto simile a IPv4, con qualche differenza nei dati contenuti e le DIMENSIONI DATAGRAMMA = 128 BITS.
- interfaccia di rete: punto di connessione tra host e router. E.g.: scheda bluetooth, ethernet etc. Può avere più IP address.

Evoluzione degli Indirizzamenti

maledetto il giorno in cui mi sono detta ma sì facciamo lettering che ci vuole

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:

CLASSE	START ADDRESS (/bin)
Classe A	0.0.0.0 / 0000 . 0000 . 00000000 . (...)
Classe B	128.0.0.0 / 1000 0000 . 00000000 . (...)
Classe C	132.0.0.0/ 11000000 . 00000000 (...)
Classe D (multicast)	224.0.0.0/ 1110 0000 . 00000000 (...)
Classe E (reserved)	240.0.0.0/ 1111 0000 . 00000000 (...)

NOTA: la struttura dell'indirizzo IP nel 2-layer classful era: (net-ID).(host-ID) la prima metà identifica la rete, la seconda l'host.
*
Quindi la CLASSE è associata alla RETE!

- Piece of cake! Questo può palesarsi all'esame. Anche se il 2-layer classful è discontinuated. Basta ricordarsi i numeri (0, 128, 192, 224, 240) o essere svelti con le conversioni dec-binario. Visti i primi 4 bit, il gioco è fatto! :)

1984 →

INDIRIZZAMENTO A 3 LIVELLI CLASSFUL:
Immagino si dica "a 3 livelli" perché a questo punto la STRUTTURA dell'indirizzo diventa:
(net-ID).(SUBNET-ID).(HOST-ID)
Di questo indirizzamento non trovo davvero nulla in giro, non sul K-R, non nelle slides, non nel web...

1993 →

CIDR - Classless Inter-Domain Routing
Eliminate le classi (CLASSLESS, DUH);
Indirizzi gestiti in modo EFFICIENTE per fare ROUTING.
IP = <prefisso, suffisso> dove
prefisso = RETE
suffisso = HOST CONNESSO ALLA RETE
questi due campi devono essere separati con qualcosa, un'altra stringa di bit; a questo serve la seguitida MASCHERA di RETE - 32 BIT disposti in X "1" e Y "0"
eg: 11111111.11111111.11111111.00000000. Operando un AND LOGICO possiamo estrarre la parte RETE da quella HOST.

Indirizzi IP particolari:

- **0.0.0.0** : indirizzo di avvio stack TCP;
- **127.0.0.1**: <u>loopback</u> localhost - permette di comunicare con la propria stessa macchina come se fosse un altro host in rete;
  Perché dovrei volerlo fare? Boh a scopo di test ad esempio;
- Net_ID. TUTTI 1: indirizzo di BROADCAST, mando pacchetti a TUTTA la rete contrassegnata con Net_ID;
- Net_ID. TUTTI 0: identifica la RETE e BASTA (o la sottorete)
- 255.255.255.255 (TUTTI i BIT A 1): broadcast locale;

• SUBNETTING •

Namely, dividere la rete raggruppando TOT indirizzi host e formando <u>LOGICAMENTE</u> una SOTTORETE.

a tal proposito, anche per verificare: conti degli esercizi, vedere → VLSMcalc.net - it's free :)

→ QUANTO E' lungo subnet_ID? DIPENDE:

- ha lunghezza <u>FISSA</u> in caso di SUBNETTING STATICO;
- ha lunghezza <u>VARIABILE</u> in caso di SUBNETTING DINAMICO; ← Variable Length Subnet Mask

Esempio: risaliamo alla rete dato l'indirizzo

193.205.92.150/25 → "barra 25" - è una notazione che descrive la subnet mask, significa "ci sono 25 "1" e il resto "0" (32-25 = 7 "0".)

quindi dobbiamo fare l'AND LOGICO:

	193	205	92	150
indirizzo	11000000.11001101.01011100.10010110			
mask	11111111.11111111.11111111.10000000			

∧ and logico

=

$$(11000000.11001101.01011100.10000000)_2 =$$

$$= (193.205.92.128)_{10} = \text{la rete!}$$

*dopo ci tocca riconfigurare la rete da capo ao!

Ci verrà anche chiesto di fare il procedimento inverso, allo scopo di PROGETTARE RETI. Sul raddoppiare la barra: è buona norma "raddoppiare": requisiti nel decidere la subnet mask, tipo se abbiamo 62 host, NON usiamo una subnetmask /24, che offre 64 indirizzi. Altrimenti ci andrà stretta all'aggiungerne solo 2!+

LAN Local Area Network

allora qui parliamo di LAN e di datalink PDU qo.in di... uhh... **Capitolo 5: datalink (?)**

K-R pag. 449 (pg. 460/775 del PDF)
Slides ch. 05 "Datalink e LAN"
Lezione A.A. 2021/22: Nov. 10, 2021

[definizione di IEE] Sistema di comunicazione che permette ad apparecchiature indipendenti di comunicare tra loro entro una 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";
- TUTTE le macchine condividono LO STESSO CANALE fisico di comunicazione;
- È una network - ECONOMICA; - VERSATILE per modifiche; - FACILE per fare manutenzione; - CAPACE di sopportare grossi carichi di lavoro; - DURATURA nel tempo (anni, se ben progettata e configurata).
- Le trasmissioni sono SEMPRE DI TIPO BROADCAST.

Downsides:

- tutti gli host collegati vanno identificati;
- bisogna definire un modo per farli comunicare;

Altre gatte da pelare:
- flessibilità → compatibilità tra host di fatte3e diverse;
- modularità;
- espandibilità, con la P in caps;
- affidabilità;
- gestibilità; ← a questo proposito ci sarebbe il SNMP, simple network mgmt protocol

Cosa serve in una LAN?
- beh, uno o più host;
- software di rete (normalmente legato all'OS);
- le NIC (Network Interface Card, wired/wireless);
- delle API;
- NETWORK HUB, o CONCENTRATORI (e.g.: switch);
- Cablaggio strutturato (cavi, antenne, etc.);

2 MACRO CATEGORIE in cui è divisa:

- **HARDWARE**: topologia, cavi, tutto ciò che la può rendere wired/wireless etc.
- **SOFTWARE**: SW applicativo, tutti i protocolli

LIVELLO 2: datalink - struttura il messaggio e lo passa al Physical layer (dal layer appena sopra, namely un messaggio IP), e lo spezzetta in FRAME delle dimensioni richieste dal layer sottostante, + aggiunge al messaggio una FCS
↳ Frame Check Sequence

La LAN è definita nello STANDARD EIA/TIA 568 e ISO/IEC 11801: questo ultimo in particolare è lo standard del cablaggio. E questo potrebbe essere una domanda d'esame. EIA/TIA568.

IEEE 802.11 è il working group che gestisce lo standard WIRELESS LAN (pure questo lo chiede);
l'IEEE 802.3 invece si occupa dello standard Ethernet;

Il layer datalink si divide in:
- **LLC** Logical Link Control
- e
- **MAC** medium access control (quello che tiene conto del mezzo fisico che c'è al di sotto)

È anche il livello che funge da "arbitro" nella gestione del canale (che è forse ciò che ho già scritto), perché con un canale UNICO condiviso in BROADCAST il problema che sorge è: COME REGOLAMENTARE L'ACCESSO AL CANALE?

Dovrò inventarmi un SISTEMA che:
- TROVI GLI INDIRIZZI di tutti gli host connessi alla mia network;
- TROVI PER CIASCUNO il proprio **indirizzo MAC**;

MAC PDU: "frame" - come è fatto un PDU mac:

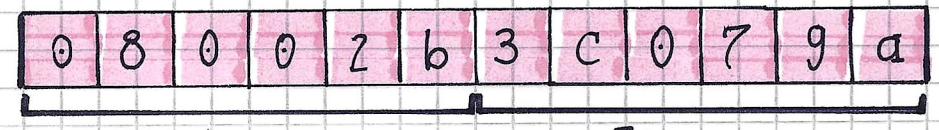
- DSAP/SSAP: destination/source SAP (Service Access Point)
  Sono: campi principali del Frame, univoci a livello mondiale;
- il payload: i dati, duh;
- la famosa FCS: è un CRC* su 32 bit, un sistema di controllo integrità;
  *Controllo a Ridondanza Ciclica

FRAME ETHERNET: come sempre c'è un header + payload

nell'header c'è un **preambolo**: 7 byte tutti composti da "01010101..."
+ l'8° byte che, invece, **termina con 11** → fondamentalmente serve per dire "ora parlo io, e quello che segue è il mio messaggio". MUCH DRAMA
poi gli indirizzi di destinatario & mittente, 2 byte per specificare la lunghezza, poi il payload e il CRC.

INDIRIZZO MAC

- 48 bit (6 byte)
- formattati in **6 coppie esadecimali:**



3 byte

OUI (o VENDOR CODE)
↳ organization unique identifier

Standardizzati dal IEEE, sono codici associati ai PRODUTTORI DI SCHEDE DI RETE (tipo Cisco)

3 byte

I meno significativi sono una numerazione progressiva decisa dai produttori.

il modo più usato per scriverlo credo sia "08:00:2b:3C:07:9a"

Scheda di Rete

divisa in

- **hardware**, interfaccia di rete
- CPU + memoria [illegible]a! Lavora indipendentemente dal PC, i dati non vanno nel processore!

- L'indirizzo fisico dev'essere: UNICO nella LAN!! in Internet, poco importa dei duplicati, ma in LAN sì.

NDR: il prof lascia intendere di essere molto judgemental ↑ al riguardo.

- L'indirizzo fisico normalmente è statico - alle volte può essere riassegnato.

rappresenta:

- **UNICAST** = una singolo host;
- **MULTICAST** = un gruppo di host;
- **BROADCAST** = TUTTE le stazioni (ff:ff:ff:ff:ff:ff);
  - N.B.: se è BROADCAST, la frame viene SEMPRE analizzata

indirizzi di gruppo: servono principalmente a fare (NEIGHBOR DISCOVERY)

2 modi di impiego:

- SOLICITATION: la stazione richiede un servizio e manda un messaggio MULTICAST con l'indirizzo del servizio; le stazioni che lo offrono rispondono;
- DISCOVERY: le stazioni che offrono un servizio inviano a cadenza regolare un messaggio MULTICAST per informare del servizio offerto; cioè SOLICITATION ma al contrario.

NEIGHBOR DISCOVERY:

- su UNIX: `ip neigh show`
- su Windows PowerShell: `get -netNeighbor`

NDP è un Network Discovery Protocol - ma è configurato per IPv6

E REITERO: in LAN si usano i MAC, non gli indirizzi IP!

ARP address resolution protocol → traduce l'indirizzo IP in indirizzo fisico (MAC)

passa da indirizzo fisico a IP ← **RARP** reverse address resolution protocol

DHCP

dynamic host configuration protocol

- help - serve per configurare gli host.

NAT è una sigla che il prof usa spesso, sta per Network Address Translation

Problemi sorti nella decisione su "COME DEV'ESSER FATTO UN SISTEMA DI ADDRESSING" (non può essere casuale, no?)

LO SO VE LE FDEGA RELATIVAMENTE BST...

- duplicazione degli indirizzi IP: conseguenza, questa, del fatto che qualcuno si assegnava un IP messo a caso (creando (!) l'ambiguità);
- riassegnazione degli indirizzi IP: ad esempio, come fare la riassegnazione nello spostare un ufficio da una rete ad un altra?
- spreco di IP non usati;
- far sì che ci sia una qualche interfaccia per vederli tutti;

da un'occhiata al sito web "showmyip"!

Fun Fact: gli indirizzi IPv4 sono stati tutti utilizzati, li abbiamo terminati :)

duh

Usando TCP si possono usare 2 PROTOCOLLI che si occupano di host config:

- BOOTP: funziona coi device diskless! (thin client), USA UDP
- DHCP: come BOOTP ma ulteriormente sviluppato

→ IMPOSTANO IP ADDRESS IN MODO AUTOMATICO!

↓

introduce il parametro di LEASE: param. di TEMPO durante il quale l'host può usare l'indirizzo IP che DHCP gli ha assegnato; NDR penso sia stato aggiunto per evitare sprechi di indirizzi.

Linux ha SAMBA ↑ di Microsoft

→ DHCP configura:

- IP;
- maschera di sottorete;
- Gateway e DNS (e a volte neanche quelli, NDR);
- altri parametri;

lab

NETBEUI

Netbeui (o NETBIOS) è un vecchio protocollo di rete, che funziona solo in locale. WINS - DNS di NETBEUI.

→ L'addressing può avvenire in 3 modalità:

i) assegn. manuale o RESERVATION: legata al MAC address;
ii) assegn. automatica: lease normalmente INFINITO;
iii) assegn. dinamica: lease molto breve (~ 1h);

→ E in 4(+1) fasi:

i) discover;
ii) offer;
iii) request;
iiii) ack;
(v) release; -> quando non uso più l'indirizzo, namely ho spento il PC

È buona norma che non si interrompa la connessione scollegando il cavo o mettendo in standby per andarsene, per che così facendo il DHCP non sa che è stata interrotta. (= spreco)

→ Banalmente, per essere raggiungibile, il DHCP server DEVE ESSERE a 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 a disposizione, un sistema Windows ha IP 169.254.0.0/16

IP e la Frammentazione dei Pacchetti

...pare il titolo di un cartone :)

Nell' header IP, ricordiamo (non so perché) la presenza dei campi:

- **PROTOCOL:** (8 bit) è letteralmente la chiave per poter leggere il payload; perché specifica di che protocollo si tratta. duh :)
- **CHECKSUM HEADER:** parity check → si prendono 16 BIT dello header e si fa il completamento a 1 della somma di tutti i 16 BIT dell'header. Deve portare tutto a 1 alla fine :)
- Ricordiamo pure che il 2° gruppo (di 32 bit) del header IP è riservato a:
  " **Frammentazione: identificazione, flag, fragment offset** "

La frammentazione, se ricordate, è quel procedimento fantastico in cui un frame a livello datalink viene appunto frammentato in frame delle dimensioni richieste dal mezzo di trasmissione sottostante. (definito dal MTU)
↓
max transmission unit!

Il RIASSEMBLAGGIO del frame viene effettuato solo una volta che questo ha raggiunto la destinazione, anche se nel tragitto passa per dei canali che possono avere MTU più grandi.

Fun fact: c'è qualche mattacchione appassionato di videogames (e [redacted] reti a quanto pare) in giro per il mondo che, in qualche impostazione God knows where, cambia la dimensione di frammentazione dei PDU Ethernet da 1500 a 1473 (odd number, letteralmente) perché PRESUMIBILMENTE in questo modo si ottiene una maggiore velocità di traffico dati, quindi riduzione del ping in ms.

Tanto gli schiaffi li prendono lo stesso ♡
Anyway cambia poco e niente, dice il prof.

«Tsk... Vuole winnare contro un pro... →

questo lo disse Giovanni Mucciaccia, che ci crediate o meno

campi di frammentazione:

- Primi 16 bit: identificativo;
- 3bit successivi: Flags
  - i) 1° BIT riservato – a cosa? ai fatti suoi che ne so ♡
  - ii) 2° BIT se = 1 → " non si può frammentare " (errore ICMP);
  - iii) 3° BIT se = 0 → " questo frammento è l'ultimo o l'unico del datagramma";
- 13 bit: offset di frammentazione

Problematiche:

- maggiore overhead (impiego di risorse non necessarie) di trasmissione;
- è facile organizzare ATTACCHI DoS mandando tantissimi pacchetti costringendo host a impiegare molte risorse;

questa funzionalità propria di IPv4, in IPv6 non c'è :D

lab

Esistono modi per determinare la MTU più piccola dato un certo percorso.
ping -f -l 153.205.92.2 on wireshark :)

> Siccome ogni router attraversato dal pacchetto ne modifica il TTL, ogni volta va modificato anche il checksum. (penso intenda ad ogni hop).

Poi vedremo l'ICMP.

per vedere su WireShark la frammentazione bisogna assicurarsi che nelle preferenze il campo "Reassemble.." sia, immagino, non spuntato. (sotto "protocols" e lì dentro sotto "IPv4".)

R • O • U • T • I • N • G
instradamento

- Fare in modo che i pacchetti arrivino a destinazione;
- ≠ forwarding (inoltro)!

tabella di instradamento: è un "database" memorizzato nel router / nella macchina, dotato di una metrica dei costi in termini di tempo, per poter valutare quali reti convenga attraversare da host a host.

Il router lavora al livello 3 (IP).

Tabella-wise, possiamo fare QUESTA DISTINZIONE tra ROUTING e FWDing:

routing	vs	forwarding
Regole per SCRIVERE le tabelle		Regole con le quali il pacchetto viene inviato a determinate porte d'uscita (cosa che tipicamente si fa consultando le tabelle)

per stampare la tabella di routing, su cmd prompt Windows c'è il comando ROUTE PRINT. non in caps ofcourse.

Nelle tabelle di instradamento, per ogni sottorete è elencato:
- il relativo NetID;
- l'indirizzo del router di inoltro;

Per usare gergo dei database, i RECORD nelle tabelle sono composti da:
- l'indirizzo della rete di destinazione;
- la netmask;
- l'interfaccia su cui inoltrare;
- indirizzo del next hop;

Ad ogni percorso è associata una METRICA, un costo che può essere in tempo approssimato o in N° di hop a seconda del protocollo di routing usato.

- 2 PROCEDIMENTI PER ROUTING: (mi pare, ndr.)

DIRETTO → se l'host mittente è nella stessa rete, inoltro direttamente al destinatario;

INDIRETTO → se sono su reti diverse, dovrò fare l'inoltro su un ROUTER INTERMEDIO; → next hop!

→ Come faccio a sapere se due host sono nella* sulla, forse stessa rete?
Risalgo alla rete dati gli indirizzi IP mittente e destinatario, se i risultati corrispondono si può procedere con instradamento diretto.

instrad. diretto: si usa il MAC address - coinvolge layer 1+2;
(o più in generale l'hardware address)

instr. ad. indiretto: si usa l'indirizzo IP del router - coinvolge layer 1+2+3;

"Badate bene," dice Loreti: la tabella di ROUTING è SEMPRE presente in TUTTE le macchine che hanno IP, che siano host o router.

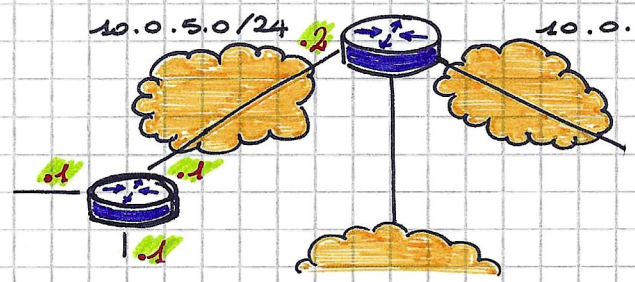
+ di solito contiene SEMPRE ALMENO il next hop migliore, più altre cose.
++ il next hop è configurato in una SOLA direzione, i percorsi sono tipicamente **asimmetrici** - la direzione opposta può scegliere un'altra route :)

ROTTE

- **STATICHE**: configurabili dall'admin di rete;
- **DINAMICHE**: da reperire con routing protocol;
- **DIRETTE**: legate alle interfacce del router;

Negli end systems e in gran parte dei router deve SEMPRE essere presente una **default route**, per quando non sa dove andare.

NOTA: in un grafico del genere:



quei ".1", ".2" and so on, stanno a specificare l'interfaccia di rete - Visto che, normalmente, i router ne hanno più di una, è buona norma che si specifichi a quale si fa riferimento.

Come si fa la tabella?

MASK	ind. destinatario	ind. next hop	flag	Ref.count	use	interfaccia

- MASK: per fare AND logico
- ind. destinatario: tipo 124.0.0.0
- ind. next hop: n° altro IP
- flag: flags varie
- Ref.count: n° utenti che stanno usando il percorso
- use: n° pkt trasmessi al receiver
- interfaccia: nome interfaccia

ICMP — Internet Control Message Protocol

È incluso nello stack TCP/IP.
È richiesto per implementazioni standard di IP.

Usato da IP per inviare **messaggi di errore.**
↳ e per farlo, ICMP a sua volta usa IP :)

header ICMP: (32 BIT)

type	code	ICMP head. checksum
8bit	8bit	16 bit

(seguono dati)

- type → formato
- code → specifica l'errore
- checksum → solito complem. a 1 di cose dell'header

valore numerico, e.g.: 0 = echo reply
5 = redirect
4 = source quench

TIPI DI MSG — segnalazione errore, e.g.: destinazione non raggiungibile
↘ richiesta informazioni

Tipo, quando si fa il comando "ping", si prepara un pacchetto ICMP di tipo 8, e la risposta è un ICMP di tipo 1.

Un altro tipo di messaggio era quello del TIMESTAMP (serviva ad ottenere data ed ora esatta degli host); ora non si usa più granché in quanto si usa NTP per SINCRONIZZARE gli orologi con un orologio di riferimento.

Il TTL varia anche a seconda del sistema operativo.

caso TRACEROUTE:
- fondamentalmente, il primo pacchetto è un MESSAGGIO ICMP con TTL settato a 1. Arriva al primo router, il TTL → 0, ICMP manda indietro un msg di errore per TTL = 0;
- il pacchetto seguente ha TTL settato a 2, così arriva esattamente al router 2 dopo;
- e così via sostanzialmente, acquisiamo info di tutti i router.

——— Torniamo a parlare di AS, border router, r. esterni, IGP EGP etc. ———

ASBR = Router di frontiera: deve avere un'istanza SIA dei router interni alla rete, SIA di quelli esterni ?

A volte sono state adottate delle POLITICHE di AUTENTICAZIONE per ragioni trivialmente di sicurezza.

protocolli di Routing AS:
- Inter AS:
  - BGP
- Intra AS (interni):
  - distance vector
    - RIP / RIP2;
    - IGRP / EIGRP;
  - link state
    - OSPF / OSPF2;
    - IS-IS;

PROTOCOLLI DISTANCE VECTOR:

- Trovano il percorso migliore, unico fattore discriminante il (n° hop.)
- ogni X secondi, il mio router invia ai vicini la sua routing table

ALGORITMO RIP

RIP era classful - RIP2 invece usa mask (classless).
• RIP vs RIP2 •

> Router riceve messaggio da Router C

↳ tabella del (router C):

Rete 2	4 hop
Rete 3	8 hop
Rete 6	4 hop
Rete 8	3 hop
Rete 9	5 hop

> aumento subito di 1 tutti gli hop reti per Router C
(dopotutto e' arrivato a me, io sono quel 1 in più)

(router C)

Rete 2	5 hop
Rete 3	9 hop
Rete 6	5 hop
Rete 8	4 hop
Rete 9	6 hop

> a questo punto guardo la (MIA) tabella, supponiamo sia:

rete	n° hop	dal router:
Rete 1	7	A
Rete 2	2	C
Rete 6	8	F
Rete 8	4	E
Rete 9	4	F

> (aggiorno la mia tabella con awareness del Router C:)

Rete 1	7	A	→ invariato;
Rete 2	5	C	→ era 2 ma ora e' aggiornato a 5;
Rete 3	9	C	→ acquisito;
Rete 6	5	C	→ sostituito perché 5 < 8;
Rete 8	4	C	→ questo poteva anche rimanere E, 4 = 4;
Rete 9	4	F	→ questo non conviene cambiarlo, 6 > 4;

BREVE ✨ CARRELLATA ✨ di ALGORITMI DI INSTRADAMENTO

→ Cosa fa il livello di Rete? [quindi siamo tornati al livello di Rete]

→ Deve trasportare pacchetti da source a destinazione, senza preoccuparsi di COME questo avviene - di quello si occupano i ROUTER (dude ROUTER si traduce letteralmente con "instradatore", tant'è!)
↓
si occupano di determinare il PERCORSO da fare.

- Riguardo il FORWARDING DIRETTO:
  all'interno dello stesso mezzo fisico, sappiamo che ci possono essere più reti IP; è compito, poi, del router, occuparsi dell'instradamento anche lì.
  MA!
  in casi del genere, è preferibile utilizzare una sola rete per mezzo fisico piuttosto che più reti, anche se sì, si può fare diversamente.

prot. ROUTABILE: (che ricordiamo essere ≠ prot. di ROUTING)
protocollo che PUÒ ESSERE UTILIZZATO PER APPLICARE ALGORITMI DI ROUTING.

routing + forwarding: insieme combinati sono, necessari per l'operatività di una rete.
fondamentale a tale scopo la tabella di ROUTING.

Tranquilli non ho dimenticato della ✨ carrellata ✨ ora arriva.

Su tutti gli host su cui c'è stack TCP/IP, che siano router, end system, w/e,
i) ci sono delle tabelle di routing;
ii) esiste ALMENO un protocollo di Routing;
plus, di default, il mio host è sempre collegato ad un
DEFAULT ROUTER, o anche DEFAULT GATEWAY
o anche FIRST-HOP ROUTER
o anche ROUTER DI PRIMO RILANCIO
... altrimenti la mia rete sarebbe isolata.

Behold now, la carrellata. (dopo un bunch di repetita...)
ALGORITMO di ROUTING: trovare il miglior percorso da SOURCE a DESTINATION. ovvh.
= con il minor COSTO, che sia in tempo, in #hops, o w/e

→ TEORIA dei GRAFI -- cosa che il prof ha "dismissato" perché in altri corsi viene affrontata molto bene, namely in AeSD. Ma fosse per me ve la dovreste ri-sorbettare.

- Grafo: un grafo $G=(N,E)$ è un insieme di N nodi ed E edge (archi).
  - per ogni arco (x,y) ci sono i nodi x e y ed un costo $c(x,y)$.
  - se il grafo è NON ORIENTATO → $c(x,y) = c(y,x)$
  - y è ADIACENTE o VICINO ad x se $(x,y) \in E$ (quindi se collegati da un arco)

- il COSTO di un percorso è la somma dei costi:
  $c(x_1,x_2) + c(x_2,x_3) + c(x_3,x_4) + \dots + c(x_{n-1},x_n)$

2 CATEGORIE di ALGORITMI:

statici: basati su tabelle manuali, percorsi che cambiano raramente;

dinamici: la topologia, i percorsi, i costi della rete possono cambiare per cui devono adeguarsi ai cambiamenti (hanno dei TIMER che triggerano la riconfigurazione)

altre 2 categorie in cui si dividono 'sti algoritmi:

sensibili al carico: a seconda del carico della rete, cambia la metrica

insensibili al carico: quelli che usiamo oggi per la maggior parte (RIP, OSPF, BGP...) → ...actually! si sta pensando di iniziare a utilizzare quelli sensibili al carico, per "migliorare la QoL." etc...

aaaltre 2 categorie:

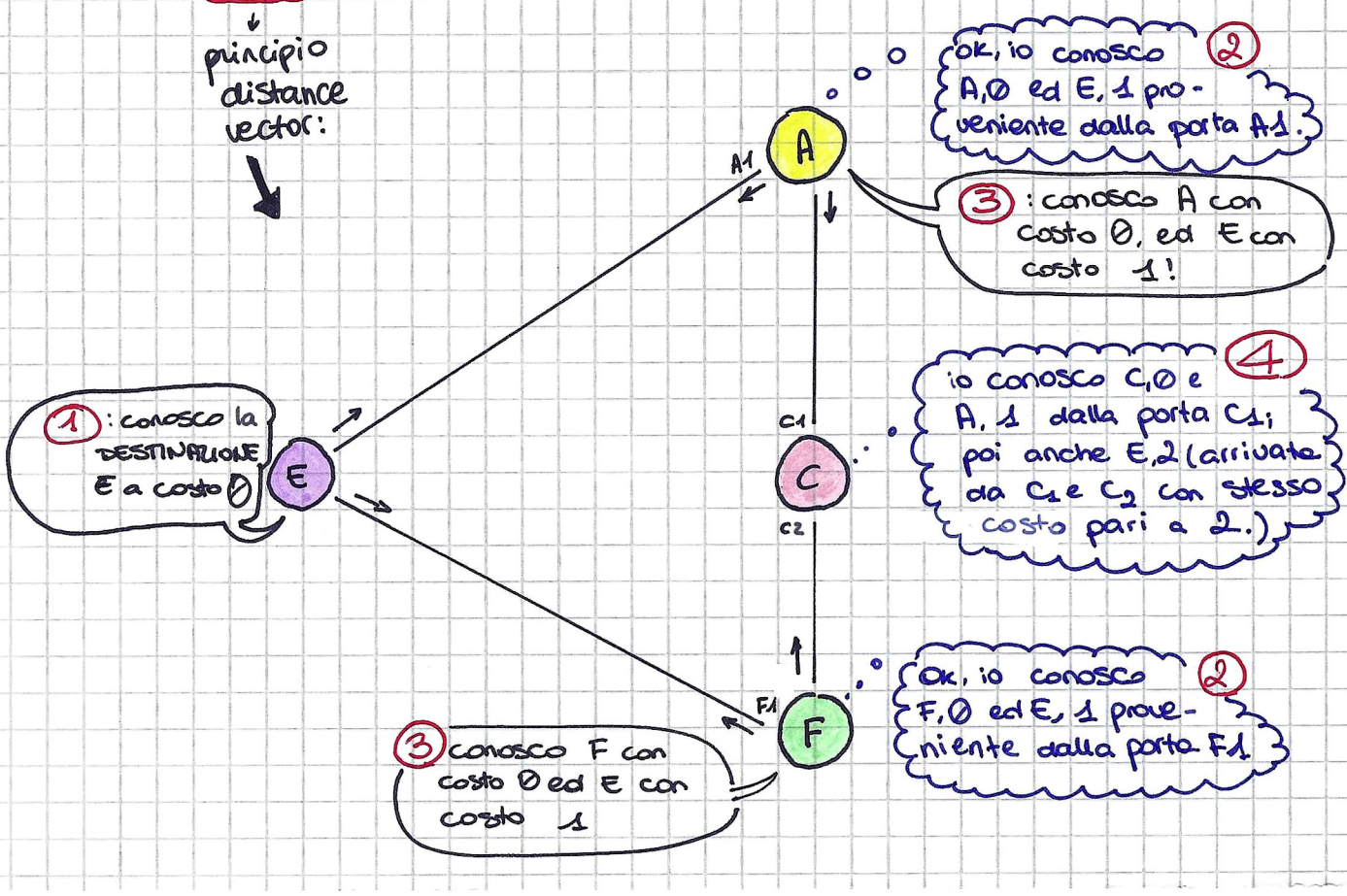
Globali: utilizzano tutti gli algoritmi a disposizione per conoscere l'intera mappa della rete - per fare ciò, fanno quella che si chiama operazione di FLOODING della rete

→ LINK STATE

Decentralizzati: qui non c'è una ~~conoscenza~~ conoscenza globale della rete, soltanto dei VICINI (più eventualmente il DEFAULT GW)
→ ogni router ha conoscenza di sé stesso

→ DISTANCE VECTOR

principio distance vector:



PRO's and CONS ...

- ... di **distance vector**:
  - ➕ semplici da implementare;
  - ➕ supportati da sistemi con poche capacità, et elaborativa et di memoria;
  - ➖ problema del LOOP (anche se per essere precisi, questo si sistema con il metodo SPLIT HORIZON);
  - ➖ convergenza lenta (dipende da # host);
  - ➖ non possono usare molti hop (es: RIP ne ha 15 max);
- ... di **Link State**:
  - ➕➕ ho una mappa completa della rete; non suscettibili a errori;
  - ➖ consumano TANTISSIMA banda;
  - ➖ non facilissimi da configurare;
  - ➖ richiedono capacità elevate;
  - si usano in topologie DENSE di ROUTER

Fun fact: la rete dell'università usa il protocollo link state OSPF.

- I pkt che il router invia per comunicare agli altri le proprie info si chiamano **hello pkt** → "hello protocol"

→ LINK STATE - algoritmi utilizzati (**SSSP** single source shortest path):

- algoritmo di Dijkstra;
- algoritmo di Bellman-Ford;

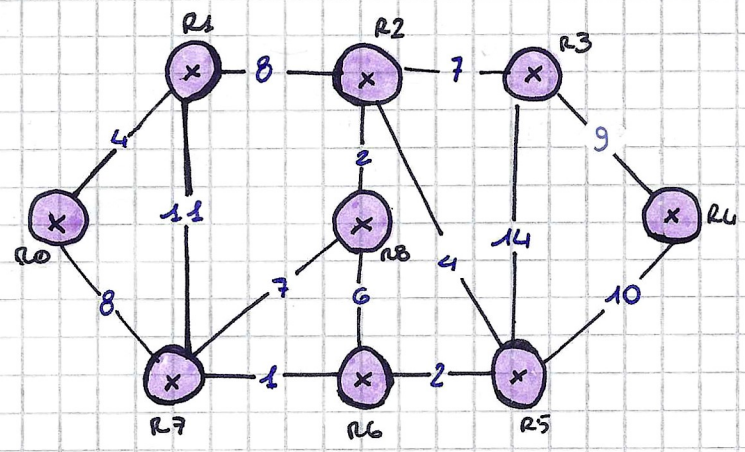
↓ parmi di capire che siano simili a meno di una differenza sostanziale: La metrica. Il Bellman-Ford può gestire anche pesi negativi. o costi, come preferite. Dijkstra, solo positivi.

Pare e dico pare che i navigatori satellitari, per trovare il percorso ottimale in strada, utilizzino proprio Dijkstra. Ora, io non ho verificato, quizás...

Ora io, da qualche parte, avevo segnato che questi algoritmi sono ad una via. NEL SENSO CHE!!. nel senso che: la strada fatta da A a B NON è necessariamente la strada ~~fatta~~ che verrà fatta da B ad A. Non trovo dove, né se esiste un modo per dire meglio 'sta cosa. Forse la scrissi qui, ben venga if-so.

L'algoritmo di Dijkstra, now, se non ve lo ricordate o non lo sapete, uhhh, okay ~~freccia~~ idk facciamolo.

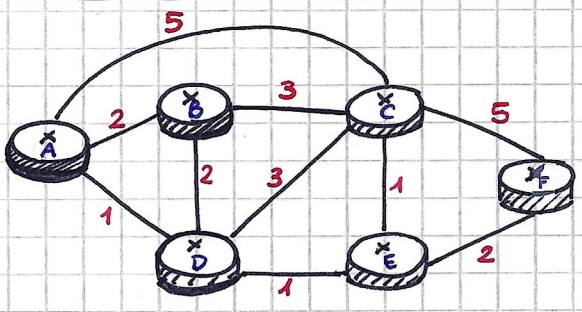
questa è la rete da considerare:



I. parto da R_0; ~~scopro~~ R_1 con C=4 e R_7 con C=8; allora ok io con Dijkstra me ce taglio ok?
II. posso raggiungere R_2, R_8, R_6:
III. QUAL È IL PERCORSO MENO COSTOSO da R_0 a R_2, R_6, R_8?
IV. R_2 si accede (?) via $R_0 \to R_1 \to R_2$ a costo 4+8 = 12;
V. R_8 si accede via $R_0 \to R_1 \to R_2 \to R_8$ a costo 4+8+2 = 14;
VI. R_6 si accede via $R_0 \to R_7 \to R_6$ a costo 8+1 = 9;
VII. posso raggiungere R_3, R_5, ora:
VIII. QUAL È IL PERCORSO MENO COSTOSO da R_0 a R_3, R_5?
IX R_3 si raggiunge via $R_0 \to R_1 \to R_2 \to R_3$ a costo 4+8+7 = 19
X R_5 si raggiunge via $R_0 \to R_7 \to R_6 \to R_5$ a costo 8+1+2 = 11
XI infine, 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

Per ogni nodo (quindi router nel nostro caso), sappiamo il percorso più conveniente da R_0 per raggiungerlo.

$R_0 \to R_8$ si potrebbe anche fare passando per R_1, ma costerebbe 4+11+7 = 22 :)



Percorsi di minor costo da A a tutte le direzioni:

passo	N	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	-	-	-	-	-

tabella di instradamento di A:

destinazione	verso	costo
F	D	4
C	D	3
B	-	2
E	D	2
D	-	1

Dunque, è una notte buia e tempestosa nel livello 2 ISO-OSI.
"un Venerdì 13 dell' anno 0 del Paradiso".

Torniamo a parlare di LAN.

- in LAN, nel momento in cui un host nella rete trasmette, esso diventa proprietario di TUTTO il mezzo trasmissivo → questo era un problema di sicurezza
- Ethernet, non ce l'ho ben chiara 'sta cosa, oltre ad essere il nome della "tecnologia" che abbiamo visto, pare sia anche il NOME dell'ALGORITMO che ha "risolto" il problema dell'arbitrare gli accessi Broadcast degli host alla LAN. "non indifferente" agli albori di LAN! che tutto era un broadcast, i.e.

Precisazione doverosa per star certi che sappiamo cosa trattano questi standard, che all'esame di solito ce li troviamo:

IEEE 802

definisce i protocolli STANDARD.
particolarmente importanti sono:
- 802.3 Ethernet
- 802.11 WIRELESS LAN e questo vai tranquillo che lo chiede 100%

EIA/TIA 568

definisce il cablaggio strutturato. poi vedremo meglio i cavi etc.

Addentrarsi nel wireless in questo momento è come darsi una martellata sui denti.

Mi sa che qui mi ripeterò, ma: a livello datalink, la PDU è detta **TRAMA**, o **FRAME**.

Non c'è uno standard ben preciso sulla struttura di un frame, ma in linea generica ci sono:
- un **trailer** con sender e receiver;
- il **payload**;
- il **CRC**, che ricordiamo stare per Controllo Ridondanza Ciclica.
  ↳ alternativamente si usano anche Controllo di Parità e somme di Controllo

2 tipi di collegamento:
- BROADCAST: più stazioni sullo stesso mezzo trasmissivo
- PUNTO-a-PUNTO: può essere cablata o wireless → in caso di p2p cablata il cavo usato in questione dovrebbe essere il CAVO CROSSOVER, giusto perché a volte è domanda d'esame.

protocolli ad accesso multiplo: Richiesti in reti ad accesso multiplo: LAN, reti via satellite...

Categorie:
- suddivisione del canale in TDM/FDM (vedi foglio # 4) - channel partitioning protocol
- protocolli ad accesso CASUALE (RAP) - e.g. Slotted ALOHA
- protocolli a rotazione (taking turn protocol) - eg. polling, token-passing

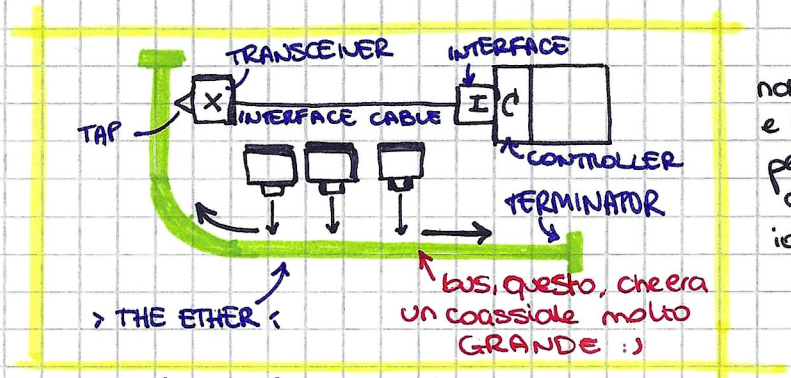
ndr: abbiamo capito che se vogliamo scrivere bellino dobbiamo stare lontani dagli anelli.
sì perché dovete sapere che ~~il~~ questo Magnum Opus è scritto su un coso del genere:

Tecniche di **allocazione** del canale trasmissivo:

- (STATICA): il mezzo viene partizionato a mo' di TDM/FDM;
- (DINAMICA): tutte le frequenze del mezzo trasmissivo vengono assegnate di volta in volta al singolo host
  ↳ ora, nel caso dinamico turn-based bisognerà inventarsi un ALGORITMO che li gestisca!

- La MAC PDU ha una lunghezza di $64 < x < 1518$ BYTES.
  ↳ aka il FRAME
- Ci sono diversi tipi di FRAME Ethernet

IEEE 802.3, copia quanto più fedele mi esca dello Sketch originale:



nar 'sto abbozzo risale al 1976 e lo ha disegnato Metcalfe in persona. il mio guaio è che, dopo aver letto della omonima "legge", io Metcalfe non riesco a prenderlo sul serio, oh to have opinions...

- c'è un SOLO BUS;
- è NON DETERMINISTICO - nessuna regola per stabilire quando parlare;
  > quando e dove trasmettere sul bus?
- ad oggi, Ethernet è la tecnologia Ethernet cablata più diffusa!
  Nel tempo, si è passati da una topologia a BUS ad una a stella;
  ↓
  SWITCHED ETHERNET
  ↓
  vedremo che si è passati all'utilizzo del cosiddetto HUB - che poi in realtà il bus rimane comunque all'interno di questo hub.

TOPOLOGIA A BUS:

- interruzione del bus → interruzione di tutta la rete;
- aggiungere un end System alla rete comporta un grosso intervento, con "FERMO" del sistema;
- una frame immessa nel bus si propaga ogni volta su entrambe le direzioni del bus -- uhh, okay?

ALOHA - protocollo per topologie a BUS indovinate dove è stato ideato lmao

- consente trasmissione in modo CASUALE
  → spesso quindi le frame mandate in rete insieme andavano in COLLISIONE!
  Collisioni che potevano essere parziali o totali a seconda dei TIMESLOT occupati da ciascuno;
- facile da realizzare, ma sicuramente poco efficiente
- poi in realtà ci sono modi per sincronizzare il tutto, ENTER CSMA-CD ...

CSMA-CD → dove CSMA sta per:

Carrier Sense } → "prima di parlare, ascolta" → COLLISION DETECTION

Multiple Access } → più host connessi, ciascuno può trasmettere

↓

"CD". infatti i protocolli che hanno sistemi di CD si chiamano CSMA-CD

mentre invece i CSMA-CA sono quelli con COLLISION AVOIDANCE !

COLLISION ~~AVOIDANCE~~

La rule-of-thumb, nei CSMA, è:

"Prima di poter trasmettere, ascolta.
E mentre trasmetti, ascolta ancora!"

... ma neanche questo scongiura collisioni!
Metti che per caso inizino a parlare 2 ENA Systemsin contemporanea: Collidono una volta, si fermano, e qui lo scenario diventa simile a quando 2 persone si parlano una sopra l'altra, quindi si fermano per poi riattaccare col "OPS SCUSA VAI PRIMA TU" - solo che anche nel dire ops scusa si parlano sopra di nuovo, e di nuovo si fermano e così via fino a che forse non ci scoppia il morto /s

- Come si scongiura questo deadlock:
  si tenta di dire, al momento della collisione, agli host coinvolti di aspettare un tempo T generato per ciascuno RANDOMICAMENTE

  - al momento della collisione viene emesso un SEGNALE SPECIALE ("jamming") ad una certa frequenza

  - SE LA COLLISIONE dopo un tempo T RICAPITA ANCORA si raddoppia l'amount di tempi T di attesa da scegliere a caso!
    nel senso, se alla prima collisione avrei aspettato un tempo nell'insieme $T = \{0, 1\}$, alla 11ª sceglierò tra $T' = \{0, 1, 2, 3\}$
    così diminuisce la prob.tà che 2 ES peschino lo stesso t.
    Questo meccanismo si chiama "Exponential Backoff"

DOMINIO:

- di BROADCAST = parti di rete in cui il broadcast riesce a raggiungere TUTTI gli host: nel caso di una LAN, il Dominio di BC è limitato dal ROUTER

- di COLLISIONE = parti di rete dove c'è possibilità che si verifichi una collisione

Tempo per rilevare una collisione $a = \frac{\text{lunghezza collegamento}}{\text{lunghezza pacchetto}}$

- Nel frame, nel header, nel preambolo (vedi foglio # 27) se ricordi c'era l'ultimo degli 8 bytes che terminava in "11". Ecco, quello ha un nome: SFD.
  Sta per Starting Frame Delimiter. :)

SPECIFICHE FISICHE ETHERNET:

dai torna a studiare

- **10 BASE T**: 10 = Banda base (non è bluepilled) è basato BASE = uhh BASE. T = "Twisted Pair" (il doppino intrecciato, ma che poi è una pessima choice of words, "intrecciato")
  - trasmissione 10 mbps in BANDA BASE;
  - sia coassiali che UTP;*
  - hub, switch, bridge;
  - Ungheria MAX cavo: (100 m) → sospetto lo chieda all'esame...
  - CONNETTORE RJ45 - affidabile ed economico, e facile da implementare, al contrario del BNC che unisce i coassiali

* la cronologia ha visto il combio progressivo andare così:
COAX THICK > COAX THIN > DOPPINO (CAT 3)

GNEGNEGNE
NON MI SENTO NEL MIO PAESE IMPONGO I CAVI COASSIALI E C'AVRÒ RAGIONE
#CoaxHoax
si scherza raga probabilmente è davvero meglio l'UTP. si scherza relativamente.

- la **100 BASE T** differisce da 10BASET in:
  - beh la velocità;
  - ha 3 tipi di possibili connettori:
    - 100BASE-T4 doppino 4 coppie;
    - 100BASE-TX doppino 2 coppie;
    - 100BASE-FX fibra ottica

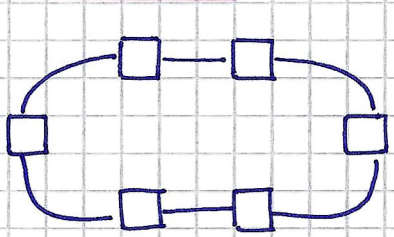
  also è RETROCOMPATIBILE con la 10 BASE-T

- GIGABIT ETHERNET (802.3Z):
  - pensa un po', 1Gbps;
  - NON retrocompatibile

IEE 802.5 - Ethernet TOKEN RING: topologia ad anello, cablaggio a stella.

Questi standard Ethernet sono TUTTI a tecnologia CSMA-CD

TOPOLOGIA AD ANELLO

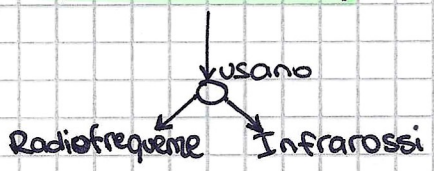


- semplifica coordinamento accessi
- più facile trovare guasti
- se si guasta un elemento, la rete è persa

TECNOLOGIA TOKEN RING: dove il "token" è un frame particolare che "dà il permesso agli host per parlare"
...in Ethernet NON C'È MODO di stabilire chi può parlare! quando è il turno di chi.
LOGICAMENTE le Token Ring sono ad anello, ma FISICAMENTE conviene disporle come "stella a doppio anello" e usare un MAU**
** Media/Multistation Access Unit

topologia FDDI: fiber distributed data interface (in disuso)
Sempre TOKEN RING, sempre cablata ad anello o stella, velocità elevate. usava il DOPPIO anello, 2 anelli indipendenti → RETE AUTORIPARANTE

Livello fisico: Reti wireless!



- devono avere un **access point** collegato alla rete cablata;
- standard 802.11: reti wireless – a onde radio
  – a infrarossi
- l'arbitraggio del canale trasmissivo si può fare con VARI ALGORITMI (non c'è uno standard preciso);
- **velocità di trasmissione:** fino a 300Mb/s;
- Moltissimi problemi del caso:
  - PROPAGAZIONE onde radio;
  - occupazione frequenze;
  - inaffidabilità;
  - potenza ridotta;
  - sicurezza;

802.11a - 5GHz
802.11b/g - 2.4GHz
DOVE b = max vel 11Mbps
g = max vel 54Mbps

Più l'host è distante dall'access point, meno c'è velocità (~50 mt)

interferenze:

- strutturali: in edifici di cemento armato, la propagazione è difficile
- su distanze elevate (e.g. ponti radio): attenzione a dove posizionare le antenne. Edifici, vegetazione, nebbia, possono disturbare il segnale;
- dovuto ad altre sorgenti: Bluetooth, interfoni perneorati -> molti apparecchi usano la f = 2.4GHz

Dice il prof: la 5GHz è usata per aspetti "marginali, ma di grande rilevanza per noi"

con la 5GHz ↵ succede meno

Modalità di collegamento:

- peer 2 peer (senza passare per l'access point, dipende dalla scheda di rete)
- client – access point;
- multiple access point + ROAMING

mobilità != nomadismo

mobilità: l'utente cambia AP MENTRE si sposta, senza interruzioni di servizio

nomadismo: vedasi uno che lavora in molti posti diversi, casa, ufficio, ufficio2, casa2, bar, ...

- BRIDGING con ANTENNA DIREZIONALE

AREA di SERVIZIO: l'ambito dove interviene l'access point ("bolla di frequenza")

SECURITY

livelli di sicurezza wireless:

- open system – o con chiave di autenticazione condivisa
- WEP – wired equivalent privacy;
- (quella "attuale") WPA/WPA2 wifi protected access;

BADATE! non è la stessa cosa che dire "possono entrare tutti" vuol dire APERTA, ma non è necessariamente la stessa cosa ☺

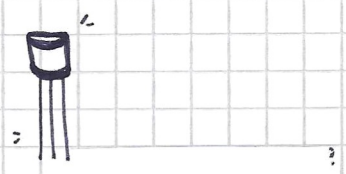
↓

Le WPA e WPA2 offrono:
cifratura dati (standard autent.ne 802.1x)
integrità dati (?)
protezione da attacchi Replay

↳ "attacco Replay": catturare pacchetti scambiati da client a server per risalire alla CHIAVE DI AUTENTICAZIONE

WPA
- personal – private
- enterprise – ha bisogno di un SERVER DI AUTENTICAZIONE (Radius, TACACS)
  Eg: Radius contiene le credenziali degli utenti

wireless sensor network – i sensori, sì, tipo quelli per l'Arduino



piccoli
economici
limitati in elaborazione e trasmissione

Caratteristiche delle reti di sensori:
- molti sensori
  - per monitorare fenomeni CONTINUAMENTE
- si dicono Reti ad Hoc
- dialogano tra loro tramite antenna
- e convergono tutti con i propri dati nel SINK (server centrale)

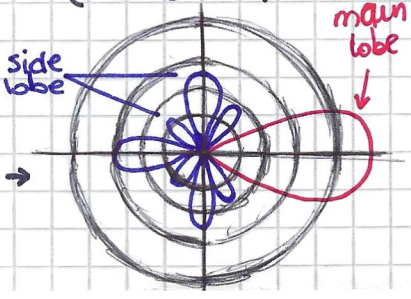
- standard ZigBee (802.15.4) RL-WPAN wireless personal area nw
  - a basso consumo, buono per sensoristica
  - opera su f = 2.4 GHz

* Effective Isotropic Radiated Power

propagazione delle onde:

potenza di radiazione: la normativa tecnica ETS 300-328 impone di NON irradiare con una potenza EIRP* > 100 mW (≈ 20 dBm)

propagazione
- DIREZIONALE: guadagno > 1
- OMNIDIREZIONALE: guadagno = 1



ogni antenna ha il suo diagramma di radiazione → per rendersi conto di come meglio orientarla

gradi di protezione IP: (che non credo sia l'Internet Protocol)

È un indicatore formato da 2 CIFRE:

i) protezione da oggetti solidi 1÷6 dove ad esempio 1 = prot. da oggetti > 50mm (credo)

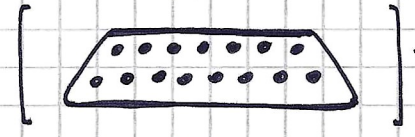
ii) permeabilità dell'acqua 1÷8 dove 8 = resistente a immersioni continue

sottostandard 802.3 il CABLAGGIO

vengono definiti come segue

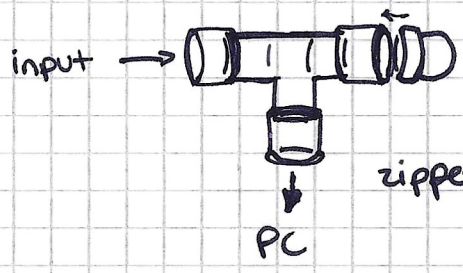
- 10BASE5 . il cosiddetto COASSIALE thick (500m)
- 10BASE2 . il cosiddetto COASSIALE thin (185m)
- 10BASET . il DOPPINO TELEFONICO (100m)
- FOIRL . standard asincrono fibra ottica x connettere : repeater (1000m)

"thicknet" - c'era un bus di coassiale e, per collegare le macchina, c'era bisogno di un TRANSCEIVER (costoso:)
- sulla scheda di rete, per collegarsi al transceiver, c'era una interfaccia AUI fatta così:



tipo una SCART

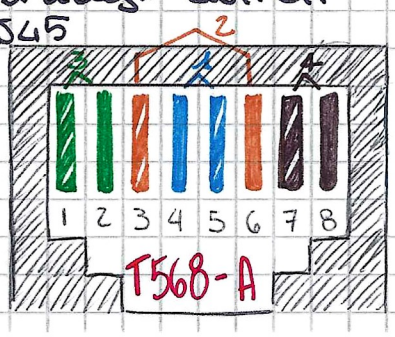
Nella "thinnet" non è più servito il transceiver ma non si potevano utilizzare PROLUNGHE

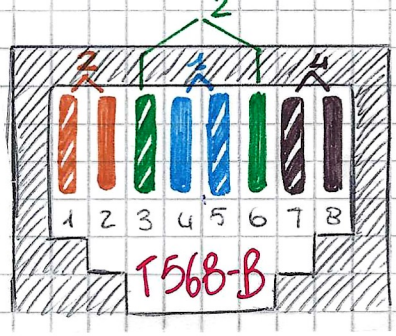


Doppino telefonico - classificazioni (le famose CAT) ups dopo c'arrivo

> Non ci si collega più al bus ma a degli SWITCH
> non più BNC -> attacco RJ45
> MAX 100m

USO di COLORI -> BINATURA
È indifferente quale si segua purché siano entrambi A-A o B-B
se lo fai A-B fai un cavo CROSS





Doppini:

- UTP doppino non schermato;
- FTP FOILED TP (schermatura con foglio di alluminio + collegamento a massa);
- STP + schermo locale;
- S-UTP / S-FTP dove S = "Shielded";

Categorie:

CAT (1): (solo) segnali vocali;

CAT (2): vel. 4 Mbps;

CAT (3): vel. 10 Mbps;

CAT (4): vel. 16 Mbps;

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

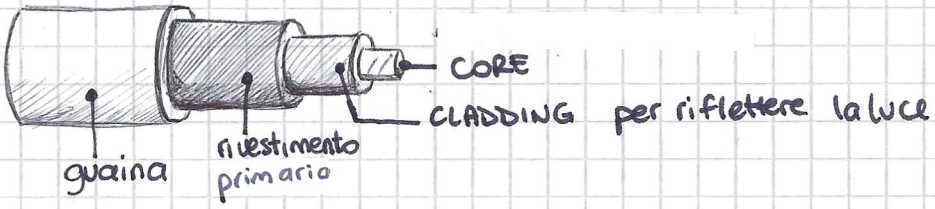
CAT (5e): vel fino a 1 Gbps;

CAT (6): 250 MHz con bp 200 MHz (non l'ho troppo capito);

CAT (7): *work in progress*

(NORME di INSTALLAZIONE) Eg: non tirare i cavi con forza > 11,3 kg
↓
sennò le binature si allentano - o come direbbero i miei colleghi de Jesi, "se sbregano"!

la Fibbra!



- è immune alle INTERFERENZE!
- può essere - monomodale
  \ multimodale → più frequenze
- tipo di connettore = ST (o più spesso SC) / ora vedo per cosa stanno:
  ST = STraight Tip
  SC = Standard Connector

Estensione di una LAN

- ... perché?

  1. per unificare LAN costruite in momenti diversi;
  2. unificare LAN in diversi edifici;
  3. suddividere CARICHI ELEVATI;
  4. aumentare distanza copribile;
  5. aumentare affidabilità;
  6. aumentare sicurezza;

  ci sono naturalmente LIMITAZIONI dovute a protocolli di accesso e al decadimento del segnale.

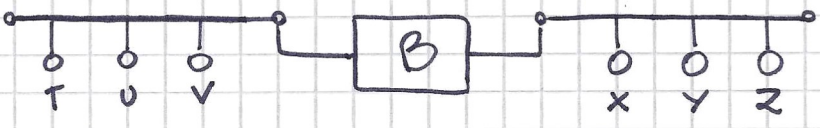
- ...come?

  - REPEATER - livello 1 (fisico)

    sono come buffer di segnale

  - BRIDGE - livello 2 - legge intestazione frame e ci fa quello che deve
    NON propaga collisioni

    adattativo: si configura a mano a mano che riceve info sulla rete

    possono anche convertire protocolli (Ethernet → [B] → TRing)

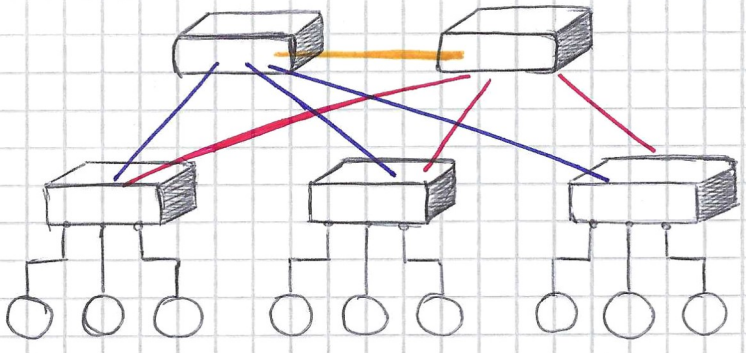
    / TRANSPARENT BRIDGE → Se x deve comunicare con z, la prima
    \ SOURCENODE BRIDGE volta la frame viene inoltrata lungo TUTTA la rete, ma già al "ritorno" il bridge avrà acquisito info che x e z sono nello stesso segmento e eviterà di inoltrare il frame a sinistra.
    = backward learning

CICLO DI BRIDGE dovuto a flooding
frame rimangono in circolo a lungo senza raggiungere destinazione
→ SOLUZIONE: DISTRIBUTED SPANNING TREE protocol STP
(per far comunicare i bridge)

fault tolerance:

doppi concentratori in modo che se smette di funzionare uno c'è l'altro a garantire il servizio (Ridondanza, Cloud computing 101)

CENTRO STELLA RIDONDATO



RETI COMMUTATE (SWITCH)

= un enorme bridge

Lavorano in 3 modalità:

I) CUT THROUGH
II) STORE & FORWARD
III) FRAGMENT FREE

I) switch legge il MAC destinatario inoltra e fine;
II) legge il MAC, controlla CRC e fwd;
III) una via di mezzo, legge i primi 64 byte e se non ci sono anomalie inoltra;

E.g.:

switch stackabili (48 + 48 = 96 porte)

come si gestiscono - GUI
- cmdline

VLAN (virtual LAN) divide lo switch in LAN diverse

Indice più o meno accurato dei contenuti