

18/11/21

## ARP (Address Resolution Protocol)

Ogni dispositivo su una rete ha associato 2 indirizzi:

→ IPv4

→ MAC

- Il protocollo ARP risolve dunque il problema di traduzione di un indirizzo IPv4 in indirizzo MAC (NEIGHBOR GREETING)

Supponendo di avere due macchine che vogliono comunicare tra loro occorre:

tramite pacchetto da inviare l'indirizzo IPv4 del destinatario.

• Prova a chiedere a livello 2 di inviare il pacchetto, ma non può perché per inviare il pacchetto ho necessità di avere un indirizzo mac.

• Per ottenere il mac del destinatario devi ricorrere al protocollo ARP che invia un pacchetto broadcast per richiedere a chi è associato quell'IPv4. => ARP REQUEST.

• La macchina che riconosce l'IPv4 risponde con ARP reply inviando il suo mac.

=> Posso inviare pacchetti tramite liv. 2

Se devo inviare ad un default router tramite il pacchetto ARP request richiedo il codice del router.

### ARP e uso cache

I dispositivi che usano ARP usano di solito anche una cache dove:

- !! • Vengono salvate le informazioni riguardanti le corrispondenze tra IP4 e MAC.  
poiché di solito per comunicare tra macchine diverse non invia un solo pacchetto ma molti quindi conviene mantenere le corrispondenze per non 'intossicare' la rete con troppe ARP.
- Solo se non ho corrispondenza in cache invia un pacchetto broadcast.
- Ogni pacchetto broadcast ARP causa un'interruzione circa 1 ms su tutti i computer delle lan.

Gli ARP si usano solo in una rete locale infatti i router non le fanno passare.

## RARP (REVERSE ARP)

Serve per sapere il mac di una macchina  
soprendo l'IPv4

→ Usato in macchine prive di HDD

Il pacchetto ARP si trova all'interno  
di un pacchetto ethernet 2.0 perché  
ha necessità di comunicare a tutte le  
macchine usando il livello 2. Non fa  
uso di LLC

- I campi hardware e protocol specificano  
il tipo di macinato al livello 2 e 3  
interessati
- I campi Hlen e Plen specificano la lunghezza  
dei 2 macinato
- Gli campi operation indicano se è una  
richiesta o una risposta.
- I campi Sender IA e Sender HA indicano  
l'indirizzo di livello 3 e di livello 2 del  
mittente
- I campi Target IA contiene l'indirizzo di  
livello 3 della macchina di cui voglio  
conoscere l'indirizzo di liv 2.

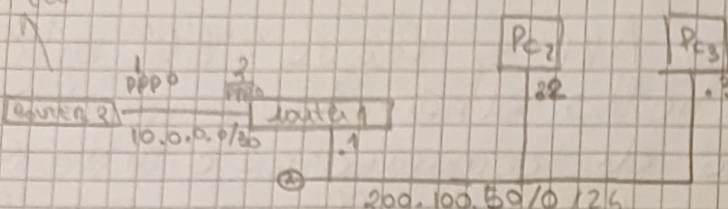
Ridondante in ARP e RARP → MAC

22/11/21

## Esercizi Subnetting

Ho uno lan con indirizzo 200.100.50.0 /24  
ossia con netmask 255.255.255.0

Velocità



1) Quante macchine possono essere collegate alla lan A oltre al router?

→ in una biretta /24 ho 256 macchine ma non posso usare lo 0 né l'uno (tutti zero/tutti 1) rimuovo il punto (.1) del router quindi posso collegare oltre 253 macchine

2) Indirizzo broadcast di lan A?

→ devo riempire la parte dell'indirizzo non occupata dal prefisso con tutti 1 prefisso è /24 quindi l'ultimo byte deve avere tutti 1 per essere broadcast

3) TABELLA INSTRADAMENTO Router 1

SUBNET	NETMASK	INT	NEXT HOP
200.100.50.0	255.255.255.0	ETH0	DIRECT, CONNECT
10.0.0.0	255.255.255.252	PPPO	D.C.
0.0.0.0	0.0.0.0	PPPO	10.0.0.1

→ tutte le destinazioni di internet fuori lan

per missione traffico  
a 0.0.0.0 devo  
aggiungere e per  
farlo devo ponere  
per 10.0.0.1

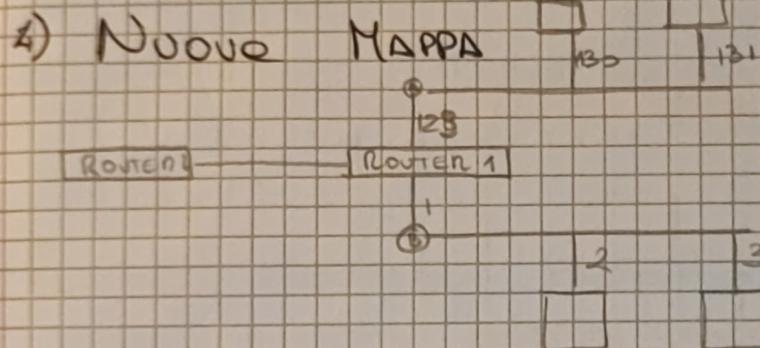
NETMASK /30:

255 255 255 252

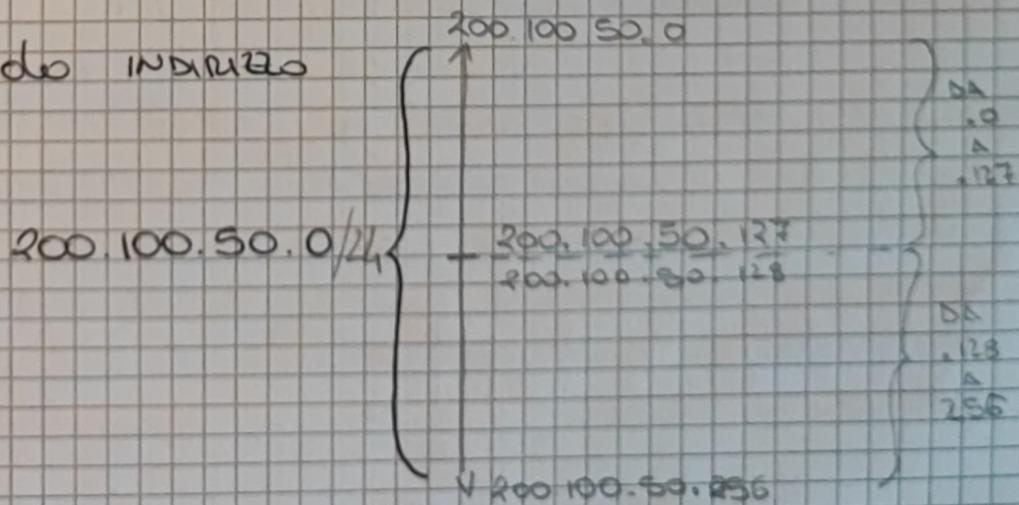
111111. 111111. 111111. 111111 00  
255. 3 - 252

Ho solo 2 macchine che posso connettere alla rete  
La 10.0.0.0 È una rete locali alla quale  
non può accedere da internet  
Non si può accedere all'interfaccia ppp0 di  
router 1

Indirizzo 10.0.0.3 è broadcast per 10.0.0.0



Suddivisione indirizzi



200.100.50.0/25 ptk uso 1 bit dell'ultimo byte  
200.100.50.128/25

le netmask sono:

255.255.255.128

200.100.50.0 /24

200.100.50.0 /25

byte sono 0  
byte sono 0

200.100.50.128 /25

byte 0 è 1  
=> 128

### TABELLA INSTRADAMENTO

SUBNET	NETMASK	INT	N. HI
200.100.50.0	255		
200.100.50.128			
10.0.0.0			
0.0.0.0			

3 scenario

200.100.50.0 /24

1.127  
- 128

1.128

1.131  
1.132

256  
256

256

1128

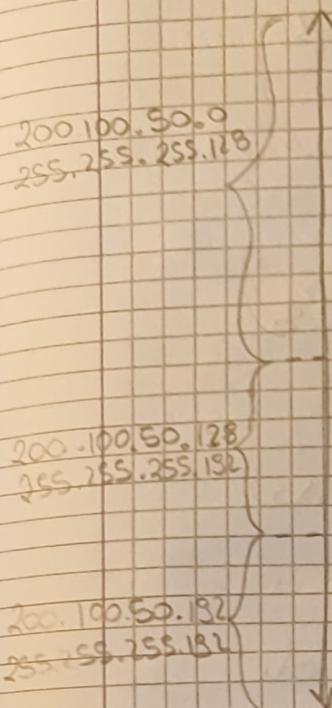
200.100.50.128 /26

1.132  
1.132

umento  
perdita  
nella parte  
posta fogliata  
1.256

subnet 255.255.255.192

• pk vado ad occupare i primi 2 bit del  
4° byte quindi ho  $128(2^7) + 64(2^6) = 192$



Se devo dividere un sottosistema specifico di indirizzi allora devo sacrificare un bit del 4° byte ma solo del sottosistema diviso. Per l'altro sottosistema di indirizzi, rimane com prefisso invariato.

200.100.50.0/25 per gruppo in alto di indirizzi,

200.100.50.128/26 per gruppo in basso di indirizzi

Per gli indirizzi broadcast di questi sottosistemi basta usare l'ultimo indirizzo disponibile per quel sottosistema (infatti avrà tutti i bit del 4° byte)  
↳ posti disponibili

Nel subnetting si può togliere e togliere (se voglio mantenere uguali le 2 subnet gli indirizzi possibili) pk usando un bit in più per il prefisso questo può avere solo 2 valori [0,1] che permettono di creare solo 2 sottogruppi.

E	<u>200.100.50.22/27</u>
	.255
R.S.	.194
D	<u>200.100.50.192/27</u>
	.193
R.H.	.130

Nella tabella di mstradamento di R.S ho 3 righe:

- 1) che mi permette di connettere tra loro i PC sulle lan E (direttamente connesso)
- 2) 0.0.0.0 per gli altri fuori lan
- 3) che mi permette di connettere i PC di E con la lan D che è direttamente connesso. (qui non posso evitare di mettere queste righe pk aumenti dovrei muovere il pacchetto per l'ID del router 4 (infatti questo indirizzo farebbe matching solo con 0.0.0.0 in R.S) per poi essere inviato me questo significa mantenere in rete un pacchetto per + tempo del necessario pk RS potrebbe già inviarlo al destinatario effettivo essendo l'ID collegato direttamente)

Se applico le tabelle di mstradamento corrette ossia quelle in cui è presente anche indirizziamento diretto nelle lan d' ho che, appena inviato il pacchetto, il router conoscendo l' destinatario <sup>(IP)</sup> (presente nelle lan D che è direttamente connesso) manda una richiesta

## REQUEST BROADCAST

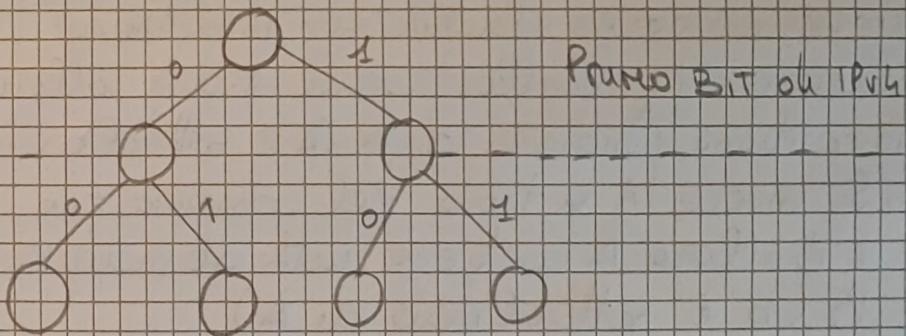
ARP sulla sotto-rete  $\rightarrow$  per sapere a quale indirizzo mac deve inviare il pacchetto.

Anche fra 2 router si ha un'ARP REQUEST specificando l'IP di cui si vuole sapere l'indirizzo MAC. (questo pk i router non sono direttamente connessi)  $\Rightarrow$  Ho next hop.

25/11/21

## INDIRIZZI SUGLI ALBERI (IPv4)

- Lo spazio degli indirizzi IPv4 può essere rappresentato con un albero binario etichettato con 0, 1
- Verso il figlio sx ho 0 ossia l'IPv4 con bit significativo a 0, mentre a dx ho 1 ossia con bit significativo pari a 1



Su ogni foglia ho un indirizzo che posso ricostruire interamente superponendo il cammino dalle foglie alla radice ormai a profondità 3?

1) A cosa corrisponde una net in questo albero?

Ad un sottoalbero radicato ad X

2) È il suo prefisso?  $(x)$

Al cammino delle foglie alla radice, completato con 0 nelle parti host (come prendo il cammino più a sx del sottoalbero scelto).

Per i broadcast invece compatti  
l'indirizzo usando il solo commissario  
e dx così quello composto da 1

3) A cosa corrisponde nella rappresentazione  
ad albero il prefisso /8 per una net?

Tutti i nodi non foglie a profondità 8

Su ogni livello ho la buona corrispondente  
a quella profondità  $\Rightarrow$  profondità 20  $\Rightarrow$  /20

4) Dati due net A e B come si  
verifica che A e B non abbiano indirizzi comuni?

$\rightarrow$  Ho A e B che sono due nodi che rappresentano  
due prefissi.

• Se voglio verificare che siano nello  
stesso comminno perché se B si trova  
nel comminno sotto alla radice di A allora  
B contiene tutti gli indirizzi di A (A è sottoalbero di B)

$\rightarrow$  Se invece A si trova nel comminno tra radice  
e B allora A contiene tutti gli indirizzi di  
B (B è sottoalbero di A)

$\rightarrow$  Negli altri casi non si sovrappongono

5) In rappresentazione ad albero come capisco se due reti sono risultati del subnetting di una rete?

Verifico che le 2 reti siano figli dello stesso genitore

6) Nella rappresentazione ad albero a cosa corrisponde le tabelle di mappamento?

Corrisponde alla lista di nodi dell'albero  
ogni nodo è associato ad un interfaccia  
o ad un next hop.

• O/o è la radice dell'albero nella sua  
totalità, significa che ho sicuramente più  
molti altri o/o.

→ Le righe di una tabella di mappamento  
sono generate in modo di organizzare gli  
indirizzi da quello con prefisso + esteso a  
quello meno esteso [o/o]

## IPv6

- Ha uno spazio di mappamento più ampio  
di IPv4 (ogni continente ha una /18)

- Diverso software per gestire IPv6 rispetto IPv4

- Sono entrambi prot. di livello 3 => sono alternativi

- Pacchetto di 40 byte senza extension header

- Version 4 byte, Traffic class 8 bit

# Protocollo ICMP

COMANDI PING e TRACEROUTE

Per IPv6 ho ICMPv6

Uso approccio Best effort come tutto up,  
ossia non garantisce la consegna dei  
pacchetti alcuni di fatto potrebbero essere  
scartati "dropped on the floor" e saranno  
ritrasmessi da livelli superiori

-> Scarto se:

TTL = 0

Congestione router.

- IP chiede a ICMP ossia internet  
control message protocol, che offre  
un servizio di Error Reporting

## Regole per ICMP

- 1) non genera messaggio di errore su un  
errore generato da un ~~per~~ messaggio di  
errore
- 2) Un messaggio frammentato ha ICMP solo  
sul primo pacchetto
- 3) Multicast non genera ICMP

## Messaggi destination Unreachable

- network unreachable: gateway vede la rete destinazione a distanza infinita.
- Non fa matching con nessuna regola delle tavole di mappamento (nemmeno ha 0/0)
- Host unreachable: l'host destinazione non risponde o chiama ARP
- Protocol unreachable: l'host destinazione non riconosce il protocollo nel pacchetto
  - se non ho il protocollo nell'host destinazione (esempio)
- Port unreachable: la porta definita non è raggiungibile
- Fragmentation needed and DF set: pacchetto non può essere frammentato
- TIME-EXCEEDED: TTL = 0 pacchetto buttato

## Messaggi redirection

Se un router vede che il prossimo router a cui dovrebbe mandare il pacchetto è sulla stessa rete del mittente.

Dico a mittente di muoversi direttamente

## Messaggio Echo

Echo Request / Reply: Controllo di raggiungibilità host

Timestamp e Reply: Come echo ma misurano in più l'ora un cui sto rispondendo.

Comando Ping (attiva e raggiungibile)

Fork: 2 protocolli separati

Invio Echo request e mi aspetto un reply e una volta conosciuti entrambi faccio la differenza e trovo il roundtrip delay.

29/01/21

## TRACE ROUTE

Voglio sapere quali router attraverso per arrivare a destinazione

• Spedisco intenzionalmente un pacchetto con TTL=1 così che arriverà a router 1 il pacchetto, decrementato il TTL, viene scartato e router 1 invia al mittente un pacchetto time exceeded (contenente IP router 1). Così so chi è Router 1

segue quanto sopra

• Aumento sempre TTL di 1 fin tanto che il pacchetto arriva a destinazione

Se invio per fare trascorrere un pacchetto Echo request quando questo arriva il destinatario risponde con Echo reply. Se invio pacchetto UDP (e devo inviare su uno certo porto in cui non ho ricevuto in ascolto altri) la macchina di cui risponde con Port unreachable.

Pacchetto UDP : Risponde pacchetto port Unreachable pk il processo non esiste sulla macchina

Il router tende o non considerare le linee Options del pacchetto IPv4 perché richiederebbe dispense eccessive di risorse

## Lo STRATO DI TRASPORTO

- Servizi di connessione tra end system per le applicazioni:  
non è presente in sistemi intermedi  
→ Servizio normalmente connesso ed affidabile
- Le primitive offerte ad altri livelli si assumono essere affidabili e devono essere facili da usare così che i programmati possano facilmente integrare

### PRIMITIVE DI TRASPORTO

• Listen

• Connect

• Send

• Receive

• Disconnect

Spesso la pila iso OSI sapeva le livelli  
e così che, se necessario, soltanto alcuni  
livelli, possono direttamente interagire con  
il livello.

## Instaurazione e Rilascio Connessione

Le primitive per l'instaurazione e per il  
rilascio di connessioni devono essere  
realizzate in modo ~~sempre~~ affidabili  
(più facilmente realizzabili per instaurazione)

### Instaurazione

- I pacchetti scambiati sono numerati così  
che ricezione possa riscontrarli
- L'instaurazione si basa sullo scelta e sul  
relativo riscontro dei n. minimi di sequenze  
usati per la numerazione dei pacchetti
  - metodo three-way handshake
- C<sub>1</sub> sceglie il numero minimo di seq. X per i suoi  
pacchetti
- C<sub>2</sub> invia a C<sub>1</sub> una connection request (C<sub>R</sub>) con X
- Arriva la richiesta a C<sub>2</sub>
- C<sub>2</sub> sceglie il proprio minimo di pacchetto Y
- C<sub>2</sub> invia notifica di accettazione di connessione  
riscontrando X e proponendo Y
- C<sub>1</sub> riceve Connection accepted da C<sub>2</sub> e  
riscontra Y

- Un numero diverso di 0 così da poter sempre uscire e che connessione mista muore in un certo pacchetto (infatti potrei avere ottime, sulle stesse 2 macchine, più connessioni)

### Rilascio Connessione

- Se il protocollo di rilascio fosse più rudimentale rischierei di perdere pacchetti

④ **Rilascio simmetrico:** Siano  $C_1$  e  $C_2$  due macchine che stanno parlando, se  $C_1$  rilascia la connessione non è detto che la percezione del rilascio sia immediata da parte di  $C_2$ , infatti alunque  $C_1$  continua a ricevere la trasmissione di  $C_2$

Per quanto tempo devo continuare ad ascoltare dopo il rilascio della connessione? Consapevole che l'ascolto completo comunque un dispendioso di risorse PROBLEMA Esercizi

uso tecniche simili al three way handshake con timeout

### TCP

servizio di dati bidirezionali contemporanei (full duplex) punto-punto  
multicasting e broadcasting non supportati

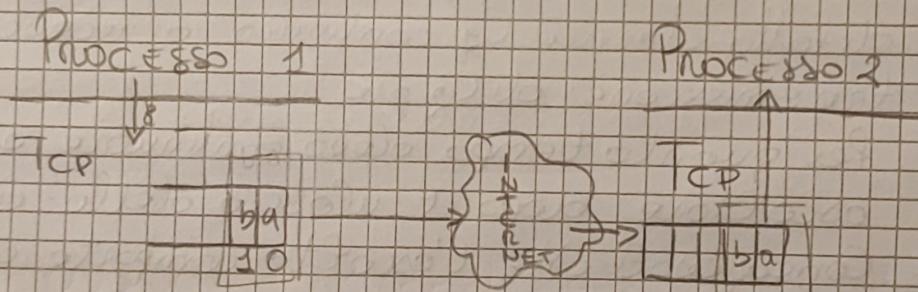
• GARANTISCE L'AFFIDABILITÀ DELLA TRASMISSIONE  
a differenza del livello 2/3

## Transmission Control Protocol

TCP individua i problemi di trasmissione li corregge e provvede a unirli.

• Servizio Connesso. La connessione avviene fra 2 processi, identificati da IP e un numero di Port in entrambe le macchine (processi).

- Stabilite la connessione il processo mittente posso chiedere al servizio TCP che gli invie al servizio TCP accettante correttamente che lo posso a processo destinatario.
- Processi vedono il canale come punto



## Port

TCP specifica come distinguere più destinazioni sulle stesse macchine.

→ Ai processi che interagiscono con altri processi TCP assegna un numero di Port.

• Port sono dunque TCP SAP

Port ha 2 Byte

↳ alcuni sono riservati (256)

MAC  
Layer  
IP  
Protocol

Port 1-100

SSH

- TCP non numera i pacchetti ma byte, ogni byte ha un n. di sequenza a 32 bit  
 $\hookrightarrow$  32 bit permettono di avere reset dei numeri di seq poco frequenti
- Tali numeri di seq. sono dati anche agli ACK

Le TPDU TCP si chiama segment - header + dati.  
 La max. coprente pacchetto TCP è dato da:

- ① limite di pacchetto IP - 65535 byte
- ② Mtu
- ③ limite segment

TCP stabilisce come inviare i dati, forme cioè i pacchetti da trasmettere

Il processo non sa quanti byte invia TCP in un pacchetto.

Il receiver di processo li prende i pacchetti quando la coda in arrivo è piena

Nel pacchetto ricevuto ho solo il numero del primo byte inviato e il numero totale di byte nel pacchetto

TCP riscontra i byte e il prossimo byte aspettato

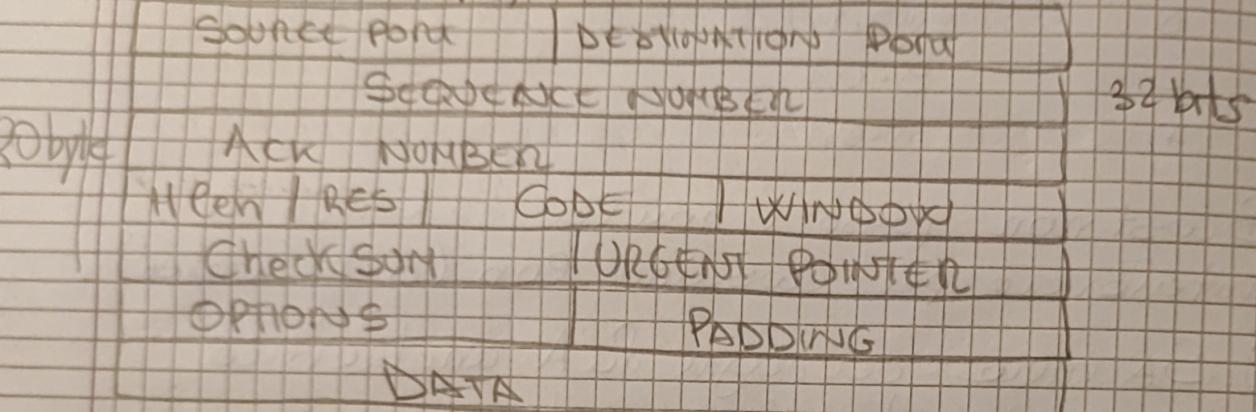
## TCP (2)

SEGMENT usato per:

- stabilire connessioni
- trasferire dati
- mandare ack
- chiudere connessioni

FORMATO SEGMENT

4 byte



• CONTENUTO IN PACCHETTO IP

- Arrivo tramite IP alla macchina destinatario e tramite Port individua a che processo deve andare il pacchetto
- Ethernet supporta 1500 byte se sottraggono 20 byte di intestazione IP senza opzioni e 20 byte di intestazione TCP senza opzioni  $1500 - 20 - 20 = \underline{1460 \text{ byte}}$  a disp. x applicazioni

Sequence number: numero byte e stabilisce la posizione del pacchetto dati nel flusso dei dati generati dal mittente

Ack NUMBER: riscontra, byte ricevuti correttamente e indica il prossimo byte atteso

Len: numero di parole di 32 bits (longword) nell'intestazione

Coode (6 bits): indica il tipo di messaggio contenuto nel segmento

bit	SIGNIFICATO	USO IN THREE WAY HANDSHAKE (apertura connessione)
SYN	sincronize seq. number	
FIN	sender has reached end of byte stream	(chiusura conn.)
ACK	ack field is valid	
URG	urgent point field is valid => important	
PUSH	push request	
RST	reset connection	

Windows specifica l'ampiezza corrente delle finestre di controllo di flusso

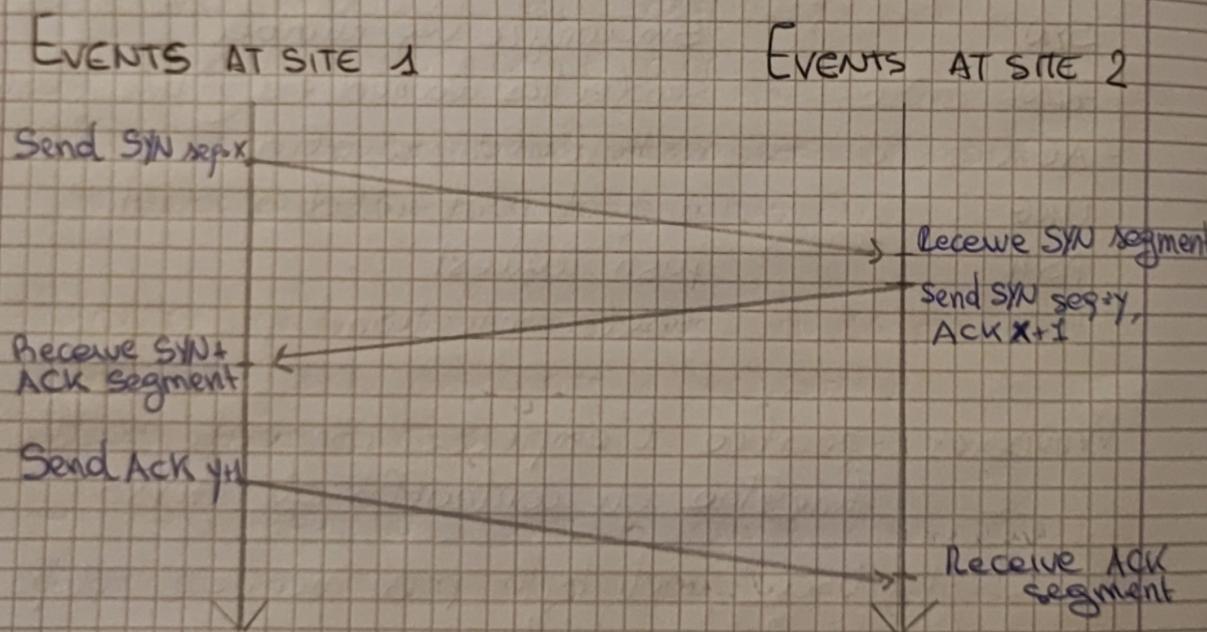
- ① stabilisce quanti byte posso ricevere dopo l'ultimo riscontro.
- ② dice quanti byte possono essere muovi dall'intelaiatura in relazione allo spazio rimanente sulla coda (strumenti perde dato)
- ③ può valere 0.

Checksum: verifica l'intero segmento  
e gli indirizzi IP del pacchetto IP

OPTION: le più usate specifica la massima  
ampiezza del campo dati

- Negozia l'ampiezza campo dati in  
fase di connessione
- almeno 536 byte di dati occettano

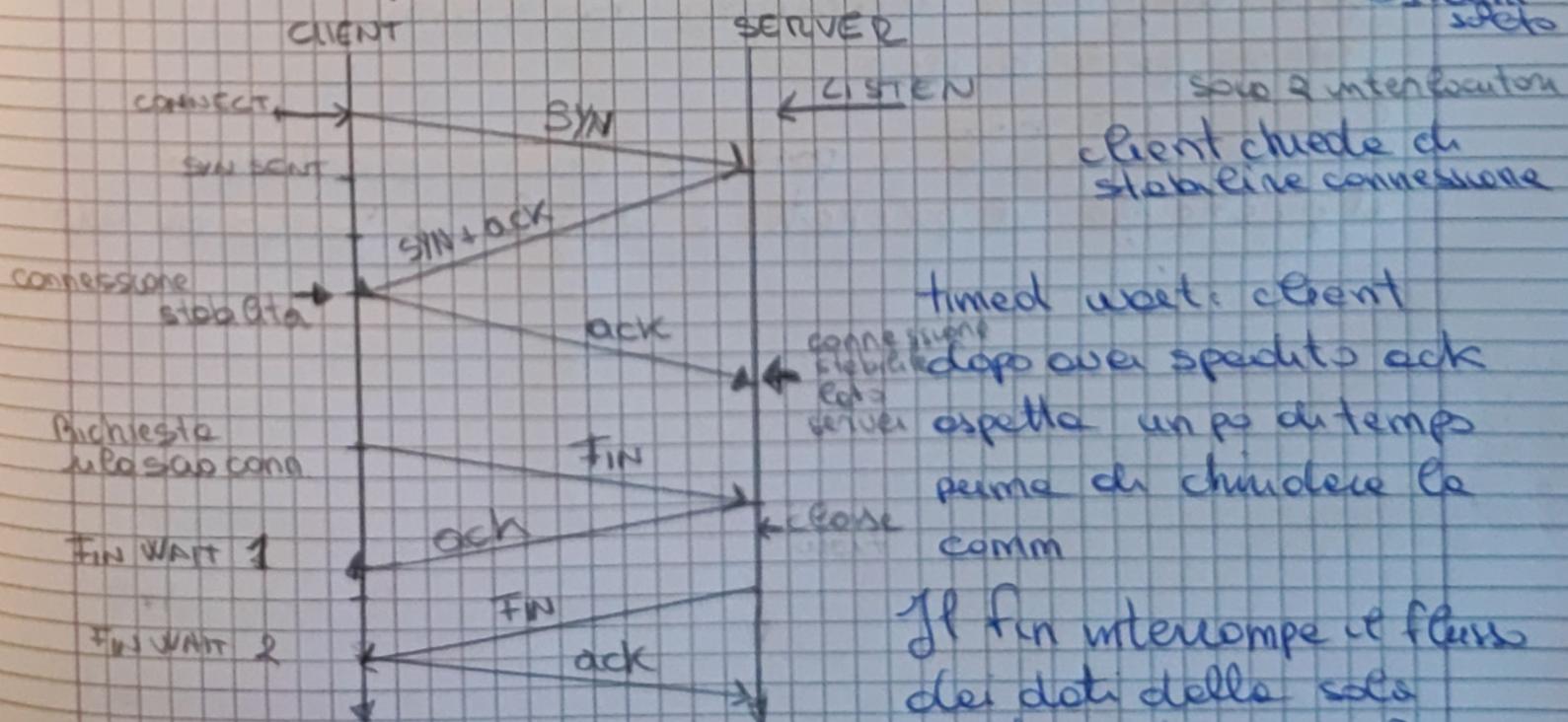
Macchina 1 manda pacchetto con bit S/N=1  
e indica il numero di sequenza



il primo byte spedito avrà numero  $x+1$   
ossia byte ottenuto dal site 2

## RILASCIO DI CONNESSIONE

- Per terminare la connessione una macchina (quella che intende interrompere la connessione per prima) invia un segment con FIN=1  
→ non ha più dati da trasmettere.
- Quando la macchina che ha inviato FIN=1 riceve in riscontro un ACK considera chiuso il suo flusso
- I dati continuano a fluire in direzione opposta
- La connessione si considera chiusa se entrambi i flussi sono chiusi
- Il suo normale richiede 4 segmenti
- Il primo ACK( $x+1$ ) e il secondo FIN possono trovarsi nello stesso segment
- Se ACK spedito per ultimo applichi un timeout pari a 2 \* tempo vita pacchetto (2-256 sec)



06/12/21

## Stato ESTABISH.

In RICEZIONE:

- Gestione suscita
- Controllo flussi

In TRASMISSIONE:

- Come spedire pacchetti e gestione congestione
- Specificazione copie

## UDP - User Data protocol

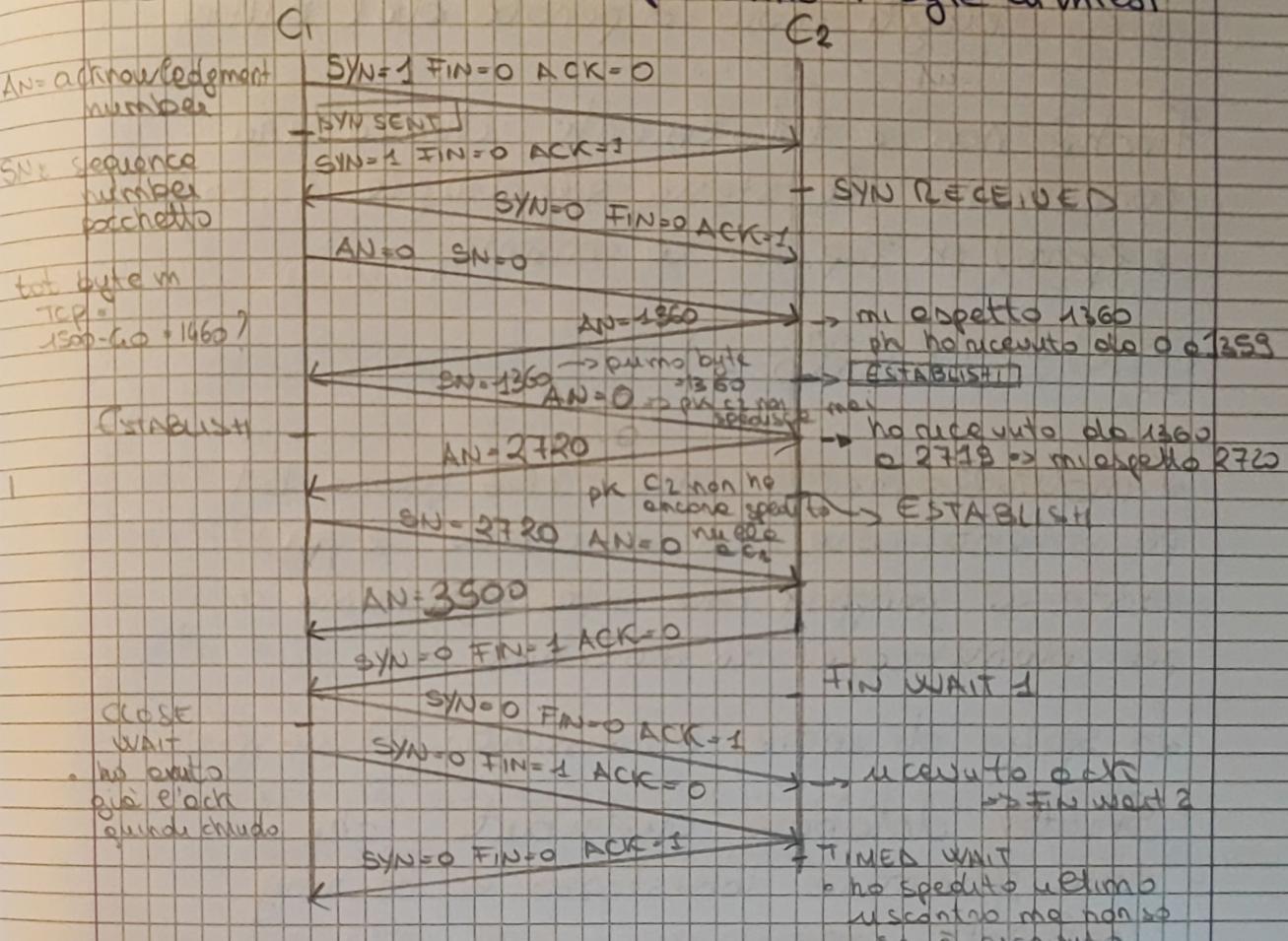
- fornisce un servizio di trasmissione non affidabile ossia non ha ack
- Si usa quando non mi interessano gli ack
  - Riscontro ack a livello applicativo
- Header di 8 campi composti da 16 bit (port, lunghezza datagram, checksum)
- Stessa port di TCP

Se non specifico i port a cui è destinato il pacchetto il pacchetto è multihop  $\Rightarrow$  Senso di Volp

# Esercizio TCP

1) applicazione  $a_1$  su calcolatore  $C_1$   
 manda 3.500 correttu (byte) ad applicazione  
 $a_2$  su  $C_2$ . MTU =  $1400 - 20 - 1360$  byte

Intestazione IPV4 e TCP hanno 20 byte di intell.



Nei primi pacchetti ho che TCP manda ,tremante opzione, un campo m cui mette il mtu che il calcolatore  $C_1/C_2$  mette a supportare in particolare  $1400 - 40 - 1360$  byte (escludendo le intestazioni dei pacchetti)

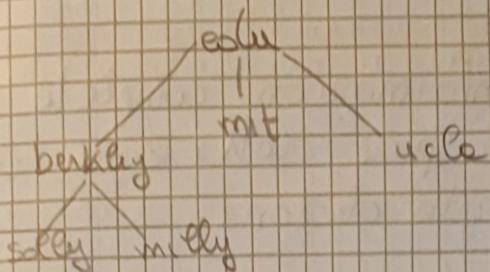
- Sequence number: numero assegnato al primo byte (parte sempre da 0)
- Acknowledgment: sempre inizia da 0
- AN di  $C_1$  è sempre 0 pk  $C_1$  non aspetta nessun byte da  $C_2$  (poiché non comunica nulla) quindi non ha nulla da riscontrare

- TCP: non sempre sceglie di ricevere un pacchetto e aspettare il riscontro ma potrebbe invece spedire in successione tutti i dati e aspettandosi tutti i riscontri dopo  $\Rightarrow$  La gestione delle mail dipende dal buffer
- Finché non è totalmente chiusa una connessione le stesse quadrupe (2, IP ~~host~~ e 2 port) per TCP non lo consente.

## • DNS (livello 7)

- L'indirizzamento IPv4 e IPv6 usano bit per identificare la macchina e poi esiste un mapping che usa un nome per identificare la macchina.  
Lo spazio di nomi ammissibili: Nome spazio
- Sfrutta database distribuiti che devono garantire robustezza e caching realizzate con:
  - replicazione
  - caching

- Usa nome space correttamente da varie strutture separate da un punto.
- tutto ciò perché vari nodi con lo stesso prefisso appartengono allo stesso albero.
- Lo gerarchico non si definisce allo livello fisico di una machine.



L'autorità top level crea sottounità berkeley.edu, mit.edu, ucla.edu e delega a loro l'assegnazione dei nomi delle machine delle sue sottounità. Tipi sally.berkeley.edu è uno sottodominio di berkeley.

**Domino:** è un sottoalbero del namespace e il suo nome è il nome della radice: berkeley.edu

• Ogni singolo host è un domino

• Toglie sono host e nodi intermedi possono rappresentare host

Tutti gli alberi specifici sono tutti figli dell'albero con nodo radice - strunge nulla

Struttura namespace è autonoma a IP

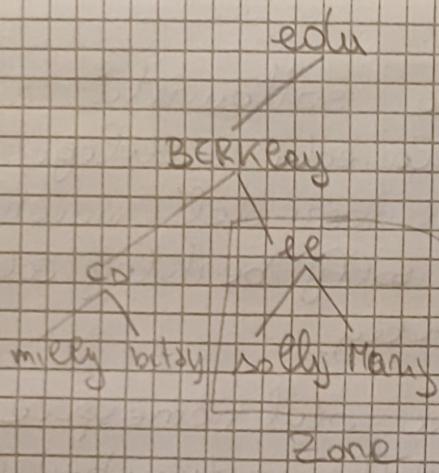
DNS: Domain name system

Oggi le macchine che realizzano mapping  
si chiamano nome server (ns)

Possono avere ologrhe per usare uno spazio  
di dominio.

## Zona

Un nome-server ha informazioni complete su  
una parte del namespace detto zona; un  
nome server è autorizzato di zone



Zone ≠ dominio

09/12/21

## NAME SERVER: PRIMARY / SECONDARY

Per ogni zone ho un solo purogno e diversi secondary interpellati quando purogno non è disponibile

### MASTER E SLAVE

È master quel server che mantiene le sue tabella di nome sempre aggiornato e gli slave si auto-aggiornano recuperando le tabella da master

- D. solito purogno è master e gli altri sono secondary.

Il nome server è un purogno per una zone e può essere secondario per un altro dominio

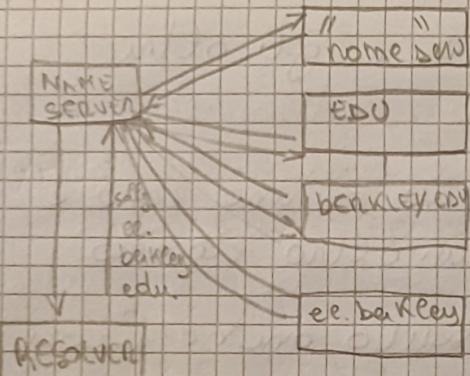
Il purogno può essere inserito anche fuori dal dominio

### RESOLVER

- I client che usano name-server si chiamano resolver
- Nei vari host ho un resolver (1 per macchina) che sono eseguite query su name-server.
- Non è obbligo che sia un processo autonomo. Oppure in browser esistono dei codici che svolgono funzioni di resolver (associa nome a ip)
- Resolver singolo (per la rete opp) non condivise dati di associmenti trovati.

## RISOLUZIONE

- Se un resolver chiede al nome server informazione se questo server ha le info la conclude direttamente
- Se non lo ha si rivolge al <sup>nome-</sup>server autorità per la radice dello spazio dei nomi
- Il server autorità fornisce al resolver un nome server di un server + specifico
- I server di più basso livello hanno svolte query



- il mio nome-server non conoscendo cosa sia l'indirizzo richiesto mi indirizzi ad un altro server radice x quello spazio
- Mi riferisco prima alla radice di tutte gli indirizzi lo stringo tutto,
- Allora non conosce ee.edu... me lui conosce il server che gestisce il nome edu e posso questo url
- edu conosce berkeley.edu e poi ~~verso~~ mondo url il mio nome server

## Approcci di risoluzione

RICORSIVA: il client chiede a quale indirizzo corrisponde il nome n e pretende come risultato l'indirizzo; se il server non possiede l'indirizzo dirà di contattare altri server.

ITERATIVA: Il nome-server a quale indirizzo corrisponde il nome n e, in caso non lo abbia, si accontenta di un indicazione di un altro server a cui rivolgersi.

Resolver si rivolge al suo name-server con query ricorsive, tra loro i server usano query iterative.

## CACHE

• Durante una query un nome server apprenderà informazioni relative ai server vicini per le zone, così che, quando avrà una query simile allora, eviterà di chiederla a tutti i server le zone destinate, solo direttamente al server che mi serve.

• Ogni informazione ha una durata temporale limitata per la <sup>sua</sup> validità (infatti può cambiare parere o sistema).

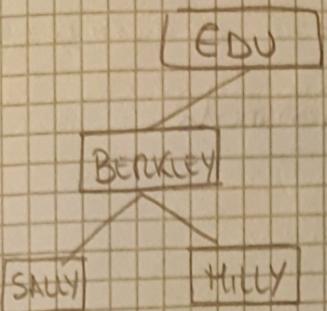
## RESOURCE RECORD

- Le informazioni DNS sono memorizzate in record (anche su singolo host) ha associato un resource record
- Contengono indirizzo IP

$\langle \text{RESOURCE RECORD} \rangle \longrightarrow \langle \text{DOMAIN NAME} \rangle \langle \text{TIME TO LIVE} \rangle$   
 $\qquad\qquad\qquad \text{(IN)}$   
 $\qquad\qquad\qquad \langle \text{CLASS} \rangle \langle \text{TYPE} \rangle \langle \text{VALUE} \rangle$   
dipende dal tipo

### TYPE:

- SOA: start of authority; info amministrativa
- • A: indirizzo di un host
- • MX: specifica il nome dell'host che accetta le mail indirizzate al dominio record
- • NS: nome server x una zona
- • AAAA: indirizzo IPv6 di un host
- • CNAME: conomico nome; meccanismo per creare alias



il nameserver berkeley.edu è il server che gestisce la zona berkeley.edu ed ha almeno 2 record:

- SALLY.berkeley.edu      IN A 70.70.70.71
- IN AAAA mohrizzo.ipv6

• il nameserver .edu è il server autorità di edu e ha i seguenti record:

- berkeley.edu      NS nameserver.berkeley.edu  
↳ riferimento di n.s che gestisce lo spazio berkeley.edu che verrà dato al n.s che fa richiesta di berkeley
- nameserver.berkeley.edu      IN A 50.50.50.50  
↳ record di mcollaggio, lo metto in una seconda riga così la macchina di cui voglio sapere l'ip possa combinare ip me non il nome infatti lo pongo in una riga stabili e la seconda riga.
- Il nameserver od ogni query restituisce l'indirizzo IP. infatti so che edu conosce il server a cui far riferimento e nella riga sottostante ho ip che mi dice come avviene

## FORMATO MESSAGGI

- Tipo messaggi con questi formati:
- richieste
- risposte

Tan i campi dell'header

QRI: è domonote (+) o risposta

RD: è recursiva (+) => recursion closed

Tra i campi delle question section

NAME: nome usato richiesto

TYPE: tipo ricerca record richiesto

Sono pacchetti molto piccoli:

di solito viaggiano con UDP

-> Port usato 53, #53

-> Se i server non rispondono ripropongo le domande

Se mi risvevo uno => risposta ampia offrone  
dove usare Tcp pk non mi basta un singolo  
pacchetto UDP

Vedi nell'ultimo pt lezione

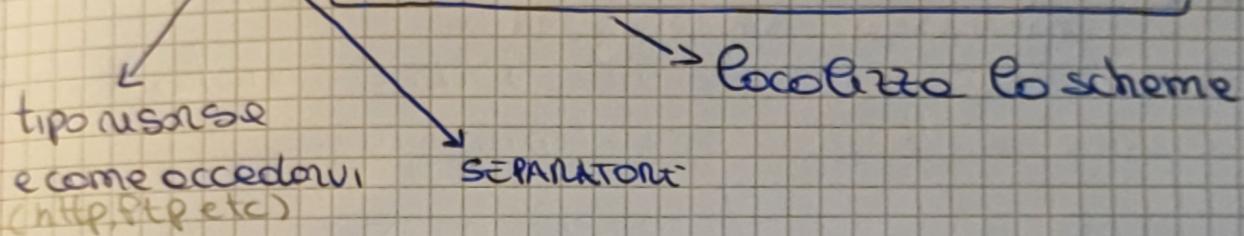
nslookup

## URL e HTTP

→ Uniform Resource Locator (URL) o Uniform

Resource Identifier (URI): Rappresentazione testuale e compatta di una risorsa disponibile in Internet (doppio ruolo di identificazione e location)

scheme: ponte dipendente - doppio-scheme



Ponte dipendente:

alcuni schemi adottano la medesima sintassi  
che per le porte dipendente doppio scheme.

Questo sintassi è denominata **CISS**  
Common Internet Scheme Syntax

//userid:password@nd.host:porta /path Rivedi  
queste porte

→ significa  
che uso  
CISS

## Scheme http

http://userid:password@md-host:porta/path

Il browser estrae md-host e porta e le usa per aprire connessione TCP-IP usando la porta (port) con la macchina specificata (md-host) se port non definita si usa porta 80

userid:password si usa per gestire user name e password

## HTTP

- basato su un semplice schema richiesta - risposta nel quale la domanda è inviata dal client e la risposta dal server

• Un colloquio server-client è una sessione

HTTP 1.0

apertura

richieste

risposte

chiusura

23/12/21

## La Posta Elettronica

- analisi requisiti:

- primitive servizi: definisco l'interfaccia di servizio
- architettura e operazione: definisco le opz. coinvolte nelle gestione del servizio
- definizione protocolli

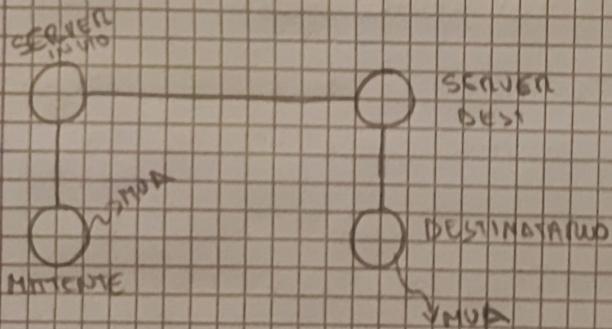
### ANALISI REQUISITI

- gestione messaggi
- spedizione messaggi
- PRIVACY

### PRIMITIVE DI SERVIZI

- GESTIONE DEI MESSAGGI
  - COMPOSIZIONE MSG
  - NOMENCLATURA
  - CANCELLAZIONE
  - STAMPA
  - CARICA/SCARICA MSG

### ARCHITETTURA



• Foto oleodinamico  
host + server configurati con + secondari e un principale

## APPLICAZIONI COINVOLTE

MUA (mail user agent)  $\Rightarrow$  YAHOO MAIL

- detto anche mailer, implementa un'interfaccia utente
- non deve essere sempre attiva

MTA (mail transmission agent)

- processo che fa da intermediazione nel processo di trasmissione. È più stabile possibile

• Il computer il cui MTA è quello a cui si rivolge il MUA per l'outgoing (OUTGOING MAIL SERVER)

• Ogni dominio offre/riceve una lista di host che hanno gli MTA che sono incaricati di ricevere posta per il dominio Mail Exchange

• Il computer che ha l'MTA del cui MUA entrano le poste (può coincidere con l'exchange)

**Incoming MAIL SERVER**

Il MUA generalmente gestisce sia l'outgoing sia l'arrivo di messaggi, quindi ha collegati a questi gli MTA di incoming sia di outgoing. Per le config di questi servizi si usa il nome della macchina  $\Rightarrow$  l'IP così può variare nel tempo

**OUTGOING MAIL SERVER MTA:**

- MSA: message submission server si usa per spedire le mail, ha MTA

MTA

## DEFINIZIONE OPERAZIONE

SALVATAGGIO MAIL:  $\rightarrow$  solo in file system locale etc.

## SPEDIZIONE:

MUA trasmette a Outgoing mail server  
MAIL SERVER chiede a DNS la lista dei  
mail e exchange della destinazione  
e li manda a soli questi. (occhiando per sicurezza)

## SERVIZI AUSILIARI

DNS: Serve per individuare i mail exchange  
 $\rightarrow$  NFS: Network file system

13/01/22

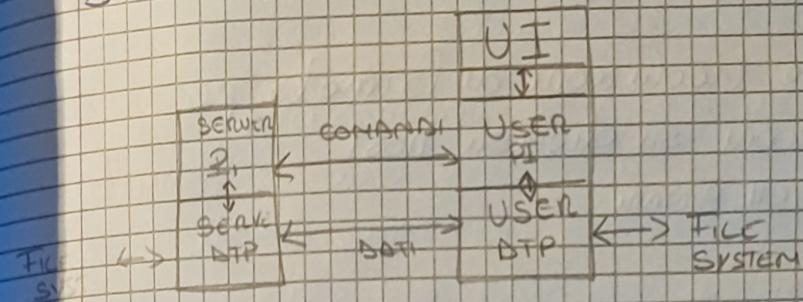
## FTP - File Transfer Protocol

DIALOGO BASATO SU TCP (PK sono grandi pacchetti)

↳ una connessione per i comandi una per i file

Pi: Protocol interpreter

DTP: Data transfer process



Mentre la sessione di trasferimento comandi è unica, la sessione di trasferimento dati ha più connessioni diverse associate per un solo file.

### COMANDI

- STORE
- DELETE
- APPEND
- RETRIEVE

### PARAMETRI DA PASSARE:

- REPRESENTATION TYPE: (come sono rappres.)
- TRANSFER MODE
- PORT

### INSTANZIAZIONE CONN. COMANDI

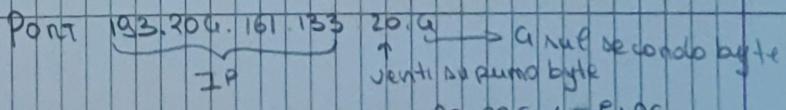
USER Pi usa porta qualsiasi (definita da S.O.)

SERWER Pi usa la porta TCP 21

### CONNESSIONE DATI

#### INSTANZIATA TRA I DUE DTP

La porta che usa il server per spedire i dati viene specificata dal server tramite comando PORT  
→ Lo USER Pi definisce cioè cosa dove vuole ricevere i pacchetti. Il server usa la porta nota 20.



comunica  
il suo numero  
porta e  
la conduttrice  
IP su  
una connetta

## CLOSURE TRANSFERIMENTO

Se il file si sposta dal USER al SERVER,  
la connessione viene chiusa dall'utente  
altrimenti vuole il contrario  
=> Chiudere una connessione significa  
che il trasferimento è finito

## RAPPRESENTAZIONE FILE

FTP effettua le traduzioni di formato per  
le rappresentazioni su diversi host

• SUPPORTA FILE: ASCII, BINARIO ...

## COMANDI

USER NAME (USER)

PASSWORD (PASS)

CHANGE WORKING DIRECTORY (CWD)

LOG OUT (QUIT)

PORT

MOUNT

PREFERENT. TYPE (TYPE)

PASSIVE (PASV): chiede al server di mettersi in  
ascolto su uno port.

## RISPOSTA IN CODICI

1 yz preliminare pos

2 yz positiva

3 yz intermedia positiva

4 yz negativa transittiva

5 yz negativa permanente

RIVEDI

Può funzionare in Active Mode / Passive

13/01/22

## ACTIVE MODE

Si ha quando il server si collega in modo  
altro verso la porta del client.  $\Rightarrow$  il server  
si comporta come client TCP per la conn. dati

$\rightarrow$  Uso porte

- Solo le porte 21 può essere raggiungibile  
dall'esterno del server

- Uso le porte note 20 come client

Lato client

- Porta qualsiasi per connessione comandi

- Porta qualsiasi per ciascuna connessione dati

d'FTP client si comporta come server PFC:

- Così le porte note del server usate siano  
sempre le porte 20 e 21

- Se non si comportasse come server all'istruzione  
della connessione dati il server si connetterebbe

, non avendo altri molenzi specifici, allo porta 20. Poiché è possibile che i processi siano altri su una macchina, se li mettessi in ascolto sulla stessa porta rischiai che i dati di uno vengano consegnati all'altro e viceversa.  
=> Ogni processo comunica al server se quel porta si mette in ascolto per ricevere i dati.

## Eutore il comando Port

Poiché per instaurare una connessione tcp efficiente è necessario avere una quadrupla (IP-PORTA, IP porta) per ogni connessione. Ossia si potrebbe evitare del fatto ~~del~~ ~~seu~~ client di usare 2 porte distinte poiché comunque non c'è s'ebbe equivoco sulle connessioni ~~dati~~/com. essendo destinate su due porte distinte del server (20/21)

→ Non posso omettere il comando test pk:

se eseguissi un breve lasso di tempo lo stesso comando del client verso il server ovviamente il server appena arriva la seconda richiesta dovrebbe abbattere la connessione instaurata prima per fare spazio . fil.

=> Poiché Tcp del server si comporta come client ~~del~~ al momento delle chiusure deve aspettare del tempo affinché i porti vengano rere di nuovo disponibili  
↳ 2x lifetime del segment

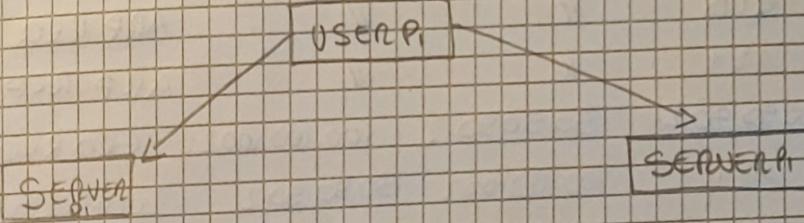
Poiché spesso non è possibile instaurare connessioni TCP con il client che si trova su un'oltre lan e di conseguente non si può raggiungere facilmente una porta  
=> Uso Passive Mode

20/01/2022

## TRASFERIMENTO FILE

### Passive Mode

Il server ~~ha~~ instaura una connessione a seguito di una richiesta PASV di User Pi



OPPURE

Rende possibile la protezione obbligata di uno user. (Lo user non espone porte)