

LIVELLO 1 FISICO

La maggior parte degli esercizi del livello fisico sono richiamati in esercizi di altri livelli. Le cose più importanti da sapere sono:
 saper disegnare il diagramma spazio-tempo, saper calcolare i tempi, il ritardo complessivo, l'ampiezza di banda, sapere le velocità dei mezzi di trasporto e sapere come funzionano i protocolli store-and-forward e cut-through

2) ritardo complessivo frame: Supporre una architettura di rete composta da un host A, host B e interconnessi dalla apparato di rete C.

A comunica con C tramite **etere** a **10 Mbit/s** per una distanza di **100mt**.

C comunica con B tramite **fibra ottica** a **1 Gbit/s** per una distanza di **2000 mt**.

L'apparato di rete può comportarsi nei seguenti modi:

- C memorizza il frame ricevuto da A; quando il frame è stato ricevuto completamente, C inizia subito la trasmissione verso B.

- C introduce un tempo di inoltro per ogni bit di **20 microsecondi**.

Realizzare il diagramma spazio tempo per i 2 casi e determinare il ritardo complessivo per un frame di **1 KByte**.

Spiegazione:**Tempo di trasmissione:**

Tempo che impiega una sequenza di bit ad uscire dall'interfaccia di rete.

$$t_{tr} = \frac{\text{numero bit}}{\text{bit rate}} \quad \text{bit rate} \rightarrow \text{velocità di trasmissione dati (Es: 10 Mbit/s)}$$

Tempo di propagazione:

Tempo che impiega un bit a percorrere il mezzo

$$t_{pr} = \frac{l}{v} \quad \begin{array}{l} l \rightarrow \text{lunghezza mezzo} \\ v \rightarrow \text{velocità di propagazione} \end{array}$$

Velocità di propagazione: **aria/etere** = $3 \cdot 10^8$ m/s **fibra ottica/rame** = $2 \cdot 10^8$ m/s

Tempo di elaborazione (o inoltro):

t_{elab} è dovuto al processamento dei dati.

Tempo di attesa:

Dipende dal carico della rete ed è trascurabile se la rete è scarica.

Ritardo complessivo:

Somma dei tempi precedenti.

Round-Trip Time (RTT):

E' il tempo di andata-ritorno di un bit.

Grafico (consiglio):

Dato un intervallo di tempo (preso dall'asse verticale):

- a t_{tr} e tempo di attesa corrisponde una linea verticale;
- a t_{pr} corrisponde una linea diagonale;

Svolgimento – store-and-forward – caso 1):

$$1 \text{ KB} = 8 \text{ Kbit} = 8000 \text{ bit} = 8 \cdot 10^3 \text{ bit}$$

$$t_{\text{tr_etere}} = \frac{\text{numero bit}}{\text{bit rate}} = \frac{8 \cdot 10^3}{10 \cdot 10^6} = \frac{8}{10 \cdot 10^3} = 8 \cdot 10^{-4} \text{ s}$$

$$t_{\text{pr_etere}} = \frac{1}{v} = \frac{100}{3 \cdot 10^8} = \frac{1}{3 \cdot 10^6} = 3,3 \cdot 10^{-7} \text{ s}$$

$$t_{\text{tr_fibra}} = \frac{8 \cdot 10^3}{1 \cdot 10^9} = \frac{8}{1 \cdot 10^6} = 8 \cdot 10^{-6} \text{ s}$$

$$t_{\text{pr_fibra}} = \frac{2000}{2 \cdot 10^8} = \frac{2}{2 \cdot 10^5} = 1 \cdot 10^{-5} \text{ s}$$

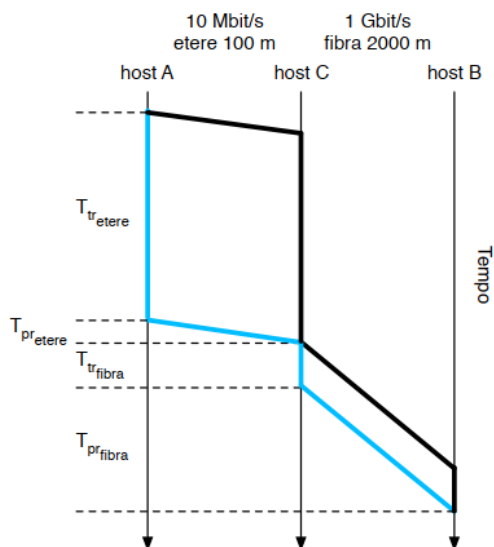
$$t_{\text{tot}} = t_{\text{tr_etere}} + t_{\text{pr_etere}} + t_{\text{tr_fibra}} + t_{\text{pr_fibra}} = (8000 + 3,3 + 80 + 100) \cdot 10^{-7} = 8183,3 \cdot 10^{-7} = 8,2 \cdot 10^{-4} = 0,82 \text{ ms} \quad (\text{ritardo complessivo})$$

($t_{\text{tr_etere}} > t_{\text{tr_fibra}}$ quindi, la “linea verticale” da A a C è più grande di quella da C a B)

($t_{\text{pr_etere}} < t_{\text{pr_fibra}}$ quindi, la “linea diagonale” da A a C è meno inclinata di quella da C a B)

DIAGRAMMA SPAZIO-TEMPO

CASO 1 STORE-AND-FORWARD:



CASO 2 CUT-THROUGH:

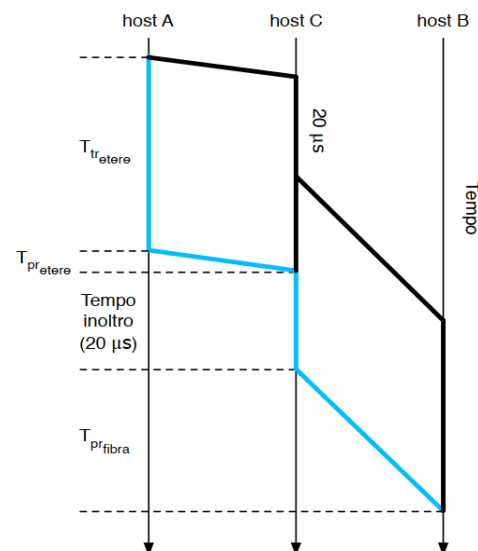


DIAGRAMMA SPAZIO-TEMPO CASO 2 CUT-THROUGH:

Tengo i dati precedenti

$$t_{\text{tr_etere}} = 8 \cdot 10^{-4} \text{ s} \quad t_{\text{pr_etere}} = 3,3 \cdot 10^{-7} \text{ s}$$

$$t_{\text{pr_fibra}} = 1 \cdot 10^{-5} \text{ s}$$

L'ultimo bit da A a C, viene inviato dopo 20 μs.

$$t_{\text{tot}} = t_{\text{tr_etere}} + t_{\text{pr_etere}} + 20 \mu\text{s} + t_{\text{pr_fibra}} = (8000 + 3,3 + 200 + 100) \cdot 10^{-7} = 8303 \cdot 10^{-7} = 8,3 \cdot 10^{-4} = 0,83 \text{ ms} \quad (\text{ritardo complessivo})$$

(Nel grafico del caso 2 si nota che la trasmissione tramite fibra non sfrutta tutta la sua velocità perché riceve dati lentamente dalla trasmissione in etere quindi il $t_{\text{tr_fibra}}$ del caso 1, non è utilizzabile).

Nota bene:

Tpr è sempre una linea dritta, segue QUALSIASI FRAME.

Ttr è una linea diagonale che segue il flusso del trasferimento di bit.

ampiezza di banda: Considerate un collegamento punto-punto in fibra ottica con lunghezza 100Km. Quale valore di ampiezza di banda (bit/s) rende un ritardo di propagazione uguale al tempo di trasmissione per pacchetti di 100byte?

Svolgimento:

$$t_{tr} = \frac{n^{\circ} \text{ bit}}{\text{bit-rate}} \quad t_{pr} = \frac{l}{c/n} = \frac{l}{v} \quad \text{Abbiamo: } t_{tr} = t_{pr} \quad \begin{matrix} T_{tr} = \text{tempo trasmissione} \\ T_{pr} = \text{tempo propagazione} \end{matrix}$$

$$\frac{n^{\circ} \text{ bit}}{\text{bit-rate}} = \frac{l}{v} \rightarrow \frac{\text{bit-rate}}{n^{\circ} \text{ bit}} = \frac{v}{l} \rightarrow \text{bit-rate} = \frac{v}{l} \cdot n^{\circ} \text{ bit}$$

Sapendo che v in fibra ottica è $2 \cdot 10^8$ e $100 \text{ byte} = 800 \text{ bit}$, allora:

$$\text{Ampiezza di banda: } \text{bit-rate} = \frac{2 \cdot 10^8}{100 \cdot 10^3} \cdot 800 = \frac{2 \cdot 10^8}{10^3} \cdot 8 = 2 \cdot 10^5 \cdot 8 = 16 \cdot 10^5 = 1600 \text{ Kb/s}$$

LIVELLO 2 LINK

bisogna conoscere bene i protocolli di controllo per l'accesso multiplo (CSMA/CD CSMA/CA) e come funziona l'algoritmo di backoff, il protocollo sliding windows.

Probabilità collisione: Tre stazioni A, B e C condividono una LAN con algoritmo esponenziale binario per la gestione del backoff e vorrebbero spedire un frame al prossimo slot.

- la stazione A inizia ora a trasmettere
- la stazione B ha avuto 2 collisioni negli slot precedenti
- la stazione C ha avuto una collisione

Qual è la probabilità di collisione al prossimo tentativo?

Info: se in d ho avuto n collisioni, allora: $p(d) = \frac{1}{2^n}$

$$\text{Probabilità: } p(a) = 1 \quad p(b) = \frac{1}{4} \quad p(c) = \frac{1}{2}$$

$$\text{Calcolo delle probabilità: } \frac{1}{4} + \frac{1}{2} - \left(\frac{1}{4} \cdot \frac{1}{2} \right) = \frac{5}{8}$$

In una rete locale 1-persistente con algoritmo di backoff esponenziale binario, le stazioni A B e C hanno dati da trasmettere. La stazione A ha già avuto 2 collisioni, la stazione B inizia ora a trasmettere, la stazione C ha avuto una collisione. Qual'è la probabilità di collisione al prossimo tentativo?

$$\text{Probabilità} = \frac{1}{2^n}$$

$$P(A) = \frac{1}{2^2} \quad P(B) = 1 \quad P(C) = \frac{1}{2}$$

$$\text{probabilità collisione} = \left(\frac{1}{2^2} + \frac{1}{2} \right) - \left(\frac{1}{2^2} \cdot \frac{1}{2} \right) = \frac{5}{8} \quad \text{cioè } 62,5\%$$

7. In una rete WiFi 802.11b a 11Mb/s, con protocollo a contesa (DCF) e il tempo SIFS di 10microsec, A deve inviare a B distante 300 metri un datagramma di 2200 byte che verrà frammentato in 2 frame della stessa dimensione. Trascurare i tempi di trasmissione dei frame di controllo.

Scrivere su questo foglio i tempi di trasmissione e di propagazione di un frammento e la velocità effettiva di trasferimento (bit trasmessi / tempo complessivo).

Sul foglio allegato scrivere il diagramma spazio-tempo della comunicazione e tutti i calcoli.

risposta:

Spiegazione:

- SIFS (Short Inter Frame Space): per separare i Frame di una singola trasmissione
- RTS, CTS e ACK (frame di controllo): quando la stazione deve trasmettere manda un RTS, quindi attende il CTS dal destinatario ed invia i Frame. Per ogni frame inviato attiva un Timer e attende un ACK. Se non arriva l'ACK (CRC errato, ..) il protocollo si ripete.

$$t_{trframe} = \frac{\text{numero bit}}{\text{bit rate}} =$$

WiFi è in etere quindi $v = 3 \cdot 10^8$

$$t_{pr} = \frac{l}{v} =$$

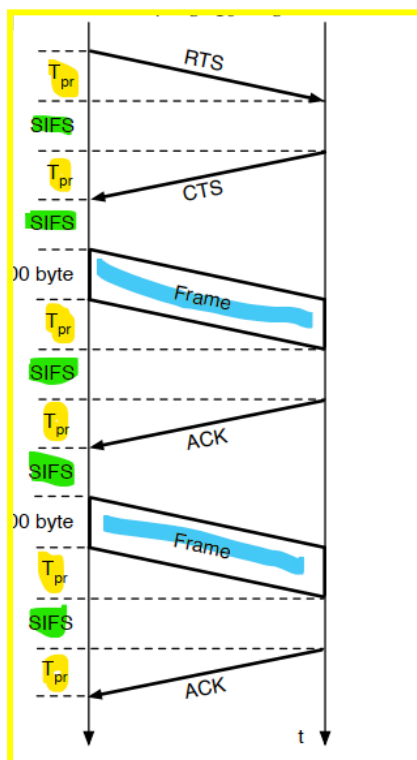
numero bit= 2200 byte= 17600 b

il frame viene frammentato in due= $17600/2 = 8800$ b

lunghezza = 300 m

sifs= 10 us

v. trasmissione= 11 Mb/s = 11000000 b/s



$$T_{tr} = n \cdot \text{bit} / V_{tr} = 8800 \text{ b} / 11000000 \text{ b/s} = 0.0008 \text{ s} = 800 \text{ us}$$

$$T_{pr} = \text{Lunghezza} / v = 300 / 3 \cdot 10^8 = 1 \text{ us}$$

$$T_{tot} = (\text{SIFS} \cdot 5) + (T_{pr} \cdot 6) + (T_{tr} \cdot 2) = (10 \cdot 5) + (1 \cdot 6) + (800 \cdot 2) = 1656 \text{ us}$$

$$\text{Velocità effettiva} = \frac{n \cdot \text{bit}_{tot}}{t_{tot}} =$$

$$17600 / 1.6 \cdot 10^{-3} = \pm 1 \text{ Mbps (1062801)}$$

In una rete WiFi 802.11g a 54Mb/s con protocollo a contesa (DCF) A deve inviare a B, distante 100 metri, un datagramma di 1000 byte che verrà frammentato in 2 frame da 500 byte. Disegnare il diagramma spazio-tempo della comunicazione ipotizzando un tempo SIFS di 10microsec e determinare la velocità effettiva di trasferimento (bit/sec). Trascurare i tempi di trasmissione dei frame di controllo.

LIVELLO 3 RETE

da sapere: Subnetting, Supernetting, divisione sotto reti, IPV6, instradamento (routing), bridge, Nat, DHCP, ARP.

(Tabelle utili: video "[1-Subnet](#)")

1) **Le classi B sono 16K. Si supponga che invece di utilizzare 16 bit per la classe B se ne fossero utilizzati 20. Quante classi B ci sarebbero state?**

In classe B ci sono 16 bit utilizzati per la rete ma considerando che i primi 2 bit sono sempre fissi a "10", allora i bit che veramente possono variare per la rete, scendono a

$$14 \cdot 2^{14} = 16384 = 16K.$$

Se quindi avessi 20 bit, tenendo 2 fissi, avrei: $2^{18} = 262144 = 262K$ classi B.

2) **Subnetting (suddividere rete C): la classe C 192.1.1.0 viene assegnata a 3 Dipartimenti A (100 host), B (60 host) e C (40 host), ciascuno con la propria LAN. Determinare il subnetting ottimale per questa situazione.**

Spiegazione: (video "[6-Subnet](#)")

Spiegazione: (video "6-Subnet")

- host: 2^x potenza di 2 che si avvicina di più (in eccesso) al valore dei "pc".
- bit host: x (l'elevamento della potenza calcolata in "host").
- bit sm (subnet mask): $32 - x$
- rete e broadcast: parte da quella iniziale (192.1.1.0) e somma ogni volta il numero di host (192.1.1.127).
- gateway: broadcast - 1
- subnet mask: in questo esercizio ho una classe C che ha subnet mask = /24 = 255.255.255.0

Usando come esempio la prima riga: bit sm - 24 = 25 - 24 = 1.

Dalle "tabelle utili (video)" ricavo che 1 → 128 valore che aggiungerò alla subnet mask della classe C ottenendo: 255.255.255.128

Svolgimento esercizio:

	pc	host	bit host	bit sm	rete	broadcast	gateway	subnet mask
A	100	127	7	25	192.1.1.0	192.1.1.127	192.1.1.126	255.255.255.128
B	60	64	6	26	192.1.1.128	192.1.1.191	192.1.1.190	255.255.255.192
C	40	64	6	26	192.1.1.192	192.1.1.255	192.1.1.254	255.255.255.192

A)

A ha bisogno di 100 host, il minimo che possiamo assegnarli è 2^7 cioè 128 hosts

B ha bisogno di 60 Host, il minimo che possiamo assegnarli è 2^6 cioè 64 hosts

C ha bisogno di 40 Host, quindi 2^6 cioè 64 hosts

quindi **A** avrà gli indirizzi da [192.1.1.0–192.1.1.127] = 128 host

di cui **192.1.1.127** indirizzo di gateway

B avrà gli indirizzi da [192.1.1.128–192.1.1.191] = 64 hosts

di cui **192.1.1.191** indirizzo di gateway

C avrà [192.1.1.192–192.1.1.255] = 64 hosts

di cui 192.1.1.255 indirizzo di gateway

3) Supernetting: un router deve annunciare che gli indirizzi da 10.10.0.0 a 10.10.50.255 e da 10.10.64.0.0 a 10.10.127.255 sono raggiungibili attraverso l'interfaccia 1, mentre gli indirizzi da 10.10.60.0 a 10.10.63.255 sono raggiungibili attraverso l'interfaccia 2. Aggregare correttamente gli indirizzi utilizzando il minor numero di reti.

Interfaccia 1:

da 10.10.0.0 = 00001010. 00001010.0| 0000000.0000000

a 10.10.127.255= 00001010. 00001010.0| 1111111.1111111

abbiamo bisogno dei ultimi 15 bit quindi $32-15=$ /17

10.10.0.0/17

INTERFACCIA 2 :

da 10.10.60.0.0 = 00001010. 00001010. 00111100.00000000

a : 10.10.63.255= 00001010. 00001010. 00111111.00000000

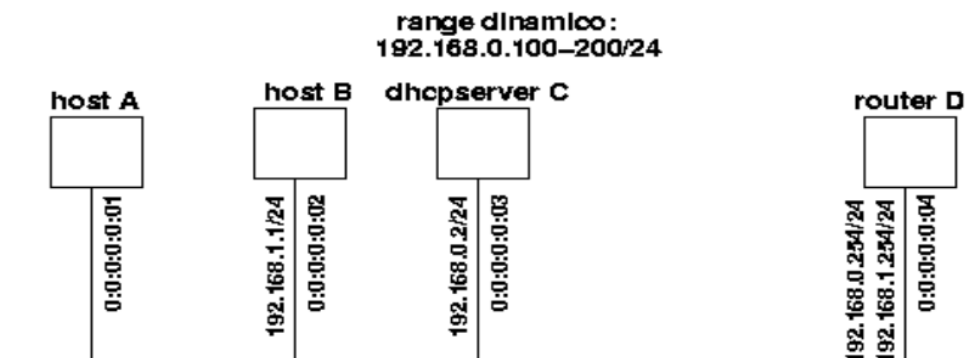
cambiano solo gli ultimi 10 bit quindi $32-10=$ /22

10.10.60.0 /22

NOTA BENE:

Gli indirizzi della seconda interfaccia sono inclusi nella prima, ma saranno scelti perché hanno un netmask più lungo.

Considerare la configurazione di rete di figura. L'host A configurato con indirizzo IP dinamico si accende e invia un ping a B. Descrivere la sequenza di tutti i pacchetti che transitano sulla rete dettagliandone il contenuto (indirizzi di mittente e destinatario) a livello 2 e 3. Specificare le azioni da intraprendere per migliorare l'instradamento di questi pacchetti.



Spiegazione MAC (lv. 2), IP (lv. 3), ARP (lv. 3):

MAC:

- Indirizzo unico; non esistono due schede di rete con indirizzo MAC uguale.

IP:

- Indirizzo che identifica un nodo all'interno di una rete locale. Viene assegnato ad un "pc" quando questo entra in rete (unico, ma è temporaneo, e serve solo dentro una rete).

ARP:

- Protocollo per conoscere il MAC Address di destinazione, noto l'IP di destinazione.

Ping:

- Comando attraverso il quale il computer tenta di inviare dei pacchetti di dati all'IP di destinazione indicato, per capire se è in grado di comunicare con quest'ultimo o meno (in specifico: tramite ping viene inviato un pacchetto ICMP di tipo "echo request" e si rimane in attesa di un pacchetto ICMP di tipo "echo reply" in risposta).

Svolgimento esercizio:

1) Acquisizione indirizzo IP dinamico (per Host A):

// Accendo host A (indirizzo IP dinamico) che chiede un indirizzo IP libero a DHCP mandando un broadcast (a tutti, perché non sa chi è il DHCP).

// Il broadcast chiede: "se c'è un servizio di DHCP, può mandarmi un indirizzo?".

- DHCP discover: mac A, IP 0.0.0.0 → broadcast

// C (il DHCP) manda il secondo pacchetto della rete verso A, per dirgli che c'è.

- DHCP offer: mac C, IP C → mac A

// A richiede l'indirizzo IP.

- DHCP request: mac A, IP 0.0.0.0 → mac C:

// A ottiene un indirizzo IP (per ipotesi è 192.168.0.100/24).

- DHCP ack: mac C, IP C → mac A, IP A

2) Ping da Host A a Host B:

L'Host B appartiene ad una rete diversa da A, quindi A non può fare una consegna diretta; fa una consegna indiretta. (A 192.168.0.*) (B 192.168.1.1)

```
// Da A a D
// Dalla tabella di routing so che devo passare dal router D per comunicare con B.
// Cerco quindi l'indirizzo mac D
- ARP request: [lv. 2] mac A → broadcast
               [lv. ARP] IP A, mac A, IP B, ?
// Il router D riceve la richiesta e restituisce ad A il mac D
- ARP reply:   [lv. 2] mac D → mac A
               [lv. ARP] IP A, mac A, IP B, mac D
// Mando il pacchetto con destinazione finale IP B e con mac D
- ICMP echo-request: [lv. 2] mac A → mac D
                    [lv. 3] IP A → IP B

// Da D a B
// Cerco l'indirizzo mac B di destinazione
- ARP request: [lv. 2] mac D → broadcast
               [lv. ARP] IP D, mac D, IP B, ?
// B riceve la richiesta e restituisce a D il mac B
- ARP reply:   [lv. 2] mac B → mac D
               [lv. ARP] IP D, mac D, IP B, mac B
// Il router D, manda il pacchetto (inviato da A) a B
- ICMP echo-request: [lv. 2] mac D → mac B
                    [lv. 3] IP D → IP B

// Da B a D ad A
// Invio risposta da Host B a Host A
- ICMP echo-reply: [lv. 2] mac B → mac D
                  [lv. 3] IP B → IP D
- ICMP echo-reply: [lv. 2] mac D → mac A
                  [lv. 3] IP D → IP A
```

1) comunicazione tra A-C (4 pacchetti DHCP)

- A si accende e chiede con un messaggio broadcast chi è il server DHCP (DHCP discover)
- dhcp(C) risponde ad A con un DHCP Offer
- A richiede un indirizzo Ip a C tramite un (DHCP Request)
- C gli offre un indirizzo con un (DHCP ACK)

2) comunicazione tra A-D (2 ARP, 1 ECHO)

- dato che B non è nella stessa rete di A, deve uscire dalla rete tramite il router D
- A chiede l'indirizzo del Router D tramite un ARP Request
- D risponde con il suo indirizzo mac tramite un ARP Reply
- A invia un pacchetto ICMP tramite un ping (ECHO-Request) a D in modo che lo instrada verso B

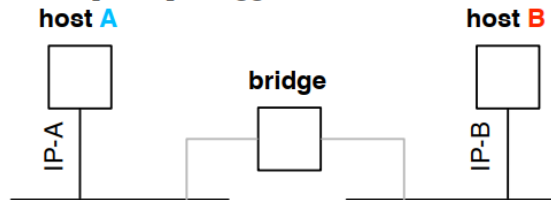
3) comunicazione tra D-B (2 ARP, 2 ECHO)

- D chiede l'indirizzo mac a B tramite un ARP Request
- B risponde con un ARP Reply
- D gli invia il pacchetto ICMP (inviato precedentemente da A) con all'interno un (ECHO-REQUEST)
- B risponde con un ECHO REPLY che D reindirizza ad A.

3) Miglioramento:

Si dice che A e B che **sono in comunicazione diretta** (aggiungendo una riga nelle tabelle di routing di entrambi gli host); si ottiene che A potrà inviare pacchetti a B in modo diretto sfruttando il collegamento che hanno

Bridge (livello 2 e 3): se nell'esercizio appena fatto, al posto del router C avessi avuto un bridge, con A e B senza connessione fisica diretta, comunque avrei potuto inviare un pacchetto (ping) da A a B in modo diretto, facendo solo questi passaggi:



1)ARP

A chiede l'indirizzo MAC di B tramite un ARP Request

B risponde con il suo indirizzo MAC tramite un ARP Reply

2)ECHO

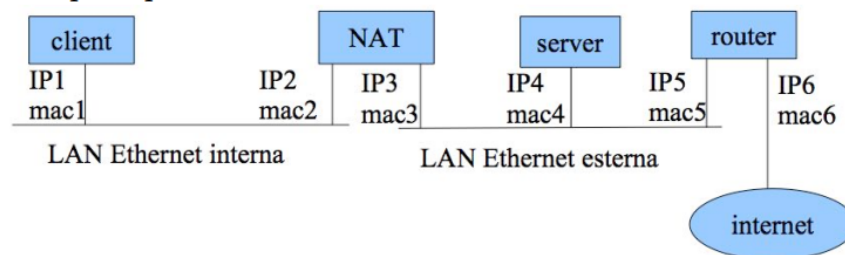
A gli invia un Ping, cioè un pacchetto ICMP contenente un (ECHO-Request)

B risponde con un ECHO- Reply

5 - [2015-2016]) NAT: Si consideri la rete di figura in cui:

- il client ha un indirizzo privato IP1 e un default router IP2
- il server ha un indirizzo pubblico IP4 e un default router IP5

Subito dopo l'accensione delle macchine il client genera "ping IP4" Elencare tutti i Frame conseguenti al ping, dattagliando per ogni frame la stratificazione dei protocolli utilizzati e le relative informazioni principali.



// Da client a NAT - LAN interna

// Il client chiede chi ha l'indirizzo IP4

- ARP request: [lv. 2] mac1 → broadcast
[lv. ARP] IP1, mac1, IP4, ?

// NAT si accorge di essere collegato con l'IP4 quindi, risponde al client

- ARP reply: [lv. 2] mac2 → mac1
[lv. ARP] IP1, mac1, IP4, mac2

```

// il client invia "echo-request" al NAT
- ICMP echo-request: [lv. 2] mac1 → mac2
                    [lv. 3] IP1 → IP4

    // Da NAT a server - LAN esterna
// NAT chiede chi ha l'indirizzo IP4
- ARP request: [lv. 2] mac3 → broadcast
              [lv. ARP] IP3, mac3, IP4, ?
// il server risponde al NAT
- ARP reply: [lv. 2] mac4 → mac3
            [lv. ARP] IP3, mac3, IP4, mac4
// NAT invia "echo-request" al server
- ICMP echo-request: [lv. 2] mac3 → mac4
                    [lv. 3] IP3 → IP4

// Il server risponde al client (attraverso il NAT)
- ICMP echo-reply: [lv. 2] mac4 → mac3
                  [lv. 3] IP4 → IP3
- ICMP echo-reply: [lv. 2] mac2 → mac1
                  [lv. 3] IP2 → IP1

```

1) Da IP1 a NAT (2 ARP, 1 ECHO)

- IP1 richiede tramite messaggio broadcast l'indirizzo di IP4 con un ARP Request
- NAT(mac 2) ha l'indirizzo di IP4 e risponde con un ARP Reply
- IP1 invia un Echo-Request diretto a IP4 tramite NAT

2) da NAT a IP4 (2 ARP, 3 ECHO)

- NAT (mac3) chiede l'indirizzo MAC di IP4 tramite ARP Request
- risponde il server con l'indirizzo IP4 con un ARP Reply
- NAT invia il pacchetto ICMP (ECHO-Request) a IP4
- IP4 risponde con un ECHO-Reply a NAT(mac 3)
- NAT(mac 2) invia il ECHO-Reply a IP1

LIVELLO 4 TCP

1) Finestra congestione (Slow Start – Fast Recovery): Si supponga che la finestra di congestione TCP sia impostata a 18 KB e che si verifichi un timeout. Quanto sarà grande la finestra se le 4 trasmissioni successive hanno successo? Si assuma che la dimensione massima del segmento sia di 1 KB.

Spiegazione:

(**cwnd** = finestra di congestione: indica quanti byte il mittente può avere/inviare sulla rete contemporaneamente).

(**MSS** = dimensione massima segmento; **RWND** = spazio libero (accettabile) sul buffer).

All'accensione:

- di solito la dimensione massima della cwnd è 64 KB e la soglia 32 KB ($64/2$)
- Parto con la cwnd di 1 KB cioè **cwnd=MSS = 1KB**.
- Ogni volta che svolgo l'invio dei segmenti con successo, la cwnd **raddoppia** (se la finestra è grossa 16 KB invio 16 segmenti e poi raddoppio la dimensione).
- Quando la dimensione della cwnd **raggiunge la soglia**, la cwnd aumenta di 1 ad ogni trasmissione avvenuta con successo (**cresce linearmente**).

Metodi diversi di proseguimento:

- **Slow Start:** quando avviene un "timeout" (perdita di un pacchetto), si ritrasmette il pacchetto perso, la soglia diventa la metà della dimensione attuale della cwnd, e la cwnd riparte da 1 KB (si riparte dal passo iniziale, ma con soglia ridotta).
- **Fast Recovery:** quando avviene un "timeout" (perdita di un pacchetto), si ritrasmette il pacchetto perso, la soglia diventa la metà della dimensione attuale della cwnd, e la cwnd riparte dalla soglia (ed aumenta di uno ad ogni trasmissione avvenuta con successo).

Svolgimento – Slow Start:

- La cwnd è impostata a 18 KB
- Arriva un timeout che porta la finestra a 1 KB; ritrasmette il segmento che è stato perso; la soglia diventa $9\text{KB} \left(\frac{18\text{KB}}{2} \right)$.
- 4 trasmissioni successive con successo:
 - 1) Trasmetto 1 segmento (1 KB); la cwnd diventa di 2 KB.
 - 2) Trasmetto 2 segmenti (2 KB); la cwnd diventa di 4 KB.
 - 3) Trasmetto 4 segmenti (4 KB); la cwnd diventa di 8 KB.
 - 4) Trasmetto 8 segmenti (8 KB); la cwnd dovrebbe diventare di 16 KB ma con la soglia, si ferma a 9 KB.
- La cwnd sarà grande 9 KB.

Svolgimento – Fast Recovery:

- La cwnd è impostata a 18 KB
- Arriva un timeout; ritrasmette il segmento che è stato perso; la soglia diventa $9\text{KB} \left(\frac{18\text{KB}}{2} \right)$; si riparte dalla soglia di 9 KB.
- 4 trasmissioni successive con successo:
 - 1) Trasmetto 9 segmenti (9 KB); la cwnd diventa di 10 KB.
 - 2) Trasmetto 10 segmenti (10 KB); la cwnd diventa di 11 KB.
 - 3) Trasmetto 11 segmenti (11 KB); la cwnd diventa di 12 KB.
 - 4) Trasmetto 12 segmenti (12 KB); la cwnd diventare di 13 KB.
- La cwnd sarà grande 13 KB.

Esercizio 4.1:

Si consideri la trasmissione in upload di un file di 6 KB (6000 byte) con finestra di ricezione $rwnd$ fissa di 2 KB, e MSS di 1 KB, valori iniziali di sequenza = 0, senza gestione della congestione.

Disegnare il diagramma spazio tempo elencando tutti segmenti scambiati e specificando per ogni segmento valori plausibili di Sequenza, Riscontro e Bit di codice attivi, compreso l'handshake iniziale e finale. Si applichi il piggybacking ove possibile.

Svolgimento:

RWND= 2 KB;

DATI = 6 KB;

MSS= 1 KB;

seq= 0;

