

Valutazione delle prestazioni di una rete

Per valutare le prestazioni di una rete in termini di capacità trasmissiva, affidabilità e qualità a livello fisico si possono utilizzare parametri quali:

1. latenza (*latency*);
2. jitter o *delay variation* (*variazione di ritardo*);
3. *Round Trip Time* (RTT) o tempo di andata e ritorno;
4. *Throughput o goodput*, è la velocità di informazione, o bit rate netto, messa a disposizione delle applicazioni che operano in rete;
5. perdita di pacchetti (*packet loss*);
6. *banda (digitale)*, a livello fisico è la capacità trasmissiva nominale, corrispondente al *bit rate lordo* messo a disposizione dalla tecnologia dello strato fisico utilizzata; per esempio nelle LAN Ethernet 10BASE-T il bit rate lordo è 10 Mbit/s, nelle 100BASE-TX è 100 Mbit/s, nelle 1000BASE-T è 1 Gbit/s.

1. Latenza

Uno dei parametri che incidono sulle prestazioni di una rete è la *latenza*.

In una *comunicazione unidirezionale senza correzione d'errore* la latenza può essere definita come il tempo che intercorre fra l'emissione di una PDU (Protocol Data Unit) da parte della sorgente e la ricezione della PDU stessa da parte del destinatario.

Poiché lo strato 2 OSI controlla e gestisce la comunicazione su un canale fisico, organizzandola come trasmissione di *frame* (2-PDU), in generale la latenza dipende da:

- tempo di trasmissione di un frame (2-PDU)¹; il tempo di trasmissione di un frame è calcolabile come rapporto fra la dimensione del frame espressa in bit (D_F), e il bit rate (BR) con cui si opera a livello fisico (velocità di trasmissione):

$$\Delta t_{Frame} = \frac{D_{F_}[bit]}{BR_ [bit/s]} \quad [s]$$

- ritardo di propagazione su ciascun mezzo trasmissivo, calcolabile come rapporto tra la lunghezza del collegamento fisico (una tratta) e la velocità di propagazione dei segnali:

$$\Delta t_{prop.} = \frac{L_{[m]}}{v_{prop.}[m/s]} \quad [s]$$

- ritardi causati dal transito attraverso gli apparati di rete (switch ecc), dovuti a
 - ritardi intrinseci ($\Delta t_{app.}$) tra la ricezione su una porta di ingresso e la trasmissione su una porta di uscita dell'apparato;
 - eventuali ritardi dovuti agli accodamenti per differenti livelli di priorità e/o per un traffico in rete elevato (Δt_Q).

¹ Nel calcolo andrebbe tenuto in conto anche il processo di incapsulamento che porta alla formazione del frame (strato 2 OSI).

Ipotizzando per semplicità che si utilizzi sempre lo stesso tipo di frame, che la dimensione dei pacchetti (strato 3 OSI) non superi l'MTU (*Maximum Transmission Unit*) consentito dai frame (assenza di frammentazione dei pacchetti), che vi siano N apparati di rete interposti tra sorgente e destinazione, la latenza (LT) si può quindi calcolare come:

$$LT = \Delta t_{Frame} + \Delta t_{Pr.op.} + \sum_1^N i (\Delta t_{App._i} + \Delta t_{Q._i}) \quad [s]$$

1.1. Latenza di uno switch Ethernet

Nel caso di switch Ethernet la latenza dell'apparato dipende anche dalla modalità con cui esso opera, modalità che può essere di due tipi:

- *store and forward*;
 lo switch deve ricevere il frame completo su una porta di ingresso prima di iniziare a trasmetterlo su una porta di uscita; in questo caso la latenza viene definita² come l'intervallo di tempo che intercorre fra la ricezione dell'ultimo bit del frame e l'inizio della trasmissione del primo bit del frame stesso (sintetizzabile come LIFO, *Last In First Out*);
 la latenza dipende quindi dal tempo di ricezione del frame

$$\Delta t_{Frame} = \frac{D_F \text{ [bit]}}{BR \text{ [bit / s]}} \quad [s]$$

e dal tempo necessario al suo trasferimento sulla porta di uscita ($\Delta t_{I \rightarrow O}$):

$$\Delta t_{switch} = \Delta t_{Frame} + \Delta t_{I \rightarrow O}$$

- *cut-through*;
 lo switch analizza i primi bit del frame almeno fino a che individua l'indirizzo MAC di destinazione, dopodiché trasferisce i bit sulla porta di uscita e inizia a trasmettere; in questo caso la latenza dello switch³ viene definita come l'intervallo di tempo che intercorre fra la ricezione del primo bit sulla porta di ingresso e la trasmissione del primo bit sulla porta di uscita, sintetizzabile come FIFO (First In First Out); in questo caso la latenza dipende dal numero di bit analizzati (almeno i primi 112 bit costituiti dal preambolo+SFD, 64 bit, e dall'indirizzo MAC di destinazione, 48 bit) e dal ritardo introdotto nel trasferimento dei bit dalla porta di ingresso alla porta di uscita.

2. Jitter

Con il termine *jitter* si indica la variazione di ritardo (*delay variation*) che delle PDU emesse a intervalli di tempo regolari subiscono nell'andare dalla sorgente alla destinazione. Il *jitter* è quindi causato da variazioni di latenza.

Fissato quindi il ritardo (latenza) nominale, un valore positivo di jitter indica che il ritardo subito da una PDU è maggiore di quello nominale, mentre un valore negativo di jitter indica che il ritardo subito da una PDU è inferiore a quello nominale.

3. Round Trip Time o tempo di andata-ritorno

In una comunicazione *bidirezionale con correzione d'errore* e controllo di flusso il parametro di interesse per quanto riguarda i ritardi è il tempo di andata-ritorno o *Round Trip Time* (RTT).

L'RTT può essere definito come il tempo che intercorre fra l'emissione di una PDU da parte di una sorgente e la ricezione della conferma di corretta ricezione (ACK) da parte della sorgente stessa.

² Per i dettagli si veda l'RFC 1242

³ Si veda l'RFC 1242

L'RTT può quindi essere considerato come la latenza totale che si ha fra l'emissione da parte della sorgente di due PDU consecutive, pari alla somma della latenza subita dalla PDU nella direzione sorgente-destinazione ($LT_{S \rightarrow D}$) e della latenza subita dall'ACK nella direzione destinazione-sorgente ($LT_{D \rightarrow S}$):

$$RTT = LT_{S \rightarrow D} + LT_{D \rightarrow S} \quad [s]$$

dove

$$LT_{S \rightarrow D} = \Delta t_{Frame_dati} + \Delta t_{Pr\ op.} + \sum_1^N i(\Delta t_{App_i} + \Delta t_{Q_i}) \quad [s]$$

$$LT_{D \rightarrow S} = \Delta t_{Frame_ACK} + \Delta t_{Pr\ op.} + \sum_1^N i(\Delta t_{App_i} + \Delta t_{Q_i}) \quad [s]$$

Inoltre:

- sullo strato 3 OSI l'RTT può essere determinato effettuando un *ping* da un host sorgente verso un host destinazione, in quanto viene misurato il tempo che intercorre fra l'emissione di un messaggio ICMP di *Echo request* e la ricezione del corrispondente messaggio di *Echo replay*;
- sullo strato 4 OSI l'RTT può essere definito come il tempo che intercorre fra l'emissione del primo bit di un segmento TCP (o 4-PDU, Protocol Data Unit dello strato 4, nella terminologia OSI) e la ricezione dell'ultimo bit del corrispondente segmento di ACK (conferma di corretta ricezione).

4. Throughput e goodput

Il termine throughput può essere impiegato per definire il bit rate netto che un qualsiasi strato OSI offre allo strato superiore, mentre il termine goodput viene impiegato per definire solo il bit rate netto (o throughput) a livello di applicazione. Dal punto di vista degli utilizzatori di una rete ciò che interessa realmente è la velocità con la quale essi possono effettivamente scambiare informazioni (dati) e quindi qual è il *bit rate netto*, indicato spesso come *throughput* o *goodput*, che le applicazioni utilizzate hanno a disposizione.

Il *throughput* (*goodput*) dipende da vari fattori:

- prestazioni della rete in termini di banda, perdita di pacchetti, jitter;
- latenza, che a sua volta dipende da
 - tempo di trasmissione di una PDU;
 - ritardo introdotto dai mezzi trasmissivi utilizzati, calcolabile come rapporto fra la loro lunghezza e la velocità di propagazione dei segnali su essi;
 - ritardi introdotti dagli apparati di rete nel trasferire le PDU da un loro ingresso a una loro uscita;
 - eventuali ritardi causati dal grado di congestione della rete, che può rendere necessario un accodamento (bufferizzazione) delle PDU negli apparati di rete prima che possano essere trasmesse;
 - nel caso di implementazione della QoS (*Quality of Service*), dalla priorità che hanno le code nella trasmissione delle PDU; per esempio PDU che trasportano voce e video (generate dal protocollo RTP) sono messe in una coda che ha priorità maggiore rispetto alla coda in cui sono messe le PDU dei protocolli di applicazione per dati (HTTP, FTP ecc.);
- tipo di metodo utilizzato per il controllo di flusso e la correzione d'errore e ritardi da essi introdotti, in quanto incidono sull'RTT.

Come parametri di riferimento con cui confrontare le effettive prestazioni di una rete è possibile calcolare i throughput (goodput) massimi teorici offerti da una rete locale (LAN) che adotta i protocolli della suite TCP/IP e che utilizza la tecnologia Ethernet per implementare gli strati OSI 1 e 2, quando si considerano:

- applicazioni che utilizzano come protocollo di trasporto il protocollo TCP;
- applicazioni che utilizzano come protocollo di trasporto il protocollo UDP.

Nell'effettuare questi calcoli inizialmente si possono fare le seguenti ipotesi semplificative:

- assenza di errori e quindi di ritrasmissioni nel caso di protocollo di trasporto TCP;
- ottimizzazione della configurazione del protocollo TCP in modo tale da consentire l'emissione continua di PDU (segmenti) senza che il controllo di flusso determini sospensioni della trasmissione;
- si trascurano le fasi di inizializzazione e chiusura delle connessioni logiche, così come i ritardi introdotti dai mezzi trasmissivi e dagli apparati di rete;
- i frame Ethernet abbiano un *payload* (campo data) avente la dimensione massima, pari a 1500 Byte, corrispondente all'MTU (*Maximum Transission Unit*).

4.1. Calcolo del throughput massimo teorico offerto dal protocollo TCP

Ipotizziamo di operare con tecnologia Ethernet 100BASE-TX per cui il bit rate lordo è pari a BR=100 Mbit/s. Procediamo quindi nel seguente modo.

- a) Si determina la dimensione (D_F) totale di un frame Ethernet II, che è composto da (B=Byte):
- Preambolo + SFD -> 8 B
 - Indirizzi MAC destinazione e sorgente -> 6+6=12 B
 - Protocol Type -> 2 B
 - Payload (o data o information) -> 1500 B
 - FCS (Frame Check sequence, ottenuto con il metodo CRC-32 che produce 32 bit) -> 4 B

Si ha quindi $D_F = 8 + 12 + 2 + 1500 + 4 = 1526 B$ e quindi $D_F = 1526 \cdot 8 = 12208 bit$

- b) Si calcola il numero massimo di frame/s, N_F , che si possono trasmettere in modo continuativo, noti il bit rate lordo (BR) e la dimensione di un frame completo (D_F [Byte]), a cui va aggiunto l'IFG (*Inter Frame Gap*), corrispondente a 12 B (96 tempi di bit):

$$N_F = \frac{BR}{(D_F + IFG) \cdot 8} = \frac{100 \cdot 10^6}{(1526 + 12) \cdot 8} = \frac{100 \cdot 10^6}{12304} = 8127 \text{ [frame / s]}$$

- c) Si determina la dimensione massima che possono avere i blocchi di bit informativi (dati) veri e propri trasportati nel payload di un segmento TCP, denominata **Maximum Segment Size (MSS)**, sottraendo all'MTU la dimensione degli header (H) dei protocolli degli strati superiori (IPv4, strato 3, e TCP, strato 4); assumendo i valori tipici, pari a $H_{IPv4}=20 B$ e $H_{TCP}=20 B$, si ha:

$$MSS = MTU - H_{IPv4} - H_{TCP} = 1500 - 20 - 20 = 1460 B$$

- d) Si calcola il bit rate netto (BR_{Netto}), o *throughput*; poiché ogni frame trasporta un blocco di dati veri e propri avente dimensioni pari all'MSS si ha che il bit rate netto disponibile per il trasferimento dei bit informativi prodotti dalle applicazioni, è calcolabile come:

$$BR_{Netto} = (N_F) \cdot (MSS) \cdot 8 = 8127 \cdot 1460 \cdot 8 = 94,9 \text{ [Mbit / s]}$$

- e) Si può calcolare quindi il tempo minimo teoricamente necessario per trasferire un messaggio avente una certa dimensione (D_M , espressa in Byte), per esempio un file di 100 MB, che risulta così pari a:

$$\Delta T_{Trasf} = \frac{D_M \cdot 8}{BR_{Netto}} = \frac{800 \cdot 10^6}{94,9 \cdot 10^6} \cong 8,4 \text{ s}$$

Se le applicazioni forniscono blocchi di byte informativi con dimensione inferiore a quella massima (MSS=1460 B), per cui si impiegano frame con *payload* inferiore all'MTU, *diminuisce anche il throughput* in quanto aumenta sia il numero di frame da trasmettere sia il peso percentuale dei byte degli header (detti anche *overhead*).

In alcuni casi, poi, si valuta il *throughput* massimo che il **protocollo IPv4** (strato 3 OSI) offre a un protocollo di trasporto, e non a un protocollo di applicazione, per determinare l'efficienza del trasferimento di pacchetti. In questo caso si ha che la dimensione massima del payload di un pacchetto (che non necessita di frammentazione) è pari a $MTU - H_{IPv4} = 1500 - 20 = 1480$ B, per cui è:

$$BR_{IPv4} = (N_F) \cdot (D_{pacchetto}) \cdot 8 = 8127,4 \cdot 1480 \cdot 8 = 96,2 [Mbit / s]$$

4.2. Calcolo del throughput massimo offerto dal protocollo UDP

Ipotizzando di operare in condizioni ideali e con *payload* Ethernet di 1500 B (pari all'MTU), sapendo inoltre che l'Header del protocollo UDP è di soli 8 B, in quanto il protocollo UDP opera in modalità *connectionless* e quindi non effettua né il controllo di flusso né la correzione d'errore, si ha che la dimensione massima di un blocco di dati incapsulato nel payload di un segmento UDP risulta pari a:

$$D_P = MTU - H_{IPv4} - H_{UDP} = 1500 - 20 - 8 = 1472 \text{ B}$$

Il bit rate netto, o throughput, offerto dal protocollo UDP risulta così pari a:

$$BR_{Netto} = (N_F) \cdot (D_P) \cdot 8 = 8127 \cdot 1472 \cdot 8 = 95,7 [Mbit / s]$$

Il tempo minimo teorico necessario per trasferire un file di 100 MB utilizzando come protocollo di trasporto l'UDP risulta così pari a:

$$\Delta T_{TRASF} = \frac{D_M \cdot 8}{BR_{TH}} = \frac{800 \cdot 10^6}{95,7 \cdot 10^6} \approx 8,3 \text{ s}$$

4.3. Calcolo del throughput offerto da un protocollo a correzione d'errore e controllo di flusso.

Nel caso in cui si implementino il controllo di flusso e la correzione d'errore il bit rate netto (*throughput*) diminuisce per via dei ritardi che queste tecniche introducono.

La correzione d'errore per ritrasmissione ed il controllo di flusso possono essere implementate con le seguenti modalità:

- a) *stop and wait*;
- b) a finestra di trasmissione (*sliding window*).

4.3.1 Stop and Wait

Il metodo *stop and wait* opera nel seguente modo (FIGURA 1):

la sorgente emette una PDU (Protocol Data Unit) alla volta e attende la conferma di corretta ricezione (ACK) prima di trasmettere la PDU successiva.

Come sempre se la PDU non è dello strato 2 (cioè non è un frame) essa viene incapsulata dai protocolli sottostanti e viene trasferita nel *payload* di un frame.

Con il metodo *stop and wait* la latenza totale corrisponde al *Round Trip Time* (RTT, o tempo di andata e ritorno) in quanto prima di passare alla PDU successiva è necessario attendere la conferma di corretta ricezione.

In prima approssimazione, in assenza di congestione di rete (nessun accodamento), trascurando i ritardi di elaborazione e quelli introdotti dagli apparati di rete, ipotizzando che le PDU seguano lo stesso percorso nelle due direzioni per cui i mezzi trasmissivi introducono lo stesso ritardo, l'RTT può essere calcolato sommando la durata di un frame, i ritardi di propagazione (Δt_r) nelle due direzioni e la durata di un frame che trasporta solo l'ACK:

$$RTT \approx \Delta t_{Frame} + 2\Delta t_r + \Delta t_{ACK} [s]$$

La durata di un frame si può calcolare dividendo il numero di bit che la compongono per il bit rate lordo:

$$\Delta t_{Frame} = \frac{D_F \cdot 8}{BR} [s]$$

Il ritardo introdotto dai mezzi trasmissivi può essere calcolato come rapporto tra la loro lunghezza (L) e la velocità di propagazione, che è pari al prodotto tra il fattore di velocità (F_V, valori tipici: 1 per l'aria, 0.67 per i cavi) e la velocità della luce (c = 3 · 10⁸ m/s):

$$t_r = \frac{L_{[m]}}{F_V \cdot 3 \cdot 10^8 [m/s]} [s]$$

L'RTT può anche essere considerato come la somma della durata di un frame e di un IFG (Inter Frame Gap) corrispondente alla somma dei ritardi di propagazione (e di quelli introdotti dagli apparati) e della durata del frame che trasporta l'ACK.

Nella pratica si può determinare l'RTT con delle misure, come per esempio le seguenti:

- l'RTT a livello 3 può essere determinato effettuando un *ping* dall'host sorgente verso l'host destinazione;
- L'RTT a livello di trasporto (strato 4), con protocollo TCP, può essere determinato nella fase di inizializzazione della connessione logica andando a misurare il ritardo che esiste fra l'invio di un SYN e la ricezione del segmento di SYN ACK.

Poiché tra l'inizio di un frame e l'inizio del frame successivo intercorre un intervallo di tempo pari all'RTT, il numero massimo di frame/s (N_F) trasferito risulta pari a:

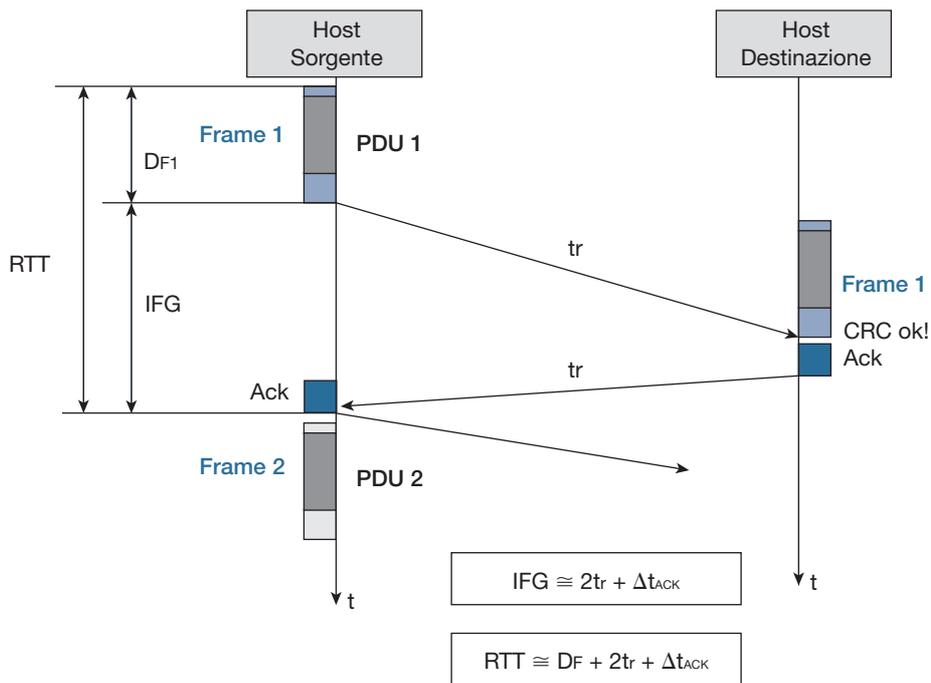
$$N_F = \frac{1}{RTT} [frame / s]$$

Il numero massimo di frame/s (N_F) che si possono trasmettere può infatti essere calcolato come rapporto tra il bit rate lordo e la dimensione dell'RTT espressa in termini di tempi di bit:

$$D_{RTT} = \frac{RTT}{t_{bit}} = RTT \cdot BR [bit] \Rightarrow N_F = \frac{BR}{D_{RTT}} = \frac{BR}{RTT \cdot BR} = \frac{1}{RTT} [frame / s]$$

Il massimo *throughput* ottenibile (BR_{Netto}) si può quindi calcolare come prodotto tra il numero di frame trasmessi e la dimensione (in bit) che ha un blocco di D_M Byte (i dati di applicazione veri e propri) incapsulati in un frame:

$$BR_{Netto} = (N_F) \cdot (D_M \cdot 8) [bit / s]$$



RTT: Round Trip Time
 IFG: InterFrame Gap
 tr: tempo di ritardo
 DF1: Durata di un frame con dati
 CRC: Cyclic Redundancy Check

FIGURA 1 Principio di funzionamento del metodo stop and wait

4.3.2. Metodo a finestra di trasmissione (sliding window)

Questo metodo viene utilizzato dal protocollo di trasporto TCP. In questo caso la sorgente può emettere in maniera continuativa un certo numero di PDU (*Protocol Data Unit*) tra una conferma di corretta ricezione (ACK) e l'altra. Il numero di PDU che si possono trasmettere in modo continuativo, senza attendere la conferma di corretta ricezione viene determinato dalla dimensione della finestra di trasmissione (*Windows Size*) espressa in Byte.

L'host destinazione comunica la dimensione della propria finestra di ricezione all'host sorgente, che in questo modo conosce quanti Byte (e quindi quante PDU) può trasmettere senza dover aspettare l'ACK di conferma di corretta ricezione.

La procedura di calcolo del bit rate netto (*throughput*) è quindi derivabile dal caso precedente (metodo *stop and wait*) tenendo conto che le dimensioni delle finestre di trasmissione e di ricezione determinano le prestazioni del sistema.

E' quindi necessario calcolare e configurare la dimensione della finestra di ricezione (*Receive Window size*, RXWIN) tenendo conto di un parametro noto come *prodotto banda-ritardo* o **BDP** (*Bandwith-Delay Product*), il quale è definito come il prodotto tra il bit rate lordo (BR) e l'RTT:

$$BDP = BR \cdot RTT \text{ bit}$$

Nel caso di più link aventi BR diverso va preso in considerazione quello con il bit rate minore.

Il *throughput* massimo si ottiene quando la dimensione della finestra di ricezione (D_{RXWIN}) ha una dimensione (in Byte) almeno pari a BDP:

$$D_{RXWIN} \geq \frac{BDP}{8} \text{ Byte}$$

Ottimizzando la dimensione della finestra di ricezione è possibile non avere interruzioni fra la ricezione di un ACK e l'altra, massimizzando così il *throughput*.

Nota la dimensione della finestra di ricezione (D_{RXWIN}), in assenza di errori e quindi di ritrasmissioni, è possibile stimare il throughput massimo (BR_{Netto}) consentito dal protocollo TCP come:

$$BR_{Netto} \approx \frac{D_{RXWIN} \cdot 8}{RTT} \text{ bit / s}$$

Infatti in un tempo pari all'RTT al massimo si possono trasferire senza interruzioni un numero di byte pari alla dimensione della finestra di ricezione (che il destinatario comunicata alla sorgente durante la fase di instaurazione della connessione logica o handshake a tre vie).

Ovviamente la velocità di trasmissione in linea (bit rate lordo) deve essere superiore al valore del bit rate netto per via del fatto che i protocolli impiegati aggiungono un loro header, per trasferire le loro informazioni di servizio e i loro controlli.

Questa relazione evidenzia come all'aumentare dei ritardi diminuisca il throughput, in quanto aumenta il tempo di attesa fra un ACK e l'altro. Le variazioni di ritardo, o jitter, influiscono così anche sul throughput.

Attività di laboratorio proposte.

1. Misura con l'analizzatore di protocollo Wireshark dell'RTT (*Round Trip Time*) a livello TCP.

La misura dell'RTT può essere effettuata operando nel seguente modo:

- si cattura con Wireshark il traffico che si ha quando ci si collega a un server web, applicando il **filtro di cattura** "host nomehost" (nell'esempio host labtele.iismajorana.it), Figura 2;
- si seleziona il frame (nell'esempio il frame n. 2) che trasporta il segmento TCP di *SYN, ACK*;
- si espande la voce *Frame 2* e si seleziona la voce *Time delta from previous displayed frame*;
- si prende nota del valore di tale parametro in quanto esso è pari all'RTT (*Round Trip Time*), poiché è la differenza di tempo che intercorre tra il frame precedente visualizzato, che trasporta un segmento TCP di *SYN* (inviato dal client per chiedere al server di aprire una connessione logica) e la risposta (*SYN, ACK*) con la quale il server accetta di aprire la connessione logica stessa; è possibile applicare tale voce come colonna cliccando su essa col tasto destro del mouse, Figura 3.



Figura 2 Applicazione di un filtro di cattura nella schermata iniziale di Wireshark

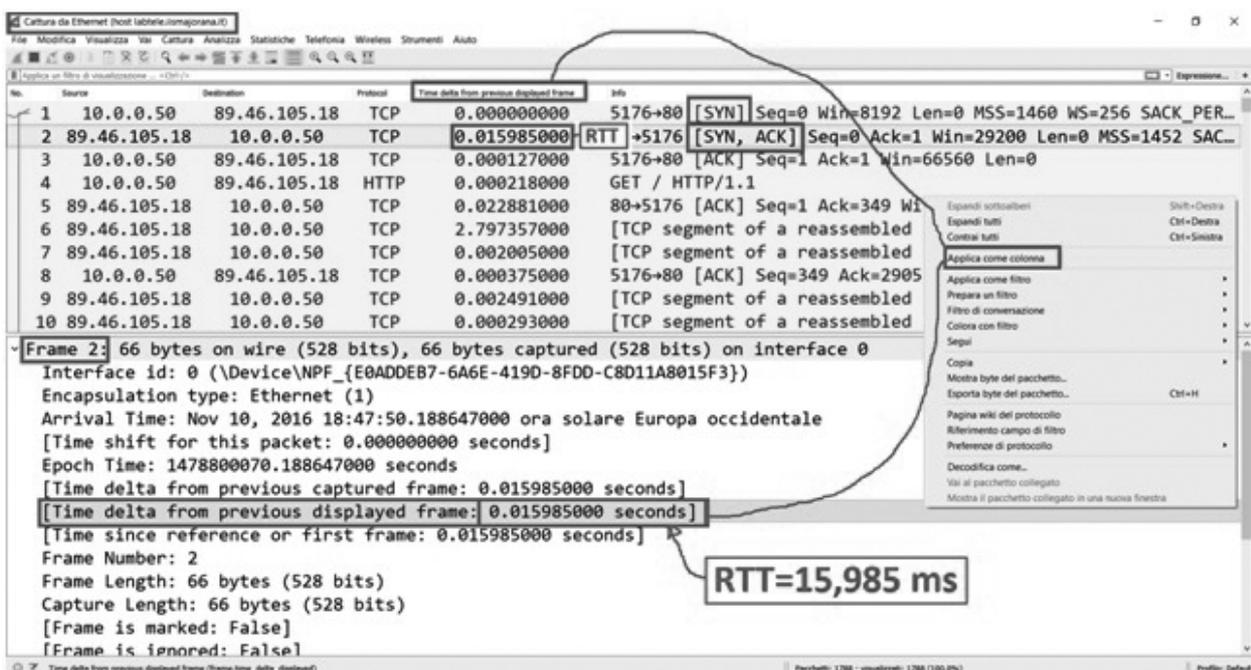
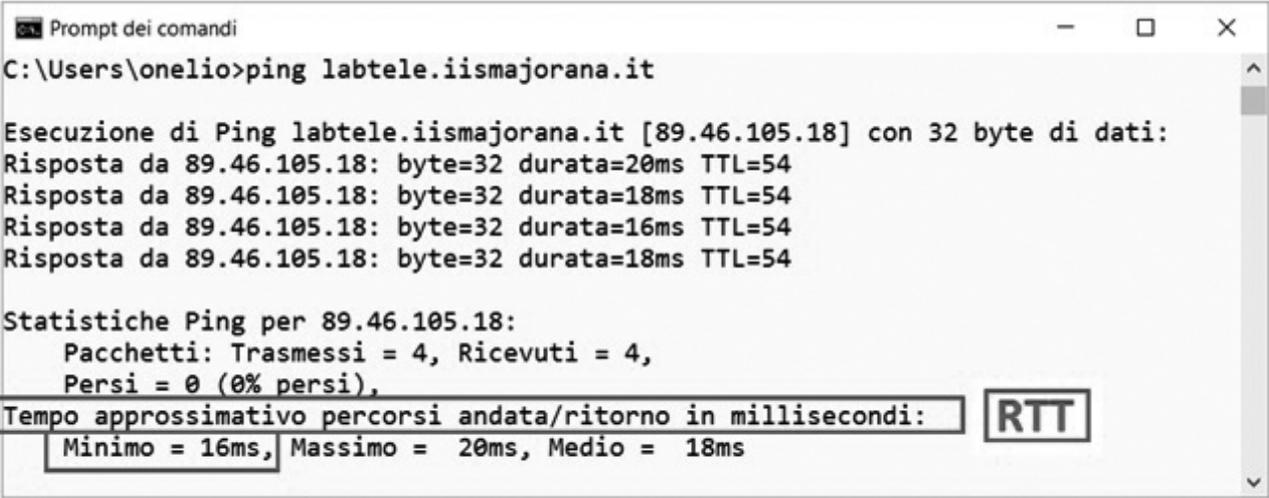


Figura 3 Misura dell'RTT a livello TCP fatta con Wireshark

E' poi possibile confrontare il valore di RTT misurato con Wireshark con quello ottenuto tramite il comando *ping nomehost* che fornisce l'RTT a livello 3 (protocollo ICMP), Figura 4.



```
Prompt dei comandi
C:\Users\onelio>ping labtele.iismajorana.it

Esecuzione di Ping labtele.iismajorana.it [89.46.105.18] con 32 byte di dati:
Risposta da 89.46.105.18: byte=32 durata=20ms TTL=54
Risposta da 89.46.105.18: byte=32 durata=18ms TTL=54
Risposta da 89.46.105.18: byte=32 durata=16ms TTL=54
Risposta da 89.46.105.18: byte=32 durata=18ms TTL=54

Statistiche Ping per 89.46.105.18:
  Pacchetti: Trasmessi = 4, Ricevuti = 4,
  Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
  Minimo = 16ms, Massimo = 20ms, Medio = 18ms
```

Figura 4 Determinazione dell'RTT a livello 3 (protocollo ICMP)

2. Determinazione con Wireshark del bit rate lordo (valutato tenendo conto anche degli header dei diversi protocolli) e del throughput a livello TCP.

Wireshark fornisce numerose statistiche sul traffico catturato, tra cui la determinazione del bit rate con cui avviene uno scambio dati tra due host e il throughput (capacità trasmissiva) a livello TCP.

Dopo aver catturato il traffico come indicato nel Laboratorio 1, per ottenere queste misure, dalla voce *Statistiche* nella barra dei menu di Wireshark si seleziona (Figura 5):

- a) *Grafico I/O*, per misurare il bit rate;
 - b) *Grafico dei flussi TCP* → *Capacità trasmissiva*, per misurare il throughput (indicato come *Capacità trasmissiva*).
- a) In *Grafico I/O*, Figura 6, si può dare un nome ai grafici e si possono impostare i seguenti *Filtri di visualizzazione*:
- *ip.dst== indirizzo_IP_PC* (per il download), in modo da tracciare il grafico con i soli pacchetti destinati al PC (il server è la sorgente, il PC è la destinazione)
 - *ip.src== indirizzo_IP_PC* (oppure *ip.dst== indirizzo_IP_server*) (per l'upload), in modo da tracciare il grafico con i soli pacchetti destinati al server (il PC è la sorgente, il server è la destinazione)

Si seleziona quindi *Bits* per l'Asse Y e *Intervallo: 1s*, in modo da avere il bit rate (bit/s) sull'asse Y. E' poi possibile tralare e ingrandire i grafici per evidenziare le parti di interesse (valori di picco, ecc.), ecc.

Cliccando su *Copia* è possibile copiare i valori dei punti che compongono i grafici e incollarli (in formato *csv*, *comma separated value*) su un foglio di calcolo per determinare, ad esempio, il valore massimo del bit rate (Figura 7).

- b) Nel grafico che compare (Figura 8) viene riportato il throughput misurato da Wireshark, sul lato destro, mentre sul lato sinistro è riportata la lunghezza dei segmenti TCP.

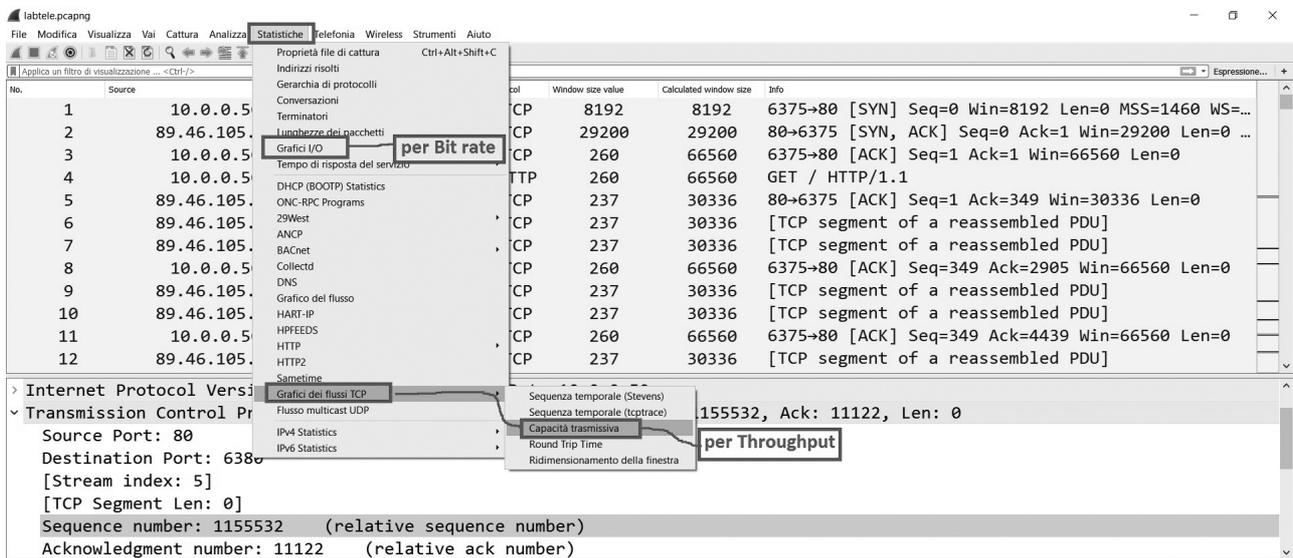


Figura 5 Menu Statistiche di Wireshark

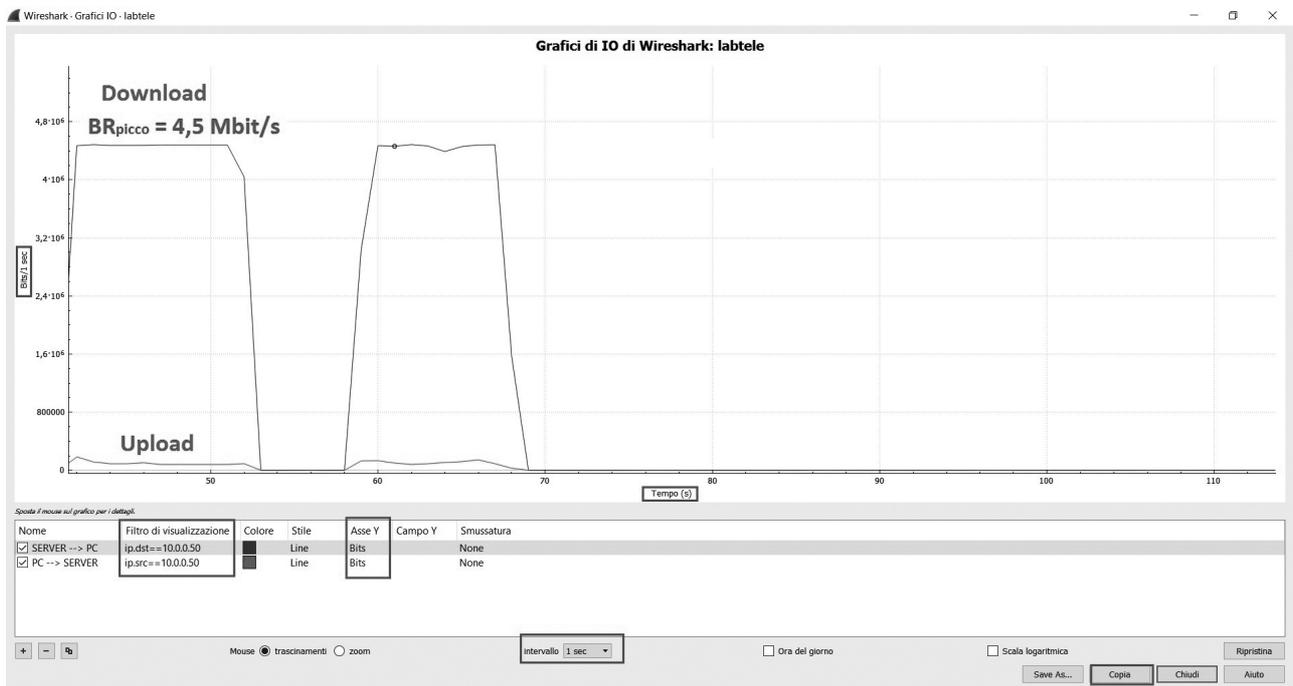


Figura 6 Grafici che forniscono il bit rate nelle direzioni Server → PC (Download) e PC → Server (Upload)

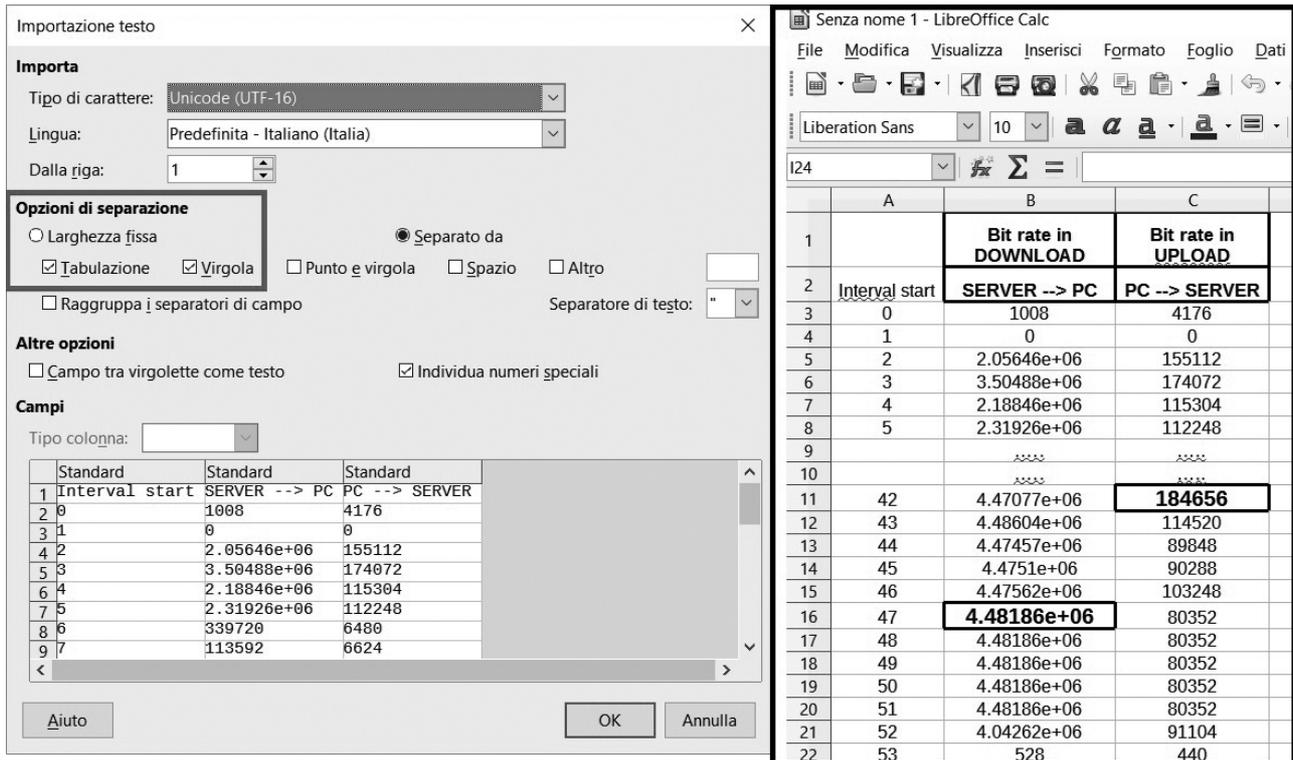


Figura 7 Valori del bit rate copiati su un foglio di calcolo.

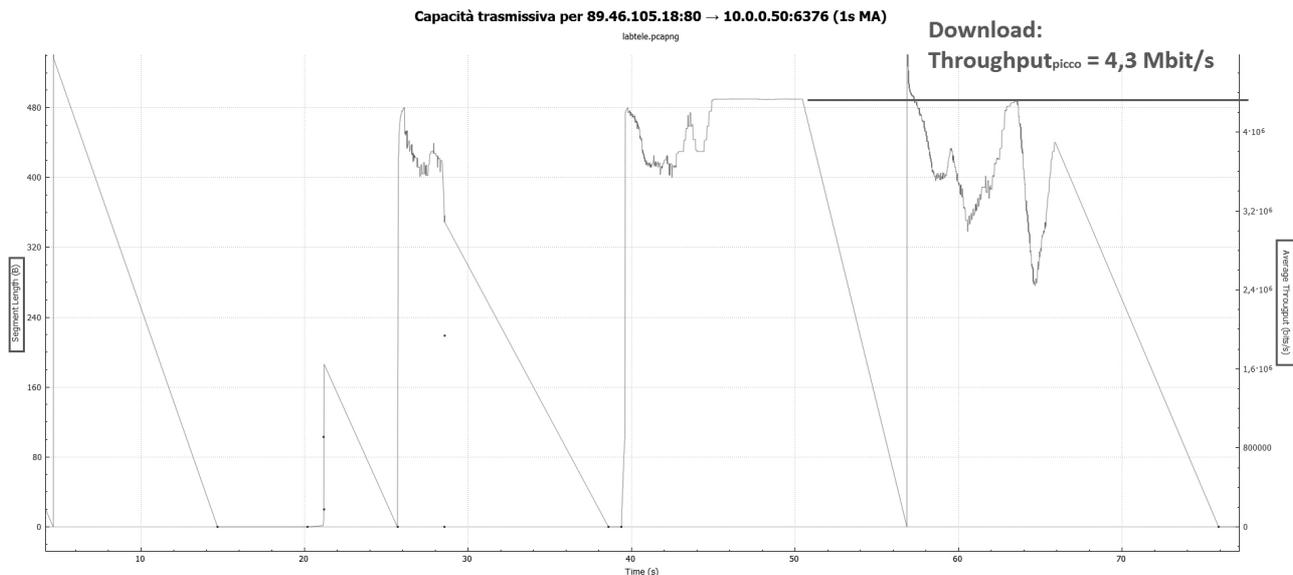


Figura 8 Grafico che presenta il throughput (capacità trasmissiva) a livello TCP nella direzione Server → PC (download)

3. Se è possibile accedere sia al PC sia al server per caricare dei programmi, è possibile determinare il throughput a livello TCP e UDP con i pacchetti software Jperf o Iperf3 (si veda il libro di testo, cap.3 Laboratorio didattico 1, per i dettagli).

ESERCIZI

Esercizio 1

Calcolare il bit rate netto massimo (throughput) per il trasferimento di dati consentito dal protocollo TCP nel caso in cui:

- si opera in una rete locale in cui si adotta la suite TCP/IP;
- si impiega la tecnologia Ethernet 100BASE-TX;
- il protocollo TCP incapsula nel payload delle proprie PDU (o segmenti TCP) blocchi di dati con dimensione pari a 1000 B;
- il controllo di flusso non intervenga in quanto si è all'interno della finestra di trasmissione e quindi si possano inviare senza interruzioni le PDU.

Calcolare quindi il tempo minimo necessario per il trasferimento di un file di 100 MB

Soluzione

Nel caso in esame la pila di protocolli che governa la trasmissione dei blocchi di dati è la seguente: TCP (Strato 4), IPv4 (Strato 3), Ethernet II (Strato 2), Stato fisico (Strato 1).

Per via del processo di incapsulamento la dimensione (D_F) di un frame Ethernet risulta pari alla somma della dimensione di un blocco di dati, degli header dei protocolli TCP, IPv4, dell'header e dell'FCS (Frame Check Sequence) aggiunti dal protocollo Ethernet II:

$$D_F = H_{Ethernet} + (H_{IPv4} + H_{TCP} + DATA)_{payload_Ethernet} + FCS = (8+12+2) + (20+20+1000) + 4 = 1066 \text{ B}$$

tenendo conto anche dell'IFG (Inter Frame Gap) minimo, di 12 B, che separa due frame Ethernet e del fatto che la tecnologia 100BASE-TX opera con una velocità di trasmissione pari a 100 Mbit/s, il numero massimo di frame/s con cui si può trasmettere è pari a:

$$N_F = \frac{BR}{(D_F + IFG) \cdot 8} = \frac{100 \cdot 10^6}{(1066 + 12) \cdot 8} = \frac{100 \cdot 10^6}{8624} = 11595 \text{ [frame / s]}$$

il bit rate netto (throughput) massimo si ottiene moltiplicando il numero di frame/s per il numero di bit informativi (dati) trasferiti con un frame:

$$BR_{Netto} = (N_F) \cdot (D_{blocco}) \cdot 8 = 11595 \cdot 1000 \cdot 8 = 92,8 \text{ Mbit / s}$$

Il tempo minimo di trasferimento di un file di 100 MB è quindi pari a:

$$T_T = \frac{D_{File} \cdot 8}{BR_{Netto}} = \frac{(100 \cdot 10^6) \cdot 8}{92,8 \cdot 10^6} \approx 8,6 \text{ s}$$

Esercizio 2

In una rete locale in cui si adotta la suite TCP/IP e la tecnologia Ethernet 1000BASE-T, cablata in categoria 6, si impiega un protocollo di applicazione per trasferire file con correzione d'errore per ritrasmissione e controllo di flusso con metodo *stop and wait*.

Sapendo che:

- il protocollo di applicazione opera formando delle 7-PDU (Protocol Data Unit dello strato 7 OSI) composte da un Header di 2 B e da un payload di 512 B;

- la conferma di corretta ricezione (ACK), invece, viene trasferita in una 7-PDU avente lunghezza pari a 4 B (2 B Header + 2 B ACK);
- il protocollo di applicazione utilizza il servizio di trasporto connectionless offerto dal protocollo UDP;
- si deve trasferire un file di 10 MB da un server a un client distanti 200 m;
- il fattore di velocità dei cavi impiegati è pari a 0,67; gli apparati di rete introducono un ritardo complessivo di 10 μs

calcolare il bit rate netto massimo (o *throughput* o velocità di informazione) con cui può essere trasferito il file e il tempo necessario al suo trasferimento in assenza di errori.

Indicare quindi come è possibile aumentare il throughput mantenendo inalterati i protocolli al di sotto dello strato di applicazione e la tecnologia Ethernet impiegata.

Soluzione

Nel caso in esame la pila di protocolli che governa la trasmissione del file è la seguente:

Protocollo di Applicazione, protocollo di trasporto UDP, protocollo dello strato 3 IPv4, protocollo dello strato 2 Ethernet II, Stato fisico.

Sapendo che l'Header del protocollo UDP è di 8 B, che il protocollo IPv4 ha un Header di 20 B, si calcola la dimensione di un frame Ethernet che trasporta una 7-PDU con dati:

$$D_F = H_{Ethernet} + (H_{IPv4} + H_{UDP} + PDU_{applic.})_{payload} + FCS = (8+12+2) + [20+8+(2+512)] + 4 = 568 B$$

La dimensione di un frame Ethernet che trasporta la 7-PDU di ACK ha un *payload* (campo informativo) avente la minima dimensione specificata dallo standard 802.3, pari a 46 B. Infatti la 7-PDU che trasporta le conferme di corretta ricezione (ACK) è di soli 4 B e aggiungendo gli header UDP e IPv4 si arriva alla formazione di un pacchetto IPv4 di 32 B; poiché la dimensione minima del payload di un frame Ethernet II è di 46 B, verranno aggiunti 14 B di riempimento (detti *padding* e costituiti da tutti 0) per arrivare a 46 B. La dimensione di un frame che trasporta gli ACK è quindi pari a:

$$D_{ACK} = (8+12+2) + (46) + 4 = 72 B$$

La tecnologia 1000BASE-T opera trasmettendo con un bit rate lordo (velocità di trasmissione) di 1000 Mbit/s, per cui un tempo di bit risulta pari a:

$$t_{bit} = \frac{1}{BR} = \frac{1}{10^9} = 1 \text{ ns}$$

Il tempo di trasmissione di un frame con dati e di un frame con l'ACK risultano così pari a:

$$\Delta t_F = \frac{D_F \cdot 8}{BR} = D_F \cdot 8 \cdot t_{bit} = 568 \cdot 8 \cdot 10^{-9} = 4,54 \mu s$$

$$\Delta t_{ACK} = D_{ACK} \cdot 8 \cdot t_{bit} = 72 \cdot 8 \cdot 10^{-9} = 0,57 \mu s$$

Si calcola il ritardo introdotto dal collegamento (comprensivo di quello degli apparati di rete):

$$t_r = \frac{L}{v_{propagaz.}} + tr_{apparati} = \frac{200}{0,67 \cdot 3 \cdot 10^8} + 10^{-5} = 11 \mu s$$

L'RTT (tempo di andata-ritorno) si può quindi stimare in:

$$RTT \approx \Delta t_{Frame} + 2t_r + \Delta t_{ACK} = 4,54 \cdot 10^{-6} + 2 \cdot 11 \cdot 10^{-6} + 0,57 \cdot 10^{-6} \approx 27 \mu s$$

Tra l'inizio di un frame e l'inizio del frame successivo intercorre un intervallo di tempo pari all'RTT per cui il numero di frame/s trasferito risulta pari a:

$$N_F = \frac{1}{RTT} = \frac{1}{27 \cdot 10^{-6}} = 37037 \text{ [frame / s]}$$

Poiché il numero di 7-PDU trasmesse coincide con il numero di frame trasmessi, sapendo che la dimensione di un blocco di dati trasferito come payload in una 7-PDU è di 512 B, possiamo quindi stimare il bit rate netto (BR_{Netto}), o throughput, come:

$$BR_{Netto} = (N_{Frame/s}) \cdot (D_{payload} \cdot 8) = (N_{7-PDU}) \cdot (D_{payload} \cdot 8) = 37037 \cdot 512 \cdot 8 = 152 \text{ Mbit / s}$$

Il valore trovato è quindi all'incirca 1/10 del bit rate lordo (1000 Mbit/s).

In assenza di errori il tempo necessario per trasferire il file da 10 MB risulta così pari a

$$\Delta t_{Trasf.} = \frac{D_M \cdot 8}{BR_{Netto}} = \frac{80 \cdot 10^6}{152 \cdot 10^6} \cong 520 \text{ ms}$$

Il throughput può essere aumentato, senza modificare i protocolli al di sotto dello strato di applicazione, se si aumenta la dimensione del blocco di dati trasferito come payload in una 7-PDU. Poiché la dimensione massima del payload di un frame Ethernet, corrispondente all'MTU (*Maximum Transmission Unit*), è pari a $MTU = 1500$ B, la dimensione massima che può avere un blocco di dati che costituisce il payload di una 7-PDU è pari a:

$$D_{Payload} = MTU - (H_{IPv4} + H_{UDP} + H_{7-PDU}) = 1500 - (20 + 8 + 2) = 1470 \text{ B}$$

Rifacendo i calcoli si può determinare così verificare che il throughput aumenta e raggiunge il suo valore massimo.

Dimensione di un frame Ethernet:

$$D_F = H_{Ethernet} + MTU_{payload} + FCS = (8 + 12 + 2) + 1500 + 4 = 1526 \text{ B}$$

Tempo di trasmissione di un frame con dati:

$$\Delta t_F = D_F \cdot 8 \cdot t_{bit} = 1526 \cdot 8 \cdot 10^{-9} = 12,2 \mu s$$

L'RTT si può quindi stimare in:

$$RTT \approx \Delta t_{Frame} + 2t_r + \Delta t_{ACK} = 12,2 \cdot 10^{-6} + 2 \cdot 11 \cdot 10^{-6} + 0,57 \cdot 10^{-6} \cong 34,77 \mu s$$

Il numero di frame/s trasferito risulta pari:

$$N_F = \frac{1}{RTT} = \frac{1}{34,77 \cdot 10^{-6}} = 28760 \text{ [frame / s]}$$

Poiché il numero di 7-PDU trasmesse coincide con il numero di frame trasmessi, sapendo che il payload delle 7-PDU è di 1470 B, possiamo quindi stimare il throughput (BR_{Netto}) come:

$$BR_{Netto} = (N_{7-PDU}) \cdot (D_{payload} \cdot 8) = 28760 \cdot 1470 \cdot 8 = 338 \text{ Mbit / s}$$

Operando con il bit rate massimo, il tempo necessario per trasferire un file di 10 MB diminuisce e diviene pari a

$$\Delta t_{Trasf.} = \frac{D_M \cdot 8}{BR_{Netto}} = \frac{80 \cdot 10^6}{338 \cdot 10^6} \cong 237 \text{ ms}$$

Esercizio 3

In rete locale 1000BASE-T un client e un server sono collegati (con cablaggio in rame) a uno stesso switch e sono posti alla massima distanza ammessa dagli standard; sapendo che il fattore di velocità del cavo è 0.7, per cui la velocità di propagazione è $2,1E+8$ m/s, e che lo switch determina un ritardo di $5 \mu s$ calcolare

- la latenza,
- l'RTT (tempo di andata-ritorno)
- il bit rate netto (throughput) massimo messo a disposizione dei protocolli di applicazione nel caso in cui:
 - si impiega un protocollo dello strato 4 che, pur operando in modalità connection oriented, viene configurato per operare la correzione d'errore con metodo *stop and wait*;
 - le 4-PDU (Protocol Data Unit del protocollo dello strato 4) sono caratterizzate da un header di 20 B;
 - la conferma di corretta ricezione (ACK) è inviata con una 4-PDU con dimensione totale pari a 20 B;
 - una 3-PDU (Protocol Data Unit del protocollo dello strato 3) ha un header di 20 B e incapsula nel suo payload una 4-PDU;
- indicare quale modifica si può apportare al protocollo di trasporto per aumentare il bit rate netto senza modificare la configurazione dei protocolli degli altri strati.

Soluzione

a) Calcolo della latenza

La distanza massima ammessa dagli standard relativi al cablaggio strutturato, a cui è necessario fare riferimento, nel caso di cablaggio in rame (U/UTP ecc.) è di 100 m, per cui la distanza totale del collegamento client-server è pari a 200 m, con interposto a metà collegamento lo switch.

La latenza nella direzione client-> server si può quindi calcolare come:

$$LT = \Delta t_{Frame} + \sum \Delta t_{Prop.} + \Delta t_{Switch} \quad s$$

Ipotizzando di operare in assenza di errori, con il payload del frame Ethernet pari all'MTU (Maximum Transmission Unit, pari a 1500 B) si ha che un frame è composto da 1526 B (non considerando l'IFG) e quindi ha una dimensione (espressa in bit) pari a:

$$D_{F_ [bit]} = D_{F_ [Byte]} \cdot 8 = 12208 \quad bit$$

Con la tecnologia Ethernet 1000BASE-T, che opera a 1 Gbit/s, la durata temporale di un frame è pari a:

$$\Delta t_{Frame} = \frac{D_{F_ [bit]}}{BR_{[bit/s]}} = \frac{12208}{1 \cdot 10^9} = 12,2 \quad [\mu s]$$

Il ritardo di propagazione su 100 m introdotto da un cavo risulta pari a:

$$\Delta t_{prop.} = \frac{L_{[m]}}{v_{prop. [m/s]}} = \frac{100}{2,1 \cdot 10^8} = 0,48 \quad [\mu s]$$

La latenza risulta così pari a:

$$LT = 12,2_{[\mu s]} + [0,48_{[\mu s]} + 0,48_{[\mu s]}] + 5 \mu s = 18,16 \quad \mu s$$

b) Calcolo RTT (Round Trip Time, tempo di andata-ritorno)

Ipotizziamo che i tempi di propagazione nelle due direzioni siano uguali.

Nel caso di trasferimento delle conferme di corretta ricezione si forma una 3-PDU (pacchetto) avente una dimensione pari a

$$D_{3-PDU} = H_{3-PDU} + D_{4-PDU} = 20 + 20 = 40 \text{ B}$$

Una 3-PDU che trasporta la conferma di corretta ricezione (4-PDU di ACK) viene incapsulata in un frame avente il payload di dimensione minima, pari a 46 B, per cui alla 3-PDU sono aggiunti 6 B di riempimento (*padding*).

L'RTT può essere definito come l'intervallo di tempo che intercorre l'istante di tempo in cui si emette il primo bit di una PDU che trasporta dati e quello in cui si riceve l'ultimo bit di una PDU di conferma di corretta ricezione (ACK).

Poiché un frame che trasporta l'ACK è composto da $8+6+6+2+46+4=72$ B e quindi da 576 bit, la durata del corrispondente frame è pari a:

$$\Delta t_{ACK} = \frac{D_{F-ACK} [bit]}{BR [bit/s]} = \frac{576}{1 \cdot 10^9} = 0,57 \text{ } [\mu s]$$

L'RTT si può quindi stimare come:

$$RTT = LT_{Client \rightarrow Server} + LT_{Server \rightarrow Client} = [18,16 \mu s] + [0,57 \mu s + 2 \cdot 0,48 \mu s + 5 \mu s] = 24,7 \text{ } [\mu s]$$

c) Bit rate netto (throughput) massimo con metodo Stop and Wait.

Il throughput (BR_{Netto}) si può calcolare come:

$$BR_{Netto} = (N_F) \cdot (D_M \cdot 8) [bit/s]$$

dove

- N_F è il numero di frame/s trasmesso, calcolabile come

$$N_F = \frac{1}{RTT} = 40,5 \cdot 10^3 [frame/s]$$

in quanto con il metodo *stop and wait* la trasmissione del frame successivo può avvenire solo dopo la ricezione la conferma di corretta ricezione (ACK);

- D_M è la dimensione massima che può avere il payload (campo informativo) di una 4-PDU (detta anche MSS, Maximum Segment Size); poiché il campo informativo (payload) di un frame Ethernet ha dimensione massima pari a 1500 B, corrispondente all'MTU (Maximum Transmission Unit), si ottiene:

$$D_M = MTU - H_{4-PDU} - H_{3-PDU} = 1500 - 20 - 20 = 1460 \text{ Byte}$$

Il bit rate netto (throughput) risulta così pari a:

$$BR_{Netto} = (N_F) \cdot (D_M \cdot 8) = (40,5 \cdot 10^3) \cdot (1460 \cdot 8) = 473 [Mbit/s]$$

- d) Il bit rate netto può essere aumentato modificando il metodo con cui il protocollo di trasporto effettua la correzione d'errore, sostituendo il metodo *stop and wait* con quello a *finestra di trasmissione (sliding windows)*. Infatti con il metodo a finestra di trasmissione (impiegato dal protocollo di trasporto TCP) la sorgente può emettere un certo numero di 4-PDU consecutivamente, senza attendere la conferma di corretta ricezione ad ogni 4-PDU inviata; un ACK, poi, conferma la corretta ricezione delle 4-PDU inviate all'interno di una finestra di trasmissione.

Esercizio 4

Si desidera valutare il bit rate netto (throughput) massimo offerto dal protocollo di trasporto TCP nel trasferire un file di 1 MB da un server a un client ed il tempo necessario al suo scaricamento.

La connessione fisica client-server avviene tramite un accesso ADSL che in downstream ha una banda di 4 Mbit/s e che costituisce il bit rate lordo minore di tutto il collegamento.

Si chiede di:

- stimare il prodotto banda-ritardo (*Bandwith Delay Product*, BDP) quando il tempo di andata-ritorno è pari a $RTT = 10$ ms;
- determinare la dimensione minima che dovrebbe avere la finestra di ricezione;
- calcolare il throughput che si ha quando l'RTT diviene pari a 40 ms, per una situazione di congestione di rete, mantenendo inalterata la finestra di ricezione e in assenza di errori.

Soluzione

- a) Si calcola il BDP (prodotto banda-ritardo) come:

$$BDP = BR \cdot RTT = 4 \cdot 10^6 \cdot 0,01 = 40 \text{ kbit}$$

- b) La finestra di ricezione dovrebbe avere una dimensione almeno pari a $D_{RXWIN} = \frac{BDP}{8} = 5 \text{ kByte}$

- c) Se l'RTT diviene pari a 40 ms, mantenendo inalterata la dimensione della finestra di ricezione, si ha che il throughput massimo risulta all'incirca pari a:

$$BR_{Netto} \approx \frac{D_{RXWIN} \cdot 8}{RTT} = \frac{40 \cdot 10^3}{40 \cdot 10^{-3}} = 1 \text{ Mbit / s}$$

Esercizio 5

In un collegamento fra un client e un server si ha un tempo di andata-ritorno $RTT = 20$ ms. Dovendo trasferire informazioni utilizzando come protocollo di trasporto il TCP, sapendo che sono state effettuate le seguenti configurazioni:

- un segmento TCP è composto da un Header di 20 B e da un payload di 1000 B
- la finestra di ricezione utilizzata ha dimensione pari a $D_{RXWIN}=66792$ B
- il protocollo dello strato 3, IPv4, opera con pacchetti aventi un Header di 20 B
- si impiega la tecnologia Ethernet per gli strati OSI 1 e 2

Si chiede di calcolare il throughput (bit rate netto) massimo ottenibile a livello TCP e il bit rate lordo minimo a livello fisico che deve essere disponibile.

Soluzione

Il throughput massimo a livello TCP si può stimare come:

$$BR_{Netto} \approx \frac{D_{RXWIN} \cdot 8}{RTT} = \frac{66792 \cdot 8}{20 \cdot 10^{-3}} \approx 26 \text{ Mbit / s}$$

In un secondo si possono quindi trasferire un numero blocchi di dati, che costituiscono il *payload* dei segmenti TCP, pari a:

$$N_{blocchi} = \frac{BR_{Netto}}{D_{Blocco} \cdot 8} = \frac{26 \cdot 10^6}{8 \cdot 10^3} = 3250 \text{ blocchi / s}$$

Si devono perciò trasmettere 3250 segmenti TCP al secondo e quindi $N_F=3250$ frame/s.

Supponendo che i dati da trasferire siano passati direttamente al protocollo TCP, la pila di protocolli che si ha è la seguente: TCP, IPv4, Ethernet II, Strato fisico.

Il processo di incapsulamento e formazione dei frame Ethernet determina così l'aggiunta degli header TCP, IPv4, Ethernet nonché dell'FCS (Frame Check Sequence) e dell'IFG (Inter Frame Gap) dei frame, per un numero di Byte e di bit (overhead) in ciascun frame pari a:

$$H_{tot} = 20_{H_TCP} + 20_{H_IPv4} + 22_{H_Ethernet} + 4_{FCS} + 12_{IFG} = 78[B] \Rightarrow H_{tot} = 78 \cdot 8 = 624 \text{ [bit]}$$

Ciascun frame ha quindi una dimensione espressa in bit e in Byte pari a:

$$D_{Frame} = 78_{H_tot} \cdot 8 + 1000_{Dati} = 1078 \text{ Byte}; D_{Frame} = 1078 \cdot 8 = 8624 \text{ bit}$$

La minima velocità di trasmissione (bit rate lordo) richiesta a livello fisico per supportare il throughput massimo a livello TCP è quindi pari a:

$$BR = 3250_{frame/s} \cdot 8624_{bit/frame} = 28 \text{ Mb/s}$$

Impiegando almeno la tecnologia Ethernet 100BASE-TX, operante a 100 Mbit/s, è quindi possibile soddisfare i requisiti a livello fisico. E' anche possibile aumentare il throughput aumentando la dimensione di ciascun blocco di dati che costituisce il payload dei segmenti TCP, che può arrivare a 1460 B (corrispondente all'*MSS - Maximum Segment Size*).

Si vedano inoltre le soluzioni della Simulazione di seconda prova di Telecomunicazioni proposta dal MIUR e della seconda prova di Telecomunicazioni relative all'Esame di Stato 2015, scaricabili dal sito

<http://online.scuola.zanichelli.it/provatecnici/telecomunicazioni/>
