

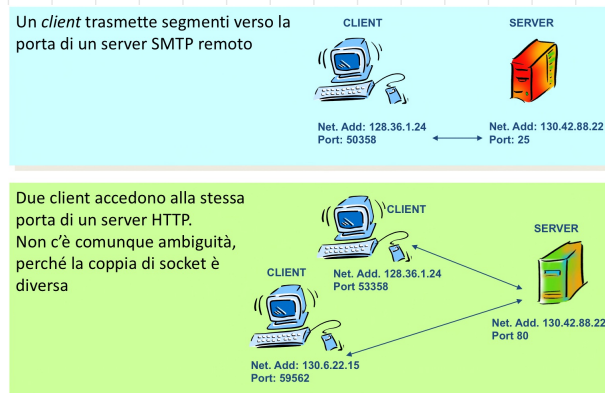
4. TRASPORTO

Il livello di trasporto ha il compito di instaurare un collegamento logico tra le applicazioni residenti su host remoti. Il trasporto rende trasparente il trasporto dei messaggi dalle applicazioni. Il livello di trasporto è presente solo negli host e consente di creare un collegamento logico tra processi applicativi. Il livello di trasporto esegue multiplexing quindi implementa indirizzamento tramite le porte.

Le porte sono salvate nelle PDU e sono valori a 16 bit. Il range è suddiviso in:

- numeri ridi [0; 1023]: numeri riservati per servizi di ampio utilizzo. Utilizzati principalmente dai server per ascolto;
- numeri registrati [1024; 49151]: numeri registrati da terzi per loro servizi (tipicamente proprietari). Utilizzati dai server per ascolto;
- numeri dinamici [49152; 65535]: numeri usati lato client per la comunicazione;

L'estensione fornita dal livello di trasporto è il socket: coppia IP-porta che identifica univocamente un applicativo.



I protocolli di trasporto sono implementati a livello S.O. A ogni socket, il S.O. crea una coda di entrata e una di uscita. Questa è la cosiddetta funzionalità di buffering.

Il livello di rete non è un livello affidabile. Il livello di trasporto, quindi, deve riuscire a garantire alle applicazioni affidabilità. Esistono due tipi di trasporto:

- TCP (TRANSPORT CONTROL PROTOCOL): orientato alla connessione + AFFIDABILE, - SPEED
- UDP (USER DATAGRAM PROTOCOL): senza connessione - SPEED, + AFFIDABILE

4.1 UDP

Le uniche cose che aggiunge al livello rete sono:

- multiplexing, demultiplexing
- controllo (sua versione) nell'header

Non esecuta nessun controllo di flusso e di errore e non garantisce la consegna.

La PDU UDP ha formato header (8 byte), payload



Il checksum viene calcolato da trasm. e ricevuto. Il ricevitore poi esegue lo stesso calcolo e, se i checksum coincidono, la PDU è valida. L'algoritmo di calcolo del checksum è molto semplice:

lato trasmettitore

- L'insieme di bit è diviso in blocchi da 16 bit
- Il campo Checksum è inizializzato a 0
- Tutti i blocchi vengono sommati in aritmetica complemento a uno
- Il risultato è complementato ed inserito nel campo di checksum del segmento inviato

lato ricevitore

- L'insieme di bit è diviso ancora in blocchi da 16 bit
- Tutti i blocchi vengono sommati in aritmetica complemento a uno
- Il risultato è complementato
 - Se sono tutti 0 il pacchetto è accettato
 - Altrimenti è scartato

Nel calcolo del checksum vengono contati lo pseudohader IP, l'header UDP e i dati.

4.2 TRASPORTO AFFIDABILE

4.2.1 PROTOCOLLI DI RITRASMISSIONE

Per operare, un protocollo di ritrasmissione necessita:

- messaggio di ACK/NACK per segnalare la correttezza (o no) ricezione
- canale di servizio per trasmettere suddetti messaggi. (anche ACK/NACK possono subire errori)

COMUNICAZIONE
BIDIREZIONALE

Per trasmettere un pacchetto, quindi, lo si manda e si aspetta l'ACK. Se arriva NACK o si va in timeout, il pacchetto viene rispedito. Si va avanti così finché il pacchetto non è stato trasmesso completamente.

Il controllo dell'errore viene eseguito ai livelli 2 e 4:

- al 2 si controlla che i bit trasmessi arrivino correttamente (oggi non si controlla più a livello 2)
- al 4 si controlla sia l'integrità che la ricezione, l'ordine di arrivo e la non duplicazione.

4.2.1.1 STOP-AND-WAIT

Il primo protocollo di ritrasmissione è lo Stop and Wait. È il più semplice in quanto necessita:

- ACK
- timer di timeout
- numerazione di pacchetti (SN) e ACK (RN)

Il funzionamento prevede la ritrasmissione se l'ACK non arriva entro il timeout. Il tempo di trasmissione dell'ACK è pari a: $2\tau + T_{ACK}$. Il protocollo funziona solo se $T_{OUT} \geq 2\tau + T_{ACK}$.

La presenza della numerazione permette di passare alle applicazioni i pacchetti nell'ordine giusto e senza duplicazioni. L'efficienza del protocollo è:

$$\eta = \frac{T}{T + 2\tau + T_{ACK}} < 1$$

Si può notare che il ritardo di trasmissione influisce molto sull'efficienza. Inoltre non è adatto a sistemi con un alto ritmo di trasmissione. Lo Stop-and-wait si presta molto bene alla comunicazione half-duplex (comunicazione bidirezionale).

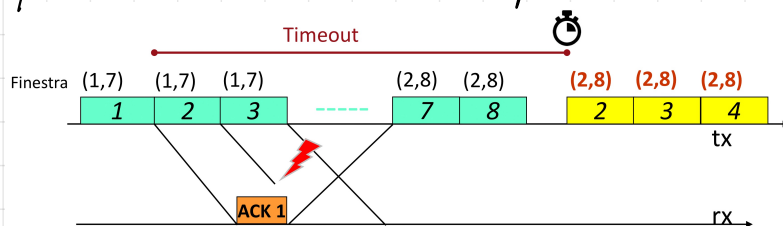
Usi dello Stop-and-wait: WIFI.

4.2.1.2 GO-BACK-N

Il protocollo GO-BACK-N è una generalizzazione dello stop-and-wait. Esso prevede la trasmissione fino a N pacchetti senza ricevere l'ACK. Questi N pacchetti sono detti finestra di trasmissione. Man mano che gli ACK vengono ricevuti, la finestra viene spostata in avanti (sliding window).

Se non ci sono errori il GO-BACK-N riesce ad offrire un'efficienza pari a $\eta = 1$ (100%).

Nel caso siano riscontrati errori, la finestra viene spostata indietro fino al primo pacchetto con errore e si ritrasmette. L'errore viene definito da un timeout come nello stop-and-wait.



I pacchetti fuori ordine vengono scartati. Lo stop-and-wait, quindi, è un GO-BACK-N con $N=1$.

L'ACK può anche essere collettivo. Se per qualche motivo gli ACK prima di ACK N sono stati persi, l'ACK N può essere interpretato come correttezza ricezione di tutti i pacchetti fino a N, rimediando alla perdita di ACK (sempre supponendo che il timeout non sia già scaduto).