

Layer di trasporto

Il livello di trasporto supporta la comunicazione tra processi di applicazioni che vengono eseguiti su due **macchine diverse**. E' proprio questo strato che è responsabile della comunicazione tra due hosts.

I protocolli di trasporto sono forniti solo dagli hosts, e **non dai routers!**

- **Send-Side:** converte i messaggi degli strati applicazioni in **segmenti al layer di trasporto**, aggiungendo un **header di livello di trasporto**, e invia i messaggi allo strato network.
- **Receive-Side:** assembla i segmenti in messaggi e li passa al layer applicazione.

Abbiamo due protocolli di trasporto

TCP - Transmission Control Protocol

- Affidabile, inoltre invia gli elementi con lo stesso ordine in cui sono stati spediti
- Controllo delle congestioni
- Controllo del flow
- Setup della connessione

UDP - User Datagram Protocol

- Non è affidabile, inoltre non invia gli elementi nell'ordine di spedizione

Reliable data transfer - Trasferimento dati affidabile

Il servizio fornito dall'rdt permette a due processi su diversi host di comunicare l'uno con l'altro, come se fossero connessi. Ad ogni modo, il servizio viene implementato al di sopra di un canale non affidabile; questo costringe il server e client ad inviarsi dei messaggi di controllo, per confermare l'affidabilità.

In questo tipo di connessione non abbiamo né perdita di dati, né corruzione di essi; inoltre viene preservato l'ordine di invio dei dati. Nel protocollo rdt capiamo l'importanza del **layer di trasporto**, e del **layer di applicazione** che è responsabile di inviare i dati dal processo inviante al processo ricevente. Quando il layer di applicazione invia i dati, passiamo al layer di trasporto, ed in questo caso viene usato un protocollo di trasferimento dati affidabile.

ACK

In ambito di telecomunicazioni informatico ACK è il simbolo che identifica un segnale di **acknowledge** emesso in risposta alla ricezione di un'informazione completa. Un esempio tipico è il pacchetto di controllo previsto dal **protocollo TCP** trasmesso dal ricevente al mittente per segnalare la corretta ricezione di un pacchetto di dati.

Un ACK può anche essere di tipo **comulativo**, ovvero un messaggio di conferma che riguarda più pacchetti di dati trasmessi correttamente.

Ad esempio, **ACK4** indica che il pacchetto 4 è stato ricevuto correttamente.

NACK

NACK sta per Not ACKnowledge, ed indica la mancata ricezione di un pacchetto, o la corretta ricezione di **n-1** pacchetti, ma la mancata ricezione di uno di essi; ad esempio **NACK4** ci dice che il pacchetto 4 non è arrivato, ma il 3,2,1 e 0 sì.

Checksum

Il tipo più semplice di checksum consiste nel sommare in trasmissione tutti i bit del messaggio e di memorizzare il valore risultante nel frame inviato. Per controllare l'integrità del messaggio sarà sufficiente effettuare in ricezione la stessa operazione di somma e confrontarla con il checksum memorizzato nel frame: se i due valori coincidono i dati possono essere considerati integri.

Questo tipo di checksum non è particolarmente efficace proprio perchè non consente di rilevare errori come il riordinamento dei bit (quindi non più FIFO) o l'inserimento di bit a zero (che non vengono quindi inclusi nella somma); infine diversi errori potrebbero avere come somma zero, e quindi non verrebbero rilevati.