

Rete: livello trasporto (TCP/IP) Parte 2 - Indice

1. Modello di servizio **TCP**
2. Il protocollo **TCP**
3. Il segmento **TCP**
4. Politica di trasmissione **TCP**
5. Intervallo di *time out*

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 376

Rete: livello trasporto (TCP/IP) Modello di servizio TCP - 1

- 2 tipi di servizio a livello trasporto
 - **TCP** (*Transmission Control Protocol*)
 - Garantisce agli utenti (*end points*) comunicazione affidabile anche in presenza di una rete inaffidabile
 - *Connection-oriented*
 - **UDP** (*User Datagram Protocol*)
 - Fornisce agli utenti la medesima qualità di comunicazione offerta dal livello **IP** sottostante
 - *Connectionless*
 - Più velocità di trasporto che affidabilità

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 377

Rete: livello trasporto (TCP/IP) Modello di servizio TCP - 2

- Il servizio **TCP** richiede la creazione di due **socket** ai 2 estremi della connessione
 - La coppia di **socket** identifica la connessione, la quale può convogliare più conversazioni tra quei punti terminali
 - Un singolo **socket** può essere punto terminale di più connessioni
 - Ogni **socket** ha un indirizzo unico, determinato dall'indirizzo **IP** del nodo ospite (32 *bit*) e dal numero di porta locale (16 *bit*)

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 378

Rete: livello trasporto (TCP/IP) Modello di servizio TCP - 3

- I numeri di porta nell'intervallo 0..255 sono preassegnati ad alcuni servizi (protocolli) standard, alcuni dei quali usano **TCP** altri **UDP**
 - Alcuni servizi standard che usano **TCP**:
 - 25 **SMTP** *Simple Mail Transfer Protocol*
 - 80 **HTTP** *HyperText Transfer Protocol*
 - 110 **POP3** *Post Office Protocol* versione 3

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 379

Rete: livello trasporto (TCP/IP) Modello di servizio TCP - 4

- La connessione **TCP** è bidirezionale (*full-duplex*) e punto a punto
 - Nessun supporto per comunicazioni a diffusione (*broadcast*) o per gruppo (*multicast*)
- La connessione **TCP** è a flusso di dati (*byte stream*)
 - Le unità di trasporto non corrispondono alle unità dati riconosciute dall'utente applicativo
 - Inviati x B \rightarrow Trasportati $x/2 + x/2 \rightarrow$ Ricevuti x B
- Vi è memorizzazione (e dunque potenziale ritardo) sia tra ricezione da mittente ed emissione su rete che tra ricezione da rete ed emissione verso destinatario
 - Le prime implementazioni di **TCP** prevedevano un'opzione per effettuare consegna immediata (**PUSH**)

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 380

Rete: livello trasporto (TCP/IP) Il protocollo TCP - 1

- L'unità di trasporto **TCP** è detta segmento, composto da un prefisso obbligatorio e da un campo dati utente di ampiezza variabile
- L'ampiezza massima di segmento è dimensionata dall'esigenza di:
 - Contenere l'intero segmento entro il campo dati di 1 **datagram** = $(64k - 1) - 20$ B
 - Limite determinato dal livello **IP**
 - Tener conto della massima dimensione delle unità di trasporto su rete per evitare costose frammentazioni
 - Limite determinato dalle caratteristiche (di rami) della rete

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 381

Rete: livello trasporto (TCP/IP) Il protocollo TCP - 2

- Un segmento troppo largo per le capacità di un tratto della rete viene frammentato in più **datagram**, ciascuno con proprio prefisso e con indicatore di frammento
 - Maggior costo trasmissivo
 - Onere di ricostituzione del segmento, con maggiore complessità nella gestione delle conferme
 - I frammenti vengono confermati nel corretto ordine di sequenza e non nel loro ordine di arrivo
 - Arrivo: 1 | 3 | 0 | 4 | 2 → Conferma: - | - | 1 | - | 4

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 382

Rete: livello trasporto (TCP/IP) Il segmento TCP - 1

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 383

Rete: livello trasporto (TCP/IP) Il segmento TCP - 2

- Così come il **datagram**, anche il prefisso di segmento indica la propria lunghezza (in parole di 32 bit), data la presenza di un campo opzionale
- Il successivo campo da 6 bit era previsto per sviluppi futuri, ma non è tuttora utilizzato
- Seguono 6 *flag* da 1 bit ciascuno
 - URG : se 1, il campo Urgent Pointer indica, in B, la posizione del segmento corrente alla quale si trovano dati "urgenti"
 - ACK : se 1, il campo Indice di Conferma è valido, altrimenti ignorato

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 384

Rete: livello trasporto (TCP/IP) Il segmento TCP - 3

- Seguono 6 *flag* da 1 bit ciascuno (segue)
 - PSH : se 1, richiede all'entità di trasporto a destinazione di consegnare il segmento al destinatario senza ritardo
 - RST : se 1, richiede il *reset* della connessione
 - SYN : usato per stabilire una connessione
 - SYN=1 ACK=0 : M → D : richiesta di connessione
 - SYN=1 ACK=1 : D → M : richiesta accettata
 - FIN : se 1, notifica la richiesta di fine connessione
 - Rilascio simmetrico: un processo può continuare a ricevere dati ben oltre la propria richiesta di fine connessione

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 385

Rete: livello trasporto (TCP/IP) Il segmento TCP - 4

- Nel campo window size l'entità di trasporto a destinazione indica l'ampiezza della propria finestra di ricezione per l'attuale connessione
 - Il valore 0 richiede ad M di sospendere ogni nuovo invio, confermando però ricezione fino al B di indice (Acknowledgment number - 1)
 - M potrà riprendere l'invio solo dopo l'emissione da parte di D di un segmento con lo stesso Indice di Conferma precedente, ma indicante ampiezza di finestra > 0
 - Dimensione massima di finestra sufficiente per contenere un segmento di massima ampiezza (< 64 kB)

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 386

Rete: livello trasporto (TCP/IP) Il segmento TCP - 5

- Il campo Checksum consente a D di effettuare un ulteriore controllo di integrità di segmento
 - Complemento ad 1 della somma in complemento ad 1 di tutte le parole a 16 bit del segmento
 - Includendo nel segmento un ulteriore prefisso virtuale comprensivo degli indirizzi IP di M e D, di 1 B a 0, di 1 B indice di protocollo (TCP=6) e di 2 B indicanti l'ampiezza in B dell'intero segmento (assente nel prefisso reale)
 - Violazione di astrazione! TCP conosce gli indirizzi IP
 - Ampiezza massima di segmento < (2¹⁶ - 1 - 20) B
 - Rendendo pari l'ampiezza del campo dati con l'aggiunta di 1 B di 0 (se necessario)
 - Considerando 0 il campo Checksum

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 387

Rete: livello trasporto (TCP/IP)
Controllo di congestione - 1

- In presenza di reti fortemente inaffidabili è difficile determinare la causa della perdita di segmento segnalata da un *time out* di ricezione
 - Congestione di rete → ritardo sulla linea, con o senza distruzione di pacchetto
 - Errori trasmissivi → distruzione di pacchetto
- Non conviene fare assunzioni se non si conosce la topologia del tratto di rete da attraversare
 - Le connessioni che usano linee fisiche hanno buona affidabilità → il *time out* è sintomo di congestione
 - Le connessioni *wireless* sono invece molto inaffidabili

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 394

Rete: livello trasporto (TCP/IP)
Controllo di congestione - 2

- Il controllo di flusso lega l'emissione di segmenti da parte di **M** alla capacità di ricezione di **D**, assumendo una rete con sufficiente capacità trasmissiva
- Il controllo di congestione aiuta ad alleviare la perdita di pacchetti in presenza di reti inaffidabili
 - Può essere necessario indipendentemente dalla capacità di ricezione di **D**

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 395

Rete: livello trasporto (TCP/IP)
Controllo di congestione - 3

- **M** mantiene 2 finestre concettuali d'invio
 - RW indicante la disponibilità manifestata da **D**
 - CW indicante la dimensione di segmento che si stima eviti congestione della rete (congestion window)
- **M** invia segmenti dimensionati al minimo tra le dimensioni delle due finestre
 - **M** usa un'euristica per approssimare l'ampiezza di CW

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 396

Rete: livello trasporto (TCP/IP)
Controllo di congestione - 4

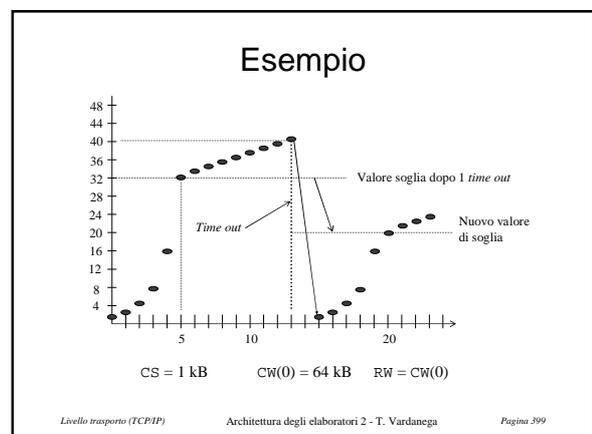
- **M** inizializza CW alla massima dimensione di segmento permesso sulla connessione (CS)
- **M** invia un segmento di massima dimensione e verifica se la conferma di ricezione arrivi prima della scadenza del *time out*
- In caso positivo, **M** raddoppia l'ampiezza di CW e ripete l'operazione fino a quando non si verifichi un *time out* o CW ecceda la dimensione di RW
- → Algoritmo detto *slow start*, adottato da tutte le implementazioni **TCP**

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 397

Rete: livello trasporto (TCP/IP)
Controllo di congestione - 5

- L'algoritmo di controllo di congestione usa un parametro soglia, inizialmente fissato alla massima dimensione di segmento (64 kB)
- Ad ogni *time out*, la soglia viene ridotta alla metà del valore corrente di CW mentre CW viene ridotto ad 1 CS
- Si usa poi l'algoritmo *slow start* per ampliare CW, raddoppiando fino al limite di soglia e poi crescendo di 1 CS alla volta

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 398



Rete: livello trasporto (TCP/IP) Intervallo di time out - 1

- **TCP** usa un'euristica per determinare la durata dell'intervallo di *time out* adatta alle prestazioni effettive della rete, che sottopone a monitoraggio continuo
 - La variabile di controllo RTT (*round-trip time*) denota la miglior stima corrente dell'intervallo che intercorre tra un invio e la sua ricezione
 - Un orologio misura l'intervallo T ad ogni invio
 - Il valore di RTT viene così aggiornato:
 - $RTT = \alpha RTT + (1 - \alpha) T$ dove α ha valore prevalente (p.es. 7/8)

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 400

Rete: livello trasporto (TCP/IP) Intervallo di time out - 2

- L'idea era quella di fissare l'intervallo di *time out* come βRTT , p.es. $\beta = 2$, ma si è poi osservato che non conviene usare un fattore β costante
- Attualmente l'intervallo di *time out* viene calcolato come $RTT + 4D$
 - dove D viene costantemente aggiornata al valore $D = \alpha D + (1 - \alpha) |RTT - T|$

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 401

Rete: livello trasporto (TCP/IP) Intervallo di time out - 3

- In presenza di reinvio a seguito di *time out* è difficile dire con esattezza se una successiva conferma di ricezione concerne il 1° invio o la sua ritrasmissione
 - Una attribuzione erronea di T corrompe RTT
- L'algoritmo di Karn prevede di non utilizzare le misurazioni relative a segmenti ritrasmessi per aggiornare RTT

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 402

Rete: livello trasporto (TCP/IP) Altre misurazioni

- Oltre ad RTT, l'entità di trasporto **TCP** usa altri 2 valori temporali
 - Un orologio misura il tempo trascorso da quando D abbia notificato disponibilità 0 senza più aggiornarla (limite di persistenza)
 - Trascorso il limite di persistenza, M chiede a D di aggiornare la sua disponibilità di ricezione
 - Un altro orologio misura il tempo trascorso dall'ultimo invio o ricezione sulla connessione (limite di attività vitale)
 - Trascorso tale limite, il lato dove questo è avvenuto invia un segmento di controllo per verificare la presenza della controparte
 - In assenza di risposta, lo stesso lato rilascia la connessione

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 403

Rete: livello trasporto (TCP/IP) Modello di servizio UDP

- **UDP** fornisce all'utente semplicemente l'accesso diretto al servizio datagram di **IP**
- Ne segue che la sua entità di trasporto non effettua alcun controllo di correttezza di trasmissione
- L'unità di trasporto **UDP** è dunque un normale datagram prefissato da:
 - 2 parole da 16 bit indicanti le porte di M e D
 - 1 parola da 16 bit indicante l'ampiezza del datagram
 - 1 parola da 16 bit con checksum calcolato con la stessa tecnica usata per il segmento **TCP**

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 404