

lo strato di trasporto

servizi di comunicazione tra end system per le applicazioni

giuseppe di battista

140-trasporto-05 copyright ©2021 g. di battista

1

nota di copyright

- questo insieme di slides è protetto dalle leggi sul copyright
- il titolo ed il copyright relativi alle slides (inclusi, ma non limitatamente, immagini, foto, animazioni, video, audio, musica e testo) sono di proprietà degli autori indicati sulla prima pagina
- le slides possono essere riprodotte ed utilizzate liberamente, non a fini di lucro, da università e scuole pubbliche e da istituti pubblici di ricerca
- ogni altro uso o riproduzione è vietata, se non esplicitamente autorizzata per iscritto, a priori, da parte degli autori
- l'informazione contenuta in queste slides è fornita per scopi didattici e non può essere usata in progetti di reti, impianti, prodotti, ecc.
- gli autori non si assumono nessuna responsabilità per il contenuto delle slides, che sono comunque soggette a cambiamento
- questa nota di copyright non deve essere mai rimossa e deve essere riportata anche in casi di uso parziale

140-trasporto-05 copyright ©2021 g. di battista

2

servizio offerto

- il servizio offerto dallo strato di trasporto allo strato superiore deve essere affidabile ed è normalmente *connesso*
- i processi che usano le primitive offerte dallo strato di trasporto assumono che esse siano *affidabili*
- le primitive di servizio dello strato di trasporto sono offerte ad una popolazione di utenti-programmatori molto *ampia*
 - primitive facili da usare

140-trasporto-05 copyright ©2021 g. di battista

3

primitive di trasporto

- esempi di primitive di trasporto
 - *listen* in attesa di qualche connessione
 - *connect* tentativo di instaurare una connessione
 - *send* invio di dati
 - *receive* in attesa di dati
 - *disconnect* rilascio della connessione

140-trasporto-05 copyright ©2021 g. di battista

4

instaurazione e rilascio di connessioni

- le primitive per l'instaurazione e per il rilascio di connessioni devono essere realizzate in modo affidabile
- il problema è più facile da risolvere per l'instaurazione che per il rilascio

140-trasporto-05 copyright ©2021 g. di battista

5

instaurazione di connessioni

- i pacchetti scambiati nell'ambito di una connessione sono numerati
- l'instaurazione è basata sulla scelta e sul relativo riscontro dei numeri iniziali di sequenza utilizzati per la numerazione dei pacchetti
 - metodo three-way handshake

140-trasporto-05 copyright ©2021 g. di battista

6

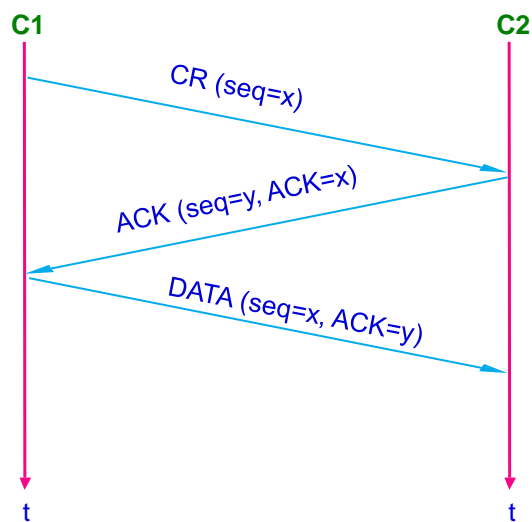
instaurazione di connessioni

- metodo three-way handshake (calcolatori C1 e C2)
 - C1 sceglie il numero di sequenza iniziale x per i propri pacchetti
 - C1 invia una connection request (CR) con x a C2
 - C2 riceve la connection request con x
 - C2 sceglie il proprio numero di sequenza iniziale y
 - C2 invia una connection accepted riscontrando x e proponendo il proprio numero iniziale y a C1
 - C1 riceve la connection accepted da C2
 - C1 riscontra y a C2

140-trasporto-05 copyright ©2021 g. di battista

7

instaurazione di connessioni



140-trasporto-05 copyright ©2021 g. di battista

8

perché non numerare i pacchetti da zero?

- usare numeri diversi per ogni connessione consente di distinguere pacchetti che, pur scambiati dalla stessa coppia di interlocutori, appartengono a connessioni diverse

140-trasporto-05 copyright ©2021 g. di battista

9

rilascio di connessioni

- attenzione: è più difficile di quanto si possa immaginare
- se il rilascio è troppo rudimentale si possono perdere dei dati
- esempio:
 - supponiamo di rilasciare la connessione come si termina una telefonata
 - ad un certo punto l'interlocutore C1 rilascia unilateralmente la connessione
 - se C2 aveva nel frattempo inviato dei dati c'è il rischio che essi arrivino a C1 dopo che C1 ha "attaccato il telefono"

140-trasporto-05 copyright ©2021 g. di battista

10

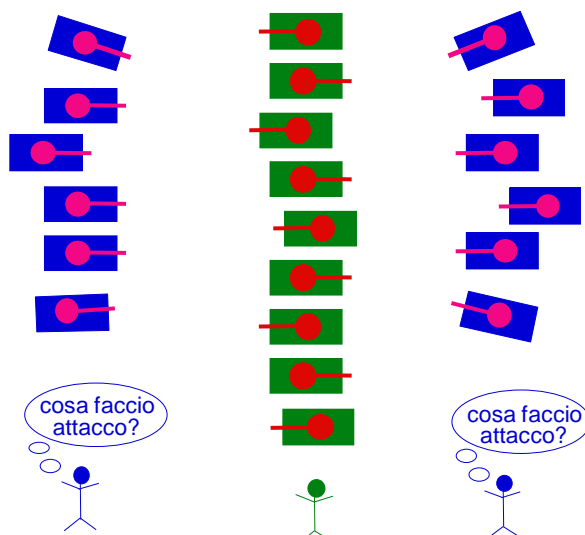
rilascio di connessioni

- una possibile soluzione: il rilascio simmetrico
 - ogni direzione è rilasciata in modo indipendente dall'altra
 - quando un interlocutore ha rilasciato continua comunque a ricevere dati eventualmente provenienti dall'altro lato
- in che istante la connessione può dirsi definitivamente rilasciata?
 - *problema dei due eserciti*

140-trasporto-05 copyright ©2021 g. di battista

11

problema dei due eserciti



140-trasporto-05 copyright ©2021 g. di battista

12

problema dei due eserciti

- i blu in totale sono più dei verdi ma i verdi sono più di ognuna delle due parti blu
- se i blu attaccano contemporaneamente vincono ma se attaccano separatamente perdono
- i blu vogliono sincronizzarsi ma per farlo devono mandare emissari attraverso il campo dei verdi, i quali potrebbero catturarli
- esiste un protocollo che faccia vincere i blu con certezza?

140-trasporto-05 copyright ©2021 g. di battista

13

problema dei due eserciti

- supponiamo che:
 - il comandante blu 1 mandi il messaggio “attacchiamo oggi alle 11:30, che ne pensi?”
 - il messaggio arrivi e che il comandante blu 2 mandi il messaggio “ok”
 - il messaggio del comandante blu 2 arrivi al comandante blu 1
- l'attacco avverrà?
- probabilmente *no* visto che il comandante blu 2 non sa se il suo "ok" è arrivato a destinazione (se l' "ok" non è arrivato il comandante blu 1 non attaccherà e quindi l'attacco del comandante blu 2 sarà sconfitto)

140-trasporto-05 copyright ©2021 g. di battista

14

problema dei due eserciti

- complichiamo il protocollo:
 - supponiamo che il comandante blu 1 debba riscontrare l' "ok"
 - supponiamo che il comandante blu 1 riscontri
- a questo punto è lui a non sapere se attaccare, visto che non sa se il suo riscontro è arrivato
- in realtà non esiste un protocollo che garantisca il successo
- il problema di rilasciare o no una connessione è assimilabile a quello di decidere se attaccare
- in pratica ci si accontenta di tecniche analoghe al three-way handshake *con timeout*

140-trasporto-05 copyright ©2021 g. di battista

15

TCP

140-trasporto-05 copyright ©2021 g. di battista

16

Transmission Control Protocol – TCP

- rfc 793, 1122 e 1323
- servizio di trasmissione dati bidirezionale contemporaneo (full duplex) punto-punto (host-to-host, end-to-end)
 - multicasting e broadcasting non supportati

140-trasporto-05 copyright ©2021 g. di battista

17

Transmission Control Protocol – TCP

- TCP è il primo tra i protocolli studiati fino ad ora nella Internet Protocol Suite a garantire l'affidabilità del servizio di trasmissione
 - a livello 2 o a livello 3, a causa del traffico eccessivo o di errori di vario genere, i pacchetti possono andare perduti, essere duplicati o essere consegnati nell'ordine sbagliato
 - TCP identifica questi problemi, richiede la ritrasmissione dei dati perduti e riordina i pacchetti fuori ordine
 - uso di riscontri (acknowledgement - ack)

140-trasporto-05 copyright ©2021 g. di battista

18

Transmission Control Protocol – TCP

- il servizio offerto è connesso e la connessione avviene fra due processi, identificati dall'**indirizzo IP** e da un numero di **port** da entrambe le parti
- una volta che la connessione è stabilita, il processo utente manda i dati allo strato TCP che li invia *correttamente* allo strato TCP del ricevente, che li passa a sua volta al processo destinazione
- l'implementazione della connessione fa in modo che i processi utente vedano un canale di comunicazione privato, permettendo una programmazione a livello utente in cui *la rete è trasparente*

140-trasporto-05 copyright ©2021 g. di battista

19

port

- TCP specifica come distinguere più destinazioni (processi) su una stessa macchina
 - ai processi che interagiscono con altri processi usando TCP viene attribuito un numero di port
- i port sono i TCP sap
- un numero di port ha 2 byte
- i port < 256 sono riservati per servizi standard (ad es. ftp = 20 e 21, telnet = 23, smtp = 25, http = 80)
 - si veda, ad es., la rfc 1700

140-trasporto-05 copyright ©2021 g. di battista

20

pacchetti e byte

- TCP non numera i pacchetti ma i byte; ogni byte spedito ha un numero di sequenza a 32 bit
 - 32 bit consentono di avere reset dei numeri di sequenza abbastanza infrequenti
- i numeri di sequenza sono usati anche per gli ack
- la t-pdu TCP si chiama segment
- segment = header + dati (opzionali)
- limiti del segment:
 - limite di 65535 byte per i dati IP
 - MTU (Maximum Transfer Unit); da cosa dipende?

140-trasporto-05 copyright ©2021 g. di battista

21

segment

- il pacchetto TCP è chiamato segment, i segment vengono usati per:
 - stabilire connessioni
 - trasferire dati
 - mandare acknowledgement
 - chiudere connessioni

140-trasporto-05 copyright ©2021 g. di battista

22

segment

- formato del segment:

source port			destination port		
sequence number					
acknowledgement number					
hlen	res.	code	window		
checksum			urgent pointer		
options				padding	
data					
.....					

140-trasporto-05 copyright ©2021 g. di battista

23

segment – sequence number e ack number

- il sequence number stabilisce la posizione del pacchetto dati nel flusso di informazioni del mittente, questi infatti trasferisce un flusso di dati generando un certo numero di pacchetti da inviare attraverso la rete
- l'acknowledgement (ack) number viene utilizzato per riscontrare i byte ricevuti correttamente (significato: prossimo byte atteso)
- il sequence number si riferisce al flusso che va nella stessa direzione del segment mentre l' ack number si riferisce al flusso che va in direzione opposta al segment

140-trasporto-05 copyright ©2021 g. di battista

24

segment – hlen e code

- hlen: numero di parole di 32 bit dell'intestazione
- code (6 bit) determina il tipo di messaggio contenuto nel segmento

bit

URG

ACK

PSH

RST

SYN

FIN

significato

urgent pointer field is valid

ack field is valid

this segment requests a push

reset the connection

synchronize sequence numbers

sender has reached end of its byte stream (rilascio)

140-trasporto-05 copyright ©2021 g. di battista

25

segment – urgent pointer e window

- urgent pointer consente di identificare eventuali dati urgenti nel segment
 - la validità del pointer dipende dal bit URG
- window specifica la ampiezza corrente della *finestra di controllo di flusso*
 - stabilisce quanti byte si possono ricevere dopo l'ultimo riscontrato
 - limita i byte che possono essere inviati dall'interlocutore
 - può assumere il valore 0

140-trasporto-05 copyright ©2021 g. di battista

26

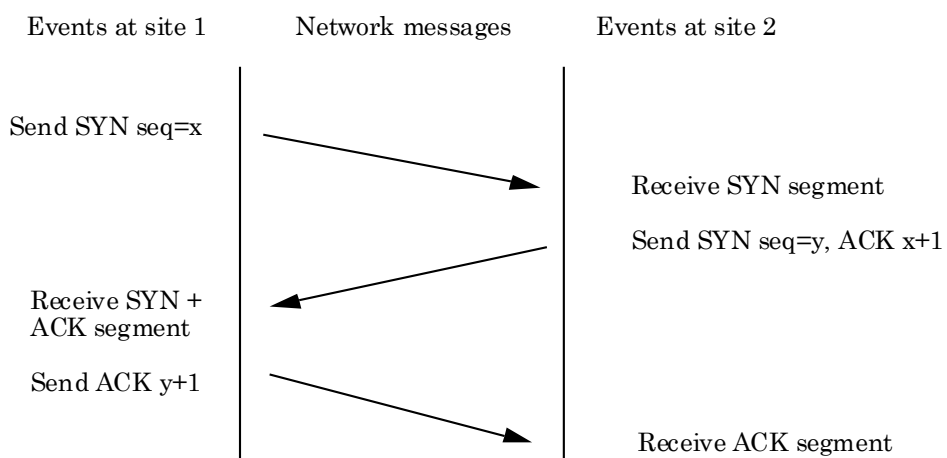
segment – checksum e opzioni

- checksum interessa l'intero segment + gli indirizzi IP + protocol
 - violazione della gerarchia dei protocolli
- opzioni: la più utilizzata specifica la massima ampiezza del campo dati
 - negoziazione durante l'instaurazione della connessione
 - gli host sono obbligati ad accettare almeno 536 byte di dati

140-trasporto-05 copyright ©2021 g. di battista

27

instaurazione di una connessione



140-trasporto-05 copyright ©2021 g. di battista

28

rilascio di una connessione

- per rilasciare la connessione una delle due parti manda un segment con FIN=1; non ha più dati da trasmettere
- quando la parte che ha mandato FIN=1 riceve in riscontro un ACK considera chiuso il suo flusso
- i dati possono continuare a fluire in direzione opposta
- la connessione è rilasciata quando entrambi i flussi sono chiusi
- il rilascio normale richiede 4 segment
- il primo ACK ed il secondo FIN possono essere nello stesso segment
- uso di timeout per il problema dei due eserciti; timeout = $2 * \text{tempo di vita stimato di un pacchetto}$

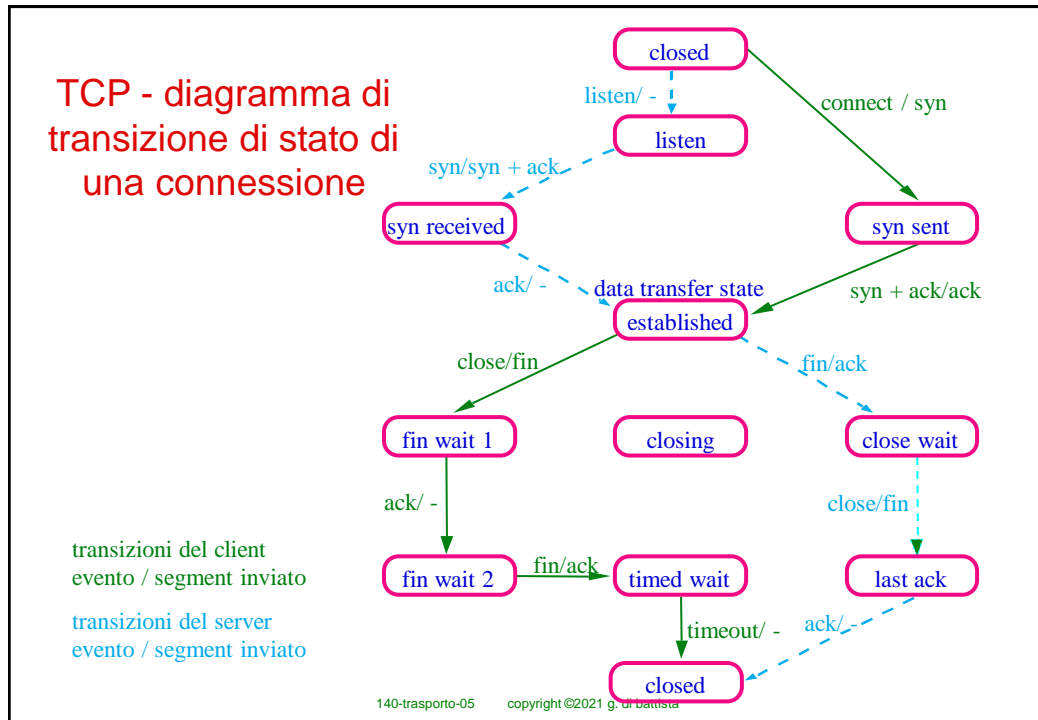
140-trasporto-05 copyright ©2021 g. di battista

29

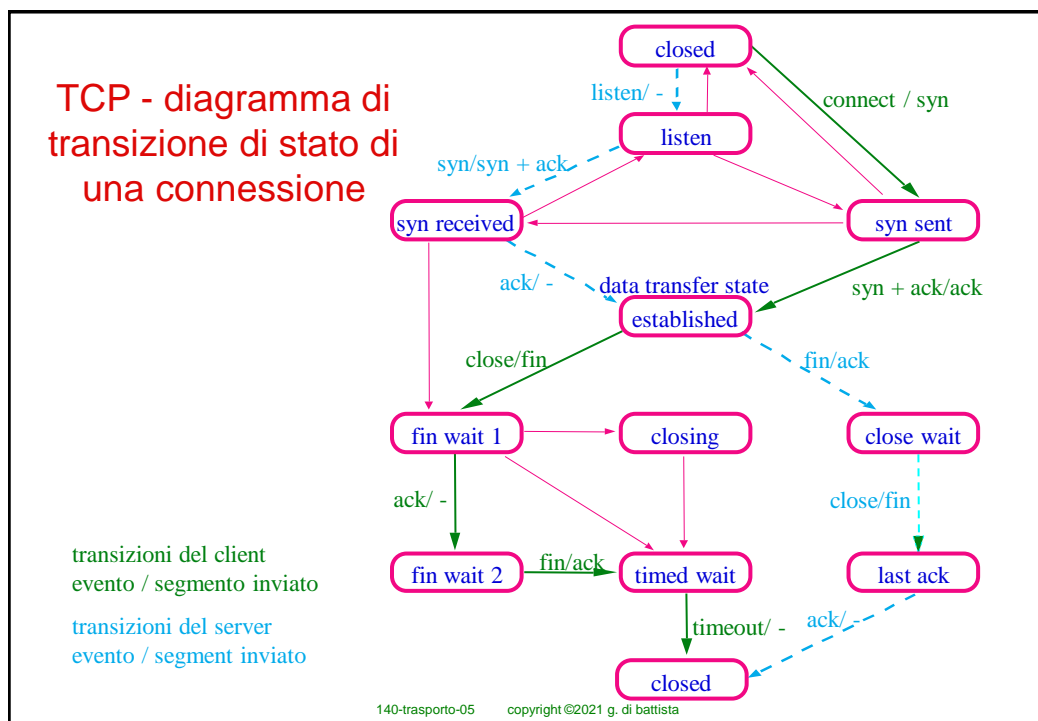
TCP - diagramma di transizione di stato di una connessione



30



31



32

stato established

- lo stato established corrisponde a molti stati più specifici, sia in ricezione
 - riscontro dei pacchetti ricevuti correttamente
 - gestione della finestra di controllo di flusso
- che in spedizione
 - scelte nella spedizione dei pacchetti, per sfruttare al massimo la banda e per contribuire al controllo delle congestioni nella rete
 - spedizione di copie dei pacchetti di cui non si è ottenuto un ack

140-trasporto-05 copyright ©2021 g. di battista

33

UDP

140-trasporto-05 copyright ©2021 g. di battista

34

User Datagram Protocol – UDP

- UDP fornisce un servizio di trasmissione non affidabile, il mittente cioè non ha informazioni sull'esito effettivo della ricezione
- viene utilizzato da servizi nei quali:
 - non è importante che un pacchetto sia riscontrato (si preferisce l'efficienza rispetto all'affidabilità)
 - oppure si preferisce gestire il meccanismo dei riscontri a livello applicativo

140-trasporto-05 copyright ©2021 g. di battista

35

User Datagram Protocol – UDP

- l'header è diviso in 4 campi di 16 bit che specificano il port da cui viene inviato il messaggio, il port destinazione, la lunghezza del datagram ed il checksum
- i port UDP sono analoghi ai port TCP

140-trasporto-05 copyright ©2021 g. di battista

36