

Corso di
RETI DI CALCOLATORI
(9 CFU)

a.a. 2020-2021
II anno / II semestre

Reti di calcolatori Prof. Gianni Fenu

1

Il Livello Trasporto III - 3

Il Livello Trasporto

Servizi forniti ai livelli superiori (entità di trasporto)

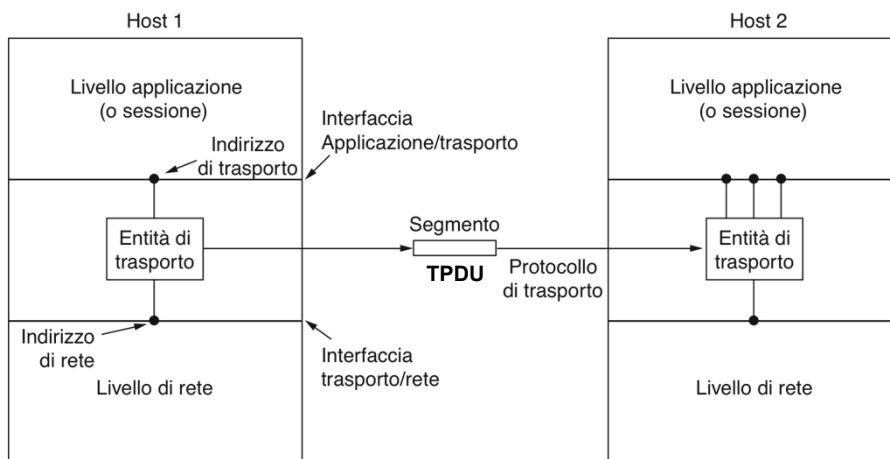
- Servizio di trasporto orientato alla connessione
- Servizio di trasporto senza connessione

Connessione END-TO-END

Reti di calcolatori 02 Prof. Gianni Fenu

2

Servizi offerti ai livelli superiori



3

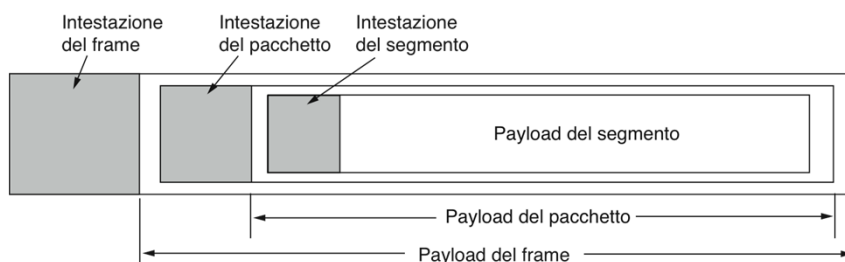
Primitive del servizio di trasporto

- Permettono agli utenti (ad esempio programmi applicativi) di accedere al servizio.
- Simili ai servizi di rete.
- Servizio in genere affidabile.

Primitiva	Pacchetto inviato	Significato
LISTEN	(nessuno)	Si blocca fino a quando un processo cerca di connettersi
CONNECT	CONNECTION REQ.	Tenta di stabilire una connessione
SEND	DATA	Invia informazioni
RECEIVE	(nessuno)	Si blocca fino all'arrivo di un pacchetto DATA
DISCONNECT	DISCONNECTION REQ.	Questo lato desidera rilasciare la connessione

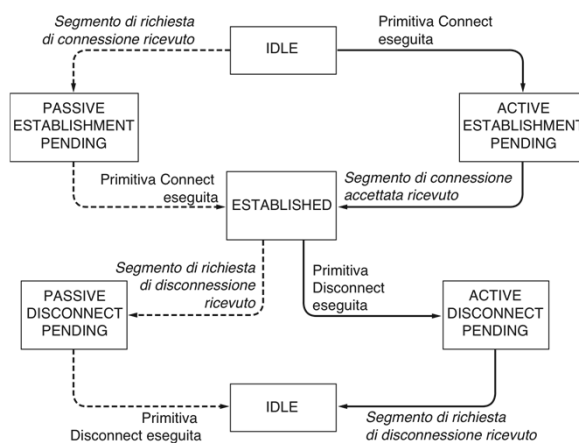
4

- TPDU (Transport Protocol Data Unit).
- Caratterizza i messaggi spediti tra entità di trasporto.
- Sono contenuti in pacchetti a livello rete, e in trame a livello data link



5

Macchina a stati finiti della gestione delle connessioni



6

Socket per TCP (primitive di trasporto)

	Primitiva	Significato
C S	SOCKET	Crea un nuovo punto finale di comunicazione
S	BIND	Associa un indirizzo locale ad un socket
S	LISTEN	Annuncia la volontà di accettare connessioni; da dimensione alla coda
S	ACCEPT	Arresta il visitatore finché non arriva la connessione
C	CONNECT	Tenta attivamente di stabilire la connessione
C S	SEND	Spedisce dati sulla connessione
C S	RECEIVE	Riceve dati dalla connessione
C S	CLOSE	Chiude la connessione

7

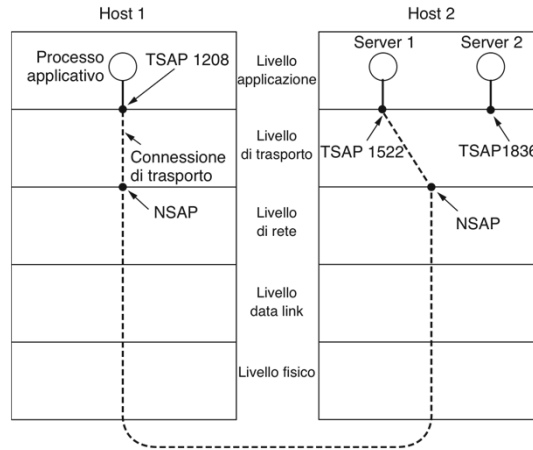
Elementi del protocollo di trasporto
Indirizzamento

- Quando un'applicazione crea una connessione, deve specificare con chi vuole connettersi.
- Il metodo usualmente utilizzato definisce gli indirizzi di trasporto utilizzabili dai processi.
- In Internet questi punti finali sono denominati porte locali (port).
- Vengono chiamati genericamente **TSAP (Transport Service Access Point)**, mentre i punti di accesso a livello rete vengono chiamati **NSAP**

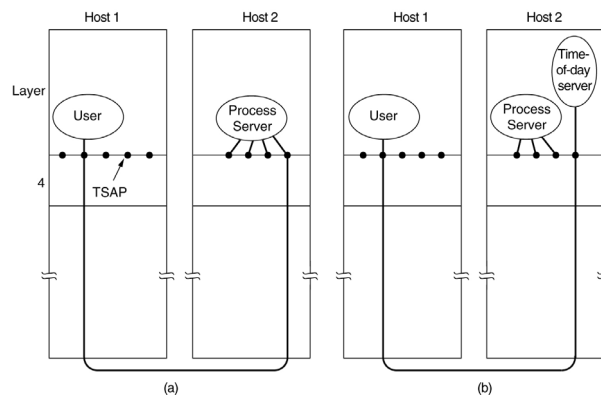
8

TSAP, NSAP e Connessioni

TSAP 1208 (origine richiesta)
TSAP 1522 (dest. richiesta)



Protocollo di connessione iniziale (Process server)



Elenco dei servizi e server disponibili (Directory server)

Attivazione di una connessione

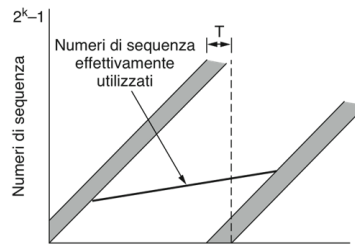
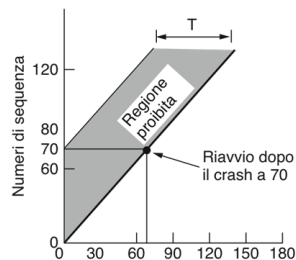
Gestione del tempo di vita di un pacchetto.

1. Progettazione ristretta delle reti.
2. Inserimento di un contatore di hop in ogni pacchetto.
3. Inserimento di un orario in ogni pacchetto.

E' necessario garantire che un pacchetto sia defunto, e, inoltre, che siano defunti tutti i suoi ack

Attivazione di una connessione

- Non è sufficiente spedire una TPDU CONNECTION REQUEST, ed attendere una risposta CONNECTION ACCEPTED.
- Risolvere problemi di perdita, memorizzazione o duplicazione dei pacchetti.



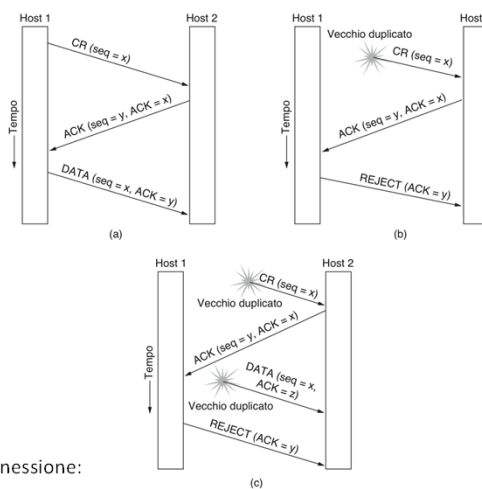
Risolvere il problema dei duplicati in una connessione.

three-way handshaking (Tomlinson, 1975)

- il richiedente invia un TPDU di tipo conn.request con un numero x proposto come inizio della sequenza;
- il destinatario invia un TPDU di tipo ack contenente:
 - la conferma di x;
 - e la proposta di un proprio numero y di inizio sequenza;
- il richiedente invia un TPDU di tipo dati contenente:
 - i primi dati del dialogo;
 - la conferma di y.

I valori x e y possono essere generati sfruttando l'orologio di sistema, in modo da avere valori ogni volta diversi.

13



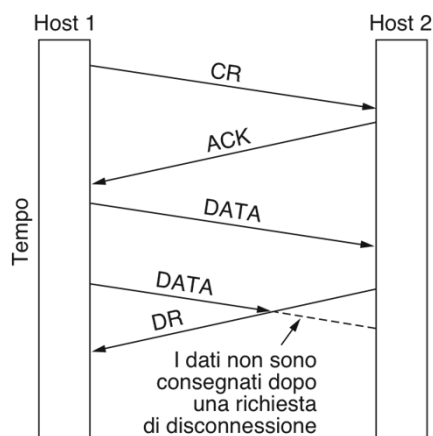
Creazione di una connessione:

- (a) instaurazione normale;
- (b) con una vecchia CONNECTION REQUEST duplicata che appare dal nulla;
- (c) con presenza di CONNECTION REQUEST e ACK duplicati.

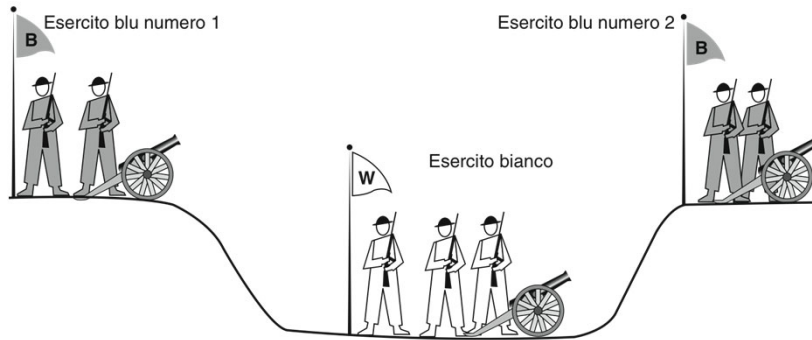
14

Rilascio di una connessione

- l'entità di trasporto rimuove le informazioni sulla connessione dalle proprie tavole e informa l'utente di livello superiore che la connessione è chiusa.
- Rilascio asimmetrico.
- Rilascio simmetrico.

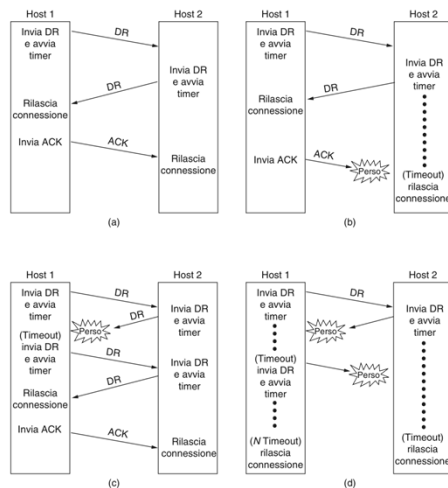


Rilascio asimmetrico improvviso con perdita di dati.



Il problema dei due eserciti

17



Rilascio di una connessione:

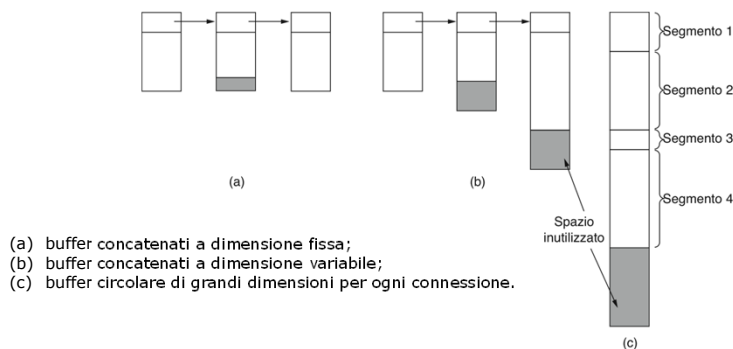
- (a) caso normale di three-handshake;
- (b) ACK finale perduto;
- (c) risposta perduta;
- (d) risposta perduta e DR successivo perduto.

18

Controllo di Flusso e gestione del buffer

- Simile al controllo del livello data link
- Solo un segmento *stop-and-wait*, altrimenti *sliding window*
- Problema di gestione dovuto alla grande variabilità di dimensione dei TPDU
 - *traffico bursty* ma poco gravoso (es. in una sessione di emulazione di terminale): niente buffer a destinazione, bastano buffer alla sorgente;
 - *traffico gravoso* (es. file transfer): buffer cospicui a destinazione, per poter sempre accettare ciò che arriva.

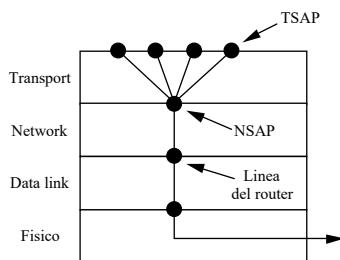
- di norma i protocolli di livello transport consentono alle peer entity di mettersi d'accordo in anticipo (al setup della connessione) sui meccanismi di bufferizzazione da usare.
- spesso la gestione della dimensione delle finestre scorrevoli è disaccoppiata dalla gestione degli ack (caso di TCP).



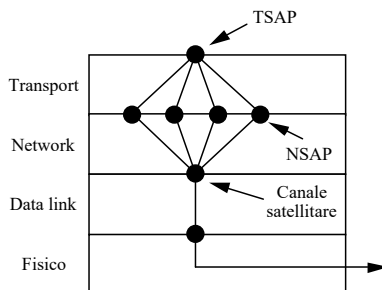
Multiplexing

- Multiplexing delle sessioni di livello transport su quelle di livello network.

In connessioni di livello network costose è conveniente convogliare molte conversazioni transport su un'unica connessione network (**upward multiplexing**).



Per ottenere una banda superiore a quella consentita a una singola connessione network, si ripartisce la conversazione transport su più connessioni network (**downward multiplexing**).



Ripristino dopo il crash

Strategia utilizzata dall'host che invia	Strategia utilizzata dall'host che riceve					
	← Prima ACK, poi scrittura			← Prima scrittura, poi ACK		
	AC(W)	AWC	C(AW)	C(WA)	W AC	WC(A)
Ritrasmetti sempre	OK	DUP	OK	OK	DUP	DUP
Non ritrasmettere mai	LOST	OK	LOST	LOST	OK	OK
Ritrasmette in S0	OK	DUP	LOST	LOST	DUP	OK
Ritrasmette in S1	LOST	OK	OK	OK	OK	DUP

OK = il protocollo funziona correttamente
 DUP = il protocollo genera un messaggio duplicato
 LOST = il protocollo perde un messaggio

23

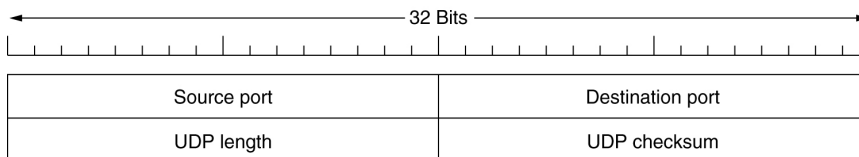
Protocolli di trasporto di Internet (UDP e TCP)

- Il livello transport di Internet è basato su due protocolli:
- **UDP (User Data Protocol)** RFC 768;
- **TCP (Transmission Control Protocol)** RFC 793, 1122 e 1323.

24

UDP (User Data Protocol)

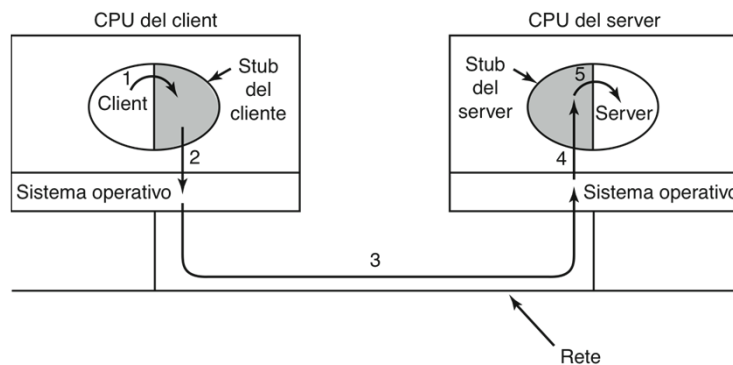
- Il protocollo UDP è di fatto IP con l'aggiunta di un breve header. Fornisce un servizio di trasporto datagram (quindi non affidabile).



- La checksum puo' non essere considerata; cio' avviene tipicamente nel caso di sessioni real-time.

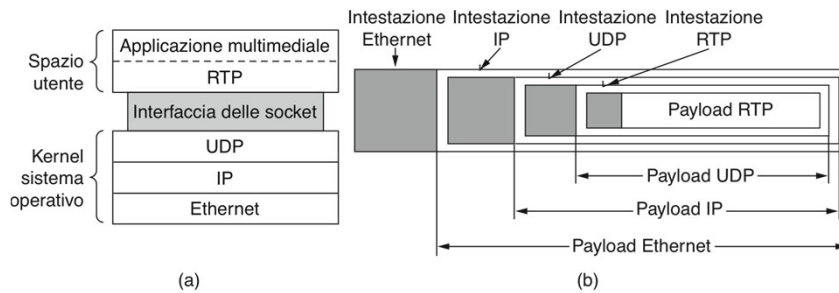
UDP (User Data Protocol)

RPC (Remote Procedure Call) caratterizzato da chiamate a procedure di libreria (stub del client, stub del server).



UDP (User Data Protocol)

- **RTP (Real-time Transport Protocol)**
- **RTCP (Real-time Transport Control Protocol)**

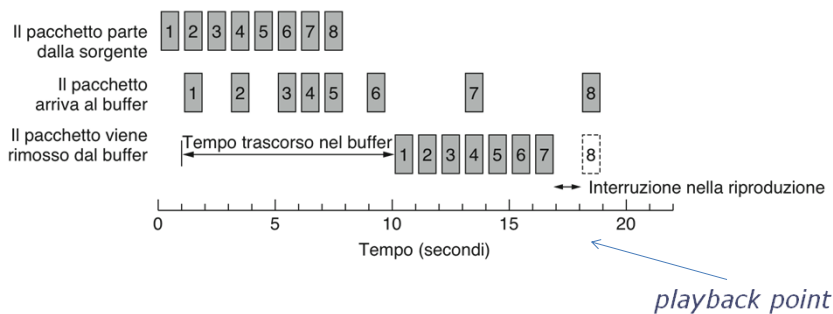


27

UDP (User Data Protocol)

- **RTCP (Real-time Transport Control Protocol)**

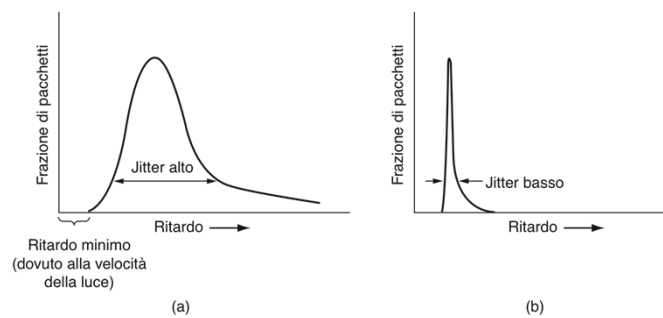
riproduzione con buffering



28

UDP (User Data Protocol)

- **RTCP (Real-time Transport Control Protocol)**

controllo del jitter**Il Protocollo TCP (Transmission Control Protocol)**

Servizio *reliable* e *connection oriented*:

- accettare dati dal livello application;
- spezzarli in **segment**, il nome usato per i TPDU (dimensione massima 64 Kbyte, tipicamente circa 1.500 byte);
- consegnarli al livello network, eventualmente ritrasmettendoli;
- ricevere segmenti dal livello network;
- rimetterli in ordine, eliminando buchi e doppioni;
- consegnare i dati, in ordine, al livello application.
- il protocollo TCP è stato progettato per fornire un flusso di byte affidabile, da sorgente a destinazione, su una rete non affidabile, e' un servizio full-duplex con gestione di ack e controllo del flusso

- I servizi di TCP si ottengono creando connessione di livello transport identificata da una coppia di punti d'accesso detti socket.
- Ogni socket ha un **socket number** che consiste della coppia:

IP address: Port number

- Il socket number costituisce il TSAP.

- I port number hanno 16 bit.
- Quelli minori di 1024 sono i cosiddetti **well-known port**, riservati per i servizi standard.

Porta	Protocollo	Utilizzo
20, 21	FTP	Trasferimento di file
22	SSH	Login remoto, rimpiazzamento di Telnet
25	SMTP	Posta elettronica
80	HTTP	World Wide Web
110	POP-3	Accesso remoto alla posta elettronica
143	IMAP	Accesso remoto alla posta elettronica
443	HTTPS	Web sicuro (HTTP su SSL/TLS)
543	RTSP	Controllo di riproduttori multimediali
631	IPP	Condivisione di stampanti

Il Livello Trasporto III - 3

- Le connessioni TCP sono full duplex e point to point.
- Sono identificate dalla coppia di socket number alle due estremità.
- E' possibile che su un singolo host più connessioni siano attestate localmente sullo stesso socket number.

Reti di calcolatori 33 Prof. Gianni Fenu

33

Il Livello Trasporto III - 3

- Esistenza di un **flag urgent** per interrompere una computazione remota già iniziata (esempio CTRL-C di una sessione di emulazione terminale).

Reti di calcolatori 34 Prof. Gianni Fenu

34

Il protocollo TCP

- Ogni byte del flusso TCP è numerato con un numero d'ordine a 32 bit, usato sia per il controllo di flusso che per la gestione degli ack;
- Un segmento TCP non può superare i 65.535 byte;

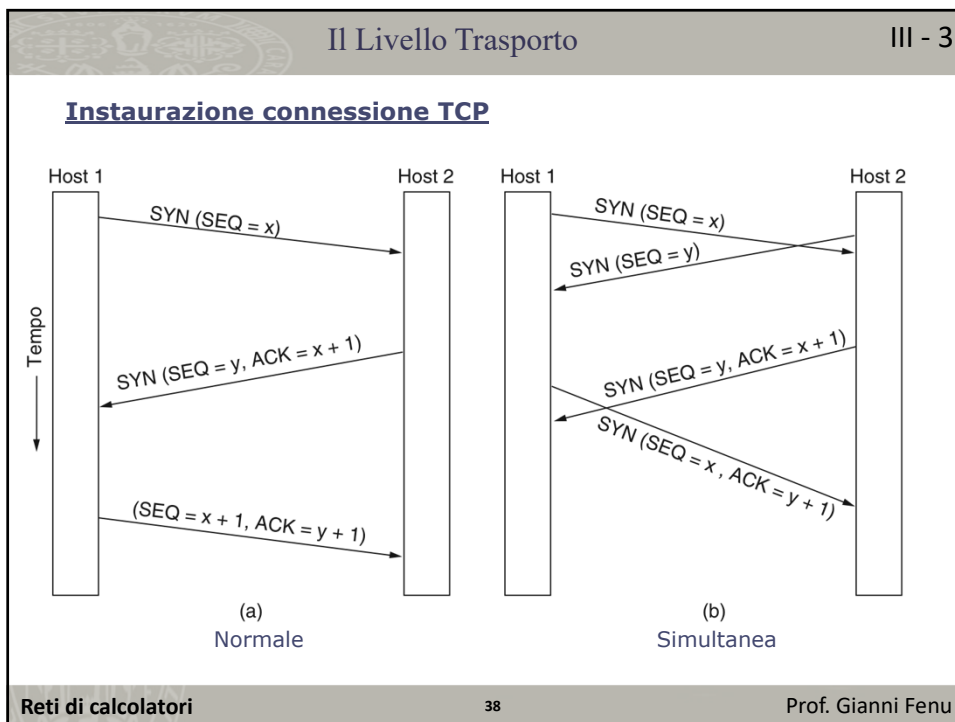
- Un segmento TCP è formato da:
 - un header, a sua volta costituito da:
 - una parte fissa di 20 byte;
 - una parte opzionale;
 - i dati da trasportare;

- TCP usa un meccanismo di sliding window di tipo go-back-n con timeout.
- Se il timeout scade, il segmento si ritrasmette.

- Le dimensioni della finestra scorrevole ed i valori degli ack sono espressi in numero di byte.



37



38

Rilascio connessione TCP

Il rilascio della connessione avviene considerando la connessione full-duplex come coppia di connessioni simplex indipendenti.

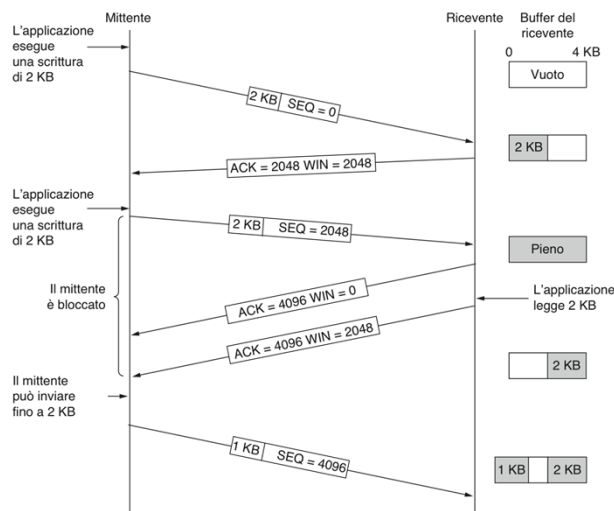
- quando una delle due parti non ha più nulla da trasmettere, invia un fin;
- quando esso viene confermato, la connessione in uscita viene rilasciata;
- quando anche l'altra parte completa lo stesso procedimento e rilascia la connessione nell'altra direzione, la connessione full-duplex termina.

- Per evitare il problema dei 2 eserciti si usano i timer, impostati al doppio della vita massima di un pacchetto.
- Il protocollo di gestione delle connessioni si rappresenta comunemente come una macchina a stati finiti.

Criterio di trasmissione a finestra scorrevole in TCP

- La dimensione delle finestre scorrevoli non è strettamente legata agli ack
- Viene continuamente adattata mediante un dialogo fra destinazione e sorgente.
- Quando la destinazione invia un ack di conferma, dice anche quanti ulteriori byte possono essere spediti.

41



esempio di controllo del flusso TCP

42

Controllo congestione TCP

- Il protocollo TCP assume che, se gli ack non tornano in tempo, ciò sia dovuto a congestione della subnet piuttosto che a errori di trasmissione
- TCP è preparato ad affrontare due tipi di problemi:
 - scarsità di buffer a destinazione;
 - congestione della subnet.

- La scarsità di buffer a destinazione e la congestione della subnet vengono gestiti da una specifica finestra mantenuta dal mittente:
 - la finestra del buffer del ricevitore
 - la **congestion window**, che rappresenta quanto si può spedire senza causare congestione.
- Il mittente si regola sulla più piccola delle due.

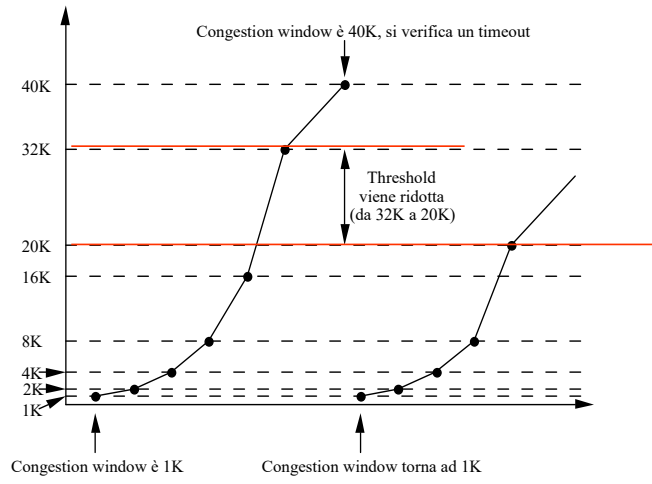
Metodi pratici di gestione delle finestre scorrevoli

- delayed acknowledgement
- algoritmo di Nagle
- silly window syndrome
- cumulative acknowledgement

Gestione dei timer

- RTT, **$SRTT = \alpha SRTT + (1 - \alpha)R$**
con **R** tempo di attraversamento e **α** pari 7/8
- **timer di persistenza**
- **timer keepalive**
- **timed wait**

Controllo della congestione



Esempio di controllo della congestione TCP