Livello di trasporto in Internet

TSAP: Transport Service Access Point

Un TSAP (Transport Service Access Point) rappresenta un indirizzo di livello trasporto.

Il TSAP destinazione (sorgente) serve per identificare univocamente l'applicazione a cui sono destinati (che ha generato) i dati contenuti in un segmento.

In Internet un TSAP ha la forma:

(IP address:port number)

dove IP address è l'indirizzo del nodo in cui è in esecuzione l'applicazione e port number è un numero a 16 bit che serve a identificare in modo univoco l'applicazione

TSAP: Transport Service Access Point

Le applicazioni di tipo server determinano il proprio numero di porta nel momento in cui, utilizzando i servizi di trasporto, si mettono in ascolto di eventuali “chiamate” da parte di applicazioni client.

Per i servizi standard di Internet vengono utilizzate le cosiddette “well known ports”, numeri predefiniti e riservati, fino a 1024

Le applicazioni di tipo client specificano all’entità di trasporto la TSAP con cui vogliono comunicare; il loro numero di porta viene assegnato direttamente dall’entità di trasporto. Si parla in questo caso di porte locali o “effimere”.

Primitive di definizione del servizio: interfaccia Socket

In Internet l'interfaccia tra livello di applicazione e livello di trasporto è basata sui socket.

Sono stati implementati per la prima volta nel 1983 nella versione 4 di Unix BSD (socket di Berkeley).

L'applicazione vede il socket come una coppia di stream, uno aperto in scrittura e uno aperto in lettura.

Scrivere nel socket corrisponde a inviare dati in rete, leggere dal socket corrisponde ricevere da dati dalla rete.

Primitive di definizione del servizio: interfaccia Socket

Nella comunicazione in rete client e server hanno comportamenti diversi:

- il server deve partire per primo, aprire il socket specificando un numero di porta e mettersi in uno stato di “ascolto” per attendere le connessioni dai client;
- il client deve partire dopo il server, aprire il socket specificando indirizzo IP e numero di porta del server e aprire la connessione;
- una volta aperta la connessione, alle operazioni di ”scrittura” nel socket da parte di una delle due applicazioni deve corrispondere una operazione di “lettura” da parte dell’altra

Principali primitive (chiamate al sistema operativo):

- **socket:** crea un punto di comunicazione;
- **connect:** stabilisce una connessione attiva con il server remoto; specifica l'indirizzo IP e porta remoti, sceglie indirizzo IP e porta locali;
- **bind:** specifica indirizzo e porta locale da associare ad un socket;
- **listen:** pone il socket in modo passivo;
- **accept:** accetta una richiesta di connessione;
- **read, write:** ricevono e inviano dati;
- **sendto, recvfrom:** inviano e ricevono dati specificando l'indirizzo IP (servizio datagram);
- **close:** chiude il socket.

Struttura delle applicazioni

Struttura client/server in comunicazione orientata alla connessione:

Server: socket → bind→ listen → accept → (read→ write)→ close

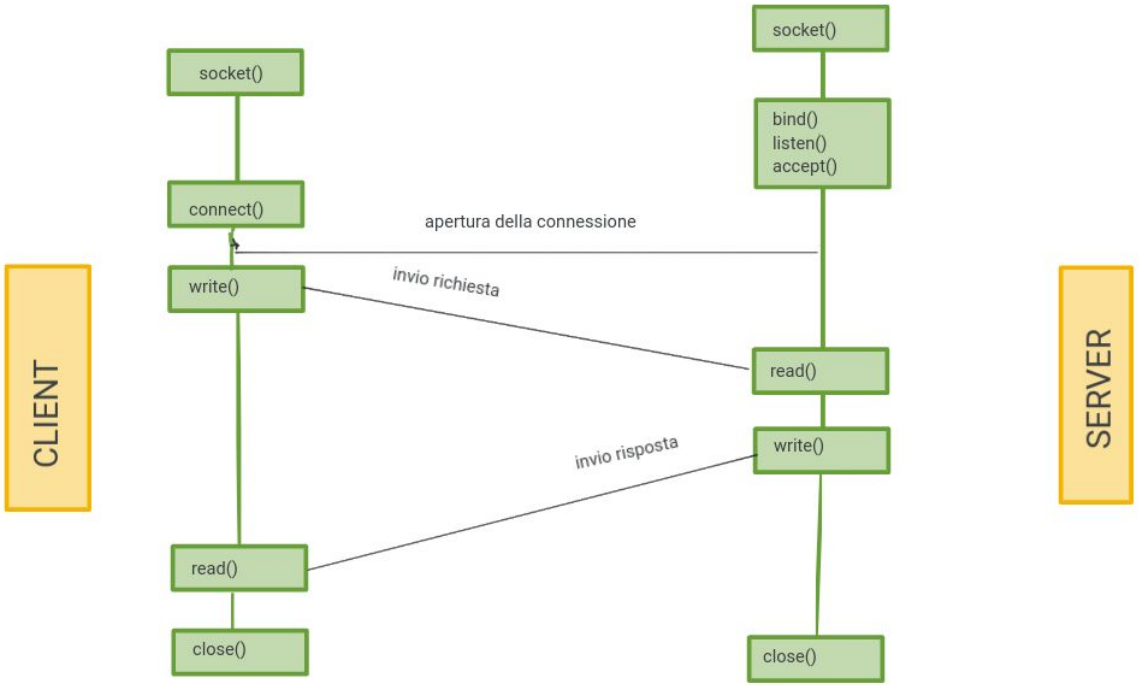
Client: socket → connect → (write→ read)→ close

Struttura client/server in comunicazione NON orientata alla connessione:

Server: socket → bind→ recvfrom → sendto→ close

Client: socket → bind → sendto→ recvfrom→ close

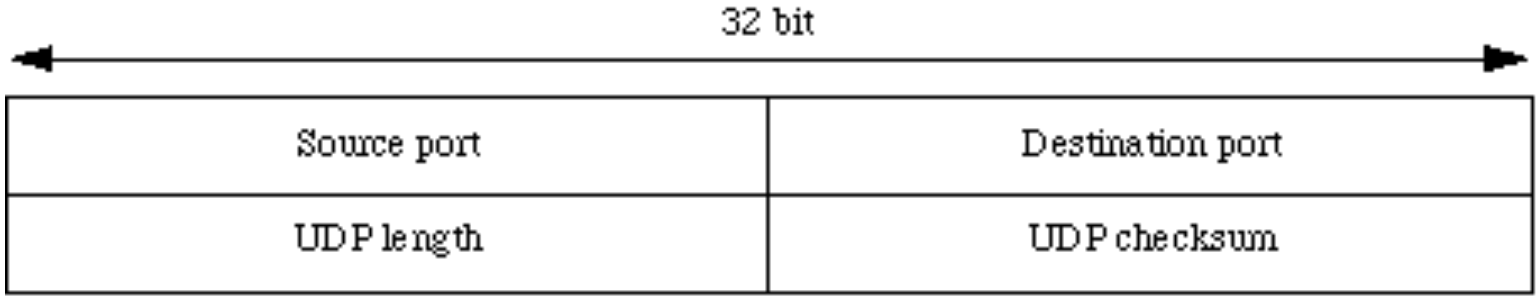
Comunicazione client-server con TCP



UDP: User Datagram Protocol

- È un protocollo non affidabile (cioè non prevede l'invio di riscontri per i segmenti ricevuti) e non orientato alla connessione.
- È utilizzato da applicazioni che prevedono semplici scambi domanda/risposta (es. il DNS), oppure da applicazioni per cui la velocità è preferibile alla perdita di dati (es. VoIP).
- Il funzionamento è semplice, i messaggi dell'applicazione vengono semplicemente inseriti in un segmento e passati a livello di rete.
- La comunicazione tra due applicazioni avviene sempre tramite una coppia di TSAP

Formato intestazione UDP



http://wwwusers.di.uniroma1.it/Reti1/Cap6.pdf

TCP: Transmission Control Protocol

Caratteristiche principali:

- progettato per fornire un flusso di byte affidabile, da sorgente a destinazione, su una rete non affidabile;
- protocollo affidabile e orientato alla connessione;
- offre un servizio full-duplex con gestione di ack e controllo del flusso.

TCP: Transmission Control Protocol

Funzionamento:

- riceve i dati dall'applicazione;
- li organizza in segmenti (dimensione massima 64 Kbyte meno la dimensione di un'intestazione IP, tipicamente circa 1.500 byte);
- li consegna al livello network, eventualmente ritrasmettendoli se necessario;
- riceve segmenti dal livello network, eventualmente rimettendoli in ordine, eliminando buchi e doppioni;
- consegnare i dati, in ordine, al livello applicazione.

TCP: Transmission Control Protocol

Le caratteristiche più importanti sono le seguenti:

- 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 e dai dati da trasportare. L'header a sua volta costituito da una parte fissa di 20 byte e da una parte opzionale.

TCP: Transmission Control Protocol

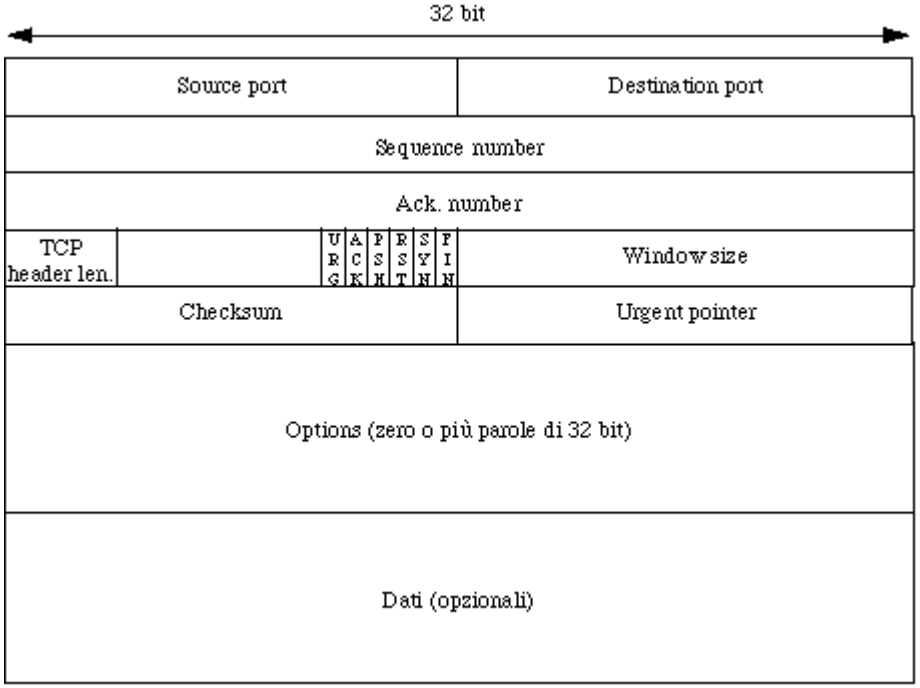
Il TCP usa un meccanismo di finestre scorrevoli di tipo go-back-n con timeout.

Se il timeout scade, il segmento viene ritrasmesso.

Numeri di sequenza dei segmenti, numeri di ack, dimensione della finestra di ricezione sono espressi in numero di byte e non in numero di segmenti.

Questo significa che ogni byte scambiato in una connessione TCP è (virtualmente) numerato.

Formato intestazione TCP



http://wwwusers.di.uniroma1.it/Reti1/Cap6.pdf

Attivazione della connessione TCP

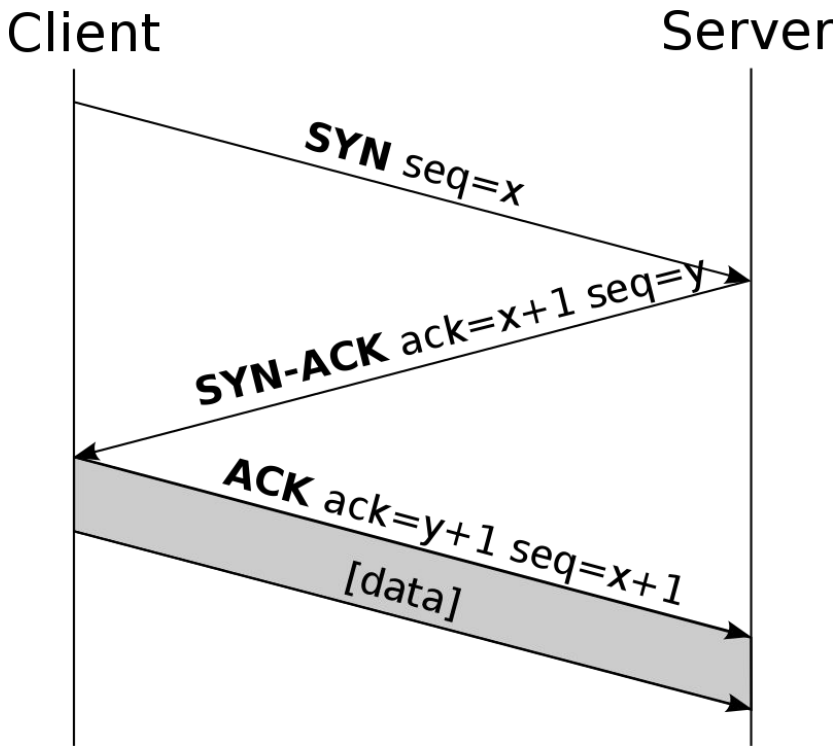
Si utilizza un protocollo three-way-handshake:

- una delle due parti (il server) apre il socket ed esegue le due primitive listen() e accept(), rimanendo così in attesa di una richiesta di connessione su un determinato port number e, quando essa arriva, accettandola;
- l'altra parte (un client) apre il socket ed esegue la primitiva connect(), specificando host, port number ed eventualmente altri parametri, questa primitiva causa l'invio di un segmento TCP con flag SYN a uno e flag ACK a zero;

Attivazione della connessione

- quando il segmento syn arriva a destinazione, l'entity di livello trasporto lato server:
  - se c'è un processo in ascolto sul port number in questione risponde con segmento di conferma con flag SYN e ACK entrambi a 1;
  - altrimenti invia un segmento con flag RST a 1 per rifiutare la connessione;
- quando l'entità TCP lato client riceve il segmento SYN di conferma, considera la connessione aperta e risponde con un segmento di ACK che può già contenere dei dati;
- alla ricezione di questo terzo segmento, anche l'entità TCP lato server considera la connessione aperta.

Apertura della connessione TCP



Numero iniziale di sequenza (ISN)

In TCP i segmenti e gli ack sono numerati. Il numero di sequenza di un segmento rappresenta il numero (virtualmente) associato al primo byte del segmento. Il numero di ack rappresenta il numero di sequenza del prossimo segmento che ci si aspetta di ricevere.

Ciascuna delle due parti all'apertura della connessione "sceglie" un numero iniziale di sequenza (x e y nella figura) prelevandolo da un contatore modulo 2^{32} associato a un timer.

Il contatore completa il suo ciclo di conteggio in circa 4.55 ore, garantendo quindi la condizione di riutilizzabilità del numero.

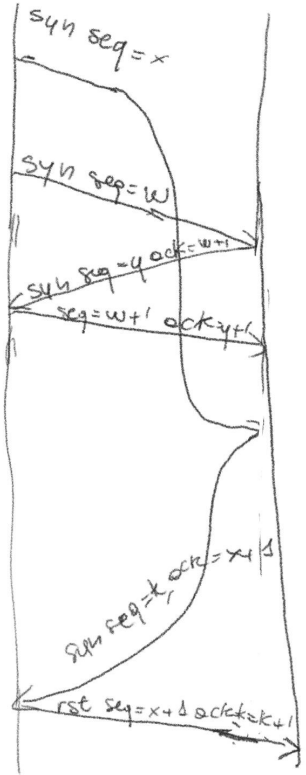
Numero iniziale di sequenza (ISN)

Questo serve a evitare problemi dovuti a eventuali segmenti duplicati ritardatari.

Quando si apre una connessione i numeri di sequenza possono essere riutilizzati solo se si è sicuri che non esistano più in rete vecchi segmenti con gli stessi numeri né come numero di sequenza né come numero di conferma.

Per questo motivo, si deve garantire che il numero iniziale di sequenza possa essere riutilizzato dopo che è trascorso due volte il tempo di vita massimo dei segmenti (Maximum Segment Lifetime – MSL), legato al TTL di IP

Gestione ritardatari duplicati TCP

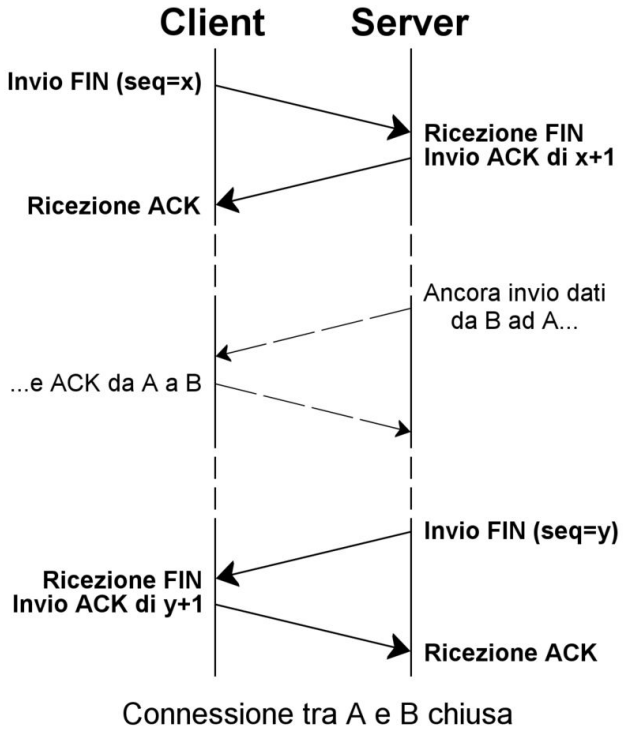


Chiusura connessione TCP

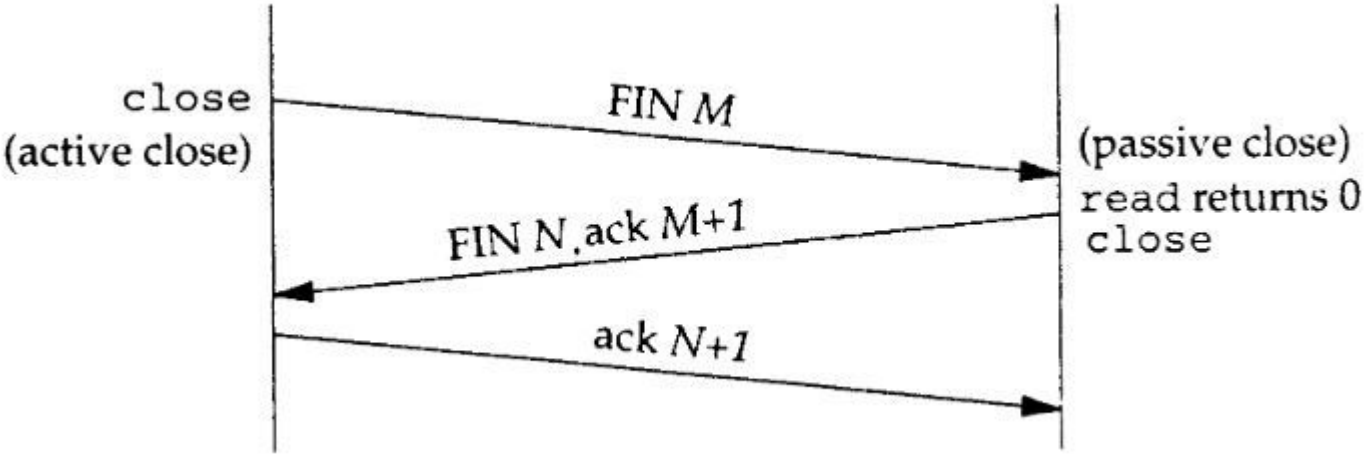
Il rilascio della connessione avviene considerando la connessione full-duplex come una coppia di connessioni simplex indipendenti, e si svolge nel seguente modo:

- quando una delle due parti non ha più nulla da trasmettere, invia un segmento fin ;
- quando il fin 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.

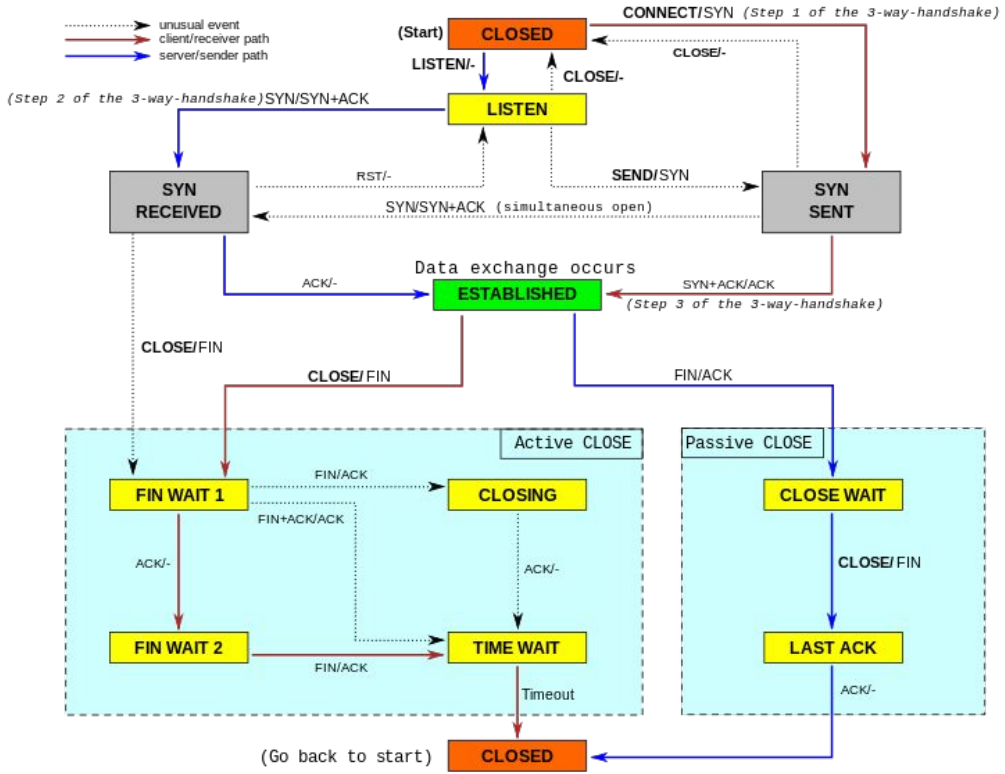
Chiusura connessione TCP (4 segmenti)



Chiusura connessione TCP (3 segmenti)

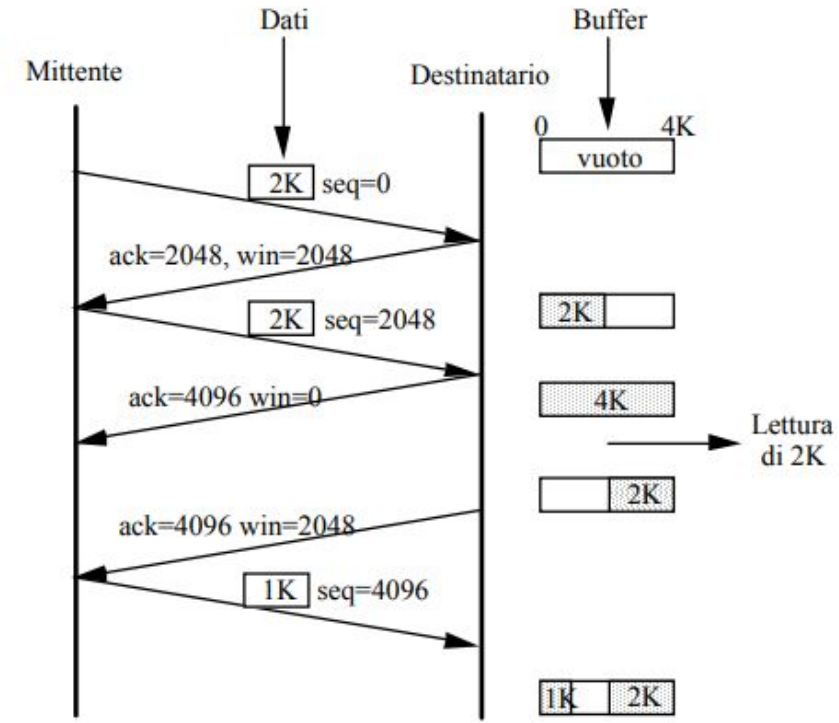


Stati connessione TCP



By Scil100, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=30810617

Controllo di flusso TCP



http://wwwusers.di.uniroma1.it/Reti1/Cap6.pdf

Controllo del flusso

Il campo window size serve per comunicare alla peer entity la dimensione (in byte) della propria finestra di ricezione.

Se si comunica una window size pari a zero, significa che non si possono ricevere dati.

In questo caso, nel momento in cui si ha nuovamente spazio (l'applicazione ha prelevato i dati), lo si deve comunicare alla peer entity in modo da sbloccare la comunicazione.

TCP: controllo della congestione

Nelle prime versioni di TCP la dimensione della finestra di trasmissione era prefissata.

Crescita di Arpanet → aumento delle situazioni di congestione della rete.

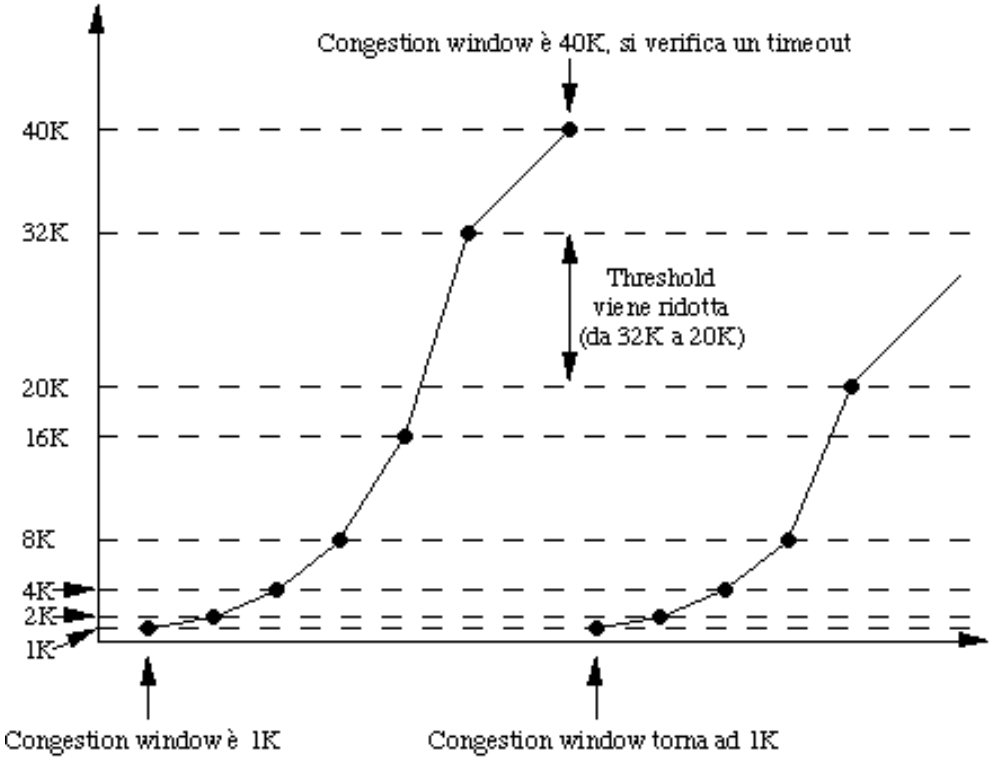
Anni ‘80 → TCP Tahoe/Reno (Van Jacobson) introduzione di sistema per evitare la congestione della rete:

- basato su finestre scorrevoli a dimensione variabile,
- senza modificare né i router né i ricevitori.

Controllo della congestione TCP Tahoe

- Per limitare il formarsi di code nei router i trasmettitori utilizzano una congestion window (cgwd) di dimensioni variabili;
- implementa AIMD (Additive Increase, Multiplicative Decrease);
- come indicazione sullo stato della rete viene utilizzata la perdita di pacchetti (ack che arrivano in ritardo).

TCP congestion window



http://wwwusers.di.uniroma1.it/Reti1/Cap6.pdf

TCP congestion window

La congestion window rappresenta quanto si può spedire senza causare congestione e viene gestita in questo modo:

- il valore iniziale è pari alla dimensione del massimo segmento usato nella connessione;
- ogni volta che un ack torna indietro in tempo la finestra si raddoppia, fino a un valore threshold (inizialmente pari a 64 Kbyte) dopodiché aumenta linearmente di 1 segmento alla volta;
- quando si verifica un timeout per un segmento il valore di threshold viene impostato alla metà della dimensione della congestion window e la dimensione della congestion window viene impostata alla dimensione del massimo segmento usato nella connessione.

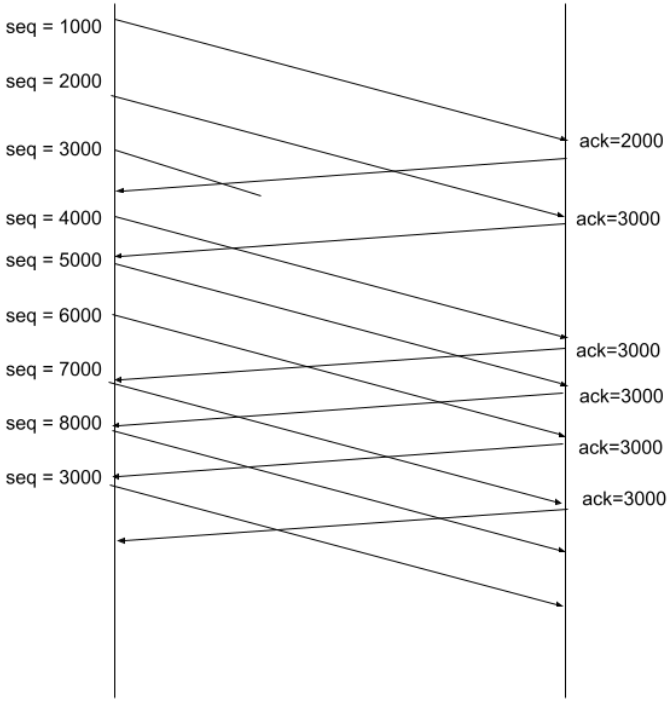
TCP slow start

Obiettivo: determinare il valore ottimale di cgwd, che dipende dall'RTT e dalle velocità dei diversi router e link attraversati.

Si parte quindi da un valore molto basso ma (se tutto va bene) si cresce rapidamente verso il valore di soglia, che approssima il valore ideale di cgwd, dopo di che la dimensione di cgwd continua a crescere molto lentamente (fase AI).

In caso di problemi, il valore di soglia viene dimezzato (fase MD) e si ricomincia con una nuova fase di slow start.

TCP fast recovery



TCP fast recovery

Gli ACK duplicati (che sono comunque possibili) vengono interpretati come sintomo di una possibile perdita di pacchetti.

Per questo motivo il terzo ACK n duplicato viene considerato un sintomo di congestione della rete, per cui:

- si ritrasmette il segmento con numero di sequenza n;
- si dimezza il valore di soglia, si re-imposta il valore di cgwd e si riparte con una fase di slow start.

TCP Reno

In caso di timer scaduto, si comporta come Tahoe.

In caso di 3 ack duplicati invece:

- ritrasmette il segmento con numero di sequenza n (senza aspettare il timeout);
- dimezza il valore di soglia e imposta il valore di cgwd al valore della soglia attuale;
- inizia una fase di incremento additivo (cgwd cresce di un segmento a ogni ack ricevuto).

ECN (Explicit Congestion Notification)

- Richiede modifiche sia al TCP che all'IP;
- se supportata, in fase di apertura di connessione le due parti lo comunicano con i flag CWR ed ECE dell'intestazione;
- se uno dei router del percorso ha i buffer riempiti oltre a una certa soglia, questo viene notificato all'entità TCP ricevente, che imposta a 1 il flag ECE (ECN echo) del segmento di risposta;
- l'entità TCP mittente comunica l'avvenuta ricezione di ECE impostando a 1 CWR (congestion window reduced).