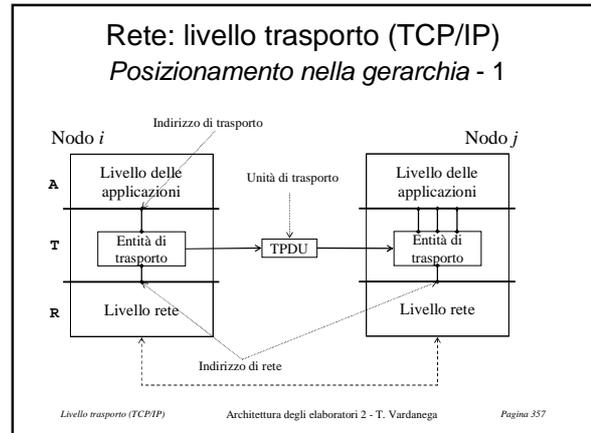


Rete: livello trasporto (TCP/IP) Parte 1 - Indice

1. Posizionamento nella gerarchia
2. **Socket**
3. Indirizzamento
4. Connessione
5. Controllo di flusso

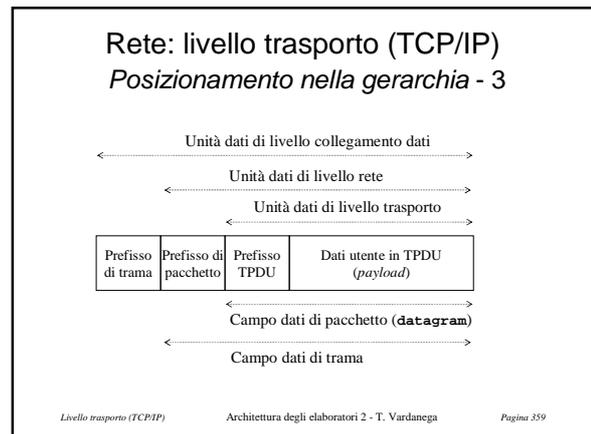
Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 356



Rete: livello trasporto (TCP/IP) Posizionamento nella gerarchia - 2

- Si pone tra il livello delle applicazioni **A** (sessione, se presente) ed il livello rete **R**
 - I servizi richiesti sono resi da “entità di trasporto”, che nascondono all’utente l’esecuzione dei relativi protocolli
 - Gli utenti in comunicazione vedono semplicemente un flusso di dati
- Offre ad **A** più punti di accesso a servizi di trasporto, mappandoli su un unico indirizzo nello spazio di **R**
- È necessario per l’inaffidabilità del livello rete, anche quando in versione orientata alle connessioni
 - Descrive come “qualità del servizio” lo sforzo impiegato per raggiungere il livello di affidabilità richiesto

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 358



Rete: livello trasporto (TCP/IP) Posizionamento nella gerarchia - 4

- Così come il livello 2 ha a che fare con un livello fisico non affidabile, il protocollo di livello trasporto deve gestire l’inaffidabilità del livello rete (flusso, congestione, errore)
 - Livello 2: connessione tra due *router* agli estremi di un collegamento punto a punto
 - Livello 4: connessione tra due nodi agli estremi di un collegamento su rete

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 360

Rete: livello trasporto (TCP/IP) Socket - 1

- Secondo il modello UNIX BSD
- Il **socket** è un terminale (*end point*) di comunicazione di livello trasporto
 - **socket** diversi per comunicazioni con o senza connessione (TCP o UDP)
- Al **socket** destinazione corrisponde un indirizzo locale (porta) associato all’indirizzo **IP** del nodo di residenza
- A livello implementativo, un **socket** è una coda di ricezione con uno stato controllato dal S/O
- Il processo utente associato al **socket** si blocca in attesa di comunicazioni in ingresso (destinatario) o di conferma di ricezione (mittente, con connessione)

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 361

Rete: livello trasporto (TCP/IP) Socket - 2

- API di UNIX BSD (1/2)
 - **socket(...)** : crea un **socket**, restituendo al chiamante l'ID del descrittore del *file* speciale corrispondente
 - **bind(...)** : associa una porta al **socket**, creando così la sua identità di rete
 - Il lato *server* può così pubblicare la propria identità
 - Il cliente usa la propria identità per stabilire il collegamento con il proprio *server*
 - **connect(...)** : blocca il chiamante fino allo stabilirsi della connessione richiesta
 - Il cliente vi indica l'indirizzo del *server* ed il proprio **socket** locale viene creato automaticamente (senza invocazione di **bind**)

Livello trasporto (TCP/IP)Architettura degli elaboratori 2 - T. VardanegaPagina 362

Rete: livello trasporto (TCP/IP) Socket - 3

- API di UNIX BSD (2/2)
 - **listen(...)** : dimensiona la coda associata al **socket** di lato *server* e pone il chiamante in attesa di richieste
 - Lo stesso *server* può servire più connessioni simultanee
 - **accept(...)** : blocca il chiamante fino ad una richiesta di connessione, ricevuta la quale le associa un **socket** identico a ma distinto da quello sul quale la richiesta è pervenuta
 - Un agente del *server* ascolta su **socket** iniziale mentre varie istanze di *server* (anche diversi tra loro) lavorano su **socket** creati dinamicamente
 - **send(...)** / **receive(...)** : invia o riceve dati su un **socket**
 - **close(...)** : mittente e destinatario rilasciano il proprio **socket** e dunque la relativa connessione

Livello trasporto (TCP/IP)Architettura degli elaboratori 2 - T. VardanegaPagina 363

Rete: livello trasporto (TCP/IP) Socket - 4

- I **socket** BSD hanno un tipo, che specifica:
 - Il dominio di indirizzamento
 - **AF_INET** denota l'indirizzamento **IP** : porta
 - Lo stile di comunicazione
 - **SOCK_DGRAM** denota l'uso di **datagram**
 - Il tipo di comunicazione
 - **IPPROTO_TCP** / **_UDP** denota i protocolli a noi noti
- **Socket** di tipo **datagram** inviano e ricevono usando **sendto** e **recvfrom** che identificano destinatario e mittente
 - **send** e **receive** operano su qualunque connessione aperta

Livello trasporto (TCP/IP)Architettura degli elaboratori 2 - T. VardanegaPagina 364

Rete: livello trasporto (TCP/IP) Indirizzamento - 1

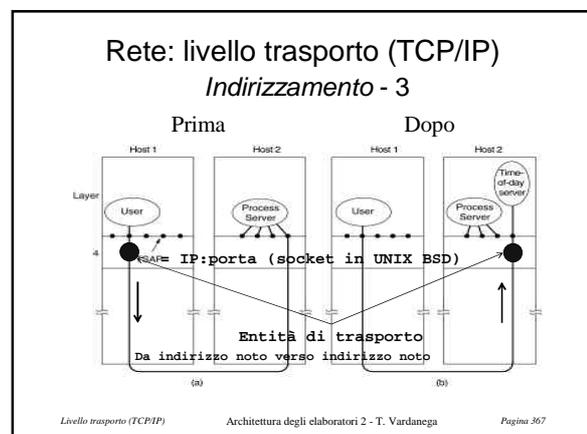
- La comunicazione a livello trasporto richiede conoscenza dell'indirizzo locale del destinatario (p.es.: **IP:porta**)
 - Tale forma di indirizzamento è detta "gerarchica"
- L'indirizzo destinazione può essere noto a priori
 - Alcune applicazioni hanno una assegnazione di porta standard, dunque nota ad ogni utente
- Altrimenti può essere identificato tramite un servizio d'anagrafe (*name server*) che lo associa al nome logico standard dell'applicazione richiesta

Livello trasporto (TCP/IP)Architettura degli elaboratori 2 - T. VardanegaPagina 365

Rete: livello trasporto (TCP/IP) Indirizzamento - 2

- Mittente e destinatario sono associati dalle rispettive entità di trasporto ad uno specifico punto di accesso locale
- Alcuni processi destinazione (*server*) sono permanentemente in attesa dietro il loro punto di accesso
- Altri processi usano invece un agente (*proxy*) che ascolta il loro vece ed assegna loro dinamicamente un punto di accesso all'arrivo di una richiesta

Livello trasporto (TCP/IP)Architettura degli elaboratori 2 - T. VardanegaPagina 366



Rete: livello trasporto (TCP/IP) Connessione - 1

- La rete ha “memoria” poiché non cancella automaticamente tutti i duplicati di pacchetti consegnati con successo né le loro conferme
 - Pacchetti duplicati arrivati tardi a destinazione possono esser trattati come nuova conversazione
 - Ogni connessione dovrebbe invece avere inizio e fine univoci, che invalidino flussi duplicati
- Il problema è reso ancor più complesso dal fatto che i nodi possono perdere traccia del loro stato di comunicazione

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 368

Rete: livello trasporto (TCP/IP) Connessione - 2

- Tutti i pacchetti che viaggiano sulla rete possono accumulare ritardo
 - Pacchetti dati, conferme e pacchetti di controllo
- Una tecnica che offre garanzie di inizializzazione consistente a mittente **M** e destinatario **D** è detta *three-way handshake*
 - **M** → **D** (*rc,x*): richiesta di connessione, con emissione di pacchetti numerati a partire da *x*
 - **D** → **M** (*cc,y,x*): conferma di connessione (*rc,x*), con emissione di pacchetti numerati a partire da *y*
 - **M** → **D** (*dat i,y*): invio del primo pacchetto dati, numerato *x*, con conferma dell'indice *y* di **D**

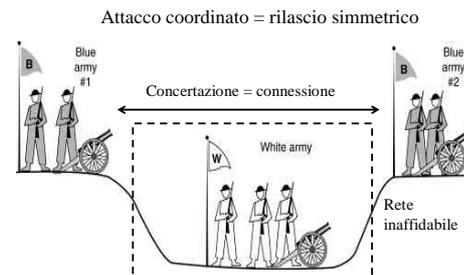
Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 369

Rete: livello trasporto (TCP/IP) Rilascio

- La terminazione di una connessione può avvenire in modo asimmetrico o simmetrico
 - Asimmetrico: conversazione telefonica; il rilascio di una parte distrugge la connessione
 - Può comportare perdita di dati → non desiderabile
 - Simmetrico: il rilascio deve avvenire da ambo i lati della connessione
 - Il **socket** richiede rilascio simmetrico
 - Può lasciare la connessione aperta all'infinito (“il problema dei 2 eserciti”)
 - I lati delle connessioni che non portano dati validi entro intervalli fissati sono autonomamente rilasciati dal rispettivo possessore

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 370

Rete: livello trasporto (TCP/IP) Illustrazione del problema (senza soluzione)



Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 371

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

- Problematica analoga a quella incontrata al livello del collegamento dati
 - Protocolli tipo **SWP** e **SRP** richiedono di trattenere pacchetti (trame) in memoria ai lati della connessione
 - I **router** connettono relativamente poche linee, il che richiede *buffer* di ampiezza accettabile
 - Nodi **M** e **D** possono ospitare numerosissimi flussi di conversazione, il che richiede molta più memoria
- Se la rete è inaffidabile (caso **Internet**) **M** e **D** devono salvare tutti i pacchetti inviati previsti dal protocollo adottato

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 372

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

- Allocazione statica di memoria pacchetti
 - Come un insieme di *buffer* di dimensione fissa ciascuno assegnato ad 1 pacchetto
 - Enorme spreco di memoria causato dal necessario dimensionamento dei *buffer* a massima ampiezza
- Allocazione dinamica
 - 1 *buffer* circolare \forall connessione
 - Spreca memoria se la connessione ha poco traffico, altrimenti ha buona resa

Livello trasporto (TCP/IP) Architettura degli elaboratori 2 - T. Vardanega Pagina 373

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

- Allocazione dinamica
 - **SWP** e **SRP** legano la gestione dei *buffer* all'arrivo di conferme di ricezione; questo non conviene in ambiente di rete
 - Meglio che **M** richieda a **D** un'ampiezza di *buffer* dimensionata alle sue aspettative di traffico e **D** indichi, all'atto di attivazione della connessione, la propria disponibilità attuale
 - Lo spazio reso disponibile da **D** si riduce progressivamente ad ogni TPDU emesso da **M**, che diventa così capace di regolare il suo flusso in uscita

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 374

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

- Allocazione dinamica
 - Un meccanismo di controllo basato solo sulla capacità di ricezione di **D** assume (erroneamente) capacità trasmissiva infinita della rete
 - Conviene ad **M** evitare di emettere troppi TPDU senza conferma
 - Con n TPDU/s capacità di assorbimento della rete e p tempo medio intercorrente tra l'emissione di TPDU e l'arrivo della sua conferma, un *buffer* di dimensione $n \times p$ consente ad **M** di operare a massima velocità avendo però il *buffer* sempre mediamente pieno

Livello trasporto (TCP/IP)

Architettura degli elaboratori 2 - T. Vardanega

Pagina 375