



ITIS G. Marconi - Catania

Le Reti Informatiche

modulo 8



Prof. Salvatore Rosta

www.byteman.it

s.rosta@byteman.it

Il Livello di Trasporto: 1

- L'utente **non ha il controllo sulla rete**; non può risolvere i problemi di un servizio inadeguato utilizzando router migliori.
- L'unica possibilità è quella di collocare un **ulteriore strato** sopra il livello di rete che migliori la qualità del servizio.
- Il livello di trasporto è di solito gestito da un software chiamato **entità di trasporto**.

Il Livello di Trasporto: 2

- L'**entità di trasporto** può essere situata:
 - nel nucleo del sistema operativo,
 - in un processo utente separato,
 - in un pacchetto di libreria collegato alle applicazioni di rete ma **NON** nei router !
- In pratica, l'esistenza del livello di trasporto fa in modo che il servizio di comunicazione sia più affidabile di quello offerto dal livello di rete sottostante.
- Eventuali problemi relativi a **pacchetti persi** o a **dati corrotti**, possono essere rilevati e risolti dal livello trasporto.

Il Livello di Trasporto: 3

- Si occupa della suddivisione dei messaggi in pacchetti e della loro trasmissione. Definisce 2 protocolli principali.
- **TCP** (Transmission Control Protocol) è un protocollo **orientato alla connessione** ed **affidabile** (ossia tutti i pacchetti arrivano, e nell'ordine giusto). *Circuito virtuale.*
- **UDP** (User Data Protocol) è un protocollo **non connesso** e **non affidabile**, i pacchetti possono arrivare in ordine diverso o anche non arrivare senza che vengano ritrasmessi. *Datagramma.*

Il Livello di Trasporto: 4

- **TCP** viene utilizzato per quelle applicazioni che richiedono un servizio **orientato alla connessione** (posta elettronica, file sharing).
 - E' caratterizzato da una **maggiore affidabilità** nel trasporto dei dati per effetto di una serie di servizi appositamente pensati (gestione del flusso, della congestione...).
 - Si comprende quindi come sia **più sicuro** ma anche **più lento** perché scambia non solo dati reali, ma anche dati di servizio.
- **UDP** per le applicazioni in tempo reale (on-line gaming, streaming audio e video). **UDP punta sulla velocità di trasmissione** a scapito della sicurezza.

Il Protocollo TCP: 1

- TCP stabilisce una **connessione** prima di trasmettere i dati. Si tratta di una **connessione virtuale** detta **sessione** (tra due computer **A** e **B**).
- Lo stabilirsi di questa connessione viene detto **Three way handshake** (tripla stretta di mano) perché consiste di tre passaggi.

Il Protocollo TCP: 2

- 1 - **A** invia un segmento di inizio connessione a **B**, contenente il suo **numero di sequenza iniziale x**.
- 2 - **B** invia un segmento di inizio connessione ad **A**, contenente l'**acknowledgment** del numero di sequenza iniziale **x** di **A** e il suo **numero di sequenza iniziale y**.
- 3 - **A** invia un segmento a **B** con l'**acknowledgment** del numero di sequenza iniziale **y** di **B**.

Il Protocollo TCP: 3

- Tra le intestazioni del livello di trasporto i **numeri di porta** sono lo strumento utilizzato per realizzare la **multiplazione** delle connessioni e delle applicazioni.
- Più **connessioni** contemporanee e più **applicazioni** di rete contemporanee, in modo che i dati contenuti nei pacchetti in arrivo vengano indirizzati al processo che li sta aspettando.

Il Protocollo TCP: 4

- Per poter inviare con successo un pacchetto con una certa porta destinazione, ci deve essere un processo **in ascolto** su quella porta, ovvero che ha chiesto al sistema operativo di ricevere connessioni su quella porta (ed è stato autorizzato).
- I numeri di porta sono a **16 bit** e vanno da **0000** a **FFFF**, ossia da **0** a **65535**.
- Quando più applicazioni diverse sono assegnate a una stessa porta si genera un conflitto.

Il Protocollo TCP: 5

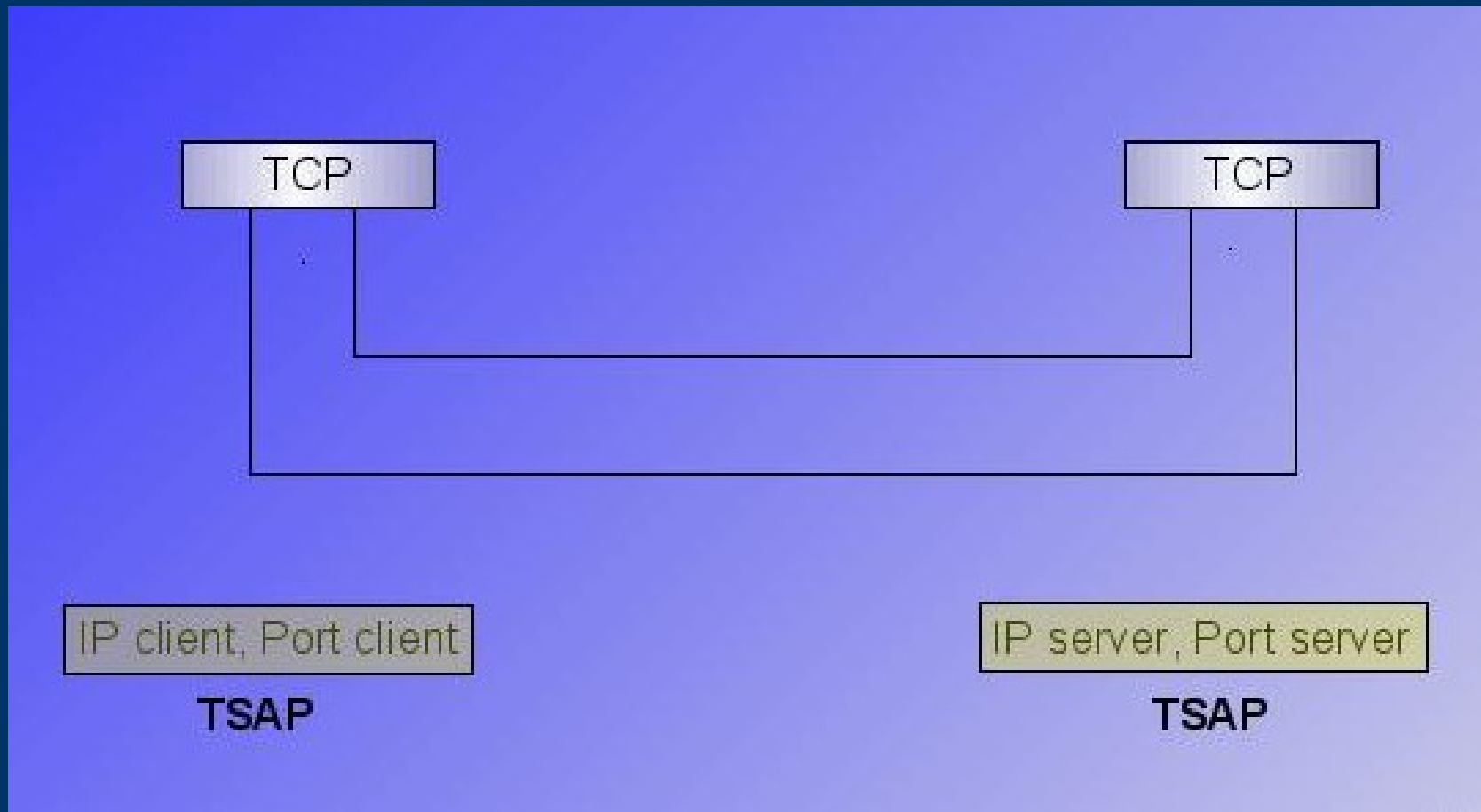
Tre categorie di porte:

- **Porte note** (*known ports*) sono le porte TCP e UDP nell'intervallo **0-1023** (primi **10 bit: 0000-03FF**) e sono assegnate a specifici servizi dall'ICANN. Si riferiscono alle applicazioni di rete standard.
- **Porte registrate** vanno da **1024 a 49151 (0400-BFFF)** e devono essere registrate presso l'ICANN relativamente all'applicazione cui si riferiscono.
- **Porte non registrate**, da **49152 a 65535 (C000-FFFF)** possono essere usate liberamente.

Il Protocollo TCP: 6

- Quando un'applicazione vuole connettersi con un processo remoto, deve specificare l'identificativo del processo.
- Quello che viene fatto è definire degli indirizzi di trasporto **TSAP** (**T**ransport **S**ervice **A**ccess **P**oint) presso i quali i processi possono attendere le richieste di connessione.
- I punti di accesso sono coppie **Indirizzo IP : Porta locale**
- Un'entità di trasporto gestisce più di un TSAP.

Il Protocollo TCP: 7



Il Protocollo TCP: 8

- Nel protocollo TCP, i punti di accesso vengono anche chiamati **socket** e, come detto, sono caratterizzati dall'indirizzo IP dell'host (4 byte) e da un numero detto **porta** (2 byte) che rappresenta l'identificativo del servizio fornito dall'host.
- Le porte inferiori alla 1023 sono dette porte **ben note** (well-know port) e sono riservate per servizi standard.
- Le altre porte possono essere scelte arbitrariamente.

Il Protocollo TCP: 9

Servizio	Porta	Prot.
FTP (File Transfer Protocol)	21	TCP
Telnet	23	TCP
SMTP (Simple Mail Transfer Protocol)	25	TCP
TFTP (Trivial File Transfer Protocol)	69	UDP
DNS (Domain Name Server)	53	TCP
HTTP (Hyper Text Transfer Protocol)	80	TCP
POP (Post Office Protocol)	110	TCP
SNMP (Simple Network Management Protocol)	161	UDP

Il Protocollo TCP: 10

Supponiamo di avere all'interno della nostra LAN una macchina con IP: 192.34.12.80.

Potremo allora riferirci ai vari servizi in questa maniera:

192.34.12.80:23 per il telnet

192.34.12.80:25 per il server della posta installato

192.34.12.80:21 per il server ftp

ecc...

Il Protocollo TCP: 11

Tutte le connessioni TCP sono full duplex e punto-punto.
Ogni connessione ha, quindi, esattamente due estremi:

socket 1 <=====> socket 2

(socket 1 , socket 2) = connessione

Il protocollo TCP non supporta il multicasting e il broadcasting.

Il Protocollo TCP: 12

Primitive dei socket:

<code>accept()</code>	Si blocca finché qualcuno cerca di connettersi
<code>receive()</code>	Si blocca finché arriva una TPDU
<code>connect()</code>	Cerca di stabilire una connessione (TPDU = conn.request)
<code>send()</code>	Invia dei dati (TPDU = dati)
<code>disconnect()</code>	Chiede di terminare la connessione (TPDU = disconn.request)

TPDU (**T**ransfer **P**rotocol **D**ata **U**nit)