

Unità Formativa (UF): RETI DI CALCOLATORI

Docente: L.MORELLO

Titolo argomento: LIVELLO 3 ISO/OSI



in collaborazione con:



per una crescita intelligente,
sostenibile ed inclusiva

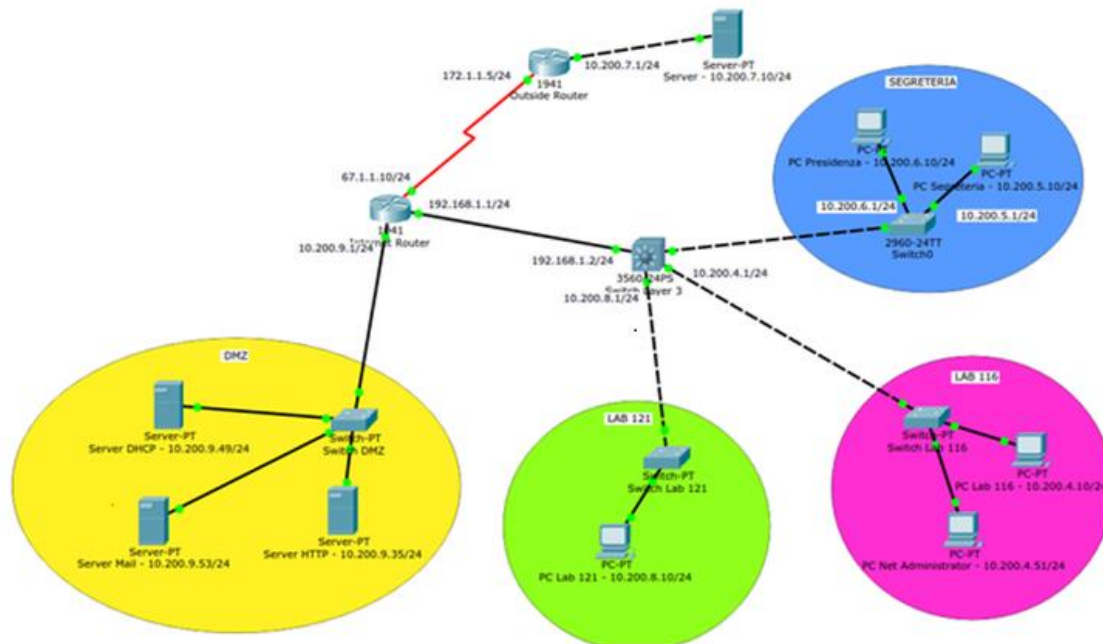
www.regione.piemonte.it/europa2020

INIZIATIVA CO-FINANZIATA CON FSE

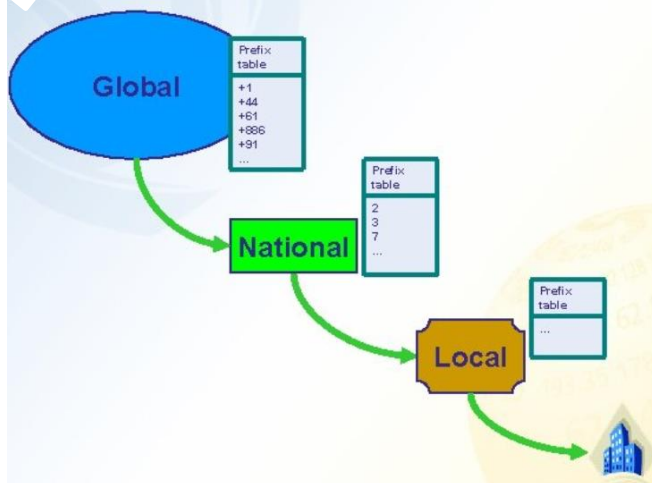
Nota di Copyright

- Questo insieme di trasparenze (detto nel seguito slide) è protetto dalle leggi sul copyright e dalle disposizioni dei trattati internazionali. Il titolo ed i copyright relativi alle slide (ivi inclusi, ma non limitatamente, ogni immagine, fotografia, animazione, video, audio, musica e testo) sono di proprietà dell' autore indicato alla prima slide o a piè di pagina. Le slide possono essere riprodotte ed utilizzate liberamente dagli studenti per scopi didattici, non a fine di lucro. In tal caso non è richiesta alcuna autorizzazione. Ogni altra utilizzazione o riproduzione (ivi incluse, ma non limitatamente, le riproduzioni su supporti magnetici, su reti di calcolatori e stampate) in toto o in parte è vietata, se non esplicitamente autorizzata per iscritto, a priori, da parte dell'autore. L'informazione contenuta in queste slide è ritenuta essere accurata alla data dell'edizione. Essa è fornita per scopi meramente didattici e non per essere utilizzata in progetti di impianti, prodotti, reti, ecc. In ogni caso essa è soggetta a cambiamenti senza preavviso. L'autore non assume alcuna responsabilità per il contenuto di queste slide (ivi incluse, ma non limitatamente, la correttezza, completezza, applicabilità, aggiornamento dell'informazione). In ogni caso non può essere dichiarata conformità all'informazione contenuta in queste slide.
- L'utilizzo del contenuto della lezione sono riservati alla fruizione personale degli studenti iscritti al corso di Reti di Calcolatori del ITS WEB AND MOBILE. Sono vietati la diffusione intera o parziale di video ed immagini della lezione, nonché la modifica dei contenuti senza il consenso, espresso per iscritto, del titolare del corso, autore dei materiali.
- In ogni caso questa nota di copyright non deve mai essere rimossa e deve essere riportata anche in utilizzi parziali.

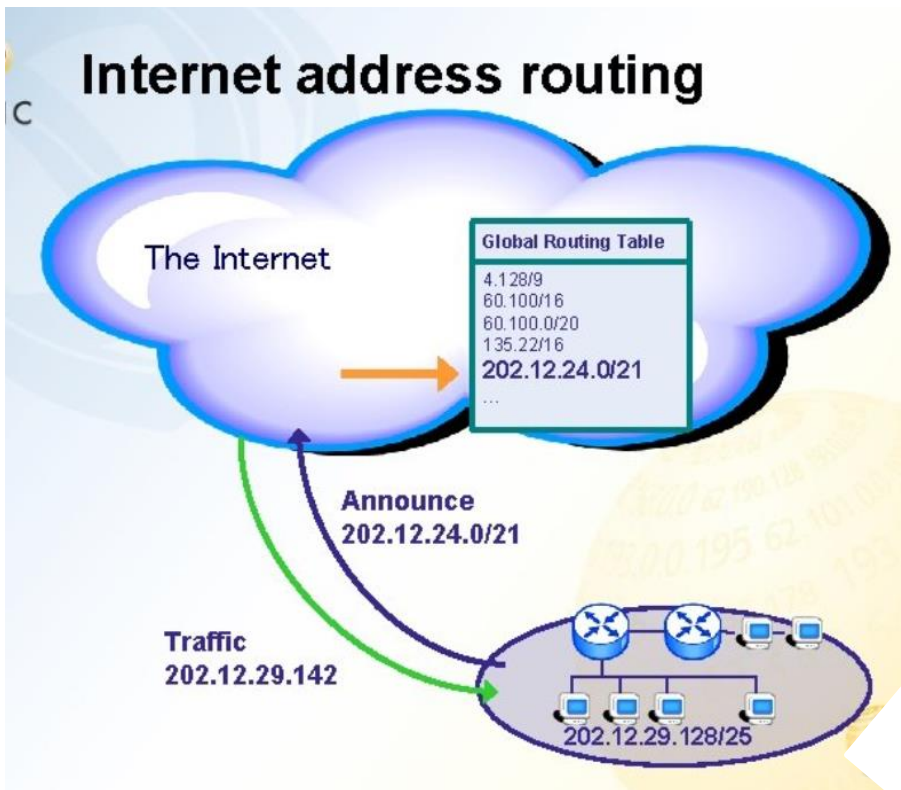
Livello 3



Telephone network routing



Internet address routing



Dal lato di scambio dell'informazione:

- ▣ Indirizzamento --- > identificazione univoca dell'interfaccia di rete di un host/router
- ▣ Inoltro/forwarding---- > funzione locale attraverso cui il router trasferisce i pacchetti dall'ingresso all'uscita
- ▣ Instradamento/Routing ---- > processo che determina i percorsi dei pacchetti dalla sorgente alla destinazione; è un processo svolto dagli algoritmi di routing

■ Dal lato del controllo della comunicazione:

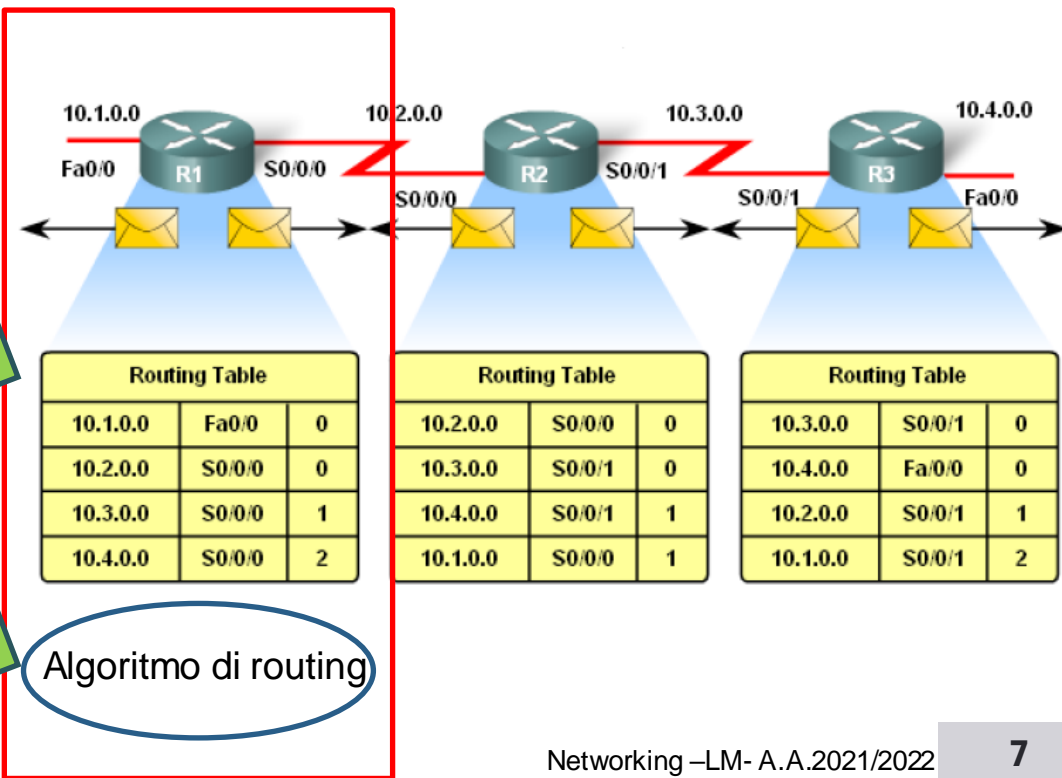
■ Si hanno protocolli quali ICMP, ARP, RARP, RIP, OSPF

■ segmenti dello strato di trasporto vengono trasferiti dallo strato di rete all'host sorgente all'host destinazione

- ▣ Lato sorgente, vengono incapsulati in pacchetti
- ▣ I Pacchetti vengono inoltrati hop-by-hop fino a destinazione
- ▣ router esaminano i campi dell'header di ciascun pacchetto IP che li attraversa
- ▣ Lato destinazione, i pacchetti vengono consegnati allo strato di trasporto

- Ancora: dal lato scambio dell'informazione:
- **Best Effort e senza connessione:** ogni router che riceve un packet, legge l'header ,e decide come/dove inoltrarlo sulla base di:
 - Un “indirizzo di destinazione” presente nell'header del packet
 - Una tabella di instradamento presente in ogni nodo (routing table)
 - I pacchetti possono (potenzialmente) percorrere strade diverse tra sorgente e destinazione, la ricomposizione viene fatta a destinazione

- La tabella di routing determina il local forwarding
- Gli algoritmi di routing determinano il percorso end-to-end attraverso la rete



Livello 3- Indirizzo IP

- E' un numero binario di 32 bit
- viene scritto nella forma x.y.z.w. Ciascuno dei x,y,z,w rappresenta 8 bit
- può assumere tutti i valori da 00000000 a 11111111 (0 –255 in base 10)
- si usa scriverlo nella forma decimale (notazione decimale puntata)

Livello 3 – Indirizzo IP

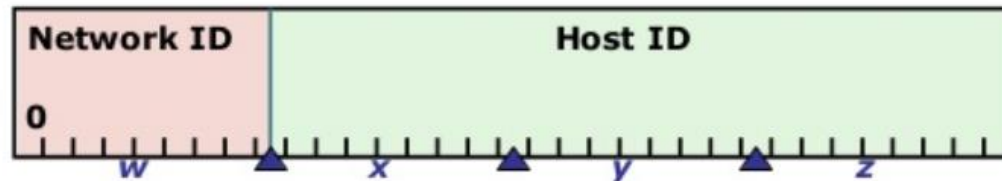
- E' associato in modo univoco ad un'interfaccia di rete di un host o di un router
- Non è associato direttamente a un host o a un router perché questi possono avere più interfacce di rete
- Indirizzo IP deve avere valenza e univocità universali (in tutto Internet)
- Il routing IP è basato sull'indirizzo dell'host destinazione
- Ogni gestore di rete ha a disposizione un blocco di indirizzi che distribuisce alle interfacce dei singoli apparati

Livello 3 – Indirizzo IP

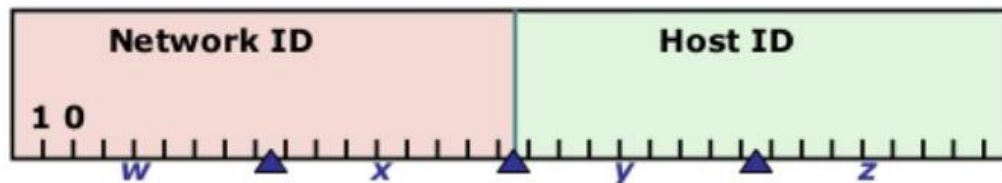
- Un blocco elementare di indirizzi IP è assegnato alle interfacce di una rete IP
- Gli indirizzi del blocco devono avere identici i primi n bit
- Questi primi n bit si chiamano prefisso(o identificativo) di rete (networkprefixo NetID)
- L'indirizzo IP è dunque diviso in due campi (o livelli)–I primi n bit (prefixo NetID) identificano la rete–I rimanenti bit (HostID) sono usati per identificare un host specifico (un'interfaccia) nell'ambito della rete
- •Il valore di n dipende dal tipo di rete

Classi di indirizzi

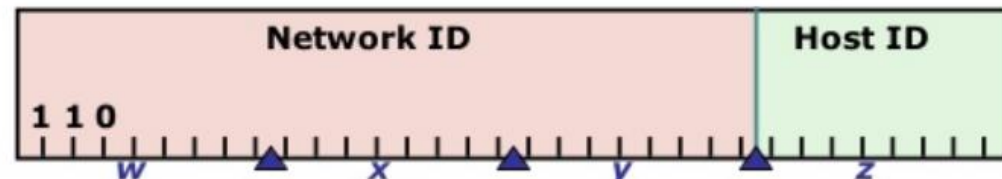
Classe A
Grandi reti



Classe B
Reti medie



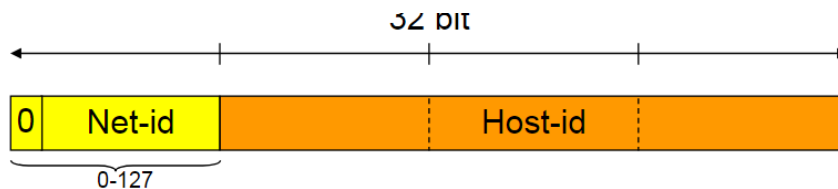
Classe C
Piccole reti



Classi di indirizzi

Classe A :

0.0.0.0 – 127.255.255.255



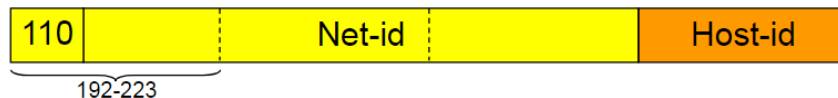
Classe B :

128.0.0.0 – 191.255.255.255



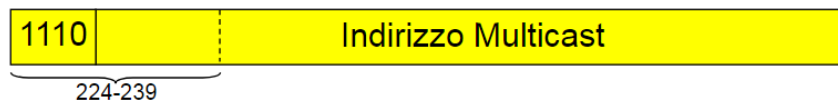
Classe C :

192.0.0.0 – 223.255.255.255



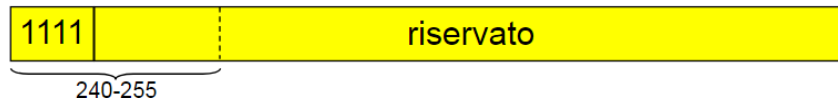
Classe D :

224.0.0.0 – 239.255.255.255



Classe E :

240.0.0.0 – 255.255.255.255



Livello 3- Classi di indirizzi

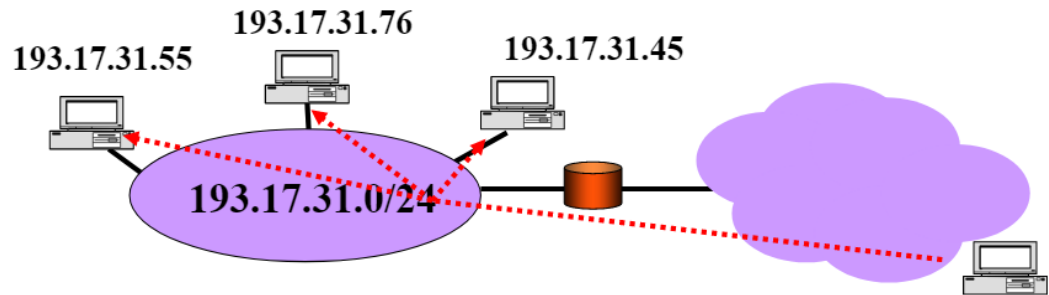
- Utilizzabili da chiunque ma solo in ambito privato.
- In Internet possono essere riutilizzati più volte ----- > Non sono univoci—Tre blocchi
 - 10.0.0.0 -10.255.255.255
 - 172.16.0.0 -172.31.255.255
 - 192.168.0.0 -192.168.255.255
- Un router non deve mai inoltrare un pacchetto con destinazione un indirizzo IP privato verso una propria interfaccia di uscita che abbia un indirizzo IP pubblico

Livello 3 - Netmask

- La netmask è un numero binario di 32 bit associato ad una rete IP
 - Inizia con n bit a 1 (dal più significativo), con n pari alla lunghezza del NetID
 - I restanti $32 - n$ bit sono a 0
 - Indica quali bit di un indirizzo IP sono assegnati al NetID
 - Viene indicata con dotted decimal notation
-
- Esempio:–Indirizzo IP 193.17.31.45 e Netmask: 255.255.255.0–La rete a cui appartiene l'indirizzo è 193.17.31.0

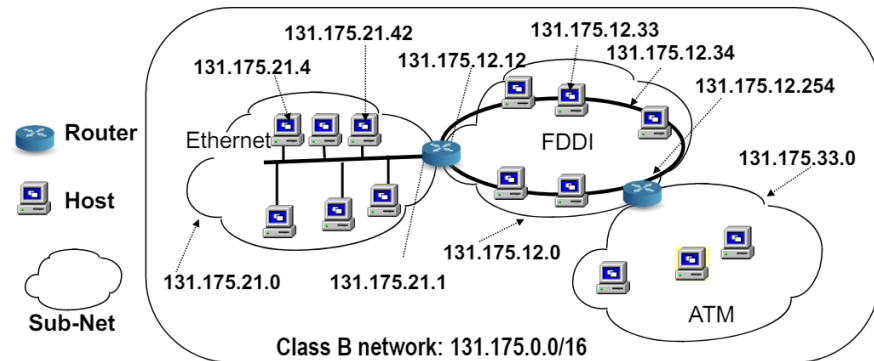
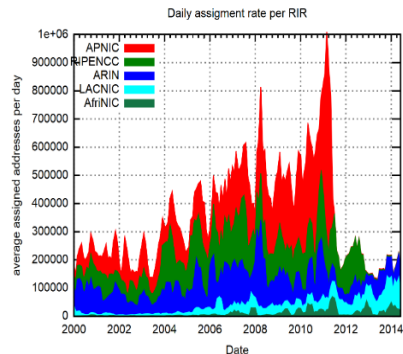
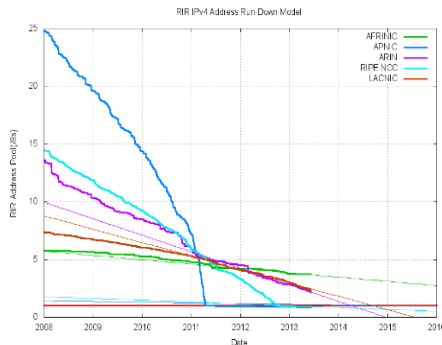
Livello 3 – Broadcast diretto

- Un indirizzo con il campo HostID di soli 1 assume il significato di indirizzo broadcast della rete indicata nel campo NetID (usato nel campo destinatario di un pacchetto IP)
- I router di transito lo trattano come un normale pacchetto (inoltrano al nexthop)
- Il router della rete di destinazione esegue il broadcast (a livello 2) solo se è abilitato a farlo
- esempio: 193.17.31.255



Livello -3 Gerarchia di reti

- Esaurimento IP v.4
- Introduzione IP v6
- Gerarchia delle reti





fondo
sociale europeo



in collaborazione con:

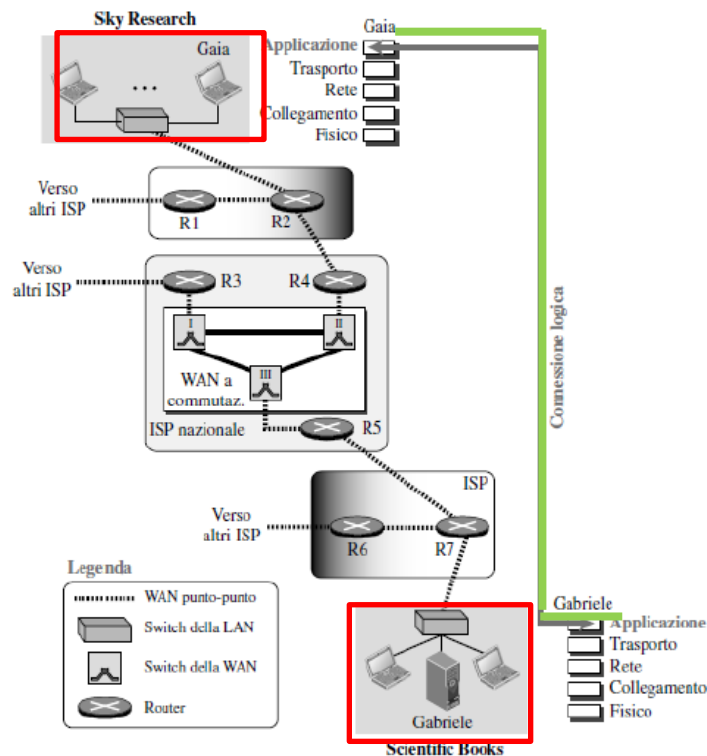


per una crescita intelligente,
sostenibile ed inclusiva

www.regione.piemonte.it/europa2020

INIZIATIVA CO-FINANZIATA CON FSE

- Il livello di Applicazione fornisce servizi alle applicazioni degli utenti della rete.
- La comunicazione è realizzata per mezzo di una:
- **connessione logica tra due identità logiche** (mittente e ricevente)
- Un programma/un'applicazione

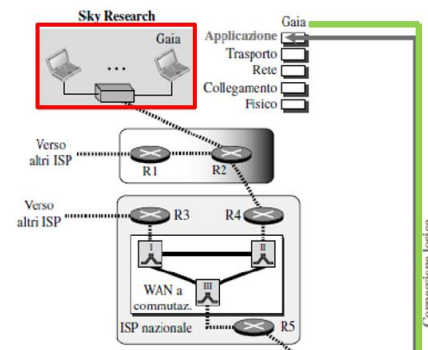


- **Un programma/un'applicazione** che vuole comunicare con un altro programma deve gestire i primi quattro livelli del modello ISO/OSI e quindi il livello applicazione dello stack +TCP/IP («aprire la connessione», inviare/ricevere dati e chiudere la connessione).
- Un insieme di istruzioni di questo tipo viene chiamato **API (Application Programming Interface)**: Le API sono set di definizioni e protocolli con i quali vengono realizzati e integrati software applicativi ----- > due processi (applicazione nel modello client/server) comunicano inviando e leggendo dati dal socket

- Applicazioni di Rete:
- Creare un'applicazione di rete significa scrivere un software che:
 - possa essere eseguito su **diversi terminali**
 - **possa comunicare tramite la rete**. EX: il browser web (FireFox, Safari, Chrome, ecc..) è un software “in esecuzione” su un dispositivo che comunica con un software in esecuzione su un server web (www.google.com, www.amazon.com, ecc..). Il linguaggio è quello esistente per la rete e nei nodi non c'è software applicativo
 - **Le applicazioni sono solo nei terminali** e possono essere facilmente sviluppate e diffuse

*****Applicazione*****

- Comunicazioni tra processi
- Host: dispositivo d'utente: Laptop, smartphone, desktop
- Processo: programma software in esecuzione su un host (molti processi possono essere in esecuzione simultaneamente sullo stesso host)
- Comunicazione inter-processo (IPC): tecnologie software il cui scopo è di consentire a diversi processi di **comunicare scambiandosi informazioni e dati**
 - Processi che risiedono sullo stesso host
 - **Processi che risiedono su host diversi (Serve una Rete!)**



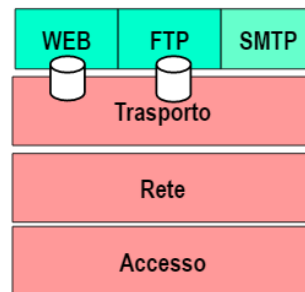
- Gli elementi fondamentali di un processo di comunicazione sono :
 - **Indirizzamento dei processi** (sapere chi sono gli interlocutori):
 - **INDIRIZZI IP**
 - **PORTA**
 - **Protocollo di comunicazione utilizzato per lo scambio dei dati:**
 - Tipi di messaggi scambiati: Richieste, risposte
 - Sintassi dei messaggi: Campi del messaggio e delimitatori
 - Semantica dei messaggi: Significato dei campi
 - Regole su come e quando inviare e ricevere i messaggi

- **Indirizzamento dei processi** (sapere chi sono gli interlocutori):

- **INDIRIZZI IP**
- **PORTA**
- **SOCKET = IP+PORTA**

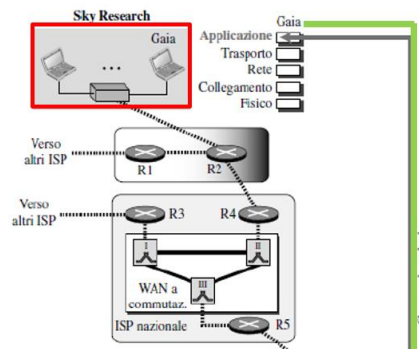
- Lo scambio di messaggi fra i processi applicativi avviene utilizzando **i servizi dei livelli inferiori attraverso i SAP (Service Access Point)** (vedi slide iniziali del corso)

• Ogni processo è associato ad un SAP



Livello controllato dall'applicazione

Livelli controllati dal sistema operativo



API e Socket

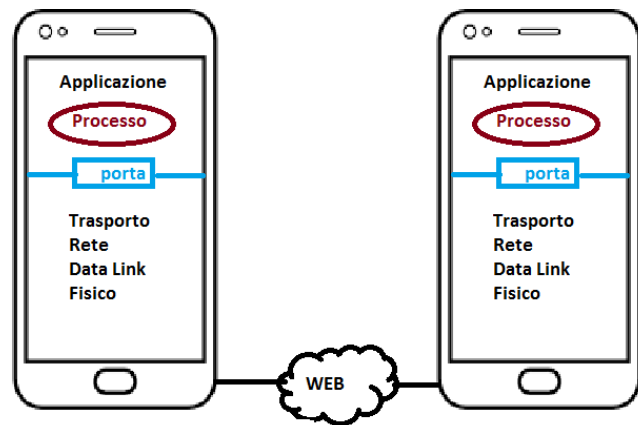
- I processi inviano e ricevono messaggi attraverso i socket
- Il socket è l'interfaccia (termine già incontrato ad inizio corso nel descrivere l'architettura a strati) tra il livello delle applicazioni e il livello di trasporto

- L'applicazione DEVE SPECIFICARE:
 - Numero di porta locale
 - –Indirizzo IP host locale
 - –Numero di porta remoto
 - –Indirizzo IP host remoto

Il SO deve specificare

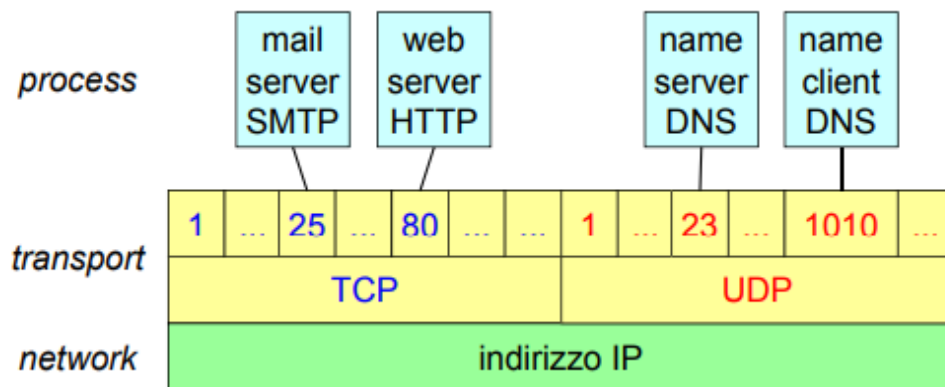
- Protocollo di Trasporto
- –Opzioni aggiuntive

- Le socket definiscono la comunicazione (vedremo più avanti come potrà essere la comunicazione ----> client/server, P2P, ibrida link □ e quale protocollo scegliere per la comunicazione link: □)
- Il processo trasmittente mette il messaggio fuori dalla
- La rete raccoglie il messaggio e lo trasporta fino alla porta del destinatario



Esempio : 142.125.7.9

Livello 4 ISO/OSI - Trasporto



- Porte: Modo per distinguere i dati
- MUX/DEMUX
- Le porte TCP e UDP sono identificate da un numero intero su 16 bit

Livello 4 ISO/OSI - Trasporto

- Le porte sono raggruppate in classi:
- le porte 0 : : 1023 sono dette porte privilegiate/note e sono usabili solo da processi di sistema lato server (HTTP,FTP,SMTP,DNS,...)
- I numeri dinamici da 1024 a 49151 sono assegnati dinamicamente ai processi applicativi lato client
- le porte da 49152 : : 65535 sono dette porte utente/registrate e sono assegnati a specifiche applicazioni da chi ne fa richiesta (tipicamente protocolli proprietari)

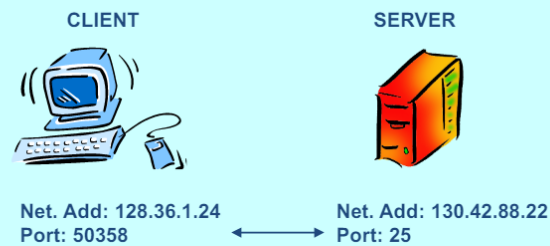


Livello 4 ISO/OSI - Trasporto

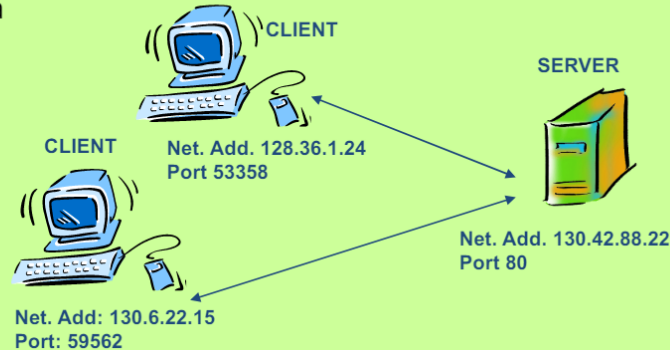
- Le porte si dividono inoltre in
- porte statiche quelle dove un server è in ascolto, ossia in attesa di richieste.
- porte dinamiche (anche dette porte effimere) sono quelle usate per completare una richiesta di connessione e svolgere un lavoro.

Multiplexing/Demultiplexing Socket

Un *client* trasmette segmenti verso la porta di un server SMTP remoto




Due client accedono alla stessa porta di un server HTTP.
Non c'è comunque ambiguità, perché la coppia di socket è diversa



Il livello 4: trasporto

- Il livello trasporto è un punto chiave della comunicazione
- Il livello di trasporto ha il compito di instaurare un collegamento logico tra le applicazioni residenti su host remoti
- Il livello di trasporto rende **trasparente il trasporto** fisico (attraverso la rete) dei messaggi alle applicazioni
- **Il livello di trasporto è presente solo negli end systems** (host)
- Esso consente il collegamento logico tra processi applicativi

Livello 4 ISO/OSI - Trasporto

- Il livello 4 provvede al trasferimento dei messaggi sulla rete procedendo in tre fasi:
 - 1. realizzazione della connessione
 - 2. trasferimento dei dati
 - 3. rimozione della connessione.
- La connessione (che deve essere **affidabile e duratura** per assicurare la corretta trasmissione dei dati e allo stesso tempo non durare più dello stretto necessario per evitare di congestionare la rete). Il livello di trasporto è preposto alla connessione logica tra due nodi di una rete
- La connessione è generalmente **dedicata** per tutta la durata della comunicazione (commutazione di circuito) ma il livello può dividere il messaggio in pacchetti da inoltrare con percorsi distinti (commutazione di pacchetto).  **Dipende dal tipo di protocollo utilizzato**

Livello 4 ISO/OSI – Trasporto – protocollo TCP

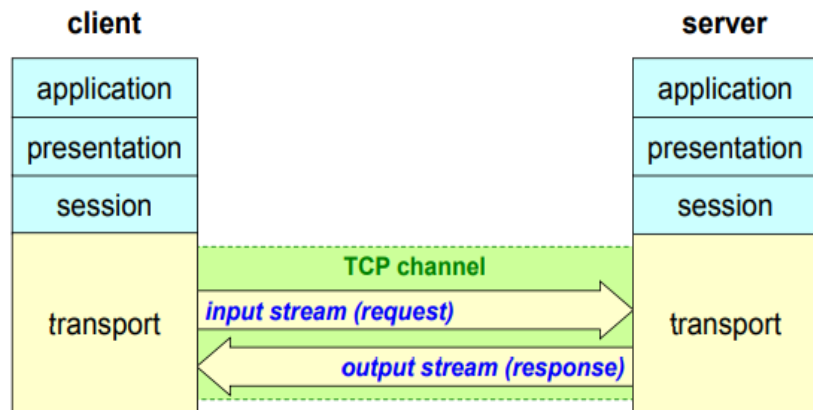
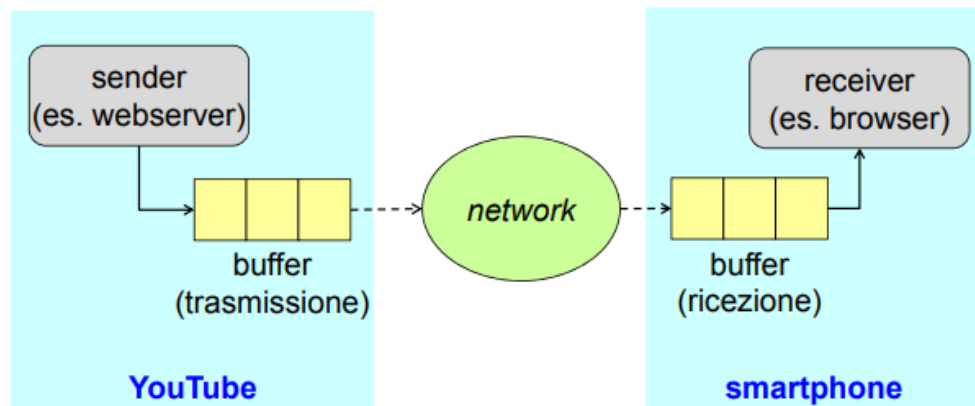


Figura 2.2: canale TCP.

- Trasmissioni simultanee
- Unità di trasmissione di 1 byte
- Buffering
- E' il più usato dalle applicazioni Internet (es. web, posta elettronica, trasferimento file).

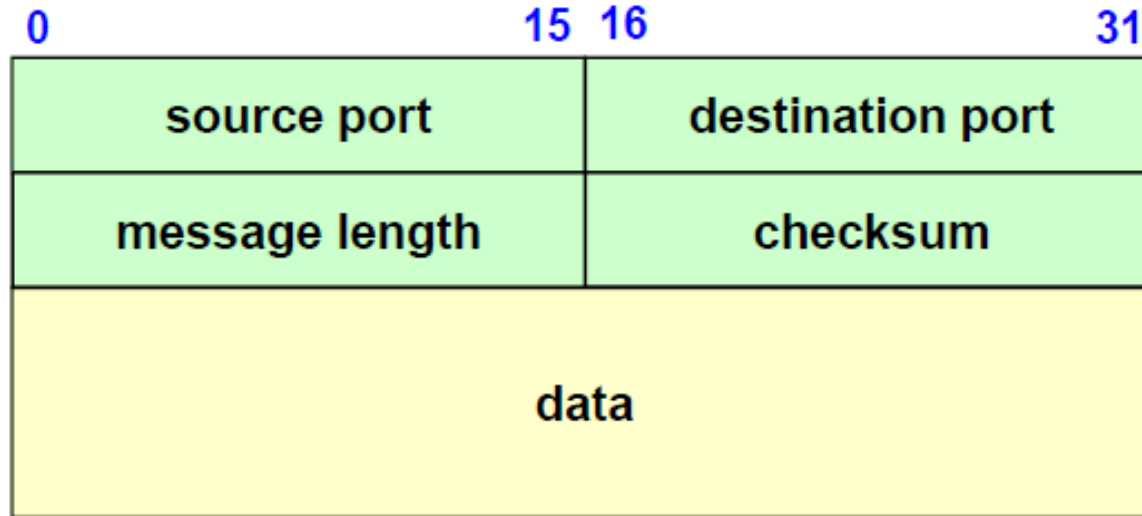
Livello 4 ISO/OSI – Trasporto – Protocollo UDP



- Non c'è il concetto di canale
- Comunicazione indipendente, scambio DI BYTES anche bidirezionale anche simultaneo
- Viene usato ad esempio per applicazioni DNS, NTP o nel gaming on-line

- UDP è un protocollo più semplice rispetto a TCP
- UDP rispetto al protocollo IP ci fornisce in più la possibilità di inviare messaggi di una dimensione maggiore di quella del semplice pacchetto IP
- UDP è un protocollo di trasporto orientato ai messaggi
- Il protocollo UDP aggiunge due funzionalità rispetto a quelle di IP: il multiplexing delle informazioni tra le varie applicazioni (tramite il concetto di porta) e la checksum (come a livello 2)

Datagram



- **E' utile usare UDP in quattro casi concettuali:**
- Se si opera su rete affidabile (esempio LAN) soprattutto quelle di tipo cablato oppure su una rete punto-punto.
- Se una singola PDU può contenere tutti i dati applicativi che si vogliono mandare, ossia se due applicazioni si stanno mandando dei dati di lunghezza inferiore a 64 kB.
- Se non importa che tutti i dati arrivino a destinazione esattamente
- Se è l'applicazione stessa che gestisce meccanismi di ritrasmissione

- Il principale problema di UDP è il controllo di congestione: UDP è connection-less e quindi non sa che cosa sta accadendo all'altro capo della rete.
- Per risolvere il problema del controllo di congestione era stato proposto il protocollo DDCP (Datagram Congestion Control Protocol) che però nella pratica non è mai stato utilizzato.

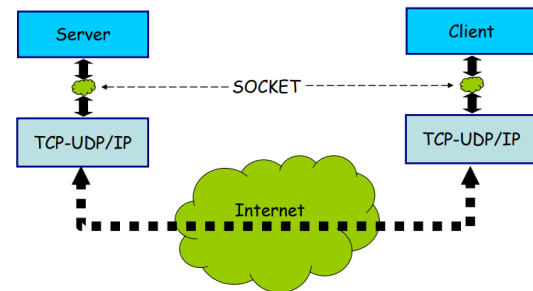
- **Le principali applicazioni sono:**
- DNS (Domain Name System)
- NFS (Network File System)
- SNMP (Simple Network Management Protocol)
- Molte applicazioni di streaming audio e video, in cui è importante la bassa latenza

- FTP (File Transfer Protocol) trasferimento file
- SMTP (Simple Mail Transfer Protocol) trasmissione e-mail
- HTTP (Hyper-Text Transfer Protocol) scambio dati tra browser e server web

- supporto della connessione tramite circuiti virtuali
- controllo di errore
- controllo di flusso:
- moltiplicazione e de-moltiplicazione
- controllo di stato e di sincronizzazione

Modello Client/server

- E' un modello astratto di comunicazione fra processi .Presuppone l'esistenza di:
 - ▣ 1 processo Client
 - ▣ 1 processo Server
- Schema di funzionamento:
 - ▣ Il Server inizialmente attende richieste di servizio
 - ▣ Il Client chiede servizio al Server
 - ▣ Il Server può accettare o rifiutare la richiesta
 - ▣ Se rifiuta, il Client può riprovare in seguito
 - ▣ Se accetta, è creata una sessione, avviene il trasferimento di informazione e infine la sessione è chiusa



- **Modello Client/server**
- I vantaggi di questa architettura sono la semplicità di realizzazione e la semplificazione del client.
- Gli svantaggi sono il sovraccarico del server, che deve elaborare richieste da più utenti e il conseguente sovraccarico del canale di comunicazione
- Ci sono diversi tipi di architetture C/S che si distinguono in base al numero di elementi che costituiscono il server. Una di queste è la client-server 2-tier. Si chiama 2-tier perchè, in inglese, tier significa strato o livello e questa architettura è appunto a due livelli



Agent per
svolgere compiti
specifici

Architettura C/S 3-tier.

- L'architettura C/S 2-tier è quella originale in cui il client interagisce direttamente con il server senza attori intermedi. E' un'architettura che può essere sfruttata su scala locale o geografica e viene usata in ambienti di piccole dimensioni, ovvero con un massimo di 50-100 client simultanei.
- Svantaggio: bassa scalabilità, poichè al crescere del numero di utenti decrescono le prestazioni del server proprio perchè viene sovraccaricato di richieste

- Configurazione C/S 3/tier
- **Adattatore, bilanciatore di carico** un sistema che si presenta come un server ai client ma non fornisce esso stesso direttamente la risposta: smista la richiesta al server **meno carico** tra tutti quelli per cui funge da interfaccia pubblica; altro esempio: può creare un cambio di protocollo per esempio da TCP a IP; altro esempio: cambio formato dati da ASCII a EBCDIC; altro esempio siti web come yahoo.it oppure amazon usano il bilanciatore per distribuire il carico.
- Quando il problema è migliorare le prestazioni di calcolo, tipicamente si adopera un bilanciatore di carico

- In un sistema C/S a tre livelli, il secondo livello è talvolta detto front-end perché è quello che si interfaccia direttamente con l'utente del servizio, mentre il terzo livello è detto back-end, perché è quello che fornisce la base del servizio pur restando nascosto nel retro (background o backstage in inglese).
- **Interfaccia utente/WEB:** Per sviluppare la parte client che si interfaccia con l'utente si può seguire l'approccio personalizzato (in inglese custom) oppure quello basato su browser web (oggi quello più diffuso)

- **l'interfaccia utente viene divisa in due parti:** quella che effettua le operazioni di input-output è basata sul browser web ed è quindi presente sulla postazione di lavoro mentre sul server si usano appositi linguaggi (come HTML) per definire tutti i campi che costituiscono l'interfaccia utente e controllarne il funzionamento.
- In questo schema, Il browser manda richieste HTTP e riceve risposte in HTML. Il server Web ha l'interfaccia descritta in HTML, **in particolare HTML statico**. Se il programma deve anche fare del lavoro, il server Web dialoga con il server applicativo per ricevere dei dati. Il server applicativo accede ai dati posizionati sulla stessa macchina o su un quarto livello e fornirà la parte dinamica dei dati. Quello che viene visualizzato da un utente è composto da dati statici e da dati dinamici

- **Modello P2P**
 - Non ci sono server sempre connessi
 - I terminali (peers) comunicano direttamente
 - I peers sono collegati in modo intermittente e possono cambiare indirizzo IP
- Esempio: BitTorrent

--Fortemente scalabile ma presenta molte difficoltà nella gestione

- L'architettura C/S è asimmetrica poichè il server svolge la maggior parte delle attività ed il suo ruolo è determinato a priori
- Invece l'architettura P2P è simmetrica perchè ogni nodo può ricoprire il ruolo sia di client sia quello di server, simultaneamente o in tempi diversi

- Si può utilizzare per esempio netstat
- Si utilizza linguaggio testuale
 - ▣ Ipconfig
 - ▣ Ping
 - ▣ Tracert
 - ▣ nslookup
 - ▣ Netstat -a -n

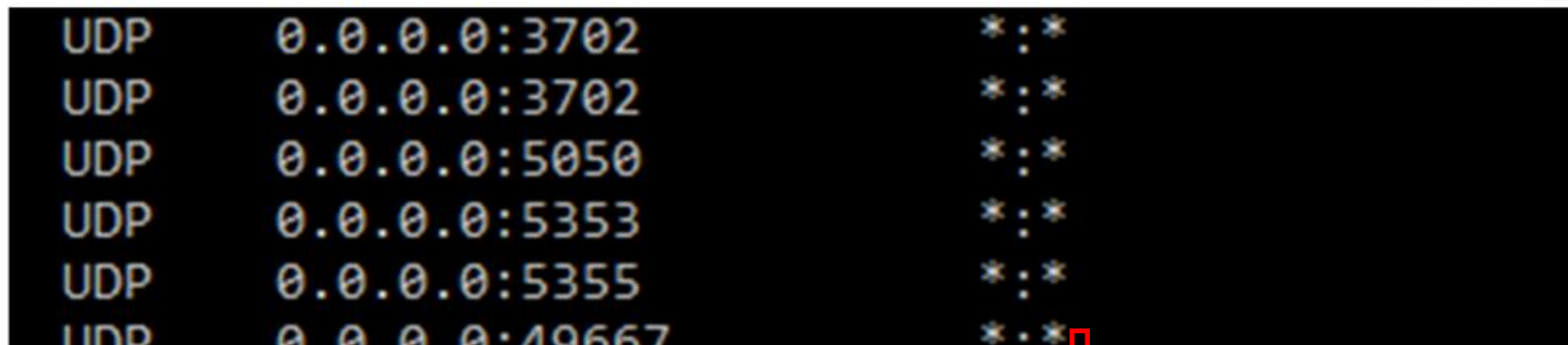
Proto	Indirizzo locale	Indirizzo esterno	Stato
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING

- Porte statiche dietro cui c'è un server in ascolto



```
TCP      [::]:135      [::]:0      LISTENING
TCP      [::]:445      [::]:0      LISTENING
TCP      [::]:49664    [::]:0      LISTENING
```

- 
- Nessuna informazione relativa all'IP v6



UDP	0.0.0.0:3702	*:*
UDP	0.0.0.0:3702	*:*
UDP	0.0.0.0:5050	*:*
UDP	0.0.0.0:5353	*:*
UDP	0.0.0.0:5355	*:*
UDP	0.0.0.0:49667	*:*

- la visualizzazione `*:*` o quella IPv6 `:::0` si verificano quando non è stata stabilita nessuna connessione con un host remoto. I servizi in ascolto sono pronti a soddisfare richieste di connessione provenienti da qualsiasi indirizzo e da qualsiasi porta remoti. Solo nel momento in cui verrà attivata una nuova connessione, verrà specificato un preciso indirizzo remoto.

- Aprire la pagina web libero.it
- Lanciare netstat -a -n
- Rispetto alla lista precedente ci sono più commessioni TCP poiché si sono aggiunte quelle generate dal browser per caricare la home page di libero.it.

TCP	0.0.0.0:49671	0.0.0.0:0	LISTENING
TCP	127.0.0.1:5939	0.0.0.0:0	LISTENING
TCP	127.0.0.1:23402	0.0.0.0:0	LISTENING
TCP	192.168.1.64:139	0.0.0.0:0	LISTENING
TCP	192.168.1.64:53838	40.67.254.36:443	ESTABLISHED
TCP	192.168.1.64:53912	95.101.114.233:443	CLOSE_WAIT
TCP	192.168.1.64:53915	87.248.107.204:443	ESTABLISHED
TCP	192.168.1.64:53916	151.101.2.2:443	ESTABLISHED
TCP	192.168.1.64:53917	23.50.144.67:443	CLOSE_WAIT
TCP	192.168.1.64:53918	95.101.114.41:443	CLOSE_WAIT
TCP	192.168.1.64:53919	95.101.114.41:443	CLOSE_WAIT
TCP	192.168.1.64:53920	95.101.114.41:443	CLOSE_WAIT
TCP	192.168.1.64:53921	95.101.114.41:443	CLOSE_WAIT
TCP	192.168.1.64:53922	95.101.114.41:443	CLOSE_WAIT
TCP	192.168.1.64:53923	95.101.114.41:443	CLOSE_WAIT
TCP	192.168.1.64:54150	52.157.234.37:443	ESTABLISHED
TCP	192.168.1.64:54152	13.107.21.200:443	ESTABLISHED
TCP	192.168.1.64:54153	13.107.18.11:443	ESTABLISHED
TCP	192.168.1.64:54154	13.107.6.254:443	ESTABLISHED
TCP	192.168.1.64:54155	13.107.215.10:443	ESTABLISHED

A causa del modo in cui TCP / IP funziona, le connessioni non possono essere chiuse immediatamente. I pacchetti possono arrivare fuori servizio o essere ritrasmessi dopo la chiusura della connessione. CLOSE_WAIT indica che l'endpoint remoto (l'altro lato della connessione) ha chiuso la connessione. TIME_WAIT indica che l'endpoint locale (questo lato) ha chiuso la connessione. La connessione viene mantenuta in modo che eventuali pacchetti ritardati possano essere abbinati alla connessione e gestiti in modo appropriato. Le connessioni verranno rimosse quando scadono entro quattro minuti