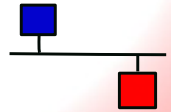
Emulatore di reti

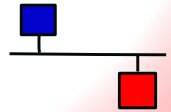
1 - Ripasso sulle reti IP

© Ing. Stefano Salvi – stefano@salvi.mn.it



Switch

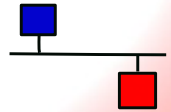
- Partiamo dalla **connessione di rete di basso livello**, quindi **Ethernet** ed il livello **Media Access Protocol (MAC)**
- In questo livello, **ogni scheda di rete (interfaccia)** è identificata da un **numero a 48 bit**, indicato con **sei coppie di cifre esadecimali**, solitamente separate da :, ad esempio **00:23:54:62:d8:4e**
- Questi indirizzi sono costituiti da due parti: **i primi 24 bit** indicano il **costruttore**, mentre i **rimanenti bit** sono il **numero progressivo** della scheda.
- Oltre ai normali **indirizzi MAC** ne esiste uno particolare, **FF:FF:FF:FF:FF:FF**, chiamato indirizzo di **broadcast**, che viene **ricevuto da tutte le schede** connesse.
- Ogni trama trasmesa conterrà **due indirizzi MAC**, uno sarà l'indirizzo della scheda stessa, il **mittente** e l'altro quello della o delle schede da raggiungere, il **destinatario**.
- Le **schede** di rete sono **connesse tra loro tramite** un apparato chiamato **switch**. Questo apparato **riceve** le trame da ogni scheda e lo inoltra alla scheda **destinataria**.
- Lo switch **impara** l'indirizzo **MAC** della o delle schede connesse ad una certa porta (presa) **analizzando l'indirizzo del mittente** delle trame che entrano da quella porta.
- Se uno switch **riceve** una trama destinata ad un **indirizzo che non conosce**, la **inoltra** su **tutte le porte**, per essere sicuro di raggiungere il destinatario.



© Ing. Stefano Salvi – stefano@salvi.mn.it

Connessione tra switch

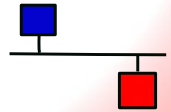
- **È possibile connettere** una porta di uno **switch** ad una porta di un **altro switch**.
- Di mano in mano **entrambi gli switch impareranno** gli indirizzi delle schede collegate all'altro switch, **analizzando** le trame che **entrano** dalla connessione tra gli switch.
- Questa **connessione tra switch** prende il nome di **trunk**.
- È anche possibile utilizzare **più cavi** per realizzare un trunk con maggire banda, creando un **channel**
- Utilizzando un trunk si può venire a creare un **problema** se viene trasmessa una **trama per una destinazione inesistente**.
- Se questa trama viene **inoltrata ad ogni porta**, verrà inviata **anche all'altro switch** tramite il trunk.
- A sua volta, **l'altro switch**, non conoscendo la porta di destinazione, **inoltrerà** la trama a **tutte le porte, compreso il trunk**.
- La trama verrà quindi **palleggiata tra i due switch** all'infinito **impegnando tutta la banda** del trunk.
- Per evitare questo basta **evitare di inoltrare** una trama per una destinazione ignota **verso la porta da cui la si è ricevuta**. Questa tecnica si chiama **split horizon**.



© Ing. Stefano Salvi – stefano@salvi.mn.it

VLAN

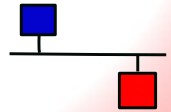
- Gli switch più sofisticati, detti **managed**, sono in grado di **raggruppare porte** in **sottoswitch indipendenti**, detti Virtual LAN o **VLAN**
- In questo modo è possibile creare **connessioni complesse**, per esempio all'interno di una sala server, **senza** necessità di utilizzare un **numero elevato di switch**.
- Naturalmente le **VLAN** sono viste come **switch indipendenti** all'interno dello stesso apparato
- Se dobbiamo **connettere** più **VLAN diverse** tra **due switch** dovremmo utilizzare **diversi trunk, uno per VLAN**.
- Questo può essere evitata utilizzando il **vlan tagging *IEEE 802.1Q***.
- Con questa tecnica **all'interno della trama** viene inserito un **codice di VLAN** che consente di far viaggiare **all' interno di un trunk, insieme ma separate**, le trame di ogni VLAN.
- Questo numero o **vlan tag** verrà poi **eliminato** quando le trame verranno inviate a **porte normali** (access) dello switch.



© Ing. Stefano Salvi – stefano@salvi.mn.it

Internet

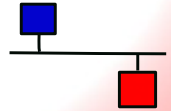
- Internet è una **rete di reti**
- Vale a dire che Internet si appoggia a **reti locali e connessioni esistenti**
- Le reti locali e occupano i **livelli 1 e 2** (*physical layer* e *data connection layer*) di **ISO-OSI**
- Quindi i protocolli di **Internet** si pongono ai **livelli 3** (*network layer*) **e superiori**



© Ing. Stefano Salvi – stefano@salvi.mn.it

livello IP

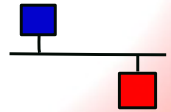
- **Internet Protocol** o **IP** è il primo livello di **Internet Protocol Suite** e si occupa di consentire ai pacchetti di rete di **raggiungere macchine in qualunque rete connessa**
- Si pone quindi a **livello 3 di ISO-OSI** o *network layer*
- Per svolgere questa funzione IP deve svolgere due **compiti**:
  - **Assegnare indirizzi** ad ogni interfaccia
  - **Gestire l'instradamento** dei pacchetti
- Per quanto riguarda l'**indirizzo**, a differenza di quello di livello 2 (MAC), oltre all'**identità dell'interfaccia**, deve indicare la **rete cui appartiene**.
- A differenza del livello 3 di ISO-OSI, **IP** è un **protocollo daragram**.



© Ing. Stefano Salvi – stefano@salvi.mn.it

Indirizzi IP - IPv4

- Gli indirizzi **IPv4** sono **numeri a 32 bit**, quindi il massimo teorico di indirizzi è **4.294.967.296**
- L'indirizzo è diviso in **due parti**:
  - Indirizzo della rete o **net**
  - Indirizzo della macchina nella rete o **host**
- i bit **piu significativi** del numero indicano la **rete** mentre i **meno significativi** la **macchina** all'interno della rete
- L'indirizzo viene **scritto** suddiviso nei **byte** che lo compongono.
- Ogni **byte** viene scritto in **decimale**, separato con un **punto** dal successivo.
- adesempio, l'indirizzo **00001010 00000000 11001000 01101000** viene scritto come **10.0.200.104**

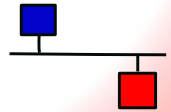


© Ing. Stefano Salvi – stefano@salvi.mn.it

Classi

- La **dimensione** del **campo net** non è fissa, ma **dipende dalla classe**.
- esistono cinque classi, di cui tre utilizzate per le normali reti:

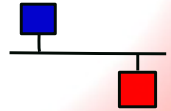
Classe	Net	Host	Formato	Intervallo	Esempio
A	8	24	0XXXXXXX.	0 .. 127	80.0.0.0
B	16	16	10XXXXXX.XXXXXXXX. ...	128 .. 191	190.220.0.0
C	24	8	110XXXXX.XXXXXXXX.XXXXXXX ...	192 .. 223	194.189.96.0
D	0	32	1110XXXX.XXXXXXXX. ...	224 .. 239	230.19.16.22
E	0	32	1111XXXX.XXXXXXXX. ...	240 .. 255	242.13.12.24



© Ing. Stefano Salvi – stefano@salvi.mn.it

Sottoreti

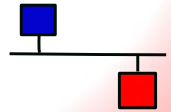
- Gli **enti internazionali** (ICANN) o **nazionali** (GARR) assegnano a grossi privati o a provider **reti IP** in base alle **classi** definite precedentemente
- Se ci viene assegnata una **rete IP** e dovessimo utilizzarla, probabilmente la troveremmo **troppo grande** e vorremmo **suddividerla** in parti.
- Per fare questo dovremmo **assegnare al campo *net*** alcuni bit del **campo *host***.
- Se, ad esempio, volessimo **suddividere in due parti** la rete di **classe C 192.168.2.0** potremmo mettere gli host dal **192.168.2.0** al **192.168.2.127** in una sottorete e quelli dal **192.168.2.128** al **192.168.2.255** nell'altra
- Questo corrisponde ad assegnare al **bit più significativo** del campo ***host*** il compito di **discernere** tra le **nuove sottoreti**.



© Ing. Stefano Salvi – stefano@salvi.mn.it

Notazione CIDR

- Se definiamo in questo modo le sottoreti, la **dimensione del campo *net*** completo (comprensivo anche della parte usata per le sottoreti) **non è più determinabile** immediatamente dalla **classe**
- Occorre **precisare** quindi in **maniera esplicita** la **dimensione del campo net** di una rete/sottorete.
- Un modo semplice è quello di indicare il **numero di bit** che fanno parte del **campo *net*** dopo una ***/***.
- Il nome di questa **notazione** è **CIDR** che indica ***Classless Inter-Domain Routing***.
- Nel nostro caso le due **nuove sottoreti** si indicheranno quindi **192.168.2.0/25** e **192.168.2.128/25** rispettivamente
- Notiamo che l'indirizzo che contiene il **numero di host 0** viene utilizzato per indicare l'i**ntera rete** e normalmente **non viene associato ad alcun host reale**, per evitare confusione.
- Naturalmente in questa notazione dovremo aggiungere:
  - **/8** alle reti di **classe A**
  - **/16** alle reti di **classe B**
  - **/24** alle reti di **classe C**
- Ricordiamo naturalmente se abbiamo indicato **24 bit di *net*** non è detto che abbiamo a che fare con una **rete di classe C**, ma magari una **sottorete** con 8 bit di host di una **rete di classe B o A**.



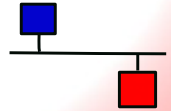
© Ing. Stefano Salvi – stefano@salvi.mn.it

Netmask

- La notazione CIDR è estremamente compatta e **facile** da usare **per no**i, ma **non è comoda per il computer**.
- Per il computer è molto più semplice utilizzare l'**operazione di and bit a bit** indicata con **&** (semplice, non doppio) in **C** o dal opcode come **`AND`** o **`ANL`** in **Assembler**.
- Se ad esempio abbiamo l'indirizzo **192.168.2.160/25** della nostra subnet di esempio possiamo fare fare un'operazione di **and** con un **numero** che abbia tutti **i bit corrispondenti al campo *net* ad 1** e tutti quelli del **campo *host* a 0**, quindi **15 bit a 1** seguiti da **7 a 0**.
- Convertiamo in binario l'indirizzo IP e facciamo l'operazione:

```
11000000 10101000 00000010 10100000 &
11111111 11111111 11111111 10000000 =
11000000 10101000 00000010 10000000 .
```

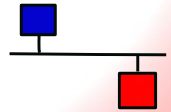
- Che, riconvertito in decimale come indirizzo IP risulta **192.168.2.128**, quindi il nostro IP **appartiene alla seconda sottorete**.
- Il numero che abbiamo utilizzato prende il nome di **netmask**, e viene normalmente scritto nella forma degli indirizzi IP, quindi **255.255.255.128**



© Ing. Stefano Salvi – stefano@salvi.mn.it

Protocollo ARP

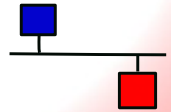
- Abbiamo visto come sono costruiti gli indirizzi IP, ma **Internet Protocol**, per **inviare pacchetti** ad altre macchine della stessa rete locale deve **appoggiarsi sulla rete locale sottostante**, che non conosce nulla di indirizzi IP
- Per **inviare un pacchetto** (o meglio una *trama*) ad una macchina **tramite una rete** locale dobbiamo **indicare l'indirizzo *MAC*** (indirizzo fisico) della macchina di **destinazione**.
- Il **pacchetto da trasmettere** contiene però esclusivamente l'**indirizzo *IP* del destinatario**
- E' quindi indispensabile trovare un **metodo** per **trasformare** un **indirizzo *IP*** nel **corrispondente indirizzo *MAC***.
- Questo metodo consiste nell' **Address Resolution Protocol** o **ARP**.
- Questo protocollo consiste nell'inviare un **pacchetto ARP di richiesta in broadcst** a tutte le schede presenti sulla rete, contente l'indirizzo IP da trovare.
- Il **protocollo ARP della macchina** che possiede **quell'indirizzo**, se esiste, **risponderà** con un pacchetto **unicast** contente tra l'altro il **suo *MAC* address**.
- Il protocollo ARP **memorizzerà per un certo tempo** l'associazione tra IP e MAC, in modo da non dover inviare pacchetti in broadcast ogni volta che si deve inviare un pacchetto IP, raddoppiando il traffico di rete



© Ing. Stefano Salvi – stefano@salvi.mn.it

Routing

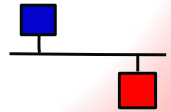
- Abbiamo visto che Internet è una **rete di reti**, quindi quando **inviamo un pacchetto** ci possono essere due casi:
  - la **destinazione** appartiene ad una delle reti cui siamo **direttamente collegati** tramite una scheda di rete (**locale**)
  - la **destinazione** appartiene ad una rete **raggiunhgibile tramite un ruoter** presente in una delle reti cui siamo connessi (**remoto**)
- Nel **primo caso**, come abbiamo visto, la nostra macchina **individuerà l'indirizzo MAC** della destinazione tramite il protocollo ***ARP***, quindi **invierà** il pacchetto **direttamente** alla macchina destinataria.
- Nel **secondo caso** invece dovrà **individuare l'indirizzo *MAC* del router** tramite ***ARP*** ed **invierà ad esso** il pacchetto contenete l'**indirizzo IP del destinatario finale**.
- A sua volta **il router**, ricevendo un **pacchetto del quale non è il destinatario**, eseguirà lo **stesso procedimento**, inviando il pacchetto a destinazione o al successivo router.
- Dato che **IP** è un **protocollo Datagram**, il procedimento descritto viene fatto **per ogni pacchetto inviato**, quindi sia per le richieste che per le risposte.



© Ing. Stefano Salvi – stefano@salvi.mn.it

Tabella di routing

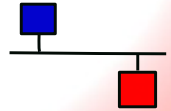
- Naturalmente questi ragionamenti hanno senso se è possibile conoscere quali sono le **reti locali**, le **reti remote** ed i **router (*gateway*)** per raggiungerle.
- Queste **informazioni** sono necessarie allo stesso modo **sia** per i router che per i normali **host**
- Per scoprirle si deve utilizzare una **tabella di routing**, che ha tre colonne principlali:
  - **rete** (in notazione **CIDR** o come **indirizzo e maschera**)
  - **gateway** (IP del router)
  - **interfaccia** (nome della scheda di rete nella macchina)
- Per prime verranno elencate le **reti direttamente connesse**. Esse avranno **0** (o **0.0.0.0**) nel campo **gateway**.
- Successivamente verranno elencate le altre reti note, indicando nel campo gateway **l'indirizzo del router** da utilizzare. Questo **gaeway** dovrà essere **raggiungibile direttamente** tramite l'interfaccia indicata nella riga.
- Naturalmente, se dovessimo indicare individualmente in questo modo **tutte le reti raggiungibili** tramite Internet, la tabella sarebbe **infinita**.
- D'altronde normalmente la connessione ad Internet avviene solitamente tramite **una sola connessione**, quindi **un solo gateway**.
- Potremo quindi indicare questo come il ***default gateway***, vale a dire il gateway cui inviare **tutti i pacchetti per i quali non conosciamo altrimenti la rete**.
- La **rete** del default gateway viene indicata come **0/0** (o **0.0.0.0/0**). Se applichiamo l'operazione di AND indicata precedentemente vediamo che questa rete **corrisponde a qualunque indirizzo IP**.



© Ing. Stefano Salvi – stefano@salvi.mn.it

ICMP

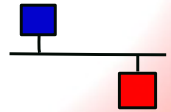
- Nel caso si stia inviando un pacchetto **ad una macchina inesistente** o spenta della nostra rete, lo scopriremo immediatamente in quanto **non avremo risposta ad *ARP***, e quindi il programma riceverà un **errore**.
- Se invece lo stiamo inviando ad una **macchina inesistente** di una **rete remota** o ad una **subnet inesistente** di una rete remota, ***ARP*** non ci avvisa dell'errore in quanto il router viene trovato.
- Per risolvere questa situazione, oltre ad altre, viene introdotto **Internet Control Message Protocol** (**ICMP**).
- Il router che non riesce ad inoltrare il pacchetto **risponderà con un pacchetto ICMP** del tipo ***destination unreachable*** che ci indicherà se la rete o l'host sono irraggiungibili
- Esistono anche **altri tipi** di messaggio, per indicare **errori** oppure per **informazioni di servizio**. Alcuni li vedremo in seguito.
- Sebbene **ICMP** sia utilizzato anche da IP, di livello 3, **è un protocollo di livello 4**.



© Ing. Stefano Salvi – stefano@salvi.mn.it

Echo e PING

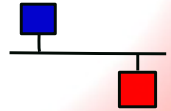
- Un utile **messaggio ICMP** è l`**echo** con la relativa risposta **echo reply**.
- Alla **ricezione** di un pacchetto ICMP **echo**, il protocollo **ICMP risponde** inviando un pacchetto **echo reply** con lo stesso contenuto.
- Questo consente di creare il **comando diagnostico ping**.
- Questo comando **invia uno o più pacchetti *echo*** all'host che indichiamo e **misura il tempo** che passa per ricevere la risposta, **se arriva**.
- Con questo comando si può in primo luogo **verificare se esiste la connettività** sia per raggiungere la destinazione che per il ritorno.
- Si può quindi verificare, se si fanno inviare più pacchetti, se la **connettività è stabile** o se vengono persi dei pacchetti.
- Per finire si può verificare la **velocità** del collegamento, controllando il **tempo della risposta**.



© Ing. Stefano Salvi – stefano@salvi.mn.it

Routing Loop e TTL

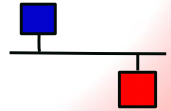
- Se inviamo un pacchetto ad una **rete esistente** ed i **router** sono **configurati correttamente**, non ci saranno problemi.
- In alcuni casi, magari per semplificare le regole di routing, è possibile che i **default gateway** facciano un **anello**, vale a dire, l'ultimo gateway rimanda il pacchetto al al primo.
- In questo caso, se si **invia** un pacchetto ad una **rete o subnet inesistente**, verrebbe continuamente **rimbalzato** tra i router dell'anello **occupando tutta la banda** disponibile.
- Per evitare questo viene inserito nella **intestazione IP** un campo, chiamato **Time To Live** o **TTL**, che viene **decrementato all'attraversamento di ogni router**.
- **Normalmente** questo campo è impostato ad un **valore tale** da riuscire a **raggiungere qualunque destinazione** (ha un massimo di 255)
- Se però il pacchetto viene **rimbalzato** tra i router, in breve questo campo **raggiungerà il valore 0**
- Quando un *router*, decrementando il TTL, **arriva 0**, scarta il pacchetto ed **invia al mittente un pacchetto ICMP time exceeded.**



© Ing. Stefano Salvi – stefano@salvi.mn.it

Traceroute

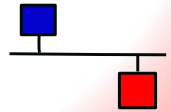
- Il meccanismo del TTL ed dell'ICMP time exceeded consente di realizzare un **interessante strumento diagnostico**
- Il comando **traceroute** è in grado di **individuare i router** che il nostro pacchetto attraversa per giungere a destinazione.
- Per questo **traceroute** invia una **serie di pacchetti** con ***TTL* crescente** a partire da 1.
- Il **primo pacchetto** verrà **scartato dal primo router**, che invierà un **ICMP time exceeded** rivelando il proprio IP.
- I **successivi pacchetti** individueranno i **successivi router** fino a quando non arriverà un errore dall'**host di destinazione**.



© Ing. Stefano Salvi – stefano@salvi.mn.it

IPv6

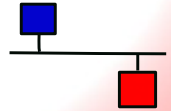
- Le **reti disponibili in IPv4** a livello mondiale sono state **esaurite** a febbraio 2011.
- Per fortuna da parecchi anni è disponibile la **prossima versione** di IP, **IPv6**.
- Questa nuova versione, oltre a **nuove caratteristiche e semplificazioni varie**, possiede un **indirizzo a 128 bit**.
- Gli indirizzi IPv6 vengono indicati divisi in **8 gruppi** di cifre **esadecimali**, ognuno rappresentante **16 bit, separati da :**
- Gli **0 non significativi** di ogni gruppo possono venire **omessi**. Una **sequenza** (ed una sola) **di gruppi a 0** può essere **sostituita da ::**
- Ad esempio, **localhost** (127.0.0.1 in IPv4) corrisponde all'**indirizzo 1 a 128 bit**, che **si scrive *::1***
- Un **indirizzo** di una scheda **reale** invece potrebbe essere **fe80::223:54ff:fe62:d84e**



© Ing. Stefano Salvi – stefano@salvi.mn.it

IPv6 - Compatibilità

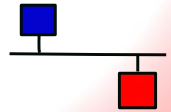
- **Gli indirizzi IPv6 ed IPv4 sono incompatibili** e quindi se la nostra macchina **non è** configurata per IPv6 **non possiamo raggiungere macchine con indirizzi IPv6** e viceversa.
- Dato che attualmente **la situazione è mista**, con una parte della rete con indirizzi IPv4 ed un'altra con indirizzi IPv6, la maggior parte delle macchine possiede **due stack IP**: uno IPv4 ed uno IPv6.
- In questo modo, **se tutto è configurato correttamente**, ogni scheda possiede almeno **due indirizzi: uno IPv4 ed uno IPv6**, potrà quindi comunicare con ogni tipo di macchina.
- Occorre però che **anche la rete sia configurata** per trasferire pacchetti di **entrambi i protocolli**, altrimenti non potremo trovare rotte per la destinazione.
- Se così non è, è possibile configurare dei **tunnel** che trasferiscano **pacchetti di un protocollo inseriti in pacchetti dell'altro**.



© Ing. Stefano Salvi – stefano@salvi.mn.it

TCP ed UDP

- Il protocollo IP, di livello 3, consente la **comunicazione tra macchine**, ma non di far comunicare specifici **programmi all'interno di una macchina**.
- Inoltre è un protocollo **datagram**, quindi **non garantisce** la **correttezza e completezza** delle informazioni ricevute, cosa che la maggior parte delle applicazioni richiede.
- Per risolvere il **primo problema**, viene introdotto il concetto di **porta**, che è un **numero a 16 bit** che identifica una **comunicazione**, quindi i **programmi che comunicano**
- Siccome **non tutti** i protocolli **richiedono** una **comunicazione confermata a connessa** (sicura e completa), vengono introdotti **due diversi protocolli di livello 4**.
- Per primo, **User Datagram Protocol**, **UDP**, si limita ad **introdurre la porta del mittente e quella del destinatario**, consentendo a **più programmi** di comunicare contemporaneamente in maniera datagram.
- Viene inoltre introdotto **Transfer Control Protocol**, **TCP**, che gestisce **connessioni prive di errori**, sempre utilizzando una **porta mittente ed una destinataria** e quindi più programmi.
- I programmi che desiderano inviare o ricevere dati si devono **registrare nello stack** indicando il tipo di **protocollo** e la **porta** cui si associeranno.
- In questo modo lo **stack** potrà **contrassegnare** i **pacchetti uscenti** con la porta scelta ed **inviare al giusto programma i paccheti entranti** in base al protocollo (TCP o UDP) e alla porta.
- Essendo **TCP ed UDP due protocolli diversi**, un numero di porta associato ad **uno dei due** identifica un **diverso interlocutore** rispetto allo **stesso numero**, associato all'**altro protocollo** (la porta 200 TCP va as un diverso programma dalla porta 200 UDP).

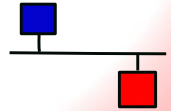


© Ing. Stefano Salvi – stefano@salvi.mn.it

Servizi base

- **TCP** ed **UDP** consentono di comunicare, ma **non entrano nel merito** della comunicazione.
- **Sopra il livello 4**, Internet Protocol Suite prevede un solo ulteriore livello, il **livello applicativo**.
- In Internet, i server dei vari **protocolli** di livello applicativo possono essere **associati a qualunque porta**, ma per semplicità vengono spesso associati ad dei **port standard** chiamati ***well known ports***.
- Non possiamo entrare in questa sede nel dettaglio dei vari **protocolli**, ma **elenchiamo i principali**, con i relativi ***well known port***:

Port	Protocollo	Sigla	Funzione
21	File Transfer Protocol	FTP	trasfermiento file
22	Secure Shell	SSH	Terminale Remoto Sicuro (anche trasferimento file e port forwarding)
23	Telnet	Telnet	Terminale remoto base
25	Simple Mail Transfer Protocol	SMTP	Trasmissione mail
53	Domain Name System	DNS	Ricerca degli IP in base al nome
80	HyperText Transfer Protocol	HTTP	Trasferimento contenuti WEB
110	Post Office Protocol v3	POP3	Recupero mail da casella di posta remota
143-220	Internet Message Access Protocol	IMAP	Gestione completa casella di posta remota
443	HyperText Transfer Protocol Secure	HTTPS	Trasferimento contenuti WEB in maniera sicura
520	Routing Information Protocol	RIP	Compila automaticamente le tabelle di routing



© Ing. Stefano Salvi – stefano@salvi.mn.it