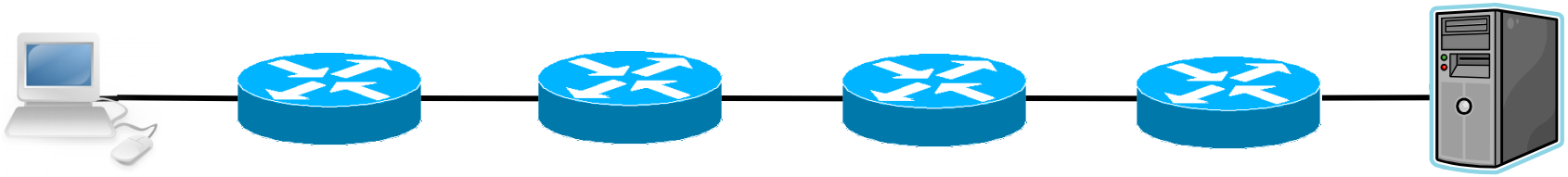
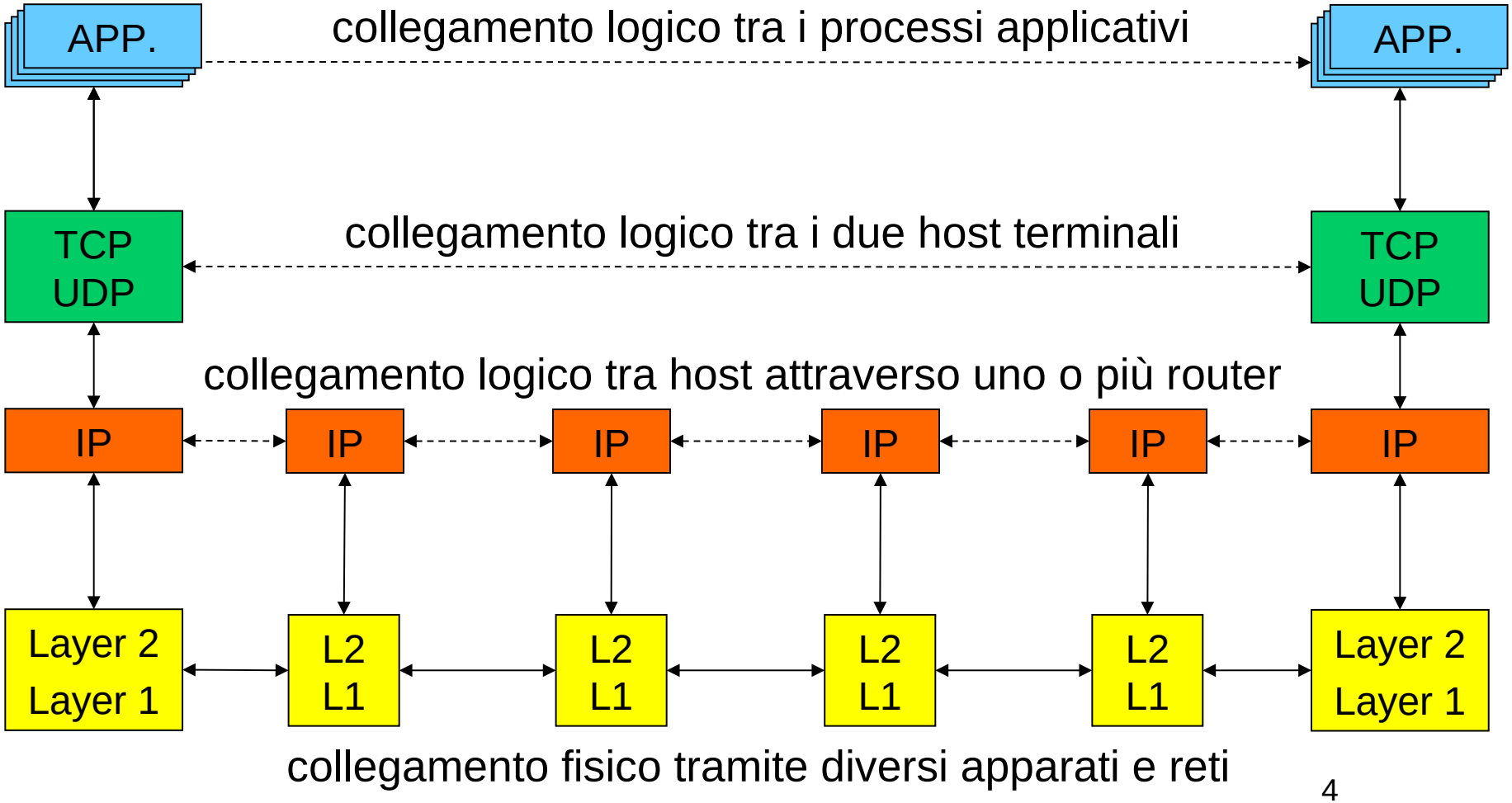
Lo strato di trasporto
Firewall e NAT

A.A. 2018/2019

Walter Cerroni

Il livello di trasporto in Internet





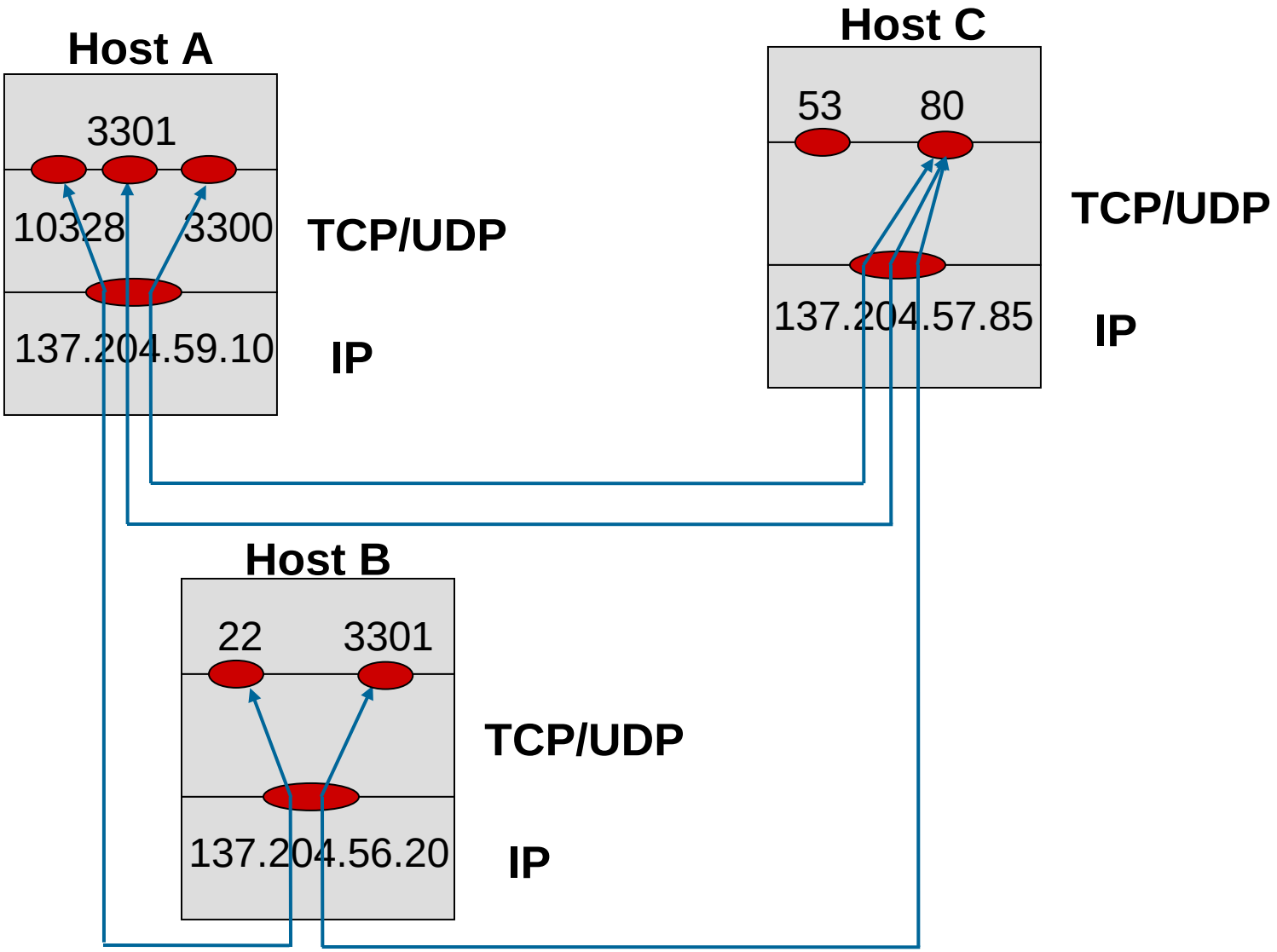
Funzioni dello strato di trasporto

- Compito dello strato di trasporto è fornire un servizio di trasporto dati tra i processi applicativi di un host sorgente e quelli di un host destinazione, svincolando gli strati superiori da tutti i problemi di rete
  - realizza una comunicazione **end-to-end**
  - rappresenta l'interfaccia fra gli strati superiori e lo strato di rete
    - l'interazione avviene attraverso un punto di accesso al servizio (**T-SAP**) che negli standard di Internet è chiamato **porta**
    - tipicamente più processi possono utilizzare le funzionalità dello stesso strato di trasporto contemporaneamente (**multiplazione**)
- Può funzionare in modalità **connectionless** e **connection-oriented**
- Protocolli principali dello strato di trasporto di Internet:
  - **UDP**: modalità **connectionless, non affidabile**
  - **TCP**: modalità **connection-oriented, affidabile**

Multiplazione/Demultiplazione

- Molteplici processi applicativi in esecuzione sullo stesso host possono aver bisogno contemporaneamente di comunicare tramite lo strato di trasporto
- Il protocollo di trasporto deve poter:
  - raccogliere i dati provenienti da applicazioni diverse e trasmetterli attraverso un unico strato di rete (**multiplexing**)
  - ricevere i dati dallo strato di rete e smistarli correttamente verso le diverse applicazioni (**demultiplexing**)
- **Numero di porta**: è un identificativo che determina univocamente un particolare processo applicativo in esecuzione su un host e che sta utilizzando il protocollo di trasporto
- **Socket**: è l'interfaccia (software) attraverso cui il livello di trasporto scambia dati con le applicazioni

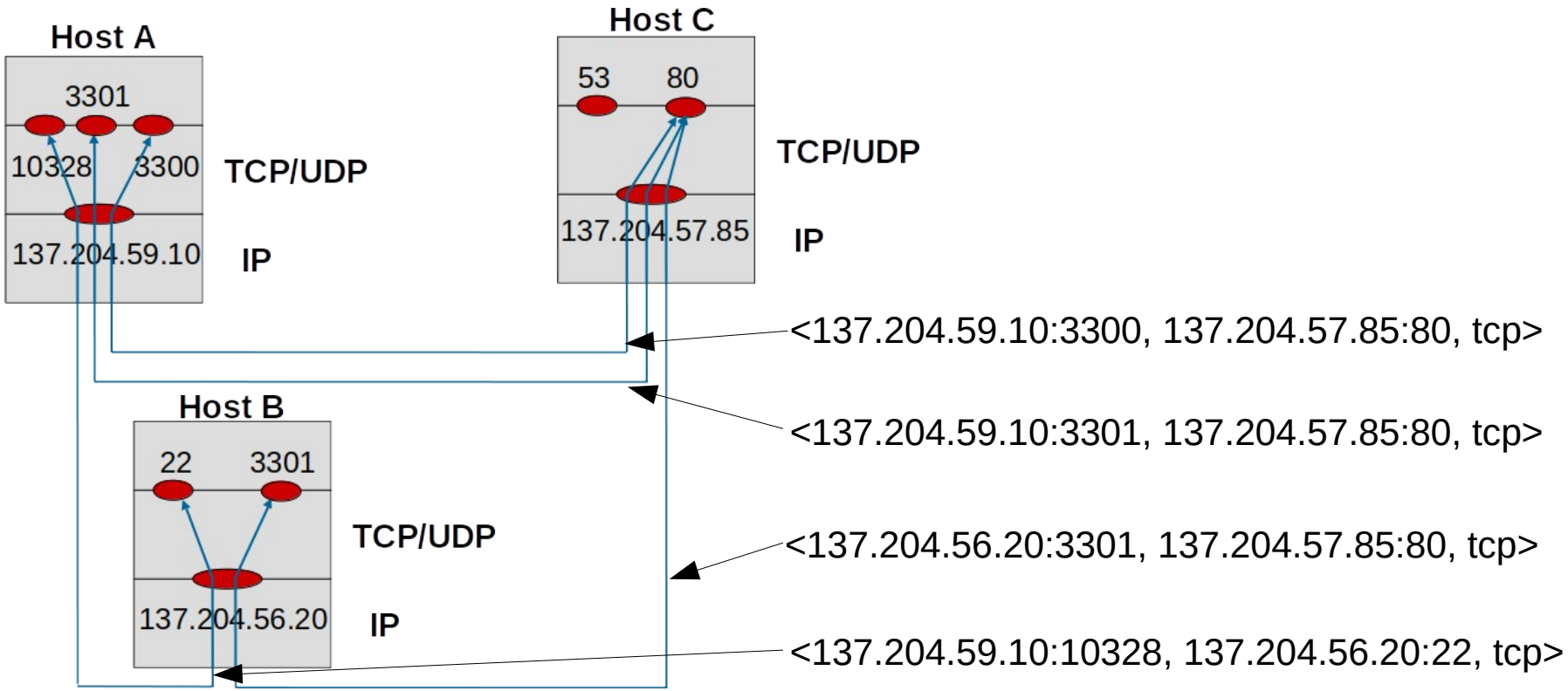
Multiplazione/Demultiplazione



Concetto di flusso

- Un **flusso** bidirezionale di dati scambiati tra applicazioni può esere identificato attraverso indirizzi e porte TCP/UDP:

  <IP-sorg:porta-sorg, IP-dest:porta-dest, protocollo>



Processi client e processi server

- Le comunicazioni fra calcolatori sono originate da processi applicativi che devono scambiare messaggi con altre applicazioni remote
- Perché un calcolatore possa effettivamente utilizzare un messaggio che arriva, occorre che in quel momento vi sia in esecuzione un processo che vada a leggere quel messaggio e sappia cosa farne
- La soluzione più frequente in Internet è l'utilizzo di applicazioni distribuite basate sul **modello client-server**
- Ci sono calcolatori su cui girano **processi Server** che erogano servizi e si aspettano di ricevere richieste di connessione da parte di **processi Client** interessati a tali servizi

Modello di servizio

- Il processo Server si predispone a ricevere una richiesta eseguendo una **apertura passiva**
  - UDP: apre una socket e si mette in ascolto sulla relativa porta, in attesa dell'arrivo di un *datagramma contenente dati applicativi*
  - TCP: apre una socket e si mette in ascolto sulla relativa porta, in attesa dell'arrivo di una *richiesta di apertura della connessione*
  - il processo server, tipicamente eseguito in background, nel mondo Linux-UNIX è chiamato **demone**
- Quando è interessato ad un determinato servizio, il processo Client esegue una **apertura attiva** tentando di collegarsi al processo server che offre tale servizio
  - UDP: apre una socket inviando direttamente un *datagramma contenente dati applicativi*
  - TCP: apre una socket inviando una *richiesta di apertura della connessione*
- Il client deve conoscere l'indirizzo IP e il numero di porta usati dal server per potersi collegare alla socket di destinazione

Scelta delle porte

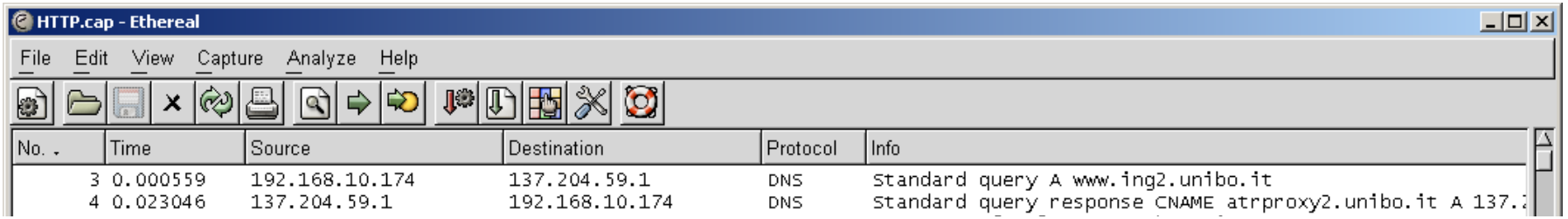
- L'indirizzo IP dell'host su cui è in esecuzione il server è:
  - inserito direttamente dall'utente
  - ottenuto tramite traduzione di un nome DNS
- Il numero di porta su cui inviare le richieste al server è scelto secondo una convenzione:
  - tutti i server di un certo tipo devono utilizzare un numero di porta definito a priori tra le cosiddette **well-known port**
    - es.: i server HTTP usano la porta TCP 80
  - l'elenco delle porte note è definito dalla **IANA** ed è reperibile su **www.iana.org** o, su sistemi Linux-UNIX, nel file **/etc/services**
  - server che non utilizzano porte note risultano "nascosti" e sono raggiungibili solo se se ne conosce l'URL completo
    - es.: http://www.nascosto.unibo.it:8088
- Un client, quando apre una socket, non dovrebbe usare localmente una porta nota

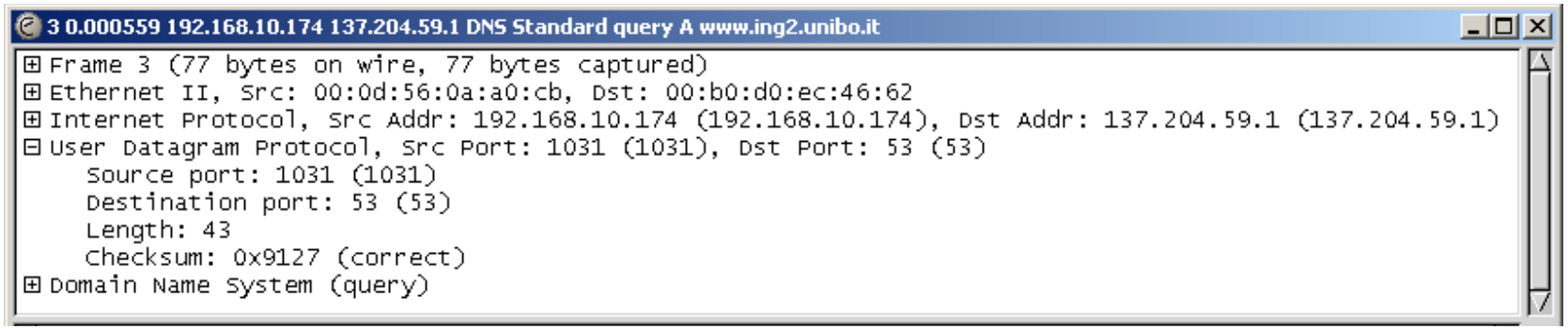
Well-known port

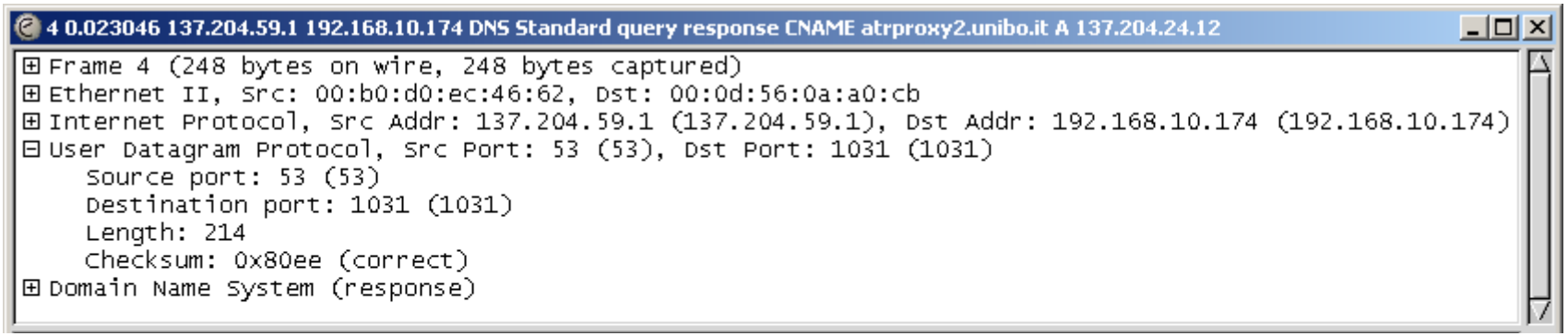
- Numeri di porta TCP/UDP a 16 bit:
  - da **1** a **1023**: porte **well-known**
    - possono essere usati solo dai server in apertura passiva
  - da **1024** a **49151**: porte **registrate**
    - sono usati da alcuni servizi ma anche dai client
  - da **49152** a **65535**: porte **dinamiche**
    - sono usati dai client

Numero		Nome	Tipo di servizio
21	TCP	FTP	trasferimento file
22	TCP	SSH	terminale virtuale criptato
23	TCP	TELNET	terminale virtuale in chiaro
25	TCP	SMTP	invio posta elettronica
53	UDP	DOMAIN	server DNS
80	TCP	HTTP	server web
110	TCP	POP3	ricezione posta elettronica
443	TCP	HTTPS	server web con connessione sicura

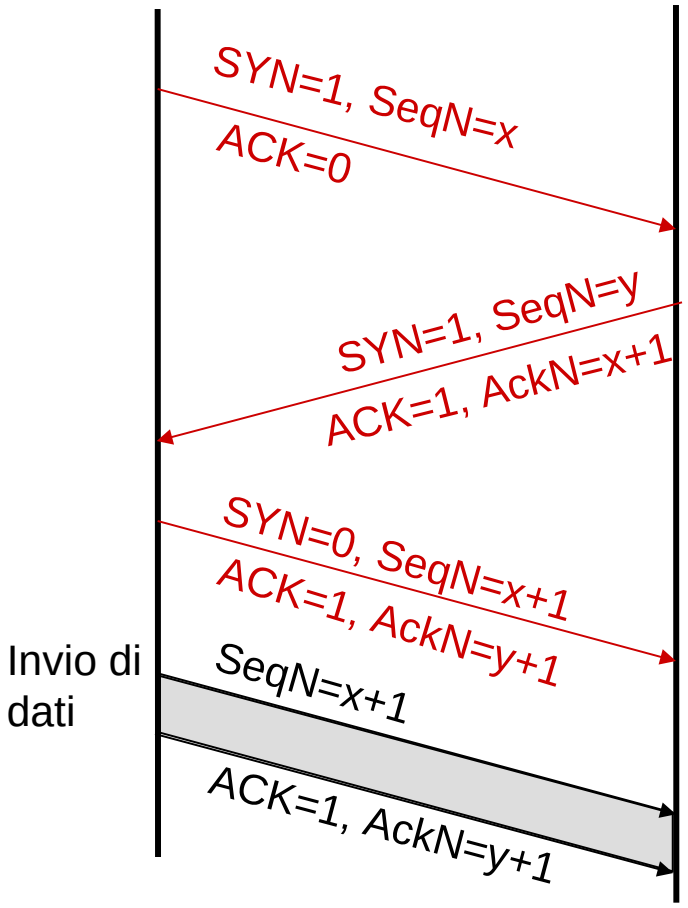
Esempio UDP: Query DNS





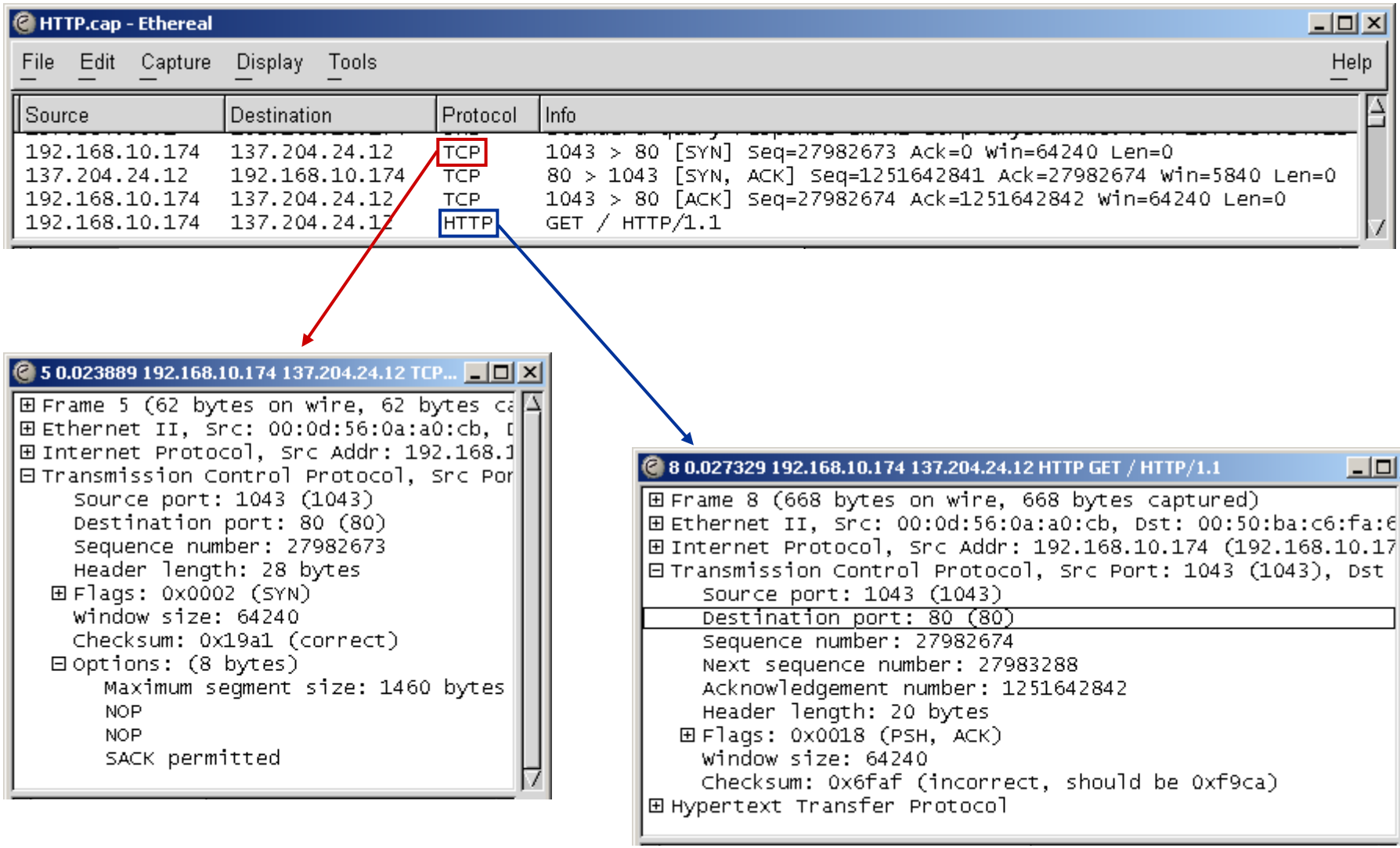


Apertura della connessione TCP

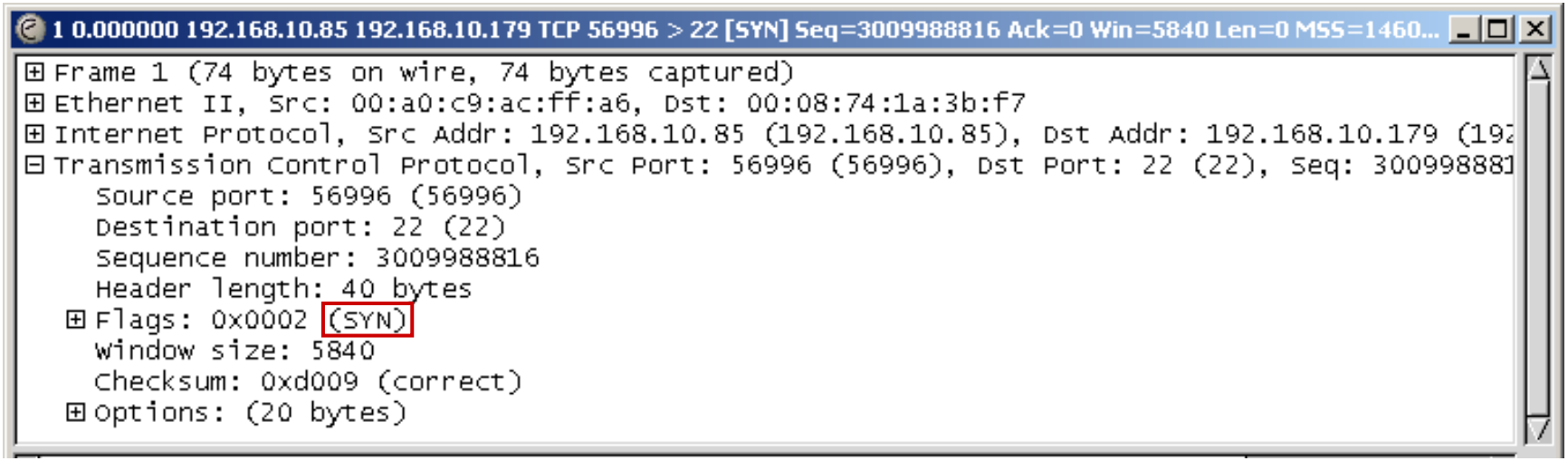


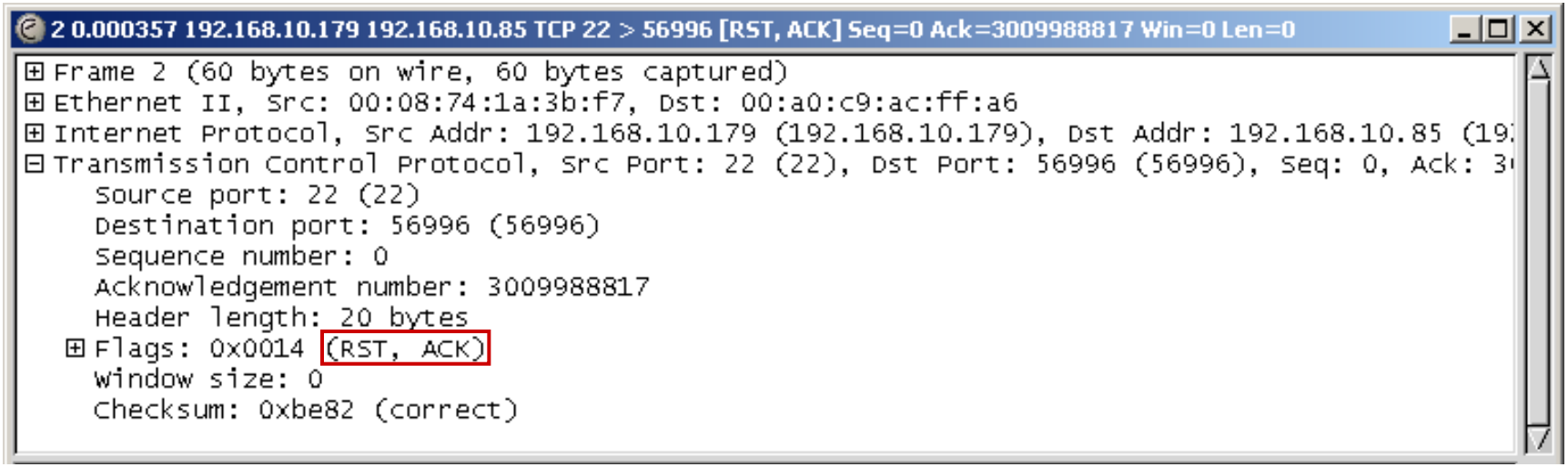
- L'apertura della connessione non è banale perché la rete può perdere,duplicare o ritardare i pacchetti
- TCP usa uno schema detto **Three Way Handshake** che risulta essere molto robusto
  - sincronizzazione di entrambi i numeri di sequenza
- Il primo pacchetto dati ha numero di sequenza uguale all'ACK precedente
  - ACK non occupa spazio di numerazione

Apertura della connessione TCP



Rifiuto di apertura della connessione TCP





Comando NETSTAT

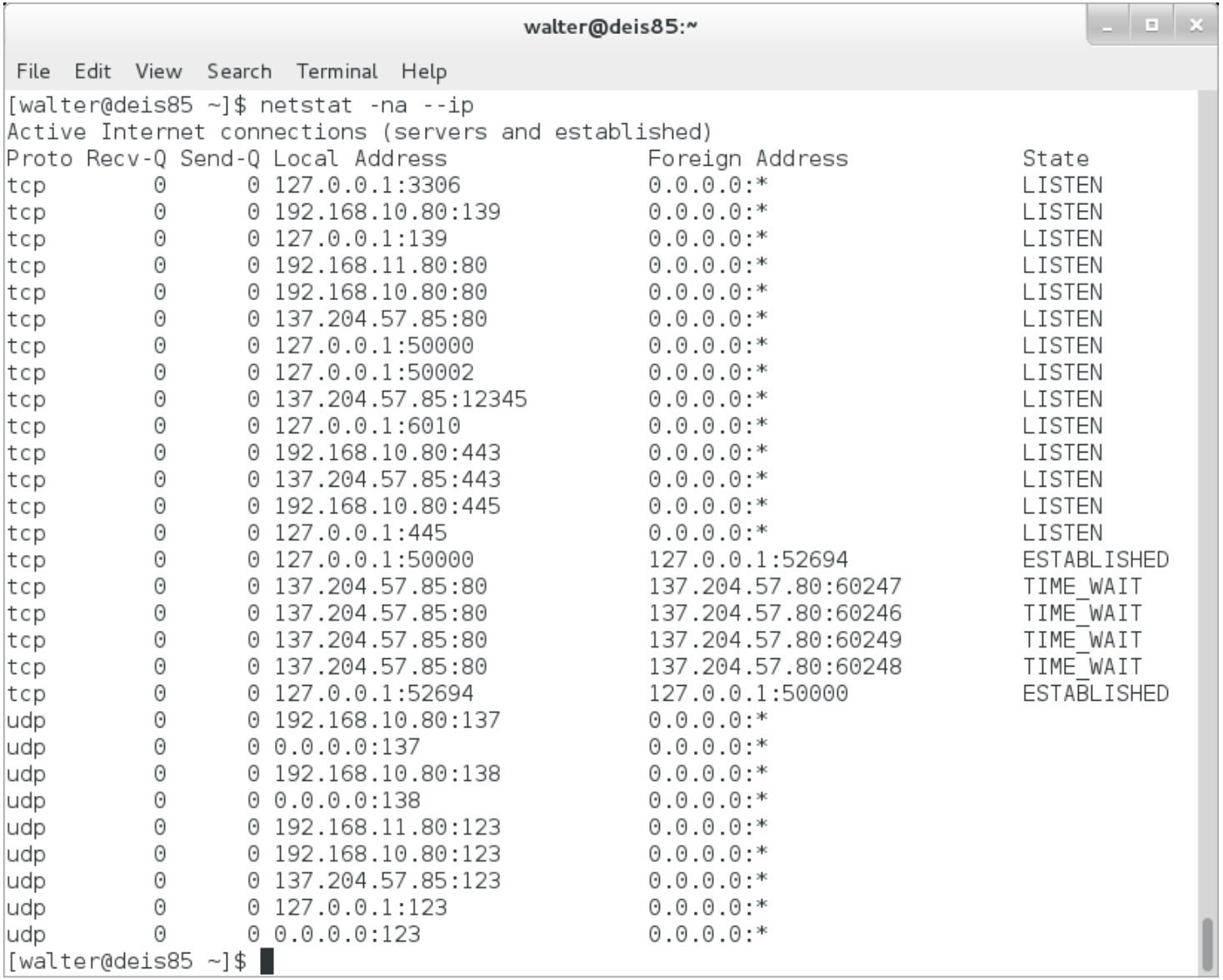
`netstat -na --ip`

visualizza le informazioni relative alle connessioni TCP attive in un host

`netstat -s`

visualizza una serie di dati di tipo statistico sul traffico relativo ai diversi protocolli

Comando NETSTAT – Esempio



Comando NMAP

`nmap HOST`

effettua un port scanning sulla macchina HOST e visualizza le porte TCP corrispondenti a processi in fase di ascolto (SERVER)

Port Scanning: si invia un SYN verso ogni porta

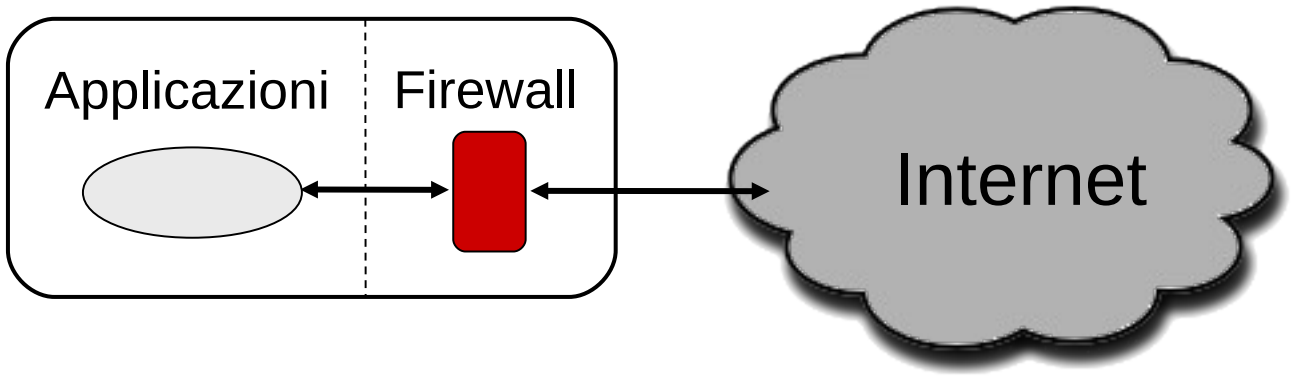
- se si riceve un SYN+ACK la porta è attiva
- se si riceve un RST+ACK la porta è chiusa
- se non si riceve nulla, la porta è filtrata

Comando NMAP – Esempio

Protezione di host: personal firewall

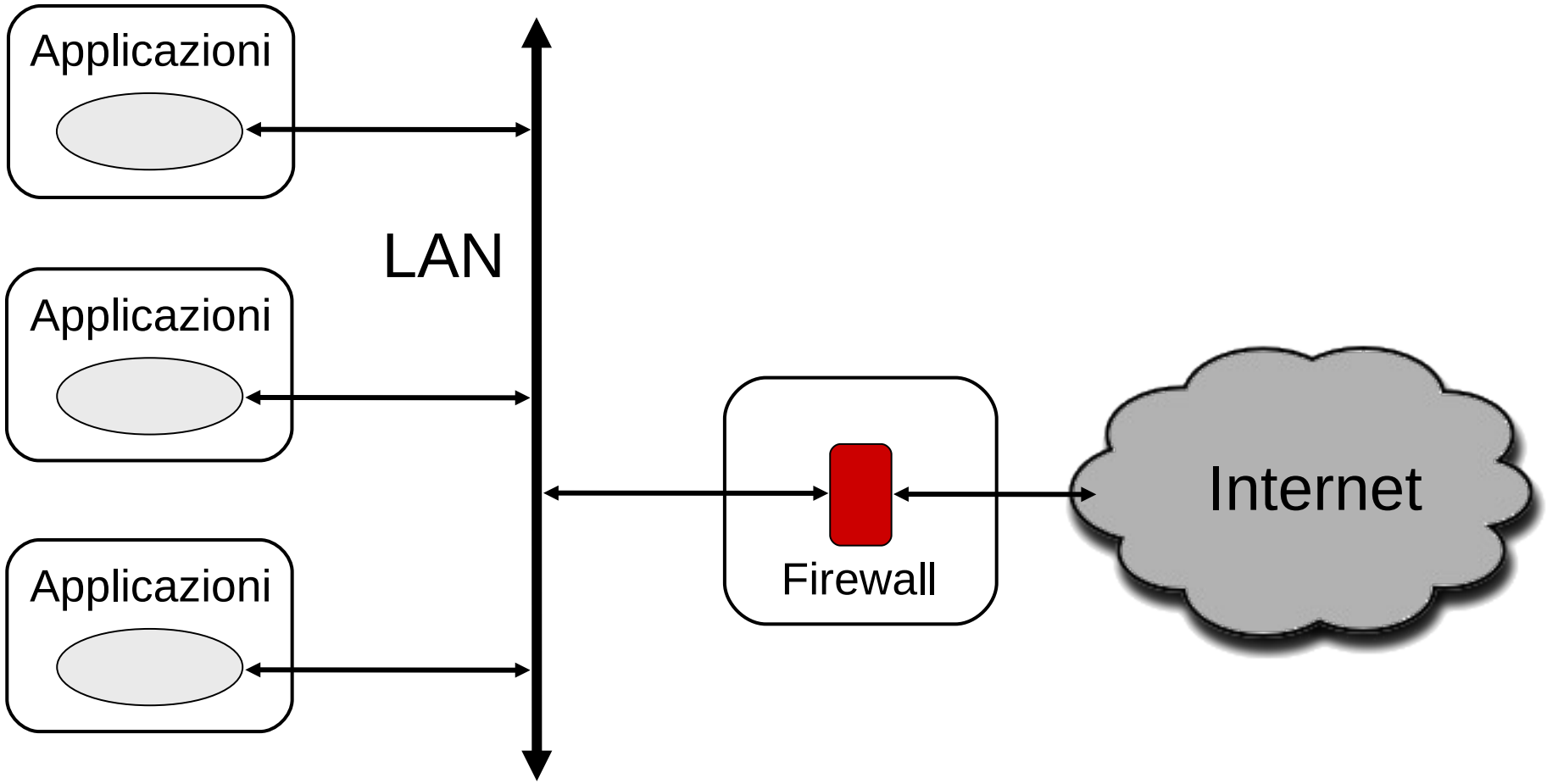
- Un firewall è un filtro software che serve a proteggersi da accessi indesiderati provenienti dall'esterno della rete

- Può essere semplicemente un programma installato sul proprio PC che
  - protegge quest'ultimo da attacchi esterni
  - protegge la rete da attacchi generati da potenziali virus presenti su PC



Protezione di rete: firewall

- Oppure può essere una macchina dedicata che filtra tutto il traffico proveniente da e diretto a una rete locale



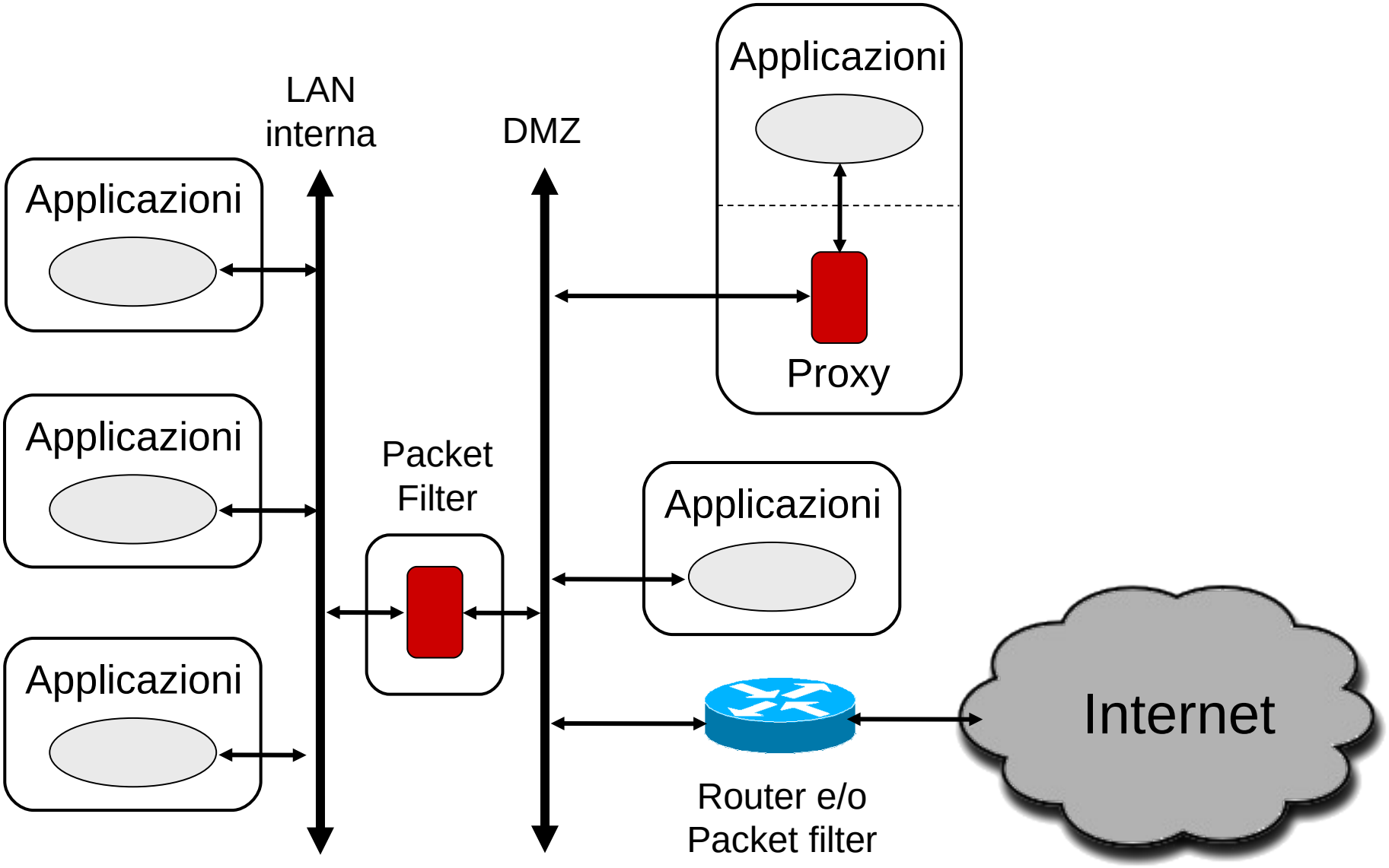
Firewall: caratteristiche

- Tutto il traffico fra la rete locale ed Internet deve essere filtrato dal firewall
- Solo il traffico autorizzato deve attraversare il firewall
- Si deve comunque permettere che i servizi di rete ritenuti necessari siano mantenuti
- Il firewall deve essere, per quanto possibile, immune da problemi di sicurezza sull'host
- In fase di configurazione di un firewall, per prima cosa si deve decidere la politica di default per i servizi di rete
  - **default deny**: tutti servizi non esplicitamente permessi sono negati
  - **default allow**: tutti i servizi non esplicitamente negati sono permessi

Livelli di implementazione

- Un firewall può essere implementato come
  - **Packet filter**
    - si interpone fra la rete locale ed Internet
    - filtra i datagrammi IP da trasferire attraverso le varie interfacce
    - il filtro scarta i datagrammi sulla base di
      - interfaccia di provenienza e/o destinazione
      - indirizzo MAC e/o IP, sorgente o destinazione
      - tipo di servizio (campo PROTOCOL o porta TCP/UDP)
    - può essere **stateful** o **stateless** nel caso in cui tenga memoria o meno delle connessioni e/o dei flussi di traffico in corso
  - **Application layer firewall** o **Proxy server**
    - filtra i dati sulla base dei protocolli applicativi (HTTP, FTP, ecc…)
    - l'accesso alla rete esterna è consentito solo attraverso il proxy

Utilizzo di packet filter e proxy server



IPTABLES

- IPTABLES implementa funzionalità di stateful packet filter nei kernel Linux 2.4 e successivi
- Lavora a livello di kernel ed ha il controllo dei pacchetti IP in transito sulle interfacce di rete (loopback compreso)
- I pacchetti IP processati da IPTABLES sono soggetti a diverse modalità di elaborazione chiamate **table** (o tabelle), ciascuna delle quali è composta da gruppi di regole denominate **chain**
- IPTABLES definisce quattro tabelle principali
  - **filter** (filtraggio di pacchetti)
  - **nat** (sostituzione di indirizzi IP)
  - **mangle** (manipolazione ulteriore dei pacchetti: TOS, TTL, …)
  - **raw** (esclusione di pacchetti dal connection tracking)

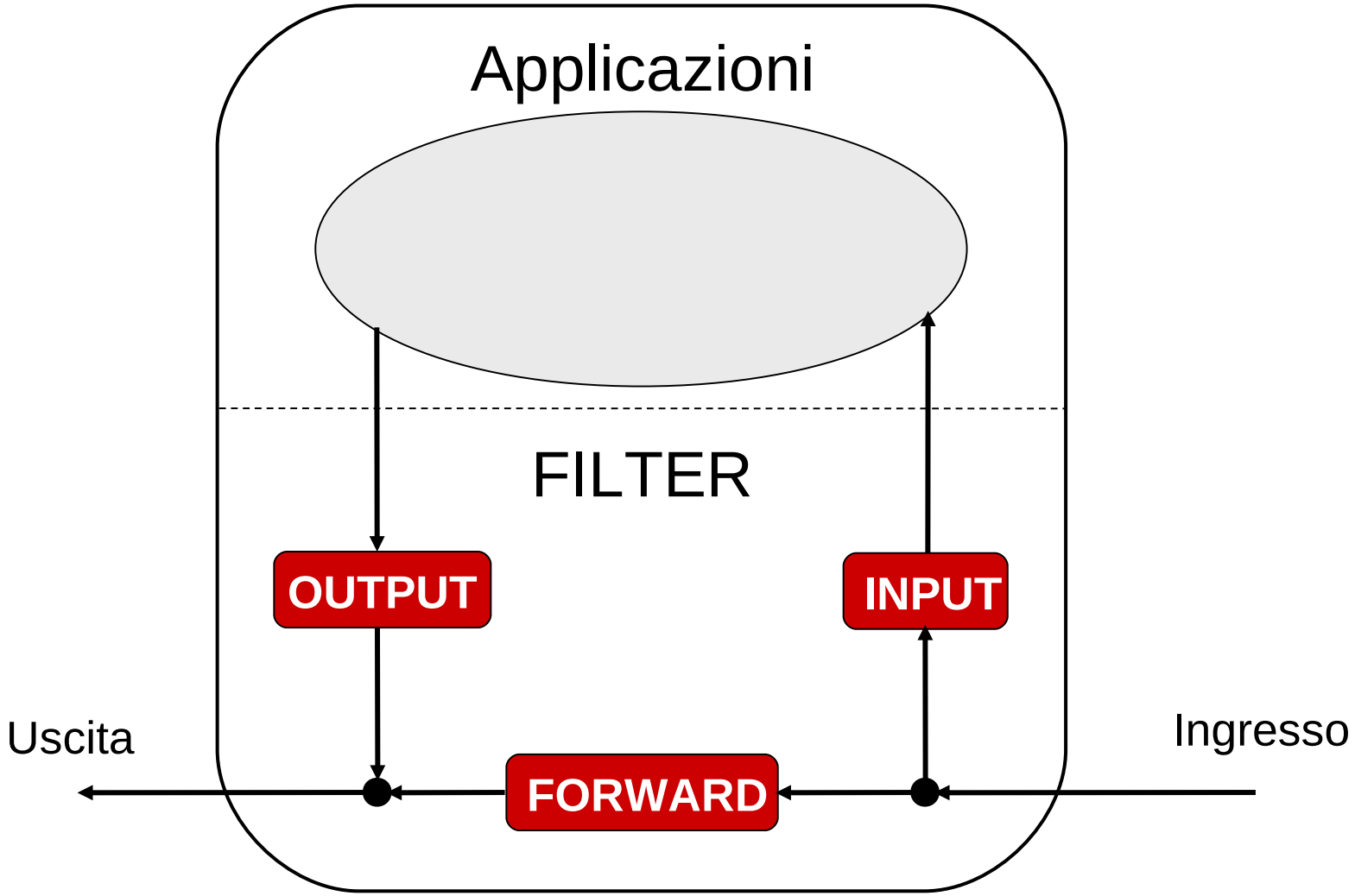
IPTABLES: la tabella FILTER

- Le funzionalità di firewall vere e proprie sono implementate dalla tabella **filter**, che si occupa di filtrare i pacchetti sulla base dell'interfaccia di provenienza e dei parametri contenuti nelle intestazioni IP e TCP

- Nella tabella filter sono presenti tre chain predefinite
  - **INPUT**: contiene le regole di filtraggio da usare sui pacchetti in arrivo al firewall (destinati all'host locale)
  - **OUTPUT**: contiene le regole da usare sui pacchetti in uscita dal firewall (originati dall'host locale)
  - **FORWARD**: contiene le regole da usare sui pacchetti in transito nel firewall (inoltrati tra interfacce diverse)

- E' possibile definire ulteriori chain

IPTABLES: chain della tabella FILTER



IPTABLES: regole della tabella FILTER

- Quando un pacchetto viene processato da una chain, esso è soggetto alle regole specificate in essa, **secondo l'ordine di inserimento**
- Una regola può stabilire se scartare (**DROP**), rifiutare esplicitamente (**REJECT**) o accettare (**ACCEPT**) un pacchetto in base a
  - interfaccia di rete coinvolta
  - indirizzo IP di origine e/o destinazione
  - protocollo (TCP, UDP, ICMP)
  - porta TCP o UDP di origine e/o destinazione
  - tipo di messaggio ICMP
  - ecc...
- Se un pacchetto non soddisfa nessuna regola, viene applicata la regola di default, o **policy**, di quella chain

IPTABLES: gestione della tabella FILTER

- Per visualizzare le regole attualmente in uso da ogni chain della tabella filter:

  iptables -L [-nv --line-num]

- Per visualizzare le regole attualmente in uso da una chain specifica:

  iptables -L <chain>

- Per impostare la policy di default di una chain:

  iptables -P <chain> <policy>

- Per aggiungere una regola in coda ad una chain:

  iptables -A <chain> <rule specs> -j <policy>

dove: **<chain> = INPUT | OUTPUT | FORWARD | ...**
<policy> = ACCEPT | DROP | REJECT | ...
<rule specs> specifica su quali pacchetti applicare la regola

Nota: **REJECT** non è ammessa come policy di default

IPTABLES: gestione della tabella FILTER

- Per inserire una regola in una chain nella posizione **<N>**:
  iptables -I <chain> <N> <rule specs> -j <policy>

- Per sostituire la regola nella posizione **<N>** di una chain:
  iptables -R <chain> <N> <rule specs> -j <policy>

- Per eliminare la regola nella posizione **<N>** di una chain:
  iptables -D <chain> <N>

- Per eliminare (flush) tutte le regole da una specifica chain o da tutte le chain (non agisce sulla policy di default):
  iptables -F <chain>
  iptables -F

IPTABLES: come specificare una regola

- Per specificare l'interfaccia di ingresso o di uscita:
  -i <interface> **-o <interface>**

- Per specificare l'IP (host o rete) di origine o destinazione:
  -s <address>/<netmask>
  -d <address>/<netmask>

  * la netmask può essere in formato decimale puntato o CIDR

- Per specificare il protocollo:
  -p tcp | udp | icmp | ...

* l'elenco dei protocolli supportati è in **/etc/protocols**

- Per specificare la porta (TCP/UDP) di origine o destinazione:
  --sport <port> **--dport <port>**

* l'elenco delle porte con i corrispondenti servizi è in **/etc/services**

IPTABLES: come specificare una regola

- Per specificare il tipo di messaggio ICMP:
  --icmp-type <typename>
  * l'elenco dei tipi di messaggi ICMP riconosciuti da IPTABLES è visualizzabile con: **iptables -p icmp -h**

- Per specificare l'indirizzo MAC di origine:
  --mac-source <MAC_address>

- E' possibile specificare il negato di un'opzione tramite l'operatore **!**
  es. **! -s <address>/<netmask>**

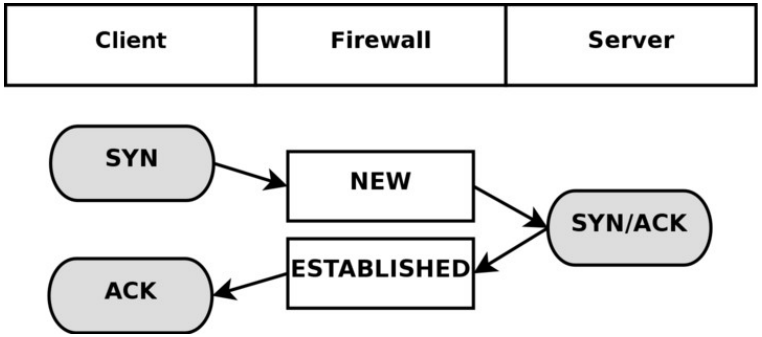
- Per specificare pacchetti di connessioni TCP o flussi UDP nuovi o già attivi (stateful packet filter):
  -m state --state NEW
  -m state --state ESTABLISHED

IPTABLES: gestione dello stato (conntrack)

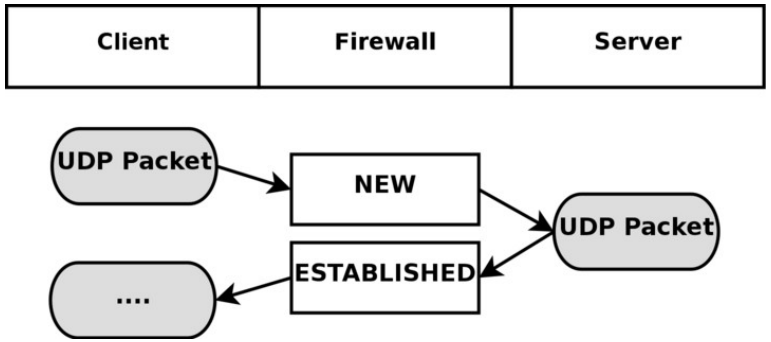
- IPTABLES definisce i seguenti stati di flussi/connessioni:
  - **NEW**: generato da un pacchetto appartenente a un flusso/connessione non presente nella tabella conntrack
  - **ESTABLISHED**: associato a flussi/connessioni dei quali sono stati già accettati pacchetti precedenti, in entrambe le direzioni
  - **RELATED**: associato a flussi/connessioni non ESTABLISHED ma che sono correlati con flussi/connessioni ESTABLISHED (es. connessioni FTP control e FTP data)
  - **INVALID**: associato a pacchetti il cui stato non è tracciabile
  - **UNTRACKED**: associato a pacchetti non soggetti alla modalità stateful (tabella **raw**)
- Per visualizzare lo stato di flussi/connessioni usare il comando **`conntrack -L`**

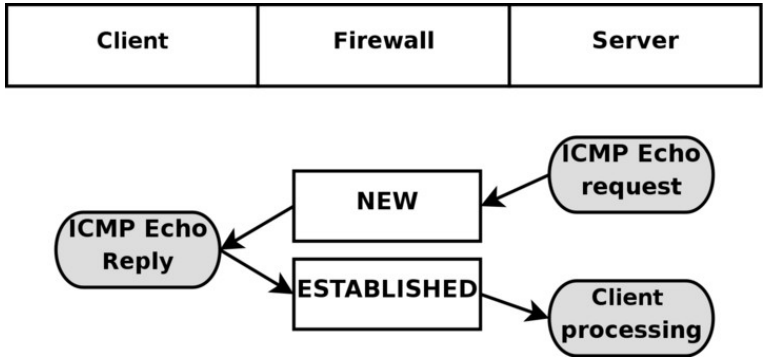
IPTABLES: gestione dello stato (conntrack)

- Connessioni TCP



- Flussi UDP e ICMP

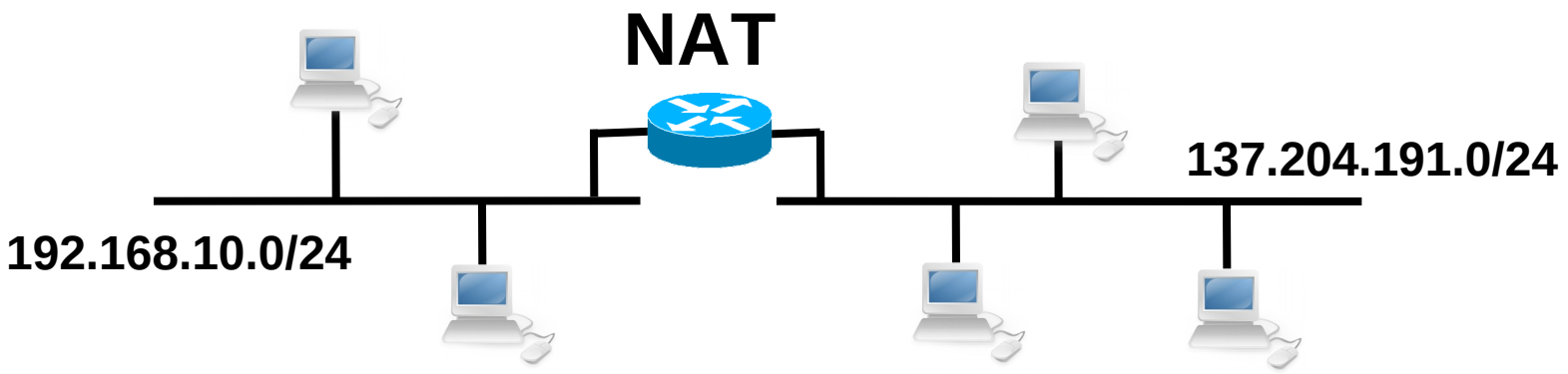




http://www.iptables.info/en/connection-state.html

Network Address Translation (NAT)

- Tecnica per il filtraggio di pacchetti IP con sostituzione degli indirizzi o mascheramento
- Definito nella RFC 3022 per permettere a host appartenenti a reti IP private l'accesso a reti IP pubbliche tramite un apposito gateway
- Utile per il risparmio di indirizzi IP pubblici e il riutilizzo di indirizzi IP privati



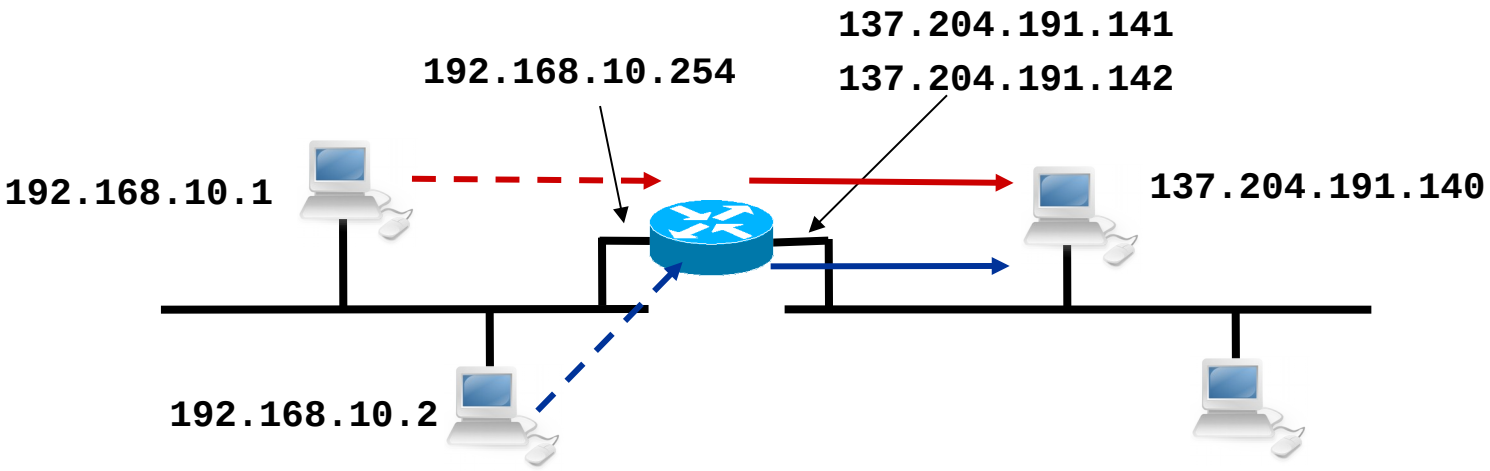
Reti IP private (RFC 1918)

- Alcuni gruppi di indirizzi sono riservati a reti IP private
- Essi non sono raggiungibili dalla rete pubblica
- I router di Internet non instradano datagrammi destinati a tali indirizzi
- Possono essere riutilizzati in reti isolate

- da **10.0.0.0** a **10.255.255.255** = **10.0.0.0/8**
- da **172.16.0.0** a **172.31.255.255** = **172.16.0.0/12**
- da **192.168.0.0** a **192.168.255.255** = **192.168.0.0/16**

Basic NAT – Conversione di indirizzo

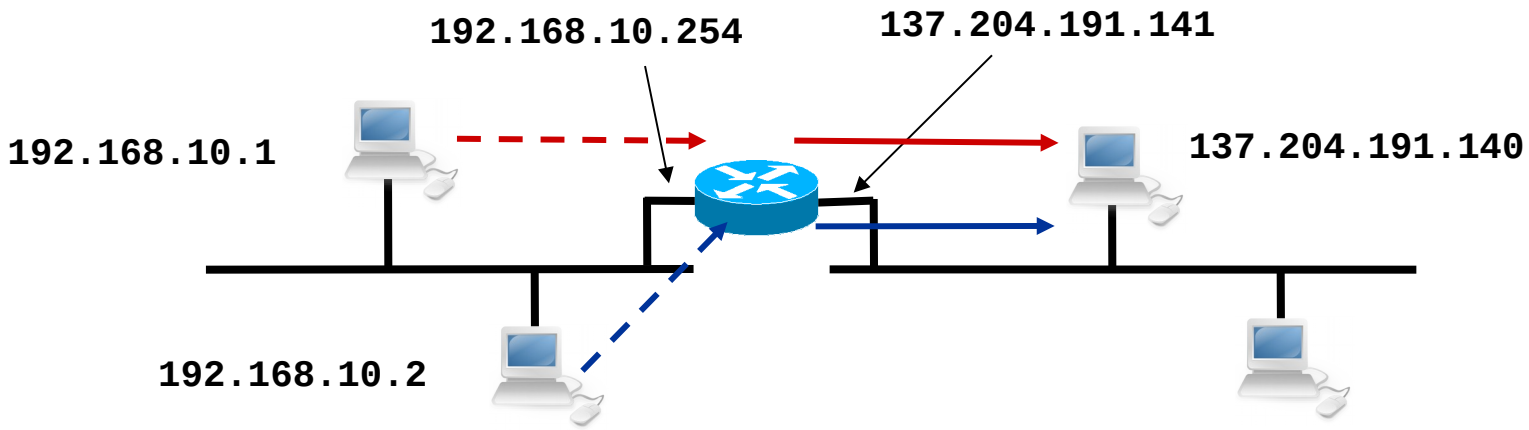
- Il NAT può fornire una semplice conversione di indirizzo IP (statica o dinamica)
- Conversioni contemporanee limitate dal numero di indirizzi IP pubblici a disposizione del gateway NAT



192.168.10.1	--------->	137.204.191.140
137.204.191.141	--------->	137.204.191.140
192.168.10.2	--------->	137.204.191.140
137.204.191.142	--------->	137.204.191.140

NAPT – Conversione di indirizzo e porta

- Il NAT può fornire anche conversione di indirizzo IP e porta TCP o UDP (anche noto come PAT)
- Conversioni contemporanee possibili anche con un unico indirizzo IP pubblico del gateway NAT



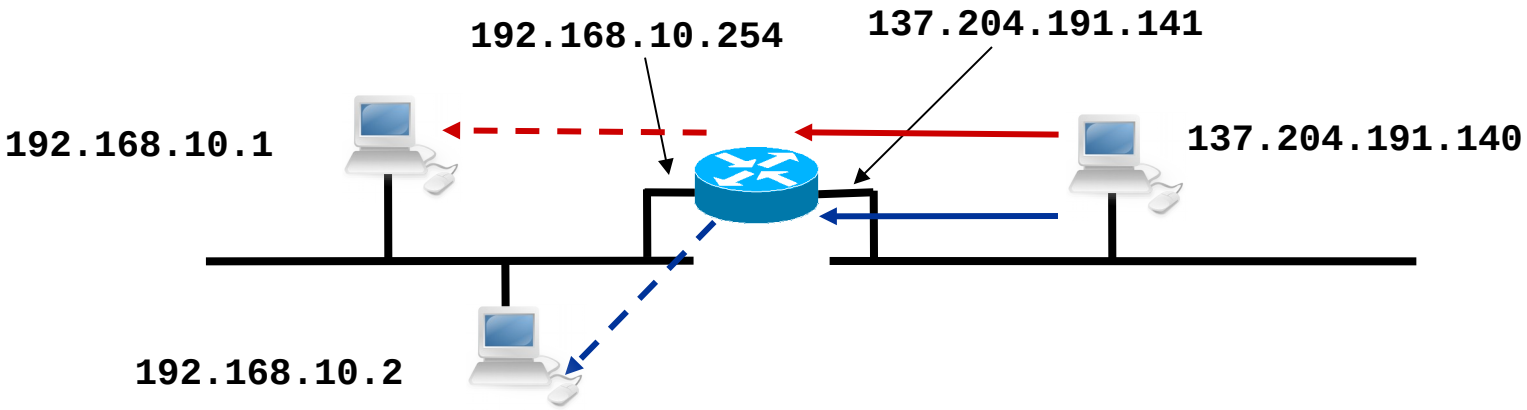
192.168.10.1:3123	⇢	137.204.191.140:80
137.204.191.141:3123	→	137.204.191.140:80
192.168.10.2:3123	⇢	137.204.191.140:80
137.204.191.141:4135	→	137.204.191.140:80

Direzione delle connessioni

- Il NAT si applica generalmente da rete privata verso rete pubblica (**Outbound NAT**)
  - tiene memoria delle connessioni e/o dei flussi di traffico che lo attraversano (intrinsecamente stateful)
  - registra le traduzioni in corso in una cache
    - traduzioni con tempo di vita limitato
  - si preoccupa di effettuare la conversione inversa quando arrivano pacchetti in direzione opposta relativi ad un flusso già attivo

Direzione delle connessioni

- E’ anche possibile contattare dalla rete pubblica un host sulla rete privata (**Bi-directional NAT**)
  - bisogna configurare esplicitamente il NAT (**Port Forwarding**)
  - bisogna utilizzare porte diverse se gli indirizzi sono limitati

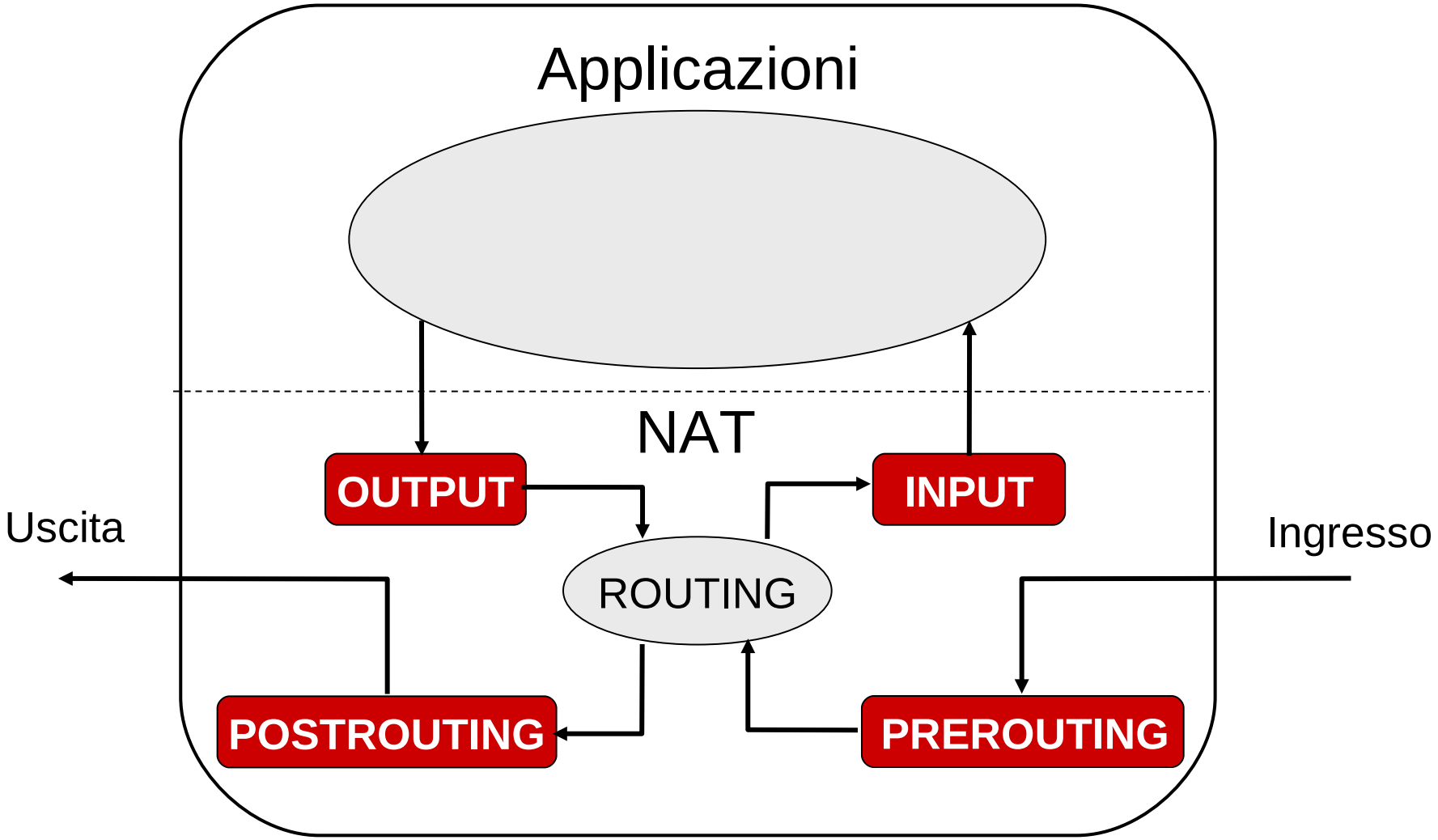


137.204.191.141:80	←	137.204.191.140:4014
192.168.10.1:80	←	137.204.191.140:4014
137.204.191.141:8080	←	137.204.191.140:4015
192.168.10.2:80	←	137.204.191.140:4015

IPTABLES: la tabella NAT

- Le funzionalità di NAT sono implementate da IPTABLES tramite la tabella **nat**
- Nella tabella nat sono presenti quattro chain predefinite
  - **PREROUTING**: contiene le regole da usare prima dell'instradamento per sostituire l'indirizzo di destinazione dei pacchetti (policy = Destination NAT o **DNAT**)
  - **POSTROUTING**: contiene le regole da usare dopo l'instradamento per sostituire l'indirizzo di origine dei pacchetti (policy = Source NAT o **SNAT**)
  - **OUTPUT/INPUT**: contiene le regole da usare per sostituire l'indirizzo di pacchetti generati/ricevuti localmente
- La policy **ACCEPT** vuol dire assenza di conversione
- La policy **MASQUERADE** vuol dire conversione implicita nell'indirizzo IP assegnato all'interfaccia di uscita

IPTABLES: chain della tabella NAT

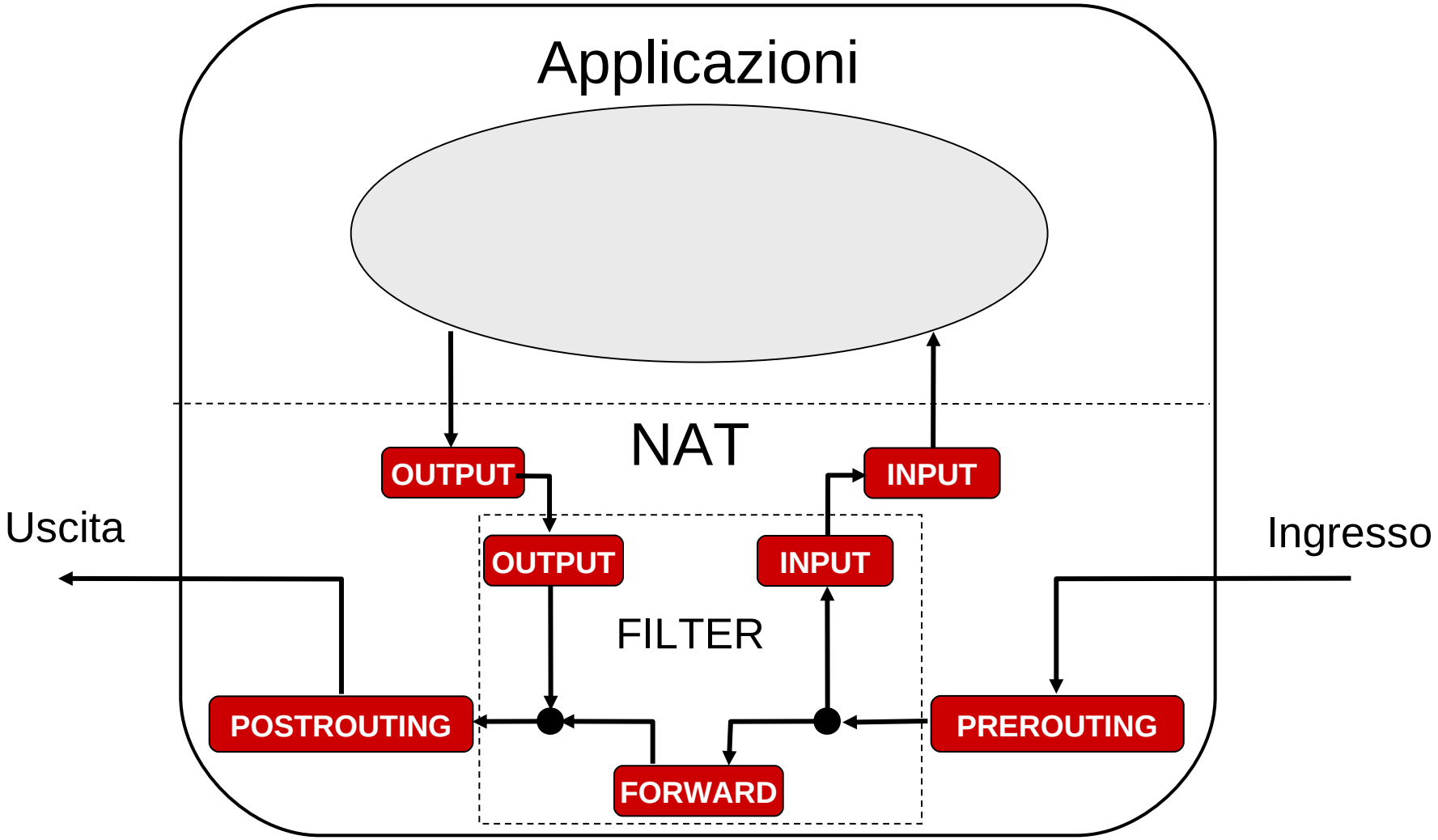


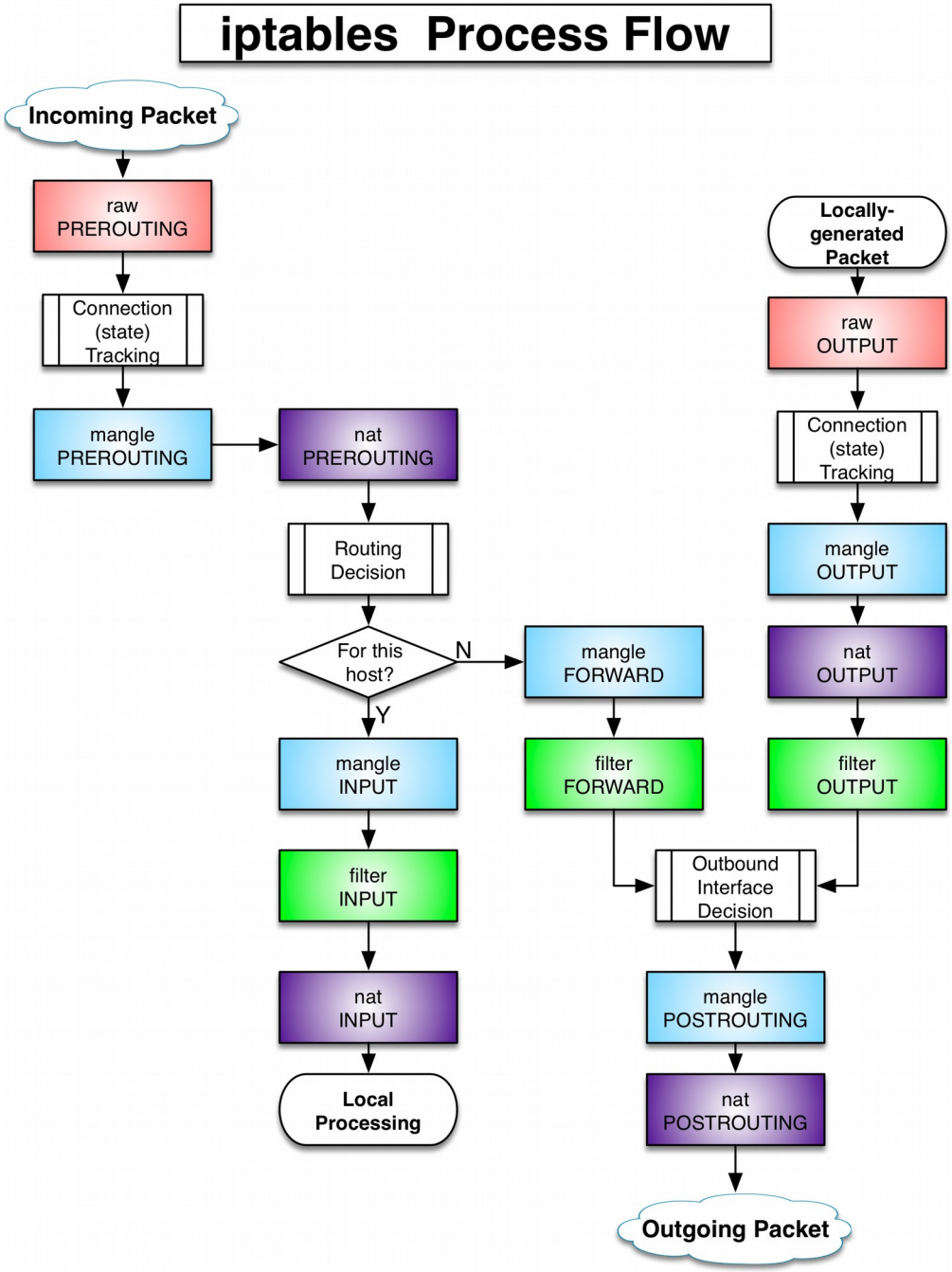
IPTABLES: gestione della tabella NAT

- Per visualizzare le regole attualmente in uso da ogni chain della tabella nat:
  iptables -t nat -L [-nv --line-num]
- Per visualizzare le regole attualmente in uso da una chain specifica:
  iptables -t nat -L <chain>
- Per aggiungere una regola in coda ad una chain:
  iptables -t nat -A <chain> <rule specs> -j <policy>

dove:
<chain> = **POSTROUTING | PREROUTING | OUTPUT | …**
<policy> = **ACCEPT | MASQUERADE |**
SNAT --to-source <addr> |
DNAT --to-destination <addr>
<addr> = **<address> | <address>:<port>**
<rule specs> = come per la tabella filter

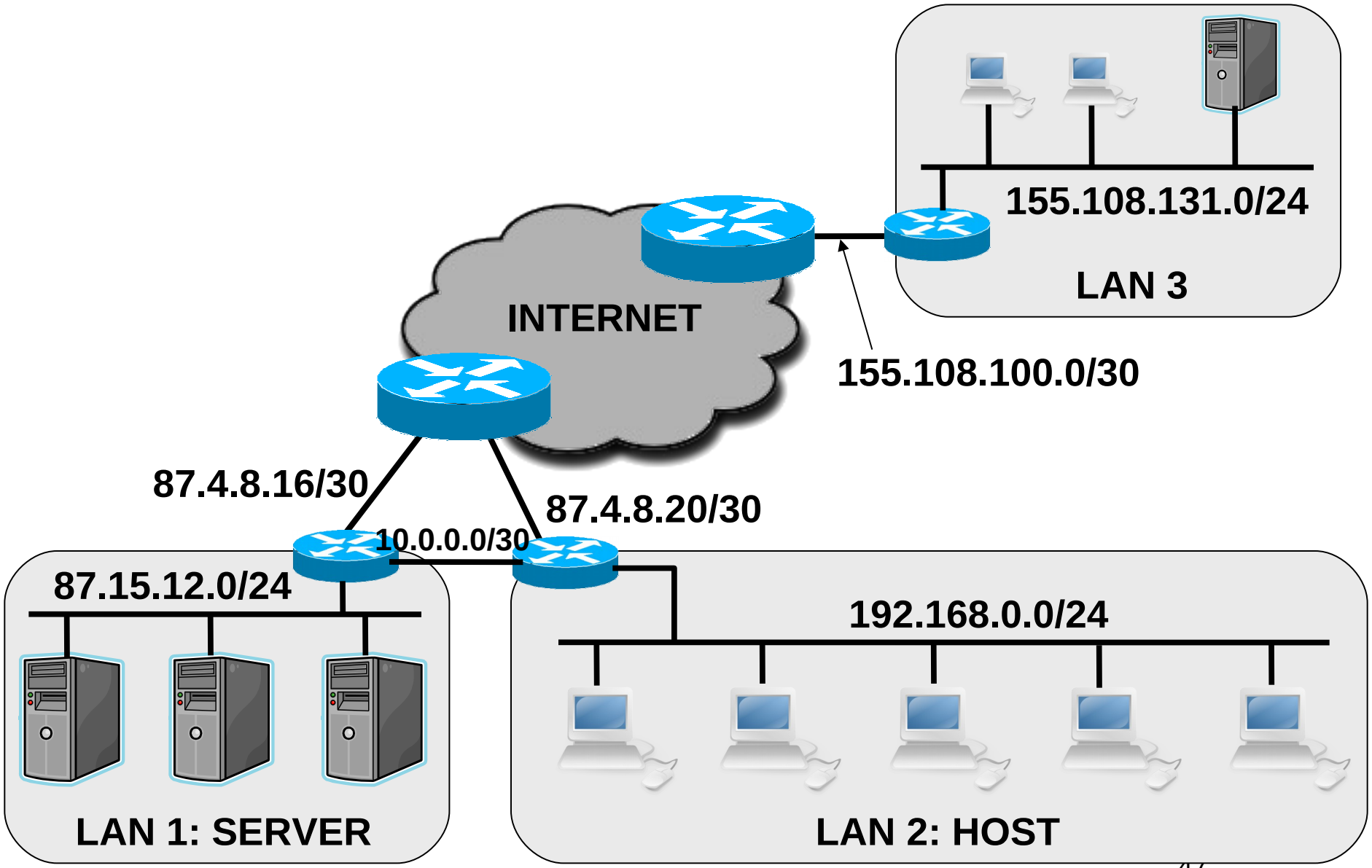
IPTABLES: FILTER + NAT





da https://stuffphilwrites.com/2014/09/iptables-processing-flowchart/

Case study: configurazione firewall e NAT



Case study: politiche di sicurezza aziendali

- Tutte le LAN aziendali che usano IP pubblici devono essere accessibili senza restrizioni da parte di connessioni provenienti dalle altre LAN aziendali
- I server nella LAN 1 devono essere accessibili da parte di connessioni non provenienti dalle LAN aziendali solo se dirette verso i servizi HTTP e HTTPS
- Almeno un host della LAN 2 deve poter essere raggiungibile via SSH dalla LAN 3
- Le altre LAN aziendali non devono essere accessibili da connessioni originate dall'esterno, ma tutti i loro host e server devono poter accedere all’esterno senza limitazioni

Case study: configurazione firewall LAN 1

- `iptables -P FORWARD DROP`
- `iptables -A FORWARD -i eth0 -m state --state NEW -j ACCEPT`
- `iptables -A FORWARD -i ppp1 -s 192.168.0.0/24 -m state --state NEW -j ACCEPT`
- `iptables -A FORWARD -i ppp0 -s 155.108.131.0/24 -m state --state NEW -j ACCEPT`
- `iptables -A FORWARD -i ppp0 -d 87.15.12.0/24 -p tcp --dport 80 -m state --state NEW -j ACCEPT`
- `iptables -A FORWARD -i ppp0 -d 87.15.12.0/24 -p tcp --dport 443 -m state --state NEW -j ACCEPT`
- `iptables -I FORWARD 1 -m state --state ESTABLISHED -j ACCEPT`

Case study: configurazione firewall e NAT LAN 2

- `iptables -P FORWARD DROP`
- `iptables -A FORWARD -i eth0 -m state --state NEW -j ACCEPT`
- `iptables -A FORWARD -i ppp1 -s 87.15.12.0/24 -m state --state NEW -j ACCEPT`
- `iptables -A FORWARD -i ppp0 -s 155.108.131.0/24 -m state --state NEW -j ACCEPT`
- `iptables -I FORWARD 1 -m state --state ESTABLISHED -j ACCEPT`

- `iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o ppp0 -j SNAT --to-source 87.4.8.21`
- `iptables -t nat -A PREROUTING -i ppp0 -d 87.4.8.21 -p tcp --dport 2222 -j DNAT --to-destination 192.168.0.1:22`

Case study: configurazione firewall LAN 3

- `iptables -P FORWARD DROP`
- `iptables -A FORWARD -i eth0 -m state --state NEW -j ACCEPT`
- `iptables -A FORWARD -i ppp0 -s 87.15.12.0/24 -m state --state NEW -j ACCEPT`
- `iptables -A FORWARD -i ppp0 -s 87.4.8.21 -m state --state NEW -j ACCEPT`
- `iptables -I FORWARD 1 -m state --state ESTABLISHED -j ACCEPT`