

## **1. INTRODUCCIÓ**

## **2. IPTABLES**

### **2.1. Kernel 2.6.2**

### **2.2. Regles**

### **2.3. Aplicacions**

#### **2.3.1. Filtrat IP**

#### **2.3.2. Comptabilitat IP**

#### **2.3.3. Masquerading & Forwarding**

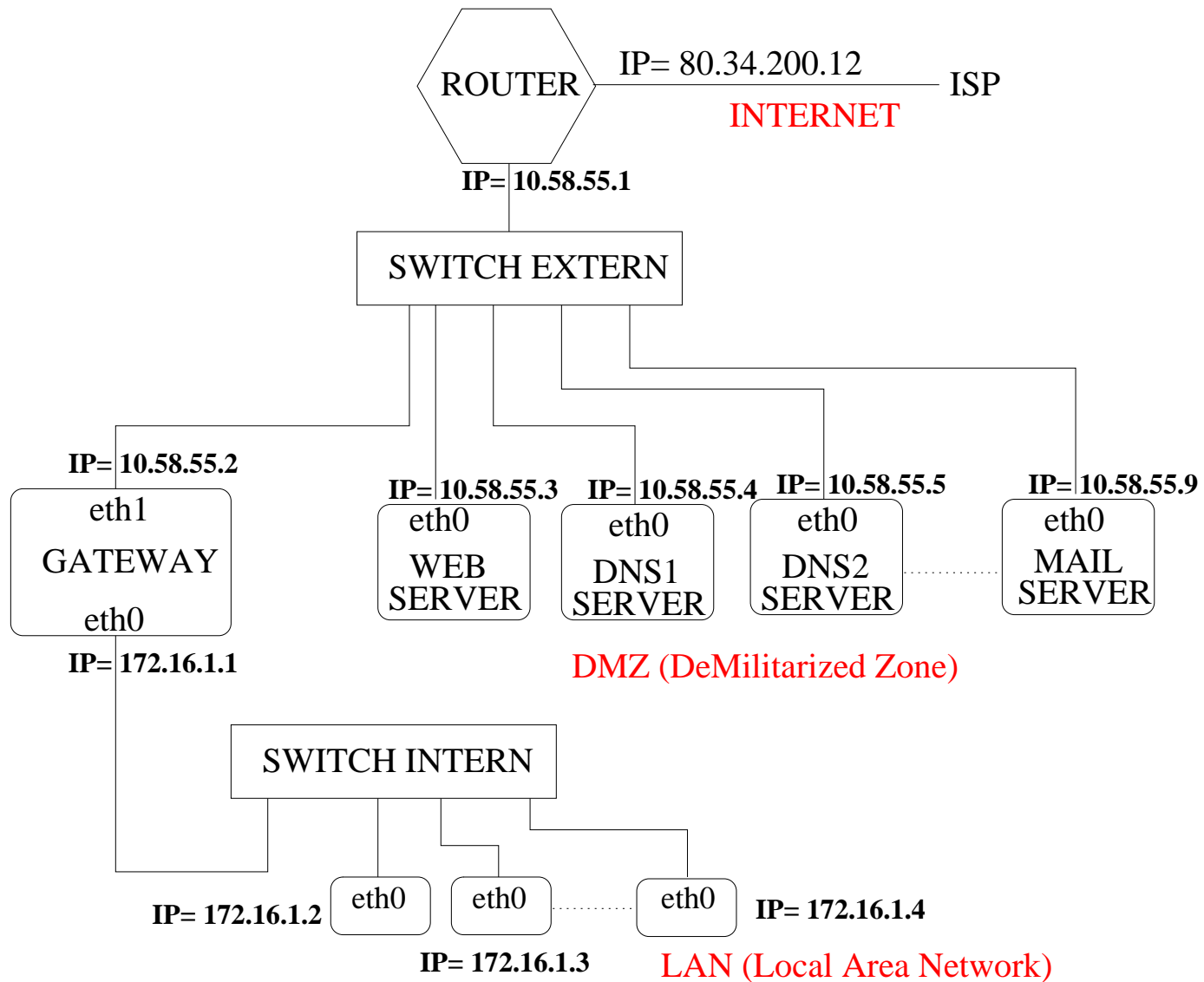
---

## 1. INTRODUCCIÓ - Màscares - Classes de xarxa

---

màscara	màxim # de màquines	classe
255.0.0.0	16.777.215	«classe A»
255.255.0.0	65.535	«classe B»
255.255.128.0	32.767	
...	...	...
255.255.255.0	255	«classe C»
...	...	...
255.255.255.240	15	
255.255.255.248	7	
255.255.255.252	3	

# 1. INTRODUCCIÓ - Esquema -Ideal- Servidors de Xarxa



---

## 1. INTRODUCCIÓ - Tipus de Servidors

---

<b>servidor</b>	<b>aplicacions</b>
mail	sendmail o qmail
web	apache
ftp	sftp
dns	named, bind
bases de dades	postgres, mysql
impressió	cups
backup	rsync, tar, dump
dos - impressió i fitxers	samba, NIS+

---

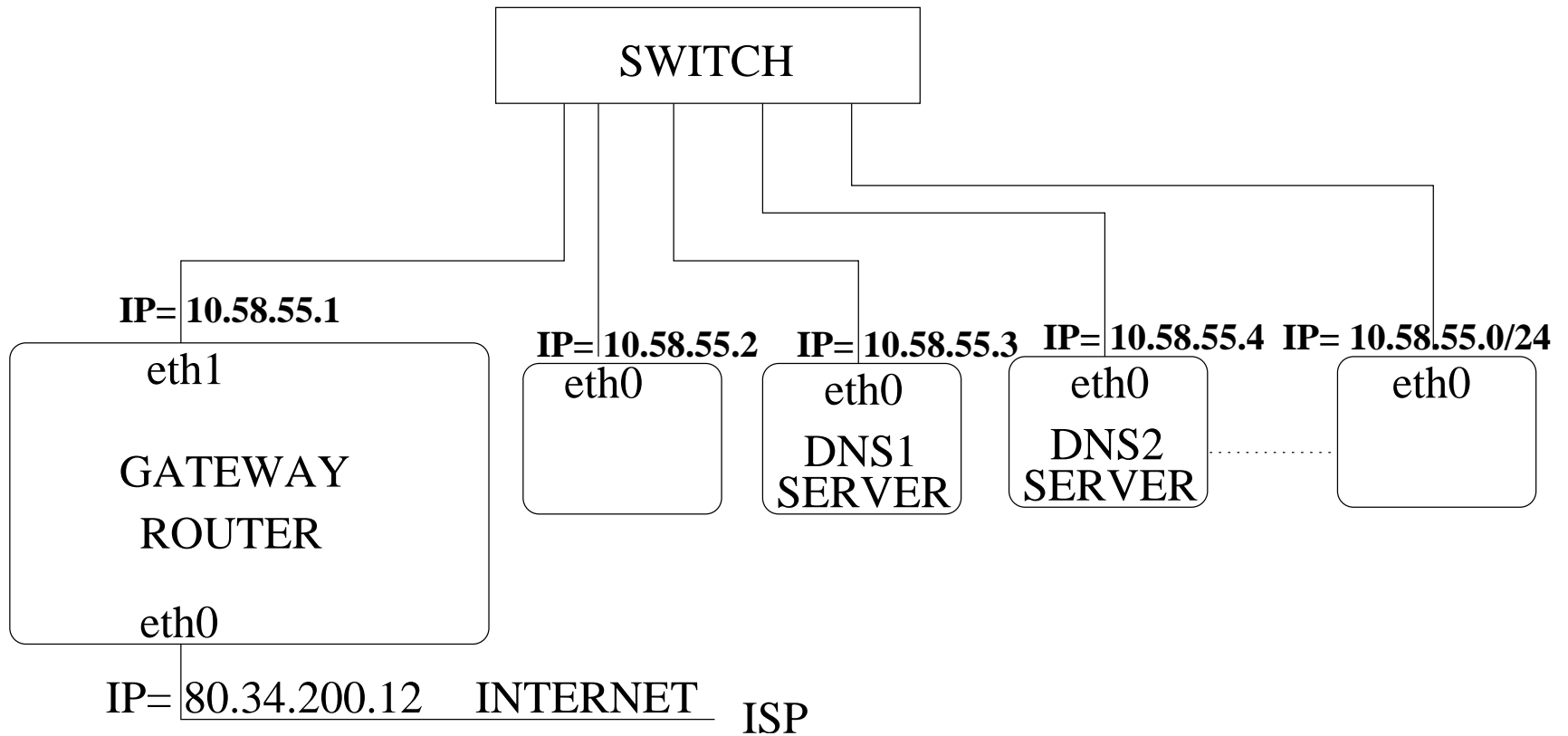
## 1. INTRODUCCIÓ - Aplicacions

---

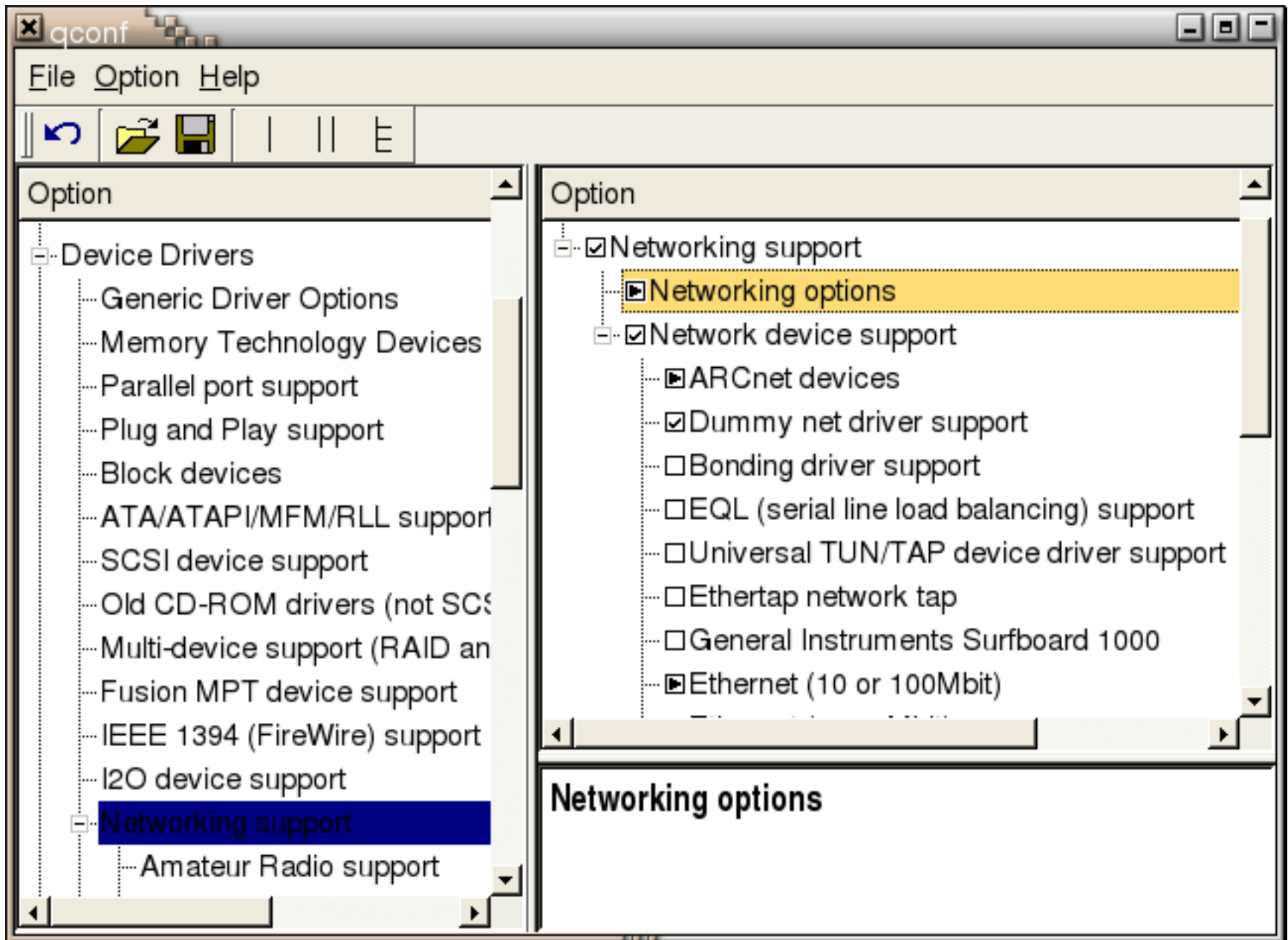
<b>aplicació</b>	<b>explicació</b>
logcheck	auditor de logs
tripwire, FAM	auditors d'alteració de fitxers
ethereal	sniffer
nmapfe/xnmap	escàner de ports
portsentry	auditor/detector d'escanneig de ports
nessus	auditor de seguretat -atacs-
procmail	filtre de mail

FAM: File Alteration Monitor

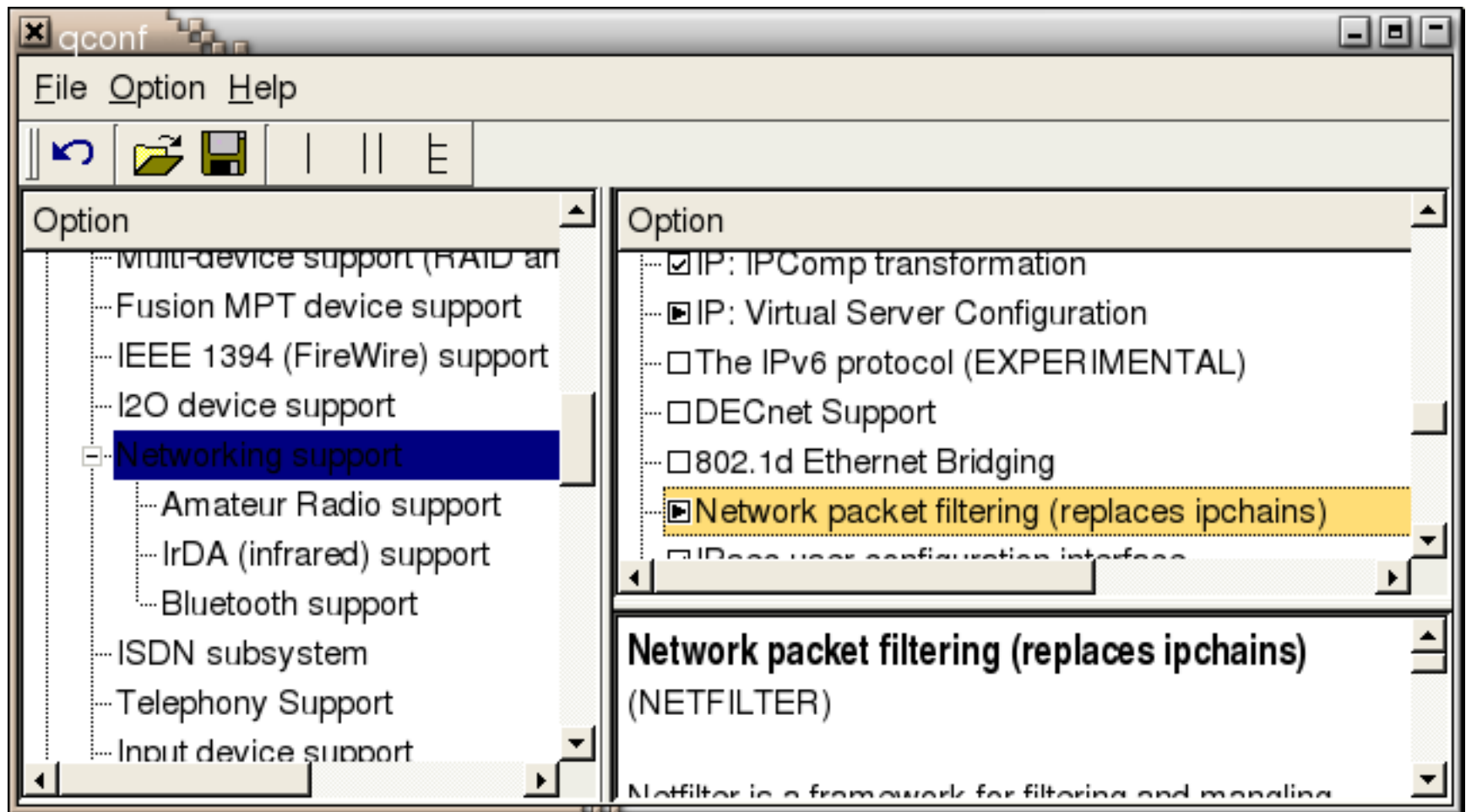
# 1. INTRODUCCIÓ - Esquema -Laboratori- Servidors de Xarxa



## 2. IPTABLES — 2.1. Kernel 2.6.2

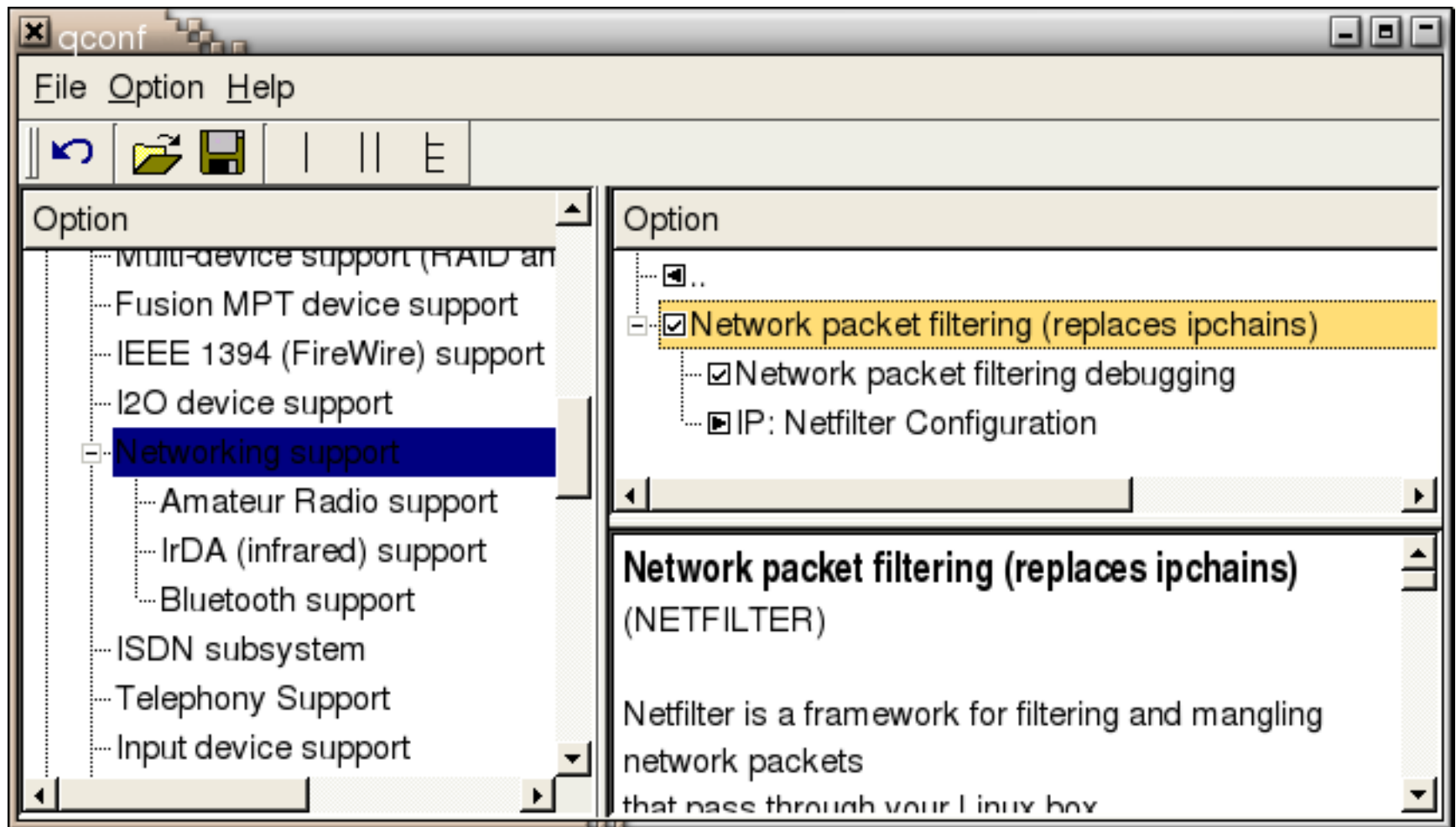


## 2.1. Kernel 2.6.2

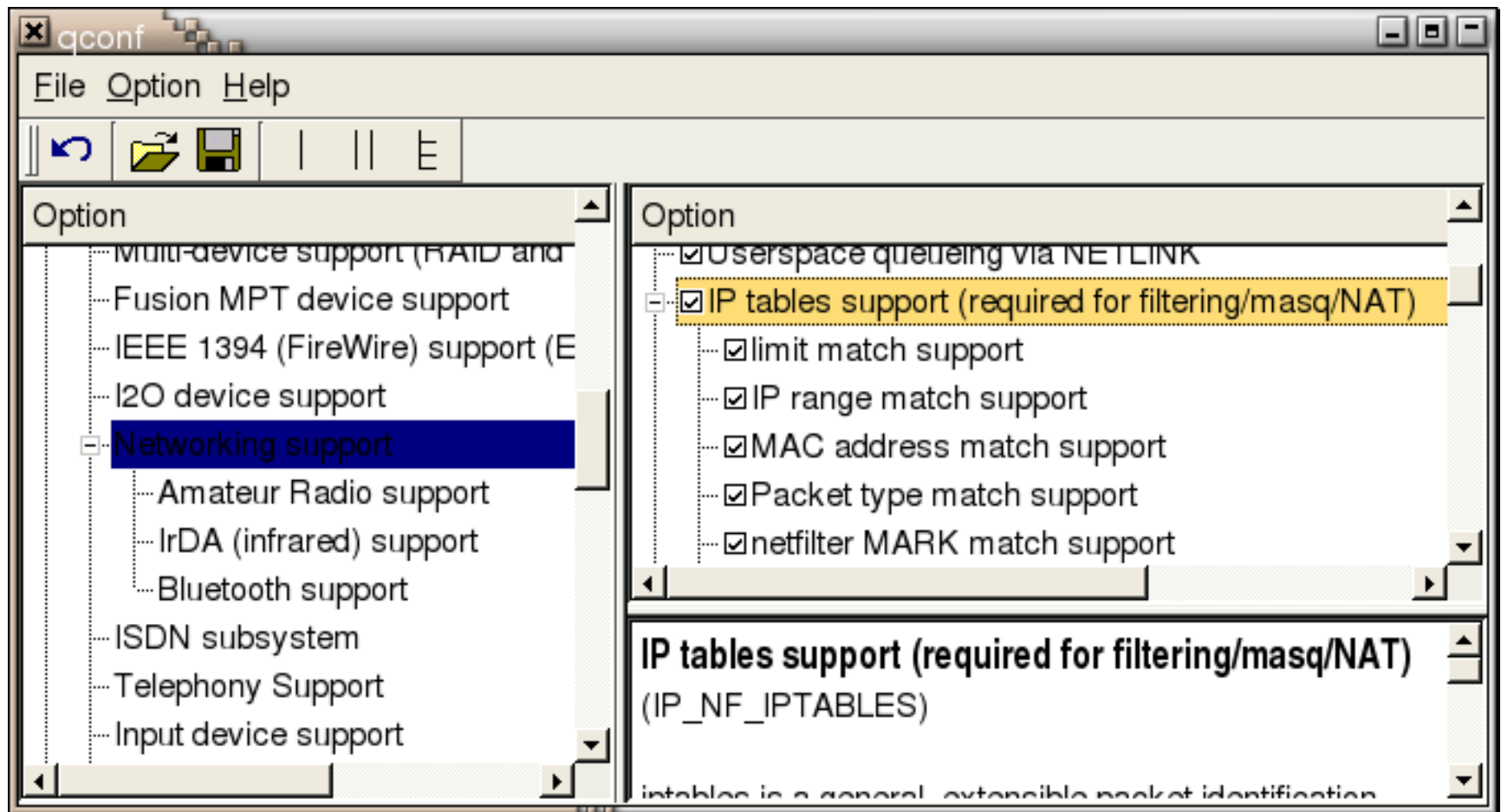




## 2.1. Kernel 2.6.2



## 2.1. Kernel 2.6.2



---

## 2.2. Regles

---

### Activació:

```
# modprobe ip_tables [altres mòduls ipt_*]
```

```
# /etc/init.d/iptables [start|restart|stop]
```

### Sintaxi regla:

```
# iptables [-t taula] ordre [[match] [acció]] [opció]
```

### Scripts:

```
/etc/init.d/iptables
```

```
/etc/rcx.d/S47iptables → /etc/init.d/iptables
```

---

## 2.2. Regles

---

### Taules (iptables)

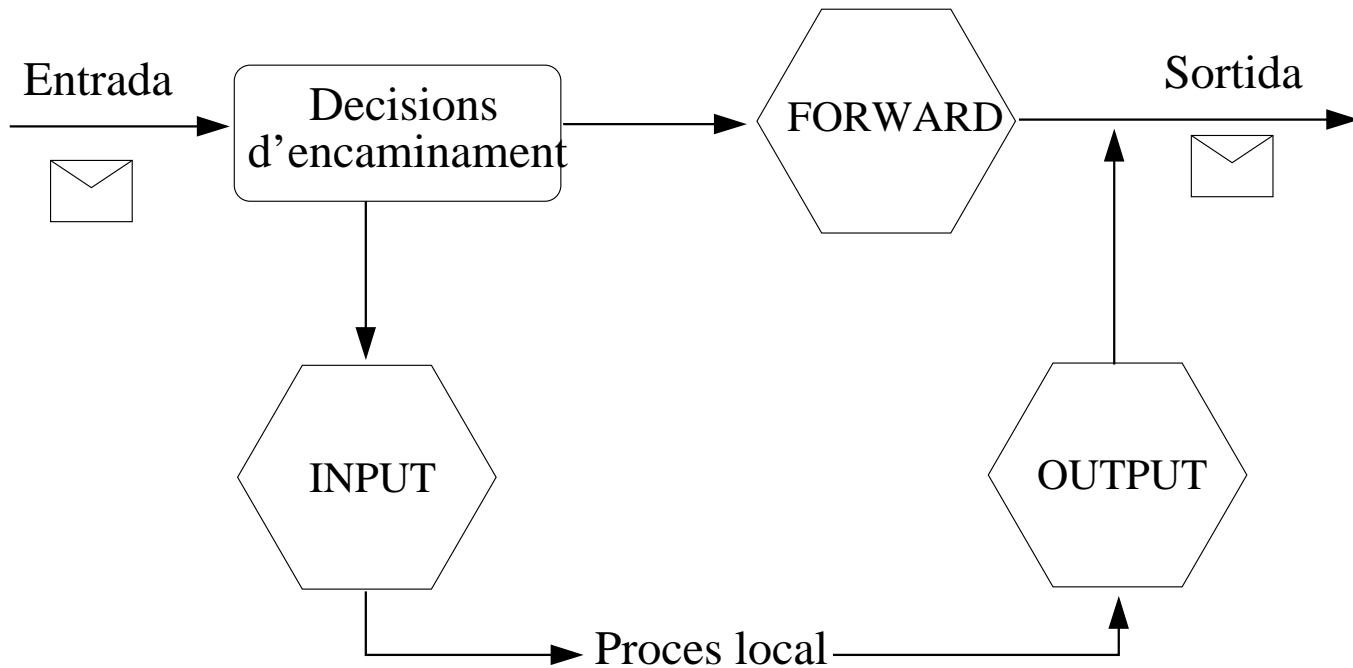
**filter:** utilitzada per filtratge. Taula per defecte (quan no es posa l'opció *-t*). Composta de 3 **cadenaes -chains-** : **INPUT** (actua en paquets entrants al host); **FORWARD** (encaminament de paquets) i **OUTPUT** (actua en paquets sortint del host).

**nat (network address translation):** utilitzada per canviar emissor i destinatari de paquets. Composta de tres cadenaes: **PREROUTING** (realitza DNAT -Destination Network Address Translation- altera destinatari dels paquets p.e. Internet → LAN); **OUTPUT** (altera paquets generats localment) i **POSTROUTING** (realitza SNAT -Source Network ...- altera emissor dels paquets, p.e. LAN → Internet).

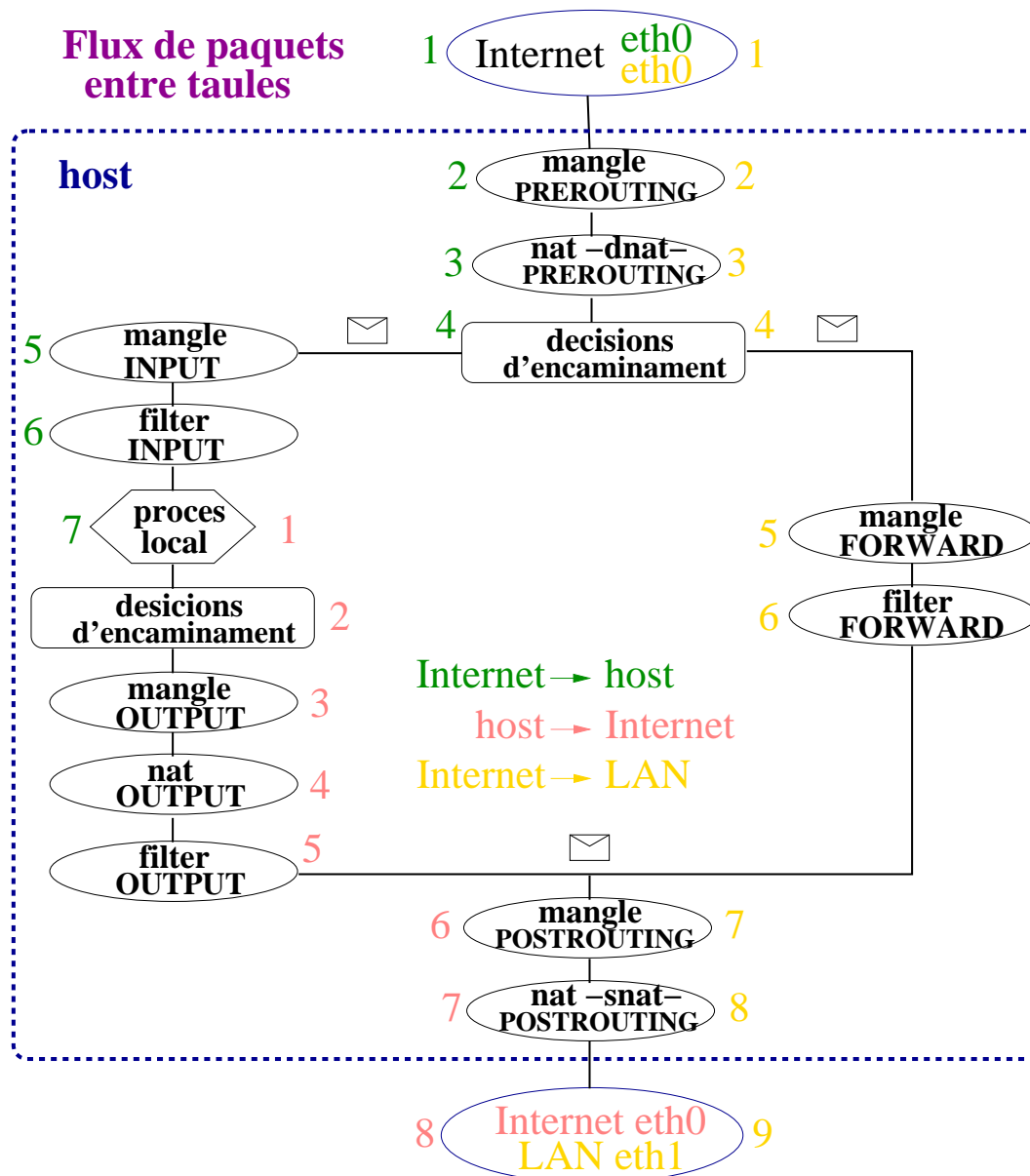
**mangle:** utilitzada per canviar capçaleres dels paquets (p.e. camps TTL, TOS i MARK). Composta per 5 cadenaes: **PREROUTING**, **OUTPUT**, **INPUT**, **FORWARD** i **POSTROUTING**.

## 2.2. Regles

### Cadenes de la taula filter



## 2.2. Regles



---

## 2.2. Regles - ordre

---

**-A *cadena*** Afegeix una o més regles al final de la cadena especificada. Ex: iptables

```
-A INPUT -j DROP
```

**-D *cadena n*** Esborra la regla número *n* de la cadena especificada. Les regles, en cada cadena, s'enumeren en ordre creixent (començant amb 1). Ex: iptables

```
-D INPUT 1
```

**-I *cadena n*** Insereix una regla en la cadena amb número *n*. Ex: iptables -I

```
INPUT 1 -p TCP --dport 80 -j DROP
```

**-R *cadena n*** Reemplaça la regla amb posició *n* de la cadena. Ex: iptables -R

```
INPUT 1 -s 172.16.1.1 -j DROP
```

**-F [*cadena*]** Esborra totes les regles [de la cadena especificada]. Ex: iptables

```
-F INPUT
```

---

## 2.2. Regles - ordre

---

**-L [cadena]** Llista totes les regles [de la cadena especificada]. Ex: `iptables -L INPUT`; `iptables -t nat -L`

**-Z [cadena]** Posa a zero els comptadors de paquets [de la cadena especificada].  
Ex: `iptables -Z OUTPUT`

**-N *cadena*** Crea una nova cadena amb el nom especificat. Ex: `iptables -N ALLOWED`

**-X [cadena]** Esborra totes les cadenes d'usuari [cadena especificada]. Ex: `iptables -X ALLOWED`

**-P [cadena] acció** Estableix acció per defecte de la cadena. Accions més utilitzades en l'ordre -P: *ACCEPT* i *DROP*. Ex: `iptables -P INPUT DROP`



---

## 2.2. Regles - opció

---

- v** verbose mode. Utilitzada en les ordres -L, -A, -I, -D, i -R. Ex: `iptables -L -v`  
// informació de taules, regles i comptabilitat
- n** adreces IP, ports ... numèriques. Utilitzada en l'ordre -L. Ex: `iptables -L -n`
- x** força que els números de sortida de iptables apareguin amb els seus valors exactes (sense cap aproximació, i.e. K, M o G). Utilitzada en l'ordre -L. Ex: `iptables -L -x`
- *-line-numbers*** mostra els números de línia de les regles. Utilitzada en l'ordre -L. Ex: `iptables -L --line-numbers`
- c [*p b*]** inicialitza comptadors de paquets a p i bytes al afegir o modificar una regla. Utilitzada en l'ordre -A, -I, i -R. Ex: `iptables -I INPUT 1 -s 172.16.1.1 -j DROP -c 2 40`

---

## 2.2. Regles - match

---

**-p [!]*[protocol]*** protocol (*tcp*, *udp* o *icmp*). !  $\equiv$  protocol diferent de l'especificat.

Valor per defecte: tots els protocols. Ex: `iptables -A INPUT -p tcp -j ACCEPT`; `iptables -A INPUT -p ! tcp -j DROP`

**Nota:** ! tcp = udp + icmp

**-s [!]*[adreça]/[màscara]*** adreça origen. Es pot utilitzar una màscara de xarxa. Ex:

`iptables -A INPUT -s 172.16.2.0/24 -j ACCEPT`

**-d [!]*[adreça]/[màscara]*** adreça destí

**-j *acció*** acció (*ACCEPT*, *DROP*, *QUEUE*, *RETURN*, *LOG* ...)

**-i [!]*[nom\_de\_interfície]*** interfície d'entrada (*ppp0*, *lo*, *eth0*, *eth1*, *eth+*). Ex: `iptables`

`-A INPUT -i lo -j ACCEPT`

---

## 2.2. Regles - match

---

**-o [!]nom\_de\_interfície** interfície de sortida (ppp0, lo, eth0, eth1, eth+). Ex: `iptables -A OUTPUT -o eth0 -j ACCEPT`

**-f** s'aplica al segon i restants fragments d'un paquet (no al primer fragment). " **! -f**" indica el primer fragment (capçalera) i/o paquets no fragmentats.

**MTU (Maximum Transfer Unit)** defineix la mida màxima d'un paquet (normalment 1500 B: 20 B capçalera IP, 20 B capçalera TCP, dades  $\leq$  1460B). El protocol IP és l'encarregat de dividir els paquets en fragments de mida  $<$  MTU. El segon i successius fragments no porten, entre d'altres, ni el port origen ni el destinatari. Per tant, només es coneixen els del primer fragment. Aquest fet dificulta el seu filtratge i/o comptabilitat.

Per assegurar-nos que capturem el segon i posteriors fragments, podem utilitzar una regla com aquesta:

```
iptables -A FORWARD -i eth0 -p tcp -f -j DROP
```

---

## 2.2. Regles - match extensions de protocol (-p)

---

Extensions TCP (o UDP) utilitzades amb `-p tcp` (o `-p udp`):

- **-sport [!] [port[:port]]** especifica port origen. El signe ":" identifica un rang de valors (p.e., 20:25, indica ports 20→25, ambdós inclosos). El signe ! és un negador. El rang "2:" indica ports 2→65535 (últim port). Ex: `iptables -A INPUT -p tcp --sport 22:80 -j ACCEPT`; `iptables -A INPUT -p tcp --sport ! 22:80 -j DROP`
- **-dport [!] [port[:port]]** especifica port destinatari. Format igual que `-sport`

Extensions solament per TCP, utilitzades amb `-p tcp`:

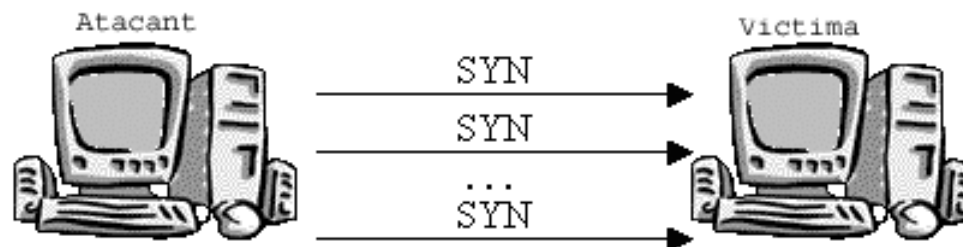
- [!] - **-syn** especifica concordança solament amb els paquets amb flag SYN  $\equiv$  1 (1<sup>er</sup> paquet d'una connexió TCP). Només TCP. Ex: `iptables -A INPUT -p tcp --syn -j DROP`
- [!] - **-tcp-flags** Flags SYN, ACK, FIN, RST, URG, PSH, ALL, NONE (en capçalera TCP). Ex: `iptables -A INPUT -p tcp --tcp-flags SYN,RST,ACK SYN -j DROP` // descarta paquets amb RST=ACK=0 i SYN=1

### Establiment d'una connexió - Protocol TCP



- **SYN:** petició de connexió (protocol TCP).
- **ACK:** justificant de recepció.

### Establiment d'una connexió - Protocol TCP



---

## 2.2. Regles - match extensions de protocol (-p)

---

- Extensions ICMP utilitzades amb `-p icmp`:

- ***-icmp-type [!] tipus*** especifica el tipus de missatge ICMP que concordi amb aquesta regla. ***tipus*** vàlids: echo-reply (pong), destination-unreachable, echo-request (ping), router-advertisement, router-solicitation, time-exceeded, parameter-problem.

Veure llista amb `"iptables -p icmp -h"`. Un tipus pot estar associat a varis sub-tipus; p.e.: destination-unreachable = host-unreachable, port-unreachable, etc. Ex:

```
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

---

## 2.2. Regles - altres extensions match (-m)

---

- Extensions MAC utilitzades amb `-m mac`:

- ***-mac-source [!] address*** especifica l'adreça MAC (Media Access Control. Nombre que identifica de forma única cada dispositiu ethernet) del host emissor que concordi amb aquesta regla. Ex:

```
iptables -A INPUT -m mac --mac-source 00:00:00:00:03:08 -j DROP
```

- Extensions MARK utilitzades amb `-m mark`:

- ***-mark #*** match de paquets prèviament marcats amb "`-j MARK`". Útil en comptabilitat IP i filtratge. Ex:

```
iptables -A INPUT -m mark --mark 1 -j DROP
```

---

## 2.2. Regles - altres extensions match (-m)

---

Extensions multiport utilitzades amb `-p tcp (udp) -m multiport`:

- ***-source-port port1,..., portN*** llista de ports origen. Ex:

```
iptables -A INPUT -p tcp -m multiport --source-port 22,53,80 -j  
DROP
```

- ***-destination-port port1,..., portN*** llista de ports destí. Ex:

```
iptables -A INPUT -p tcp -m multiport --destination-port 22,53,80  
-j DROP
```

- ***-port port1,..., portN*** llista de ports origen + destí. Ex:

```
iptables -A INPUT -p tcp -m multiport --port 22,53,80 -j DROP
```



---

## 2.2. Regles - altres extensions match (-m)

---

Extensions d'estat utilitzades amb `-m state`:

- ***-state stat1,...,statN*** match si la connexió es troba en un estat de la llista. Si no s'especifica el protocol (amb "-p"), es consideren tots els protocols (tcp, udp i icmp). Estats:

**INVALID** paquet no associat a cap flux o connexió

**ESTABLISHED** paquet pertanyent a una connexió plenament establerta

**RELATED** paquet que vol iniciar una nova connexió relacionada amb un altra connexió plenament establerta (ESTABLISHED)

**NEW** paquet corresponent a una nova connexió

**Ex:** `iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT`

---

## 2.2. Regles - altres extensions match (-m)

---

- Extensions **tos** utilitzades amb `-m tos`:

- **-tos tos\_valor** match amb type of service (tos) tos\_valor. Si no s'especifica el protocol (amb "-p") vol dir tots els protocols (tcp, udp i icmp) Ex:

```
iptables -A INPUT -m tos --tos maximize-throughput -j DROP
```

- Extensions **ttl** utilitzades amb `-m ttl`:

- **-ttl ttl\_valor** match amb time to life (ttl) dels paquets. Si no s'especifica el protocol (amb "-p") vol dir tots els protocols (tcp, udp i icmp) Ex:

```
iptables -A INPUT -p tcp -m ttl --ttl 60 -j DROP
```

**NOTA:** TOS i TTL estan explicats en la secció **accions** (tot seguit)

---

## 2.2. Regles - accions (utilitzades p.e. amb -P o -j)

---

**ACCEPT** accepta. Ex: `iptables -A INPUT -i eth0 -j ACCEPT`

**DROP** descarta. Ex: `iptables -A INPUT -i eth0 -j DROP`

**QUEUE** enllista paquet en l'espai d'usuari per un posterior processament. Ex:  
`iptables -A INPUT -i eth0 -j QUEUE`

**RETURN** retorna a la següent regla de la cadena que ha cridat la cadena actual.  
Ex:

```
iptables -N NOVA_CADENA
```

```
iptables -A NOVA_CADENA -i eth0 -j ACCEPT
```

```
iptables -A NOVA_CADENA -i lo -j RETURN
```

```
iptables -A INPUT -j NOVA_CADENA
```

```
iptables -A INPUT -i lo -p tcp -j DROP
```

```
iptables -A INPUT -i lo -p ! tcp -j ACCEPT
```

---

## 2.2. Regles - accions

---

**LOG** activa el logging de paquets (no acaba el processat de regles). Per llegir-los utilitzar `dmesg`. Opcions addicionals:

- **-log-level *nivell*** nivell de logging (veure `syslog.conf(5)`).
- **-log-prefix *prefix*** log amb prefix (fins 29 caràcters) per diferenciar missatges.
- **-log-tcp-sequence** log TCP sequence numbers (perillós).
- **-log-tcp-options** log options from the TCP packet header.
- **-log-ip-options** log options from the IP packet header.

**Ex:** `iptables -A INPUT -p TCP ! --syn -m state --state NEW -j LOG  
--log-prefix "Nou no syn"`

---

## 2.2. Regles - accions

---

**REJECT** utilitzat per retornar un paquet d'error en resposta d'un "matching". Vàlid solament en les cadenes INPUT, FORWARD, OUTPUT i les definides per l'usuari. Les opcions següents controlen el paquet retornat:

**--reject-with *tipus*** on *tipus* pot ser:

**icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable,  
icmp-proto-unreachable, icmp-net-prohibited, icmp-host-prohibited,  
icmp-admin-prohibited**

**Ex:** iptables -A INPUT -p TCP ! --syn -m state --state NEW -j REJECT  
--reject-with admin-prohib

**NOTA:** els tipus s'han d'especificar mitjançant els alias. Obteniu els alias mitjançant

"iptables -J REJECT -h"

---

## 2.2. Regles - accions

---

**MIRROR** inverteix els camps origen i destí del paquet. A continuació retransmet el paquet. Cadenes vàlides: INPUT, FORWARD, PREROUTING i d'usuari

**REDIRECT** altera l'adreça IP destinatària cap al host (i.e. 127.0.0.1). Taula vàlida: nat, en les cadenes PREROUTING, OUTPUT i d'usuari. Opció addicional:

**-to-ports *port[-port]*** readreçament del port destinatari. Vàlida només si en la regla també s'especifa -p tcp or -p udp. Ex: iptables -t nat -A PREROUTING -p TCP --dport 80 -j REDIRECT --to-ports 8080

**MARK** serveix per marcar paquets (útil en filtratge i comptabilitat IP). Marques vàlides només dins del host. Si es vol veure les marques en altres hosts utilitzar l'acció TOS. Opció addicional:

**-set-mark #** Ex: iptables -t mangle -A PREROUTING -p tcp --dport 22 -j MARK --set-mark 2

---

## 2.2. Regles - accions

---

**TOS (Type Of Service)** Camp de la capçalera IP. Opcions (generades amb "iptables -j TOS -h"):

**--set-tos *valor*** on valor pot ser:

**Minimize-Delay** 16 (0x10)

**Maximize-Throughput** 8 (0x08)

**Maximize-Reliability** 4 (0x04)

**Minimize-Cost** 2 (0x02)

**Normal-Service** 0 (0x00)

**Ex:** iptables -t mangle -A PREROUTING -p tcp --dport 22 -j TOS  
--set-tos 0x10

---

## 2.2. Regles - accions

---

**TTL (Time To Live)** Camp de la capçalera IP. útil per si volem posar el mateix temps de vida dels paquets de vàries màquines connectades a Internet. De vegades l'ISP sol en vol una. Opcions (generades amb "iptables -j TTL -h"):

**-ttl-set** value Set TTL to <value>

**-ttl-dec** value Decrement TTL by <value>

**-ttl-inc** value Increment TTL by <value>

**Ex:** iptables -t mangle -A PREROUTING -i eth0 -j TTL  
--ttl-set 0x10



---

## 2.2. Regles - accions

---

**DNAT (Destination NAT)** reescriu l'adreça IP destinatària d'un paquet (INTERNET → LAN). Sol es pot utilitzar en les cadenes PREROUTING i OUTPUT de la taula `nat`, Opció: `--to-destination adreça_IP`

**Ex.:** `iptables -t nat -A PREROUTING -p tcp -d 80.30.40.219 --dport 80 -j DNAT --to-destination 172.16.0.5`

**SNAT (Source NAT) i MASQUERADE** reescriu l'adreça IP origen d'un paquet (LAN → INTERNET). Sol es pot utilitzar en les cadenes PREROUTING i OUTPUT de la taula `nat`. La diferència està en que MASQUERADE s'utilitza per connexions via mòdem o DHCP

---

## 2.2. Regles - accions

---

**Opció MASQUERADE:** `--to-ports llista_ports`

**Ex.:** `iptables -t nat -A POSTROUTING -p tcp -j MASQUERADE  
--to-ports 1024-31000`

**Opció SNAT:** `--to-ports llista_ports`

**Ex.:** `iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT  
--to-source 80.37.50.219:1024-3200`

**Filtrat IP:** mecanisme que decideix quin tipus de paquets IP seran processats normalment i quins no. És un tipus de firewall. En aquí en veurem un exemple: IPTABLES.

### **Criteris de filtratge:**

- Tipus de protocol: TCP, UDP, ICMP, etc.
- Adreça origen paquet (IP).
- Adreça destí paquet (IP).

---

### 2.3.1. Filtrat IP - localhost (lo) - ping

---

```
# ping 127.0.0.1
```

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
```

```
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.1 ms
```

```
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.0 ms
```

```
— 127.0.0.1 ping statistics —
```

```
2 packets transmitted, 2 packets received, 0% packet loss
```

```
round-trip min/avg/max = 0.0/0.0/0.1 ms
```

```
# iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP
```

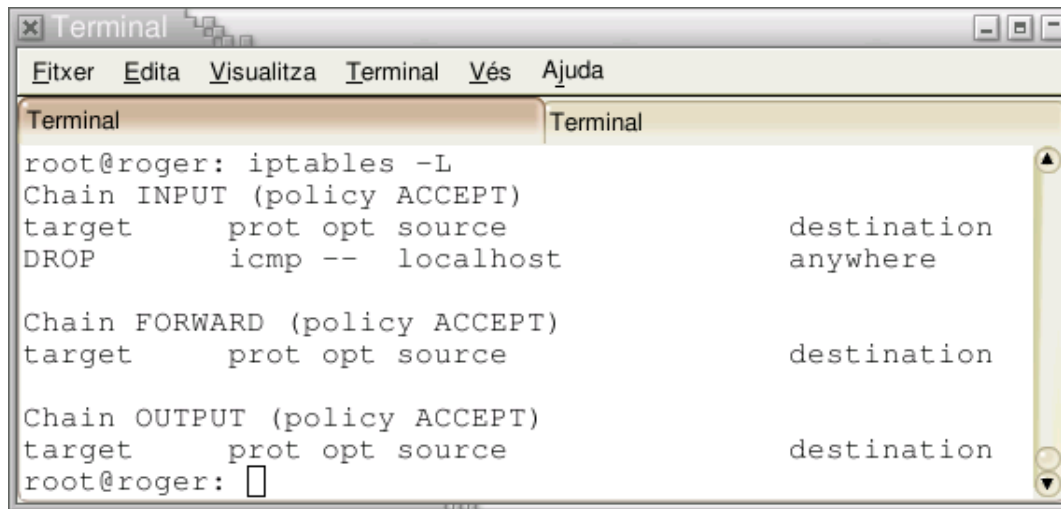
```
# ping 127.0.0.1
```

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
```

```
— 127.0.0.1 ping statistics —
```

```
5 packets transmitted, 0 packets received, 100% packet loss
```

## 2.3.1. Filtrat IP - localhost (lo) - ping

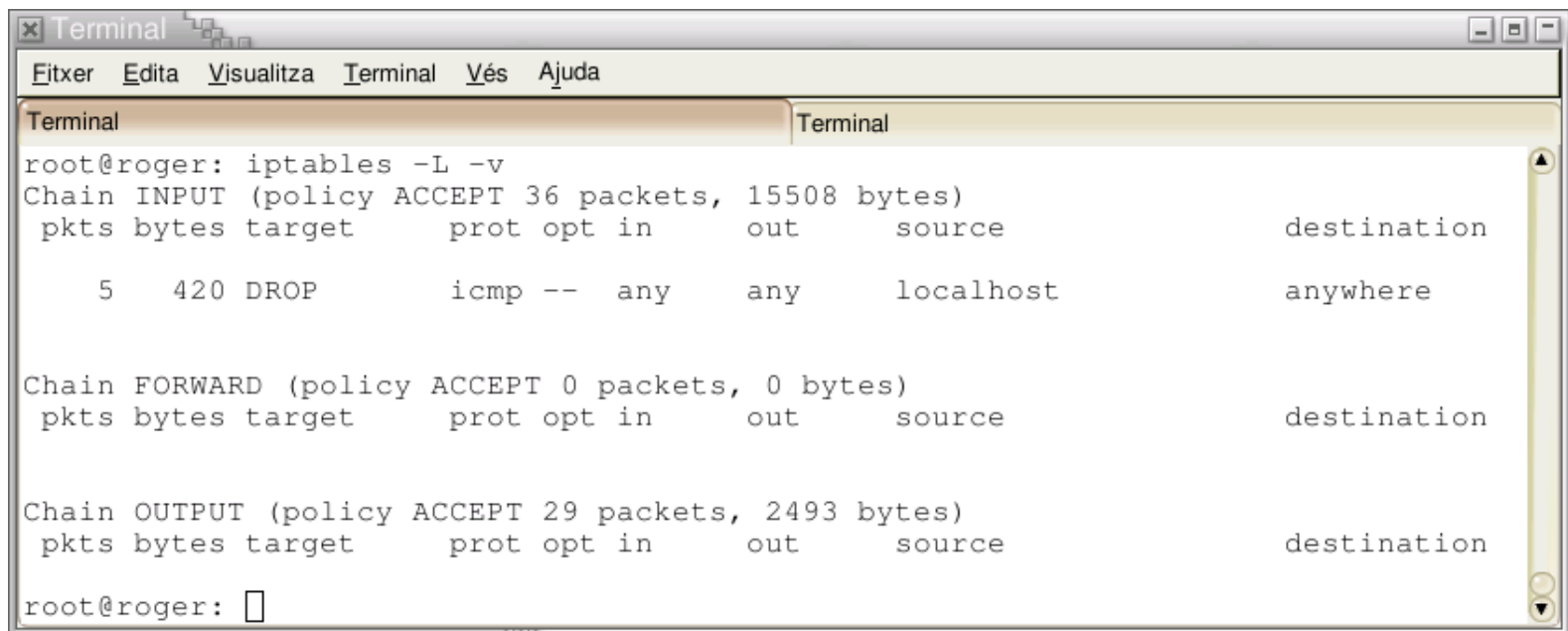


A terminal window titled "Terminal" with a menu bar containing "Fitxer", "Edita", "Visualitza", "Terminal", "Vés", and "Ajuda". The terminal shows the output of the command "iptables -L". The output lists three chains: INPUT, FORWARD, and OUTPUT, all with a policy of ACCEPT. The INPUT chain has a rule for ICMP traffic from localhost to anywhere, with a target of DROP. The FORWARD and OUTPUT chains have no rules listed.

```
root@roger: iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP      icmp -- localhost            anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@roger: 
```



A terminal window titled "Terminal" with a menu bar containing "Fitxer", "Edita", "Visualitza", "Terminal", "Vés", and "Ajuda". The terminal shows the output of the command "iptables -L -v". The output provides statistics for the chains and rules. The INPUT chain has 36 packets and 15508 bytes. The FORWARD chain has 0 packets and 0 bytes. The OUTPUT chain has 29 packets and 2493 bytes. The INPUT chain rule for ICMP traffic from localhost to anywhere has 5 packets and 420 bytes.

```
root@roger: iptables -L -v
Chain INPUT (policy ACCEPT 36 packets, 15508 bytes)
 pkts bytes target     prot opt in     out     source            destination
    5  420 DROP      icmp -- any    any    localhost        anywhere

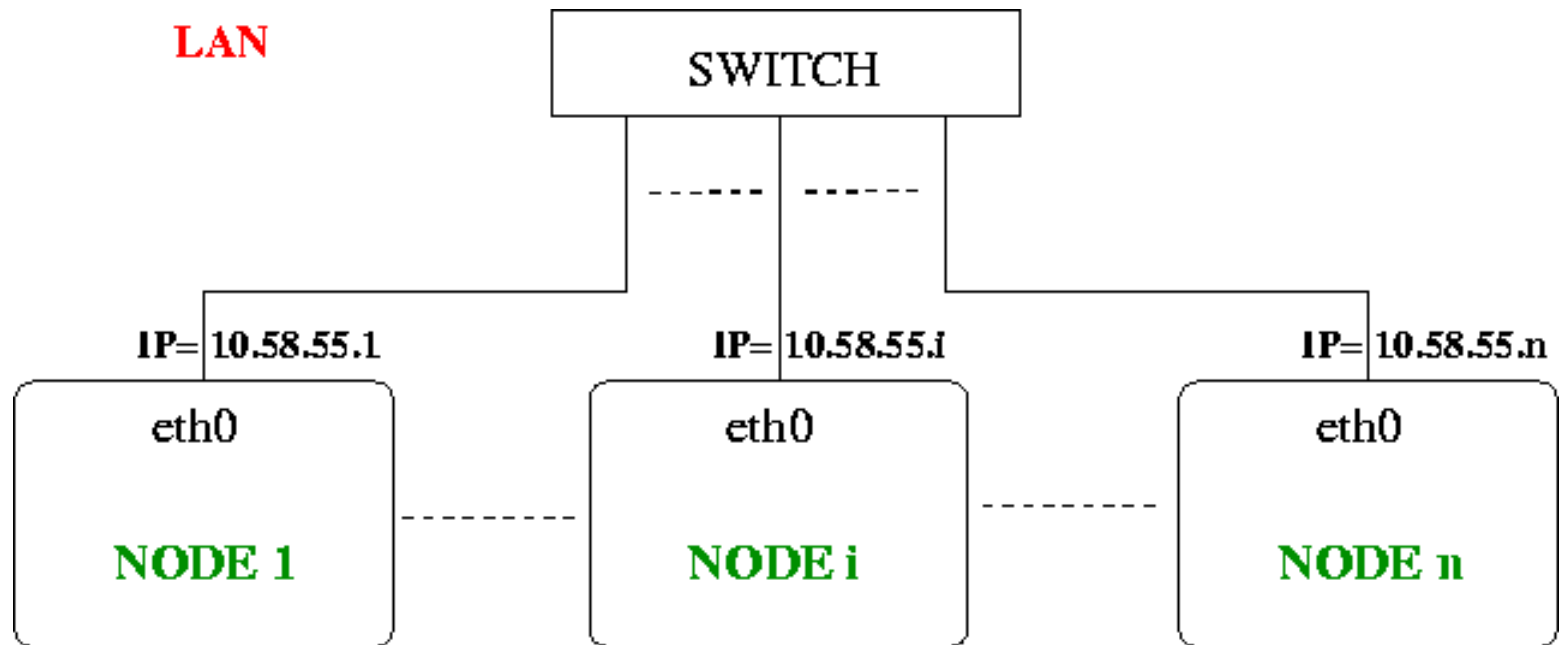
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source            destination

Chain OUTPUT (policy ACCEPT 29 packets, 2493 bytes)
 pkts bytes target     prot opt in     out     source            destination
root@roger: 
```

---

### 2.3.1. Filtrat IP - Firewall senzill (node LAN)

---



---

### 2.3.1. Filtrat IP - Firewall senzill (node LAN)

---

```
#!/bin/sh
# iptables senzill - Fitxer iptables.INPUT1
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
case "$1" in
start)
echo "Iniciant iptables"
### Comencen les regles
# Accio per defecte: ho rebutgem tot
iptables -F
iptables -X
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
# Si volem acceptar connexions iniciades des de fora (p.e. ssh), descomentar la
linia següent
# iptables -A INPUT -p TCP --syn -j ACCEPT
# Acceptar paquets de sessions ja iniciades (localment). P.e. amb mozilla
iptables -A INPUT -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
```

---

### 2.3.1. Filtrat IP - Firewall senzill (node LAN)

---

```
# Paquets UDP i ICMP s'accepten
iptables -A INPUT -p UDP -s 0/0 -j ACCEPT
iptables -A INPUT -p ICMP -j ACCEPT
;;
stop)
echo "Parant iptables"
iptables -F
iptables -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
;;
restart)
$0 stop
$0 start
;;
*)
echo "Usage: iptables {start|stop|restart}"
;;
esac
exit 0
```



## 2.3.1. Filtrat IP - Firewall senzill (node LAN)

```
Terminal
Fitxer  Edita  Visualitza  Terminal  Vés  Ajuda

root@roger: iptables -L
Chain INPUT (policy DROP)
target      prot opt source                destination          state RELATED,ESTABLISHED
ACCEPT      tcp  --  anywhere               anywhere
ACCEPT      udp  --  anywhere               anywhere
ACCEPT      icmp --  anywhere               anywhere

Chain FORWARD (policy ACCEPT)
target      prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target      prot opt source                destination
root@roger: █
```

```
Terminal
Fitxer  Edita  Visualitza  Terminal  Vés  Ajuda

root@roger: iptables -L -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target      prot opt in     out    source        destination
   59 67068 ACCEPT      tcp  --  any    any    anywhere      anywhere
      state RELATED,ESTABLISHED
   17  1507 ACCEPT      udp  --  any    any    anywhere      anywhere
    0     0 ACCEPT      icmp --  any    any    anywhere      anywhere

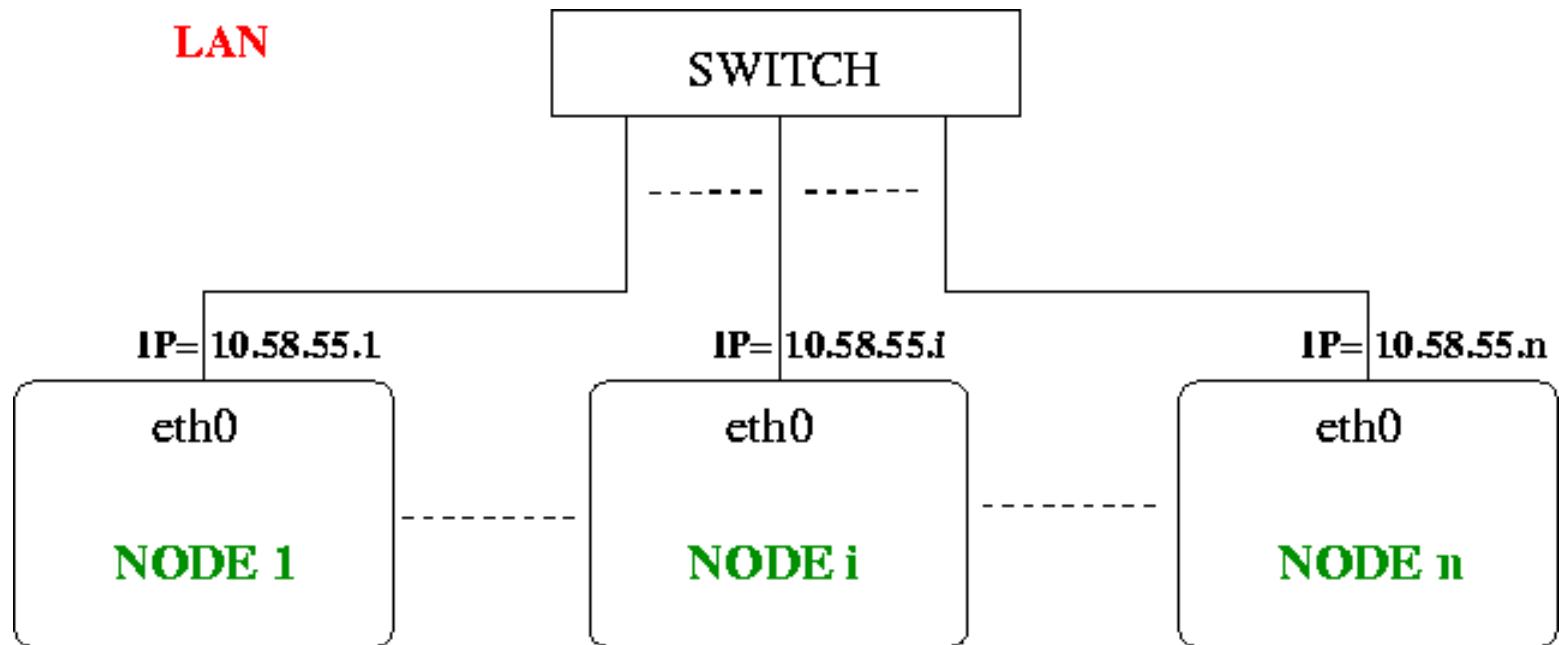
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target      prot opt in     out    source        destination

Chain OUTPUT (policy ACCEPT 70 packets, 6172 bytes)
 pkts bytes target      prot opt in     out    source        destination
root@roger: █
```

---

### 2.3.1. Filtrat IP - Firewall senzill. Nova Cadena -ALLOWED- (node LAN)

---



---

### 2.3.1. Filtrat IP - Firewall senzill. Nova Cadena -ALLOWED- (node LAN)

---

```
#!/bin/sh
```

```
# iptables senzill amb una nova cadena -ALLOWED-
```

```
# Fitxer iptables.INPUT2
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
case "$1" in
```

```
start)
```

```
echo "Iniciant iptables"
```

```
### Comencen les regles
```

```
# Accio per defecte: ho rebutgem tot
```

```
iptables -F
```

```
iptables -X
```

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -P FORWARD ACCEPT
```

```
iptables -N ALLOWED
```

---

### 2.3.1. Filtrat IP - Firewall senzill. Nova Cadena -ALLOWED- (node LAN)

---

# Si no volem acceptar connexions iniciades des de fora (p.e. ssh), comentar la linia següent

```
iptables -A ALLOWED -p TCP --syn -j ACCEPT
```

# Acceptar paquets de sessions ja iniciades (localment). P.e. amb mozilla

```
iptables -A ALLOWED -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
```

# INPUT (TCP) -> ALLOWED

```
iptables -A INPUT -p TCP -j ALLOWED
```

# Paquets UDP i ICMP s'accepten

```
iptables -A INPUT -p UDP -s 0/0 -j ACCEPT
```

```
iptables -A INPUT -p ICMP -j ACCEPT
```

```
::
```

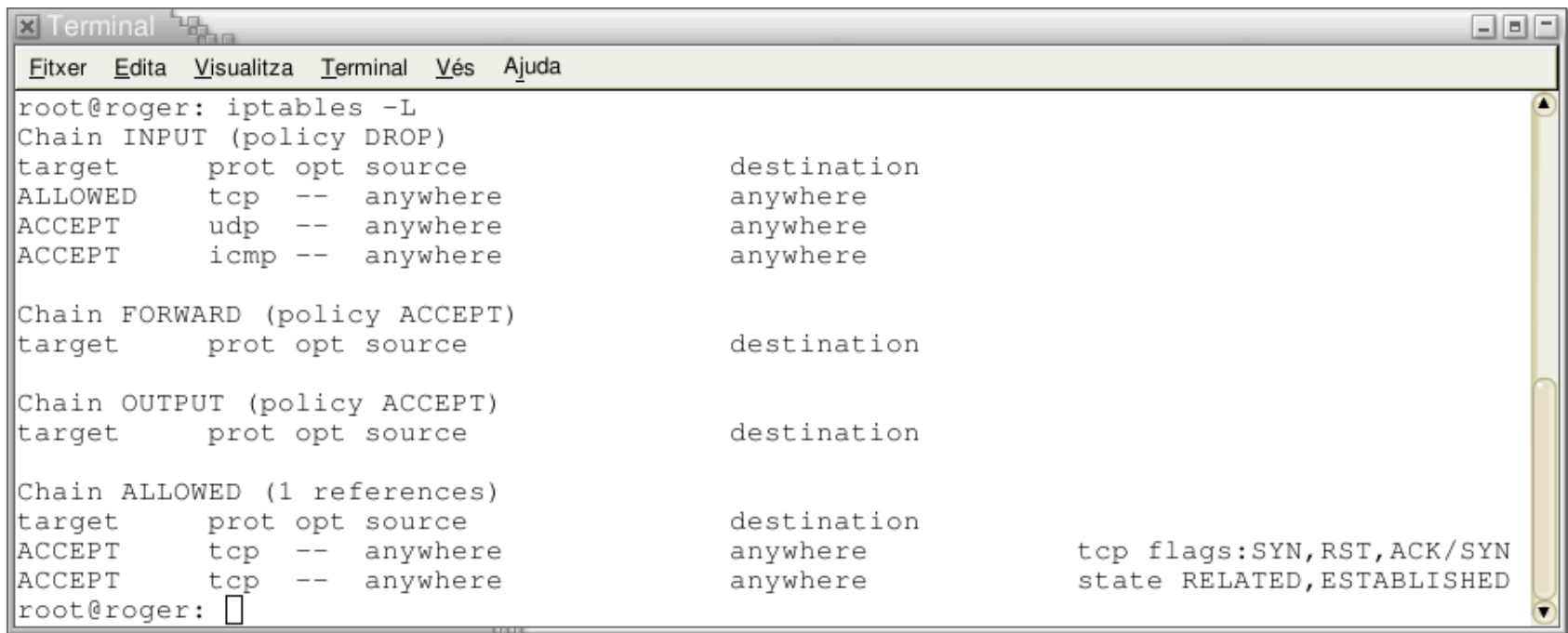
---

### 2.3.1. Filtrat IP - Firewall senzill. Nova Cadena -ALLOWED- (node LAN)

---

```
stop)
echo "Parant iptables"
iptables -F
iptables -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
;;
restart)
$0 stop
$0 start
;;
*)
echo "Usage: iptables {start|stop|restart}"
;;
esac
exit 0
```

## 2.3.1. Filtrat IP - Firewall senzill. Nova Cadena -ALLOWED- (node LAN)



A terminal window titled "Terminal" with a menu bar containing "Fitxer", "Edita", "Visualitza", "Terminal", "Vés", and "Ajuda". The terminal shows the output of the command "iptables -L". The output lists three chains: INPUT (policy DROP), FORWARD (policy ACCEPT), and OUTPUT (policy ACCEPT). Each chain has a table of rules with columns for target, protocol, options, source, and destination. The INPUT chain has three rules: ALLOWED (tcp, --, anywhere, anywhere), ACCEPT (udp, --, anywhere, anywhere), and ACCEPT (icmp, --, anywhere, anywhere). The FORWARD chain has one rule: target (prot, opt, source, destination). The OUTPUT chain has one rule: target (prot, opt, source, destination). The ALLOWED chain has two rules: ACCEPT (tcp, --, anywhere, anywhere) and ACCEPT (tcp, --, anywhere, anywhere). The terminal also shows the command "root@roger: iptables -L" and the prompt "root@roger: ".

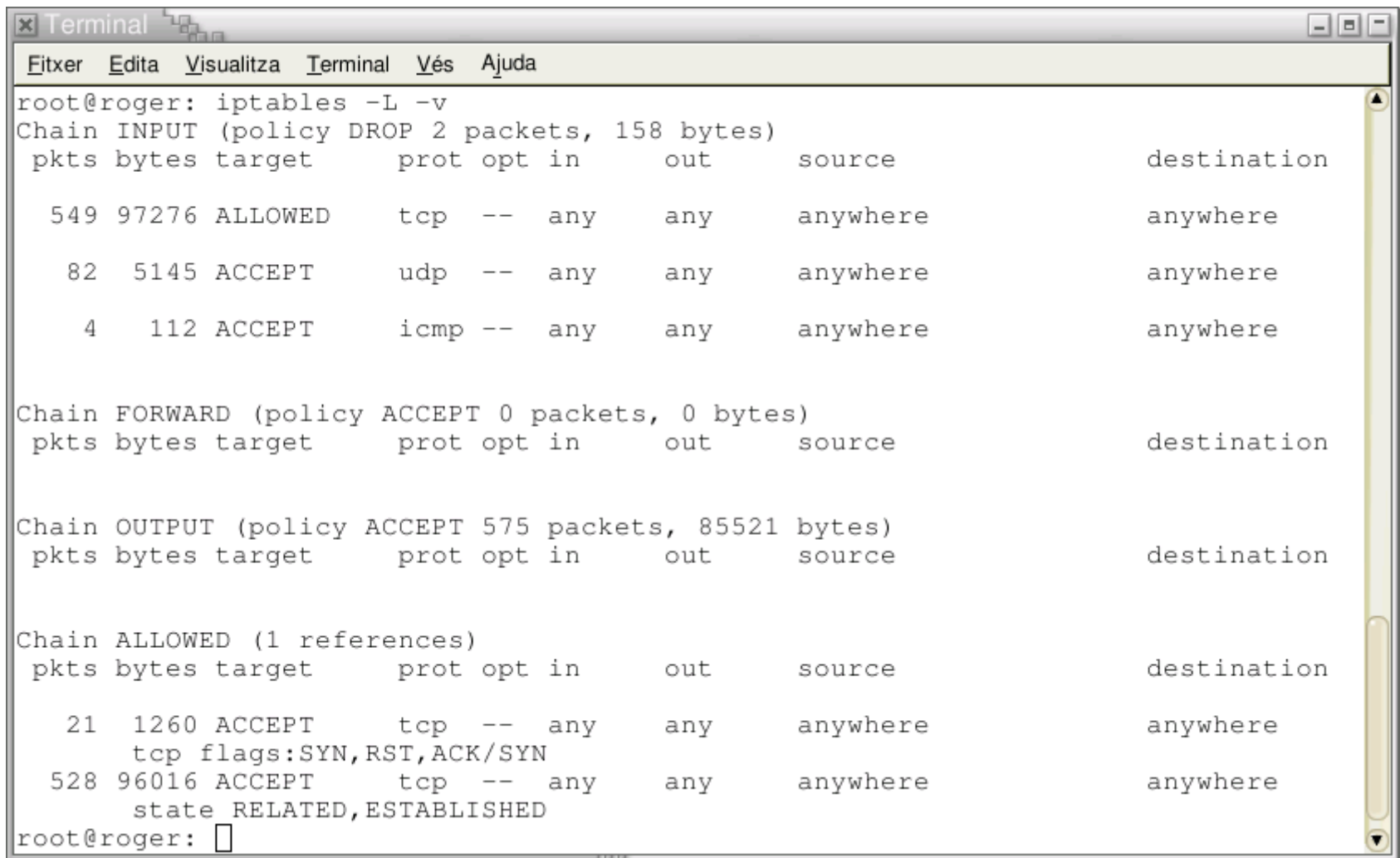
```
root@roger: iptables -L
Chain INPUT (policy DROP)
target      prot opt source                destination
ALLOWED    tcp  --  anywhere              anywhere
ACCEPT      udp  --  anywhere              anywhere
ACCEPT      icmp --  anywhere              anywhere

Chain FORWARD (policy ACCEPT)
target      prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target      prot opt source                destination

Chain ALLOWED (1 references)
target      prot opt source                destination
ACCEPT      tcp  --  anywhere              anywhere      tcp flags:SYN,RST,ACK/SYN
ACCEPT      tcp  --  anywhere              anywhere      state RELATED,ESTABLISHED
root@roger: 
```

## 2.3.1. Filtrat IP - Firewall senzill. Nova Cadena -ALLOWED- (node LAN)

A terminal window titled "Terminal" with a menu bar containing "Fitxer", "Edita", "Visualitza", "Terminal", "Vés", and "Ajuda". The terminal shows the output of the command "iptables -L -v". It displays the configuration for three chains: INPUT, FORWARD, and OUTPUT, and a user-defined chain named ALLOWED. The INPUT chain has a policy of DROP and 2 packets (158 bytes) dropped. It contains three rules: a TCP rule allowing traffic from anywhere to anywhere (549 packets, 97276 bytes), a UDP rule accepting traffic from anywhere to anywhere (82 packets, 5145 bytes), and an ICMP rule accepting traffic from anywhere to anywhere (4 packets, 112 bytes). The FORWARD chain has a policy of ACCEPT and 0 packets (0 bytes) dropped. The OUTPUT chain has a policy of ACCEPT and 575 packets (85521 bytes) dropped. The ALLOWED chain has 1 reference and contains two rules: a TCP rule accepting SYN, RST, and ACK/SYN traffic from anywhere to anywhere (21 packets, 1260 bytes), and a TCP rule accepting traffic in a RELATED, ESTABLISHED state from anywhere to anywhere (528 packets, 96016 bytes). The terminal prompt is "root@roger: " followed by a cursor.

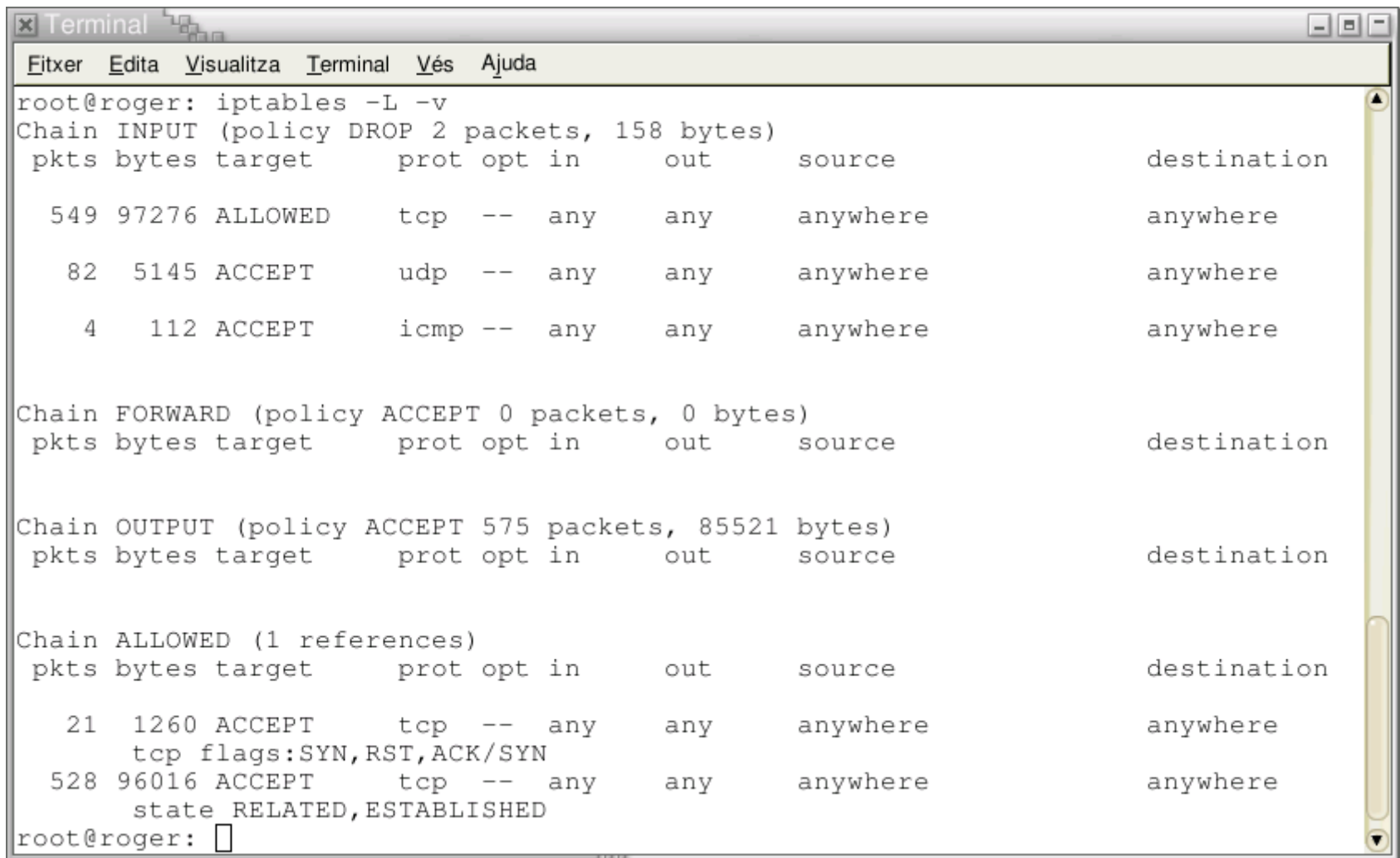
```
root@roger: iptables -L -v
Chain INPUT (policy DROP 2 packets, 158 bytes)
  pkts bytes target     prot opt in     out     source            destination
    549 97276 ALLOWED    tcp  --  any    any     anywhere          anywhere
     82  5145 ACCEPT    udp  --  any    any     anywhere          anywhere
      4   112 ACCEPT    icmp --  any    any     anywhere          anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination

Chain OUTPUT (policy ACCEPT 575 packets, 85521 bytes)
  pkts bytes target     prot opt in     out     source            destination

Chain ALLOWED (1 references)
  pkts bytes target     prot opt in     out     source            destination
     21  1260 ACCEPT     tcp  --  any    any     anywhere          anywhere
        tcp flags:SYN,RST,ACK/SYN
    528 96016 ACCEPT     tcp  --  any    any     anywhere          anywhere
        state RELATED,ESTABLISHED
root@roger: 
```

## 2.3.1. Filtrat IP - Firewall senzill. Nova Cadena -ALLOWED- (node LAN)

A terminal window titled "Terminal" with a menu bar containing "Fitxer", "Edita", "Visualitza", "Terminal", "Vés", and "Ajuda". The terminal shows the output of the command "iptables -L -v". It displays the configuration for three chains: INPUT, FORWARD, and OUTPUT, and a user-defined chain named ALLOWED. The INPUT chain has a policy of DROP and 2 packets (158 bytes) dropped. The FORWARD chain has a policy of ACCEPT and 0 packets (0 bytes) dropped. The OUTPUT chain has a policy of ACCEPT and 575 packets (85521 bytes) dropped. The ALLOWED chain has 1 reference and contains two rules: one for TCP SYN, RST, ACK/SYN packets and another for TCP packets in a RELATED, ESTABLISHED state. The terminal prompt is "root@roger: " followed by a cursor.

```
root@roger: iptables -L -v
Chain INPUT (policy DROP 2 packets, 158 bytes)
 pkts bytes target    prot opt in     out     source    destination
   549 97276 ALLOWED    tcp  --  any    any     anywhere  anywhere
    82  5145 ACCEPT     udp  --  any    any     anywhere  anywhere
     4   112 ACCEPT     icmp --  any    any     anywhere  anywhere

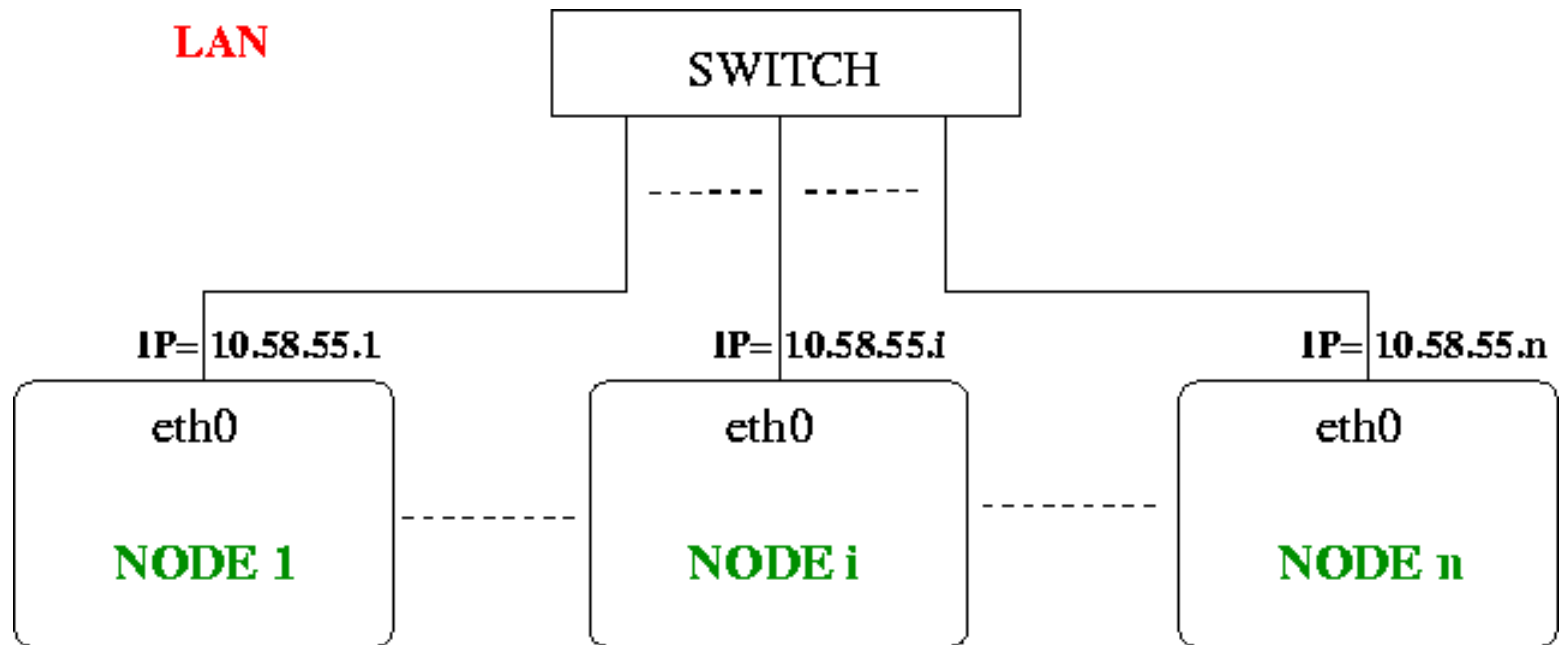
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source    destination

Chain OUTPUT (policy ACCEPT 575 packets, 85521 bytes)
 pkts bytes target    prot opt in     out     source    destination

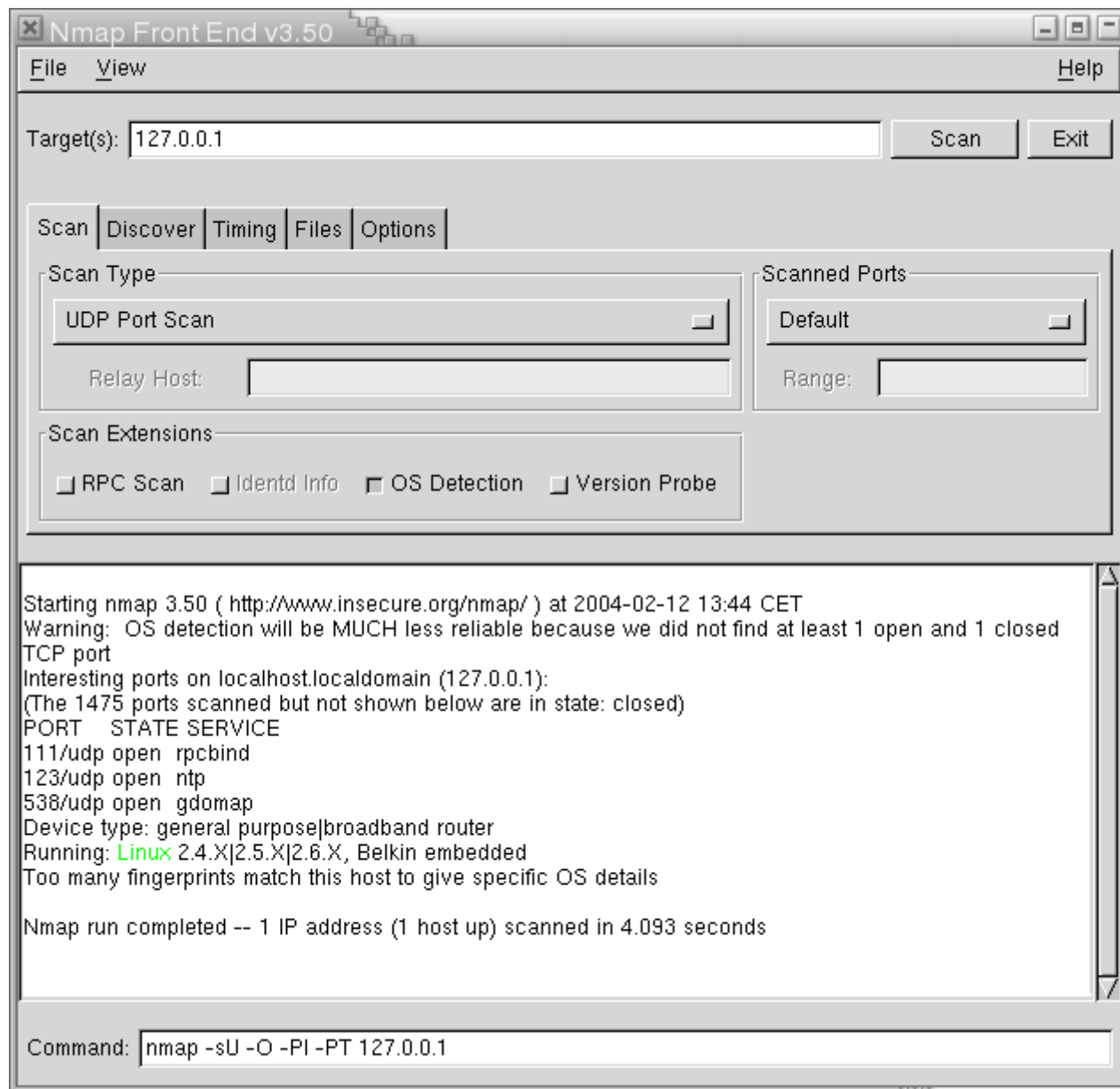
Chain ALLOWED (1 references)
 pkts bytes target    prot opt in     out     source    destination
    21  1260 ACCEPT     tcp  --  any    any     anywhere  anywhere
      tcp flags:SYN,RST,ACK/SYN
   528 96016 ACCEPT     tcp  --  any    any     anywhere  anywhere
      state RELATED,ESTABLISHED
root@roger: 
```



### 2.3.1. Filtrat IP - Firewall més sofisticat - (node LAN)



## 2.3.1. Filtrat IP - Firewall més sofisticat - (node LAN)



---

### 2.3.1. Filtrat IP - Firewall més sofisticat - (node LAN)

---

```
#!/bin/sh
# iptables mes sofisticat
# Fitxer iptables.INPUT3
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
case "$1" in
start)
echo "Iniciant iptables"
### Comencen les regles
# Accio per defecte: ho rebutgem tot
iptables -F
iptables -X
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -N FILTRE_TCP
iptables -N ALLOWED_TCP
```

---

### 2.3.1. Filtrat IP - Firewall més sofisticat - (node LAN)

---

```
iptables -N PAQUETS_TCP
```

```
iptables -N PAQUETS_UDP
```

```
iptables -N PAQUETS_ICMP
```

```
# FILTRE_TCP
```

```
iptables -A FILTRE_TCP -p TCP ! --syn -m state --state NEW -j LOG --log-prefix "Nou  
no syn"
```

```
iptables -A FILTRE_TCP -p TCP ! --syn -m state --state NEW -j REJECT --reject-with  
admin-prohib
```

```
# ALLOWED_TCP
```

```
iptables -A ALLOWED_TCP -p TCP --syn -j ACCEPT
```

```
iptables -A ALLOWED_TCP -p TCP -m state --state ESTABLISHED,RELATED -j  
ACCEPT
```

```
iptables -A ALLOWED_TCP -p TCP -j DROP
```

---

### 2.3.1. Filtrat IP - Firewall més sofisticat - (node LAN)

---

#### # Ports TCP

```
iptables -A PAQUETS_TCP -p TCP --dport 13 -j ALLOWED_TCP
```

```
iptables -A PAQUETS_TCP -p TCP --dport 22 -j ALLOWED_TCP
```

```
iptables -A PAQUETS_TCP -p TCP --dport 25 -j ALLOWED_TCP
```

```
iptables -A PAQUETS_TCP -p TCP --dport 37 -j ALLOWED_TCP
```

```
iptables -A PAQUETS_TCP -p TCP --dport 111 -j ALLOWED_TCP
```

```
iptables -A PAQUETS_TCP -p TCP --dport 113 -j ALLOWED_TCP
```

```
iptables -A PAQUETS_TCP -p TCP --dport 538 -j ALLOWED_TCP
```

```
iptables -A PAQUETS_TCP -p TCP --dport 631 -j ALLOWED_TCP
```

#### # Ports UDP

```
iptables -A PAQUETS_UDP -p UDP --dport 111 -j ACCEPT
```

```
iptables -A PAQUETS_UDP -p UDP --dport 123 -j ACCEPT
```

```
iptables -A PAQUETS_UDP -p UDP --dport 538 -j ACCEPT
```

---

### 2.3.1. Filtrat IP - Firewall més sofisticat- (node LAN)

---

# Paquets ICMP

```
iptables -A PAQUETS_ICMP -p ICMP --icmp-type 8 -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level DEBUG --log-prefix "INPUT em fan ping"
```

```
iptables -A PAQUETS_ICMP -p ICMP --icmp-type 8 -j ACCEPT
```

# INPUT (TCP) -> FILTRE\_TCP

```
iptables -A INPUT -p tcp -j FILTRE_TCP
```

# Si passen el filtre TCP ...

```
iptables -A INPUT -p ALL -d 10.50.54.15 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -A INPUT -p TCP -j PAQUETS_TCP
```

```
iptables -A INPUT -p UDP -j PAQUETS_UDP
```

```
iptables -A INPUT -p ICMP -j PAQUETS_ICMP
```

# altres paquets -> LOG

```
iptables -A INPUT -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level DEBUG --log-prefix "INPUT Paquet mort"
```

::  
;;

---

### 2.3.1. Filtrat IP - Firewall més sofisticat- (node LAN)

---

```
stop)
echo "Parant iptables"
iptables -F
iptables -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
;;
restart)
$0 stop
$0 start
;;
*)
echo "Usage: iptables {start|stop|restart}"
;;
esac
exit 0
```

---

## 2.3.2. Comptabilitat IP

---

**Idea:** compatibilitzar el tràfic de xarxa.

**Sintaxi general:**

**# iptables** *-A cadena especificació-de-regla*



---

### 2.3.2. Comptabilitat IP

---

- Comptabilitat per adreça (tràfic entre ppp0 i les xarxes 172.16.3.0 i 172.16.4.0)

**# iptables -A FORWARD -i ppp0 -d 172.16.3.0/24**

**# iptables -A FORWARD -o ppp0 -s 172.16.3.0/24**

**# iptables -A FORWARD -i ppp0 -d 172.16.4.0/24**

**# iptables -A FORWARD -o ppp0 -s 172.16.4.0/24**

---

### 2.3.2. Comptabilitat IP

---

- Quantes dades viatgen entre els dos departaments?

```
# iptables -A FORWARD -s 172.16.3.0/24 -d 172.16.4.0/24
```

```
# iptables -A FORWARD -s 172.16.4.0/24 -d 172.16.3.0/24
```

- Comptabilitat ping (utilitza el protocol ICMP). ICMP no utilitza ports com ho fa *TCP* i *UDP*.

```
# iptables -A FORWARD -m icmp -p icmp --sports echo-request
```

```
# iptables -A FORWARD -m icmp -p icmp --sports echo-reply
```

```
# iptables -A FORWARD -m icmp -p icmp -f // f: fragments
```

---

### 2.3.2. Comptabilitat IP

---

- Comptabilitat pel Port de Servei (tràfic per l'enllaç ppp0 dels serveis ftp, smtp i web)

***iptables -A FORWARD -i ppp0 -m tcp -p tcp --sport ftp-data:ftp // (20:21)***

***iptables -A FORWARD -o ppp0 -m tcp -p tcp --dport ftp-data:ftp***

***iptables -A FORWARD -i ppp0 -m tcp -p tcp --sport smtp***

***iptables -A FORWARD -o ppp0 -m tcp -p tcp --dport smtp***

***iptables -A FORWARD -i ppp0 -m tcp -p tcp --sport www***

***iptables -A FORWARD -o ppp0 -m tcp -p tcp --dport www***

---

### 2.3.2. Comptabilitat IP

---

- Comptabilitat per protocol (TCP, UDP o ICMP)

**# iptables -A FORWARD -i ppp0 -m tcp -p tcp**

**# iptables -A FORWARD -o ppp0 -m tcp -p tcp**

**# iptables -A FORWARD -i ppp0 -m udp -p udp**

**# iptables -A FORWARD -o ppp0 -m udp -p udp**

**# iptables -A FORWARD -i ppp0 -m icmp -p icmp**

**# iptables -A FORWARD -o ppp0 -m icmp -p icmp**

---

### 2.3.3. Masquerading & Forwarding

---

**Idea:** IP masquerading & Forwarding és el nom que rep un tipus de traducció d'adreces de xarxa (i ports) que permet que totes les màquines d'una xarxa privada utilitzin internet contant amb una única connexió a internet (una única adreça IP).

- Es sol realitzar en un gateway o en un router.
- NAT (Network address translation). SNAT (Source NAT) i DNAT (Destination NAT).
- MASQUERADE es sol utilitzar en connexions via modem o DHCP

---

### 2.3.3. Masquerading & Forwarding - comandes

---

Ordres iptables relacionades amb l'emascament (en el router o gateway):

**# echo 1 >/proc/sys/net/ipv4/ip\_forward** activa l'emascament (reenviament) IP (en un gateway o router)

**# echo 0 >/proc/sys/net/ipv4/ip\_forward** desactiva l'emascament (reenviament) IP

**# iptables -j MASQUERADE** habilita l'emascament

**# iptables -t nat -A PREROUTING -j DROP**

**# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE**

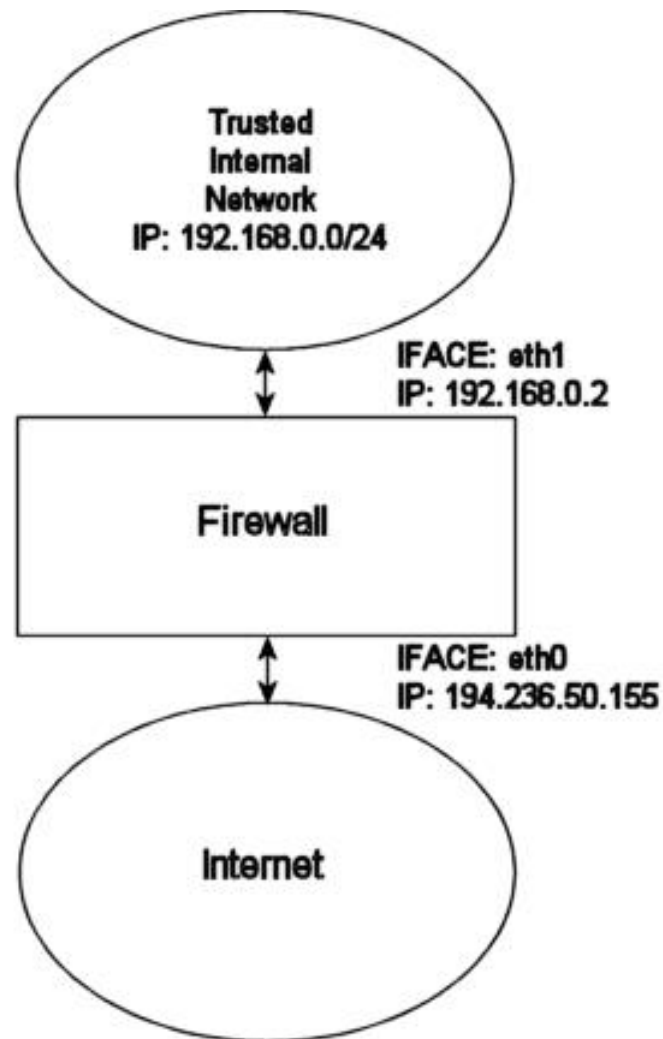
**# iptables -t nat -L**

---

### 2.3.3. Masquerading & Forwarding - Gateway

---

En aquest exemple es veuen les regles d'un host que actua com gateway (Figura de sota). Sol es veuen les regles corresponents a les cadenes FORWARD i OUTPUT.



---

### 2.3.3. Masquerading & Forwarding - Gateway

---

```
#!/bin/sh #
```

```
# # 1.1 Internet Configuration. #
```

```
INET_IP="194.236.50.155" && INET_IFACE="eth0"
```

```
INET_BROADCAST="194.236.50.255"
```

```
# # 1.2 Local Area Network configuration.
```

```
LAN_IP="192.168.0.2" && LAN_IP_RANGE="192.168.0.0/16"
```

```
LAN_IFACE="eth1"
```

```
# # 1.4 Localhost Configuration. #
```

```
LO_IFACE="lo"
```

```
LO_IP="127.0.0.1"
```

```
# # Activació de forwarding
```

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```



---

### 2.3.3. Masquerading & Forwarding - Gateway

---

# # 4.1.1 Set policies #

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT DROP
```

```
iptables -P FORWARD DROP
```

```
iptables -N FILTRE_TCP
```

# # Cadena FILTRE\_TCP #

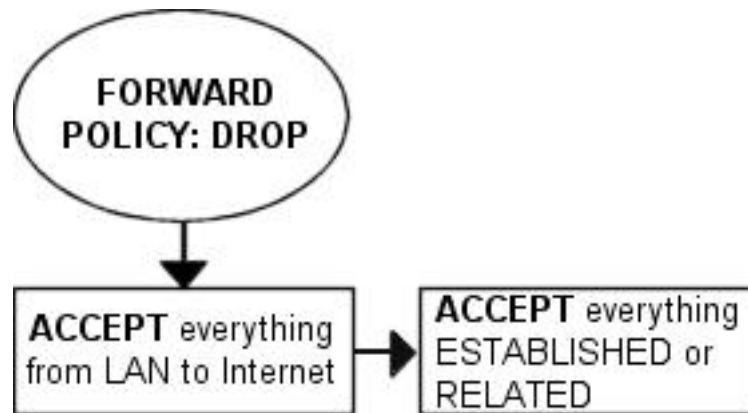
```
iptables -A FILTRE_TCP -p tcp --tcp-flags SYN,ACK SYN,ACK -m state --state NEW  
-j REJECT --reject-with tcp-reset
```

```
iptables -A FILTRE_TCP -p tcp ! --syn -m state --state NEW -j LOG \  
--log-prefix "New not syn:"
```

```
iptables -A FILTRE_TCP -p tcp ! --syn -m state --state NEW -j DROP
```

### 2.3.3. Masquerading & Forwarding - Gateway

**# # Cadena FORWARD #**



**# # Bad TCP packets we don't want #**

```
iptables -A FORWARD -p tcp -j FILTRE_TCP
```

**# # Accept the packets we actually want to forward #**

```
iptables -A FORWARD -i $LAN_IFACE -j ACCEPT
```

```
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

**# # Log weird packets that don't match the above. #**

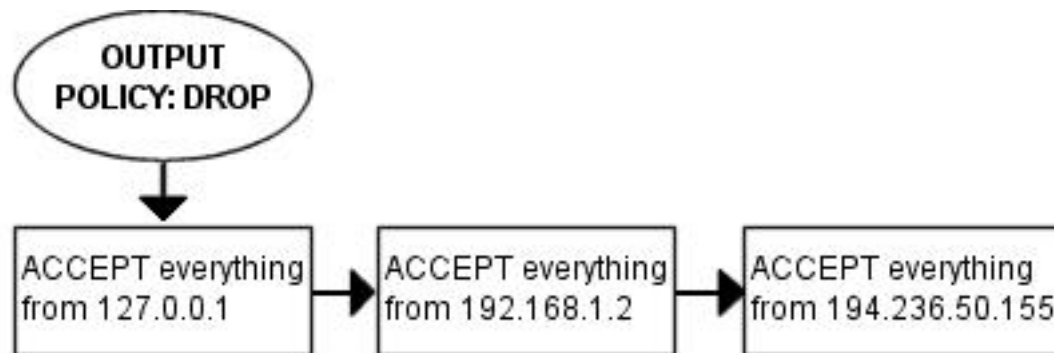
```
iptables -A FORWARD -m limit --limit 3/minute --limit-burst 3 -j LOG \  
--log-level DEBUG --log-prefix "Cadena FORWARD packet died: "
```

---

### 2.3.3. Masquerading & Forwarding - Gateway

---

#### # # 4.1.6 OUTPUT chain #



```
iptables -A OUTPUT -p tcp -j FILTER_TCP
```

```
iptables -A OUTPUT -p ALL -s $LO_IP -j ACCEPT
```

```
iptables -A OUTPUT -p ALL -s $LAN_IP -j ACCEPT
```

```
iptables -A OUTPUT -p ALL -s $INET_IP -j ACCEPT
```

```
# # Log weird packets that don't match the above. #
```

```
iptables -A OUTPUT -m limit --limit 3/minute -j LOG --log-level DEBUG --log-prefix  
"IPT OUTPUT packet died: "
```

---

### 2.3.3. Masquerading & Forwarding - Gateway

---

##### # 4.2 nat table #

# # 4.2.4 PREROUTING chain #

```
iptables -t nat -A PREROUTING -p tcp -d INET_IP --dport 80 -j DNAT --to-destination  
LAN_IP_RANGE
```

# # 4.2.5 POSTROUTING chain #

# # Enable simple IP Forwarding and Network Address Translation #

```
iptables -t nat -A POSTROUTING -o $INET_IFACE -j SNAT --to-source $INET_IP
```