
# RouterOs Firewall

Massimo Nuvoli

TRAINER #TR0368

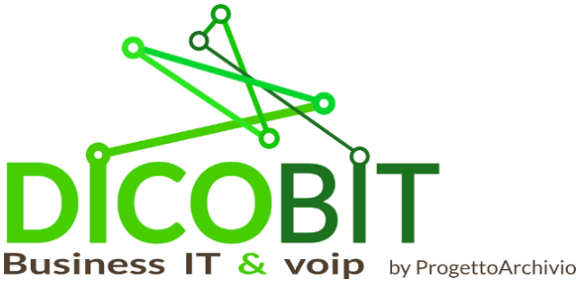
MUM Europe 2017 Milan Italy

# Massimo Nuvoli (maxnuv)

Owner of Progetto Archivio SRL and DICOBIT

System Engineer
System Architect

Please, call me Max!

# First of all..

- at the last Europe MUM..

  my talk was about Switching
  and there was a request

  Please add “hardware spanning tree”

  and from 6.38...

# Switch Hardware Spanning Tree

- Make a switch (as usual)
- Add the master port to a bridge
- Then from the bridge menu IF STP is on then the STP is active on hardware
- Slave ports are shown on the bridge to show the STP status

Look documentation:

https://wiki.mikrotik.com/wiki/Manual:Switch_Chip_Features#Spanning_Tree_Protocol

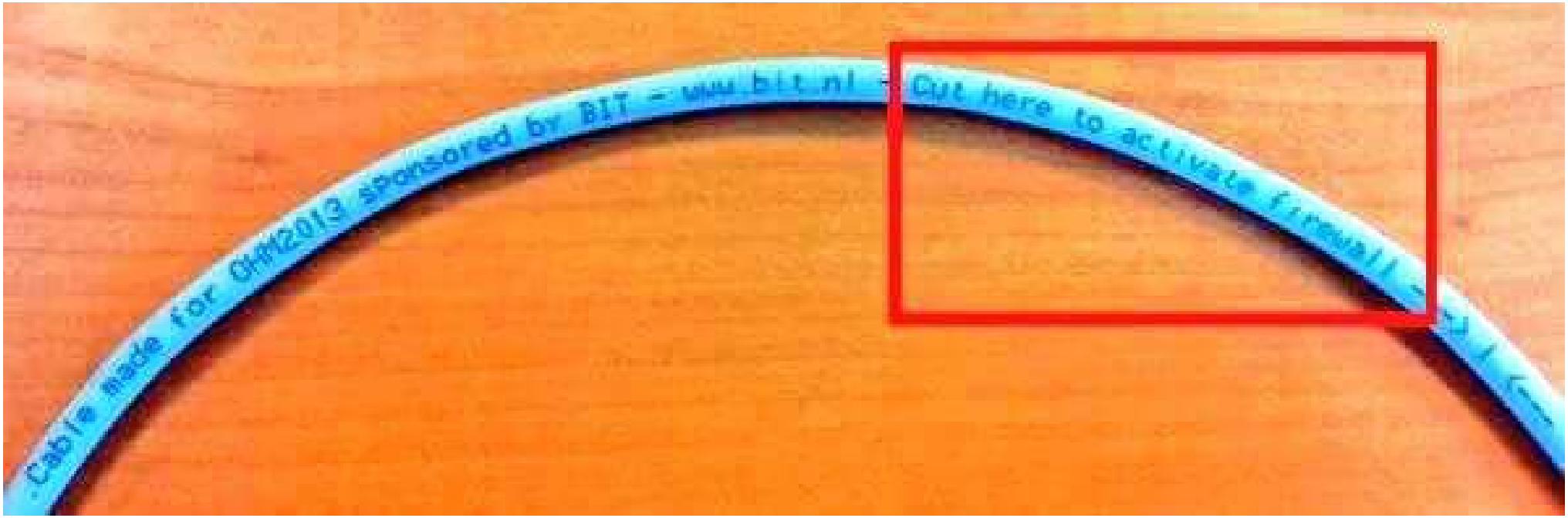
# Today goals

- Know about firewall design in RouterOs
- Know where is, and what to do with
- Changes of the firewall in the last year
- Two examples

# What is a “firewall?”

- Try to isolate the “less protected” outside area from the “more protected” inside area
- It's security device, but own only a firewall is not enough to be protected
- Security is a process, and firewall is only one part of
- The less secure item is between the keyboard and the chair

# Cut here to activate firewall :-)



# Where is “the firewall”

- L2 firewall

  Bridge → Filter

  Switch → Rule or Access List and other
- L3 (and up) firewall IPv4

  IP → Firewall and IP → Web Proxy
- L3 firewall IPv6

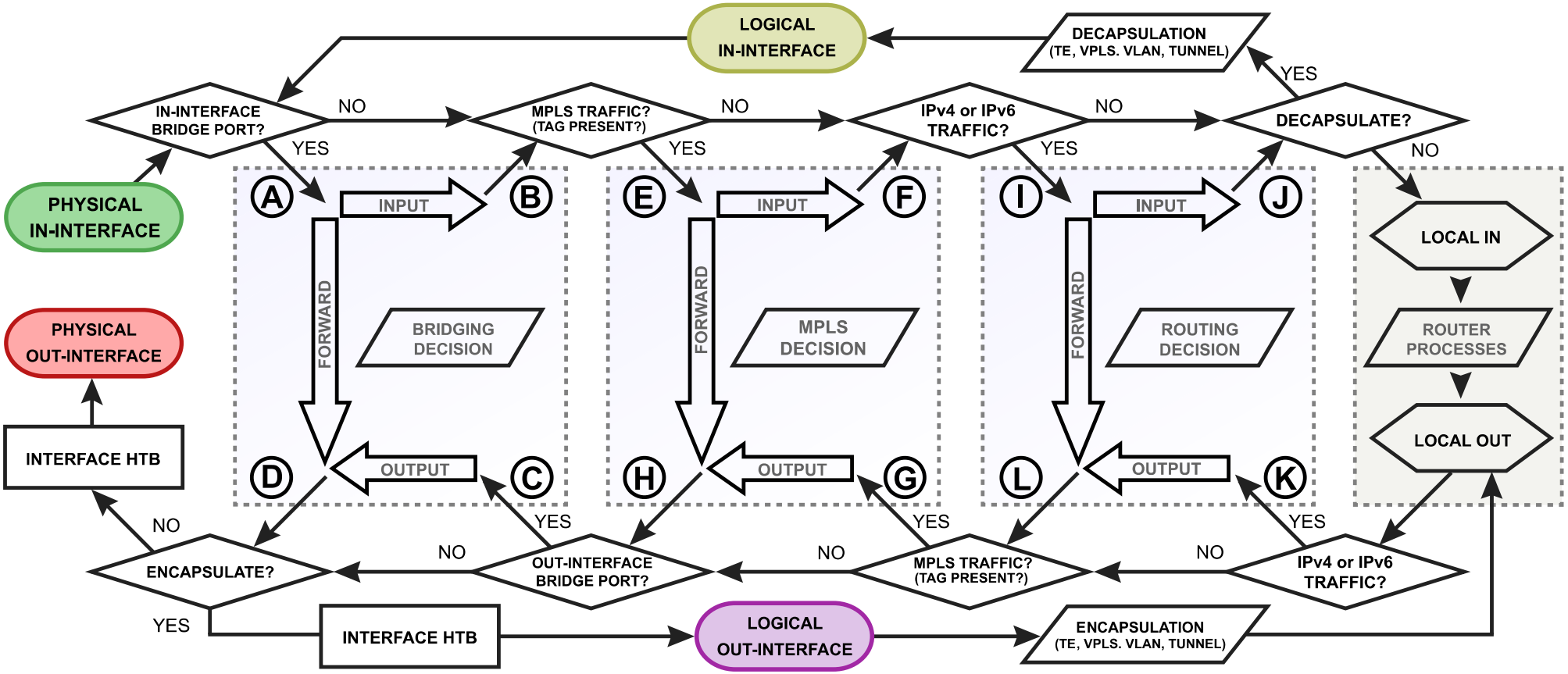
  IPv6 → Firewall

# L2 firewall

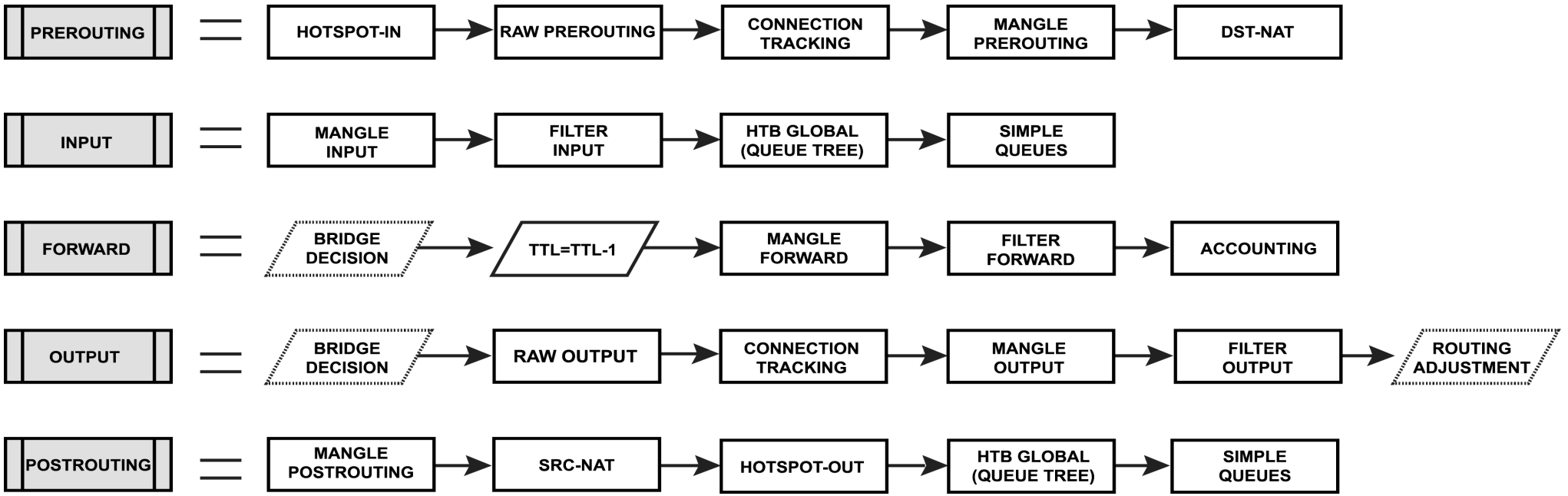
Take the fight at L2, but not only MAC ADDRESS...

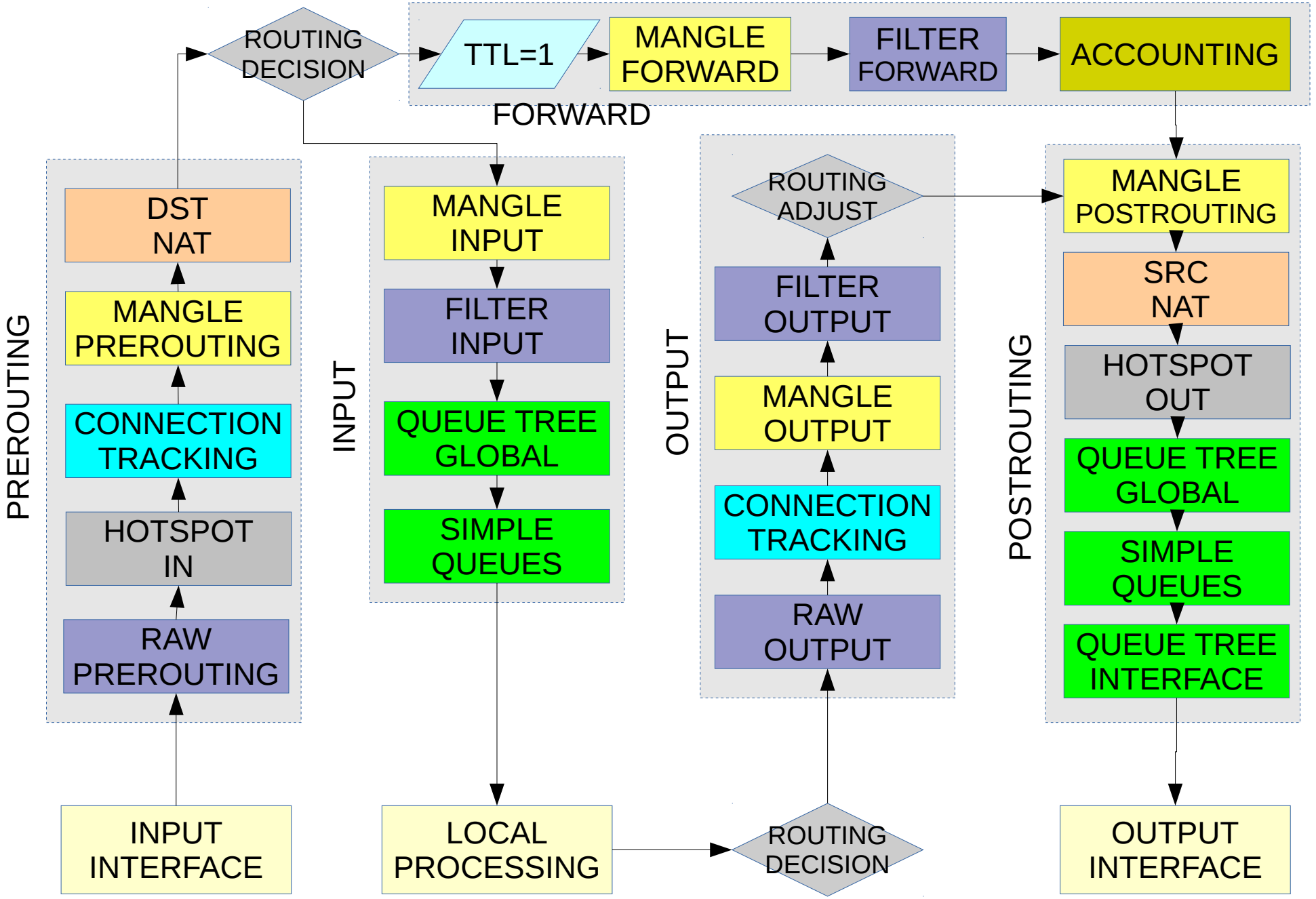
- On switch chipset with ACL (hardware)
- On bridge interface with ACL (software)

# RouterOs Packet Flow 1



# RouterOs Packet Flow 2

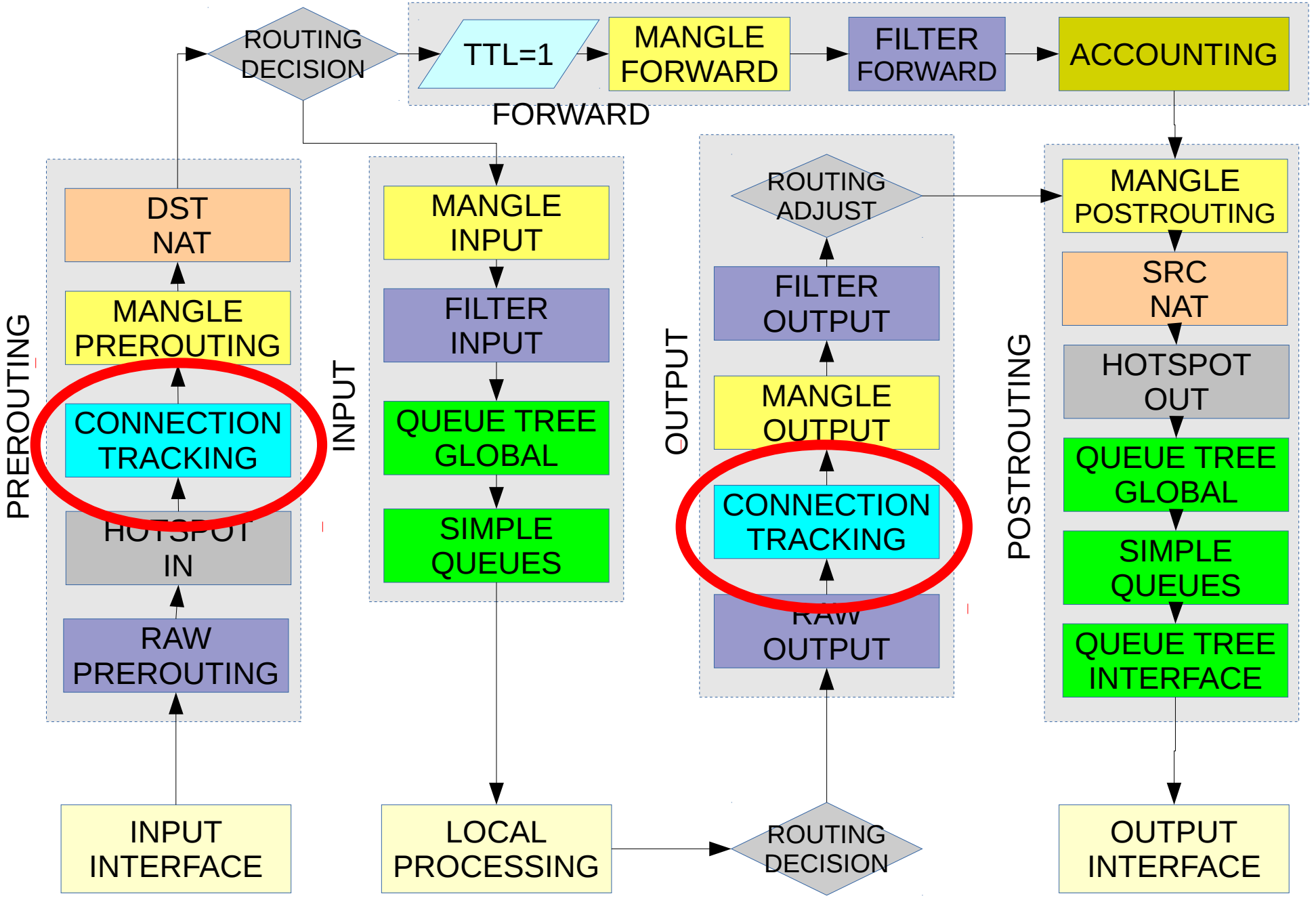



ROUTING DECISION
TTL=1
MANGLE FORWARD
FILTER FORWARD
ACCOUNTING
FORWARD
PREROUTING
DST NAT
MANGLE PREROUTING
CONNECTION TRACKING
HOTSPOT IN
RAW PREROUTING
INPUT
MANGLE INPUT
FILTER INPUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
OUTPUT
ROUTING ADJUST
FILTER OUTPUT
MANGLE OUTPUT
CONNECTION TRACKING
RAW OUTPUT
POSTROUTING
MANGLE POSTROUTING
SRC NAT
HOTSPOT OUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
QUEUE TREE INTERFACE
INPUT INTERFACE
LOCAL PROCESSING
ROUTING DECISION
OUTPUT INTERFACE

# Connection Tracking

- RouterOs can “detect” the status of a connection (TCP/UDP) and try to give us a more powerful way to check packets
- Connection state can be “new” “established” “related” but also “unknown” or “invalid”
- Particular protocols (eg SIP and FTP) needs “connection helpers” to track complex connections

```
/ip firewall connection
```


ROUTING DECISION
TTL=1
MANGLE FORWARD
FILTER FORWARD
ACCOUNTING
FORWARD
PREROUTING
DST NAT
MANGLE PREROUTING
CONNECTION TRACKING
HOTSPOT IN
RAW PREROUTING
INPUT
MANGLE INPUT
FILTER INPUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
OUTPUT
ROUTING ADJUST
FILTER OUTPUT
MANGLE OUTPUT
CONNECTION TRACKING
RAW OUTPUT
POSTROUTING
MANGLE POSTROUTING
SRC NAT
HOTSPOT OUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
QUEUE TREE INTERFACE
INPUT INTERFACE
LOCAL PROCESSING
ROUTING DECISION
OUTPUT INTERFACE

# L3 firewall IPv4 and IPv6

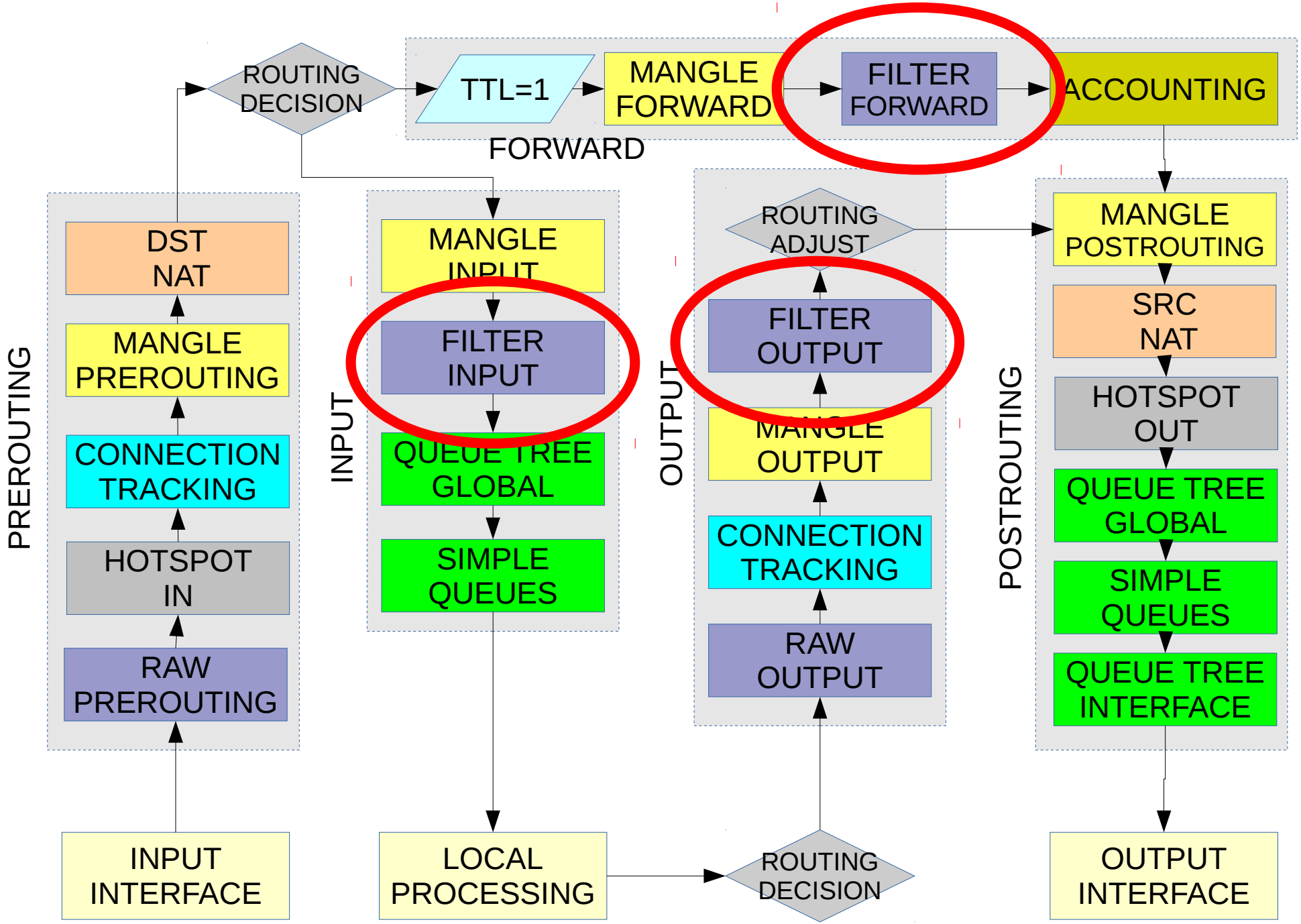
- Packet flow show “where firewall act”
- Each “position” is a “default chain”
- A “chain” is a set of sequential rules, the order IS important
- Check and action are different in each flow position
- You can jump and also return back on a chain

# Filter table

Filter chains can be used to allow and deny connections

- Input
- Output
- Forward

```
/ip firewall filter
/ipv6 firewall filter
```


ROUTING DECISION
TTL=1
MANGLE FORWARD
FILTER FORWARD
ACCOUNTING
FORWARD
PREROUTING
DST NAT
MANGLE PREROUTING
CONNECTION TRACKING
HOTSPOT IN
RAW PREROUTING
INPUT
MANGLE INPUT
FILTER INPUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
OUTPUT
ROUTING ADJUST
FILTER OUTPUT
MANGLE OUTPUT
CONNECTION TRACKING
RAW OUTPUT
POSTROUTING
MANGLE POSTROUTING
SRC NAT
HOTSPOT OUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
QUEUE TREE INTERFACE
INPUT INTERFACE
LOCAL PROCESSING
ROUTING DECISION
OUTPUT INTERFACE

# Default filter table

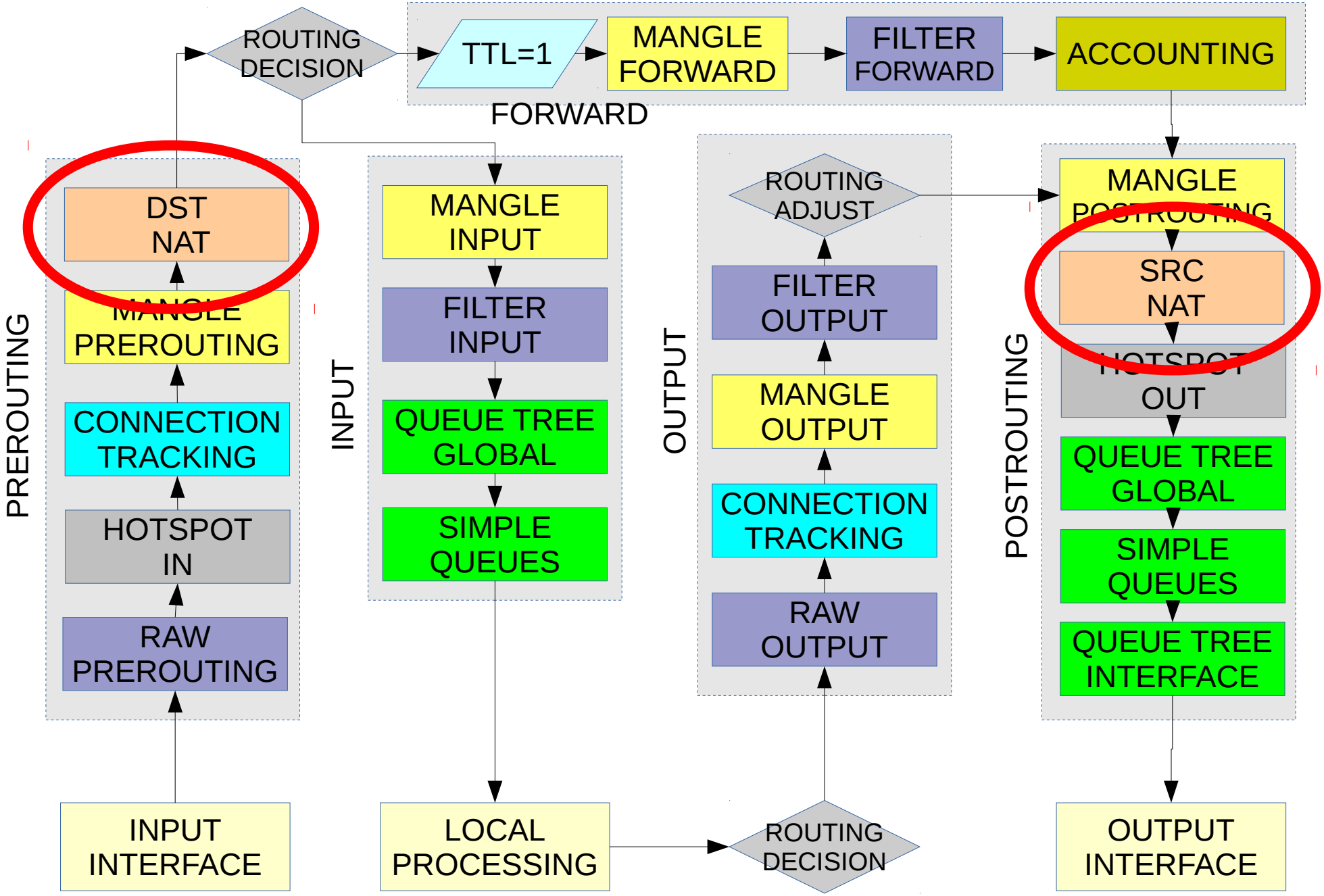
- With connection tracking:
  - accept established/related connections
  - drop invalid connections
  - after we have only “new” connections so no need to check the connection state
  - other rules

# Nat table

In the nat chains we can change address and port of connections, only in IPv4

- src nat
- dst nat

```
/ip firewall nat
```

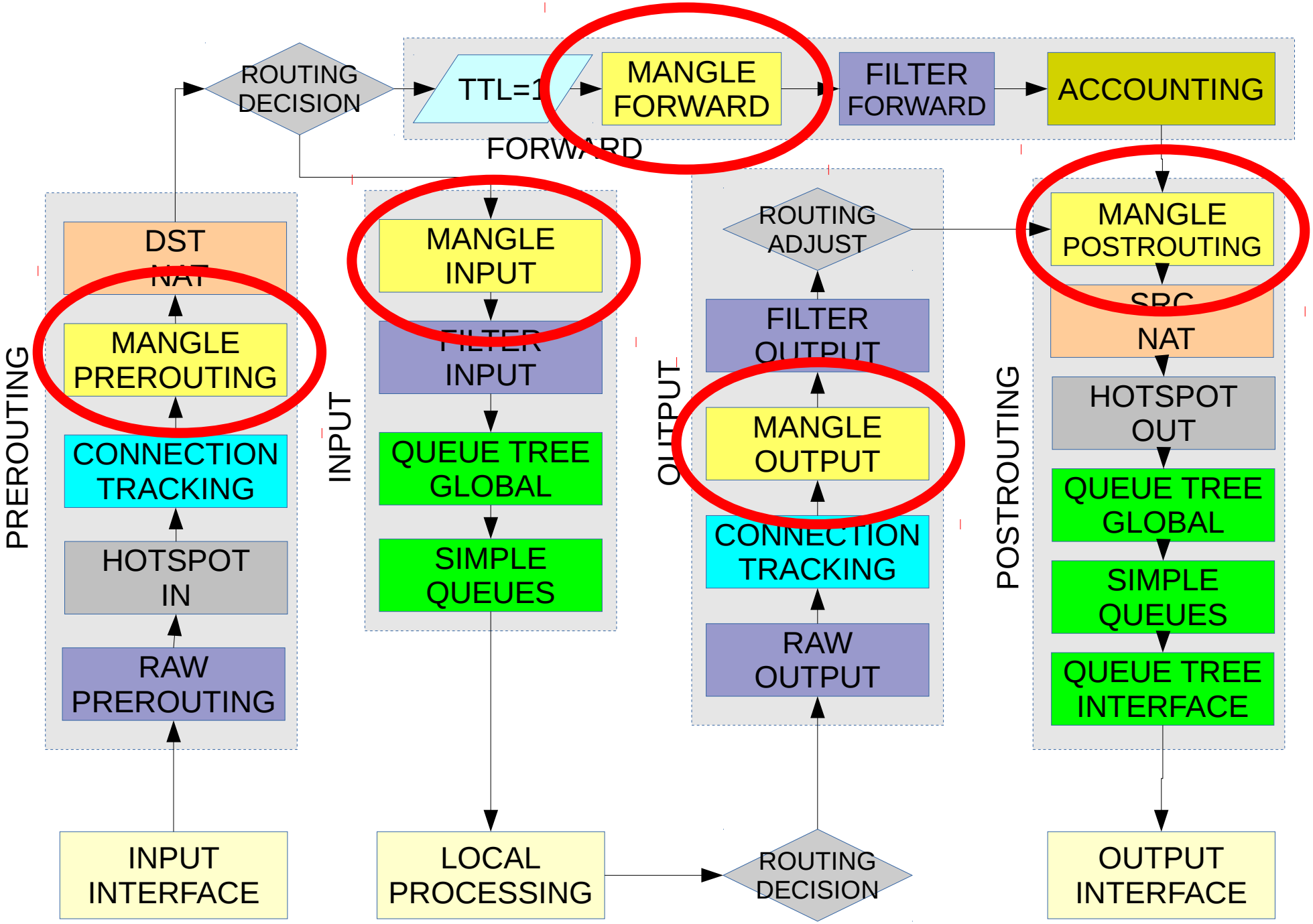

ROUTING DECISION
TTL=1
MANGLE FORWARD
FILTER FORWARD
ACCOUNTING
FORWARD
PREROUTING
DST NAT
MANGLE PREROUTING
CONNECTION TRACKING
HOTSPOT IN
RAW PREROUTING
INPUT
MANGLE INPUT
FILTER INPUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
OUTPUT
ROUTING ADJUST
FILTER OUTPUT
MANGLE OUTPUT
CONNECTION TRACKING
RAW OUTPUT
POSTROUTING
MANGLE POSTROUTING
SRC NAT
HOTSPOT OUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
QUEUE TREE INTERFACE
INPUT INTERFACE
LOCAL PROCESSING
ROUTING DECISION
OUTPUT INTERFACE

# Mangle table

The mangle chain is useful to manage all other detail of a connection (e.g. ttl or qos)

- input
- output
- forward
- prerouting
- Postrouting

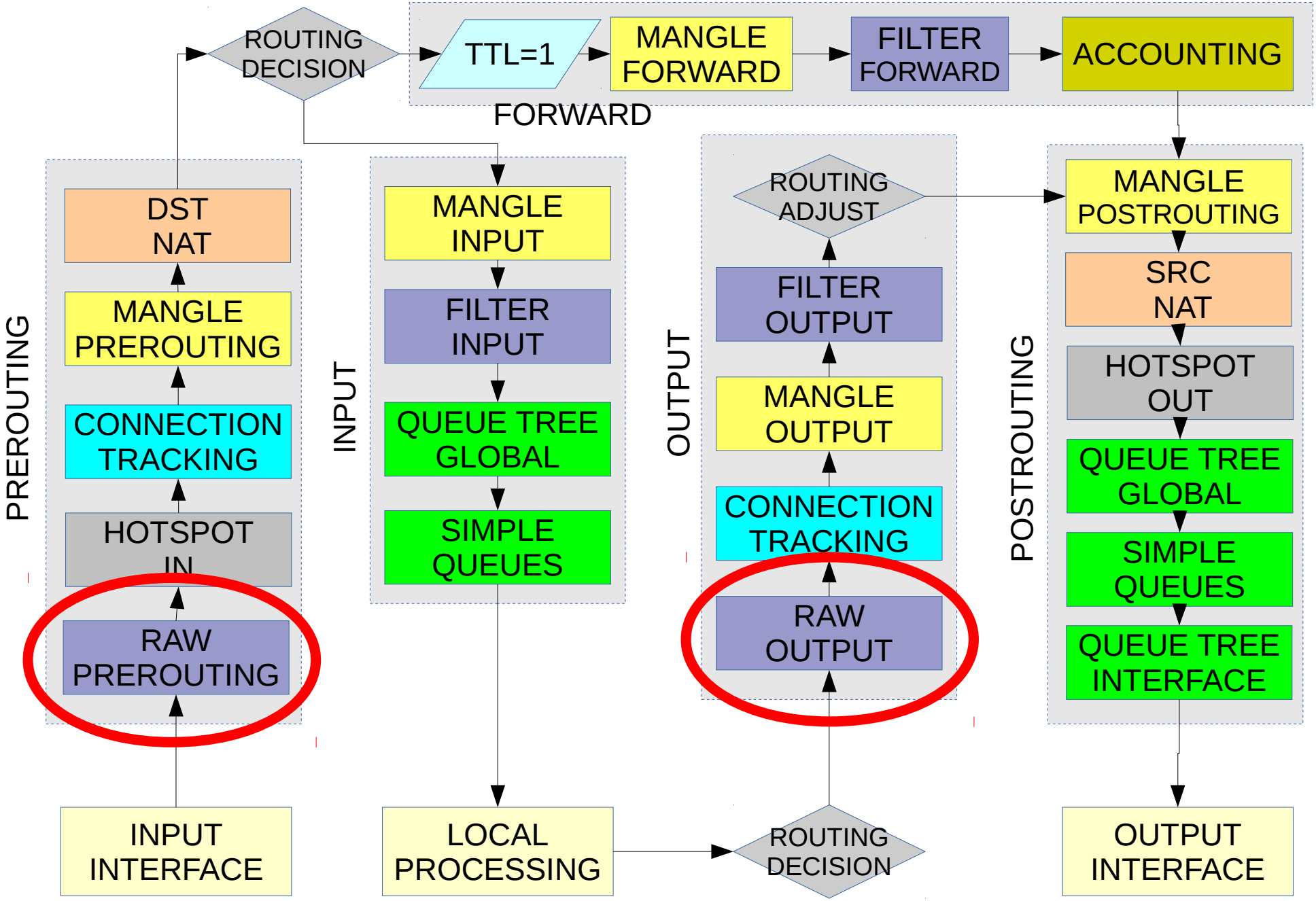
```
/ip firewall mangle
/ipv6 firewall mangle
```


ROUTING DECISION
TTL=1
MANGLE FORWARD
FILTER FORWARD
ACCOUNTING
FORWARD
PREROUTING
DST NAT
MANGLE PREROUTING
CONNECTION TRACKING
HOTSPOT IN
RAW PREROUTING
INPUT
MANGLE INPUT
FILTER INPUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
OUTPUT
ROUTING ADJUST
FILTER OUTPUT
MANGLE OUTPUT
CONNECTION TRACKING
RAW OUTPUT
POSTROUTING
MANGLE POSTROUTING
SRC NAT
HOTSPOT OUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
QUEUE TREE INTERFACE
INPUT INTERFACE
LOCAL PROCESSING
ROUTING DECISION
OUTPUT INTERFACE

# New from 6.36 raw table

- only two chains
- INPUT
- OUTPUT

```
/ip firewall raw
/ipv6 firewall raw
```


ROUTING DECISION
TTL=1
MANGLE FORWARD
FILTER FORWARD
ACCOUNTING
FORWARD
PREROUTING
DST NAT
MANGLE PREROUTING
CONNECTION TRACKING
HOTSPOT IN
RAW PREROUTING
INPUT
MANGLE INPUT
FILTER INPUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
OUTPUT
ROUTING ADJUST
FILTER OUTPUT
MANGLE OUTPUT
CONNECTION TRACKING
RAW OUTPUT
POSTROUTING
MANGLE POSTROUTING
SRC NAT
HOTSPOT OUT
QUEUE TREE GLOBAL
SIMPLE QUEUES
QUEUE TREE INTERFACE
INPUT INTERFACE
LOCAL PROCESSING
ROUTING DECISION
OUTPUT INTERFACE

# How to do it better

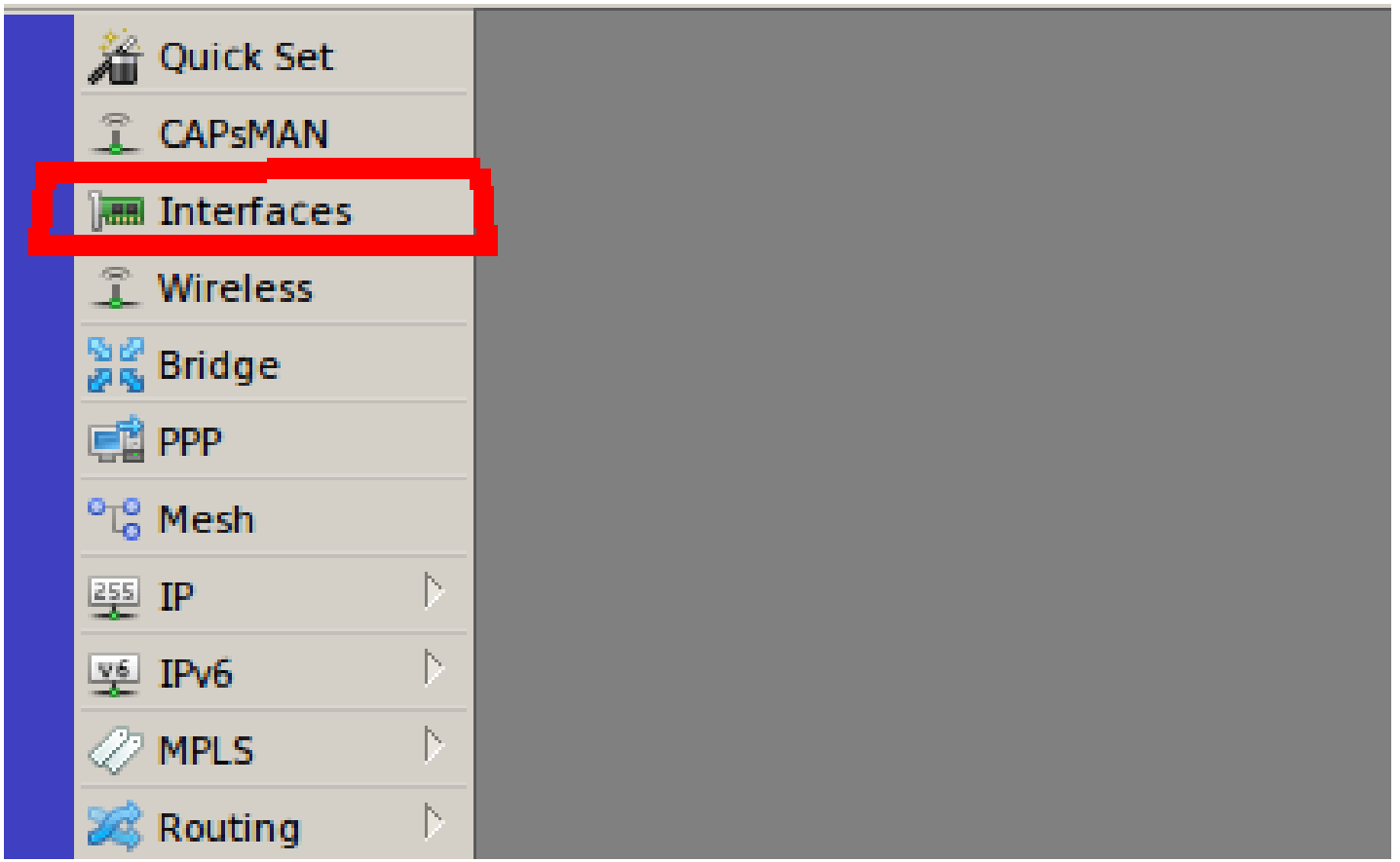
- use “interface list” and “address list”
- use “jump” and “return”
- define new chains
- define less rules as possible

later we see...

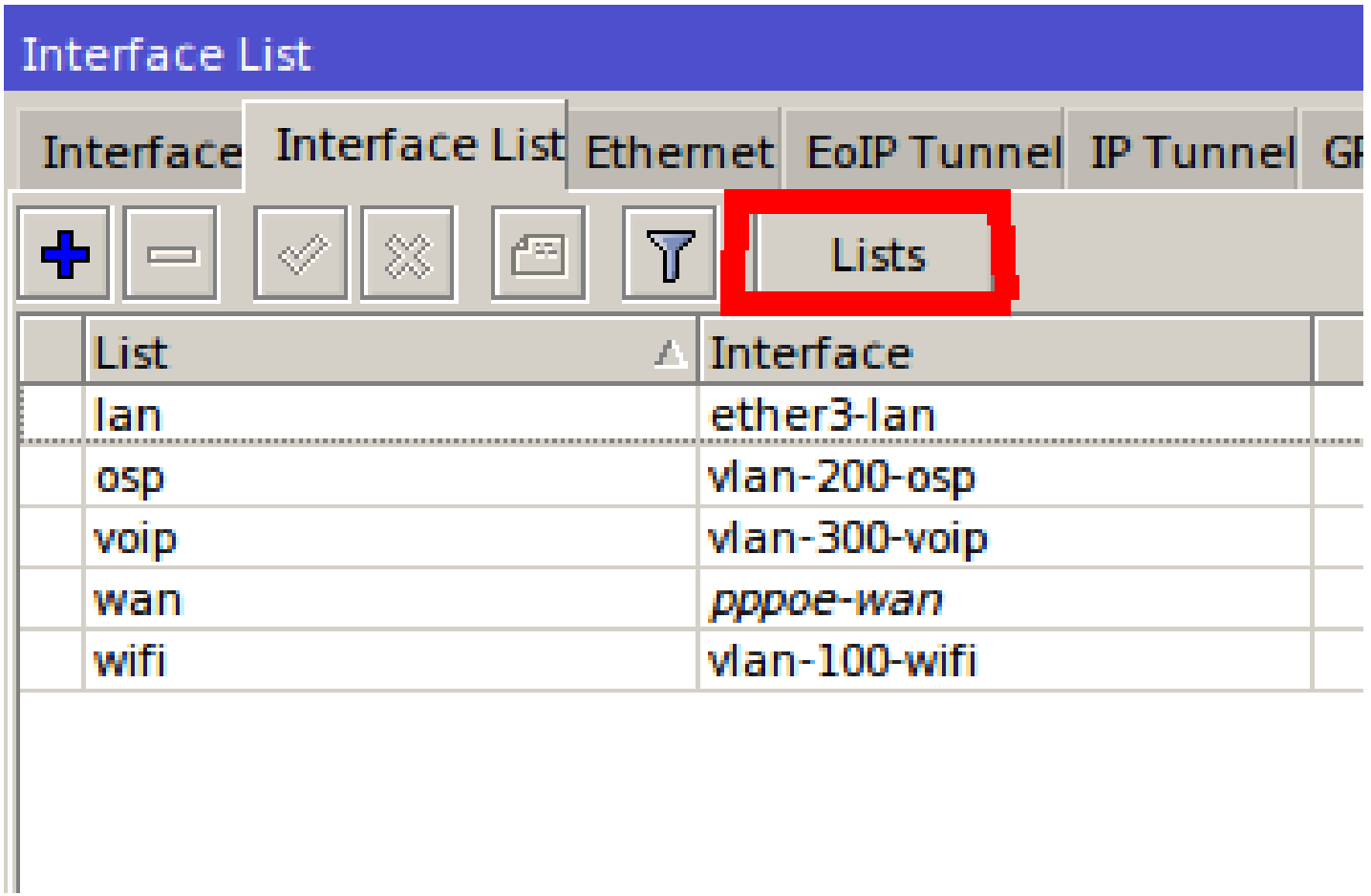
# New! “Interface Lists”

- Define a group of interfaces
- `/interface list`
- useful to simplify configuration

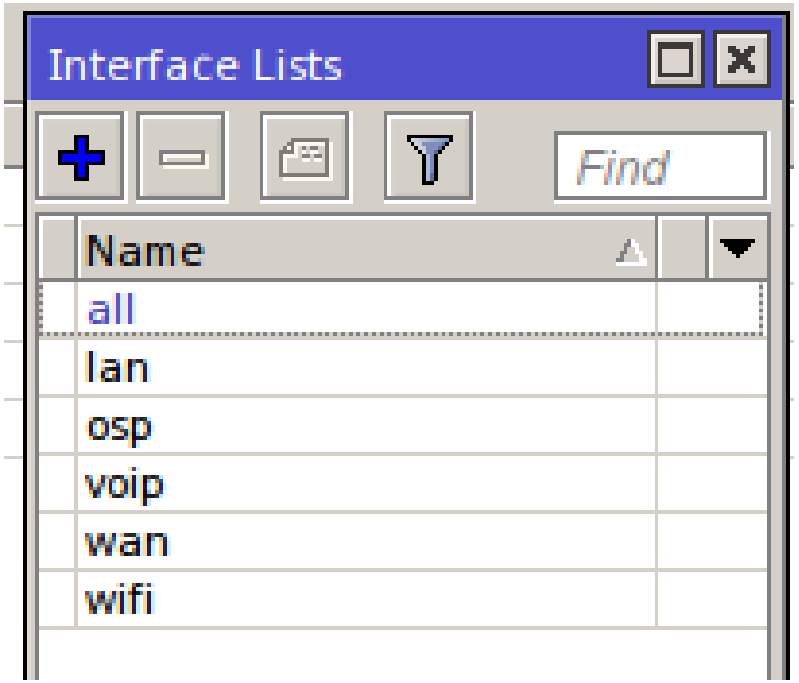
# Interface lists



# Interface lists



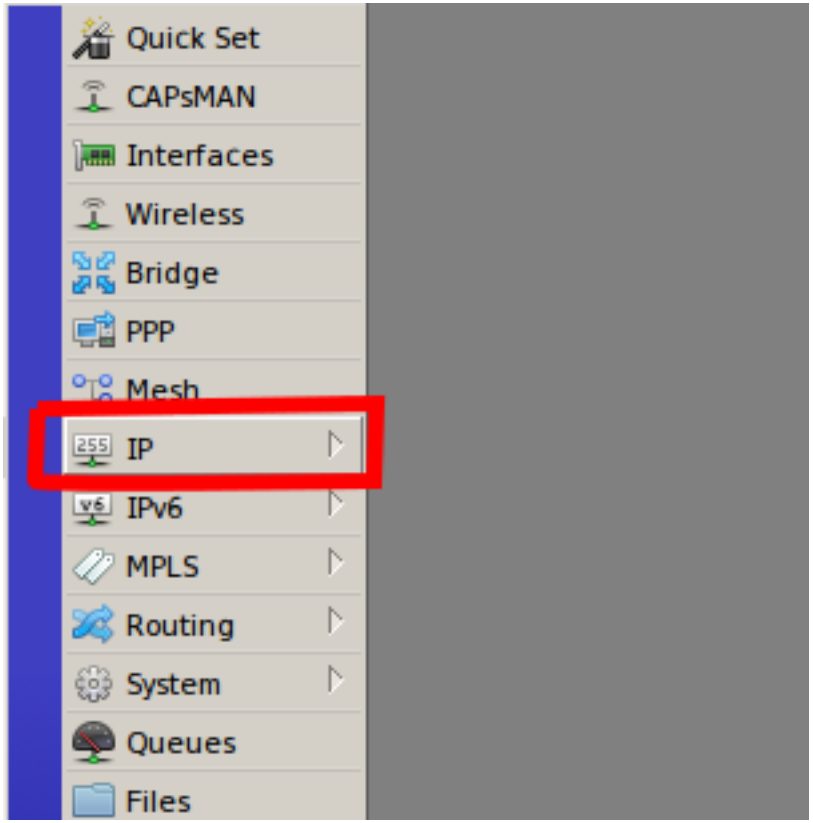
# Interface lists



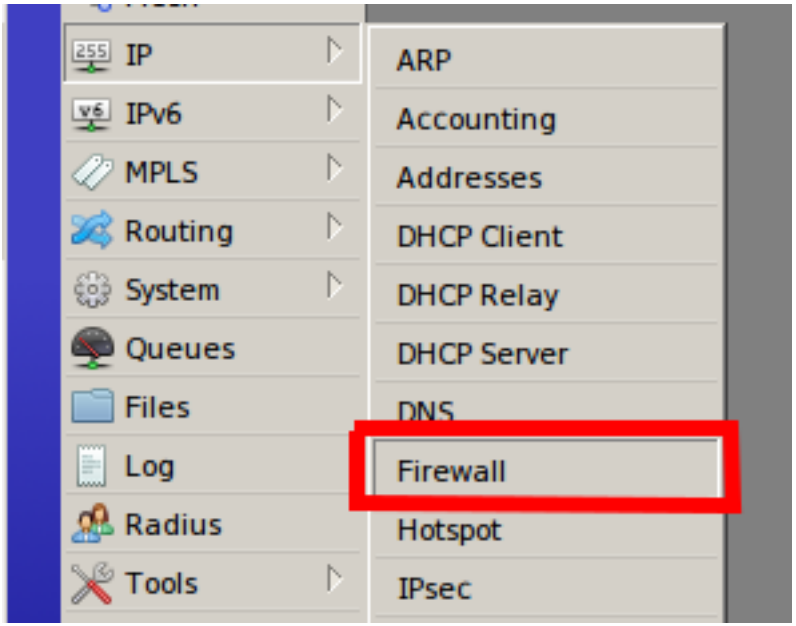
# Address Lists

- Define group of addresses
- I think MANDATORY for IPv6!!
- As “action” address can be added to address lists dynamically, also with time-out
- New from 6.36 dns names can be used in address lists!

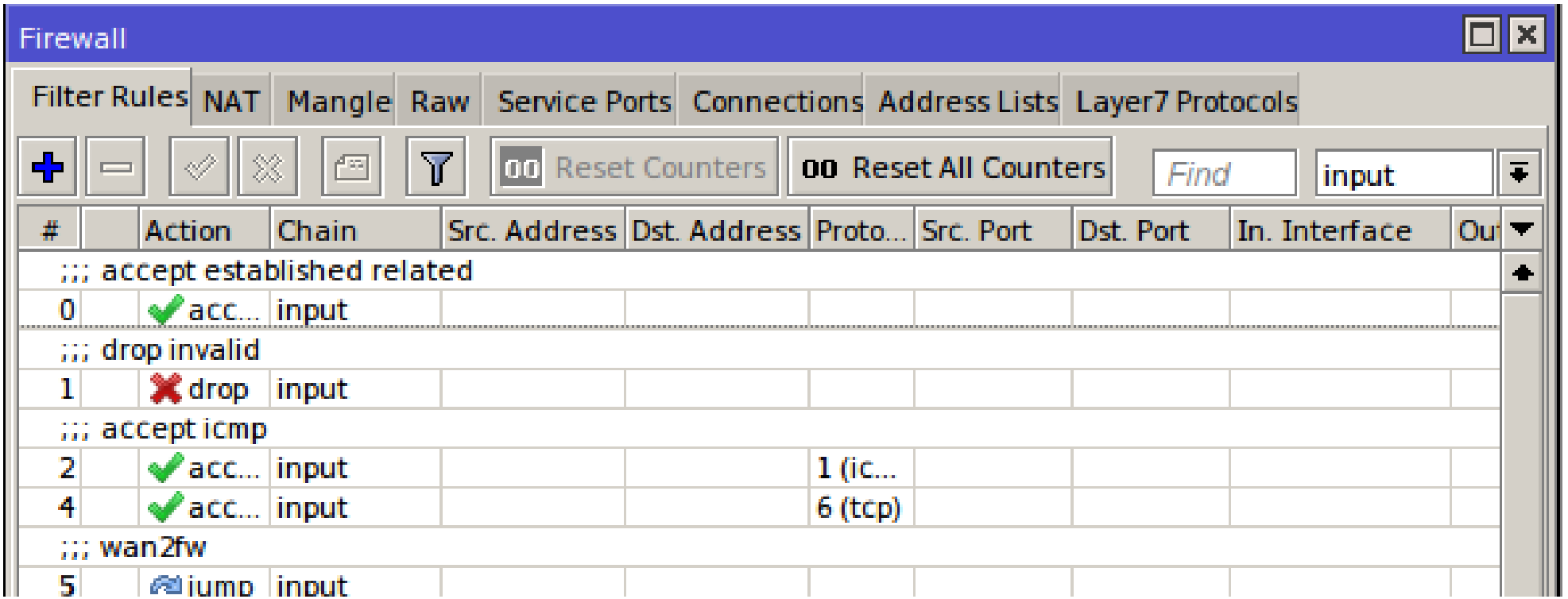
# Firewall IPv4



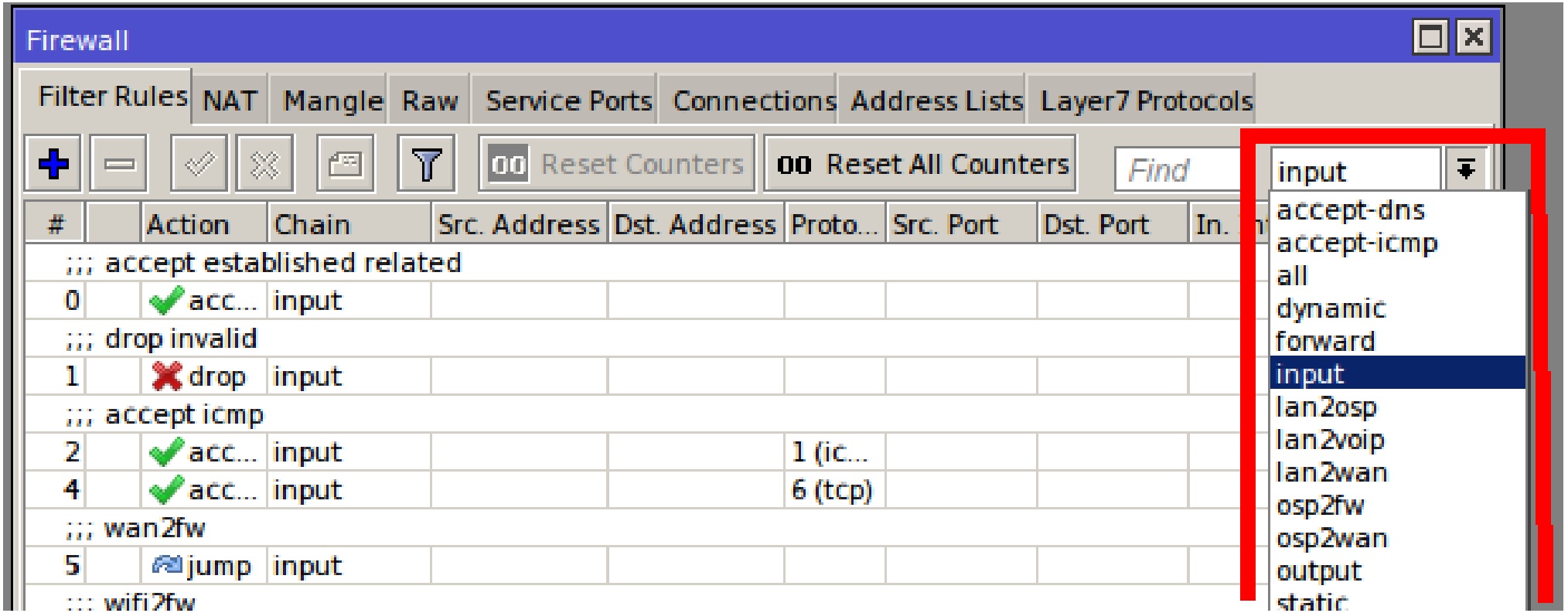
# Firewall IPv4



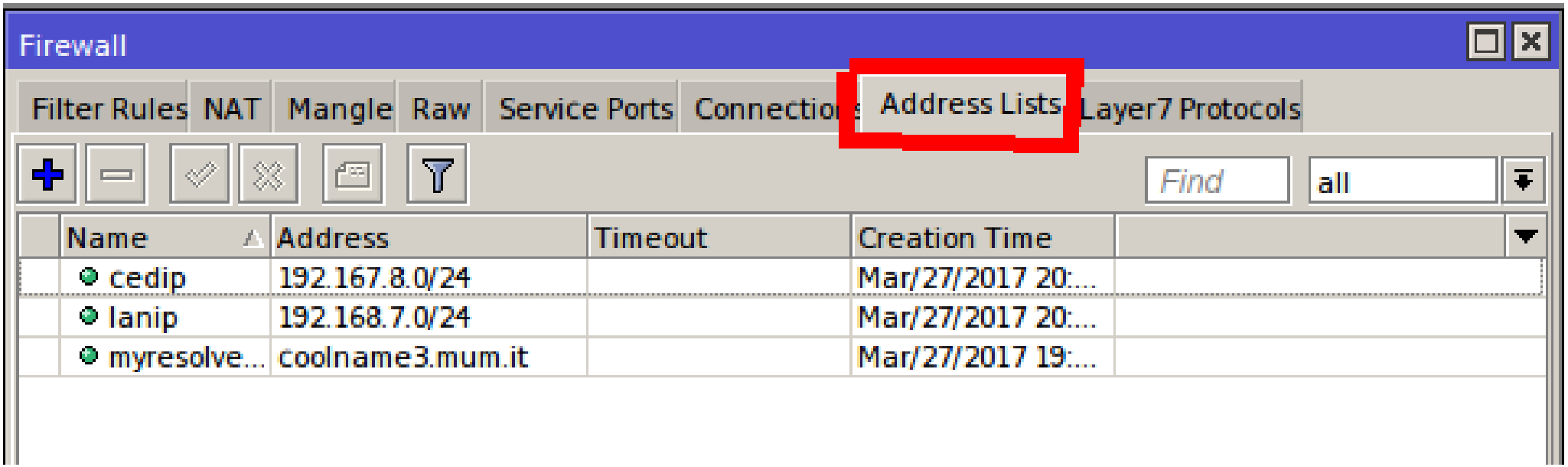
# Firewall IPv4



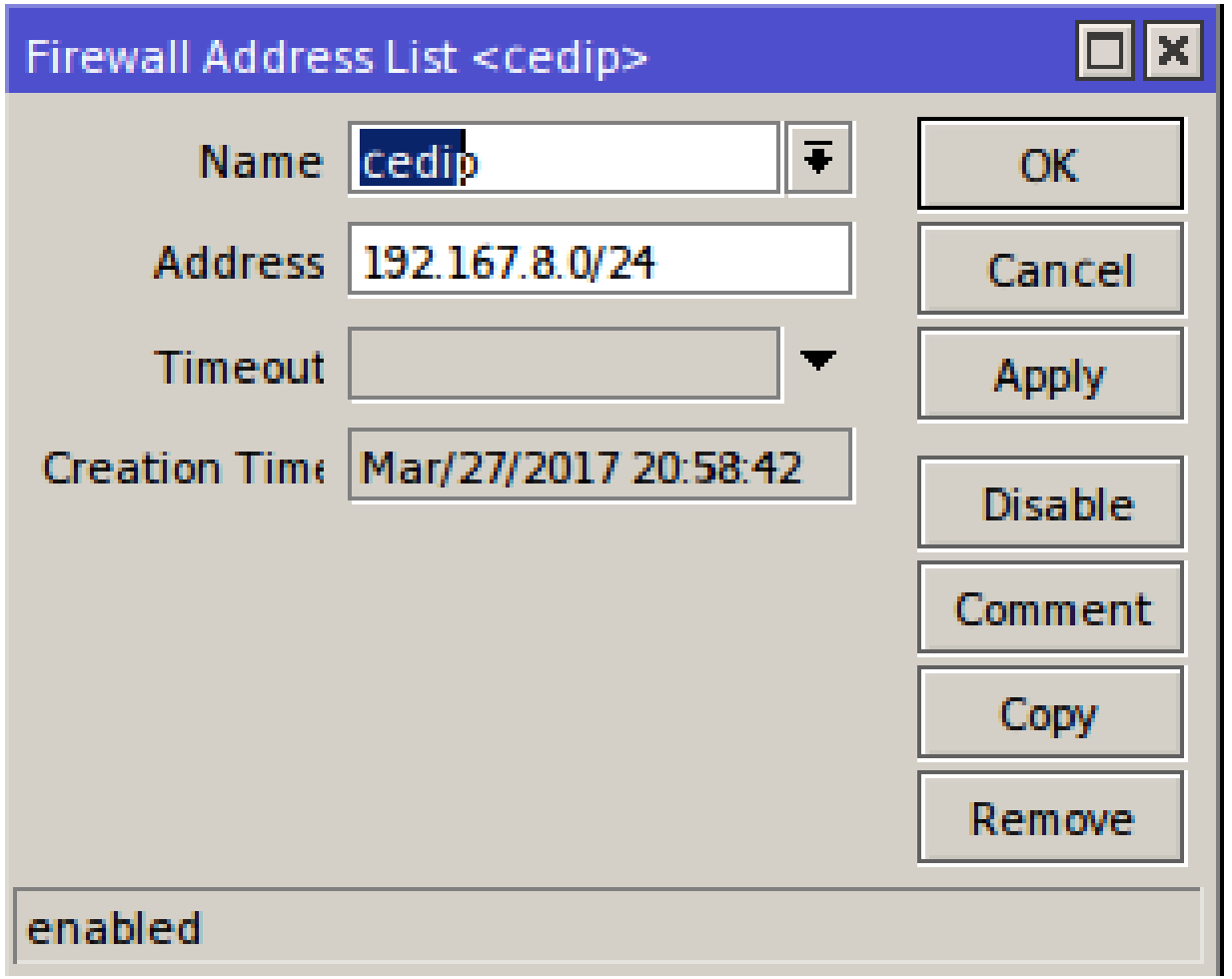
# Firewall IPv4



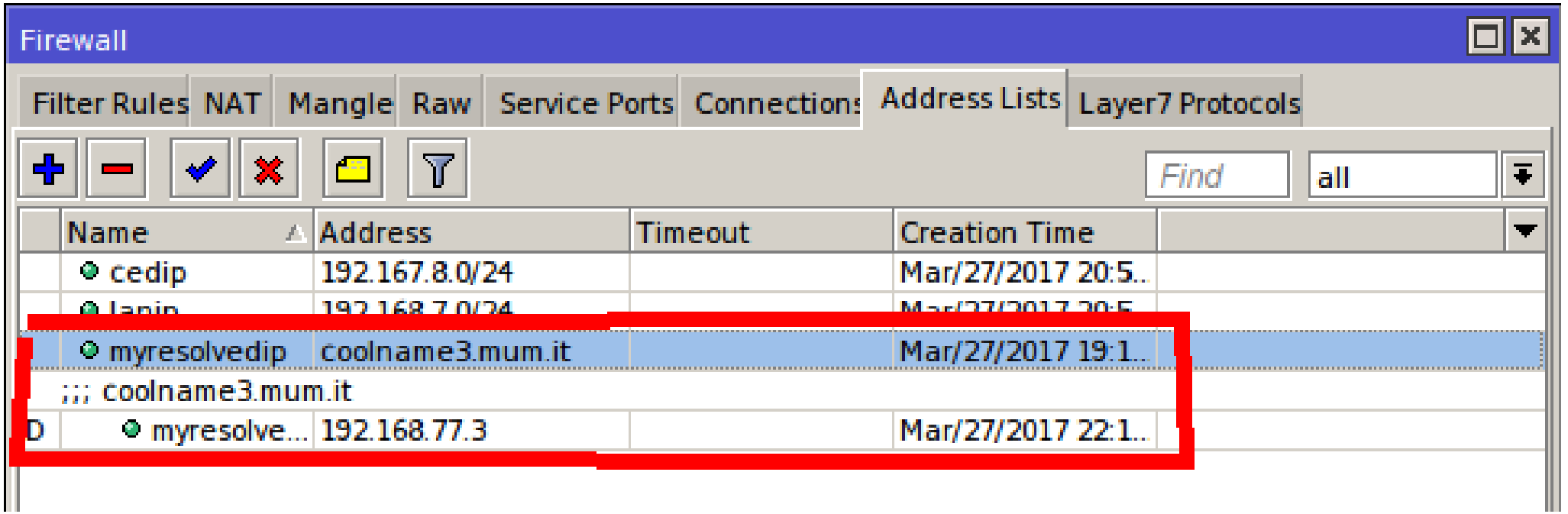
# New! “Address Lists”



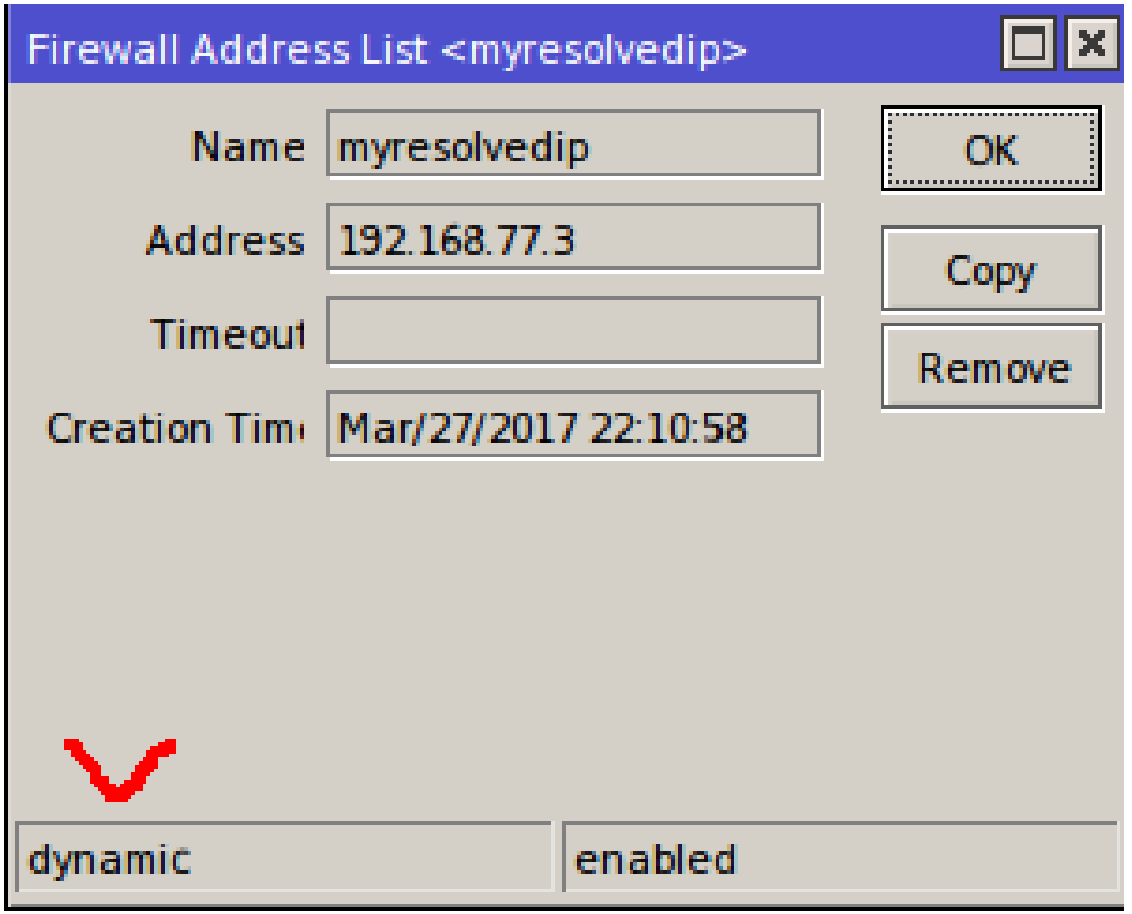
# New! “Address Lists”



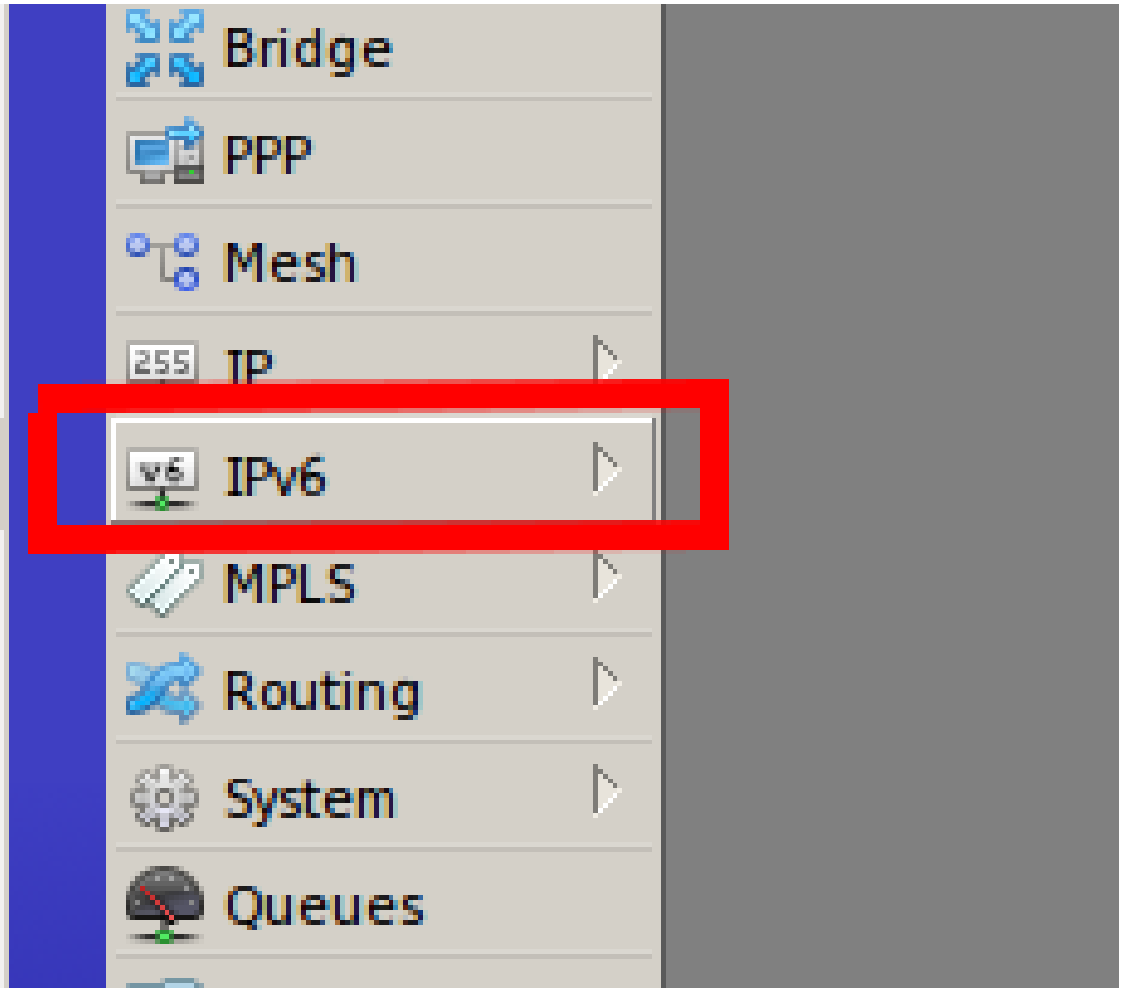
# New! “Address Lists”



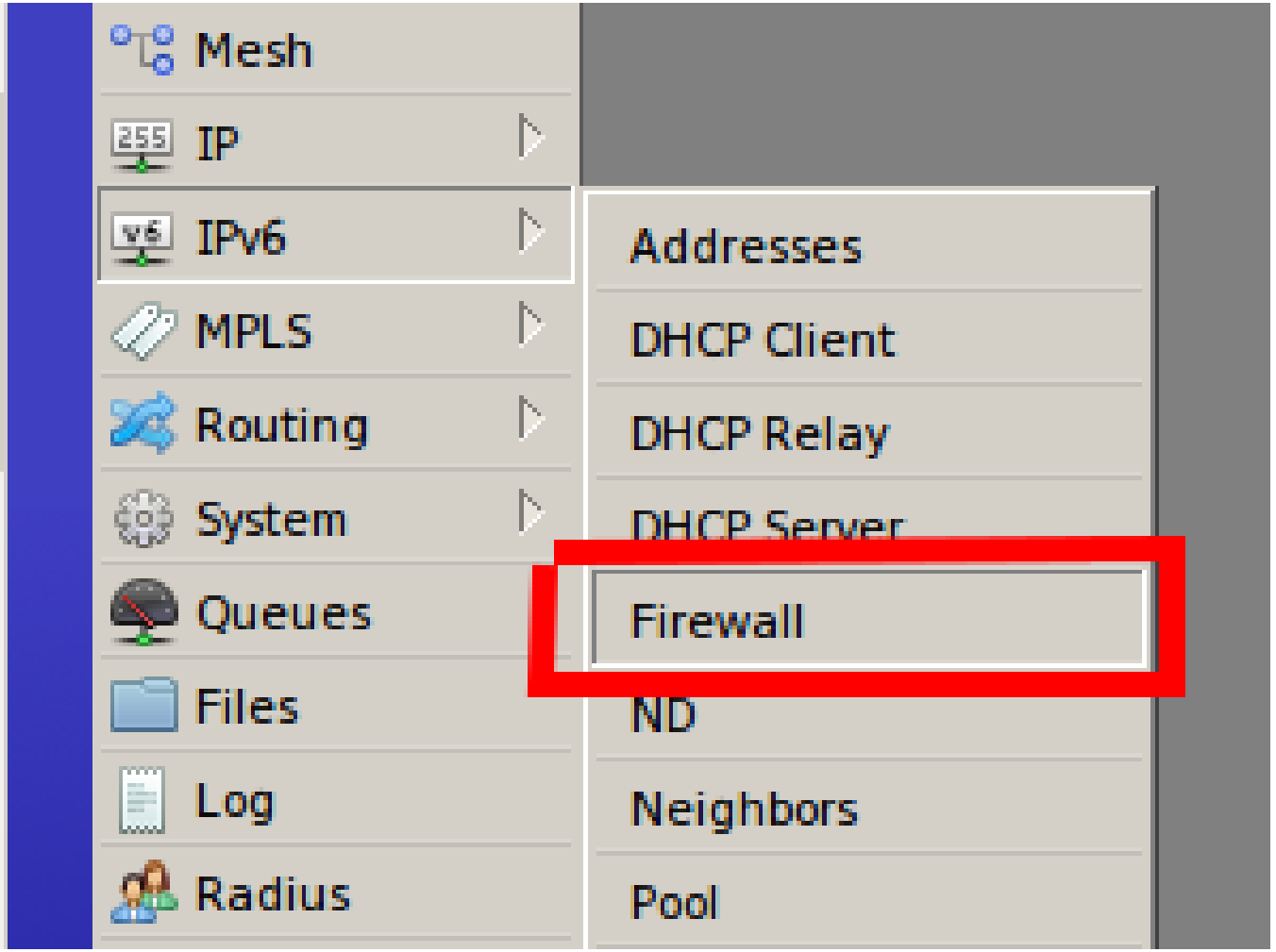
# New! “Address Lists”



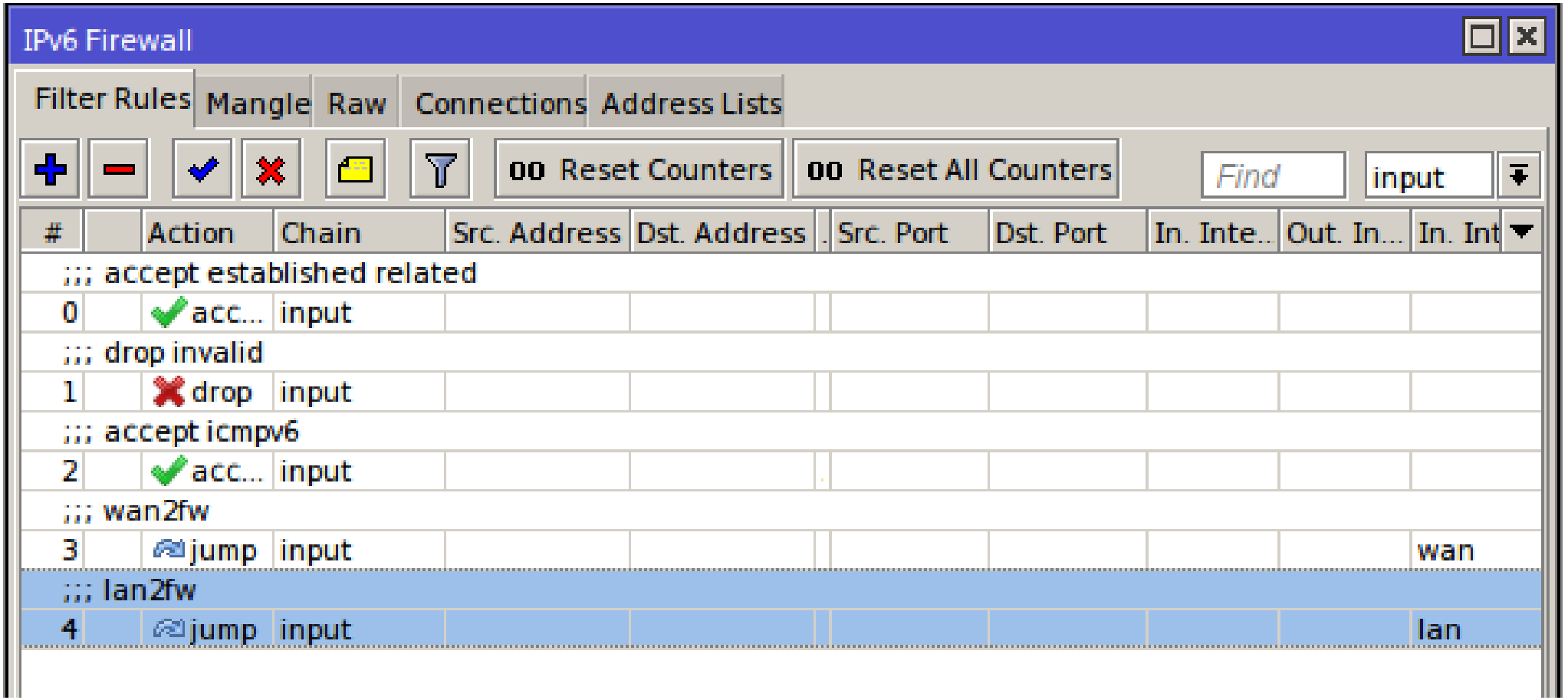
# Firewall IPv6



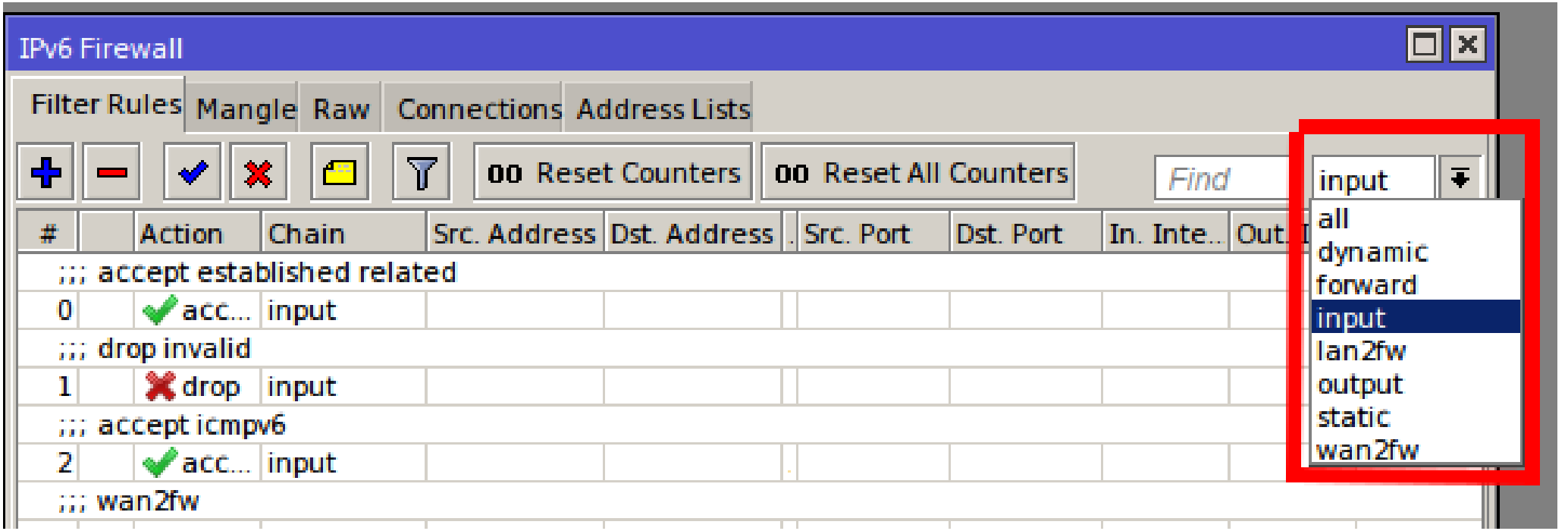
# Firewall IPv6



# Firewall IPv6



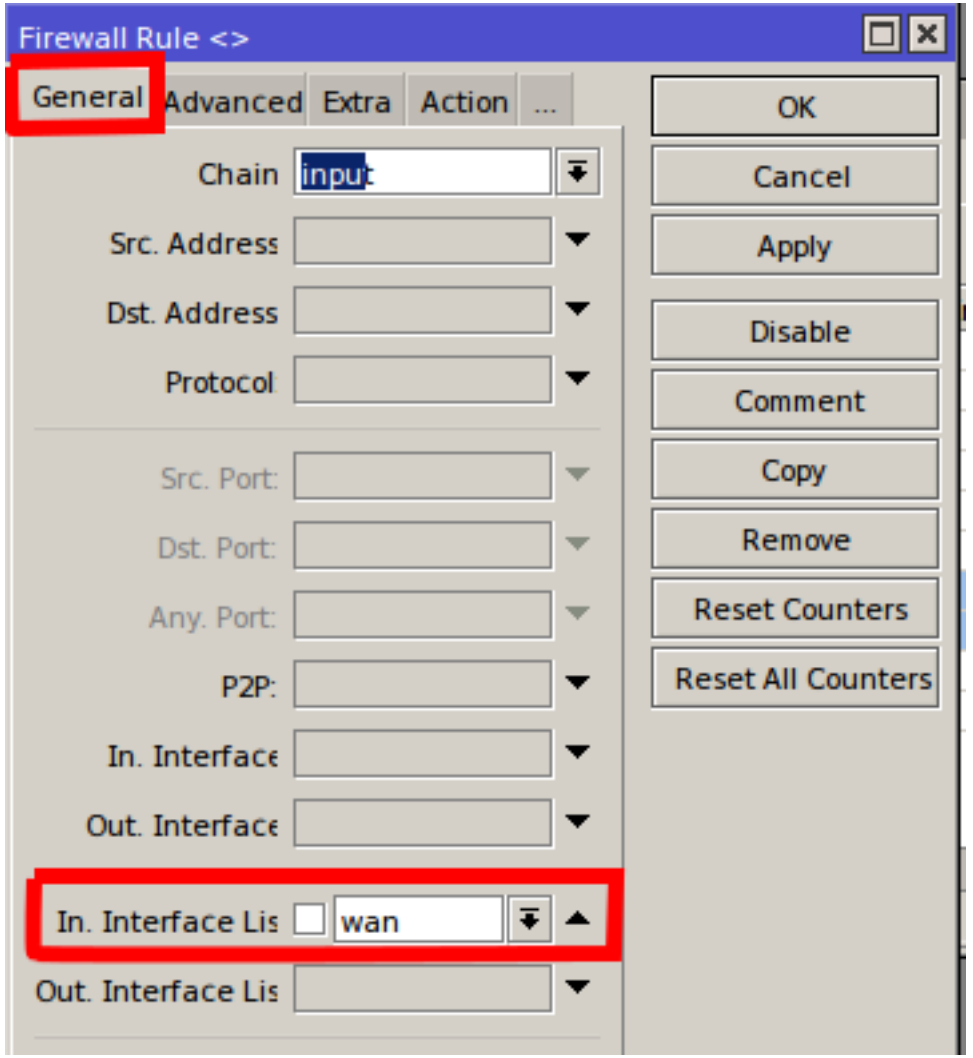
# Firewall IPv6



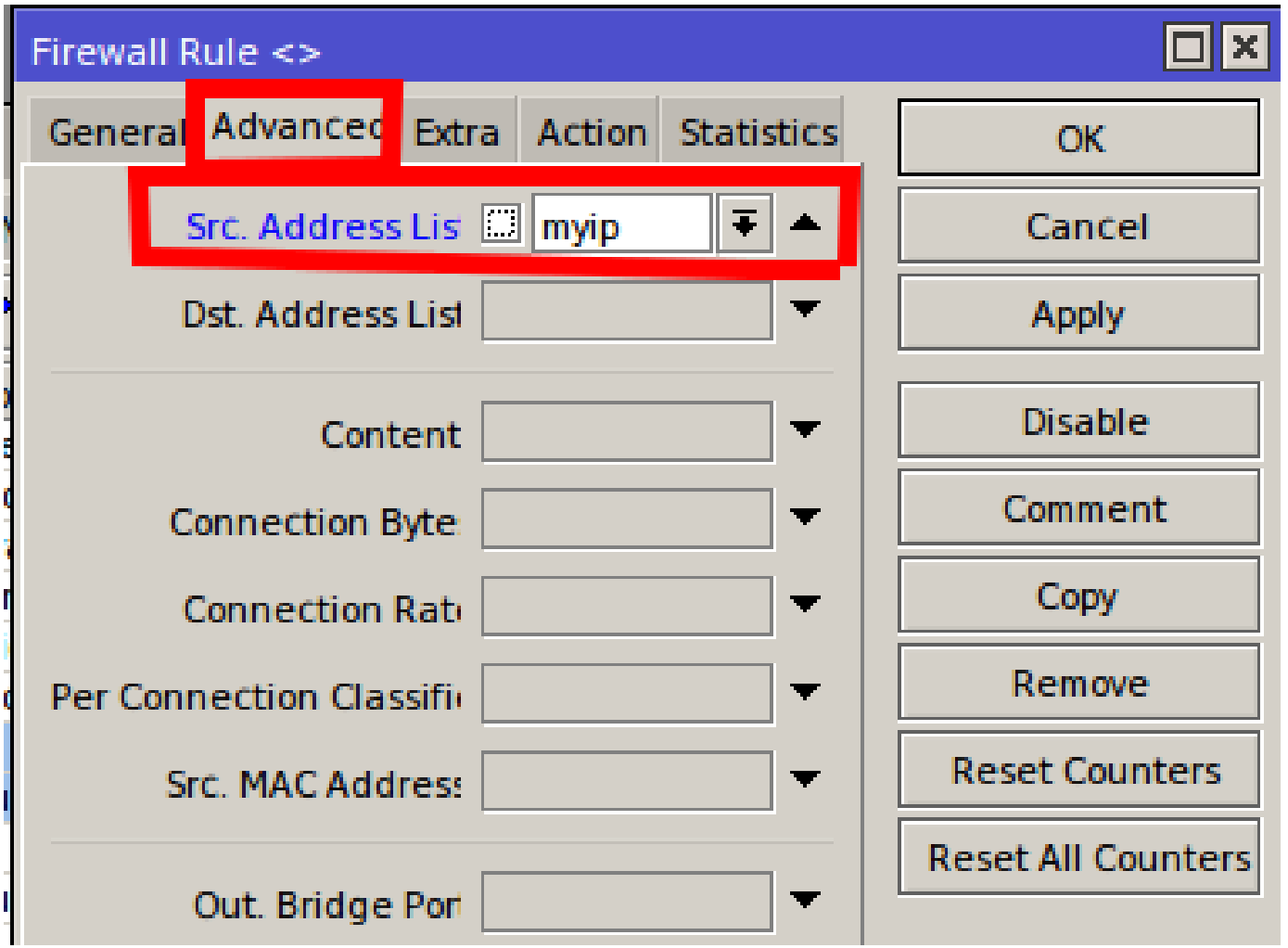
# Where we can use “lists”?

- Today only the “check”, not action

# Interface Lists



# Address Lists

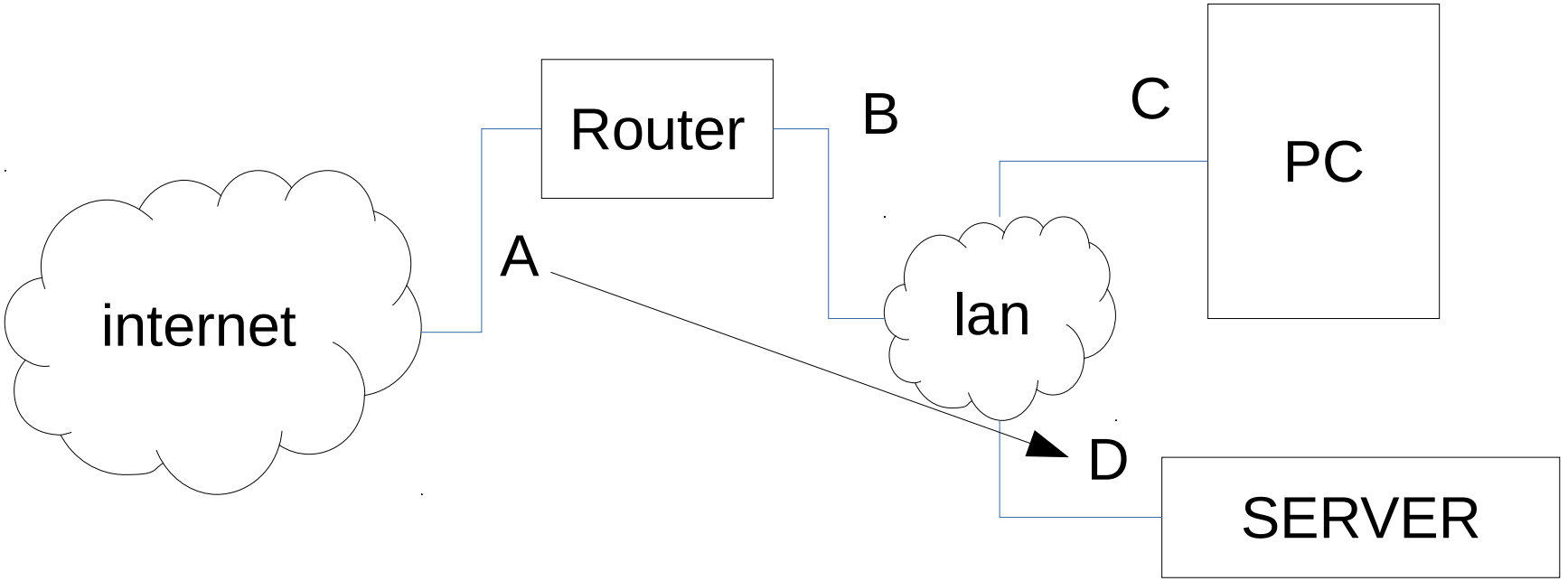


# And... improved firewall

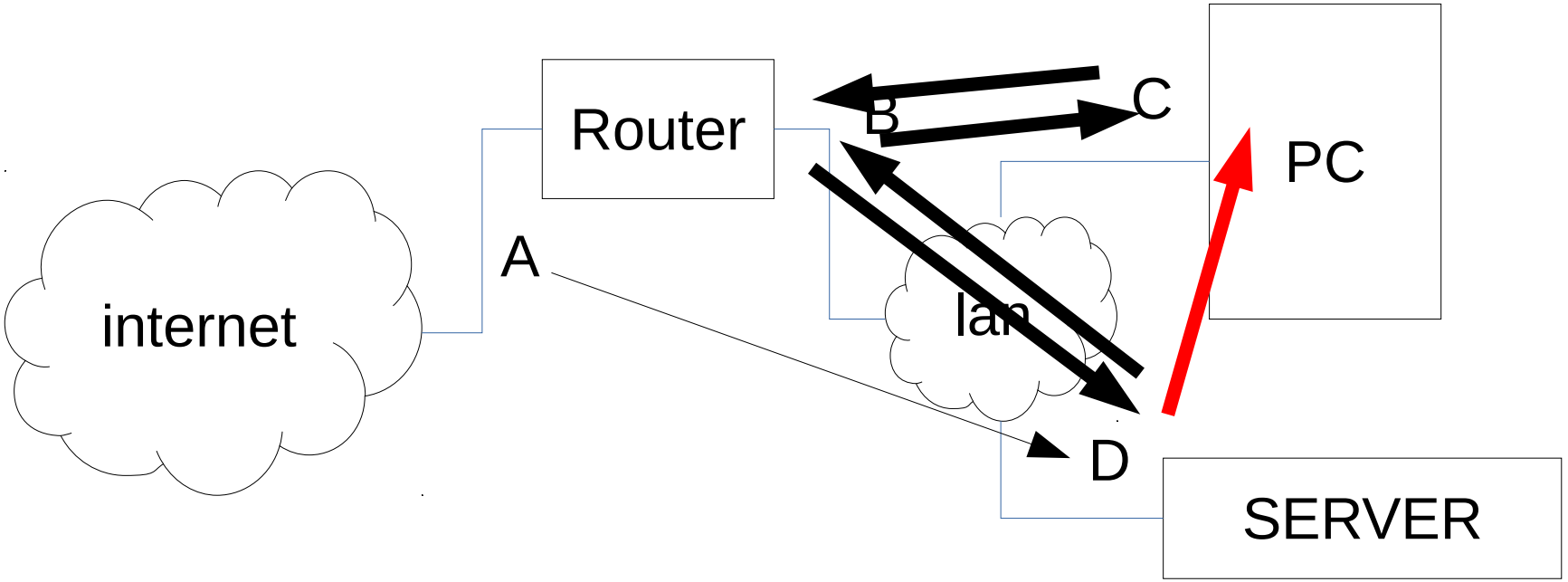
- faster “connection-limit”
- raw filter
- interface list
- address list with dns names
- limit (connections, packets, bits)

  check the wiki... all there..

# Example: routeback



# Example: routeback



# Goal

- PC with private address C need to talk to the server with private address D
- The server is on DNAT from the address A on the wan side of the router
- Use “dns name” of the server

# Routeback!

- First a dnat on the public ip address, and the packet is routed back to the lan
- Then i need a source nat, as the packet must route back to the router and then to the pc
- But... if the public ip address is dynamic?

# Address list!

- Configure the “cloud” option, so we have a dns address name with the public ip address
- Configure one address list with this dns name, then use the address list on the destination nat rule!

# Sample code part 1

```
/ip firewall address-list
add address=coolname3.mum.it list=myresolvedip
/ip firewall filter
add action=accept chain=input comment="accept
established related" connection-state=\
    established,related
add action=drop chain=input comment="drop invalid"
connection-state=invalid
add action=accept chain=input protocol=icmp
add action=drop chain=input comment="drop all from
wan" in-interface=pppoe-wan
```

# Sample code part 2

```
/ip firewall nat
add action=masquerade chain=srcnat comment="normal
masq" out-interface=pppoe-wan
add action=dst-nat chain=dstnat comment="nat to
192.168.7.2" dst-address-list=myresolvedip \
    to-addresses=192.168.7.2
add action=src-nat chain=srcnat comment="routeback
from 192.168.90.0/24 to lan (eq lan to lan)" \
    out-interface=ether3-lan src-
address=192.168.7.0/24 to-addresses=192.168.7.1
```

# A complex firewall

- One wan
- More than one lan
- Define and update frequently all rules
- Avoid to hard code all

# All code here... address list

```
/ip firewall address-list
add address=coolname3.mum.it list=myresolvedip
add address=192.168.7.0/24 list=lanip
add address=192.167.8.0/24 list=cedip
```

# All code here... input chain

```
/ip firewall filter
add action=accept chain=input comment="accept established related" \
connection-state=established,related
add action=drop chain=input comment="drop invalid" connection-state=invalid
add action=accept chain=input comment="accept icmp" protocol=icmp
add action=accept chain=input port=8291 protocol=tcp
add action=jump chain=input comment=wan2fw in-interface-list=wan jump-target=\
    wan2fw
add action=jump chain=input comment=wifi2fw in-interface-list=wifi jump-target=\
    wifi2fw
add action=jump chain=input comment=osp2fw in-interface-list=osp jump-target=\
    osp2fw
add action=jump chain=input comment=voip2fw in-interface-list=voip jump-target=\
    voip2fw
```

# All code here... forward chain 1

```
add action=accept chain=forward comment="accept established related" \
    connection-state=established,related
add action=drop chain=forward comment="drop invalid" \
    connection-state=invalid
add action=jump chain=forward comment="filtro icmp" \
    jump-target=accept-icmp protocol=icmp
add action=jump chain=forward comment="lan (ip) to wan" disabled=yes \
    in-interface-list=lan jump-target=lan out-interface-list=wan \
    src-address-list=lanip
add action=jump chain=forward comment="ced (ip) to wan" disabled=yes \
    in-interface-list=lan jump-target=lan out-interface-list=wan \
    src-address-list=cedip
```

# All code here... forward chain 2

```
add action=jump chain=forward in-interface-list=lan jump-target=lan2wan \
    out-interface-list=wan
add action=jump chain=forward in-interface-list=lan jump-target=lan2voip \
    out-interface-list=voip
add action=jump chain=forward in-interface-list=lan jump-target=lan2osp \
    out-interface-list=osp
add action=jump chain=forward in-interface-list=osp jump-target=osp2wan \
    out-interface-list=wan
add action=jump chain=forward in-interface-list=voip jump-target=voip2wan \
    out-interface-list=wan
add action=jump chain=forward in-interface-list=voip jump-target=voip2lan \
    out-interface-list=lan
add action=jump chain=forward in-interface-list=wan jump-target=wan2lan \
    out-interface-list=lan
```

# All code here... zone to zone

```
add action=drop chain=lan2osp comment="default drop"
add action=drop chain=lan2voip comment="default drop"
add action=drop chain=forward comment="default drop all2all"
add action=drop chain=input comment="drop all2fw" log-prefix=all2fw
add action=drop chain=voip2fw comment="default drop"
add action=drop chain=voip2lan comment="default drop"
add action=drop chain=voip2wan comment="default drop"
add action=drop chain=wan2lan comment="default drop"
add action=jump chain=wifi2fw comment="accept dns" jump-target=accept-dns
add action=drop chain=wifi2fw comment="default drop"
add action=jump chain=lan2wan jump-target=accept-dns
add action=drop chain=lan2wan comment="default drop"
add action=jump chain=wan2fw comment="protect ssh" jump-target=ssh
add action=drop chain=wan2fw comment="drop all from wan"
```

# All code here...
# dns check

```
add action=accept chain=accept-dns dst-port=53
protocol=udp

add action=accept chain=accept-dns dst-port=53
protocol=tcp

add action=return chain=accept-dns
```

# All code here... icmp check

```
add action=accept chain=accept-icmp comment="echo reply" icmp-options=0:0 \
    protocol=icmp
add action=accept chain=accept-icmp comment="net unreachable" icmp-options=3:0 \
    protocol=icmp
add action=accept chain=accept-icmp comment="host unreachable" icmp-options=3:1 \
    protocol=icmp
add action=accept chain=accept-icmp comment=\
    "host unreachable fragmentation required" icmp-options=3:4 protocol=icmp
add action=accept chain=accept-icmp comment="allow source quench" icmp-options=\
    4:0 protocol=icmp
add action=accept chain=accept-icmp comment="allow echo request" icmp-options=\
    8:0 protocol=icmp
add action=accept chain=accept-icmp comment="allow time exceed" icmp-options=\
    11:0 protocol=icmp
add action=accept chain=accept-icmp icmp-options=12:0 protocol=icmp
add action=drop chain=accept-icmp comment="deny all other types"
```

# All code here... ssh protection

```
add action=drop chain=ssh comment="drop ssh brute forcers" dst-
port=22 protocol=tcp src-address-list=badip

add action=add-src-to-address-list address-list=badip address-list-
timeout=1w3d chain=ssh dst-port=22 protocol=tcp src-address-
list=ssh_stage3

add action=add-src-to-address-list address-list=ssh_stage3 address-
list-timeout=1m chain=ssh dst-port=22 protocol=tcp src-address-
list=ssh_stage2

add action=add-src-to-address-list address-list=ssh_stage2 address-
list-timeout=1m chain=ssh dst-port=22 protocol=tcp src-address-
list=ssh_stage1

add action=add-src-to-address-list address-list=ssh_stage1 address-
list-timeout=1m chain=ssh dst-port=22 protocol=tcp

add action=return chain=ssh
```

# All code here... icmp check

```
/ip firewall nat

add action=masquerade chain=srcnat out-
interface=pppoe-wan

/ip firewall raw

add action=drop chain=prerouting comment="drop
bad ip" in-interface-list=wan \

    src-address-list=badip
```

# What you've seen

- Compex firewall
- And configuration can be exported and imported to another routerboard, with NO ERROR
- And all “specific” configuration is on the “interface lists” and “address lists”
- Recycle firewall rules

# This year request

- Complete IPv6 firewall
- Please add some kind of “global” generic constant values like objects
- ip addresses
- ports

# Questions?

# Thank you!

massimo@dicobit.it