RouterOs L2 filtering

Massimo Nuvoli

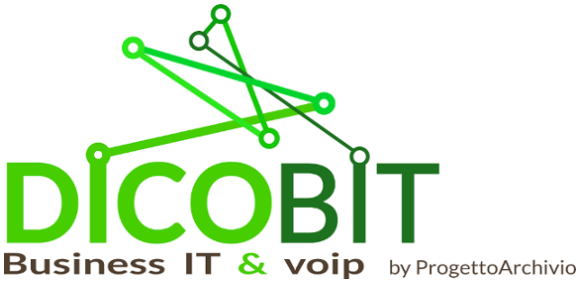
TRAINER #TR0368

MUM Europe 2018 Berlin

Massimo Nuvoli (maxnuv)

Owner of Progetto Archivio SRL and DICOBIT

System Engineer
System Architect

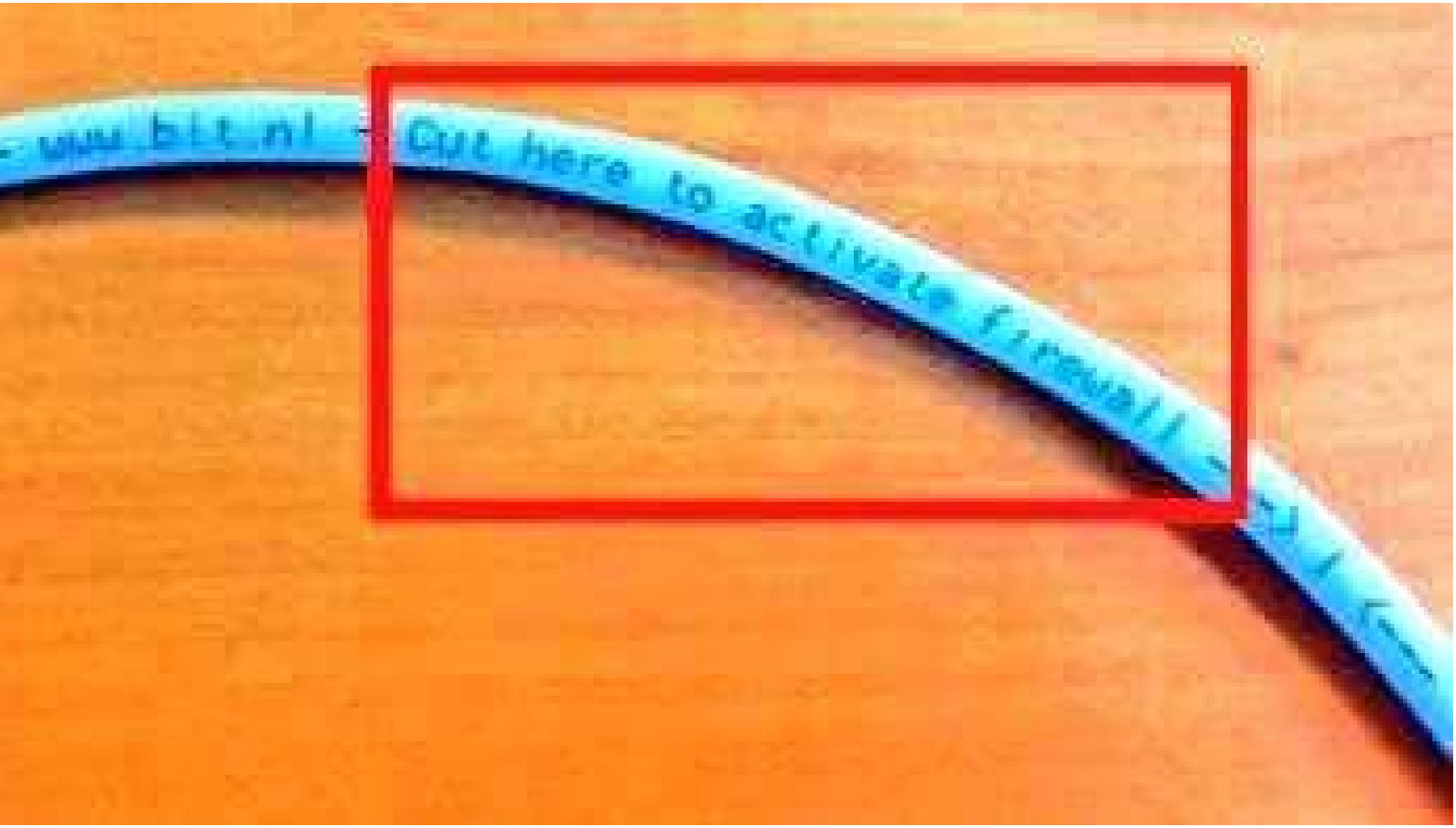


Today goals

- Know about L2 filtering in RouterOs
- Know where is, and what to do with
- Changes in the last year
- A lot of examples!

Before start...

Some L1 filtering


www.bit.nl
Cut here to activate firewall

New L2 configuration

- From RouterOs 6.41 major changes
- No more “master” “slave” interface
- Everything is “bridge” but there is “hardware acceleration”, so nothing changed
- New “H” label on bridge port, mean “this port and the bridge are doing hardware acceleration”

New L2 configuration

```
/interface bridge port
add bridge=bridge hw=no interface=ether5
add bridge=bridge hw=no interface=ether4
add bridge=bridge hw=no interface=ether3
```

- We can enable/disable hardware acceleration “by port”

New L2 configuration

- New vlan management at bridge level
- /interface bridge vlan
- STP (802.1D)
- RSTP (802.1w) (rapid)
- **MSTP (802.1s) (multiple)**

“Configuration driven”

- If you enable bridge with ports then you “may” have hardware acceleration
- If you use some feature that require hardware acceleration disabled then the “software bridge” is used instead
- Performance of the device depends on “configuration” and “active commands”

Hands on...

So… why L2 filtering?

- L2 filtering is done to achieve
  - Security
  - Performance
  - Both
  - L2 filtering can go “inside” the packet (ip address, ip protocol) but only if this is “normal” packet (no vpn, no ipsec, no mpls, no…), then in some case the packet need more steps on the chain to be processed!

Insecure MAC address!

- Sniff and change MAC address of interface is pretty easy on most operating systems
- Security based on MAC address is, what? There is no sense to put “security” and “MAC address” in the same statement
- “When” you are connected into a network (wired or not) security is broken
- “Secure” is 802.1x authentication with certificate, no support on RouterOs at wired level.

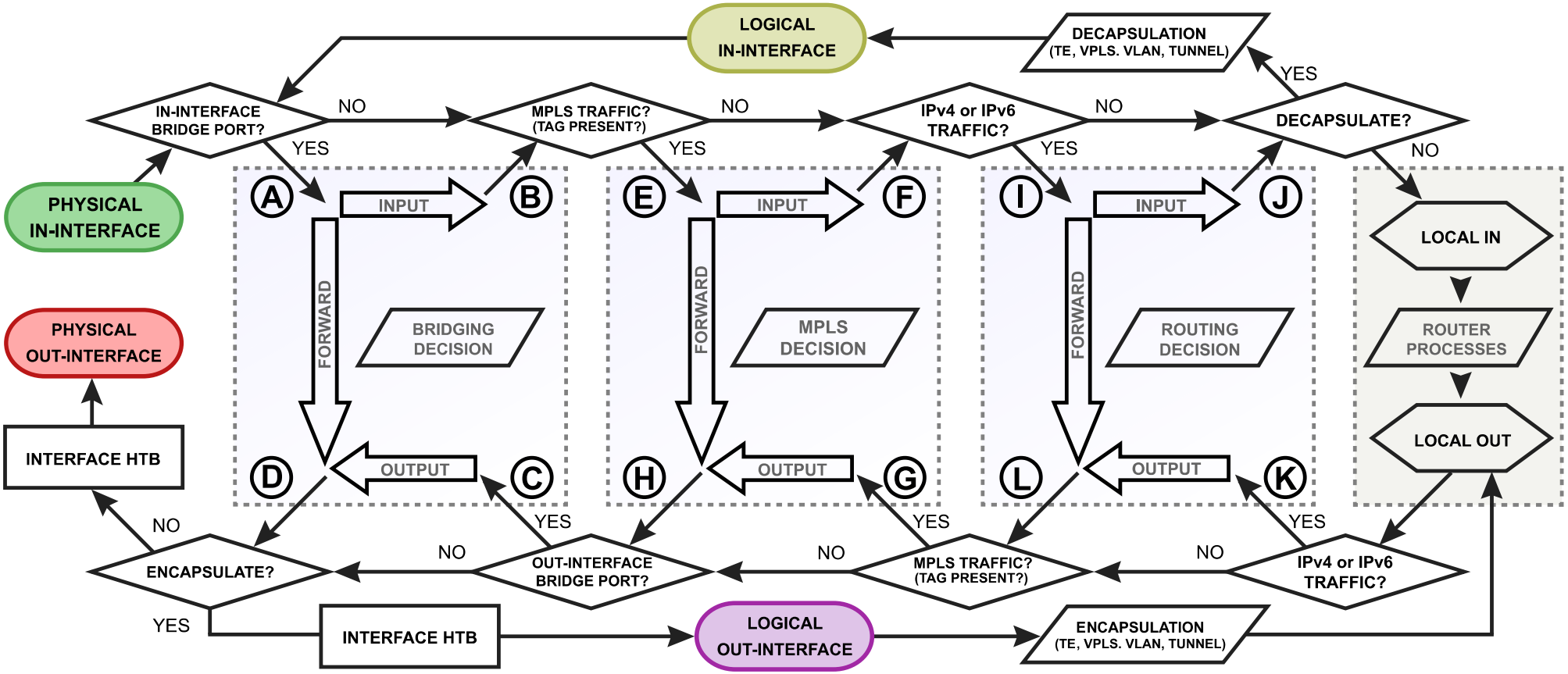
Filtering L2 protocols

- More useful, filter protocols!
- Eg, allow pppoe discovery and session and disable everything else
- Eg, check more inside packet, block dhcp..

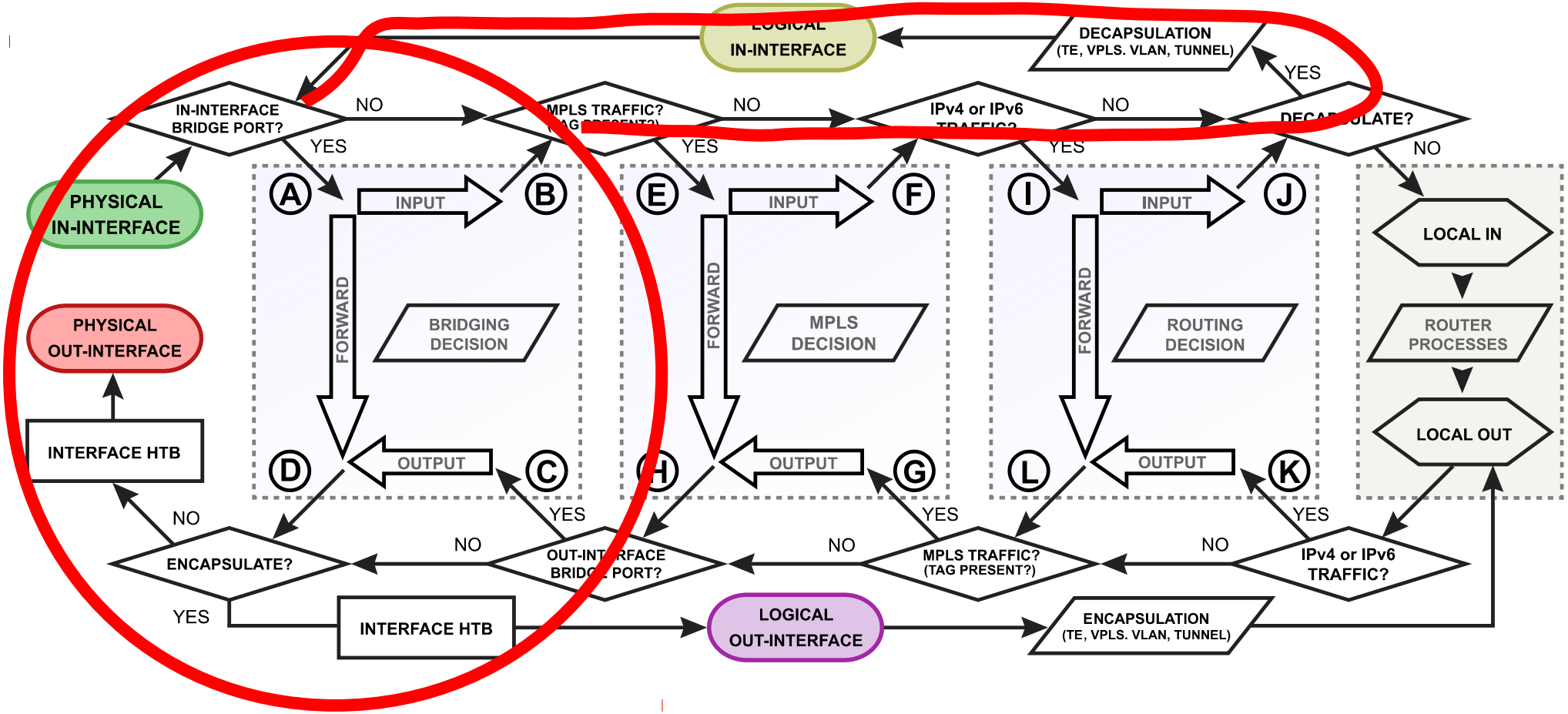
Filtering L2 headers

- Eg, filter packets on VLAN header
  - VLAN
  - Priority
  - Service VLAN
  - QinQ

RouterOs Packet Flow 1



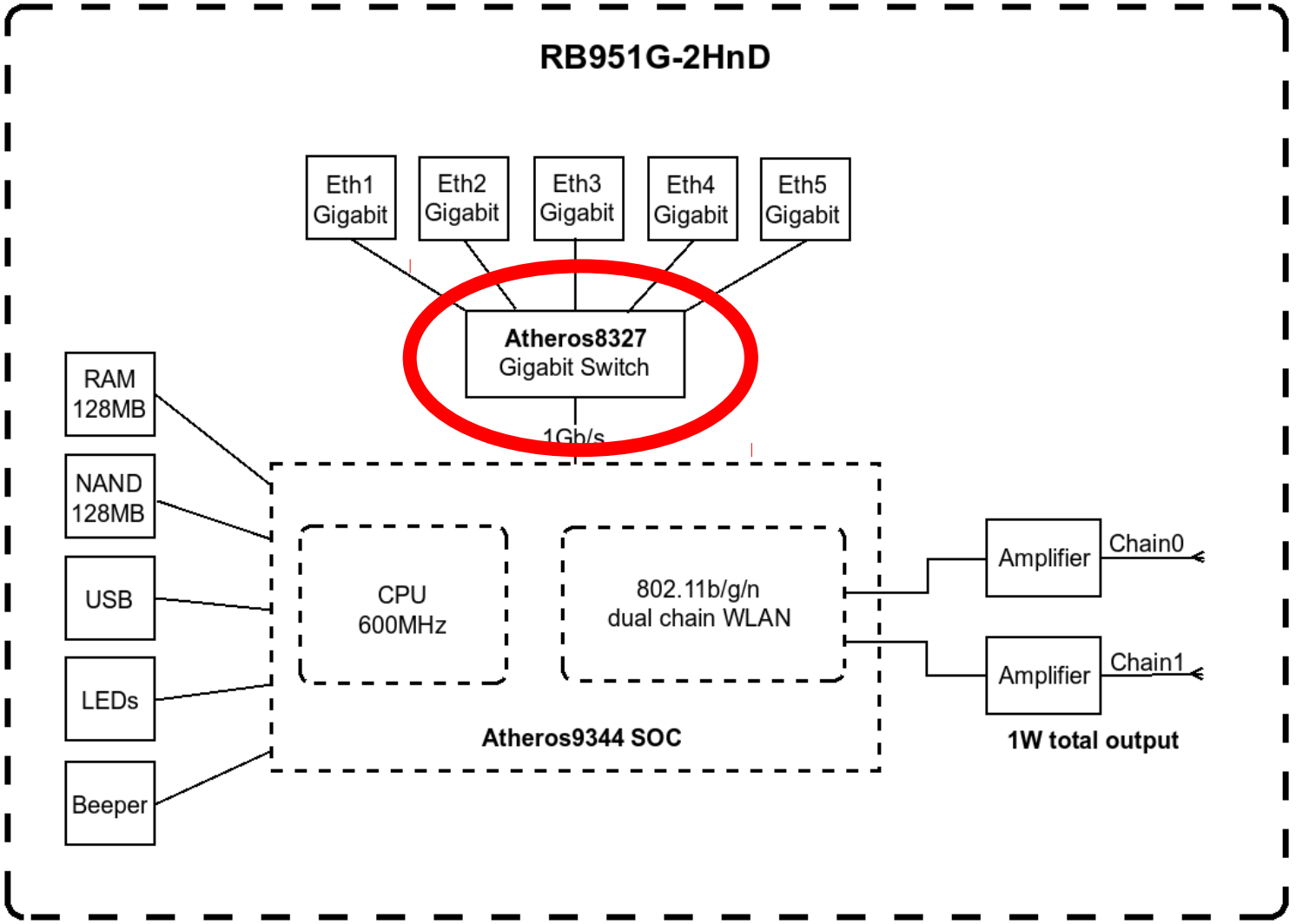
RouterOs Packet Flow 1



L2 Filtering “where”

- RouterOs can filter at L2 level in
  - Hardware switch chipset (low cpu cost)
  - Bridge level (medium cpu cost)
  - Firewall filter level (raw and normal) (high cpu cost)

Hardware Switch



Hardware Switch

- Basic Switch can do only some vlan check and some redirect
- Also basic settings can act on priority
- Advanced Switch (CRS125 or similar) can do more nasty things
- There is a limit on number of rules, kind of checks etc. etc. due to hardware limits!
- Check on wiki for limits!

Hands on...

And... SwOs

- Some RouterOs device can run SwOs
- No routing (switch only)
- Simpler setup
- Also L2 filtering but only hardware supported
- Same limits on number of rules, groups, etc etc.
- Check wiki for details!

Hardware Switch

Please..

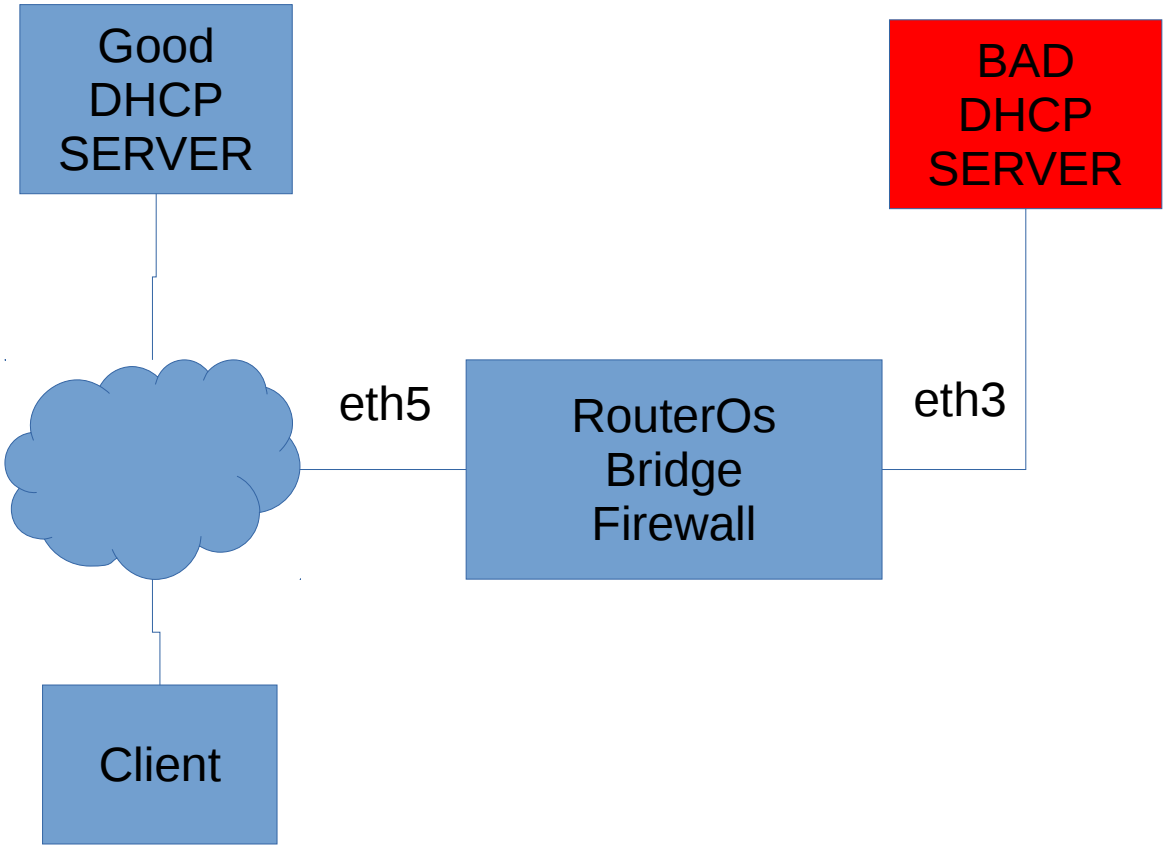
- Use only one filtering so
  - Setup vlan at switch hardware level

    or
  - Setup vlan at bridge level

Bridge filtering sample 1

```
/interface bridge filter

add action=drop chain=forward dst-
mac-
address=FF:FF:FF:FF:FF:FF/FF:FF:FF:F
F:FF:FF dst-port=67 in-
interface=ether5 \

ip-protocol=udp log=yes mac-
protocol=ip out-interface=!ether5
```

Bridge filtering sample 1



No bad dhcp servers!

- One rule
- But

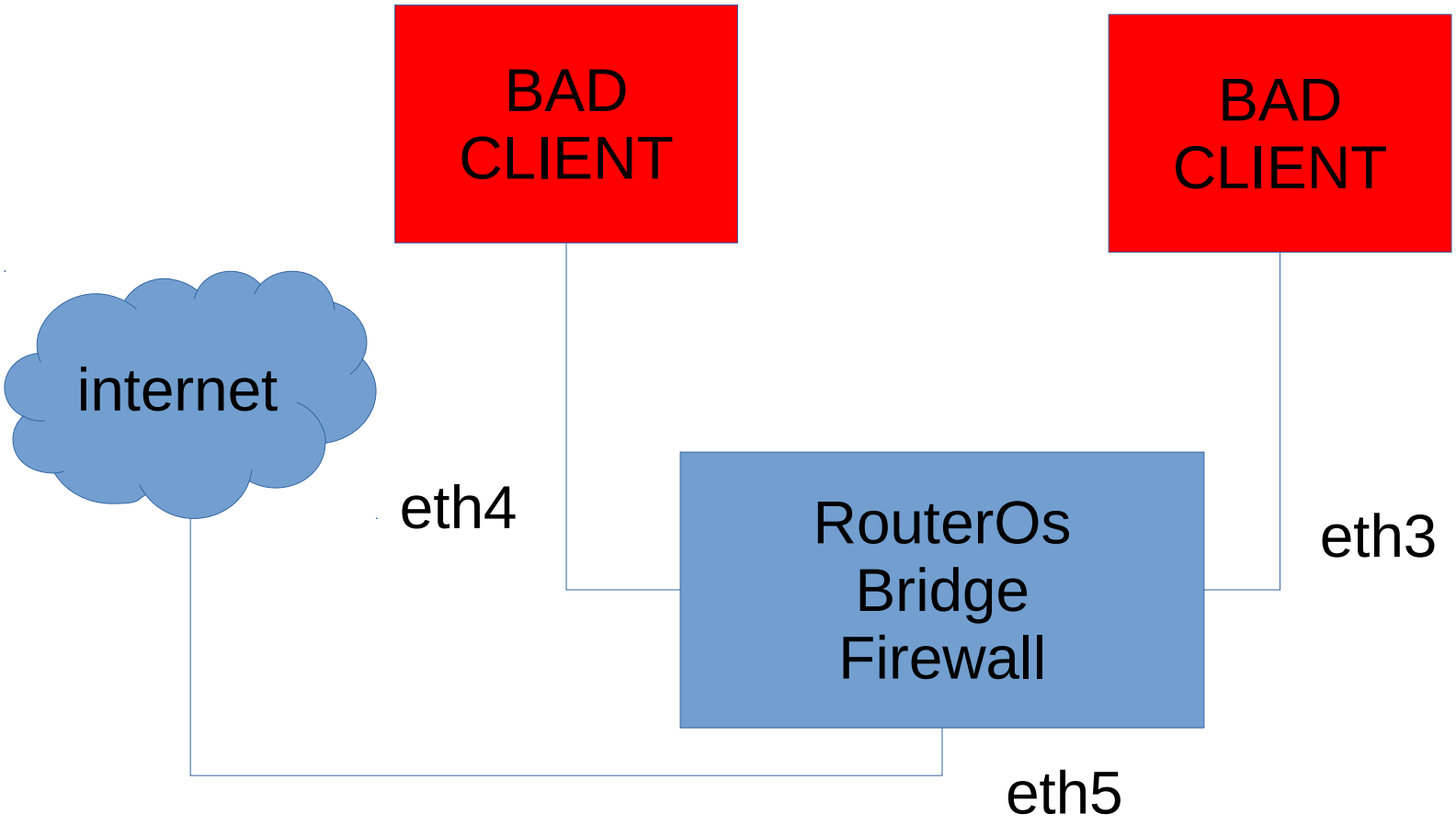
  On all ports must disable hardware offload

  !! CPU load !!

Bridge filtering sample 2

```
/interface bridge filter

add action=accept chain=input mac-
protocol=pppoe-discovery

add action=accept chain=input mac-
protocol=pppoe

add action=drop chain=input log=yes

add action=drop chain=forward in-
interface-list=all out-interface-list=all
```

Bridge filtering sample 2



No bad pppoe servers!

- Enable pppoe session and discover
- No need to bridge bad clients
- No other protocol enabled
- ether5 is not on the bridge!
- Same as before, no hardware offload!

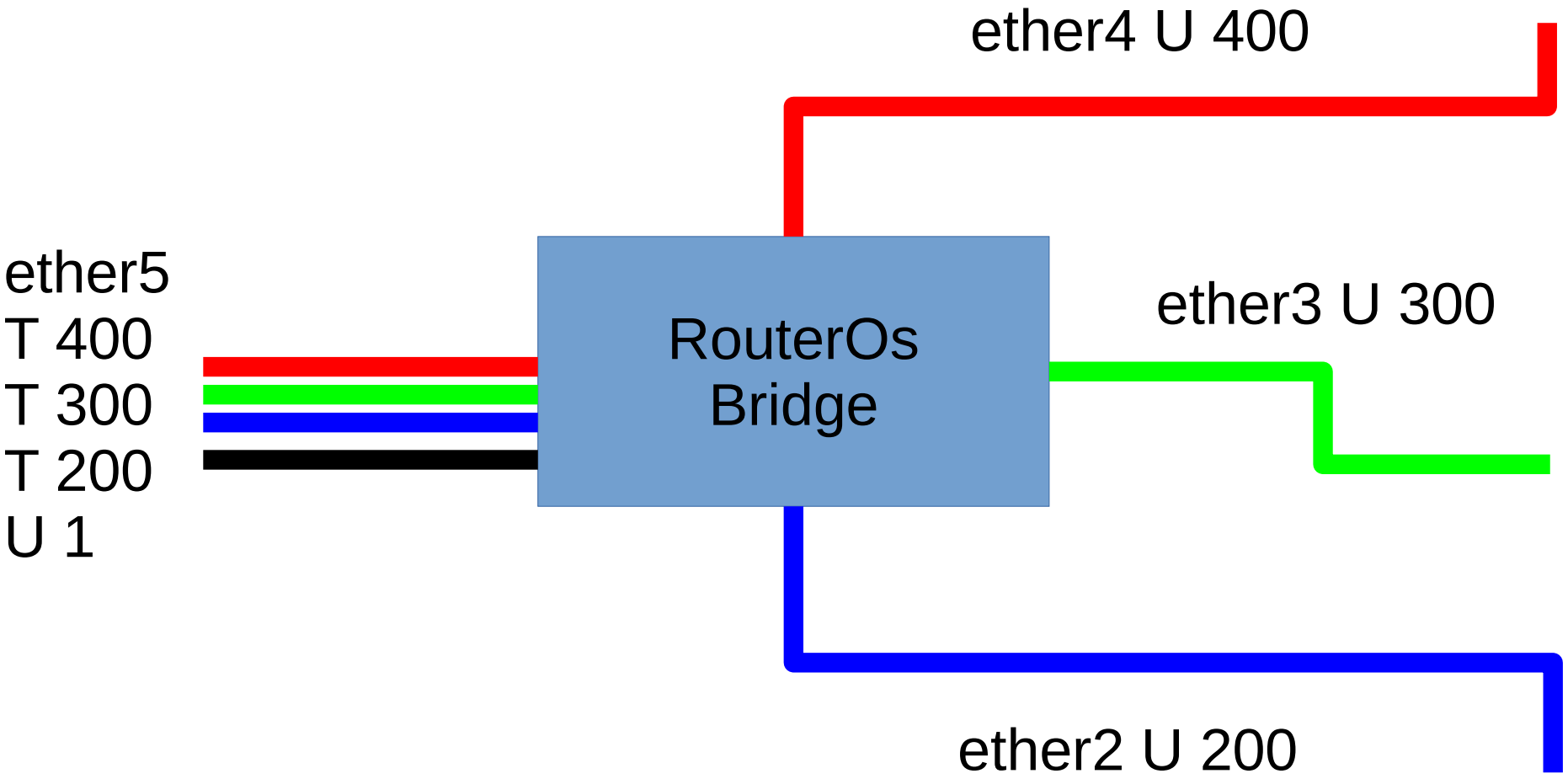
Bridge filtering sample 3

```
/interface bridge add name=bridge1
/interface bridge port
add bridge=bridge1 interface=ether5
add bridge=bridge1 interface=ether4 pvid=400
add bridge=bridge1 interface=ether3 pvid=300
add bridge=bridge1 interface=ether2 pvid=200
```

Bridge filtering sample 3

```
/interface bridge vlan

add bridge=bridge1 tagged=ether5
untagged=ether2 vlan-ids=200

add bridge=bridge1 tagged=ether5
untagged=ether3 vlan-ids=300

add bridge=bridge1 tagged=ether5
untagged=ether4 vlan-ids=400
```

Bridge filtering sample 3



Bridge filtering sample 3

Then

```
/interface bridge set vlan-filtering=yes bridge1
```

Now all vlan management is done by the bridge

On devices supported also hardware offloaded

On devices unsupported hardware offload is disabled (let me show how)

Bridge filtering sample 3

If you want to “access” vlan 200 packets on the bridge from RouterOs you MUST add the bridge itself on the filtering:

```
/interface bridge vlan

add bridge=bridge1
tagged=ether5,bridge1 untagged=ether2
vlan-ids=200
```

This mean “from the bridge you can see vlan 200”

Bridge filtering sample 4

- Assign and change priority
- Only WMM packets and VLAN tagged packets brings inside priority
- For all others priority is “0”

Bridge filtering sample 4

- First we define interface list
- Then we define filter rule based on the port of the bridge
- Then some mangle and queue trick to use assigned priority

Bridge filtering sample 4

```
/interface list
add name=phones
/interface list member
add interface=ether1 list=phones
add interface=ether3 list=phones
```

Bridge filtering sample 4

```
/interface bridge filter

add chain=forward
in-interface-list=phones
ingress-priority=0
action=set-priority
new-priority=7
```

Bridge filtering sample 4

```
/ip firewall mangle

add action=mark-packet chain=forward
disabled=no priority=7 new-packet-
mark=priority_7 passthrough=no

/queue simple

add name="Priority 7" target=0.0.0/0
priority=7/7 limit-at=xxx/xxx max-
limit=xxx/xxx packet-mark=priority_7
```

Questions?

Thank you!

massimo@dicobit.it