WIRELESS & NETWORKING EXPERTS

# Layer2 Packet Flow

March 2024 © Jono Thompson
Multithread Consultants / LinITX

LinITX
WIRELESS & NETWORKING EXPERTS

MIKROTIK

# E-COMMERCE TRAINING

CONSULTANCY

## ABOUT US:

- LinITX.com eCommerce website started in 2002
- Official MikroTik Master Distributor
- MikroTik Certified Training Partner
- Consultancy including WiFi site surveys
- Two UK warehouses with high stock levels
- Highly trained technical team

SCAN ME

@ team@linitx.com

LinITX.com

01449 888000

## EXPERT TRAINERS:



JONO
**THOMPSON**

Jono has over 20 years of experience in networking. He holds multiple MikroTIk and Ubiquiti qualifications. He is also a fully Certified Ubiquiti and MikroTik Training Partner.



RON
**TOUW**

Ron has 40 years of experience in wireless and networking protocols. He holds multiple certifications in Ubiquiti, MikroTik, Ruckus, Meru, HP, Rohde & Schwarz and more.

# Jono Thompson

- Networking background started as a Cisco Engineer
- Started using ROS June 2010
- MikroTik Consultant Since Dec 2014
- MikroTik Trainer since March 2017
  - MTCNA
  - MTCRE
  - MTCINE
  - MTCWE
  - MTCTCE
  - MTCUME
  - MTCSE
  - MTCIPv6
  - MTCSWE
  - MTCEWE

# Presentation Objectives

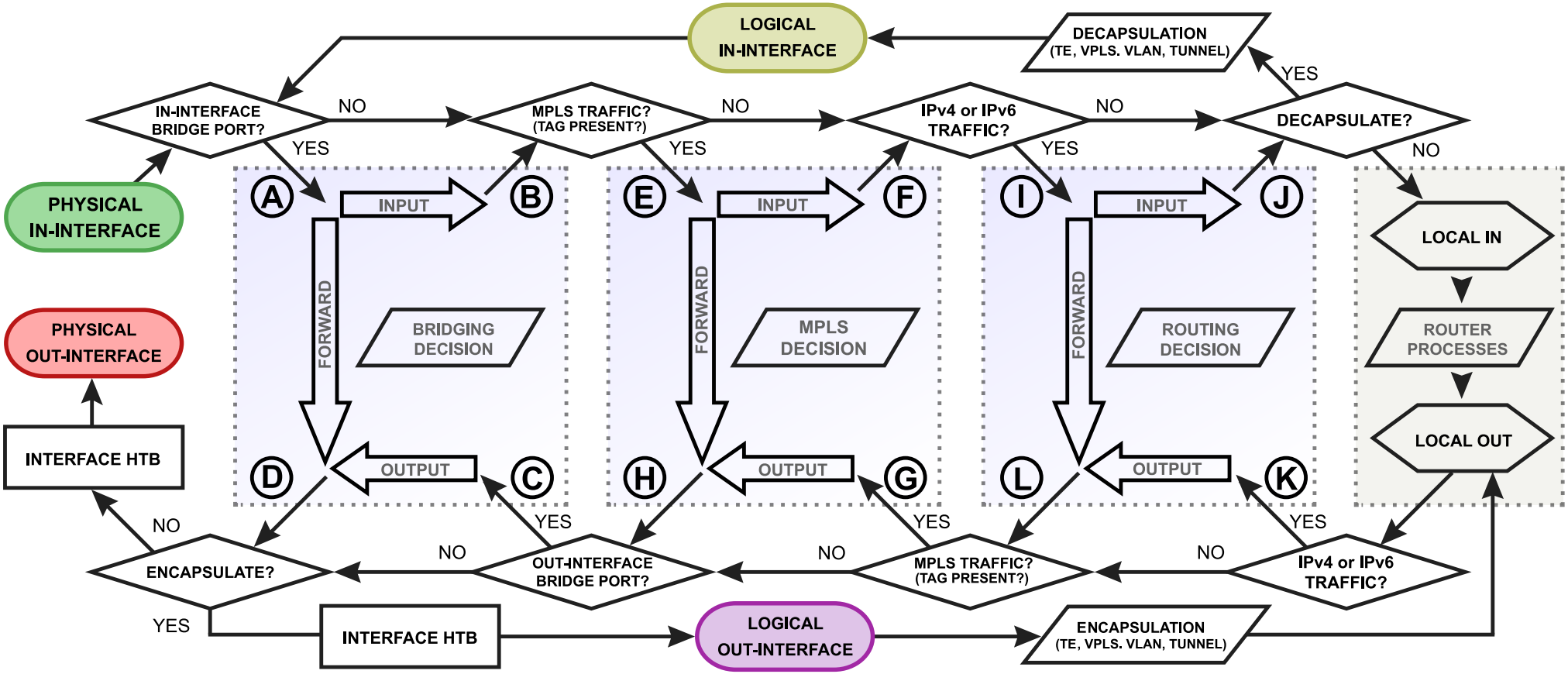
- Look at packet flow though RouterOS bridges when using Hardware Offloading.
- Look at how this changes when adding in a none hardware offloaded port.
- Look at how Layer2 traffic can be managed on a hardware offloaded bridge.

# Packet Flow

- When using more than one RouterOS Features, it is important to understand how these features work together.

- Understanding how packets flow though RouterOS will answer these questions.

# RouterOS Packet Flow



# Flow of a Hardware Offloaded Packet

- Most MikroTik devices are equipped with dedicated switching hardware – the switch chip.
- The switch chip allows us to offload some bridging functions e.g. packet forwarding between bridge ports or packet filtering, onto this specialized hardware chip without consuming any CPU resources.
- In RouterOS this is called Bridge Hardware Offloading.

# MikroTik Switch Chip – Features

- There are several different types of switch chips on RouterBOARDs. These have different features

| Switch Chip | Model (example units) | Port Switching | Port Mirroring | TX Limit[1] | RX Limit[1] | Host Table | VLAN Table | Rule Table |
|---|---|---|---|---|---|---|---|---|
| QCA8337 | hAP ac / hEX PoE | ✓ | ✓ | ✓ | ✓ | 2048 | 4096 | 92 |
| AR8327 | hAP ac[2] | ✓ | ✓ | ✓ | ✓ | 2048 | 4096 | 92 |
| AR8227 | hAP/hEX) | ✓ | ✓ | ✓ | × | 1024 | 4096 | × |
| AR8316 | | ✓ | ✓ | ✓ | × | 2048 | 4096 | 32 |
| AR7240 | | ✓ | ✓ | ✓ | × | 2048 | 16 | × |
| IPQ-PPE | hAP ax[2], hAP ax[3], Chateau ax, cAP ax | ✓ | × | × | × | 2048 | × | × |
| MT7621, MT7531 | hEX (750Gr3), hAP ax lite, | ✓ | ✓ | × | × | 2048 | 4096[3] | × |
| RTL8367 | 1100AHx4/RB4011 | ✓ | ✓ | × | × | 2048 | 4096[3] | × |
| ICPlus175D | | ✓ | ✓ | × | × | × | × | × |
| 88E6393X | RB5009 | ✓ | ✓ | ✓ | ✓ | 16k | 4096[3] | 256 |
| 88E6191X,88E6190 | L009, CCR2004-16G-2S+ | ✓ | ✓ | ✓ | ✓ | 16k | 4096[3] | × |
| 98PX1012 | | × | × | × | × | × | × | × |
| Others | | ✓ | × | × | × | × | × | × |

Complete list https://help.mikrotik.com/docs/display/ROS/Switch+Chip+Features

# Bridge – HW Offloading

- Enabling some bridge features can, depending on the switch chip, disable HW-offloading e.g.:-
  - Spanning Tree
  - Rapid Spanning Tree
  - Multiple Spanning Tree
  - IGMP Snooping
  - DHCP Snooping
  - VLAN Filtering

# Bridge – HW Offloading

- Enabling some bridge features can, depending on the switch chip, disable HW-offloading only on the interface e.g.:-
  - Bonding
  - Bridge Horizon

# Bridge – HW Offloading

| Switch Chip | Model (example units) | STP/RSTP | MSTP | IGMP Snooping | DHCP Snooping | VLAN Filtering | Bonding[4,5] | Horizon |
|---|---|---|---|---|---|---|---|---|
| | CRS3xx, CRS5xx CCR2116, CCR2216 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ×[4] |
| | CRS1xx/2xx | ✓ | × | ✓[2] | ✓[1] | × | × | ×[4] |
| QCA8337 | hAP ac / hEX PoE | ✓ | × | × | ✓[2] | × | × | ×[4] |
| AR8327 | hAP ac[2] | ✓ | × | × | ✓[2] | × | × | ×[4] |
| AR8227 | hAP/hEX lite | ✓ | × | × | × | × | × | ×[4] |
| AR8316 | | ✓ | × | × | ✓[2] | × | × | ×[4] |
| AR7240 | | ✓ | × | × | × | × | × | ×[4] |
| IPQ-PPE[6] | hAP ax[2], hAP ax[3], Chateau ax, cAP ax | × | × | × | × | × | × | ×[4] |
| ICPlus175D | | × | × | × | × | × | × | ×[4] |
| MT7621, MT7531 | hEX (750Gr3) | ✓[3] | ✓[3] | × | × | ✓[3] | × | ×[4] |
| RTL8367 | 1100AHx4/RB4011 | ✓[3] | ✓[3] | × | × | ✓[3] | × | ×[4] |
| 88E6393X, 88E6191X, 88E6190 | RB5009, L009, CCR2004-12G-2S+ | ✓ | ✓ | ✓ | ✓ | ✓[3] | ✓[7] | ×[4] |

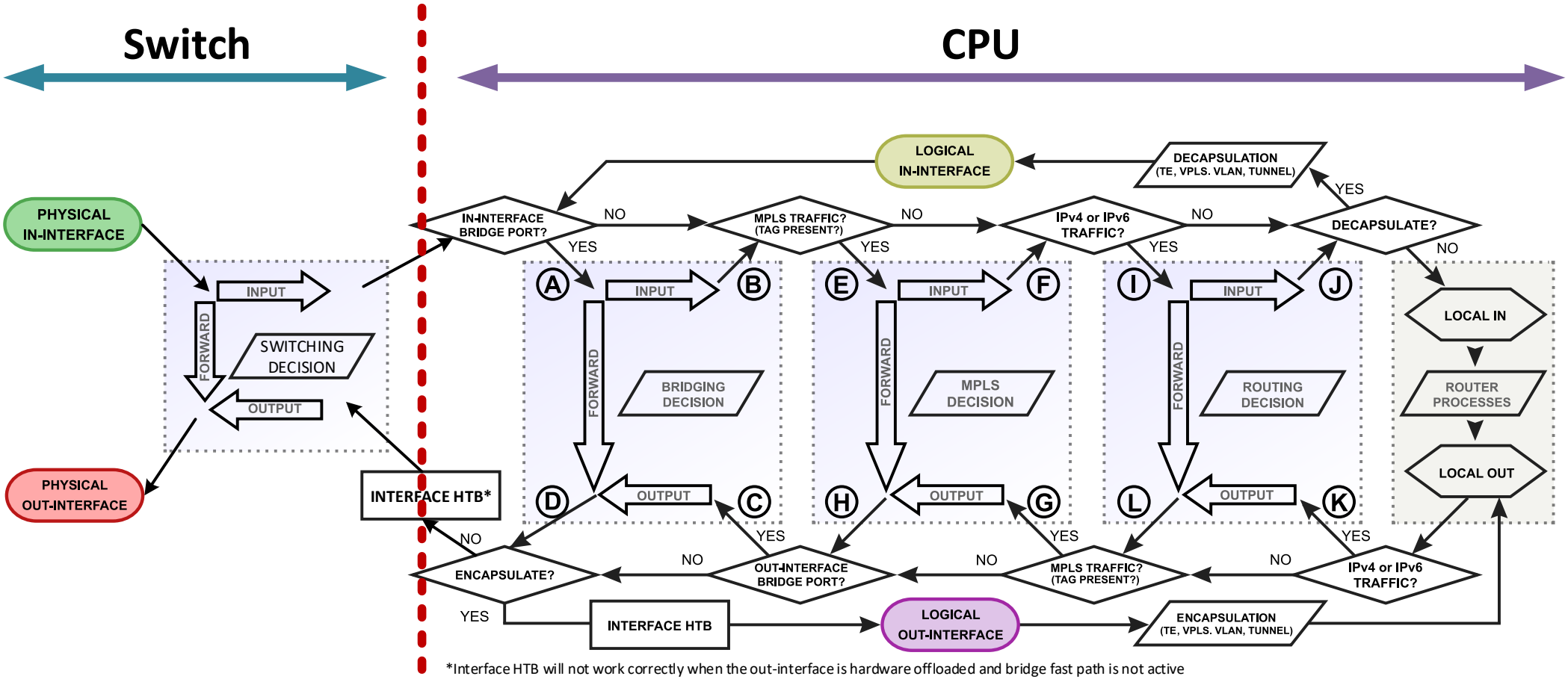
1. Feature will not work properly in VLAN switching setups, however, can be achieved using a switch ACL rule
2. Feature will not work properly in VLAN switching setups.
3. Hardware offloading for vlan-filtering only for ether-type 0x8100. The use of other ether-types and tag-stacking will disable hardware offloading.
4. Hardware offloading will only be disable for the specific bridge port not the entire bridge.
5. Bridge hardware offloading only supported using 802.3ad bonding and balance-xor bonding modes.
6. Hardware offloading support for IPQ-PPE is currently incomplete. It is recommended to use none-hw offloading bridge by enabling RSTP on the bridge
7. 802.3ad mode is only supported when R/M/STP is enabled on the bridge

Complete list https://help.mikrotik.com/docs/display/ROS/Bridging+and+Switching#BridgingandSwitching-BridgeHardwareOffloading.

# Bridge HW Offloading Packet Flow

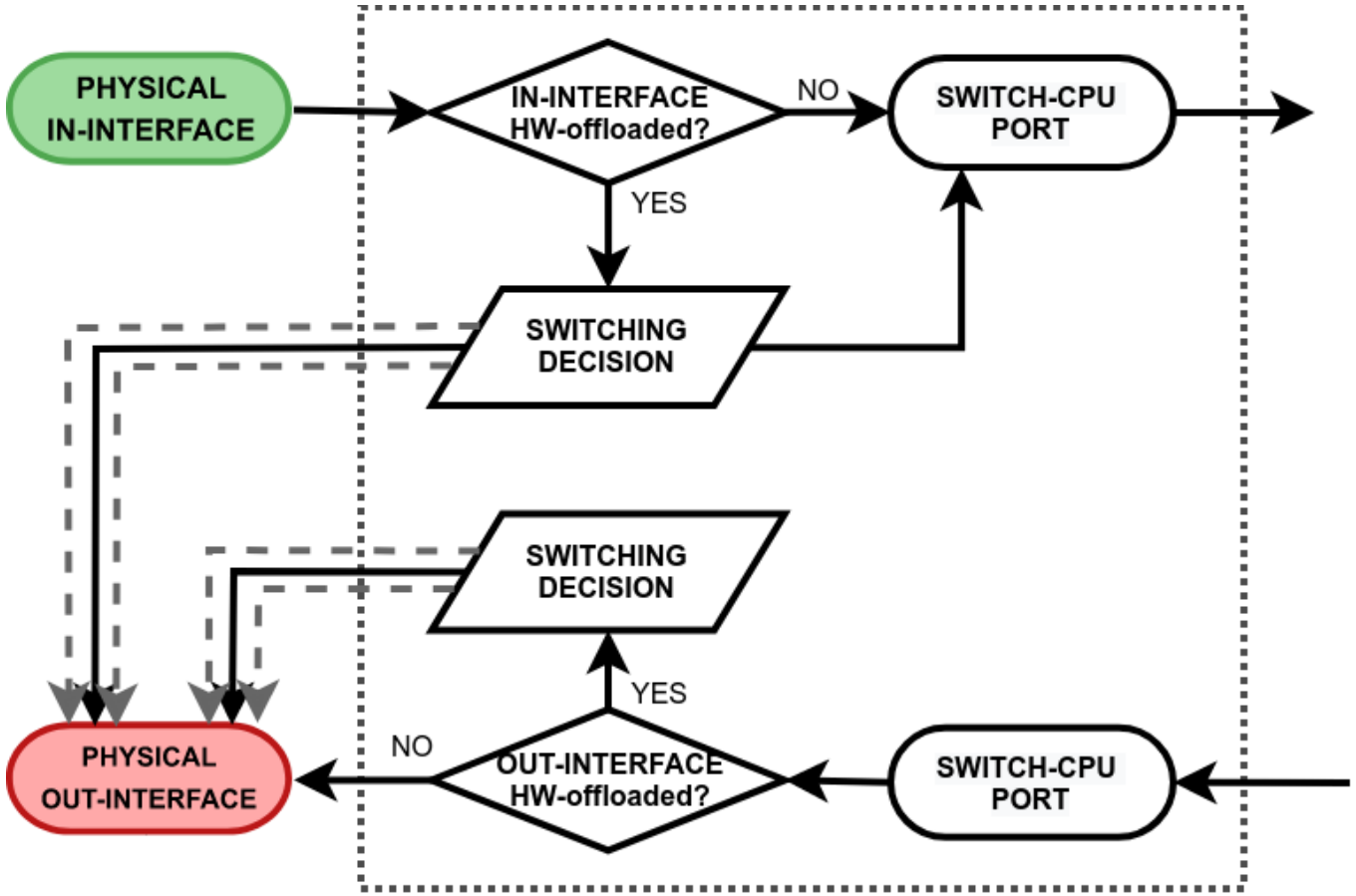
- Adding Bridge Hardware Offloading adds new elements and logic gates to the packet flow.
- Hardware Offloading does not restrict a device to only hardware limited features.
- It is possible to take advantage of both hardware and software processing at the same time.
- This requires understanding of how the packets flow.

# Packet Flow with a Switch Chip




*Interface HTB will not work correctly when the out-interface is hardware offloaded and bridge fast path is not active

# Switching


PHYSICAL
IN-INTERFACE
IN-INTERFACE
HW-offloaded?
NO
SWITCH-CPU
PORT
YES
SWITCHING
DECISION
SWITCHING
DECISION
YES
PHYSICAL
OUT-INTERFACE
NO
OUT-INTERFACE
HW-offloaded?
SWITCH-CPU
PORT

# Switching

- Inside the switching block, there are two processes
  - Switching Decision
  - Switch-CPU-Port

# Switching Decision

- Functions widely depend on the switch model.
- Controls all the switching related tasks includes:-
  - host learning,
  - packet forwarding,
  - filtering,
  - ~~rate limiting~~,
  - VLAN tagging/untagging,
  - ~~mirroring,~~ etc.
- Certain switch configuration can alter the packet flow.

# Switch-CPU Port

- Switch-CPU a special purpose switch port for communication between the main CPU and other switch ports.
- The switch-cpu port does not show up anywhere on RouterOS except for the switch menu.
- None of the software related configuration (e.g. interface-list) can be applied to this port.
- Packets that reach the CPU are automatically associated with the physical in-interface.

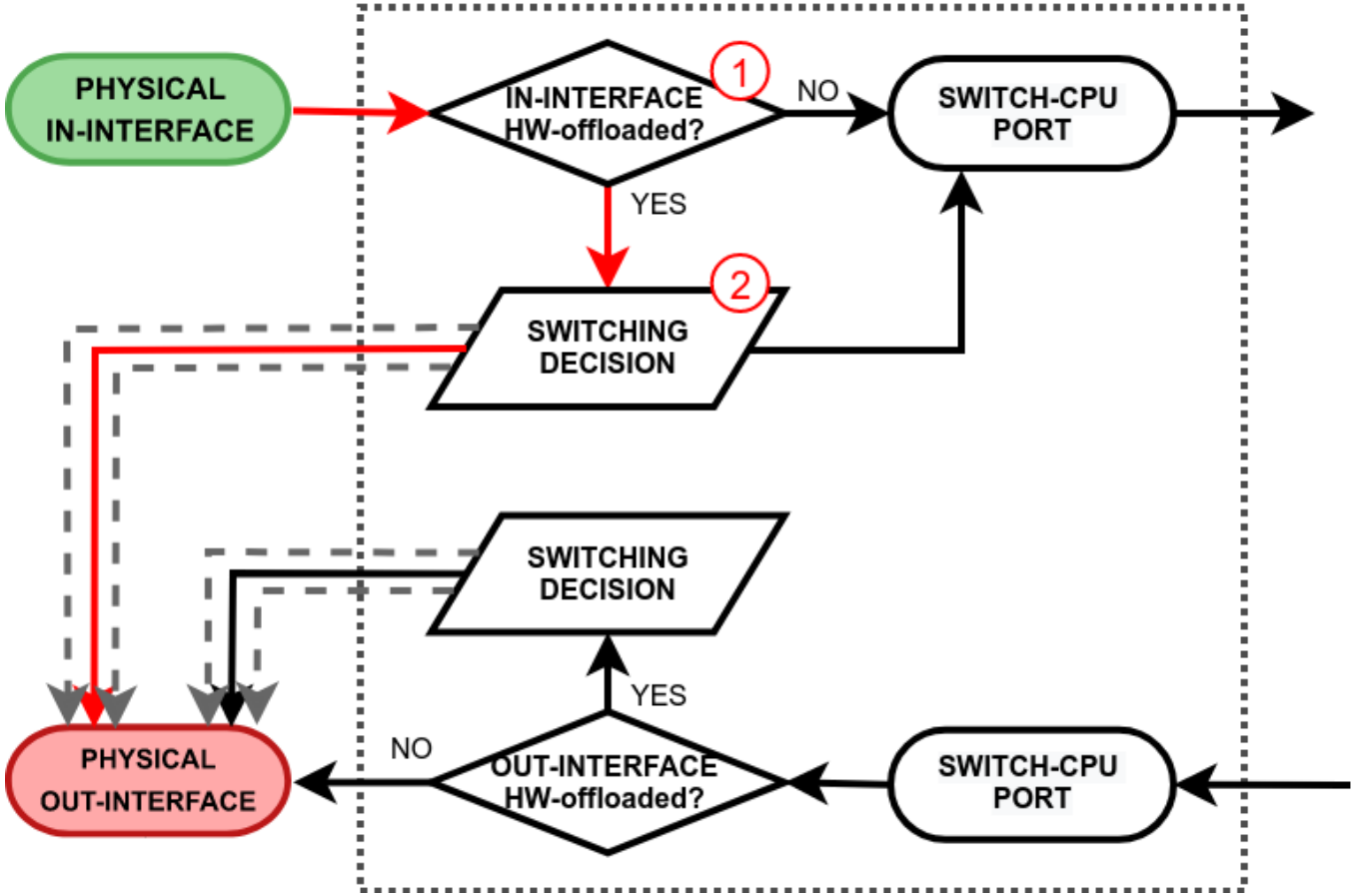
# Flow of a HW Offloaded Packet

- There are 3 ways a packet may flow though the switching logic:-
  - Switch Forward
  - Switch Input
  - Switch Output

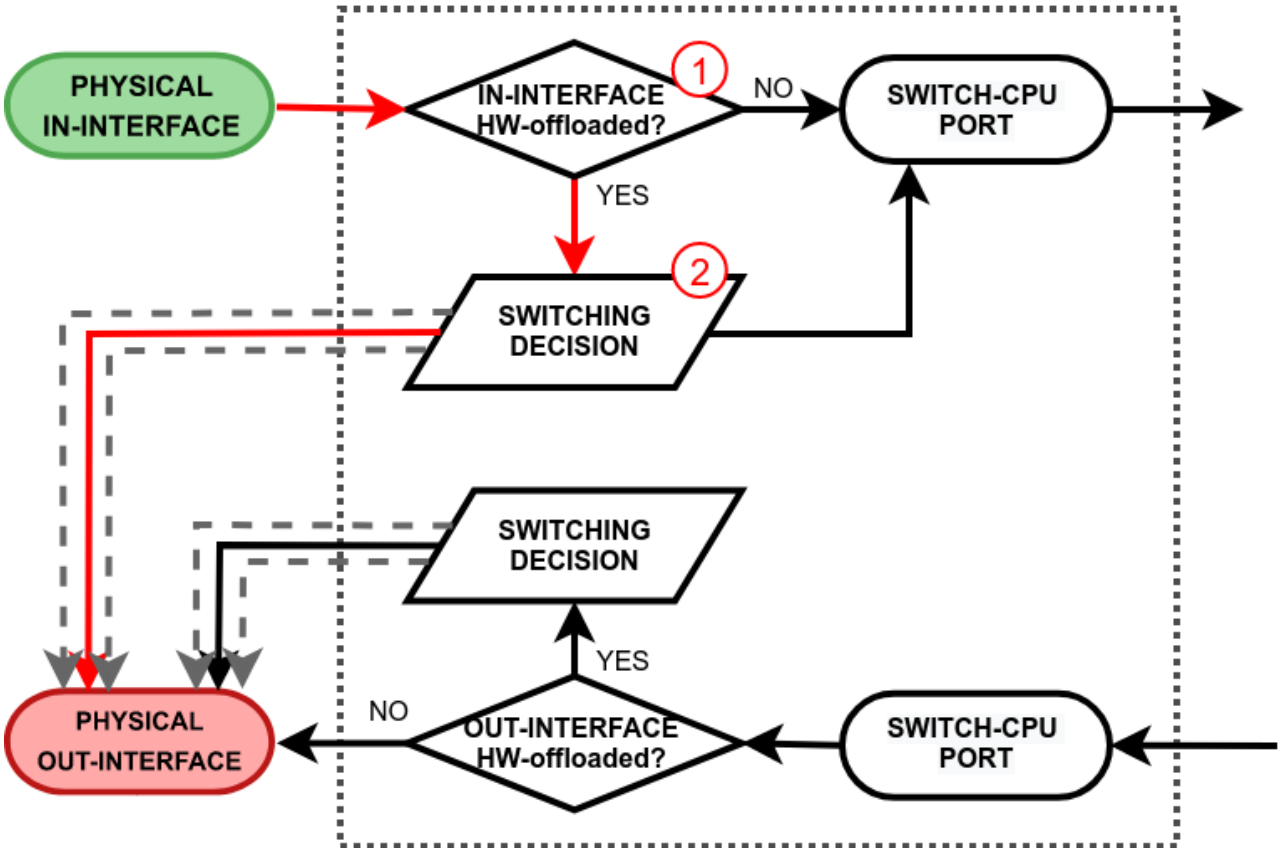
# Switch Forward

- When a packet is forward between two switch ports on the same switch chip and the interfaces are hardware offloaded.

# Switch Forward

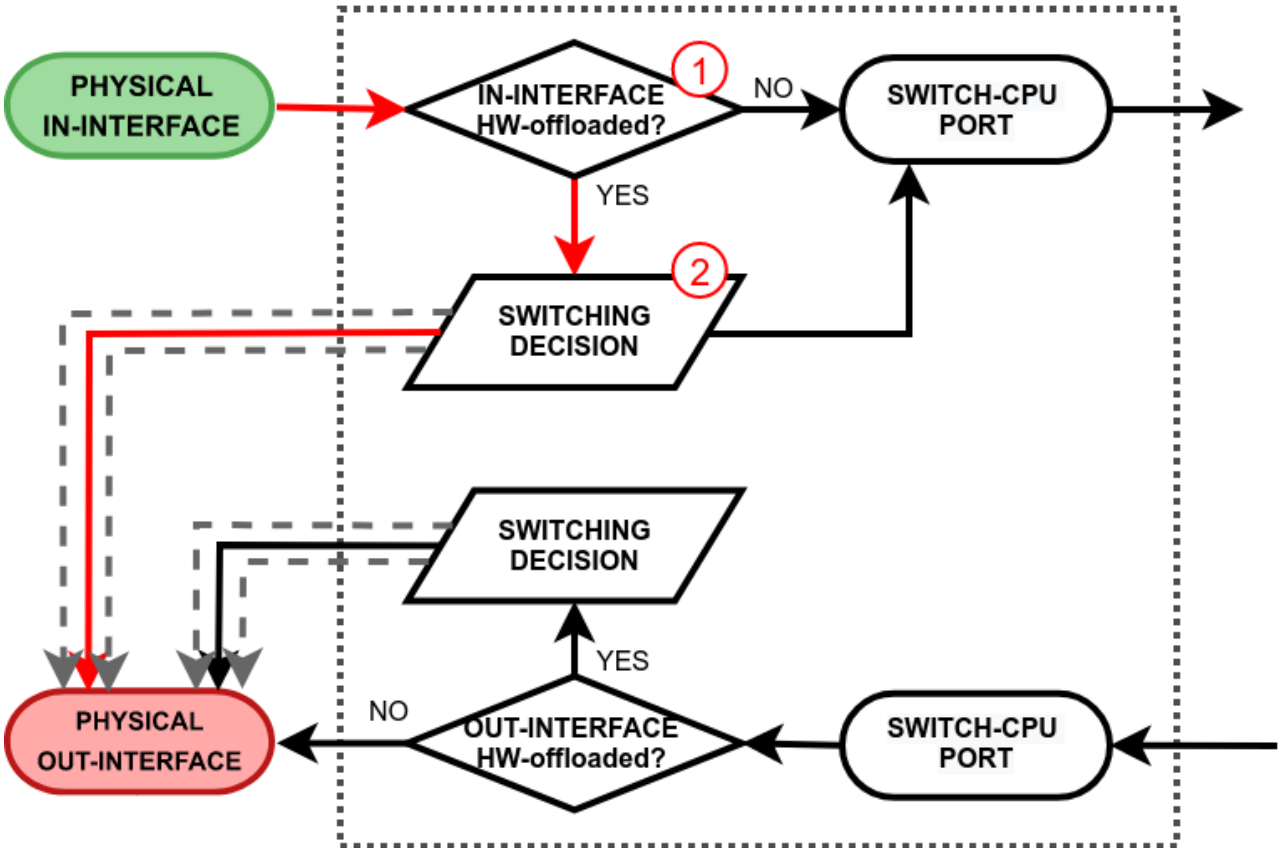


# Switch Forward



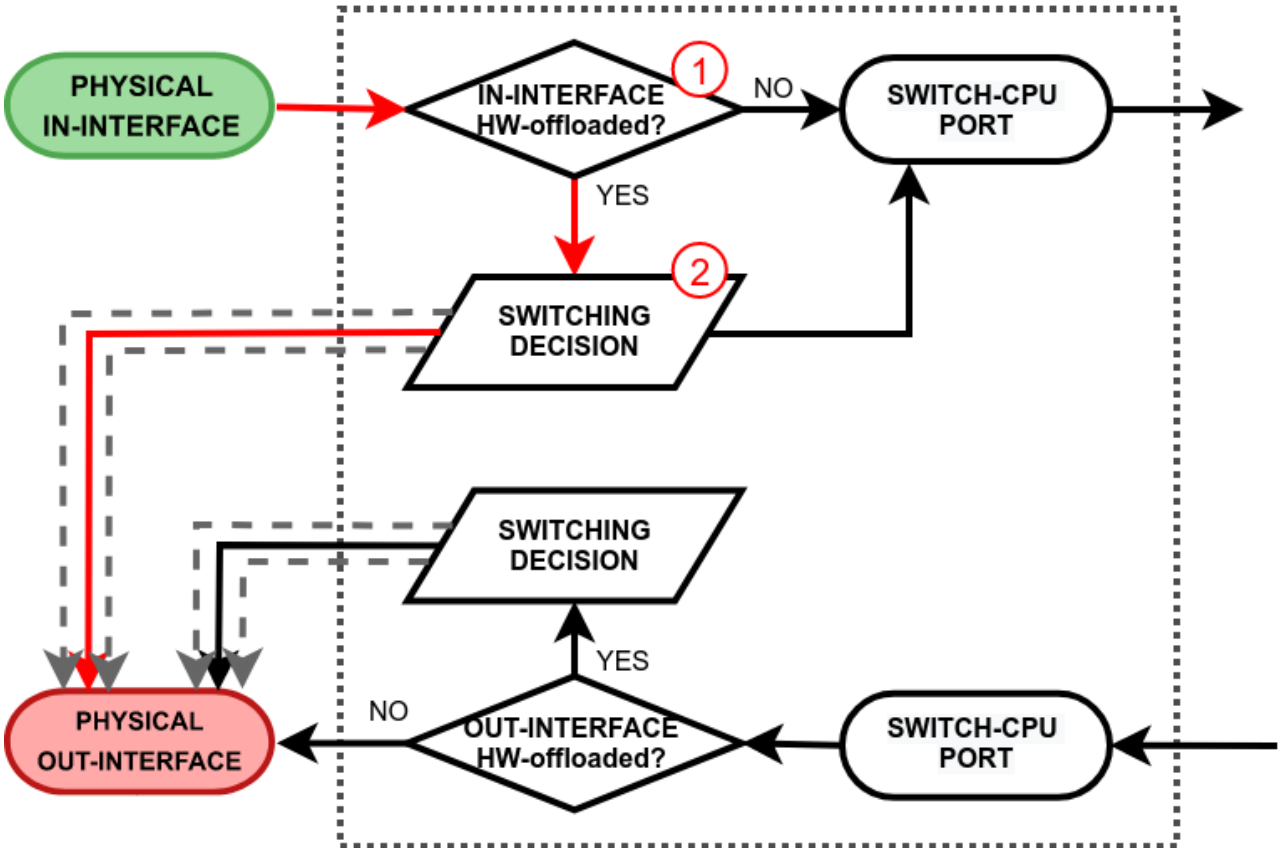

The switch checks whether the in-interface is a hardware offloaded interface.

# Switch Forward



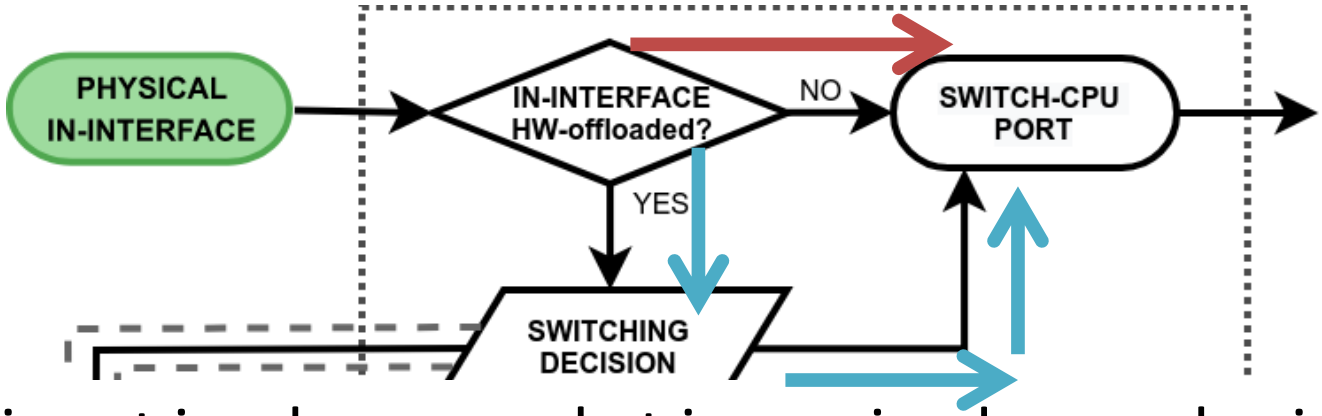

The packet through the switch host table to make a forwarding decision. If the switch finds a match for the destination MAC address, the packet is sent out through the physical interface.

# Switch Forward



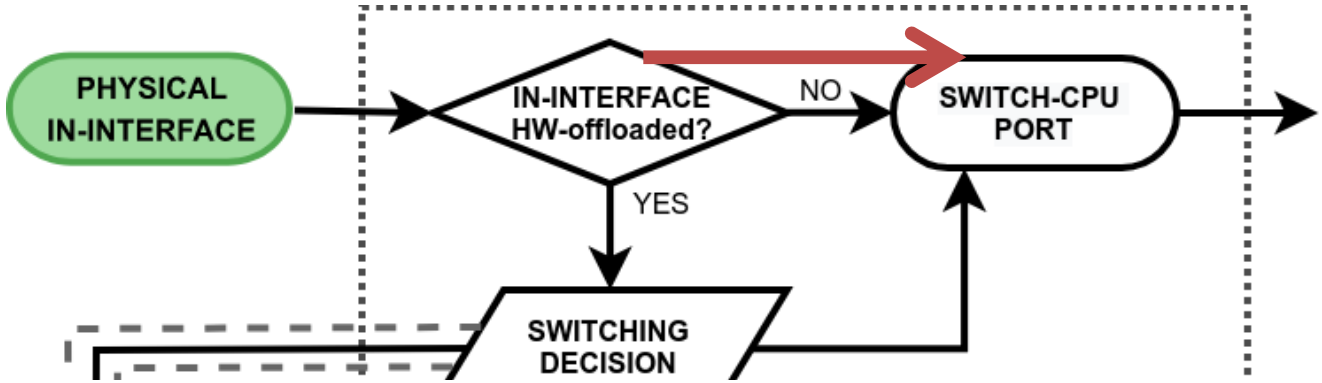

A packet that ends up being flooded (e.g. broadcast, multicast, unknown unicast traffic) gets multiplied and sent out to every hardware offloaded switch port.

# Switch to CPU Input




- Switch input is when a packet is received on a physical interface and it is destined to the switch-cpu port for further software processing
- There are 2 paths to the switch-cpu port.

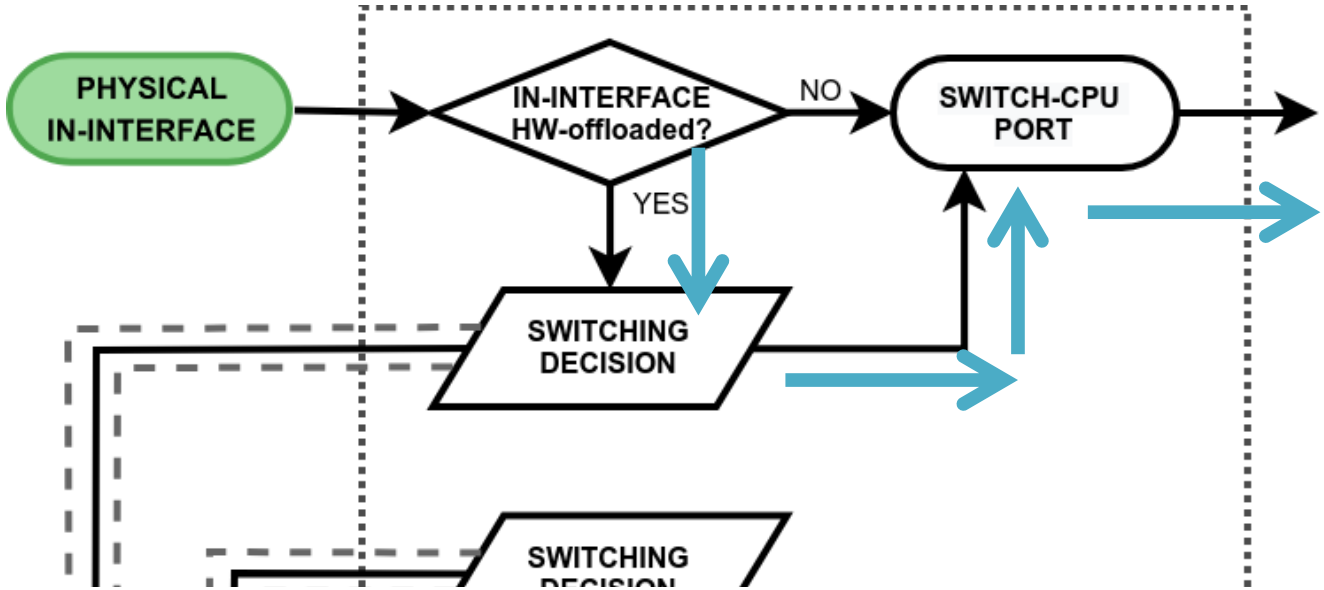
# Switch to CPU Input




1. Hardware offloading and switching is not used e.g.
   - a standalone interface for routing
   - a bridged interface with disabled HW-offloading

   The packet is simply passed to the CPU for further processing by RouterOS.

# Switch to CPU Input




2. When hw-offloading is active on the in-interface. This causes the packet to pass through to the switching decision. This may happen for several reasons:

# Switch to CPU Input

- Dst MAC-Address matches with a local MAC address e.g. when a packet is destined to a local bridge interface
- A packet is flooded to all switch ports (inc Switch-CPU) e.g. broadcast or multicast traffic or unknown unicast is received
- Switch may have learnt that some hosts can only be reached though Switch-CPU Port, e.g. none hw offloaded interfaces (wireless, EoIP and some ethernet interfaces)

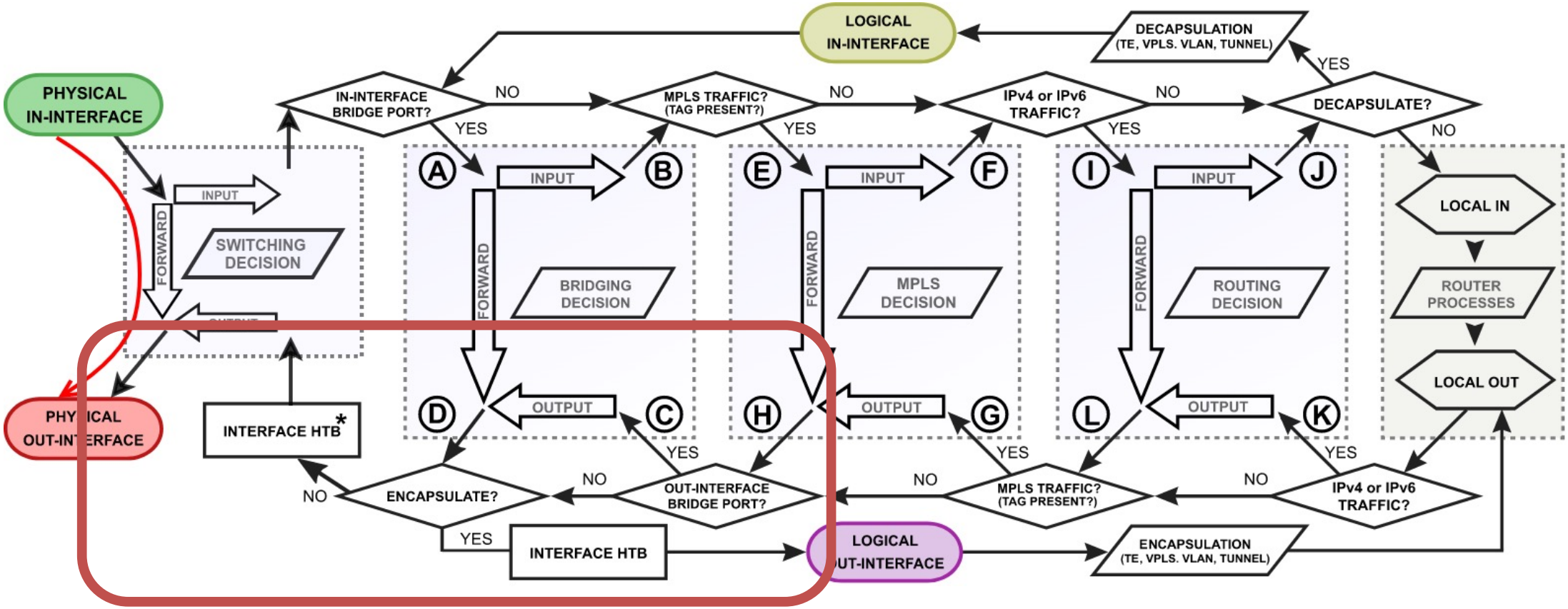
# Switch to CPU Input

- Packet is copied to the switch-cpu e.g. for packet inspection.
- Packet is trigged by the switch configuration and should be processed in software e.g. DHCP or IGMP snooping.

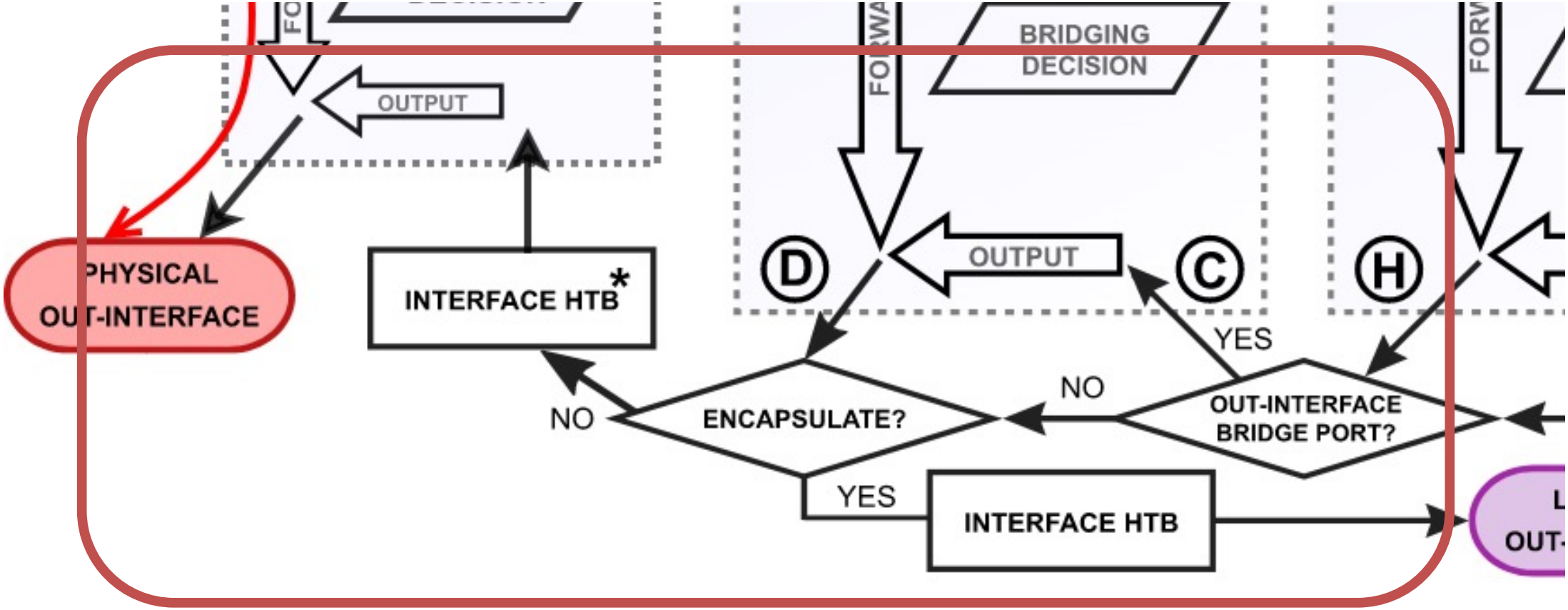
# CPU Output to Switch

- This happens when a packet exits RouterOS software processing and it is received on the switch CPU port.

# CPU Output to Switch

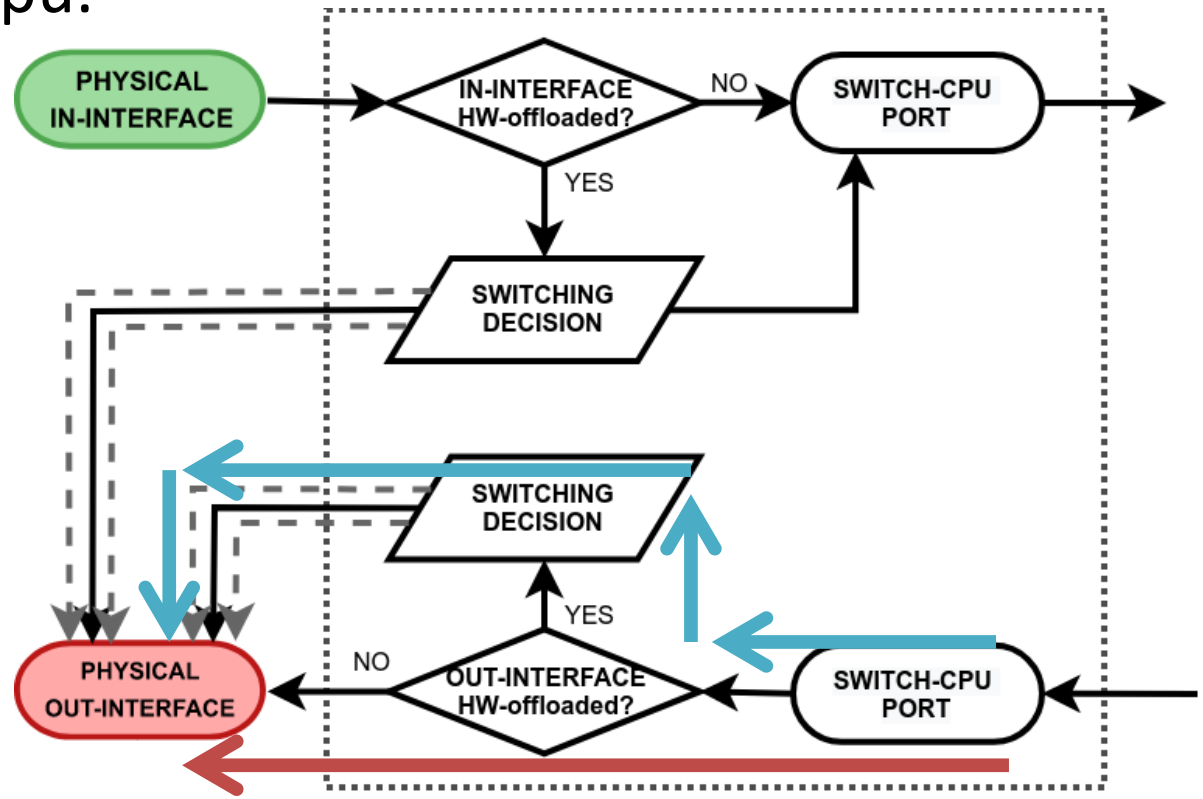


# CPU Output to Switch



# CPU Output to Switch

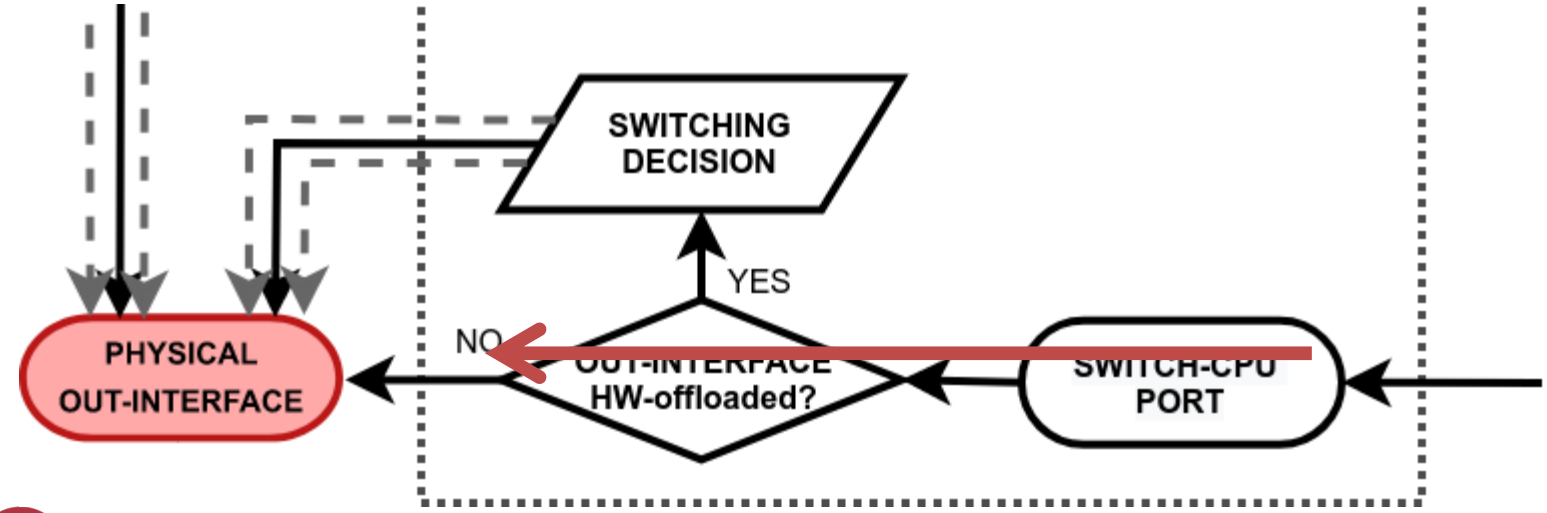
- There are 2 paths a packet can take after leaving the swich-cpu.



# CPU Output to Switch

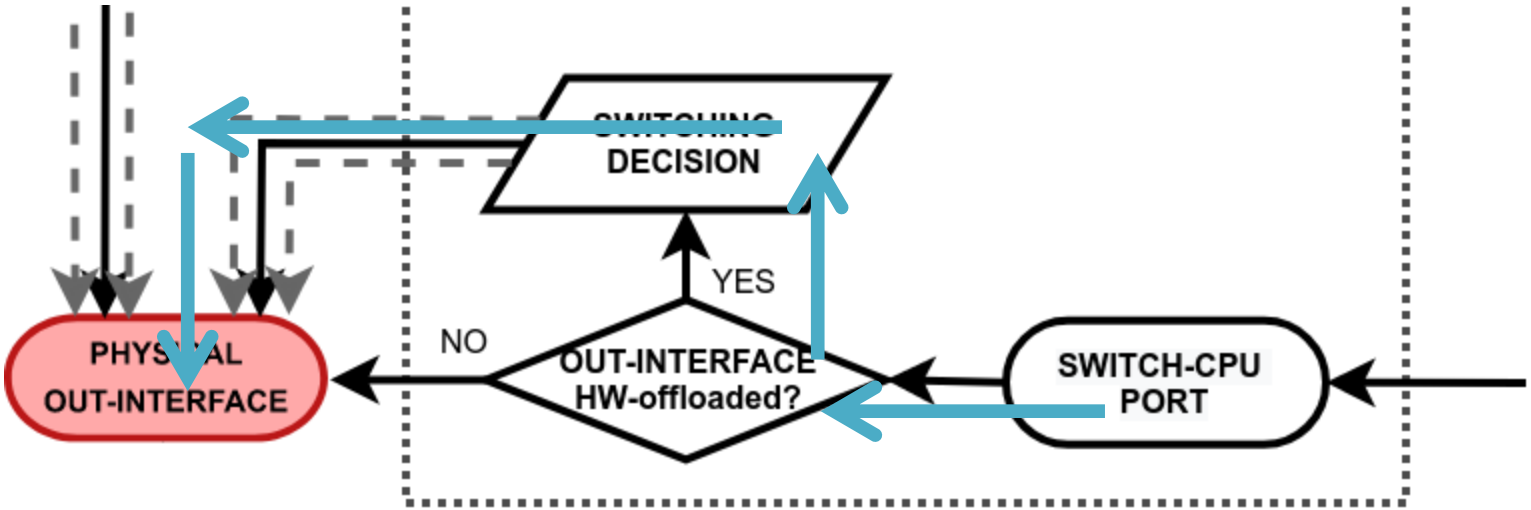
1. Where hardware offloading and switching is not even used e.g.
   - A standalone interface used for routing
   - a bridged interface with hw-offloading disabled

   Here the packet is just sent out of the physical interface.



# CPU Output to Switch

2. When HW-offloading is active on the out-interface this will cause the packet to pass though the switching decision and will learn the src-mac address from the packet.



# CPU Output to Switch

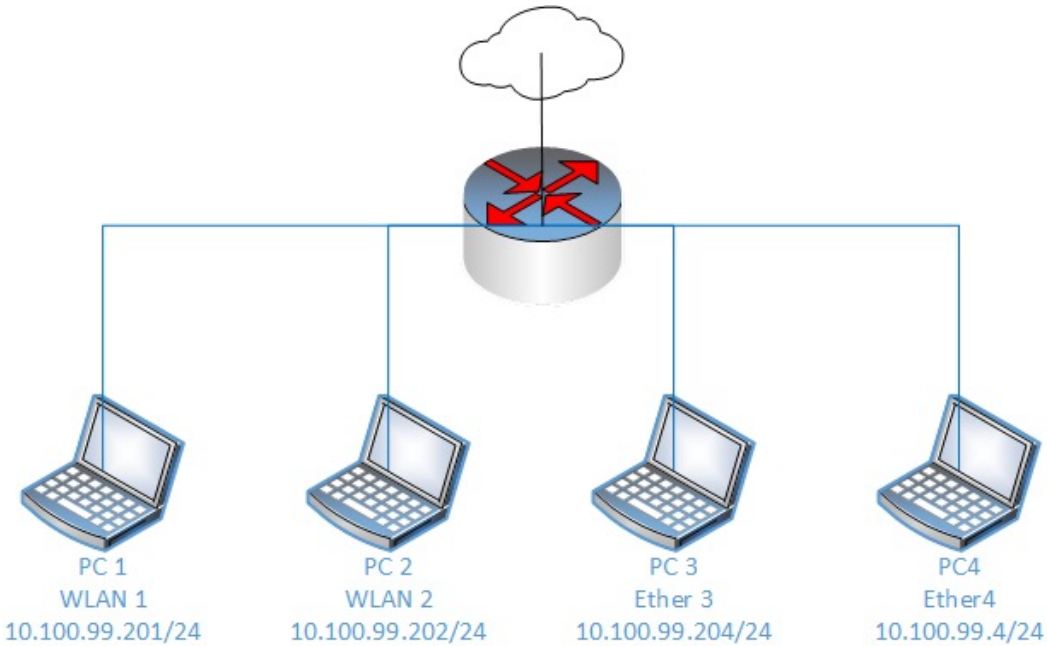
- The switching decision of learning mac addresses is especially useful when a bridge contains both HW and none-hw offloaded interfaces e.g. ethernet and wireless.
- The switch can learn which frames needed to be forward to the CPU.

# Hardware and Software Bridge

- Hardware offloading does not restrict a device to only hardware limited features.
- It is possible to take advantage of the hardware and software processing at the same time.
- This does require a very deep understanding of how packets travel through the switch chip and when exactly they are passed to the main CPU.

# Example Config – hAPac^2

- Ether 5 – WAN
- Ether 1-4, all in bridge-lan.



# Example Config – hAPac^2

- Ether3 and Ether4 have hw-offloading disabled

Bridge

Bridge | Ports | Port Extensions | VLANs | MSTIs | Port MST Overrides | Filters | NAT | Hosts | MDB

Find

| # | | Interface | Bridge | Horizon | Trusted | Priority (... | Path Cost | PVID | Role |
|---|---|---|---|---|---|---|---|---|---|
| 0 | H | ether1 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 1 | H | ether2 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 2 | | ether3 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 3 | | ether4 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 4 | I | wlan1 | bridge-lan | | no | | | | |
| 5 | I | wlan2 | bridge-lan | | no | | | | |

6 items (1 selected)

Terminal <2>

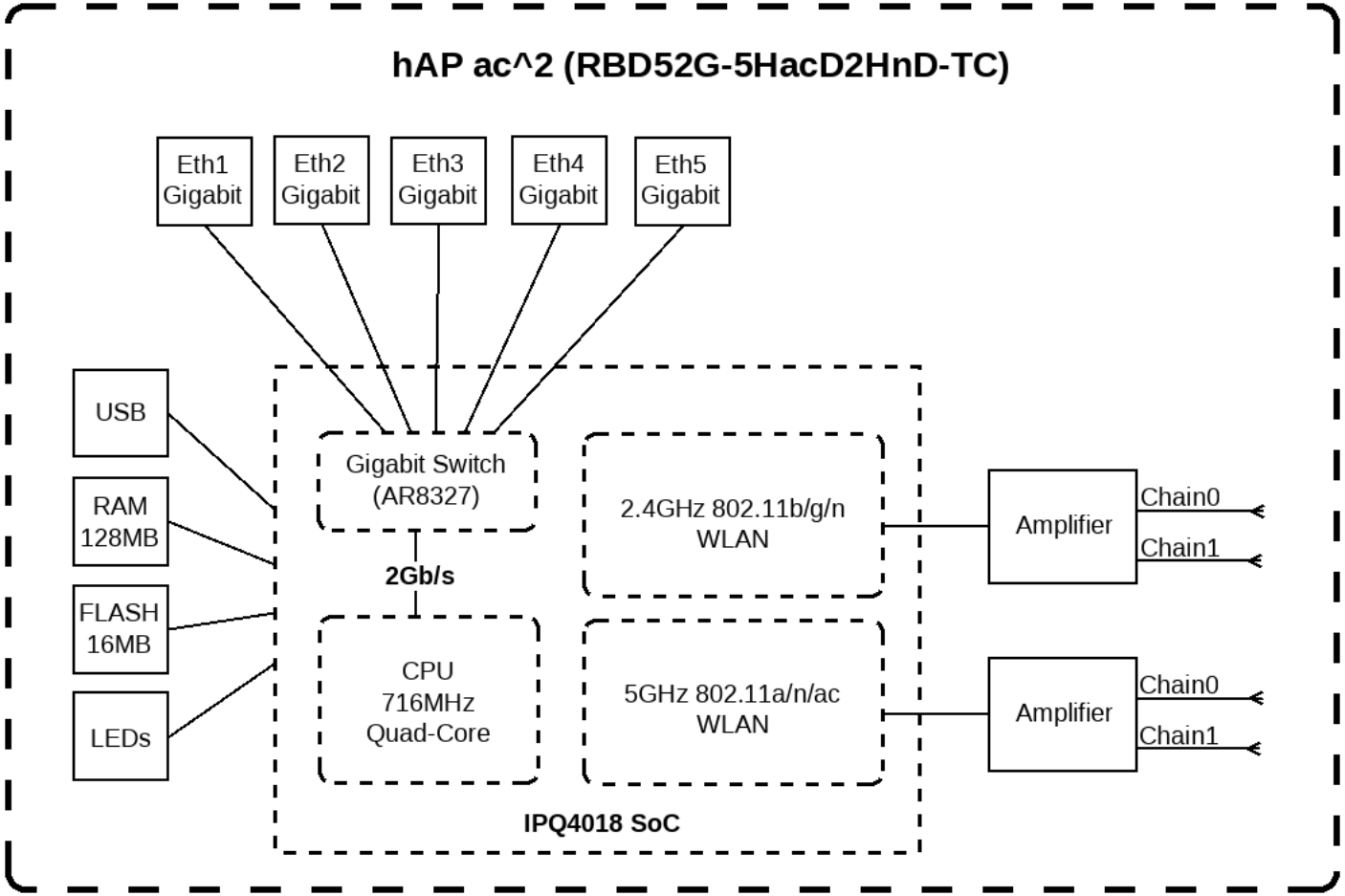
```
/interface bridge port
add bridge=bridge-lan interface=ether1
add bridge=bridge-lan interface=ether2
add bridge=bridge-lan hw=no interface=ether3
add bridge=bridge-lan hw=no interface=ether4
add bridge=bridge-lan interface=wlan1
add bridge=bridge-lan interface=wlan2
[admin@MikroTik] /interface/bridge/port>
```

Terminal <2>

```
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL>
#    INTERFACE  BRIDGE      HW   PVID  PRIORITY  PA  IN  HORIZON
0  H ether1     bridge-lan  yes     1  0x80      10  10  none
1  H ether2     bridge-lan  yes     1  0x80      10  10  none
2    ether3     bridge-lan  no      1  0x80      10  10  none
3    ether4     bridge-lan  no      1  0x80      10  10  none
4  I wlan1      bridge-lan          1  0x80      10  10  none
5  I wlan2      bridge-lan          1  0x80      10  10  none

[admin@MikroTik] /interface/bridge/port>
```

# hAPac^2 – block digram



# MikroTik Switch Chip – Features

- There are several different types of switch chips on RouterBOARDs. These have different features

| Switch Chip | Model (example units) | Port Switching | Port Mirroring | TX Limit[1] | RX Limit[1] | Host Table | VLAN Table | Rule Table |
|---|---|---|---|---|---|---|---|---|
| QCA8337 | hAP ac / hEX PoE | ✓ | ✓ | ✓ | ✓ | 2048 | 4096 | 92 |
| AR8327 | hAP ac[2] | ✓ | ✓ | ✓ | ✓ | 2048 | 4096 | 92 |
| AR8227 | hAP(hEX) | ✓ | ✓ | ✓ | × | 1024 | 4096 | × |
| AR8316 | | ✓ | ✓ | ✓ | × | 2048 | 4096 | 32 |
| AR7240 | | ✓ | ✓ | ✓ | × | 2048 | 16 | × |
| IPQ-PPE | hAP ax[2], hAP ax[3], Chateau ax, cAP ax | ✓ | × | × | × | 2048 | × | × |
| MT7621, MT7531 | hEX (750Gr3), hAP ax lite, | ✓ | ✓ | × | × | 2048 | 4096[3] | × |
| RTL8367 | 1100AHx4/RB4011 | ✓ | ✓ | × | × | 2048 | 4096[3] | × |
| ICPlus175D | | ✓ | ✓ | × | × | × | × | × |
| 88E6393X | RB5009 | ✓ | ✓ | ✓ | ✓ | 16k | 4096[3] | 256 |
| 88E6191X,88E6190 | L009, CCR2004-16G-2S+ | ✓ | ✓ | ✓ | ✓ | 16k | 4096[3] | × |
| 98PX1012 | | × | × | × | × | × | × | × |
| Others | | ✓ | × | × | × | × | × | × |

Complete list https://help.mikrotik.com/docs/display/ROS/Switch+Chip+Features

# Bridge – HW Offloading

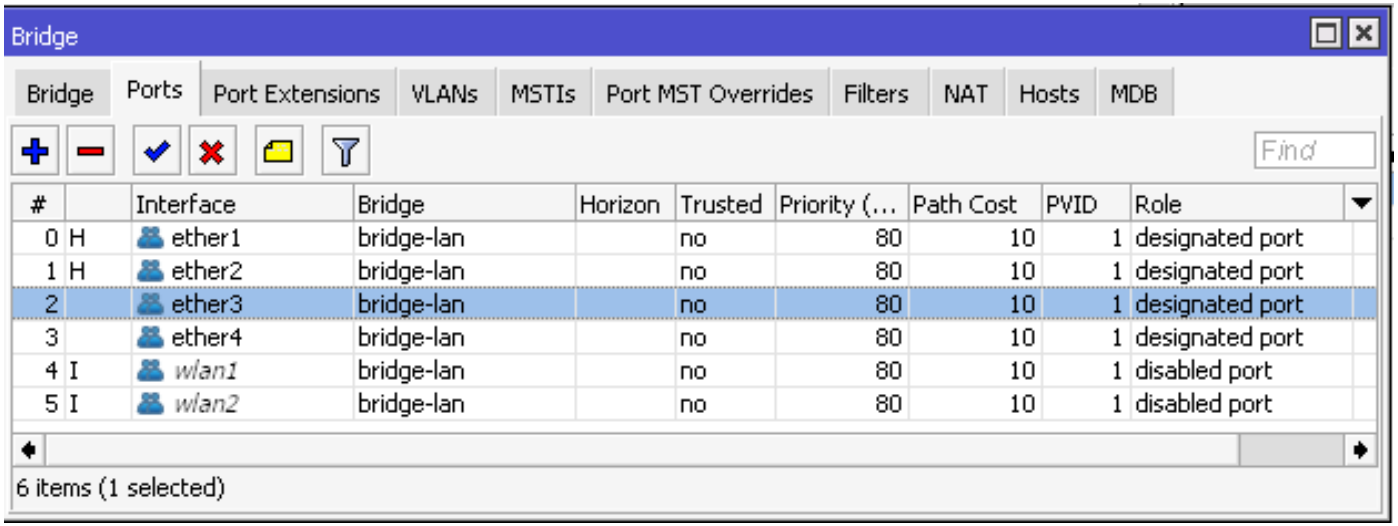
| Switch Chip | Model (example units) | STP/RSTP | MSTP | IGMP Snooping | DHCP Snooping | VLAN Filtering | Bonding[4,5] | Horizon |
|---|---|---|---|---|---|---|---|---|
| | CRS3xx, CRS5xx CCR2116, CCR2216 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ×[4] |
| | CRS1xx/2xx | ✓ | × | ✓[2] | ✓[1] | × | × | ×[4] |
| QCA8337 | hAP ac/ hEX PoE | ✓ | × | × | ✓[2] | × | × | ×[4] |
| AR8327 | hAP ac[2] | ✓ | × | × | ✓[2] | × | × | ×[4] |
| AR8327 | hAP/hEX lite | ✓ | × | × | × | × | × | ×[4] |
| AR8316 | | ✓ | × | × | ✓[2] | × | × | ×[4] |
| AR7240 | | ✓ | × | × | × | × | × | ×[4] |
| IPQ-PPE[6] | hAP ax[2], hAP ax[3], Chateau ax, cAP ax | × | × | × | × | × | × | ×[4] |
| ICPlus175D | | × | × | × | × | × | × | ×[4] |
| MT7621, MT7531 | hEX (750Gr3) | ✓[3] | ✓[3] | × | × | ✓[3] | × | ×[4] |
| RTL8367 | 1100AHx4/RB4011 | ✓[3] | ✓[3] | × | × | ✓[3] | × | ×[4] |
| 88E6393X, 88E6191X, 88E6190 | RB5009, L009, CCR2004-12G-2S+ | ✓ | ✓ | ✓ | ✓ | ✓[3] | ✓[7] | ×[4] |

1. Feature will not work properly in VLAN switching setups, however, can be achieved using a switch ACL rule
2. Feature will not work properly in VLAN switching setups.
3. Hardware offloading for vlan-filtering only for ether-type 0x8100. The use of other ether-types and tag-stacking will disable hardware offloading.
4. Hardware offloading will only be disable for the specific bridge port not the entire bridge.
5. Bridge hardware offloading only supported using 802.3ad bonding and balance-xor bonding modes.
6. Hardware offloading support for IPQ-PPE is currently incomplete. It is recommended to use none-hw offloading bridge by enabling RSTP on the bridge
7. 802.3ad mode is only supported when R/M/STP is enabled on the bridge

Complete list https://help.mikrotik.com/docs/display/ROS/Bridging+and+Switching#BridgingandSwitching-BridgeHardwareOffloading.

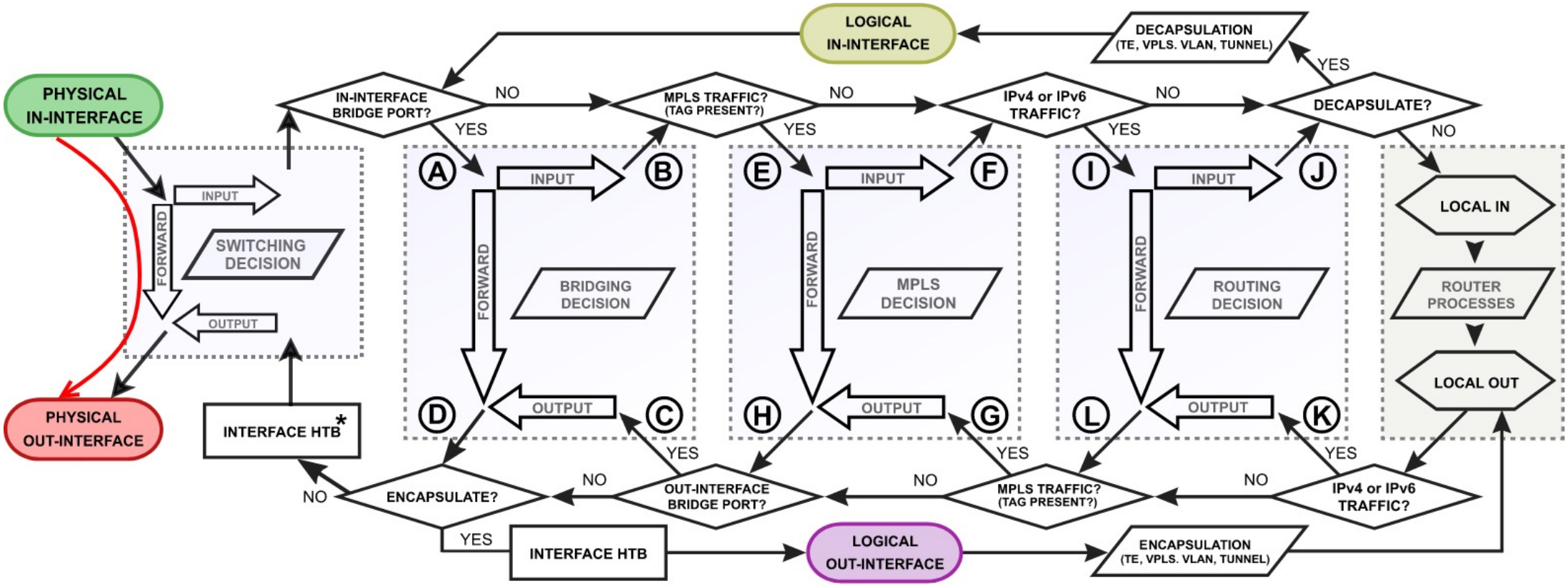
# Example 1 – In HW Offloaded / Out HW-Offloaded

- Traffic flow between Ether1 (PC1) and Ether2 (PC2).
- Bridge hardware offloading in enabled.
- Packet is forwarded between two switch ports on a single switch.

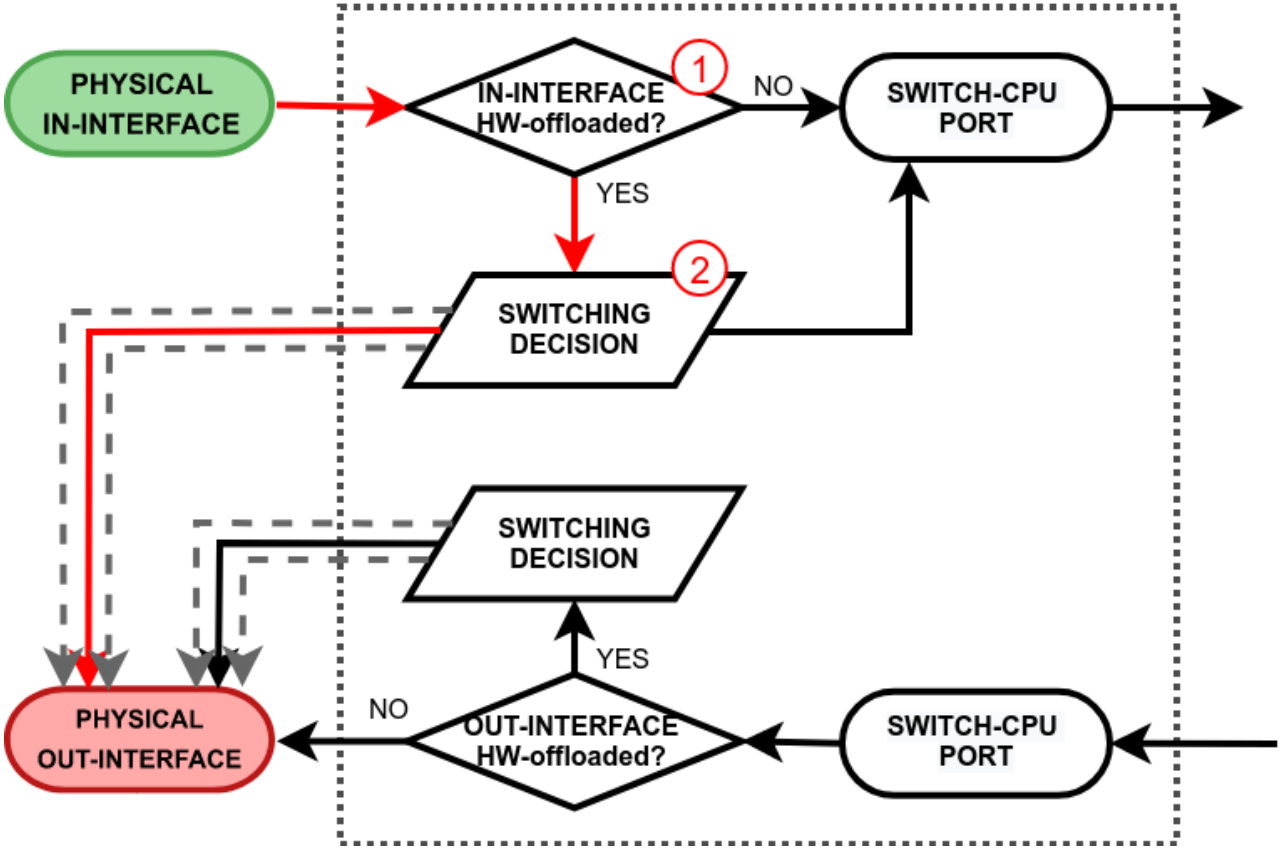



| # | | Interface | Bridge | Horizon | Trusted | Priority (... | Path Cost | PVID | Role |
|---|---|---|---|---|---|---|---|---|---|
| 0 | H | ether1 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 1 | H | ether2 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 2 | | ether3 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 3 | | ether4 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 4 | I | *wlan1* | bridge-lan | | no | 80 | 10 | 1 | disabled port |
| 5 | I | *wlan2* | bridge-lan | | no | 80 | 10 | 1 | disabled port |

# Example 1 – In HW Offloaded / Out HW-Offloaded

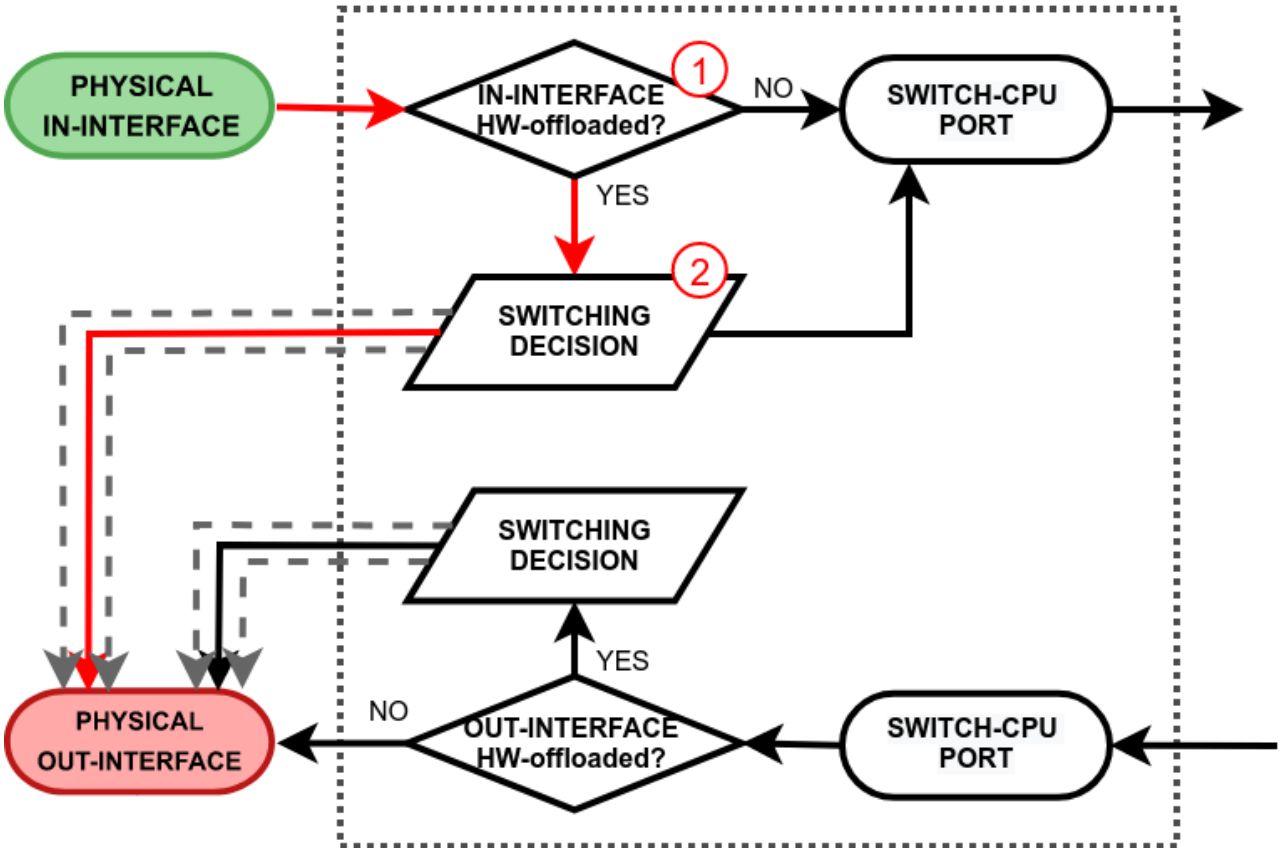


# Example 1 – In HW Offloaded / Out HW-Offloaded




The switch checks whether the in-interface is a hardware offloaded interface.

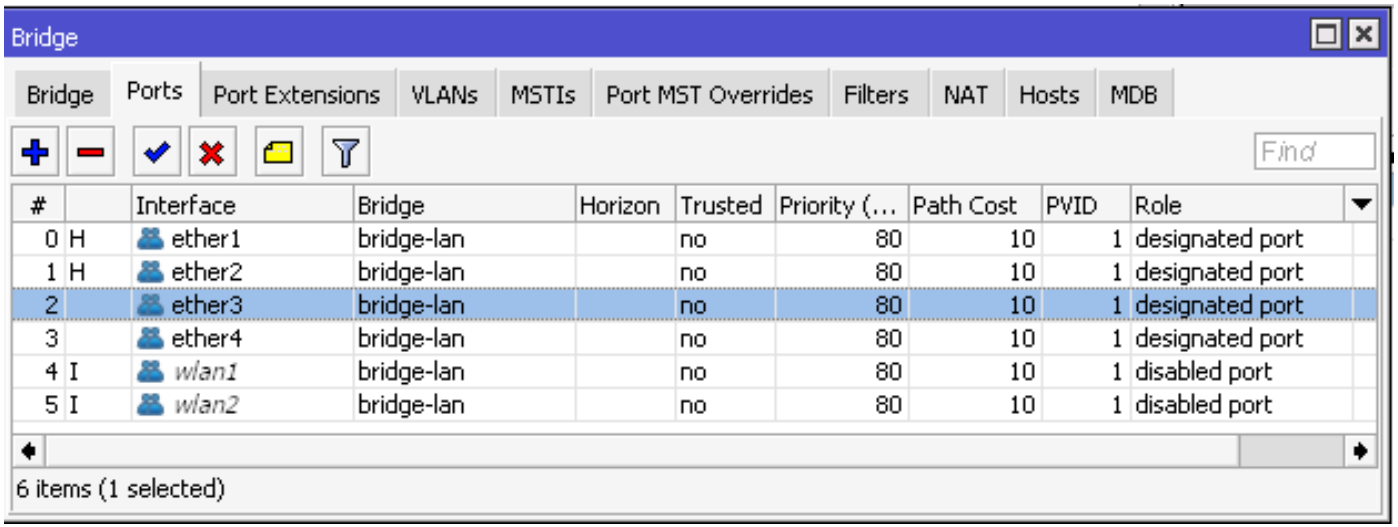
# Example 1 – In HW Offloaded / Out HW-Offloaded




The packet through the switch host table to make a forwarding decision. If the switch finds a match for the destination MAC address, the packet is sent out through the physical interface.

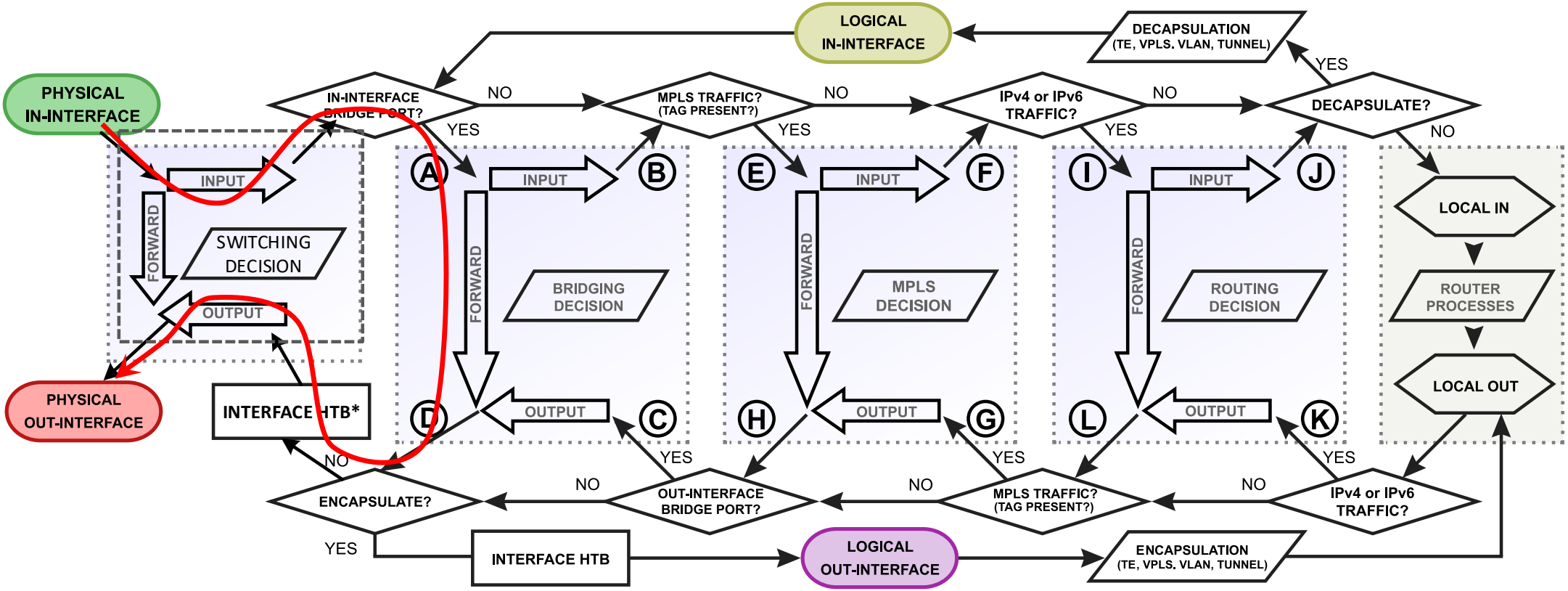
# Example 2 – In HW Offloaded / Out  Not HW-Offloaded

- Traffic Flow between Ether1 (PC1) and Ether4 (PC4)
- In-interface HW offloaded Out-Interface not hw-offloaded



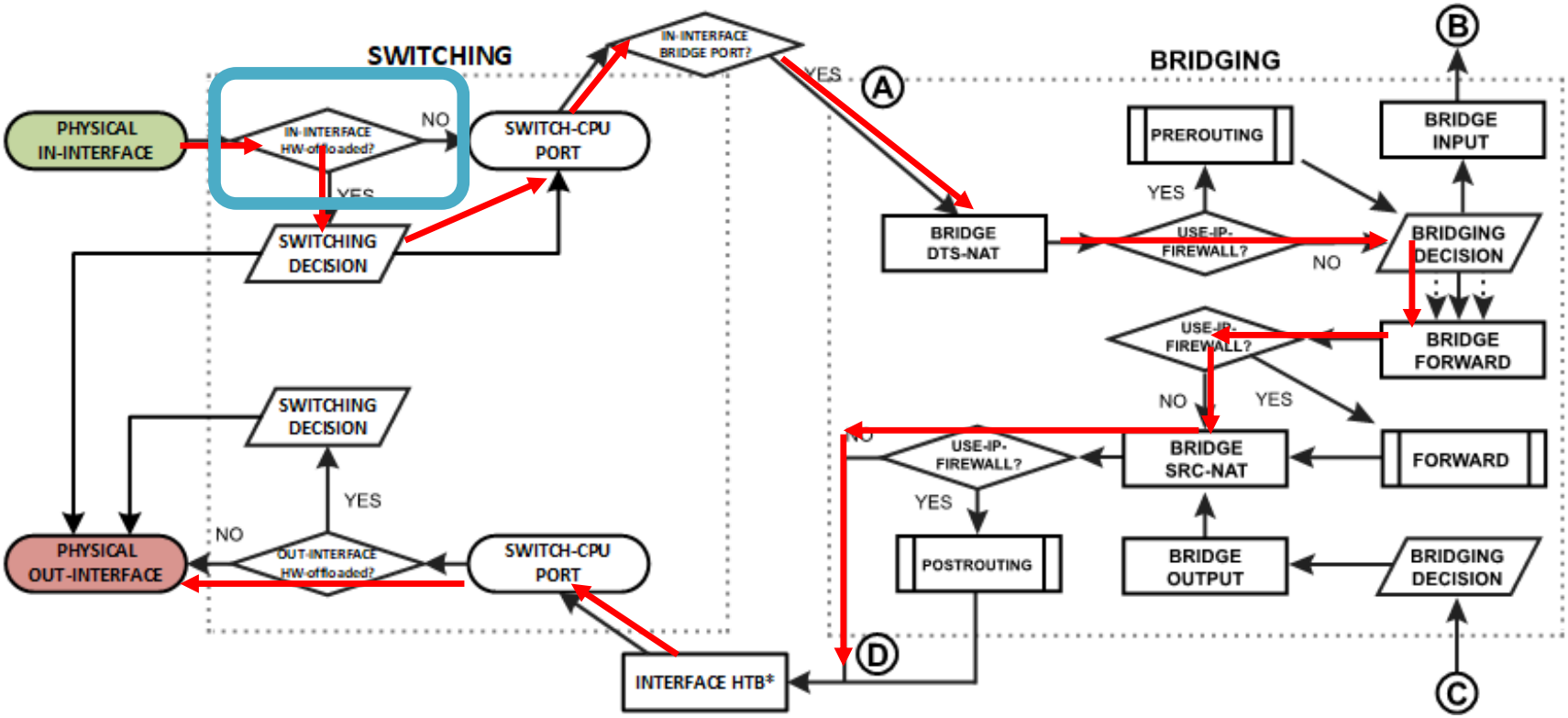

| # | | Interface | Bridge | Horizon | Trusted | Priority (... | Path Cost | PVID | Role |
|---|---|---|---|---|---|---|---|---|---|
| 0 | H | ether1 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 1 | H | ether2 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 2 | | ether3 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 3 | | ether4 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 4 | I | *wlan1* | bridge-lan | | no | 80 | 10 | 1 | disabled port |
| 5 | I | *wlan2* | bridge-lan | | no | 80 | 10 | 1 | disabled port |

# Example 2 – In HW / Out Not HW



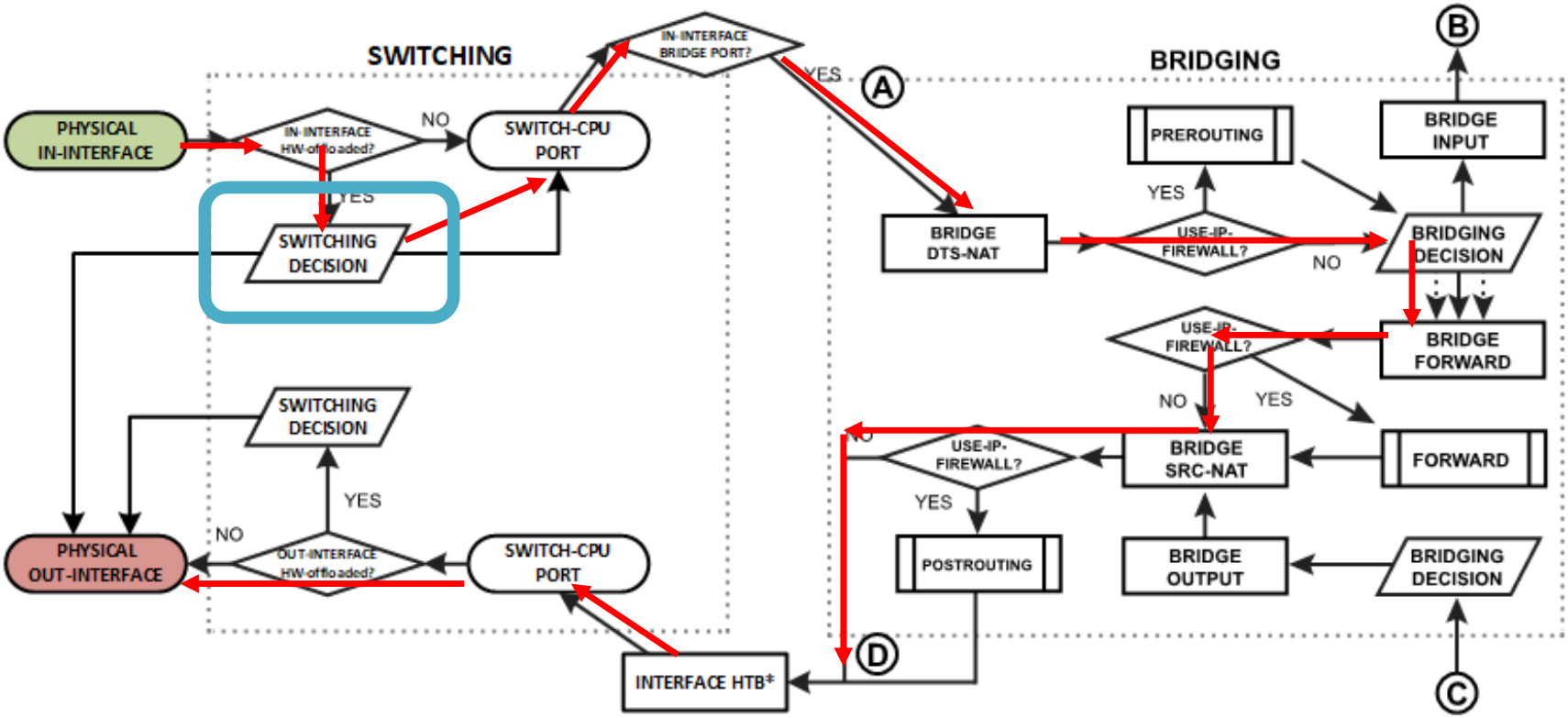

*Interface HTB will not work correctly when the out-interface is hardware offloaded and bridge fast path is not active

# Example 2 – In HW / Out  Not HW



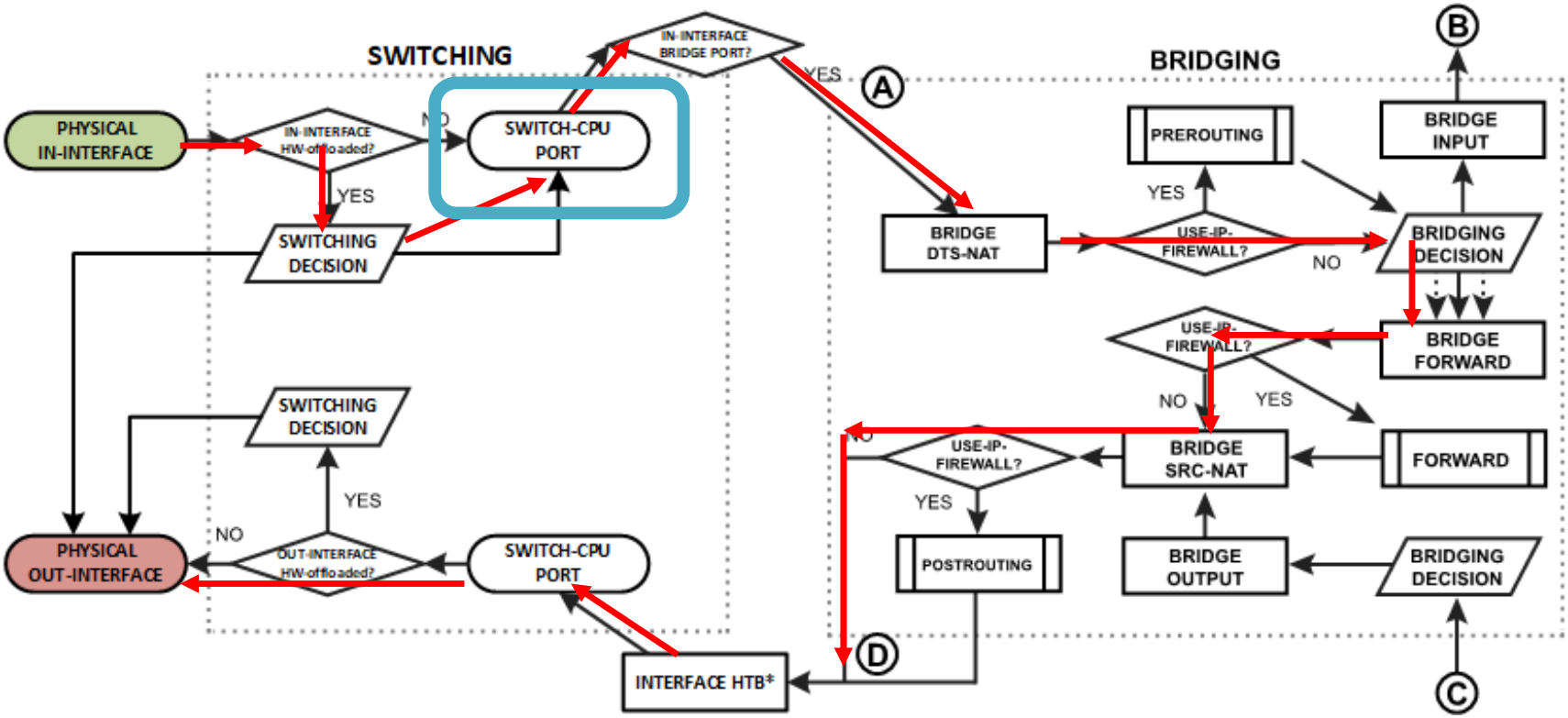

1. Switch checks if in-interface is hw-offloaded interface.

# Example 2 – In HW / Out  Not HW



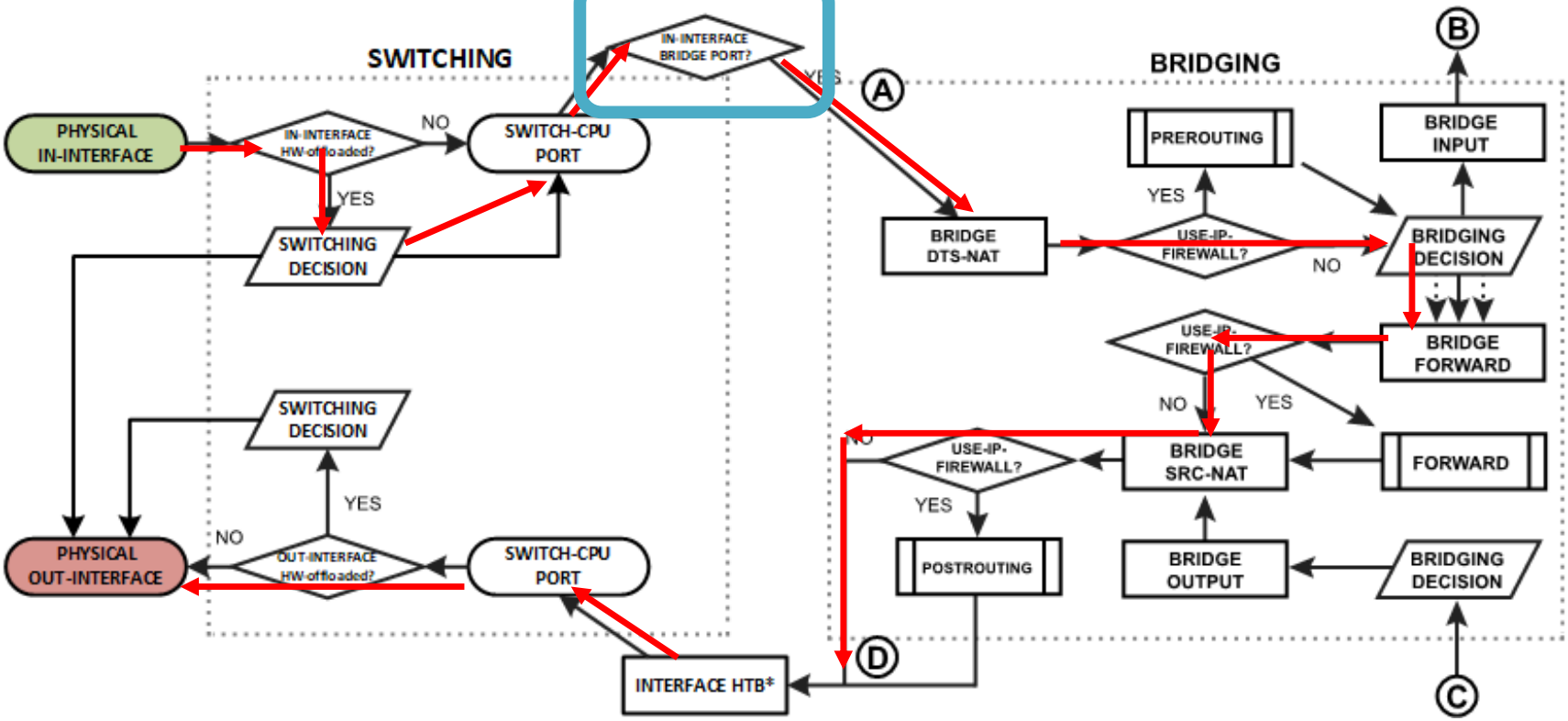

2. Packet dst-mac address is run though the switch host table to make a forwarding decision. The packet gets forwarded to the switch-cpu port.

# Example 2 – In HW / Out  Not HW



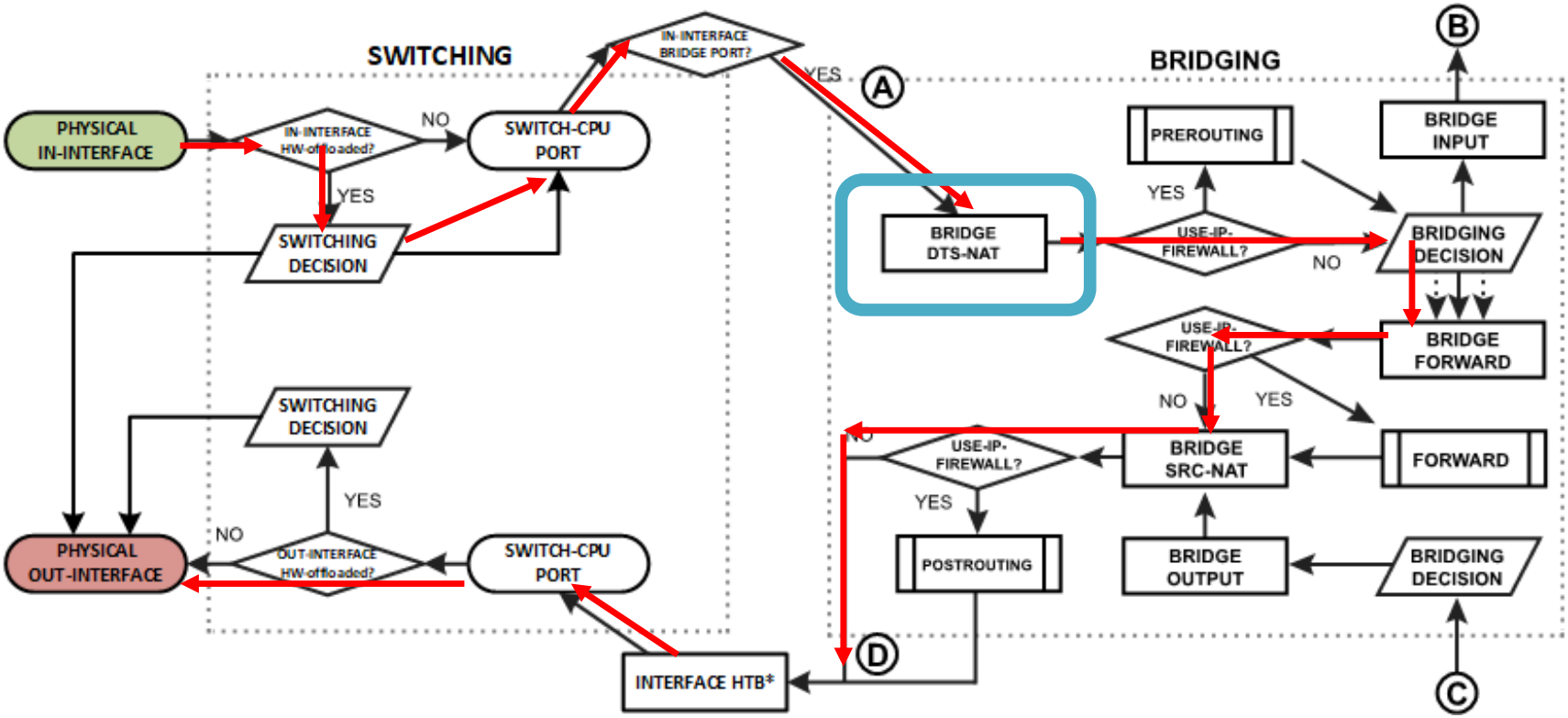

3. The packet exits though the switch-cpu port and it will be further processed by the RouterOS packet flow.

# Example 2 – In HW / Out Not HW



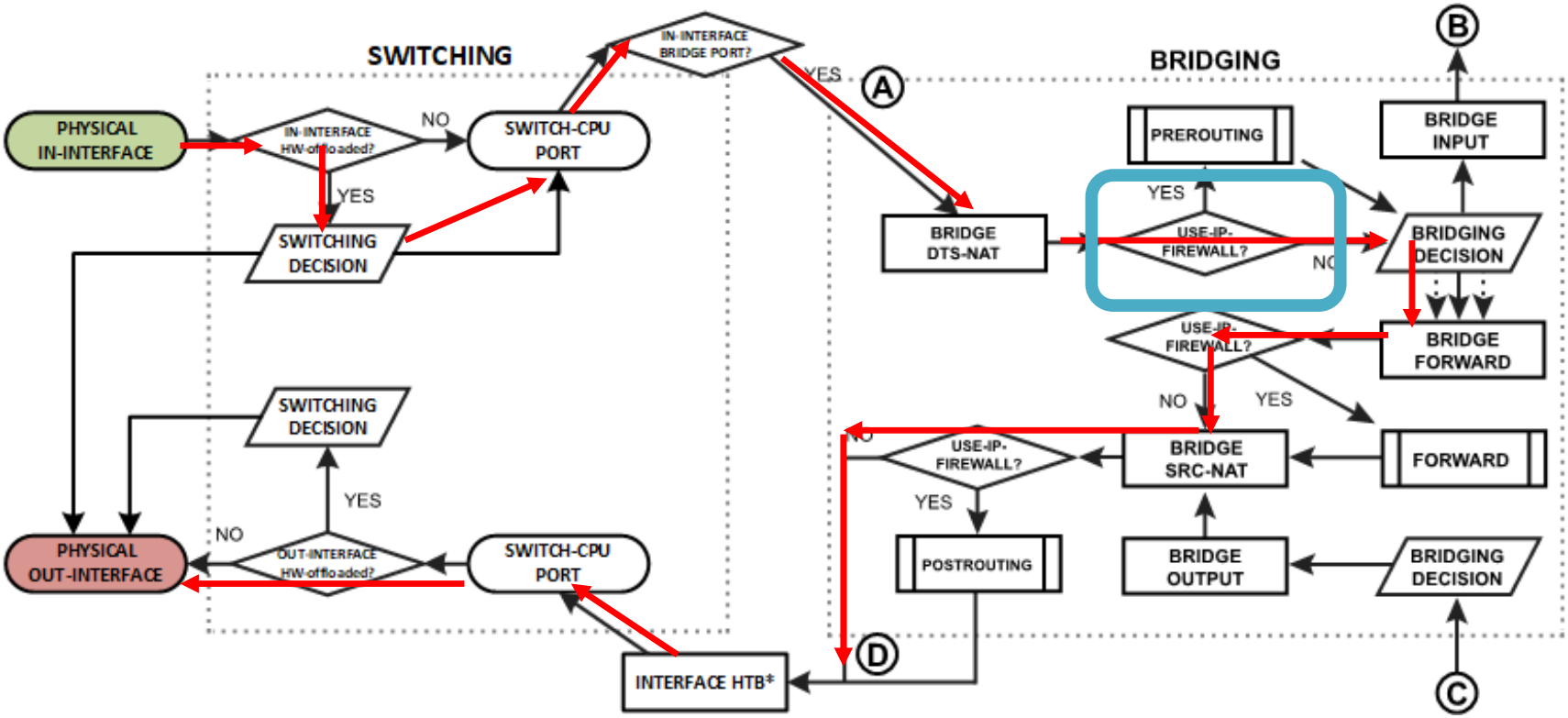

4. The device determines that in-interface is a bridge port, so it gets passed through the bridging process.

# Example 2 – In HW / Out  Not HW



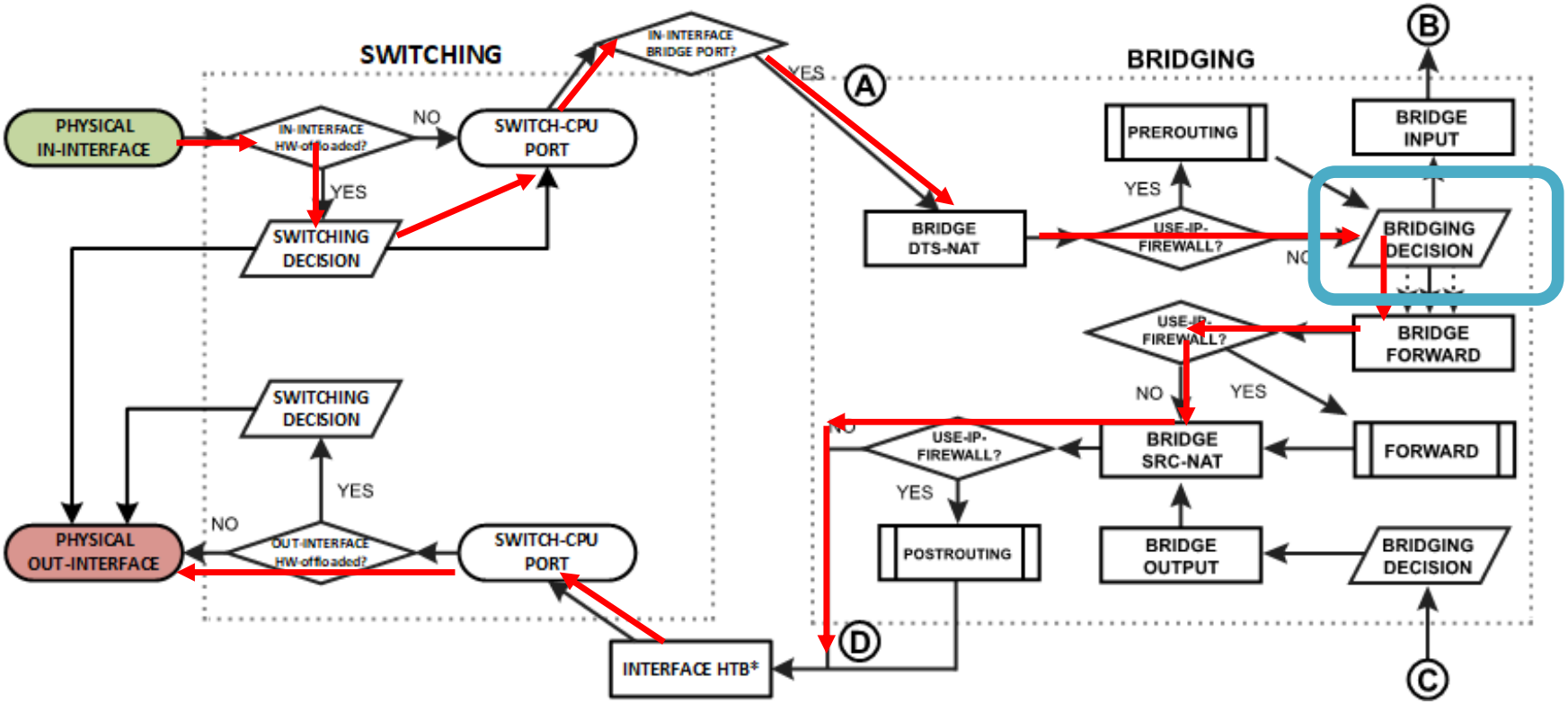

5. The packet goes through the bridge NAT dst-nat chain, where MAC destination and priority can be changed, apart from that, a packet can be simply accepted, dropped, or marked.

# Example 2 – In HW / Out  Not HW



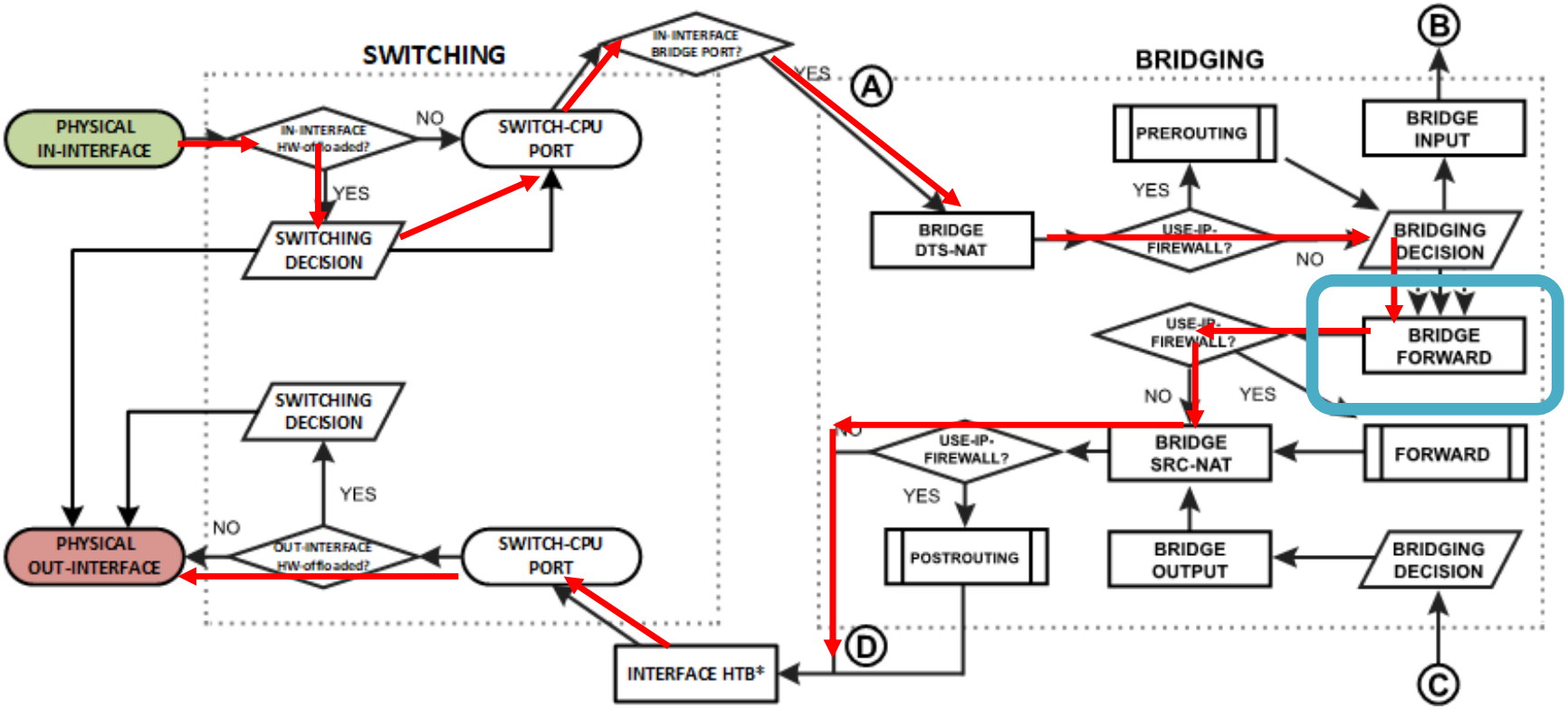

6. Checks whether the use-ip-firewall option is enabled in the bridge settings.

# Example 2 – In HW / Out  Not HW



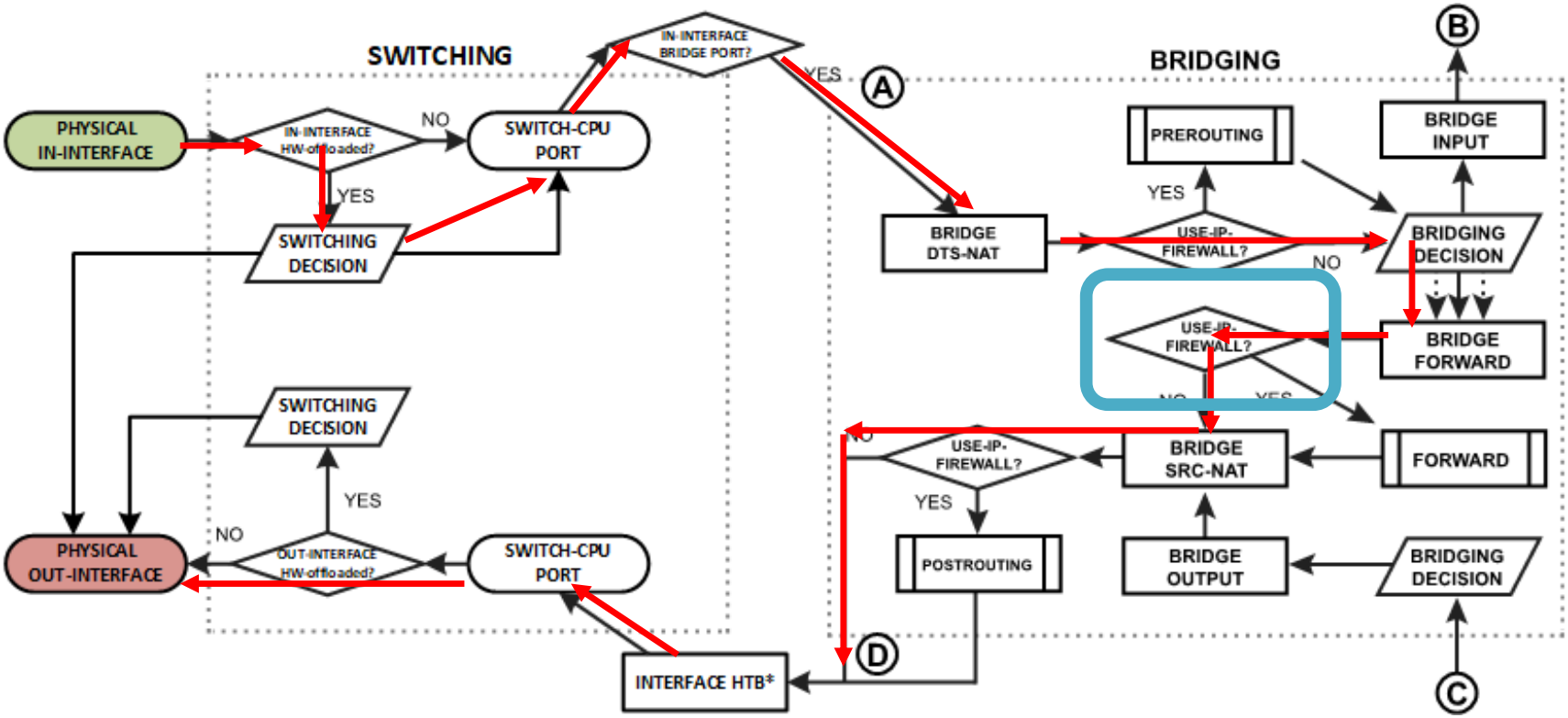

7. Run packet through the bridge host table to make a forwarding decision. A packet that ends up being flooded (e.g. broadcast, multicast, unknown unicast traffic), gets multiplied per bridge port and then processed in the bridge forward chain.

# Example 2 – In HW / Out  Not HW



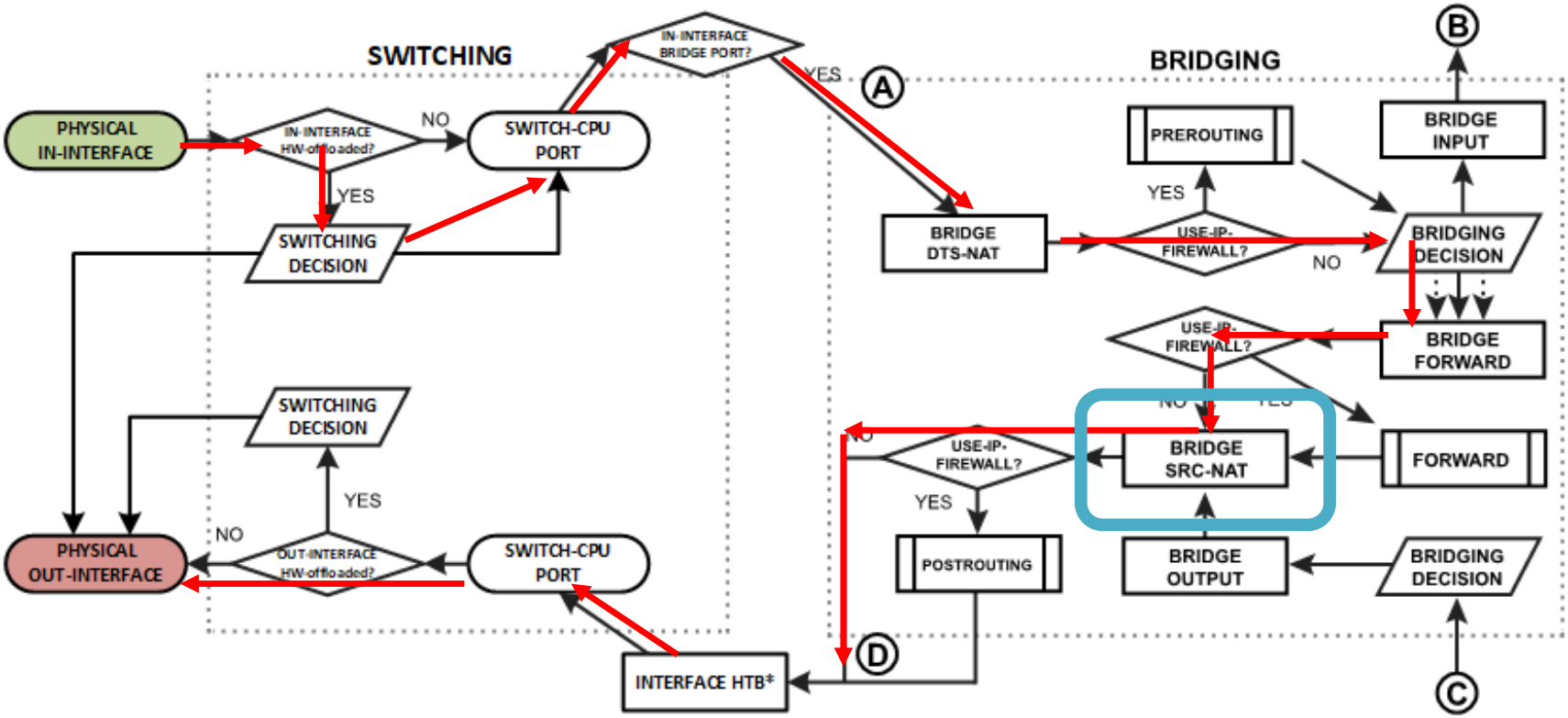

8. Packet goes through the bridge filter forward chain, where priority can be changed or packet can be simply accepted, dropped, or marked.

# Example 2 – In HW / Out  Not HW



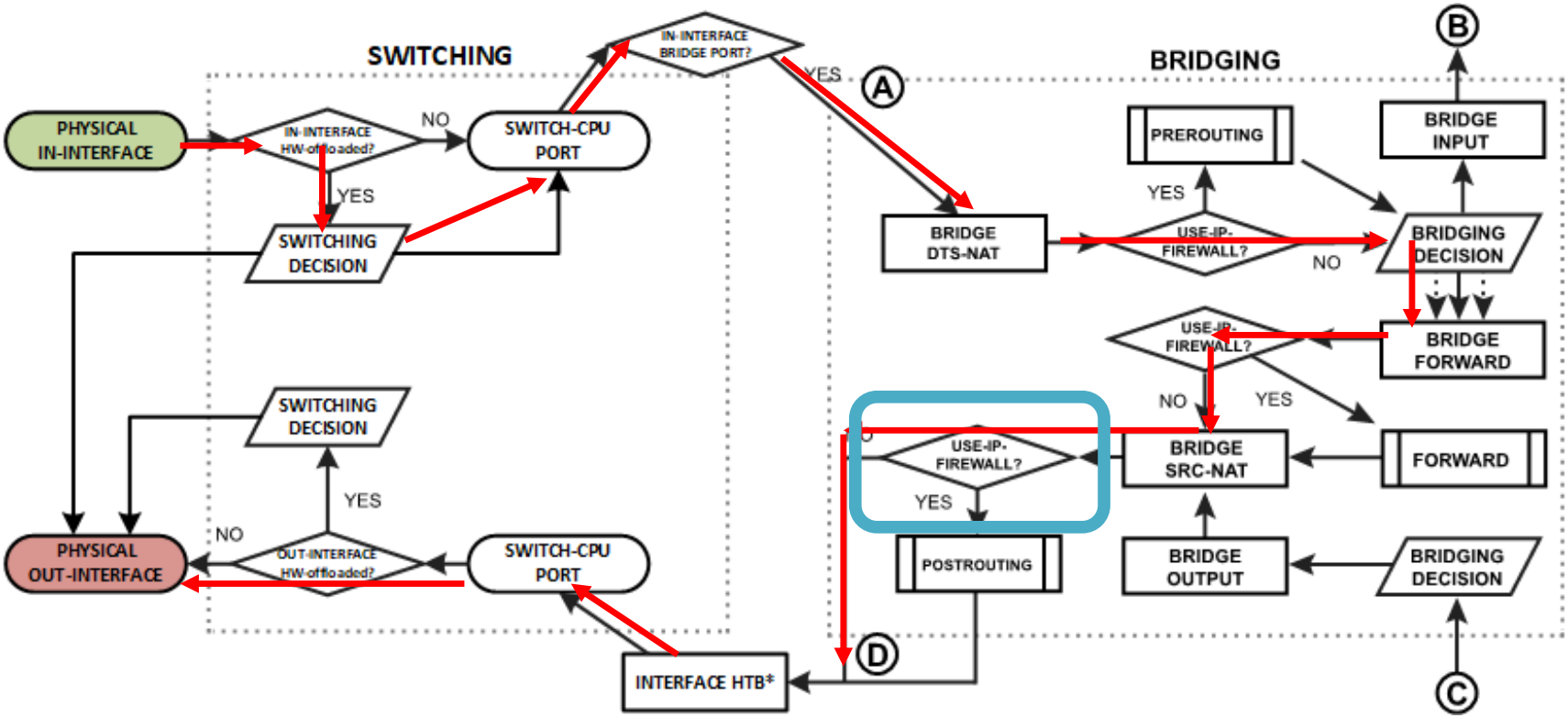

9. Checks whether the use-ip-firewall option is enabled in the bridge settings.

# Example 2 – In HW / Out  Not HW



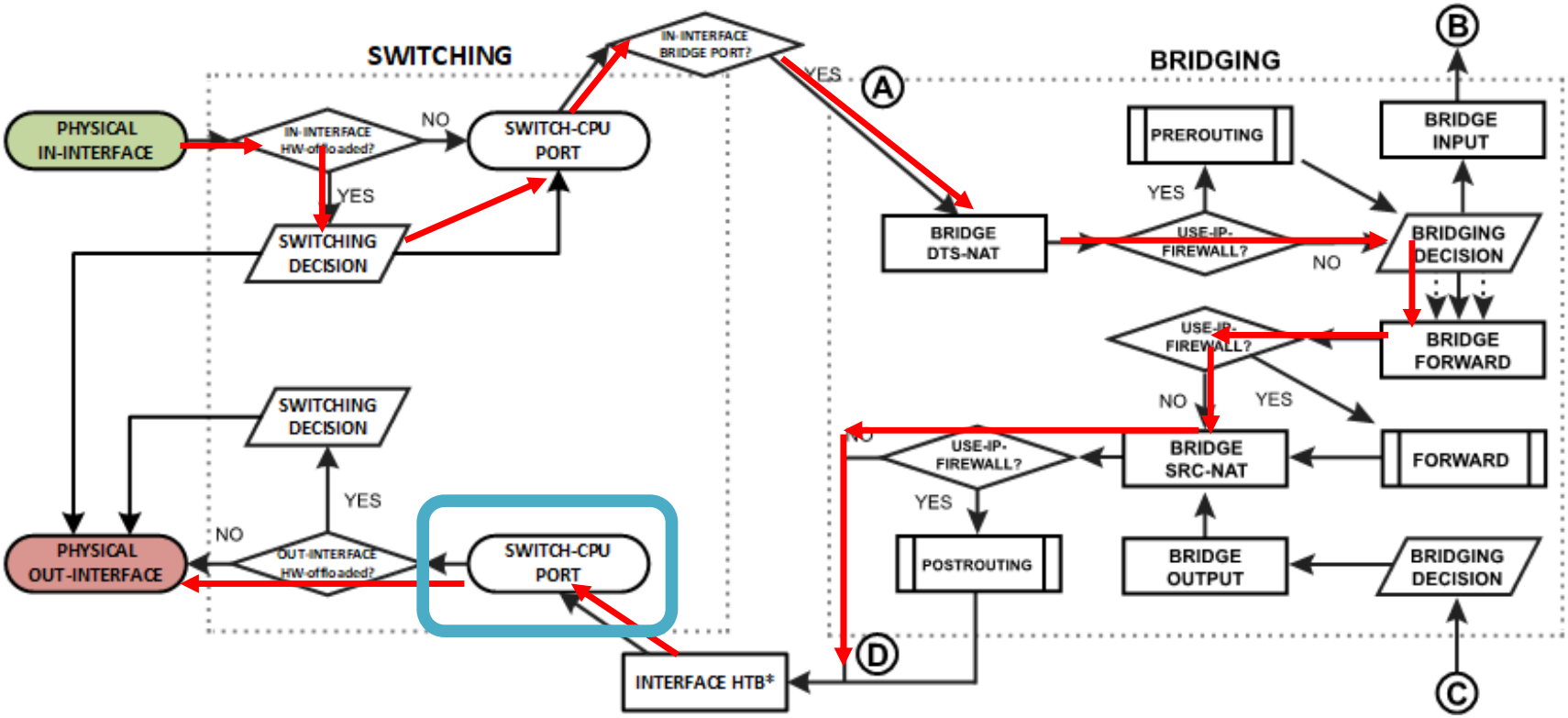

10. Packet goes through the bridge NAT src-nat chain, where MAC source and priority can be changed, apart from that, a packet can be simply accepted, dropped, or marked.

# Example 2 – In HW / Out  Not HW



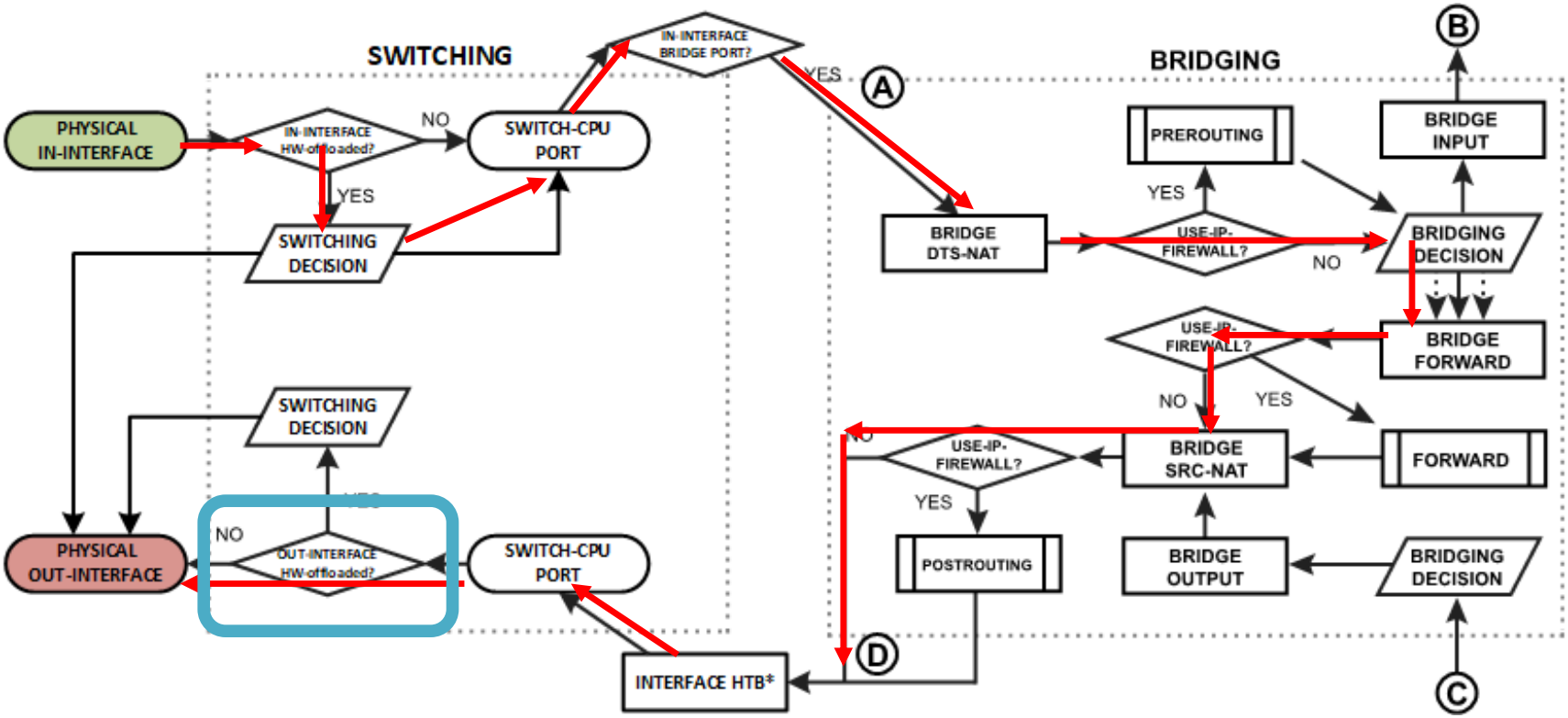

11. Checks whether the use-ip-firewall option is enabled in the bridge settings. The packet now leaves the bridge process.

# Example 2 – In HW / Out  Not HW




12. The packet that exits the RouterOS software processing is received on the switch-cpu port.

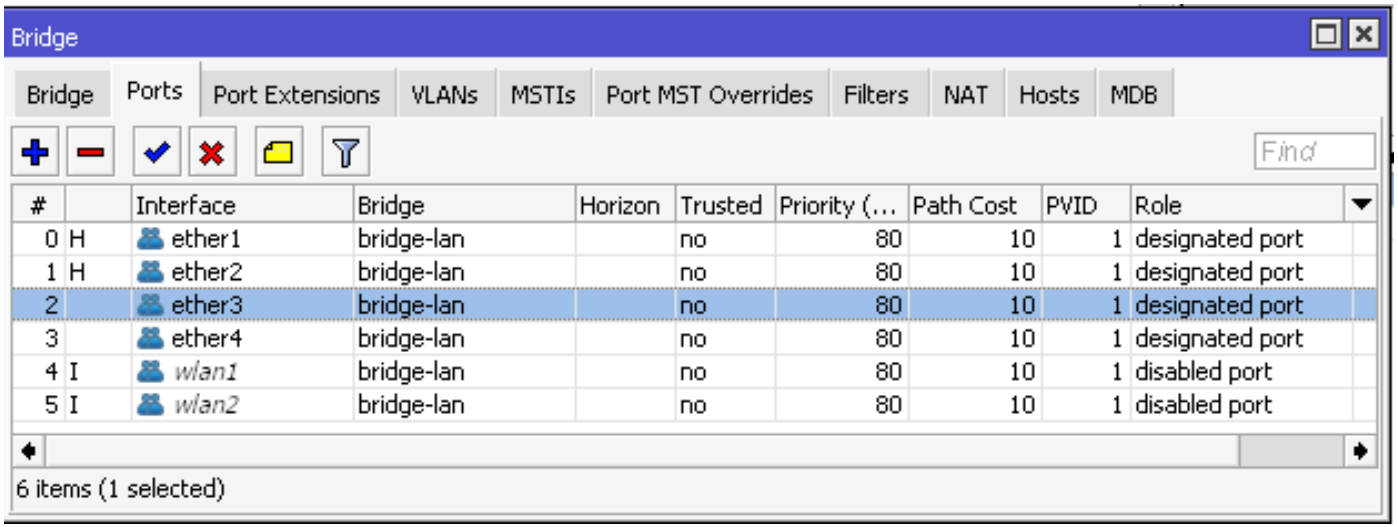
# Example 2 – In HW / Out  Not HW




13. The switch checks whether the out-interface is a hardware offloaded interface and the packet now leaves on the physical interface.

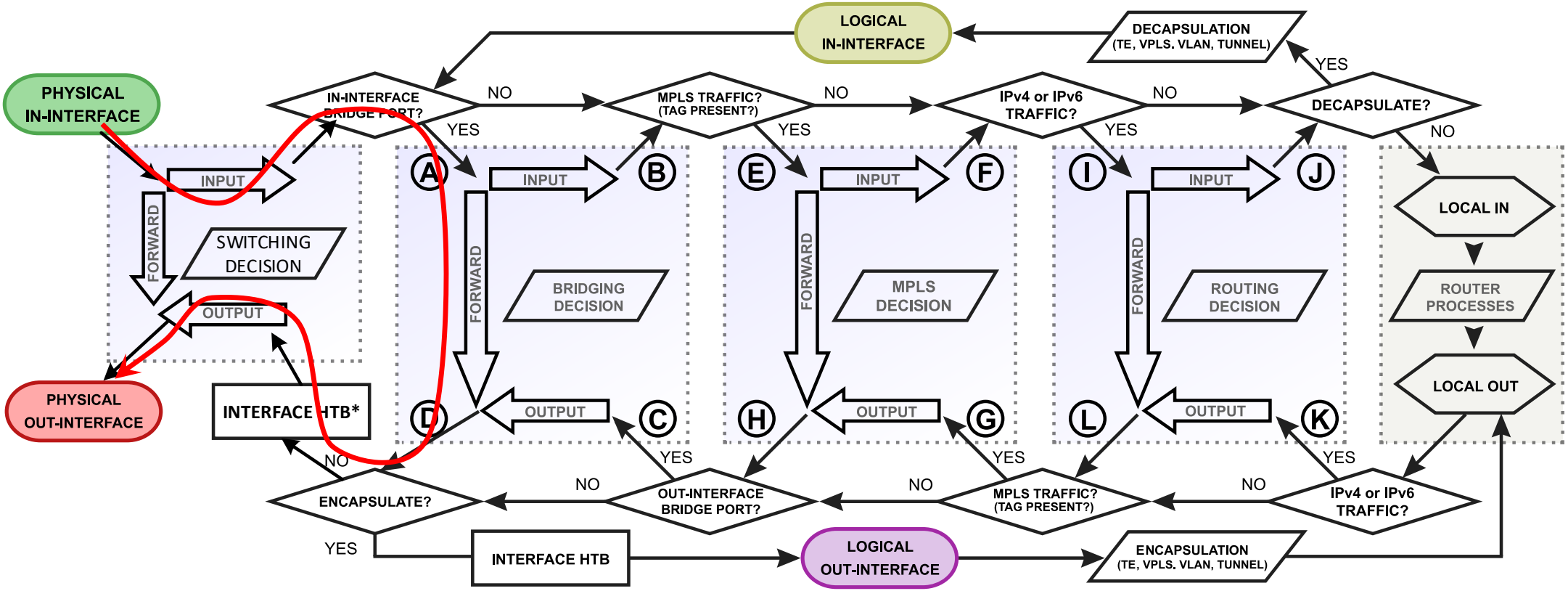
# Example 3 – In Not HW-Offloaded / Out HW Offloaded

- Traffic Flow between ether3 (PC3) and ether2 (PC2)
- In-interface HW offloaded, Out-Interface not hw-offloaded



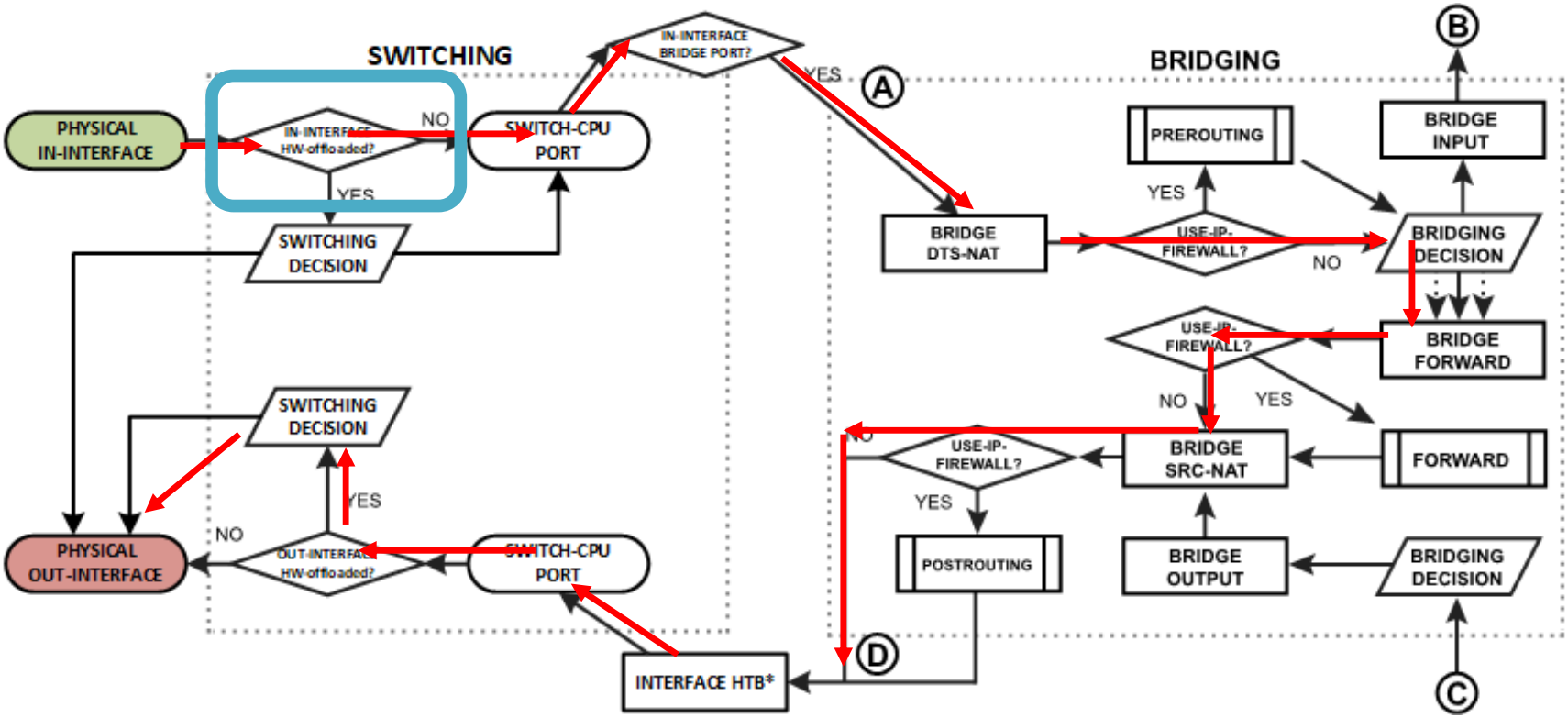

| # | | Interface | Bridge | Horizon | Trusted | Priority (... | Path Cost | PVID | Role |
|---|---|---|---|---|---|---|---|---|---|
| 0 | H | ether1 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 1 | H | ether2 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 2 | | ether3 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 3 | | ether4 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 4 | I | *wlan1* | bridge-lan | | no | 80 | 10 | 1 | disabled port |
| 5 | I | *wlan2* | bridge-lan | | no | 80 | 10 | 1 | disabled port |

# Example 3 – In Not HW / Out HW



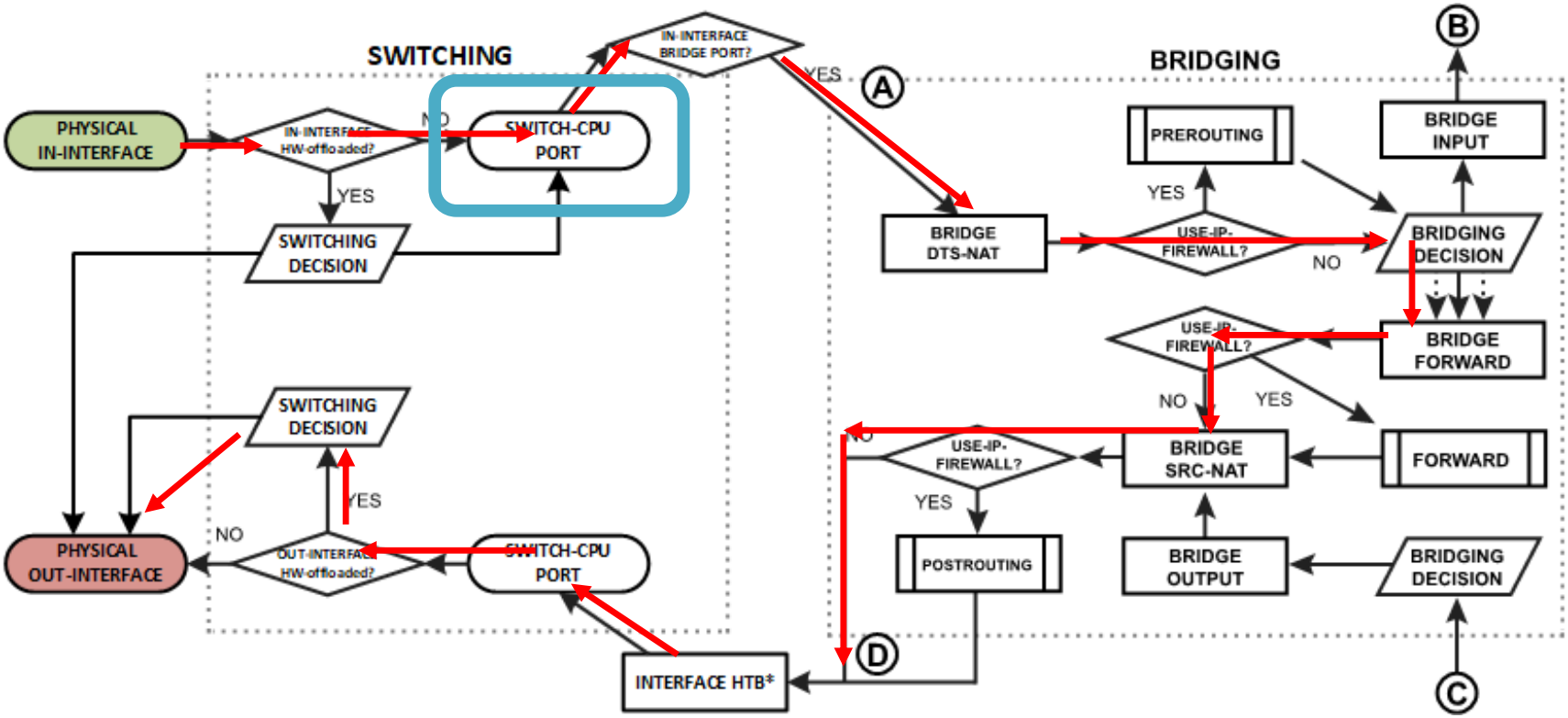

*Interface HTB will not work correctly when the out-interface is hardware offloaded and bridge fast path is not active

# Example 3 – In Not HW / Out HW



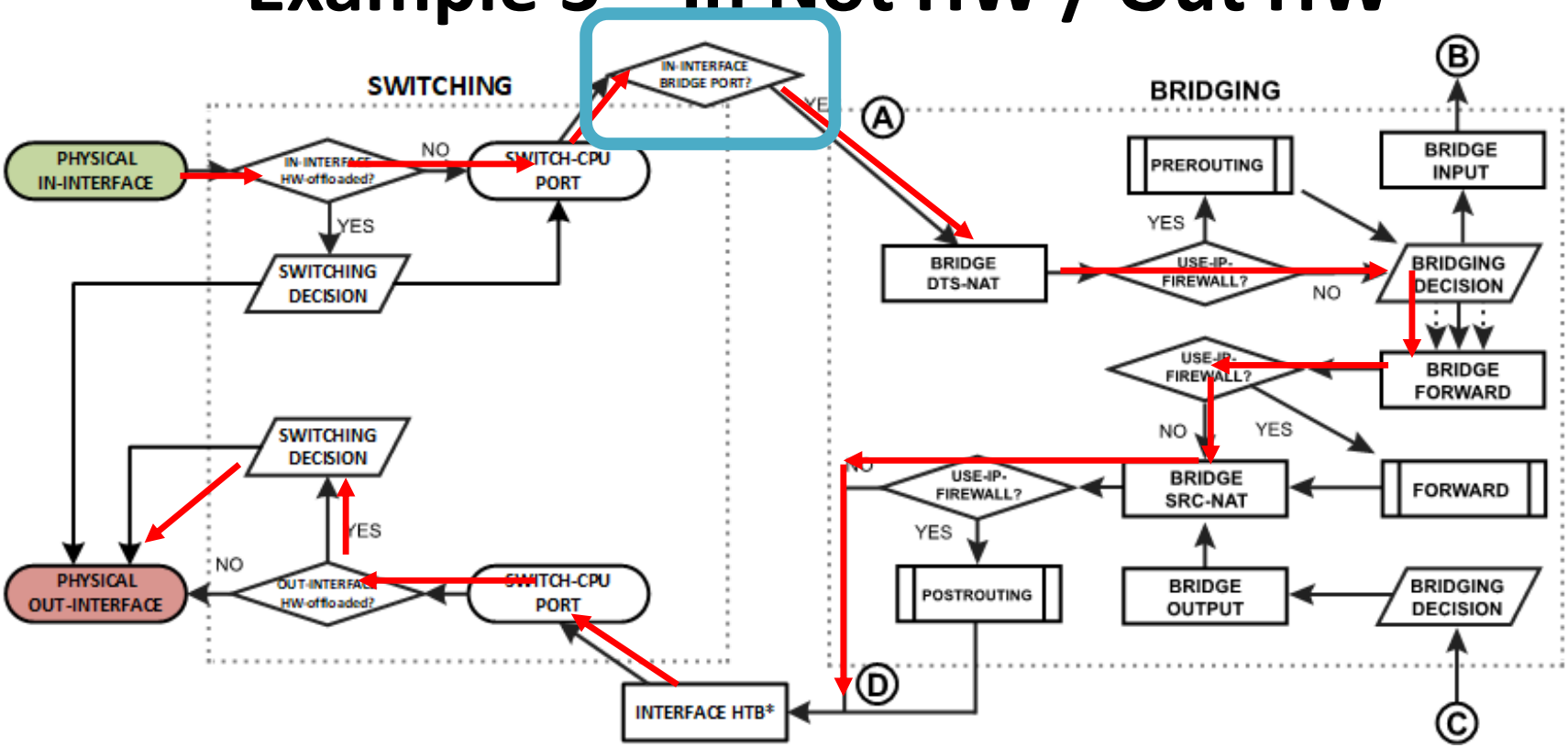

1. The switch checks whether the in-interface is a hardware offloaded interface; The packet is forward to the switch-cpu port.

# Example 3 – In Not HW / Out HW



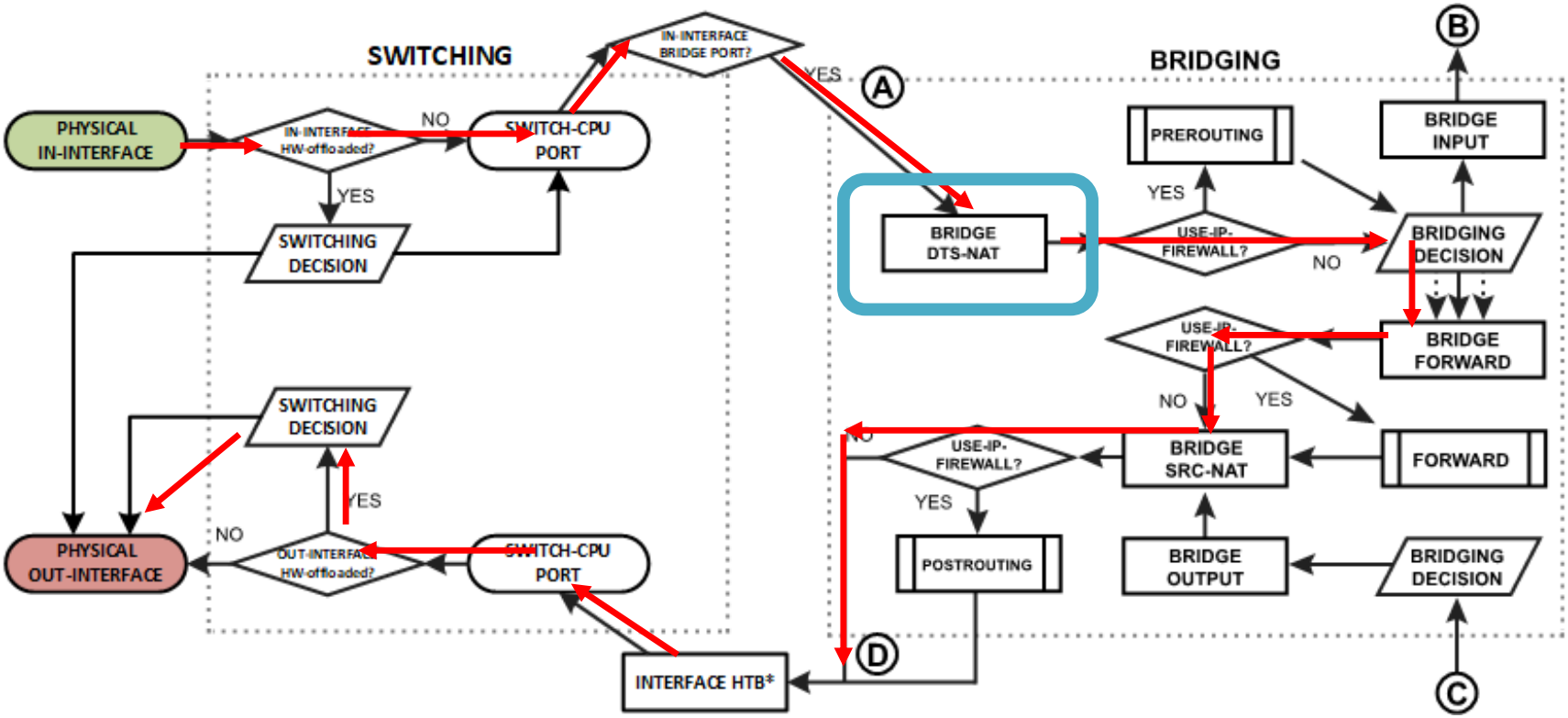

2. The packet exits though the switch CPU Port and it will be further processed by the RouterOS packet flow.

# Example 3 – In Not HW / Out HW



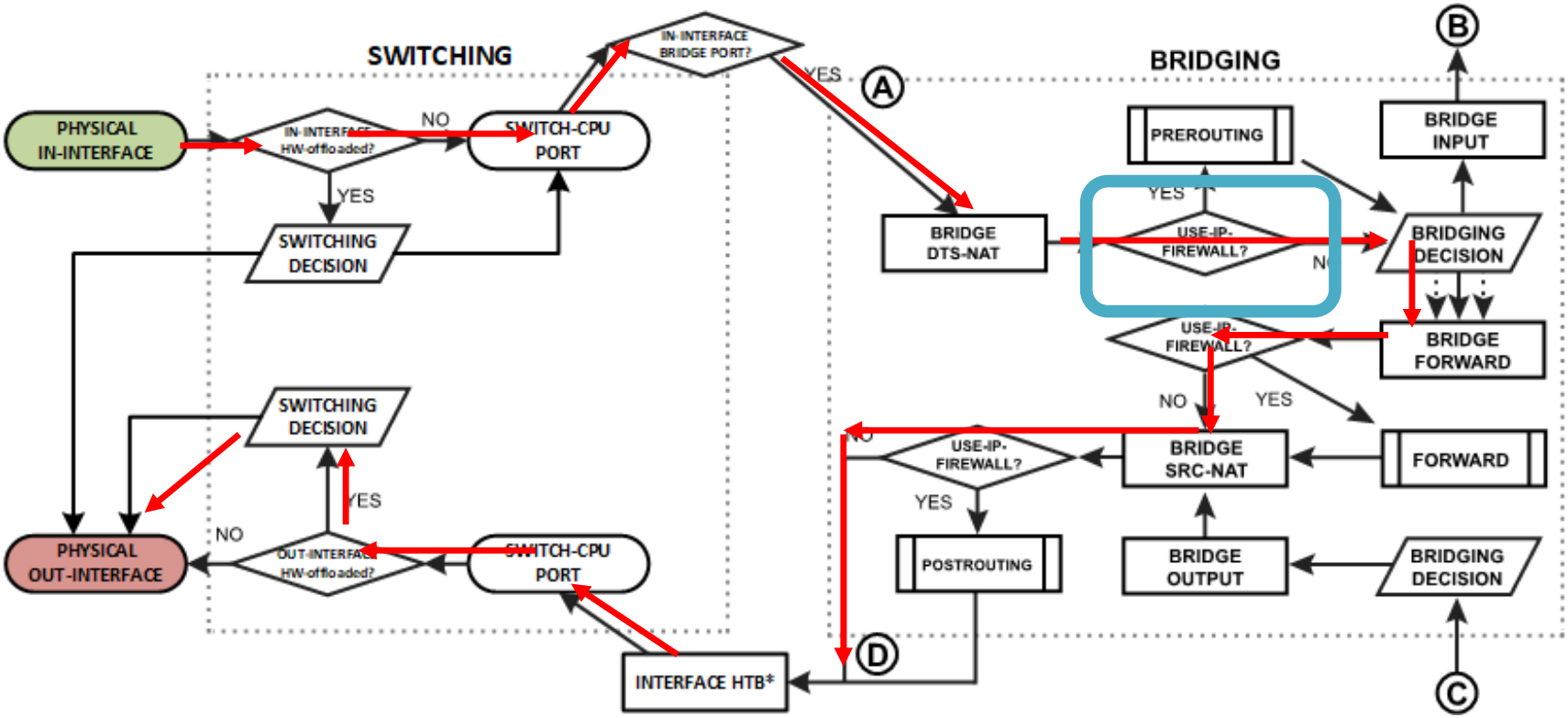

3. The device determines that in-interface is a bridge port, so it gets passed through the bridging process.

# Example 3 – In Not HW / Out HW



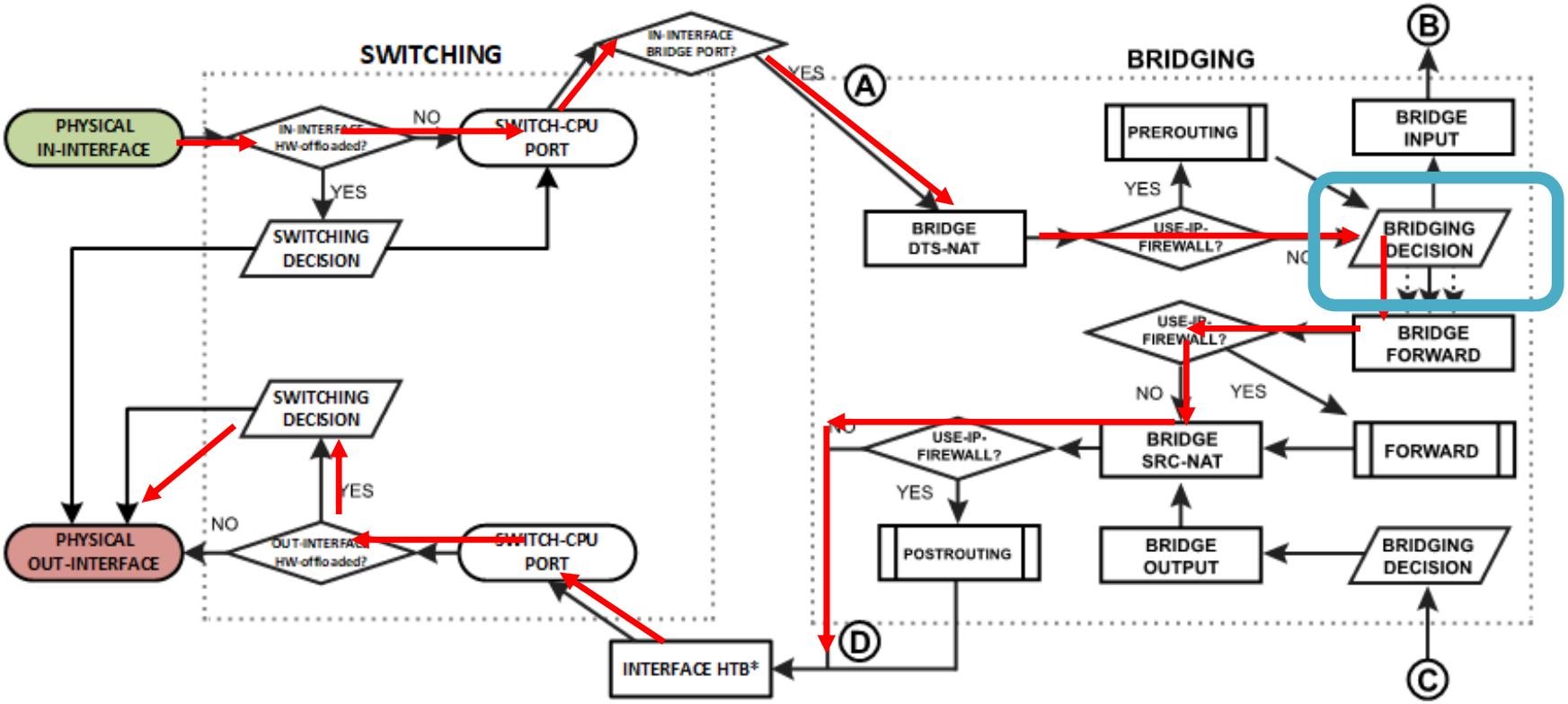

4. The packet goes through the bridge NAT dst-nat chain, where MAC destination and priority can be changed, apart from that, a packet can be simply accepted, dropped, or marked.

# Example 3 – In Not HW / Out HW



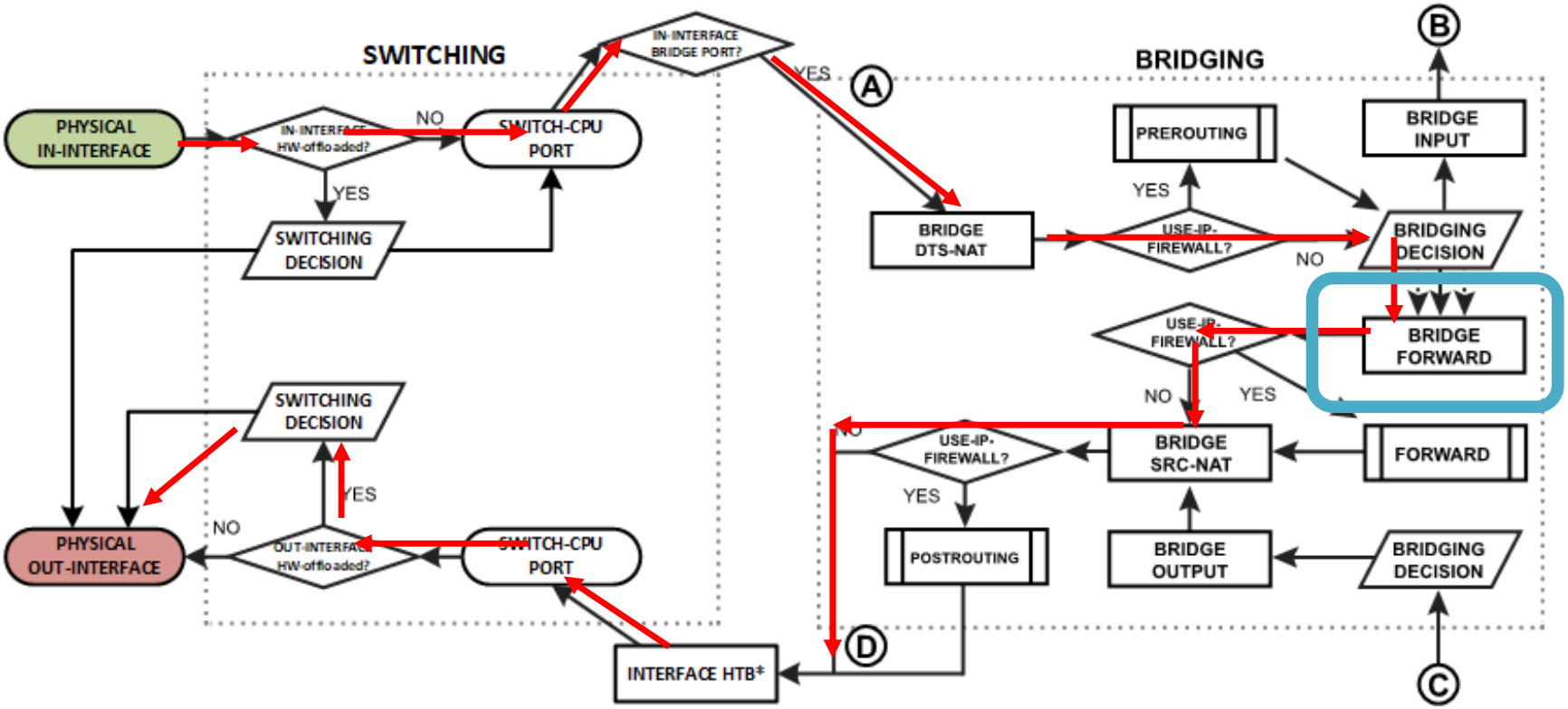

5. Checks whether the use-ip-firewall option is enabled in the bridge settings.

# Example 3 – In Not HW / Out HW



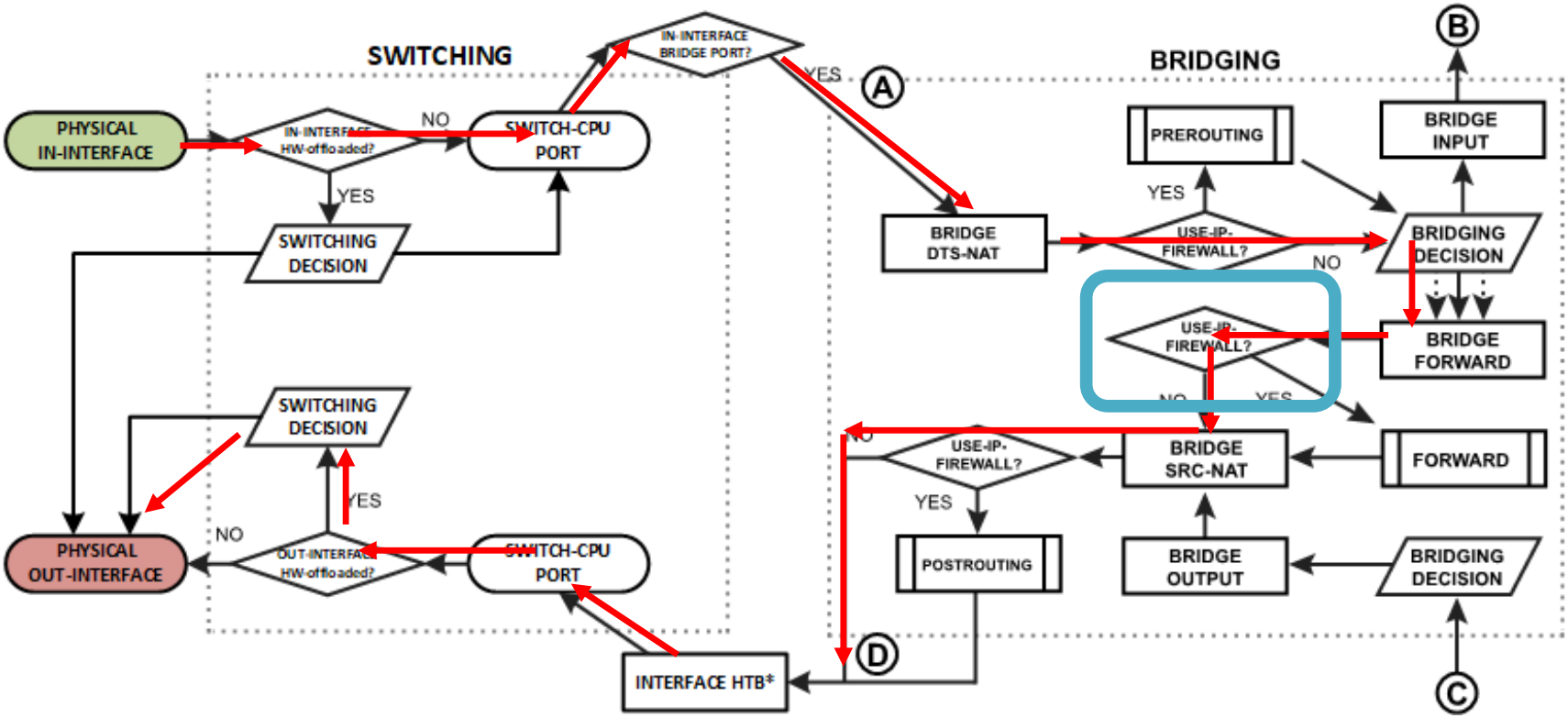

6. Run packet through the bridge host table to make a forwarding decision. A packet that ends up being flooded (e.g. broadcast, multicast, unknown unicast traffic), gets sent out of all bridge port and then processed in the bridge forward chain.

# Example 3 – In Not HW / Out HW



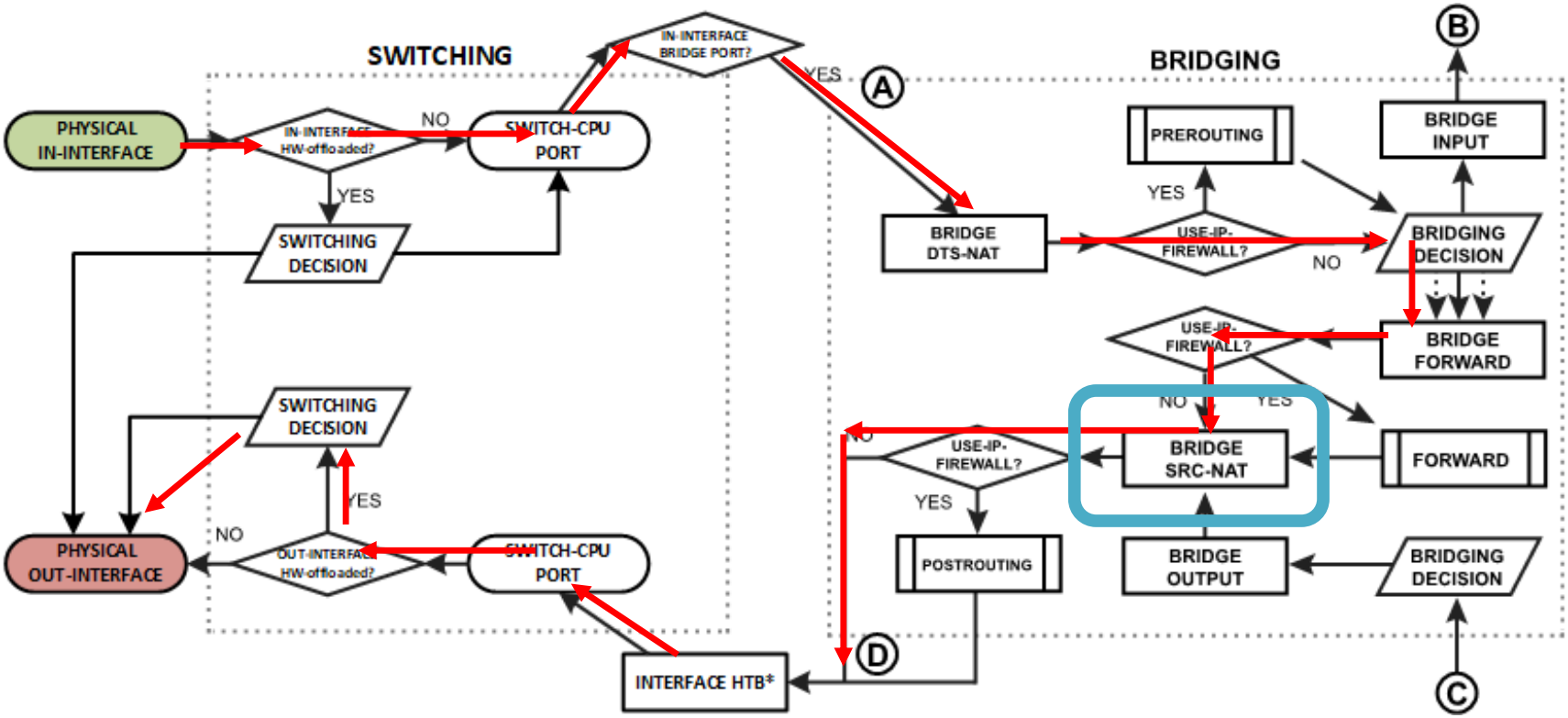

7. The packet goes through the bridge filter forward chain, where priority can be changed or packet can be simply accepted, dropped, or marked.

# Example 3 – In Not HW / Out HW



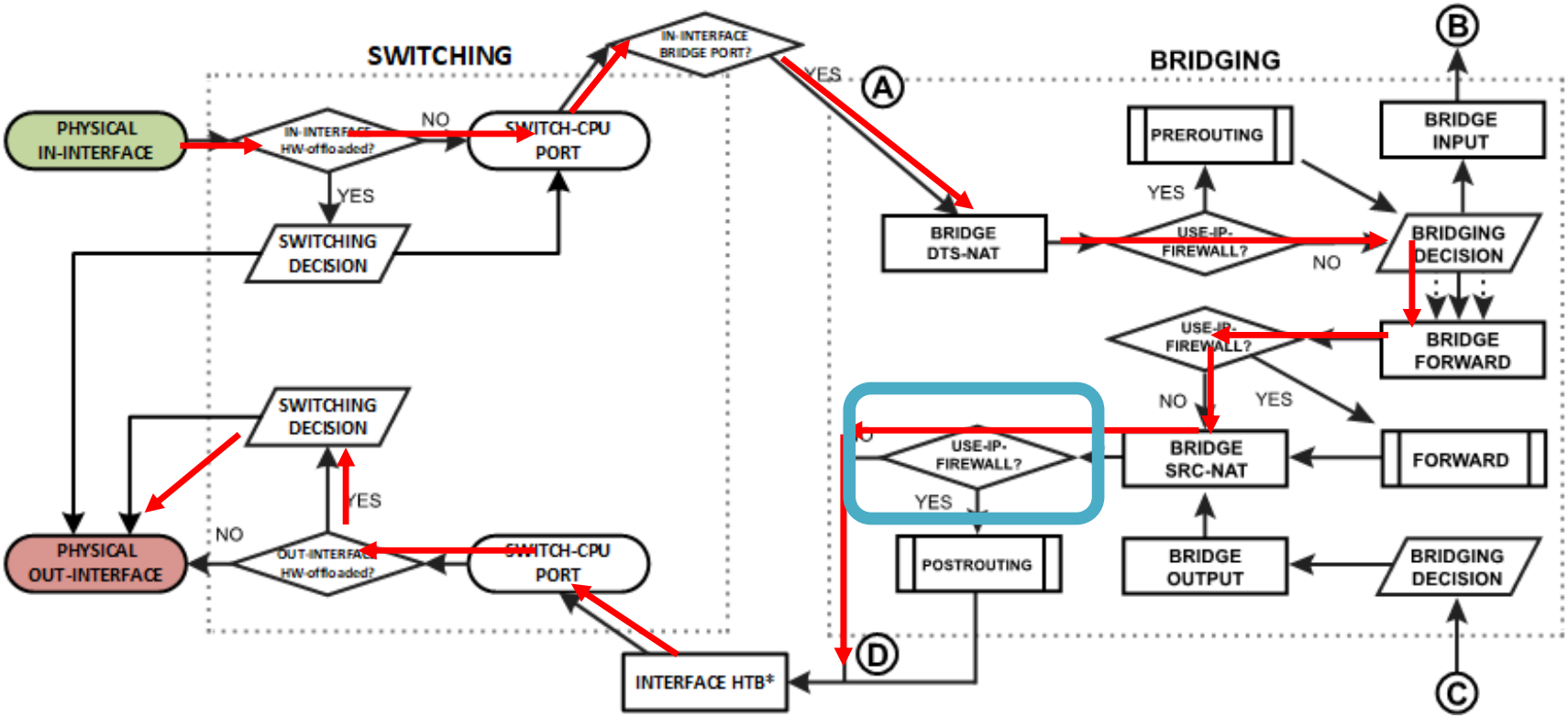

8. Checks whether the use-ip-firewall option is enabled in the bridge settings.

# Example 3 – In Not HW / Out HW



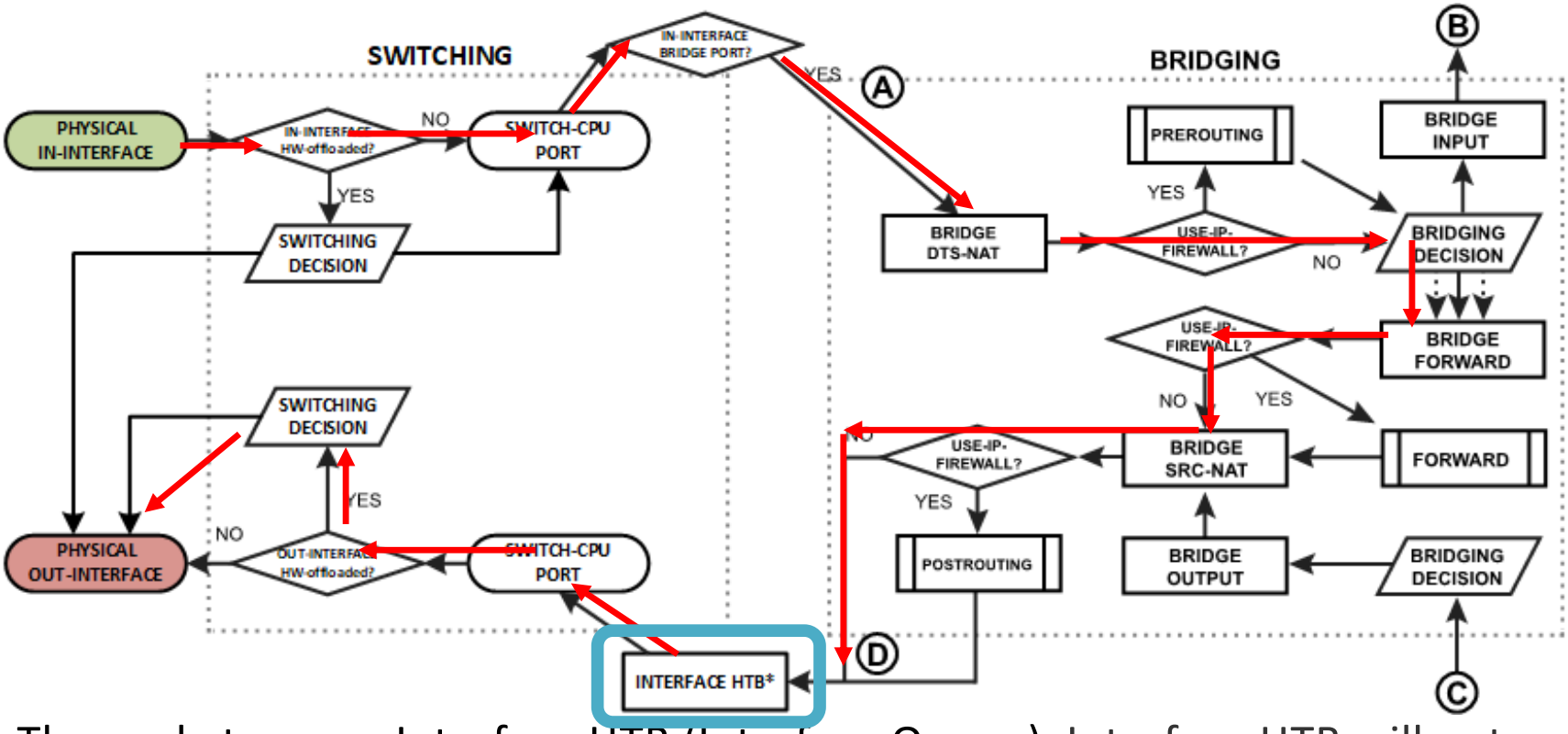

9. The packet goes through the bridge NAT src-nat chain, where MAC source and priority can be changed, apart from that, a packet can be simply accepted, dropped, or marked.

# Example 3 – In Not HW / Out HW



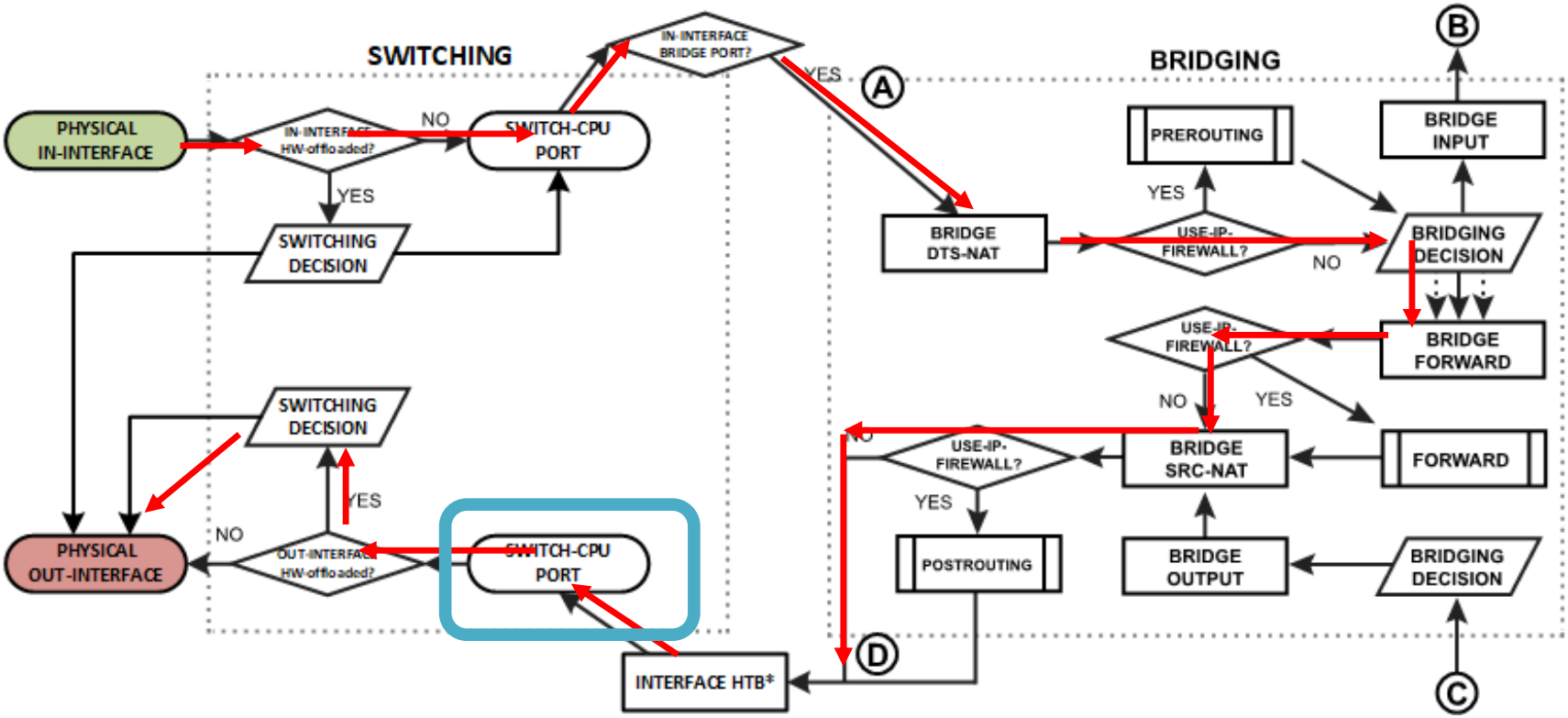

10. Checks whether the use-ip-firewall option is enabled in the bridge settings and the packet now leaves the bridge process.

# Example 3 – In Not HW / Out HW



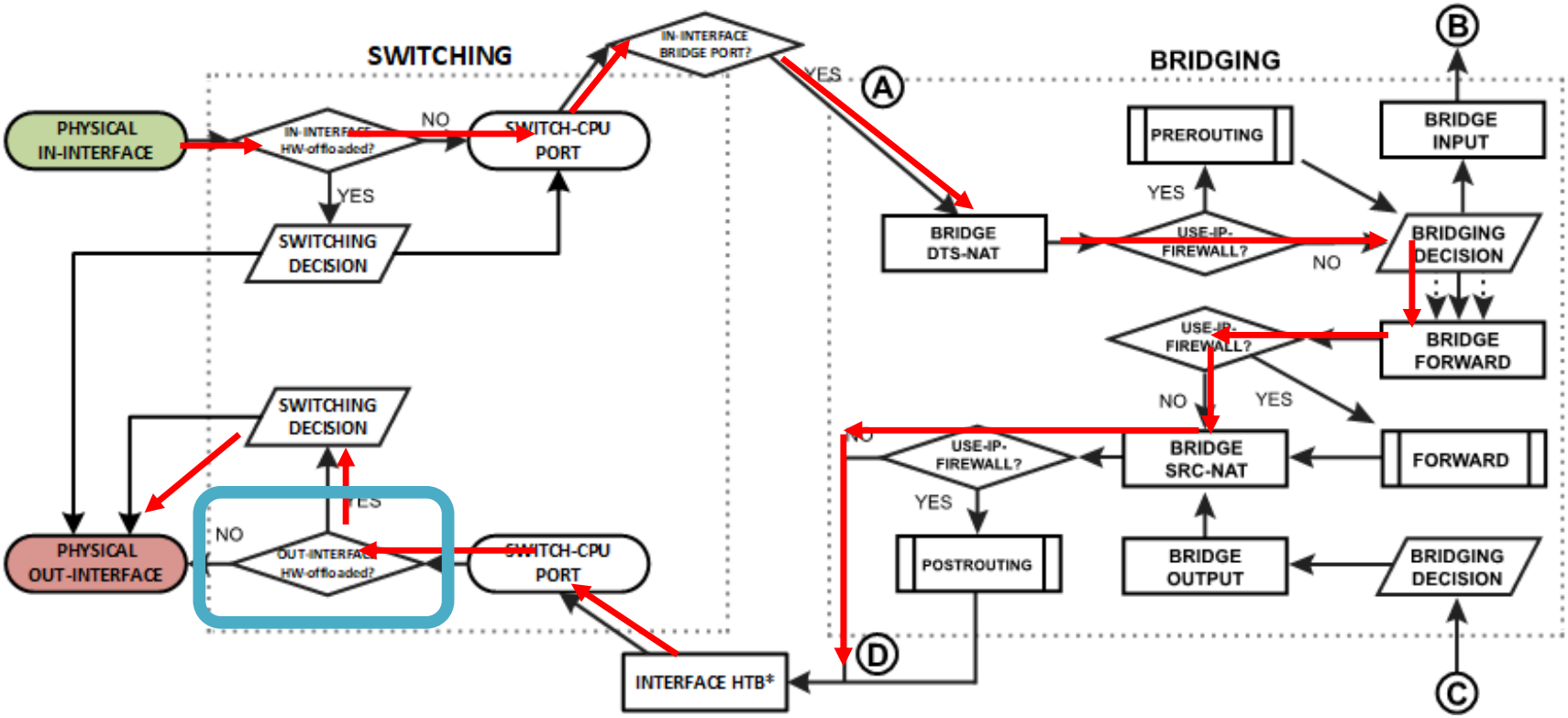

11. The packet passes Interface HTB (Interface Queue). Interface HTB will not work correctly when the out-interface is hardware offloaded and the bridge Fast Path is not active.

# Example 3 – In Not HW / Out HW



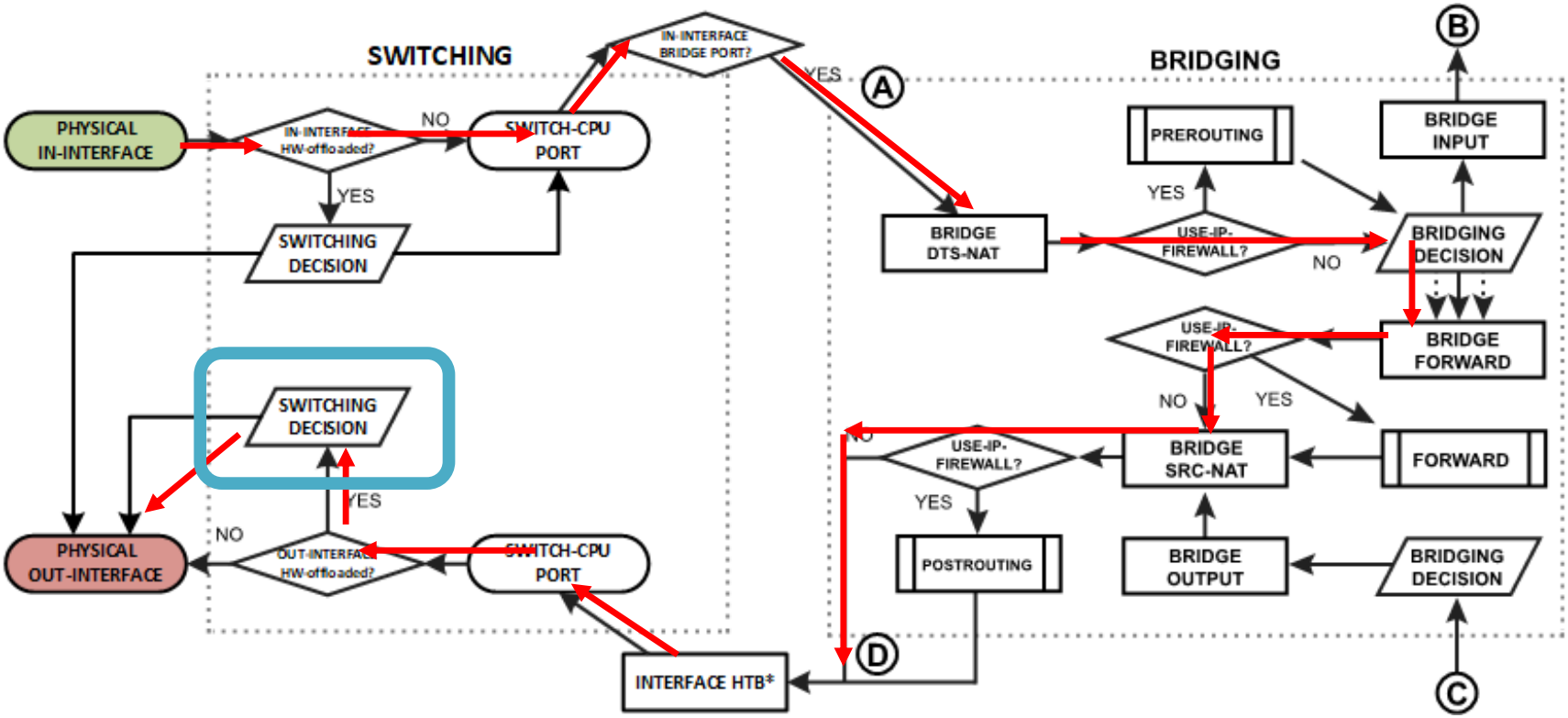

12. The packet that exits the RouterOS software processing is received on the switch-cpu port.

# Example 3 – In Not HW / Out HW




13. The switch checks whether the out-interface is a hardware offloaded interface.

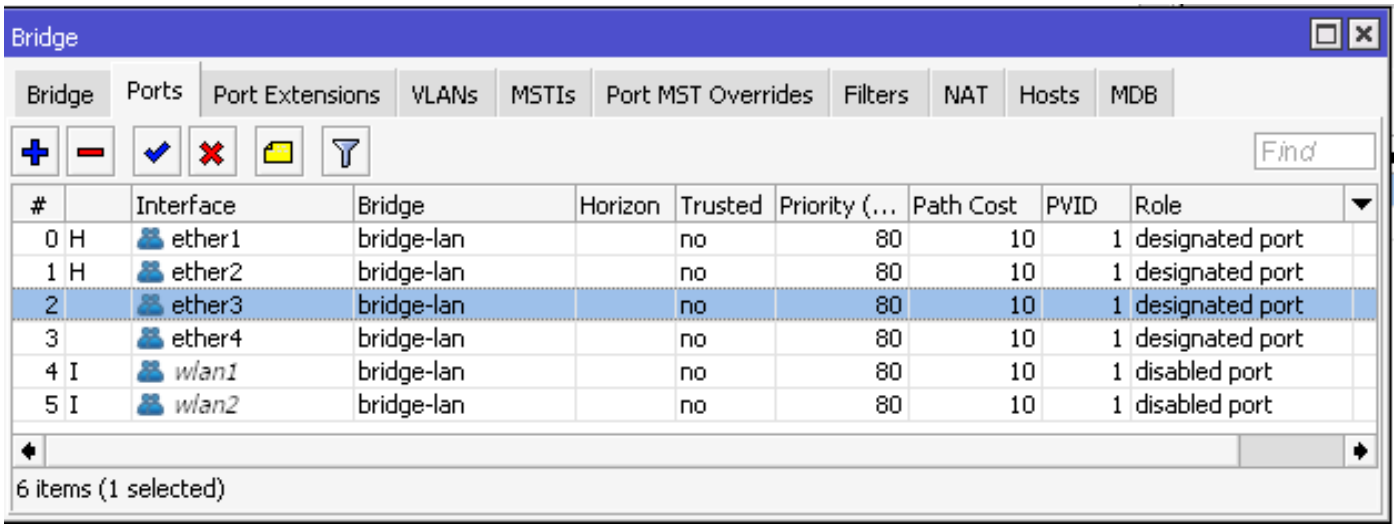
# Example 3 – In Not HW / Out HW




14. The packet passes through the switch host table to make a forwarding decision. If the switch finds a match for the destination MAC address. The packet is now sent out through the physical interface.

# Example 4 – In Not HW-Offloaded / Out Not HW-Offloaded

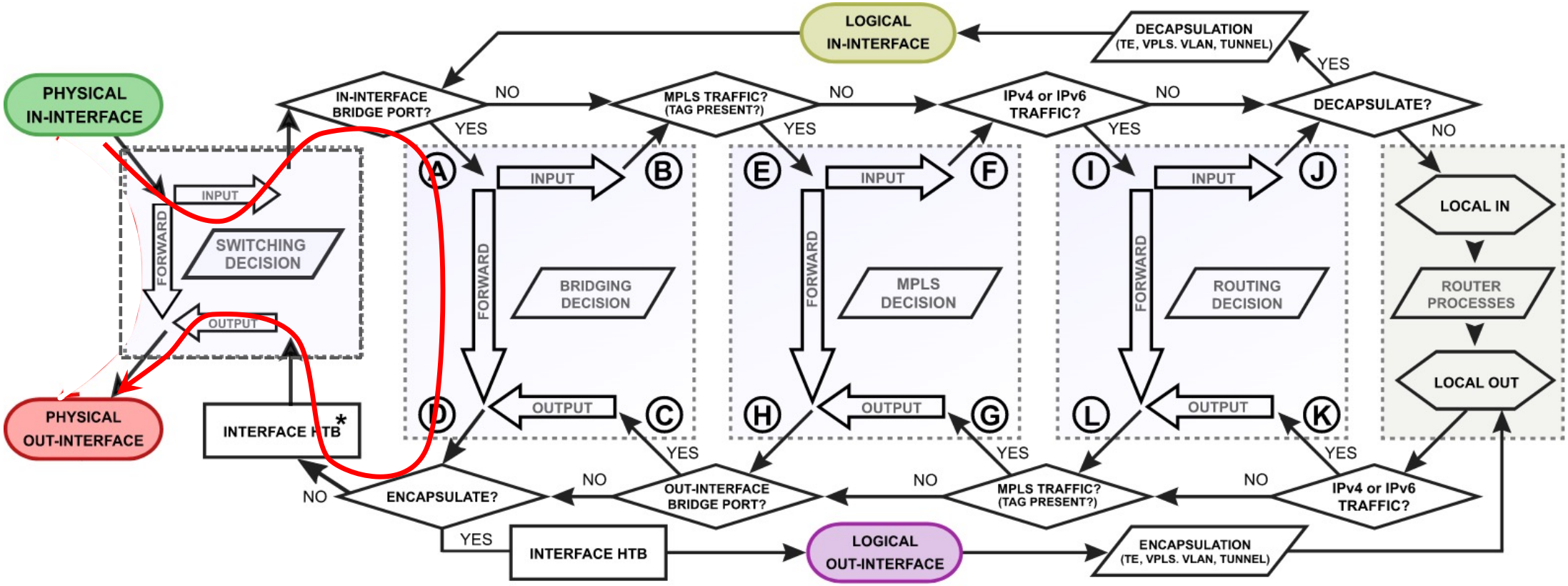
- Traffic flow from ether3 (PC3) to ether4 (PC4)
- In-interface HW offloaded, Out-Interface not hw-offloaded



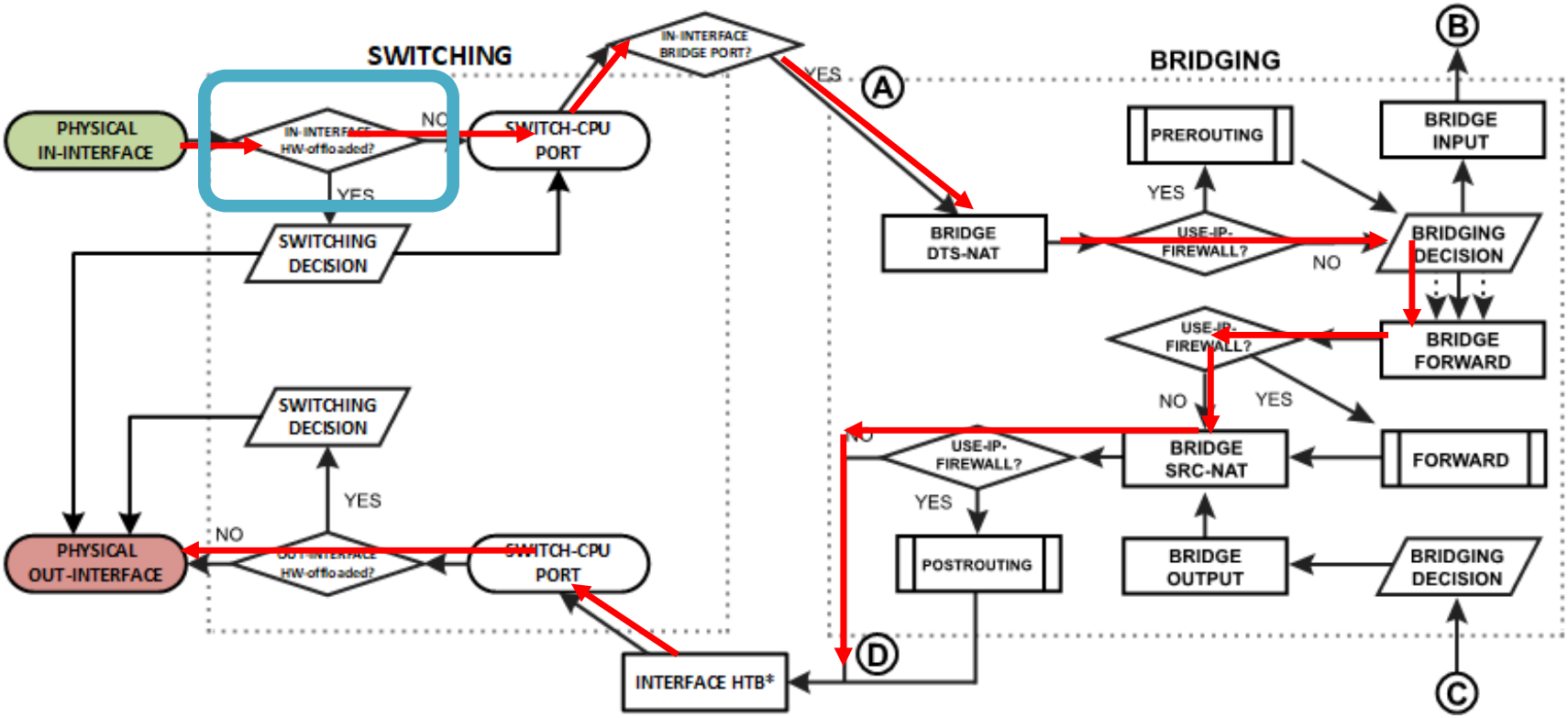

| # | | Interface | Bridge | Horizon | Trusted | Priority (... | Path Cost | PVID | Role |
|---|---|---|---|---|---|---|---|---|---|
| 0 | H | ether1 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 1 | H | ether2 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 2 | | ether3 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 3 | | ether4 | bridge-lan | | no | 80 | 10 | 1 | designated port |
| 4 | I | *wlan1* | bridge-lan | | no | 80 | 10 | 1 | disabled port |
| 5 | I | *wlan2* | bridge-lan | | no | 80 | 10 | 1 | disabled port |

6 items (1 selected)

# Example 4 – In Not HW / Out Not HW

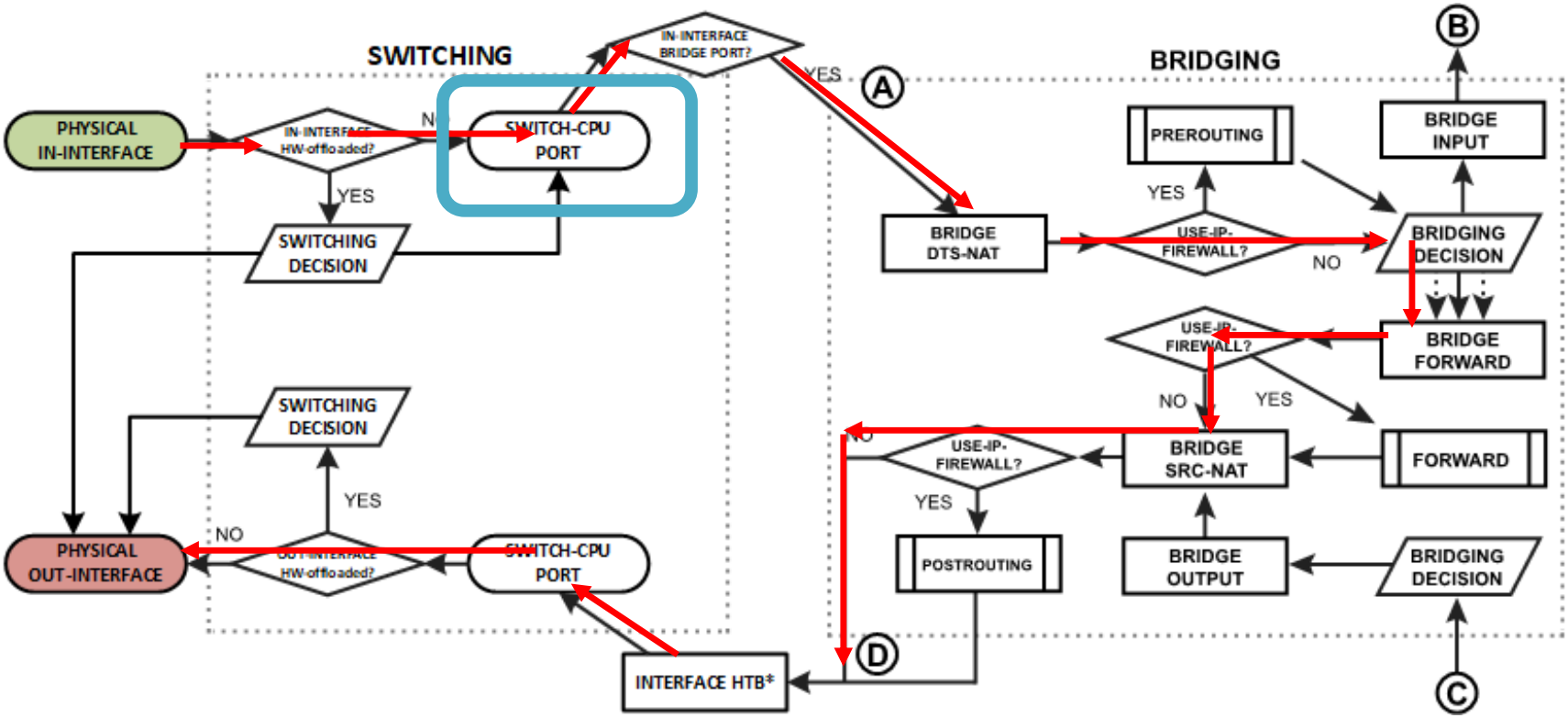


# Example 4 – In Not HW / Out Not HW



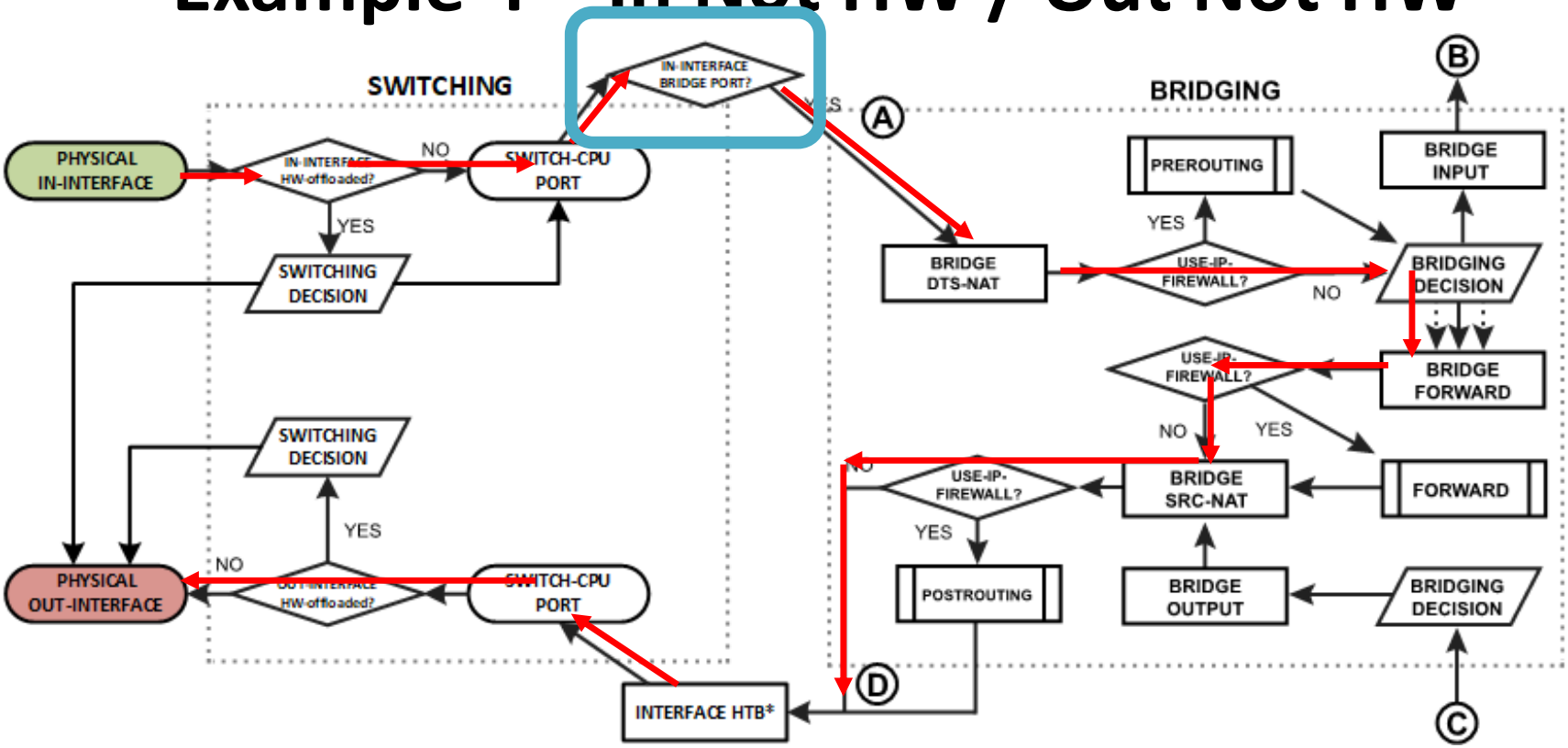

1. The switch checks whether the in-interface is a hardware offloaded interface.

# Example 4 – In Not HW / Out Not HW



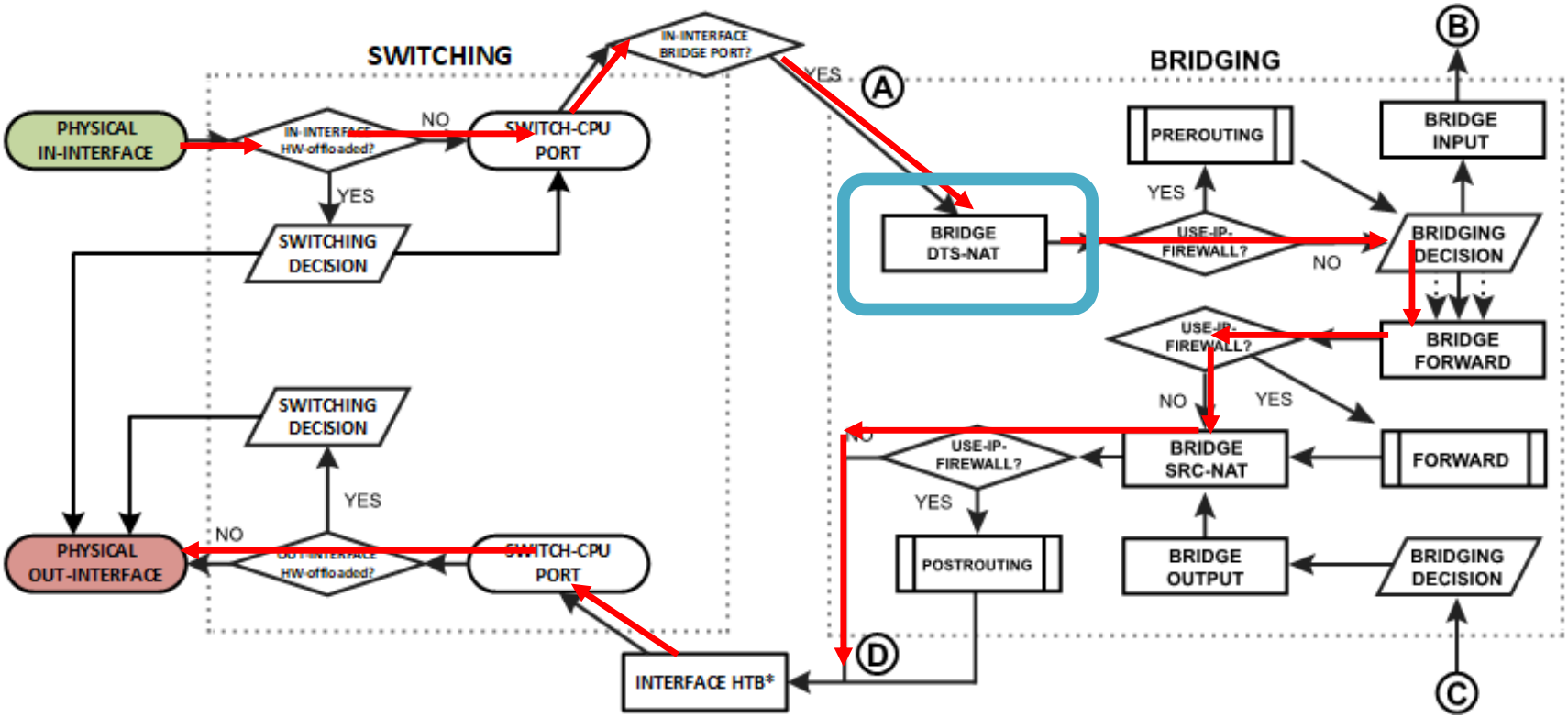

2. The packet exits though the switch CPU Port and it will be further processed by the RouterOS packet flow.

# Example 4 – In Not HW / Out Not HW



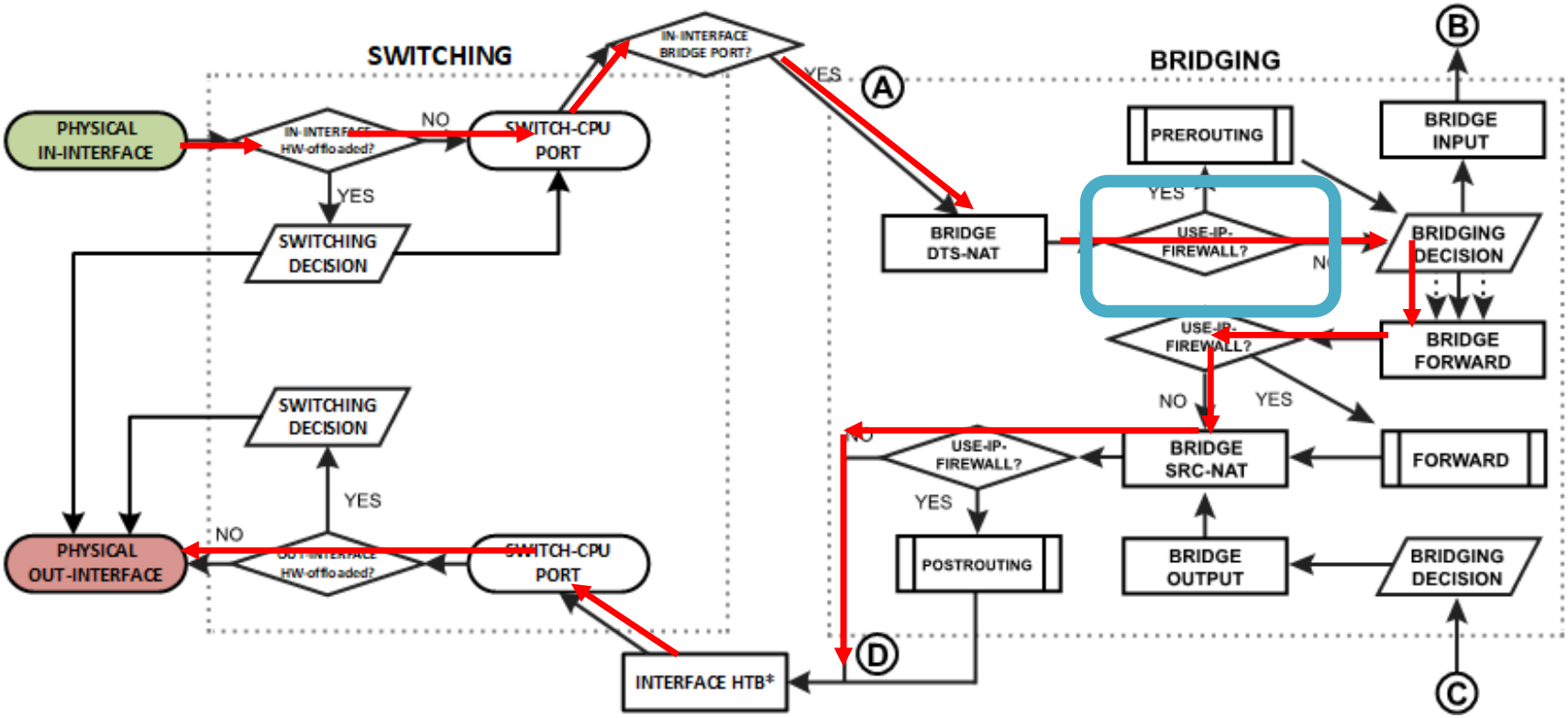

3. The device determines that in-interface is a bridge port, so it gets passed through the bridging process.

# Example 4 – In Not HW / Out Not HW



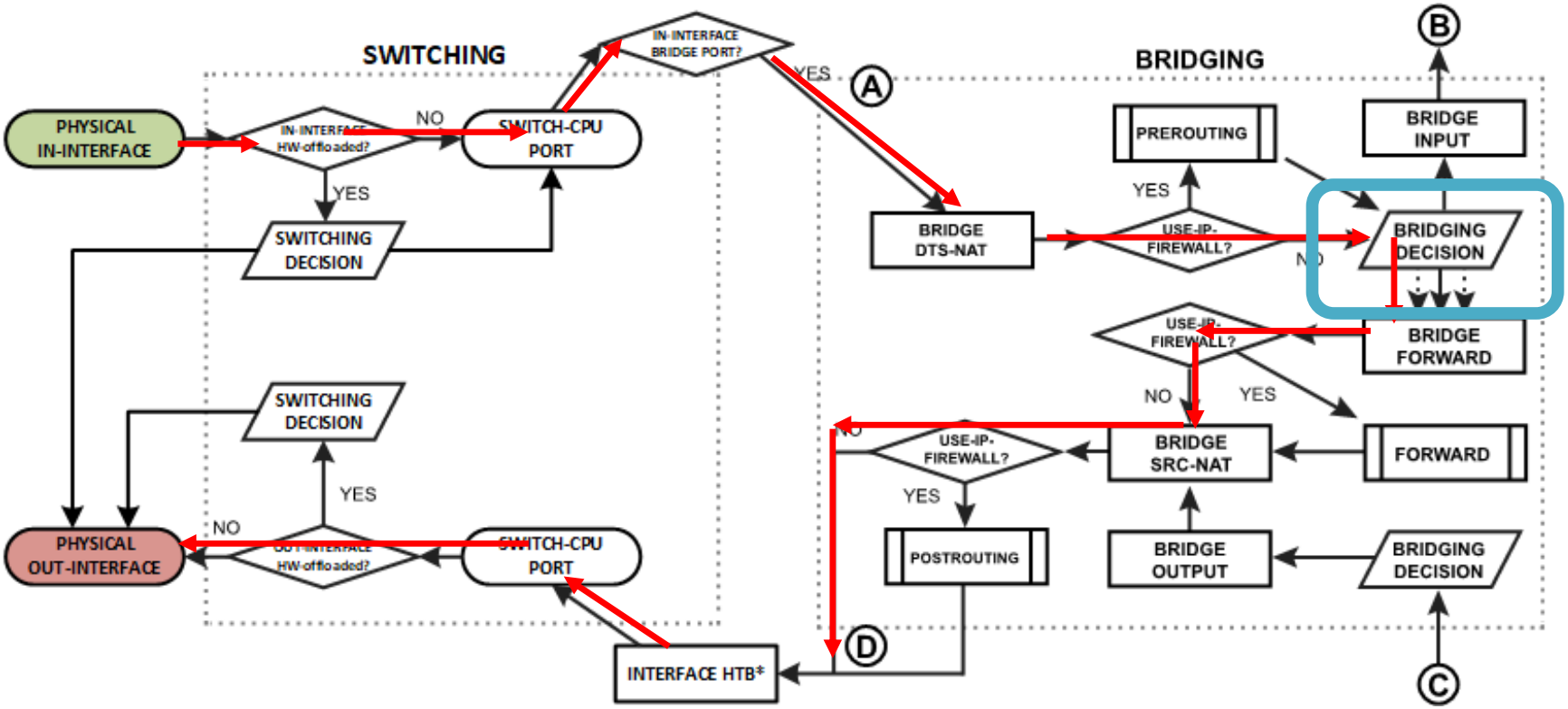

4. The packet goes through the bridge NAT dst-nat chain, where MAC destination and priority can be changed, apart from that, a packet can be simply accepted, dropped, or marked.

# Example 4 – In Not HW / Out Not HW



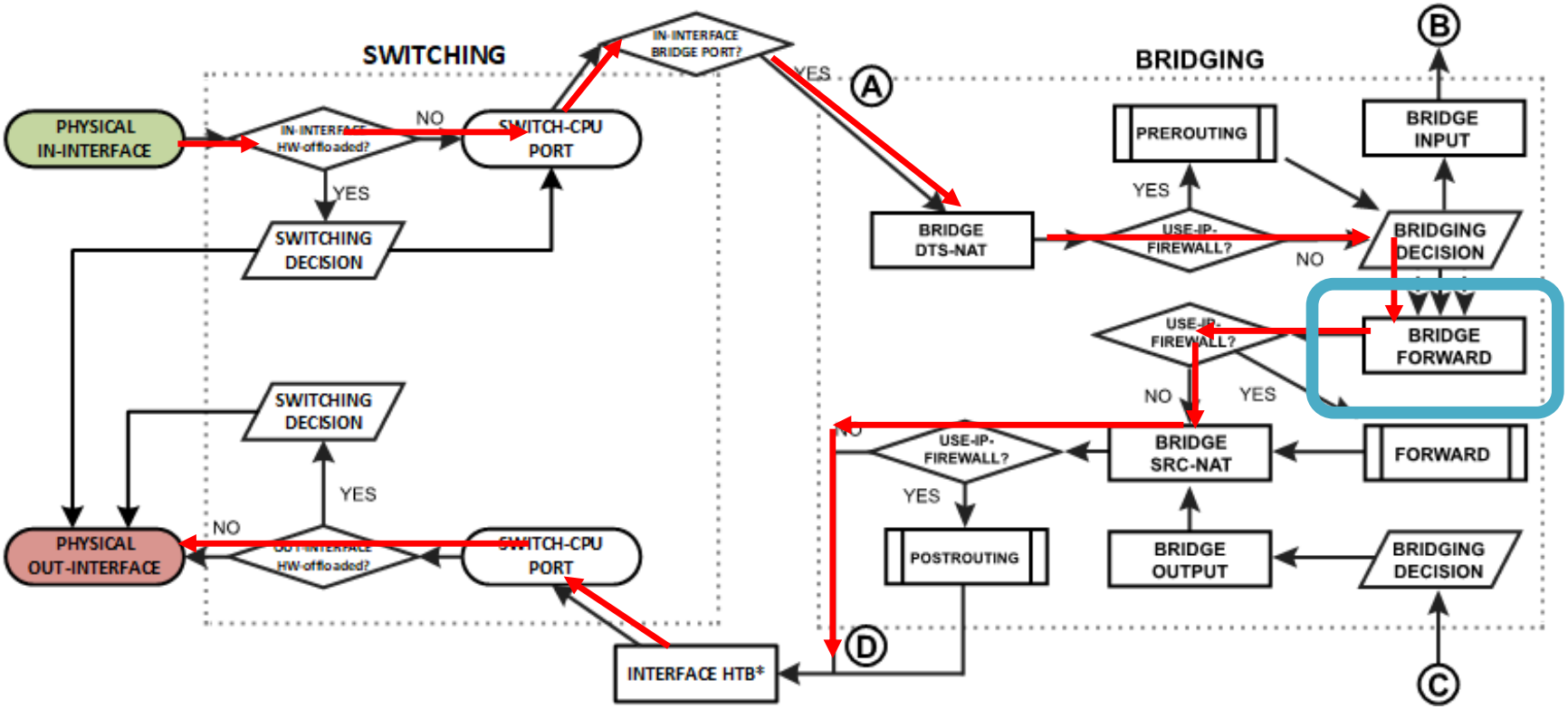

5. Checks whether the use-ip-firewall option is enabled in the bridge settings.

# Example 4 – In Not HW / Out Not HW



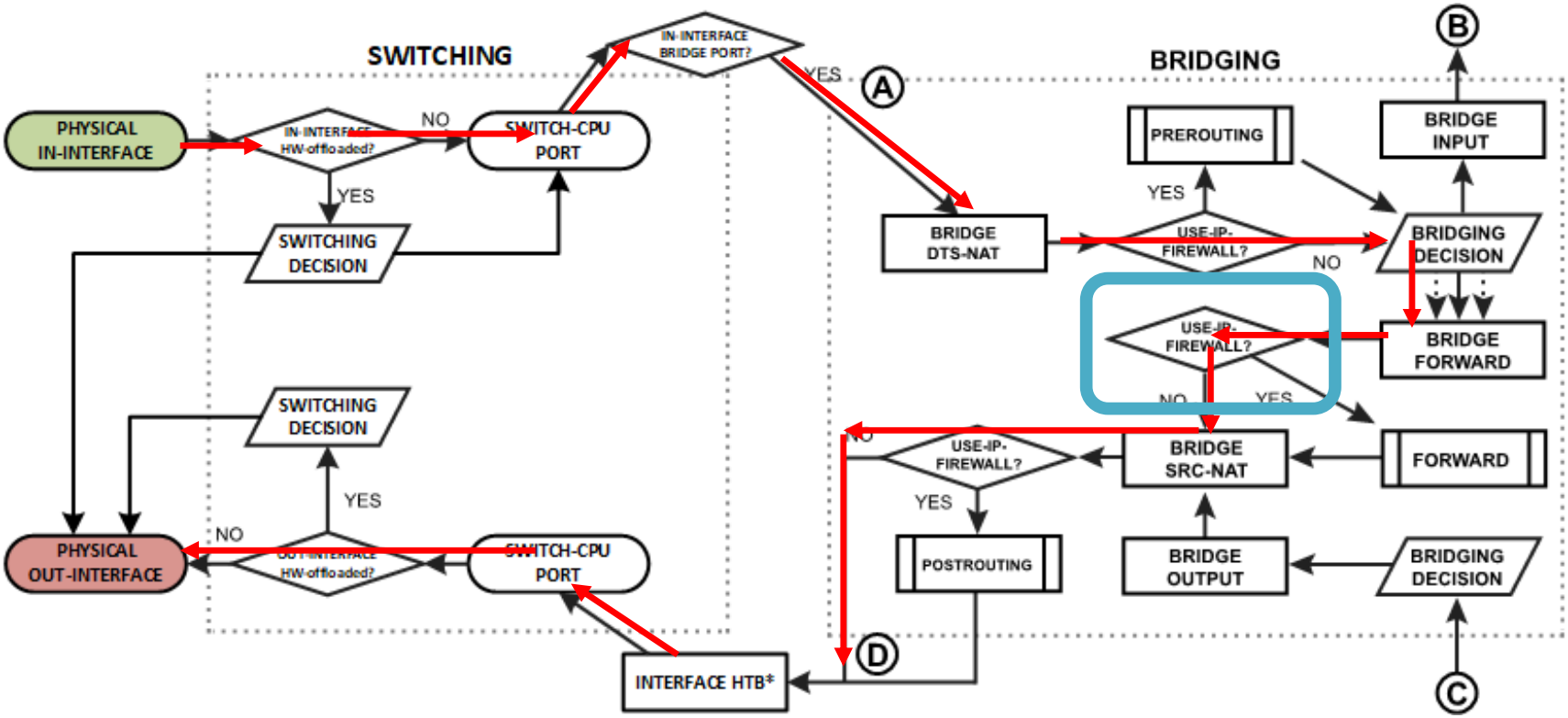

6. Run packet through the bridge host table to make a forwarding decision. A packet that ends up being flooded (e.g. broadcast, multicast, unknown unicast traffic), gets multiplied per bridge port and then processed in the bridge forward chain.

# Example 4 – In Not HW / Out Not HW



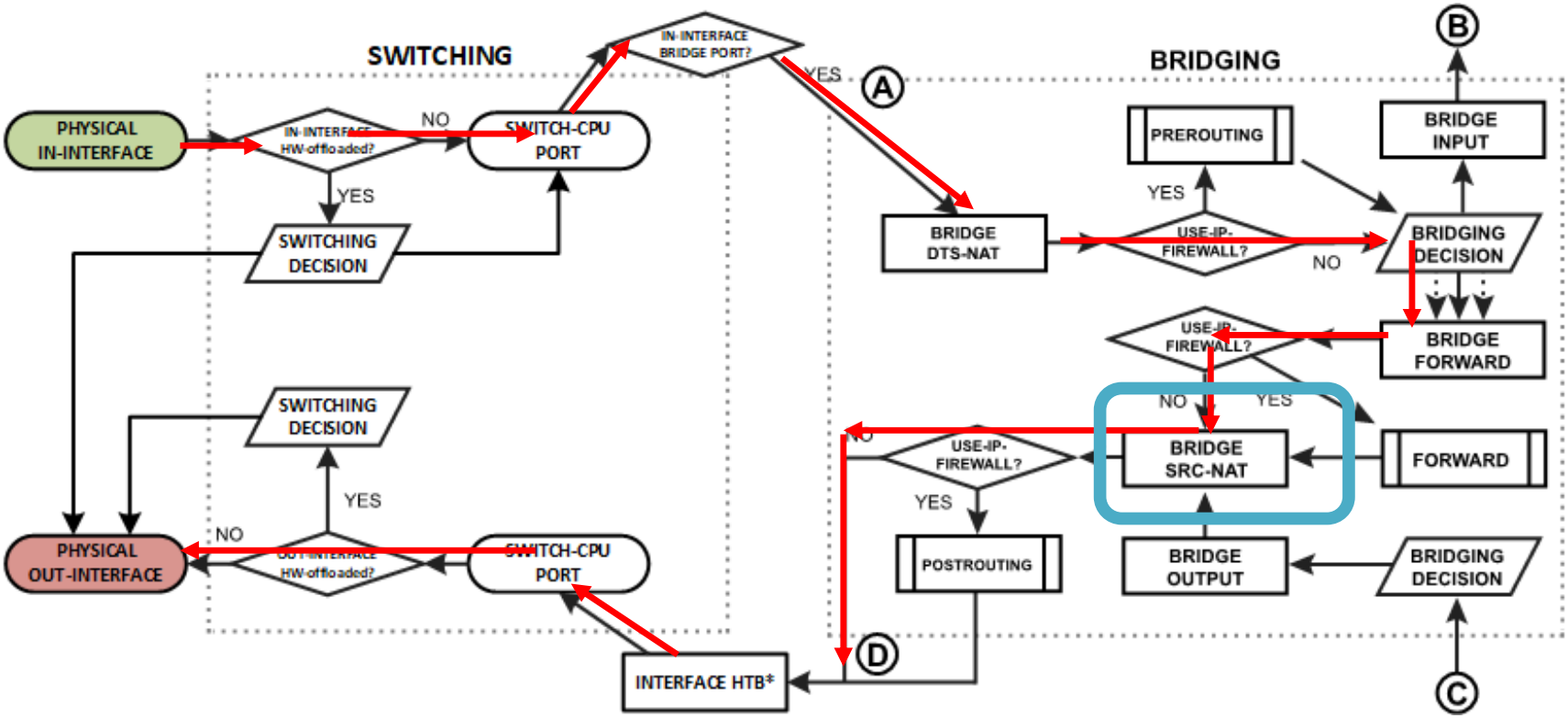

7. The packet goes through the bridge filter forward chain, where priority can be changed or packet can be simply accepted, dropped, or marked.

# Example 4 – In Not HW / Out Not HW



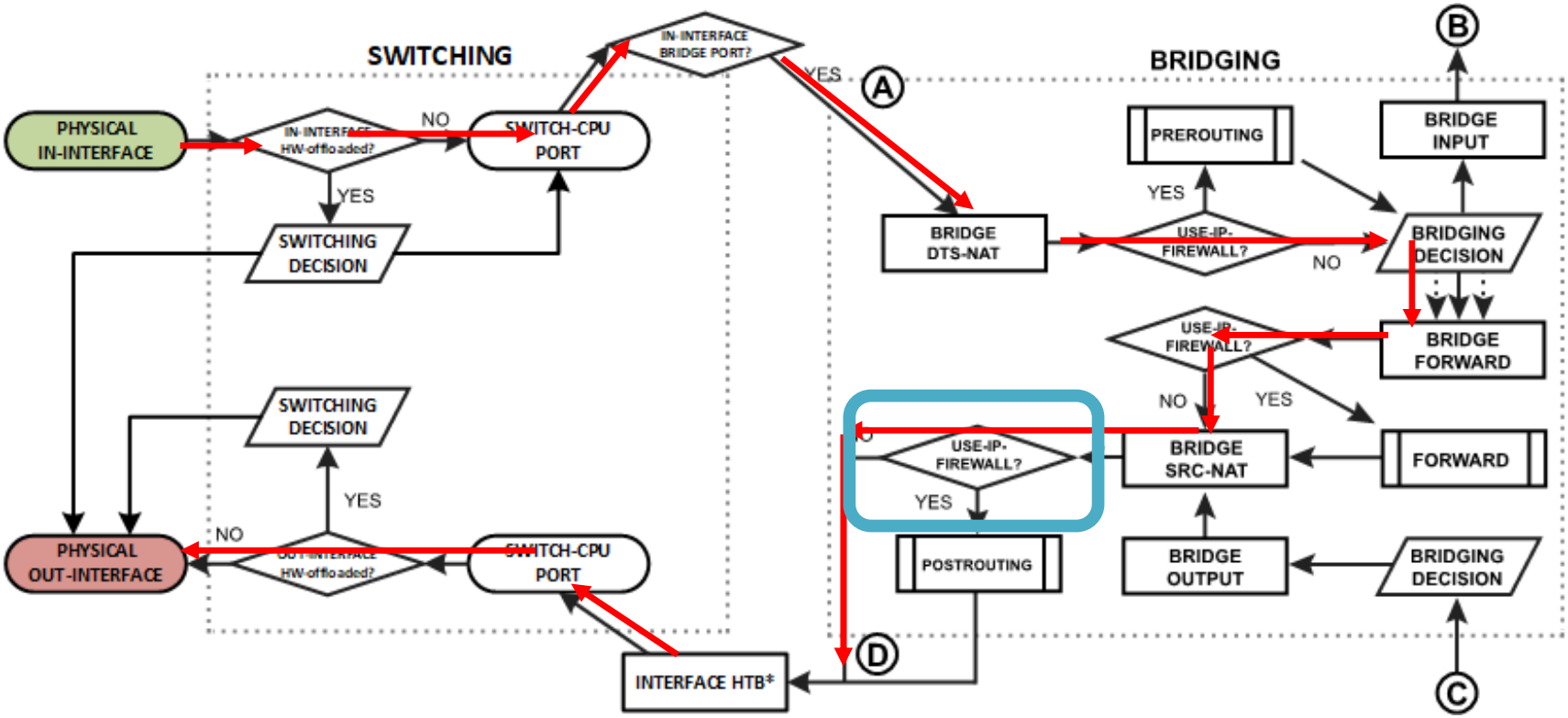

8. Checks whether the use-ip-firewall option is enabled in the bridge settings.

# Example 4 – In Not HW / Out Not HW



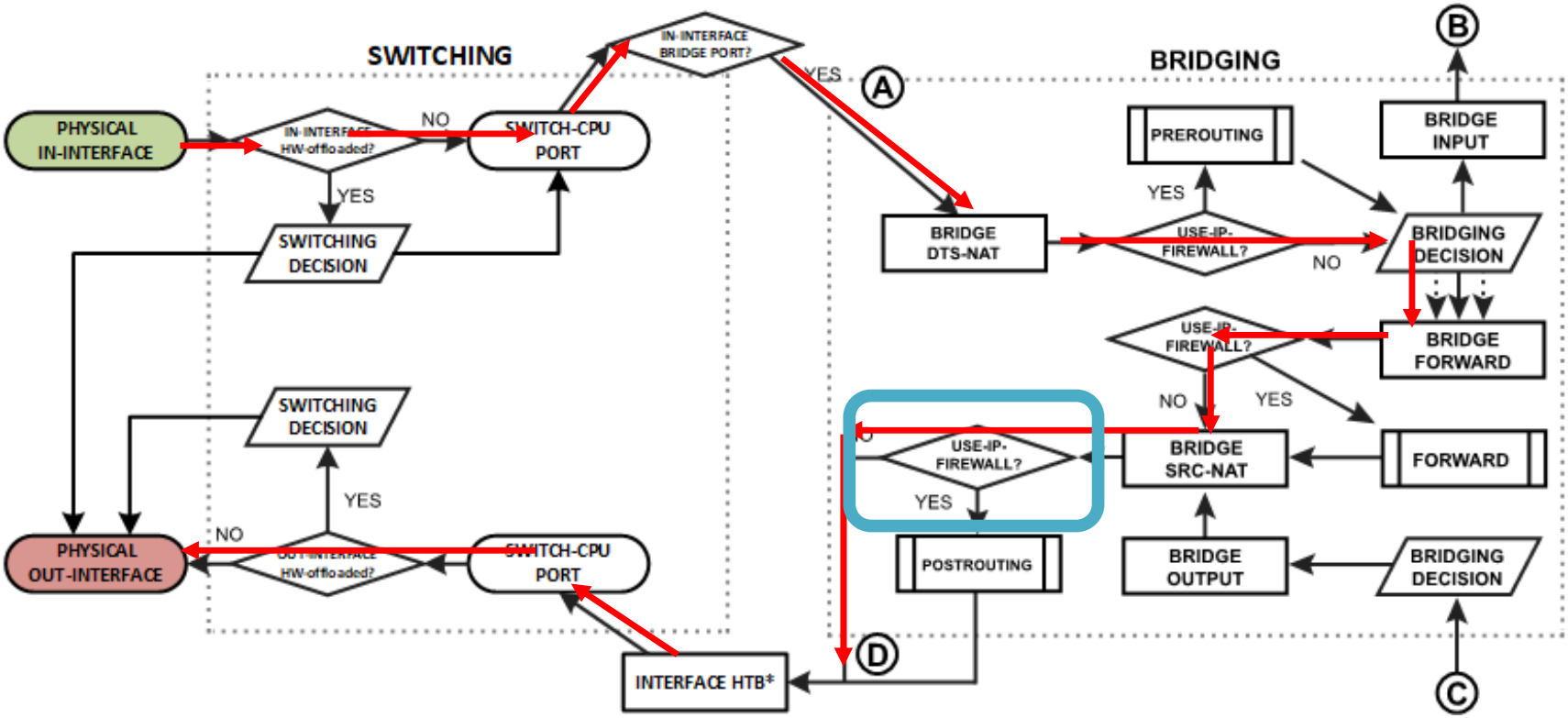

9. The packet goes through the bridge NAT src-nat chain, where MAC source and priority can be changed, apart from that, a packet can be simply accepted, dropped, or marked.

# Example 4 – In Not HW / Out Not HW



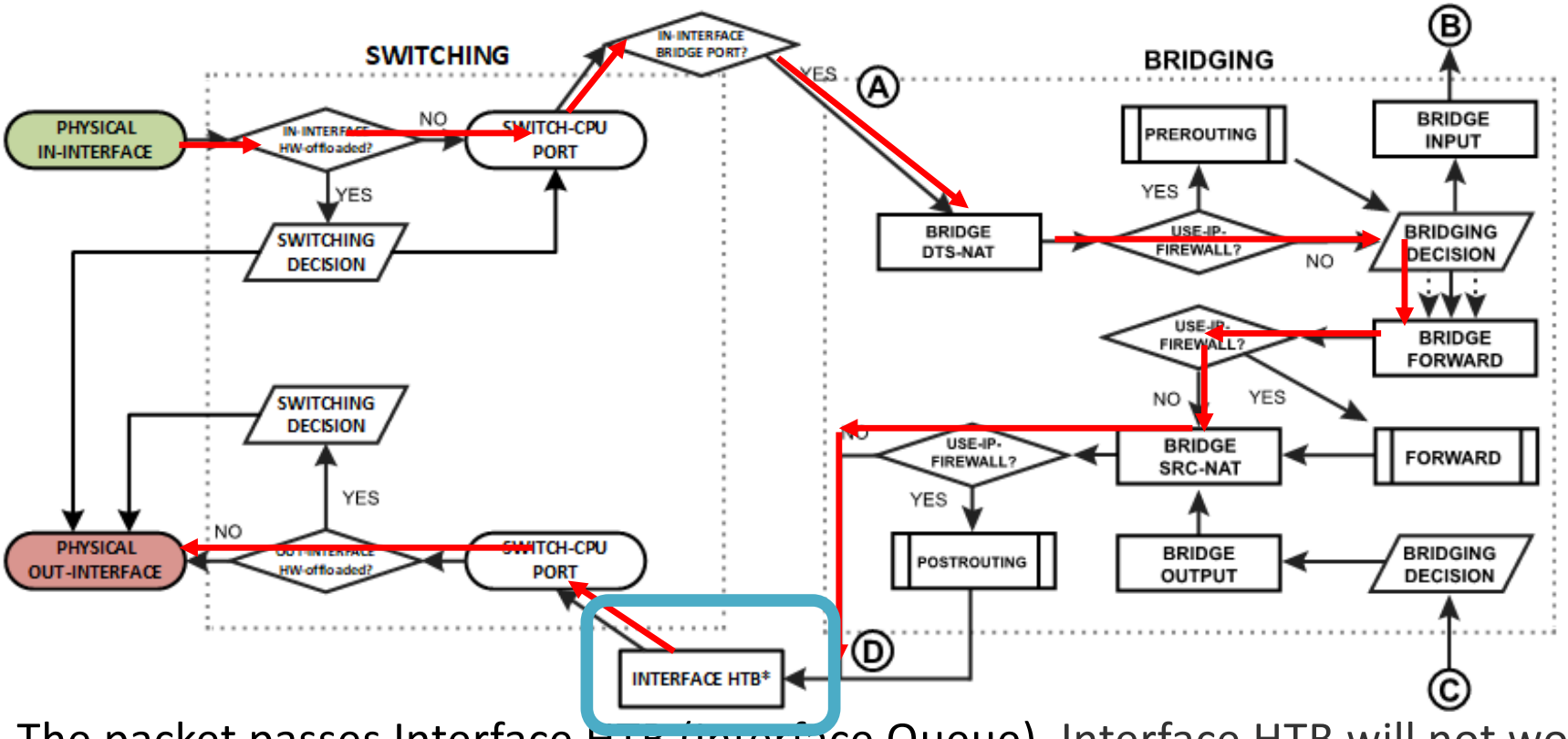

10. Checks whether the use-ip-firewall option is enabled in the bridge settings.

# Example 4 – In Not HW / Out Not HW



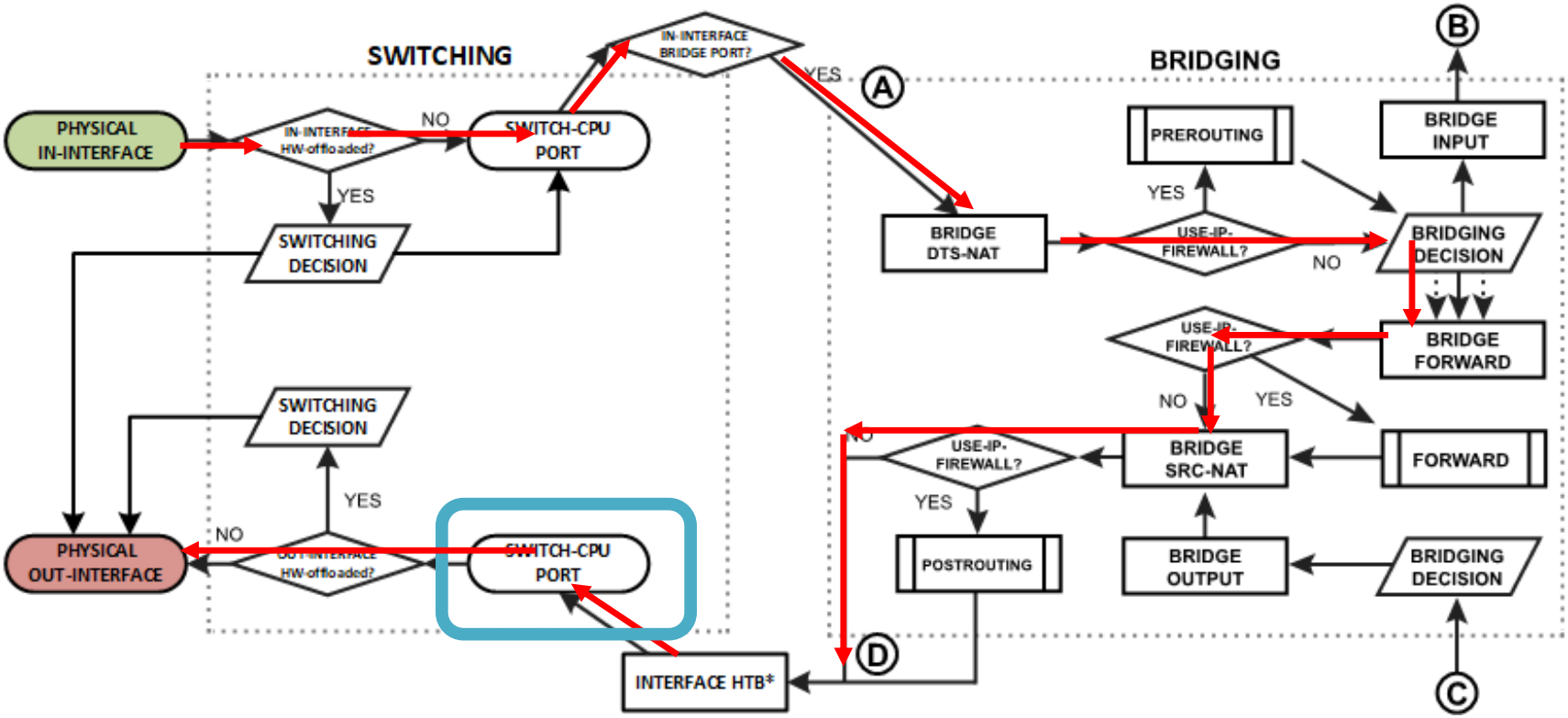

11. Checks whether the use-ip-firewall option is enabled in the bridge settings. The packet now leaves the bridge process.

# Example 4 – In Not HW / Out Not HW



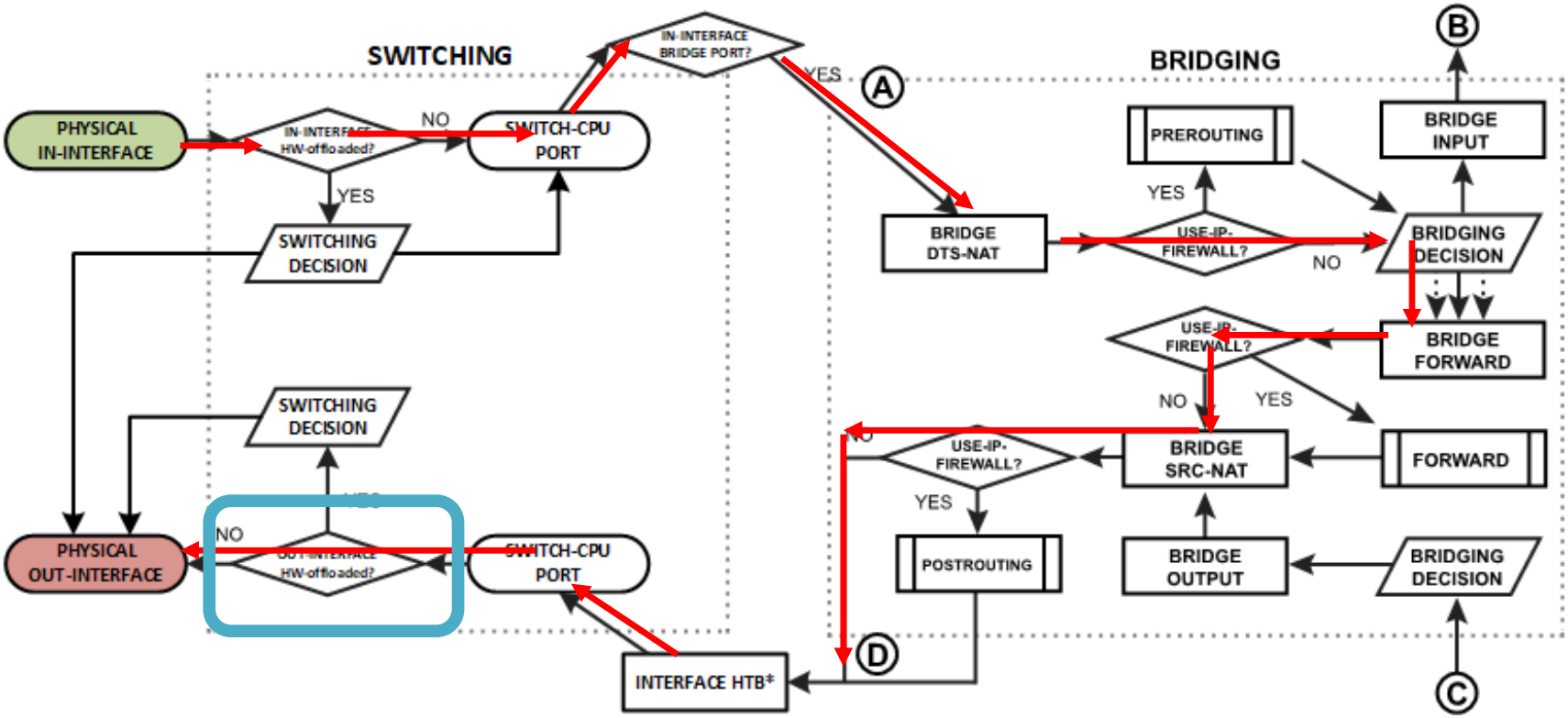

12. The packet passes Interface HTB (Interface Queue). Interface HTB will not work correctly when the out-interface is hardware offloaded and the bridge Fast Path is not active.

# Example 4 – In Not HW / Out Not HW




13. The packet that exits the RouterOS software processing is received on the switch-cpu port.

# Example 4 – In Not HW / Out Not HW




14. The switch checks whether the out-interface is a hardware offloaded interface and the packet is send out of the physical out-interface.

# Controlling Layer 2 Traffic

- RouterOS has a number of places where Layer2 packets can either be filtered or limited some features include:
  - Bridge Filter
  - Bridge NAT
  - Simple Queue
  - Interface Queue
  - Switch Rules
  - Switch Rate control
  - IP Firewall
  - Bridge Horizon

# Controlling Layer 2 Traffic

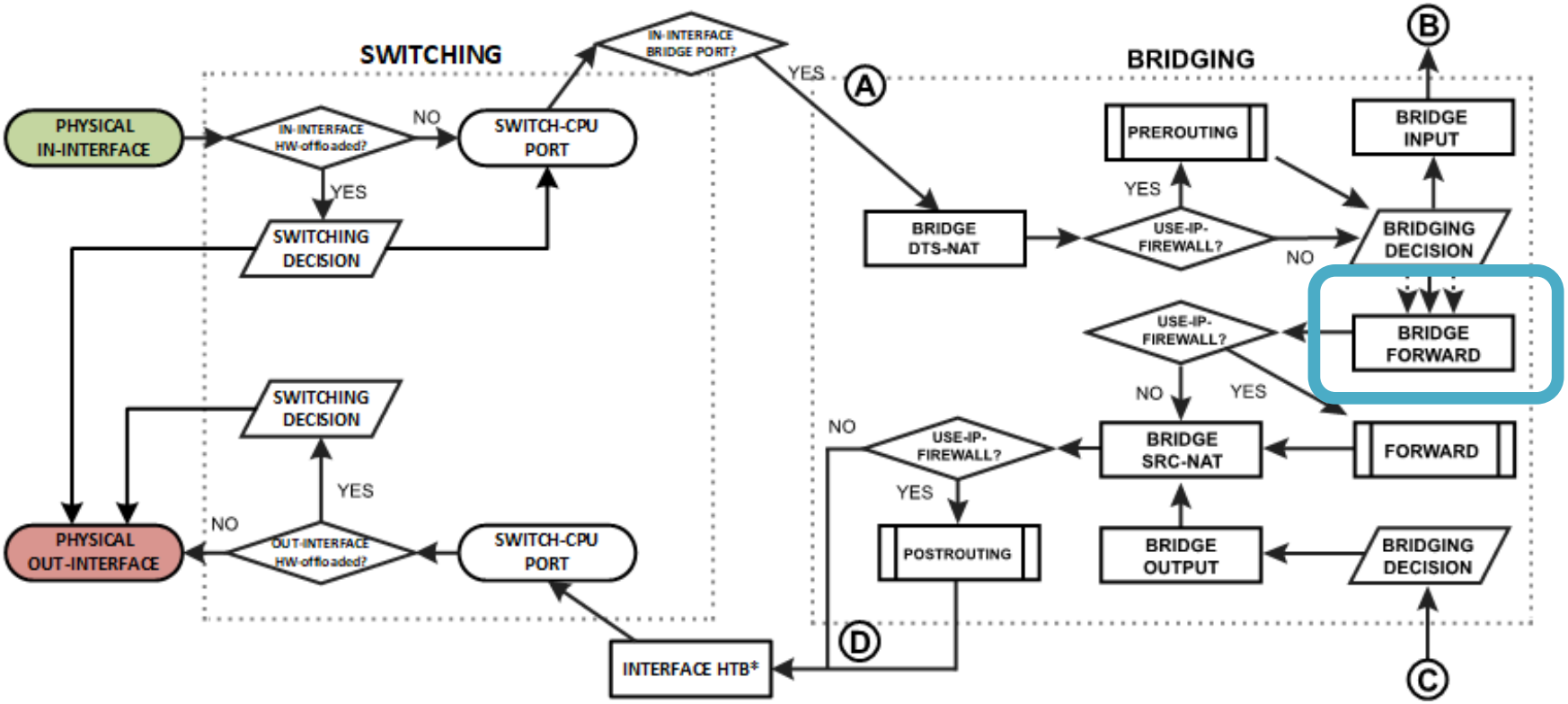
- Some of these features run in software and some run in hardware.
- Enabling some of these features can disable HW-offloading on either just the interface or for the whole bridge.

# Bridge Filter Forward

- Bridge filter forward rule to block TCP/80 with in-interface ether1
- What can PC1 get to?
  - Webserver on PC2?
  - Webserver on PC4?
  - Webfig on Router?
  - http web pages on the internet?

```
add action=accept chain=forward dst-port=80 in-interface=ether1 ip-protocol=tcp mac-protocol=ip
```

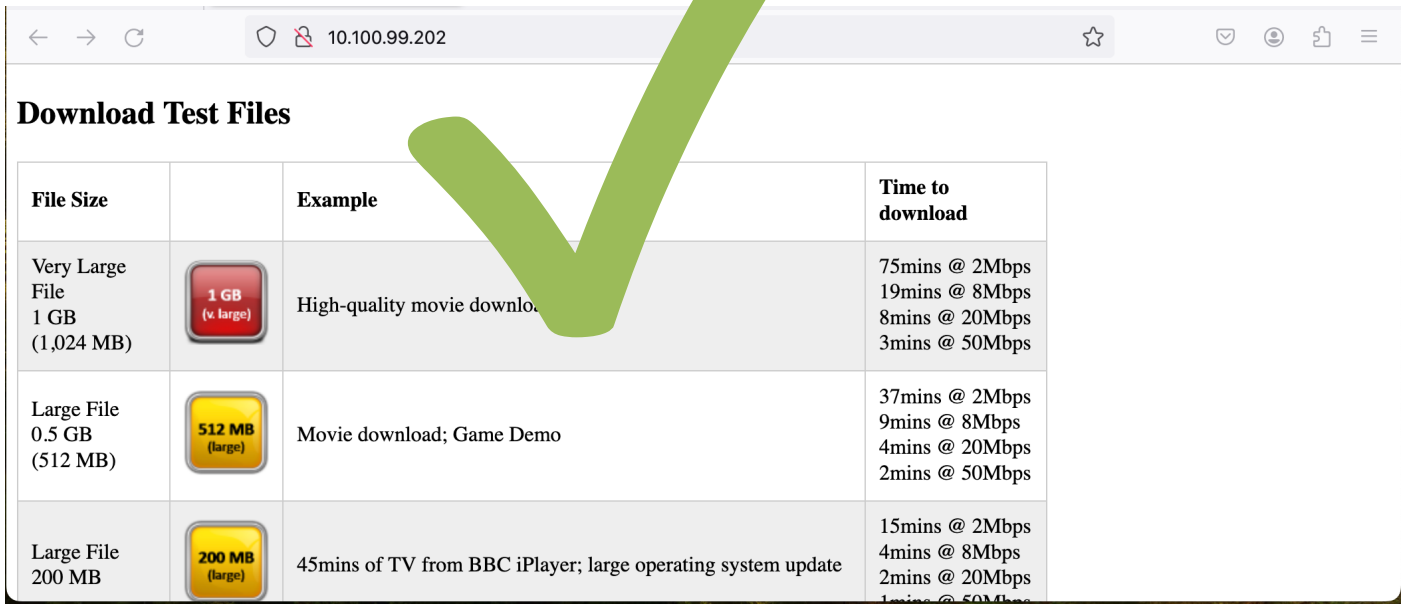
# Bridge Filter Forward



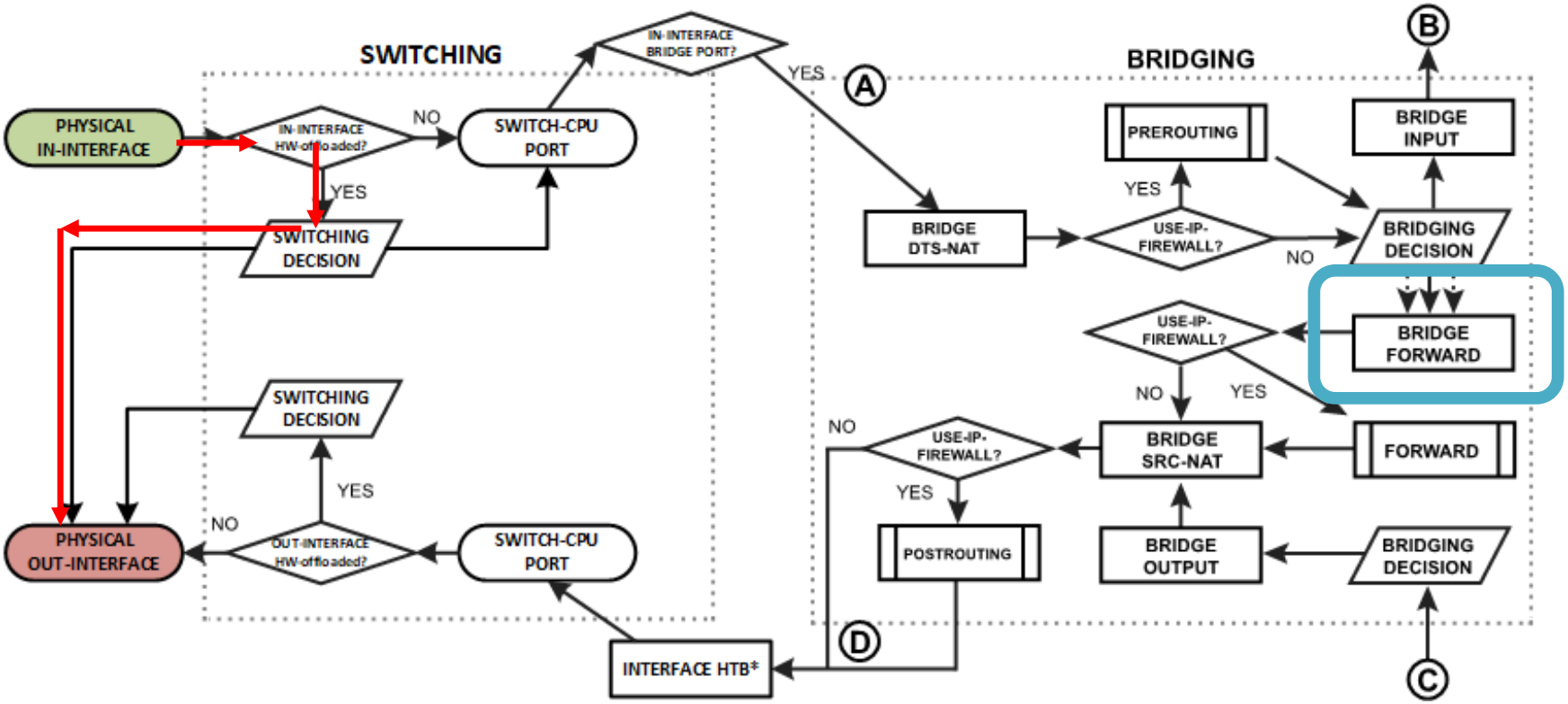

- Bridge Filter forward is part of Bridge Forward Function.
- Bridge Filter rules do not disable HW-offloading.

# Bridge Filter forward – Webserver on PC2

- Can we get to the webserver on PC2 (ether2)?
- PC1 (ether1) is HW offloaded
- PC2 (ether2) is HW offloaded



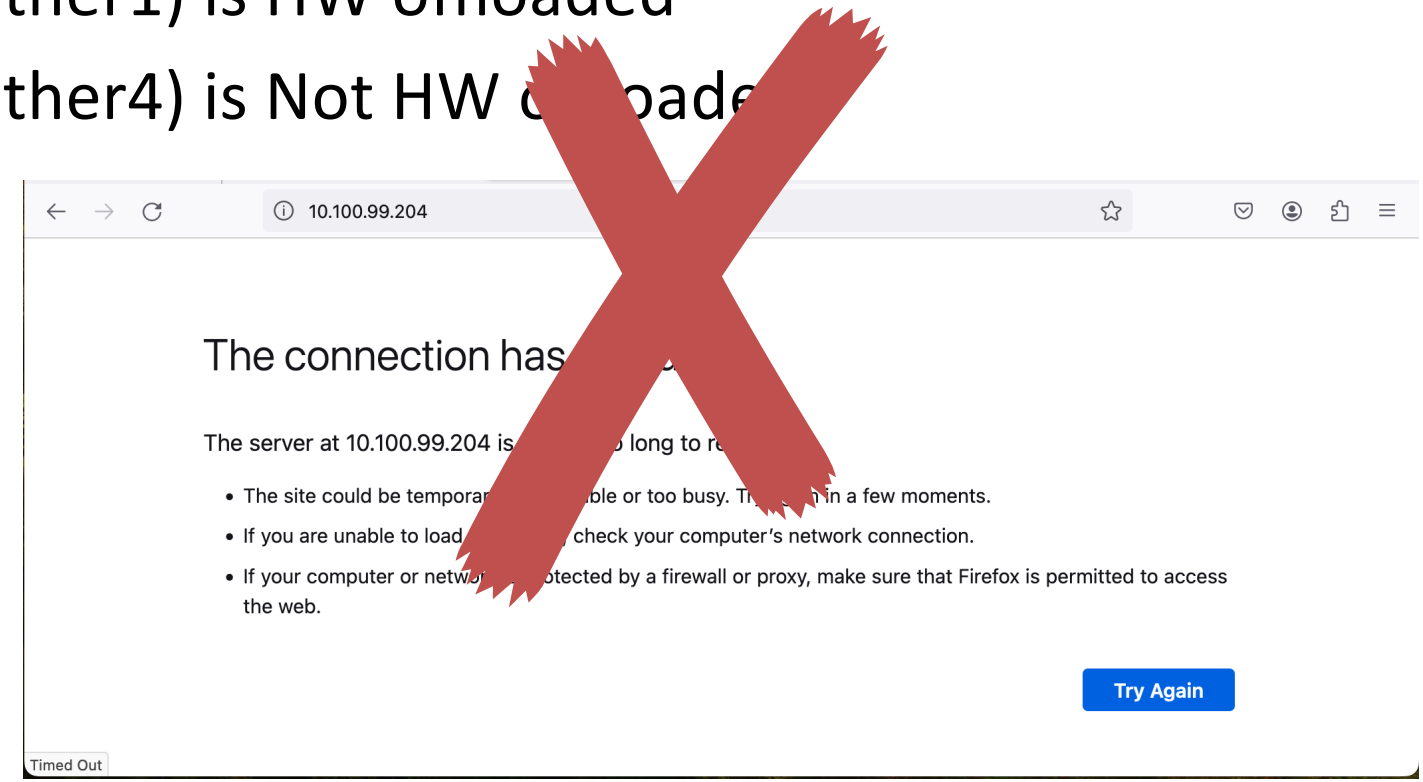
# Bridge Filter Forward – Webserver on PC2



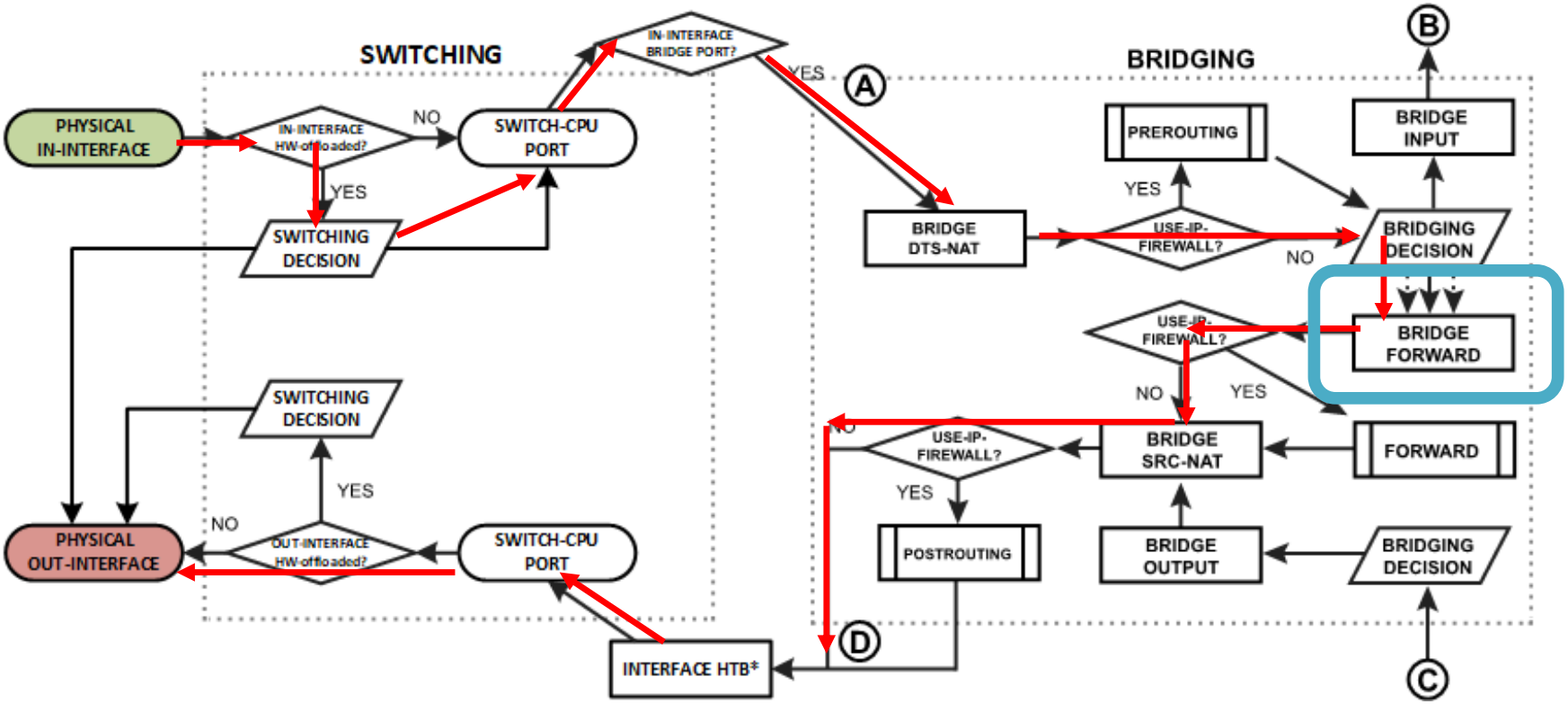

- Bridge filter rules will not apply and web page loads.

# Bridge Filter Forward – Webserver on PC4

- Can we get to the webserver on PC4 (ether4)?
- PC1 (ether1) is HW offloaded
- PC4 (ether4) is Not HW offloaded



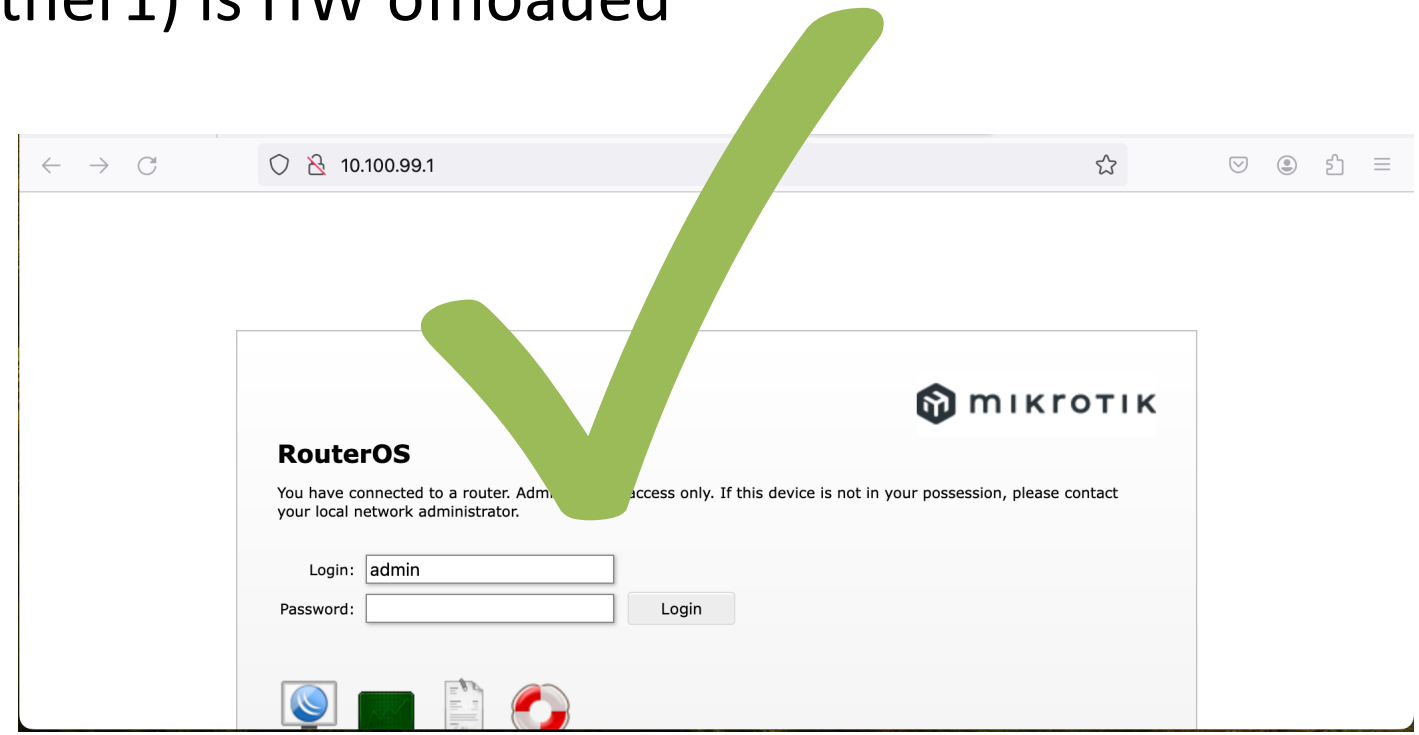
# Bridge Filter Forward – Webserver on PC4



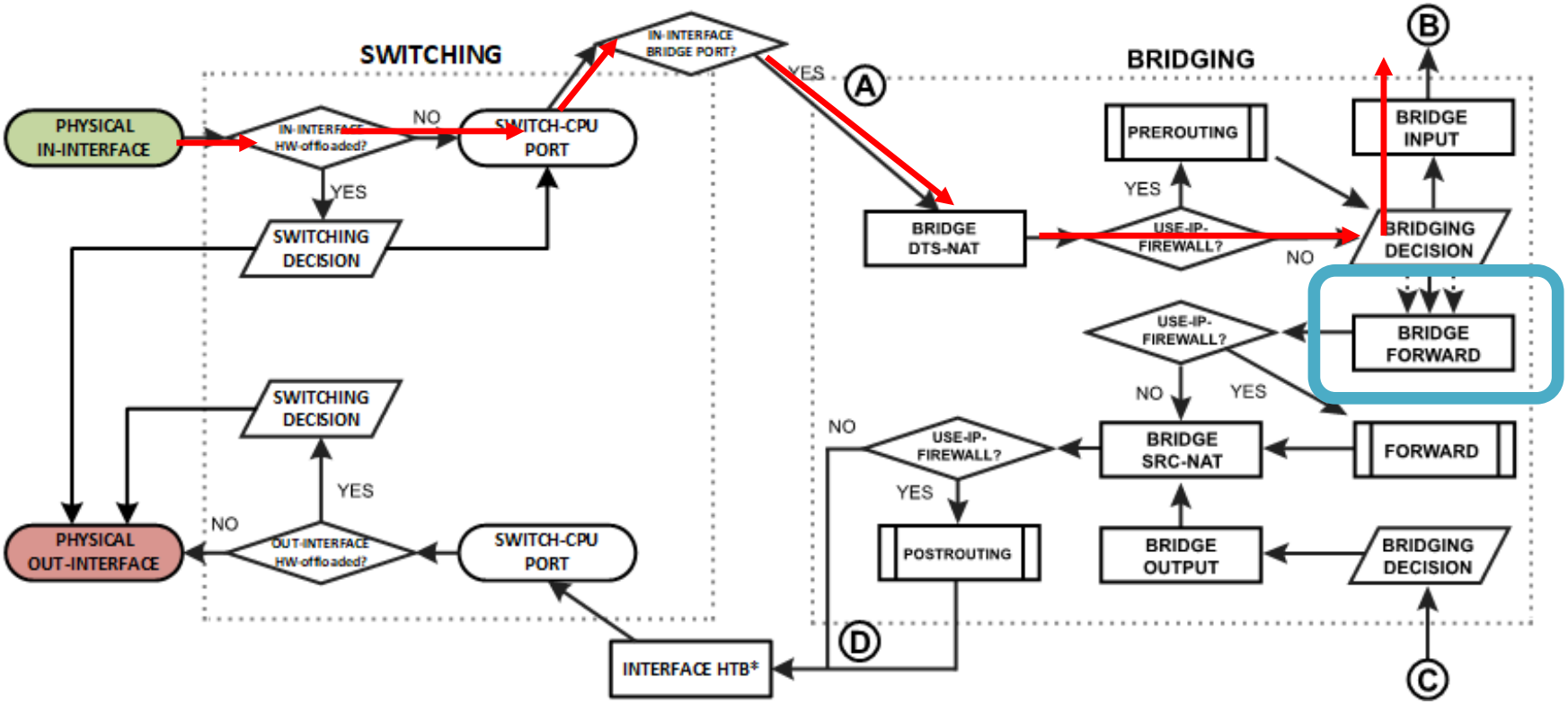

- Bridge Filter forward will apply when in-interface is HW offloaded and Out interface is not HW-offloaded.

# Bridge Filter Forward – Webfig on Router

- Can we get to the webfig on router?
- PC1 (ether1) is HW offloaded



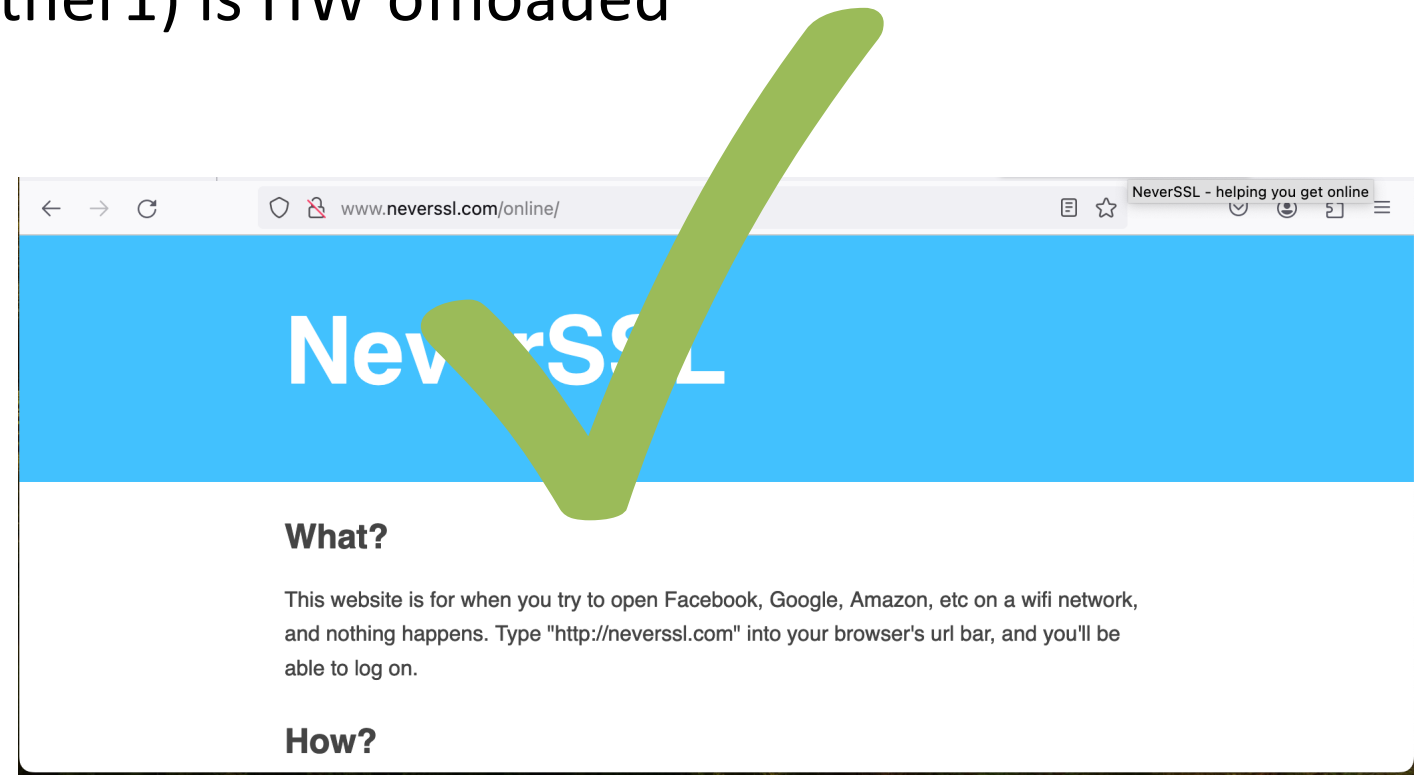
# Bridge Filter Forward – webfig on router



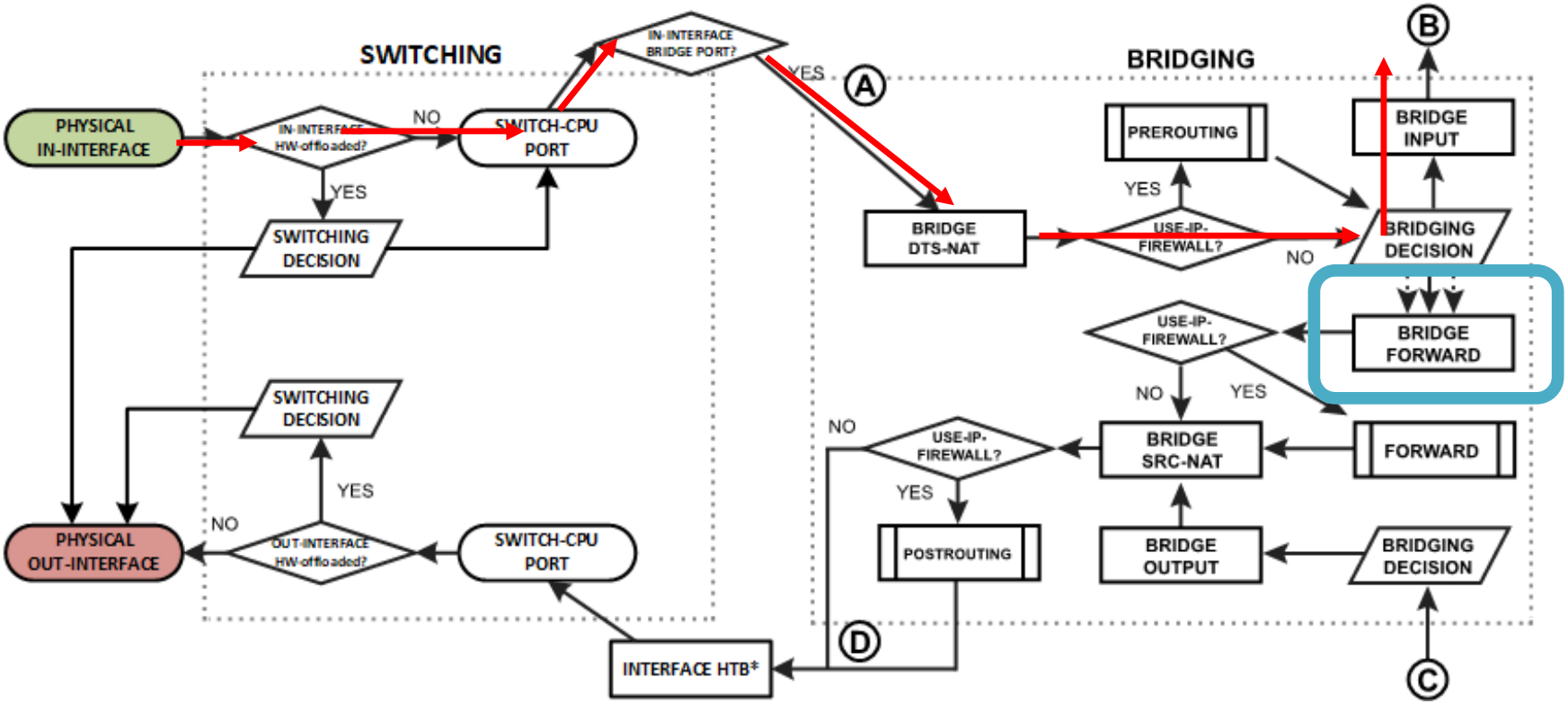

- Bridge Filter forward rules does not apply here as the traffic is bridge input.

# Bridge Filter forward – http websites on internet

- Can we get to the http websites on the internet?
- PC1 (ether1) is HW offloaded



# Bridge Filter Forward – http webpages on internet



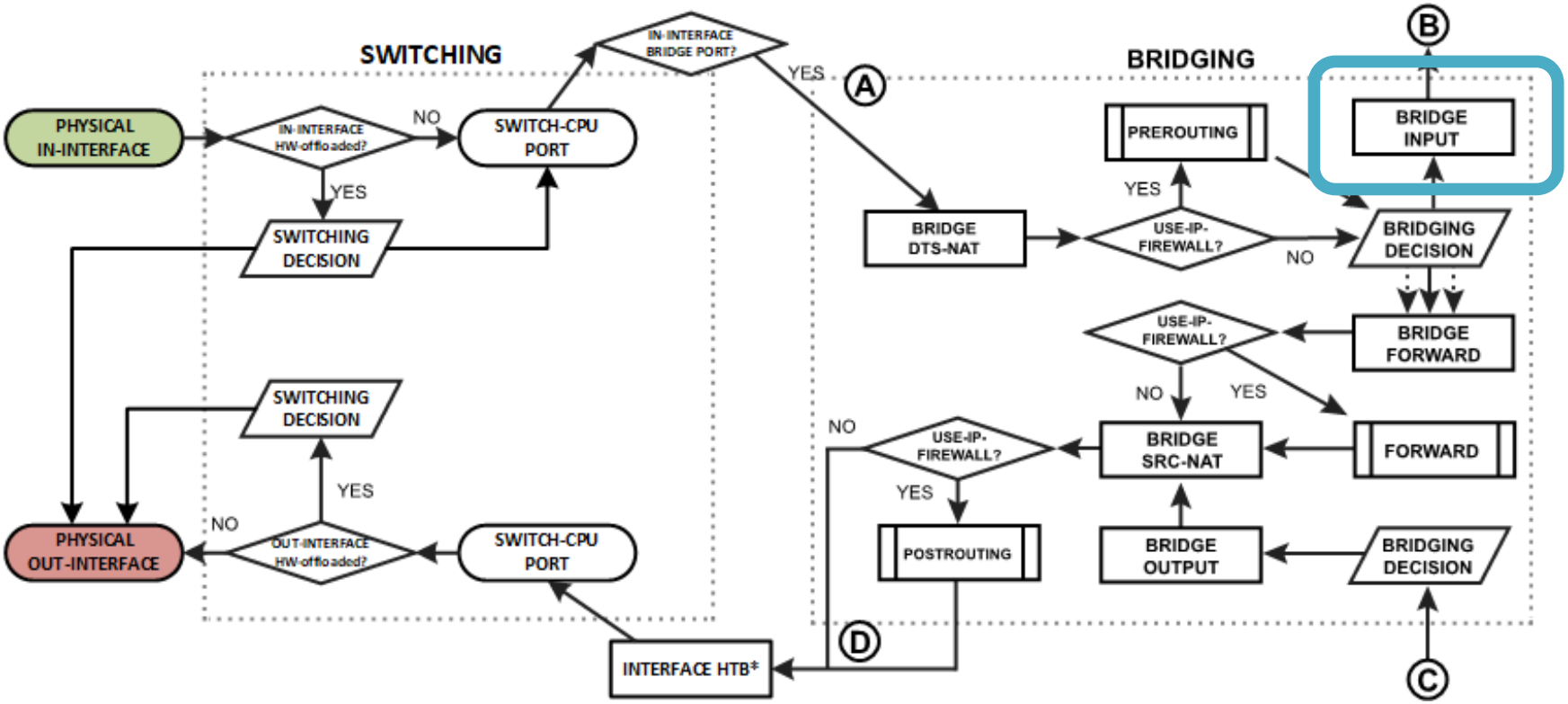

- Bridge Filter forward rules does not apply here as the traffic is bridge input.

# Bridge Filter Input

- Bridge filter input rule to block TCP/80 with in-interface ether1
- What can PC1 get to?
  - Webserver on PC2?
  - Webserver on PC4?
  - Webfig on Router?
  - http web pages on the internet?

```
add action=accept chain=input dst-port=80 in-interface=ether1 ip-protocol=tcp mac-protocol=ip
```

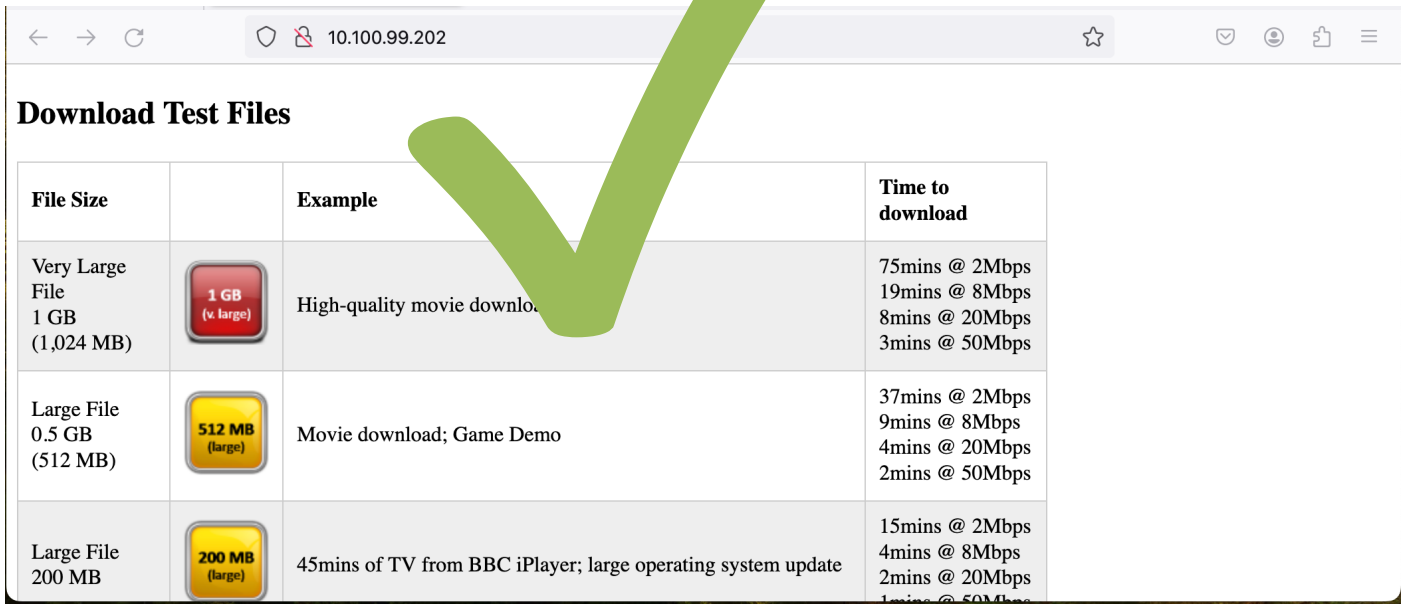
# Bridge Filter Input



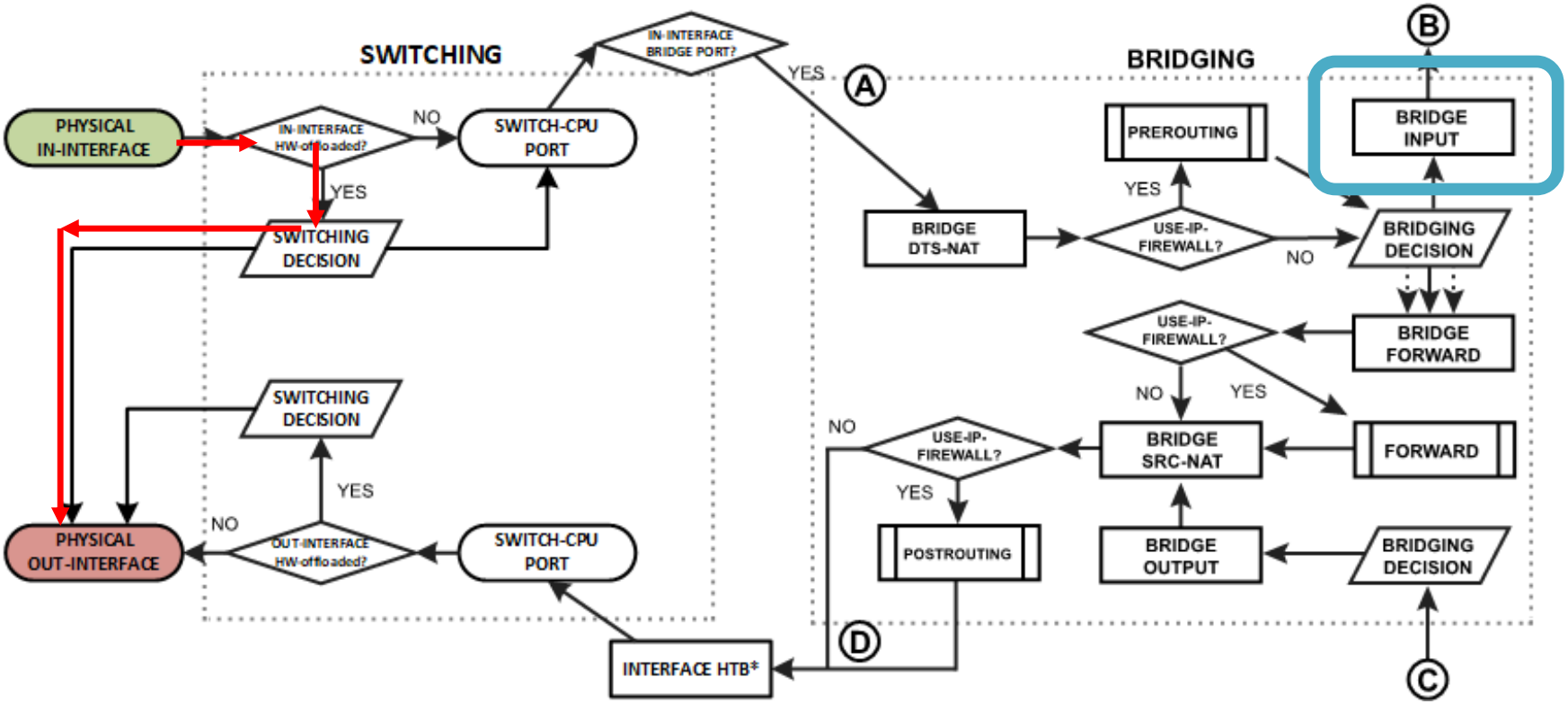

- Bridge Filter input is part of Bridge input function.
- Bridge Filter input rules do not disable HW-offloading.

# Bridge Filter Input – Webserver on PC2

- Can we get to the webserver on PC2 (ether2)?
- PC1 (ether1) is HW offloaded
- PC2 (ether2) is HW offloaded



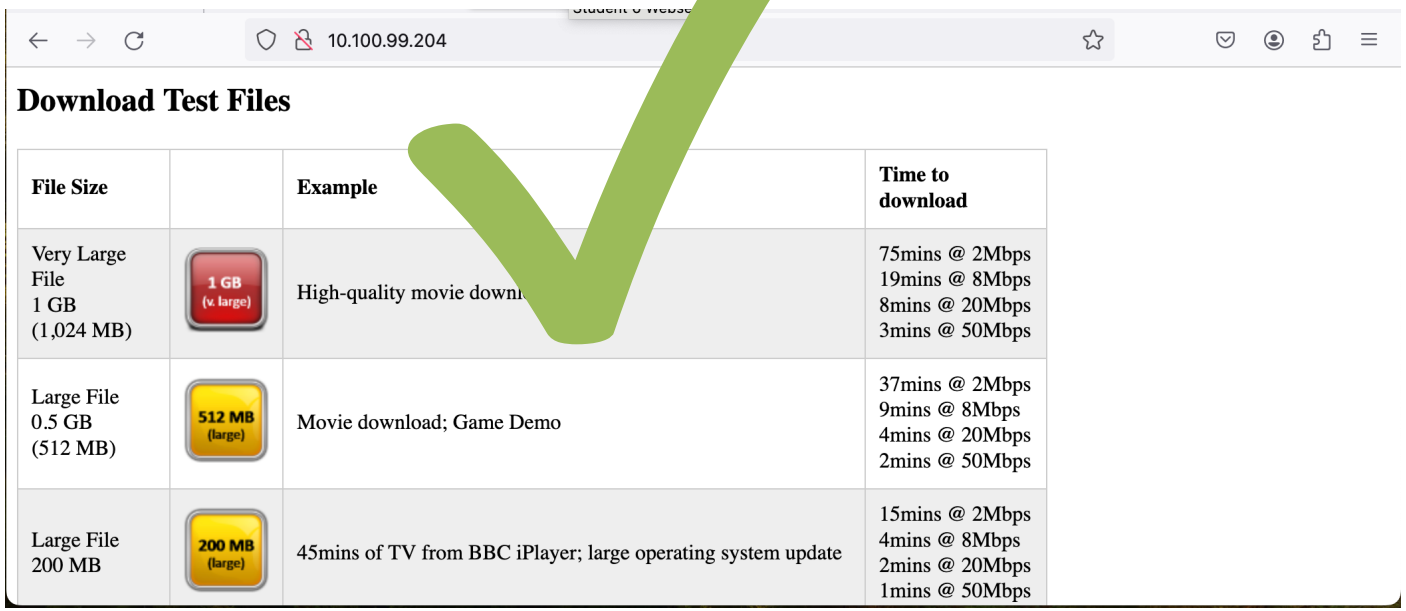
# Bridge Filter Input – Webserver on PC2



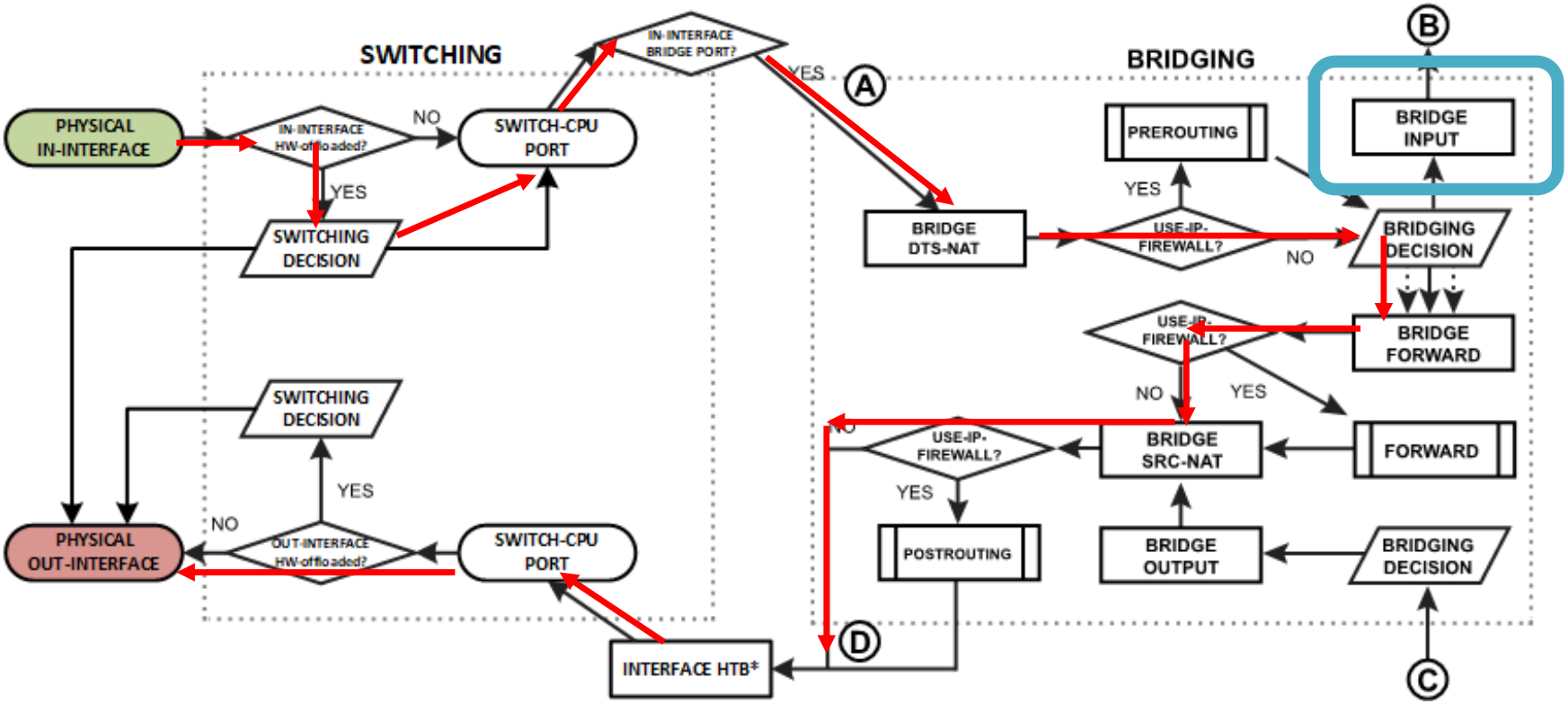

- Bridge filter input rules will not apply and web page loads.

# Bridge Filter Input – Webserver on PC4

- Can we get to the webserver on PC4 (ether4)?
- PC1 (ether1) is HW offloaded
- PC4 (ether4) is Not HW offloade



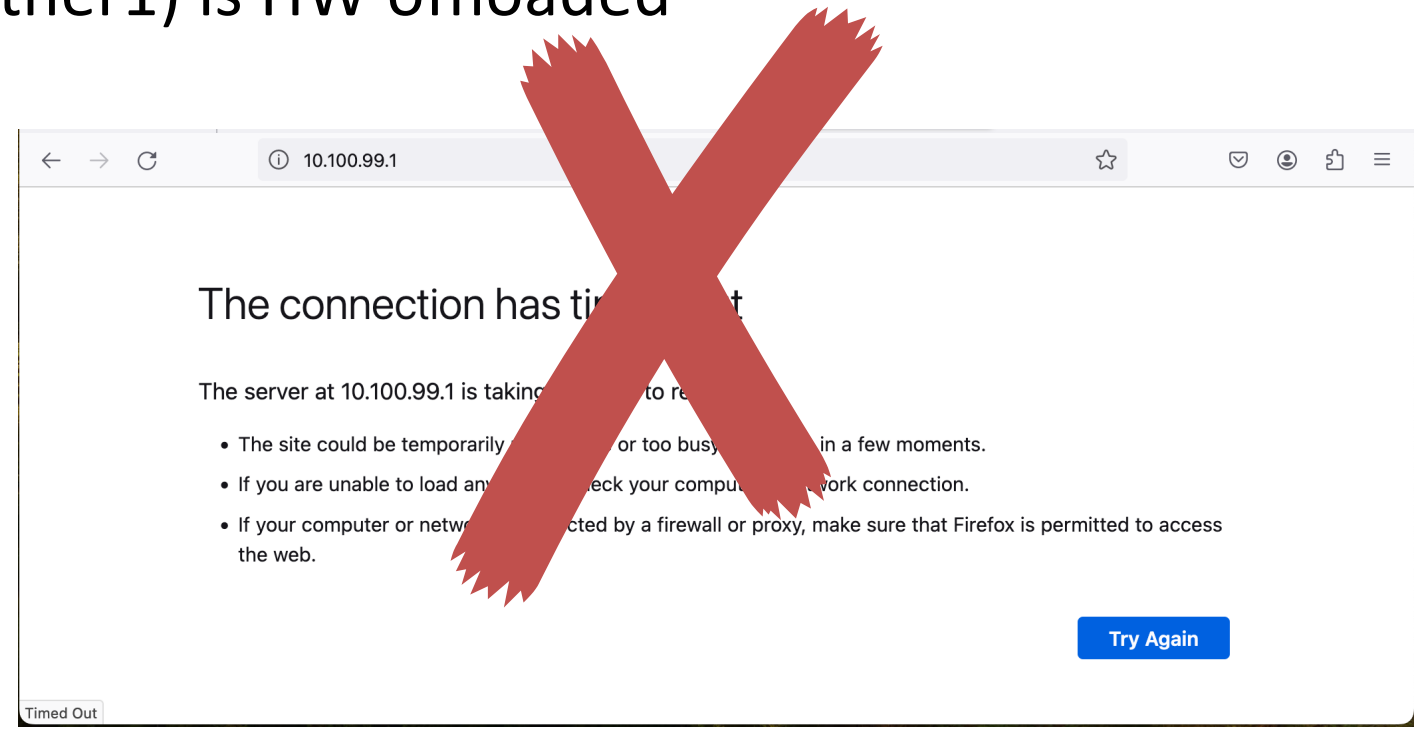
# Bridge Filter Input – Webserver on PC4



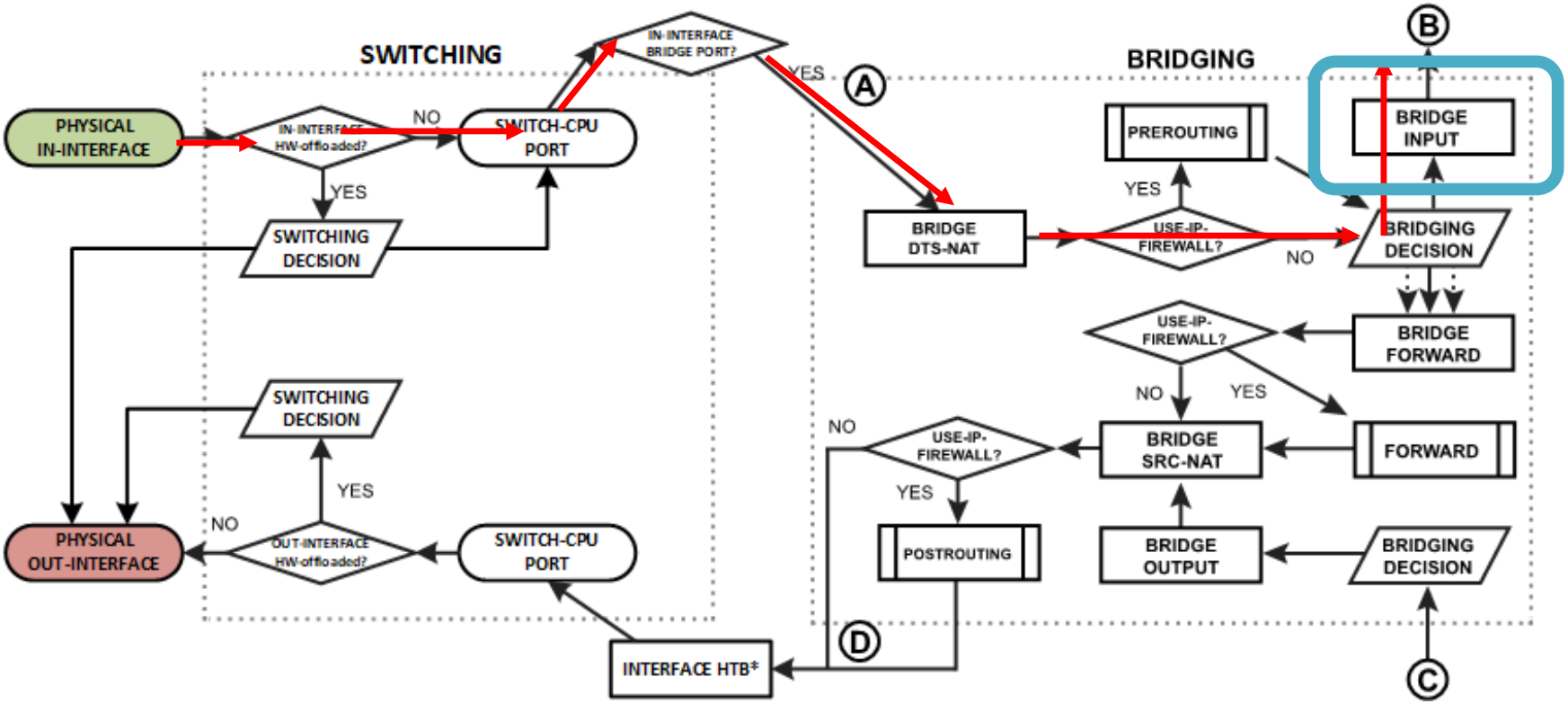

- Bridge Filter input will not apply here and web page loads.

# Bridge Filter Input – Webfig on Router

- Can we get to the webfig on router?
- PC1 (ether1) is HW offloaded



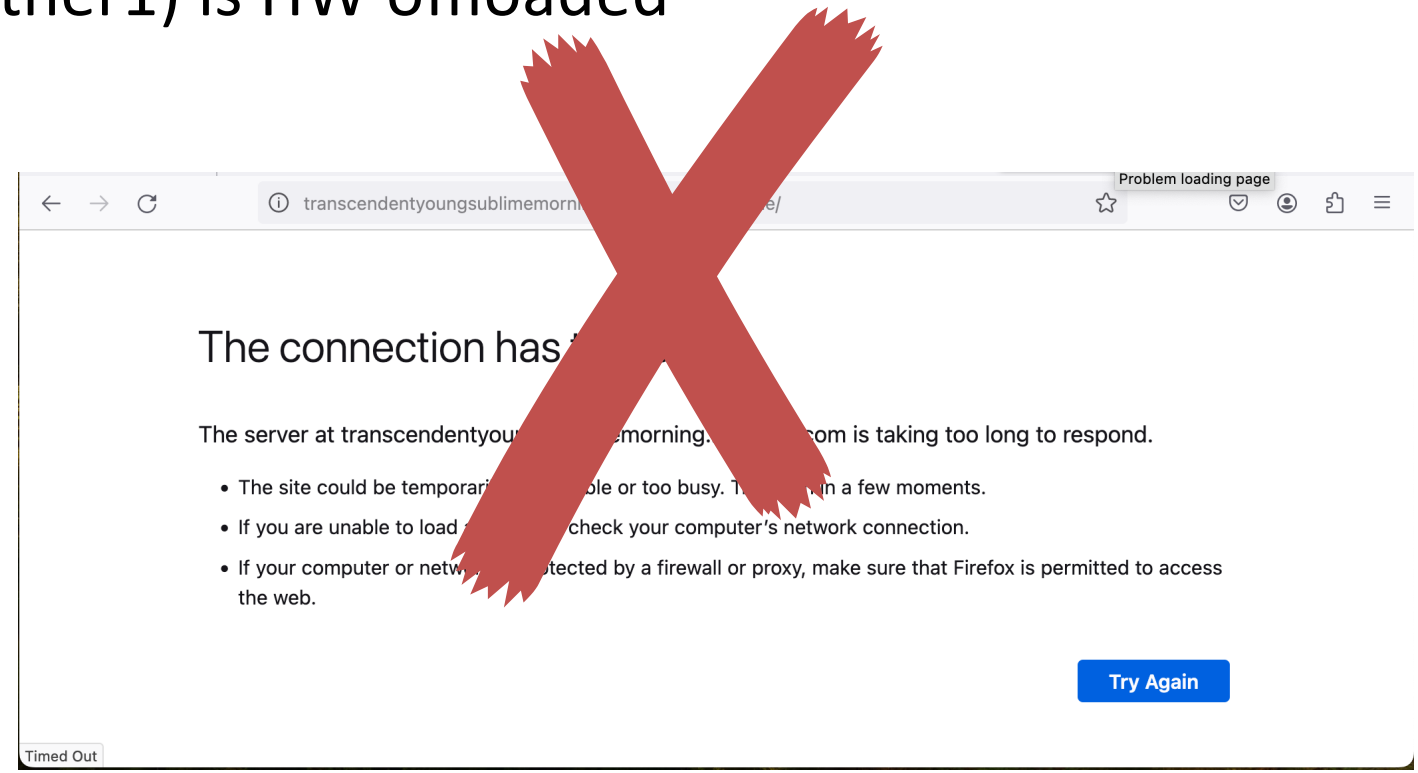
# Bridge Filter Input – Webfig on router



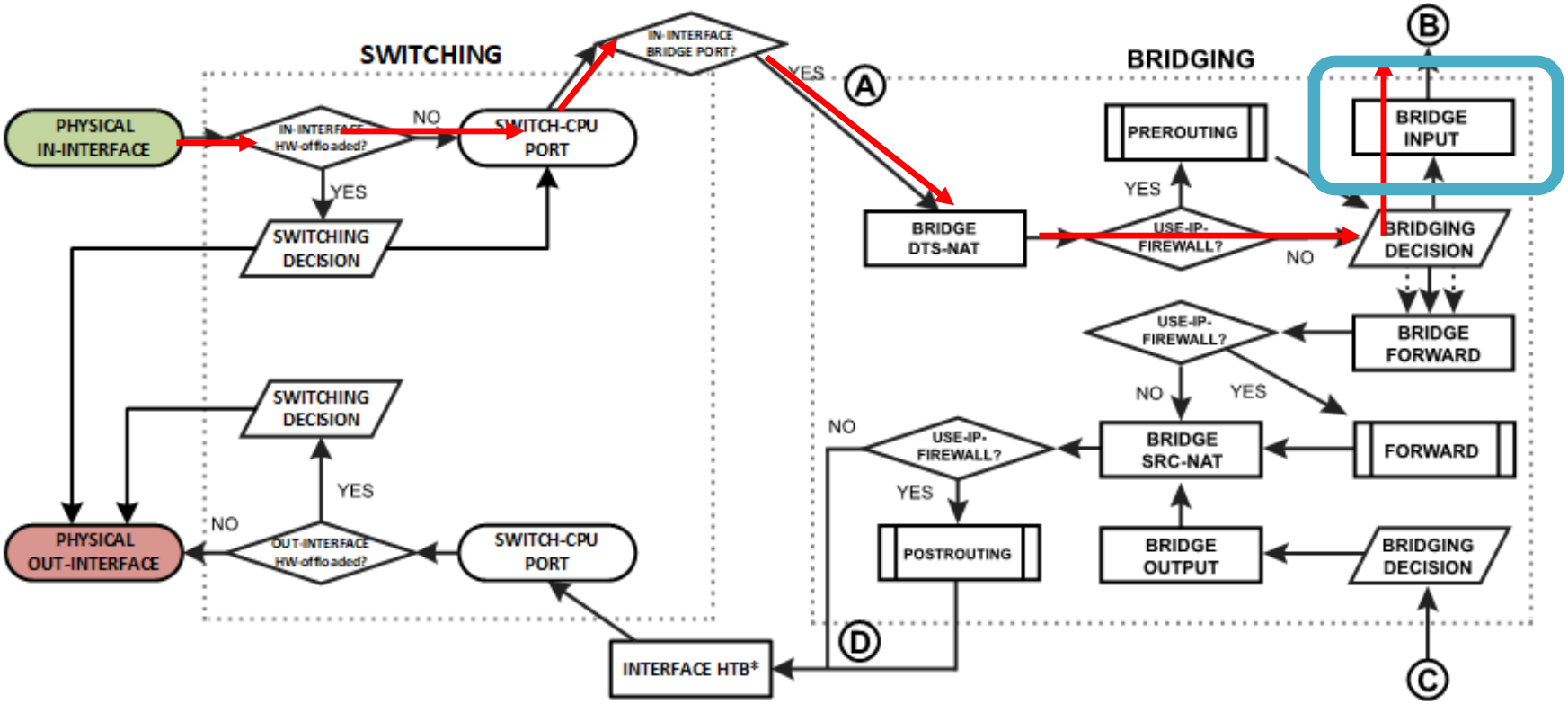

- Bridge Filter input rules does apply here as the traffic is bridge input and webfig is blocked.

# Switch Rules – http websites on internet

- Can we get to the http websites on the internet?
- PC1 (ether1) is HW offloaded



# Bridge Filter Input – http webpages on internet



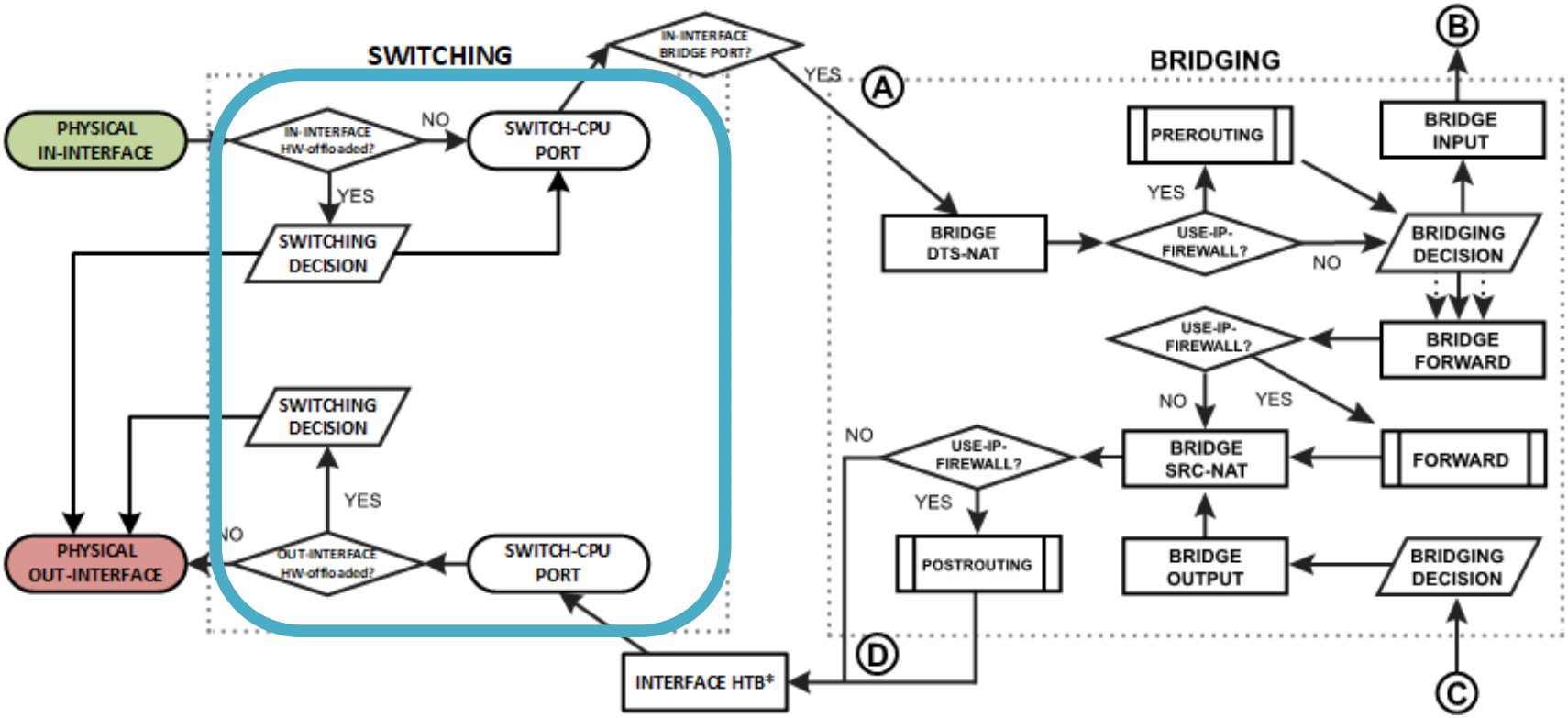

- Bridge Filter input rules does apply here as the traffic is bridge input and web page does not load.

# Switch Rules

- Switch filter rule to block TCP/80 with on port ether1
- What can PC1 get to?
  - Webserver on PC2?
  - Webserver on PC4?
  - Webfig on Router?
  - http web pages on the internet?

```
/interface ethernet switch rule
add dst-port=80 new-dst-ports="" ports=ether1 protocol=tcp switch=switch1
```

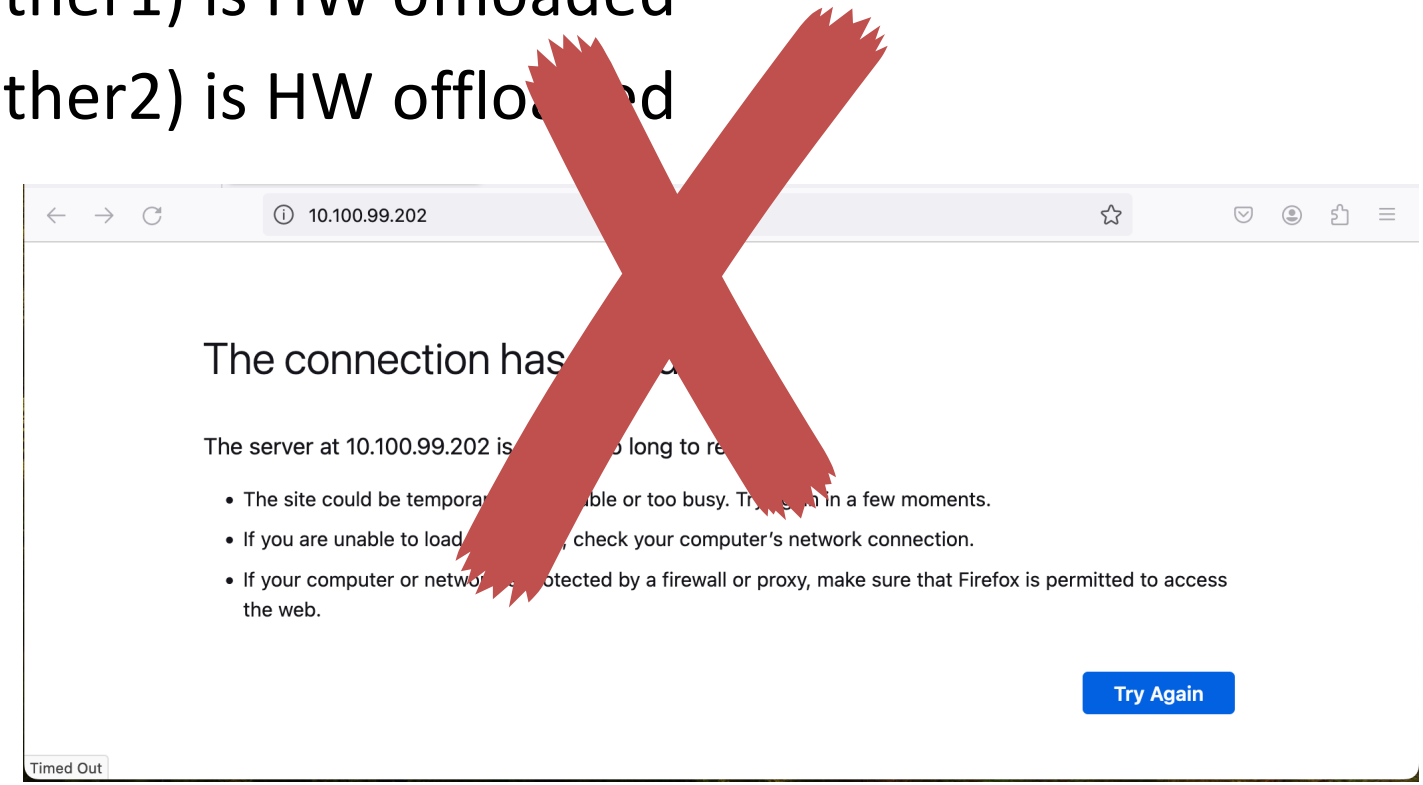
# Switch Rules (ACL)



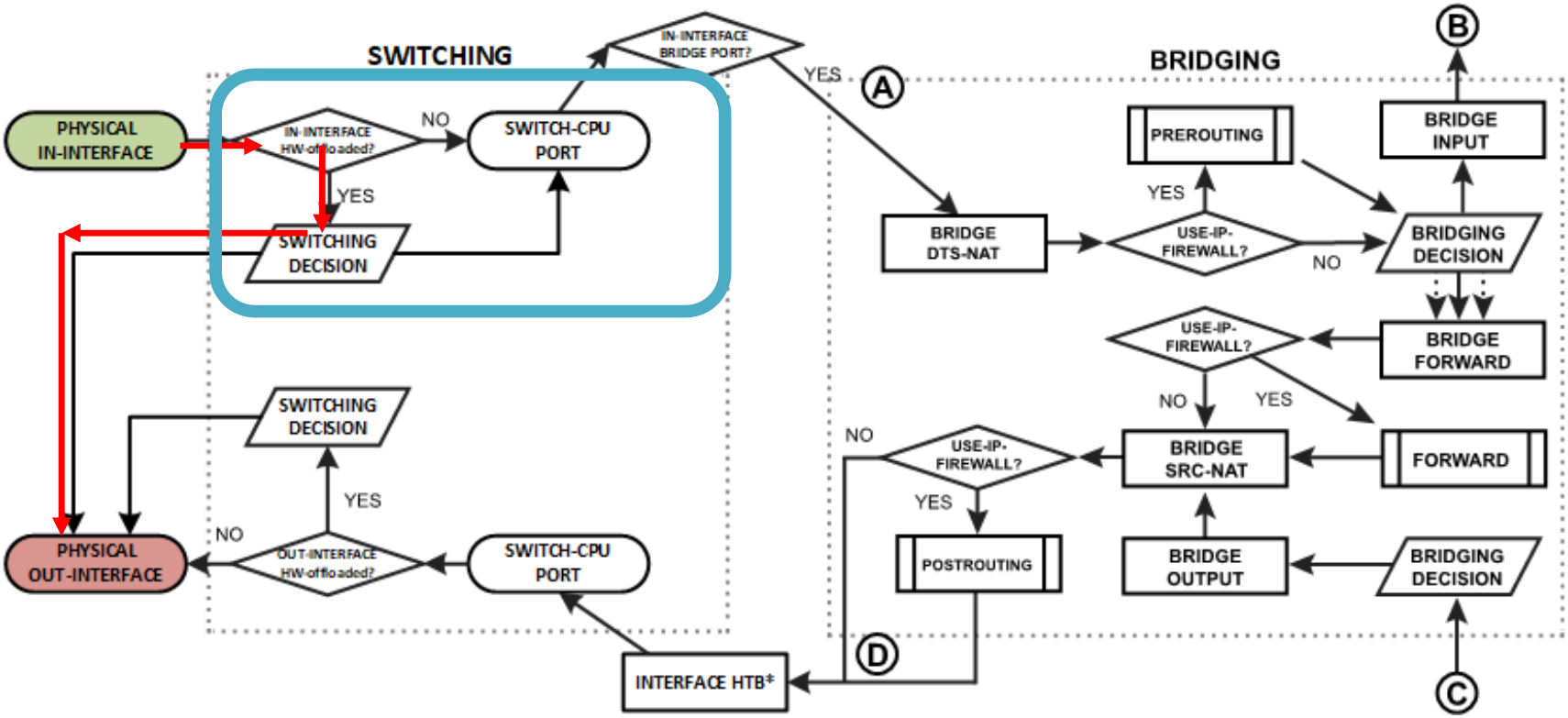

- Switch rules (ACL) apply in the switching function and on ingress.

# Switch rules – Webserver on PC2

- Can we get to the webserver on PC2 (ether2)?
- PC1 (ether1) is HW offloaded
- PC2 (ether2) is HW offlo d



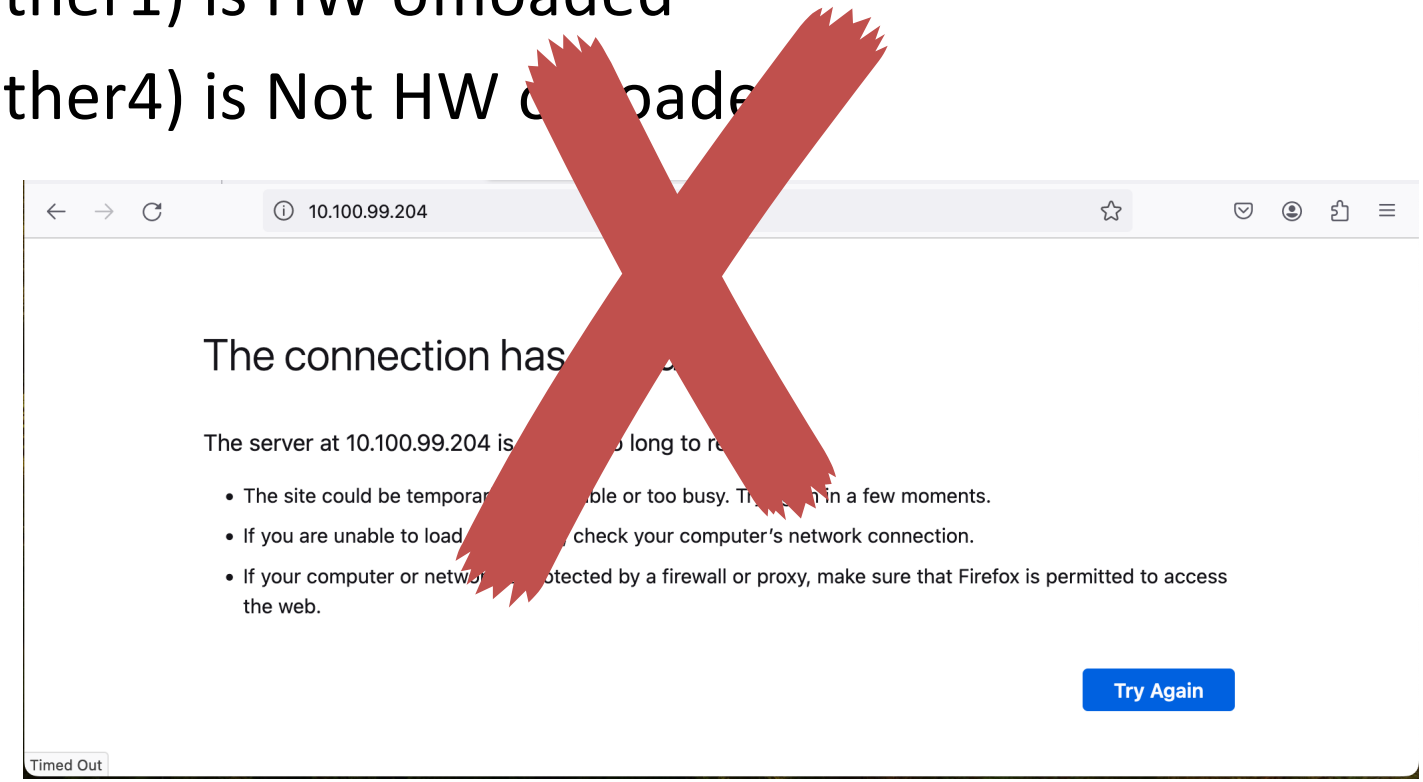
# Switch rules – Webserver on PC2



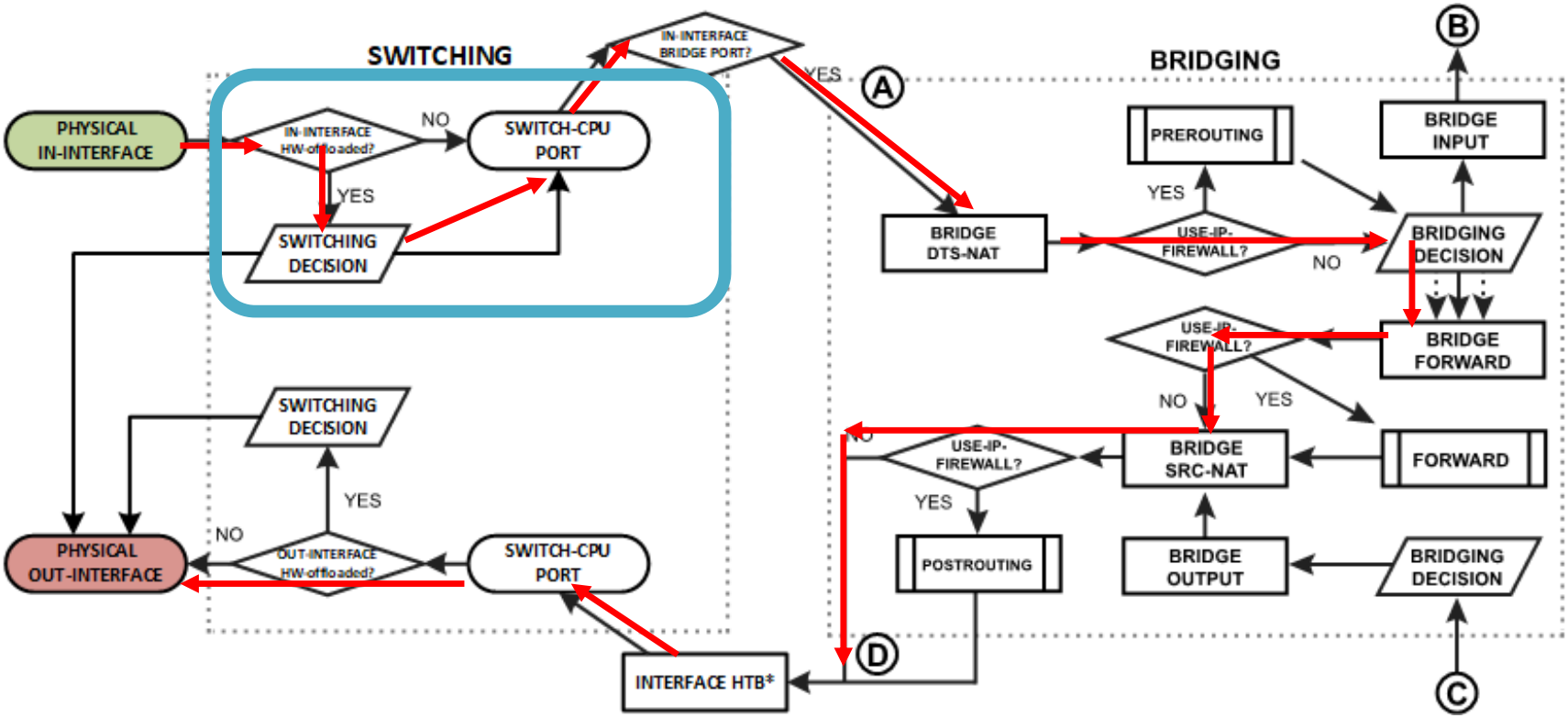

- Switch rules will apply here.

# Switch rules – Webserver on PC4

- Can we get to the webserver on PC4 (ether4)?
- PC1 (ether1) is HW offloaded
- PC4 (ether4) is Not HW offloaded



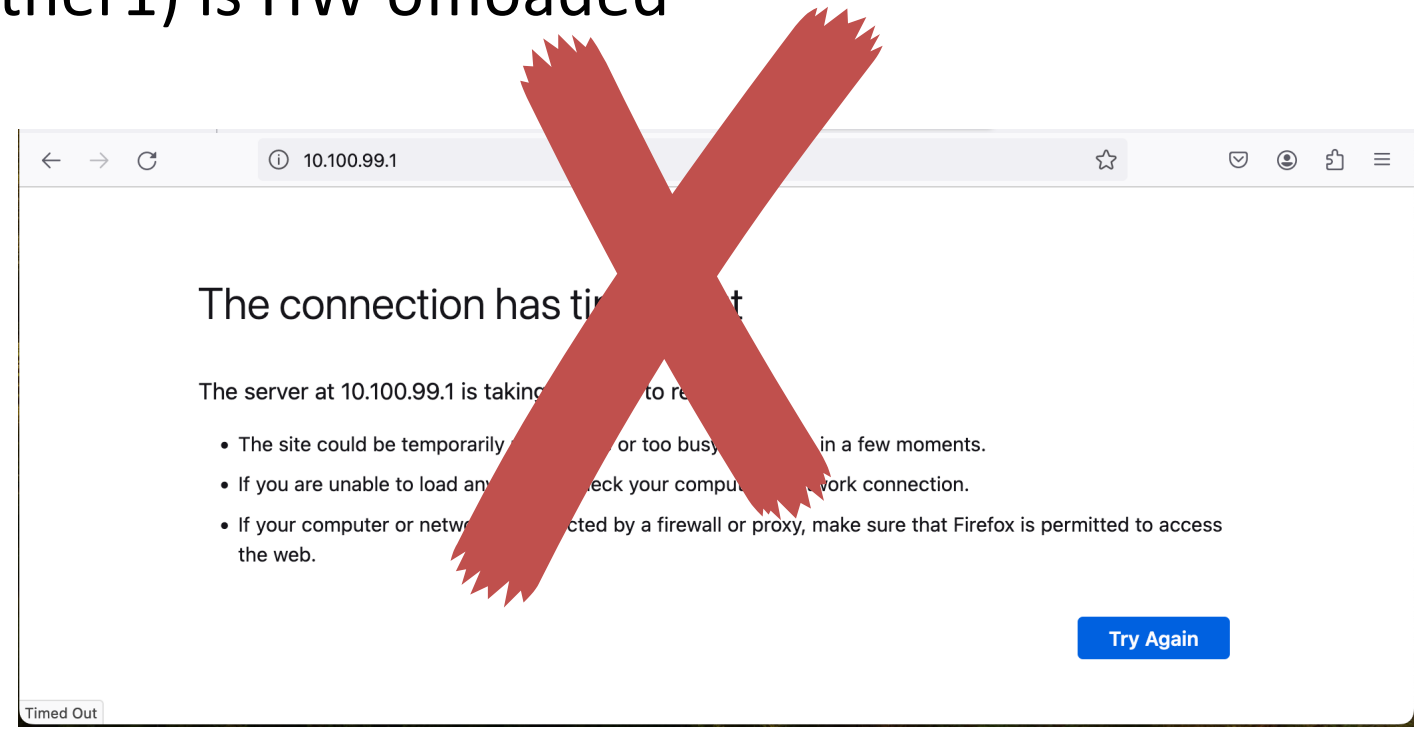
# Switch rules – Webserver on PC4



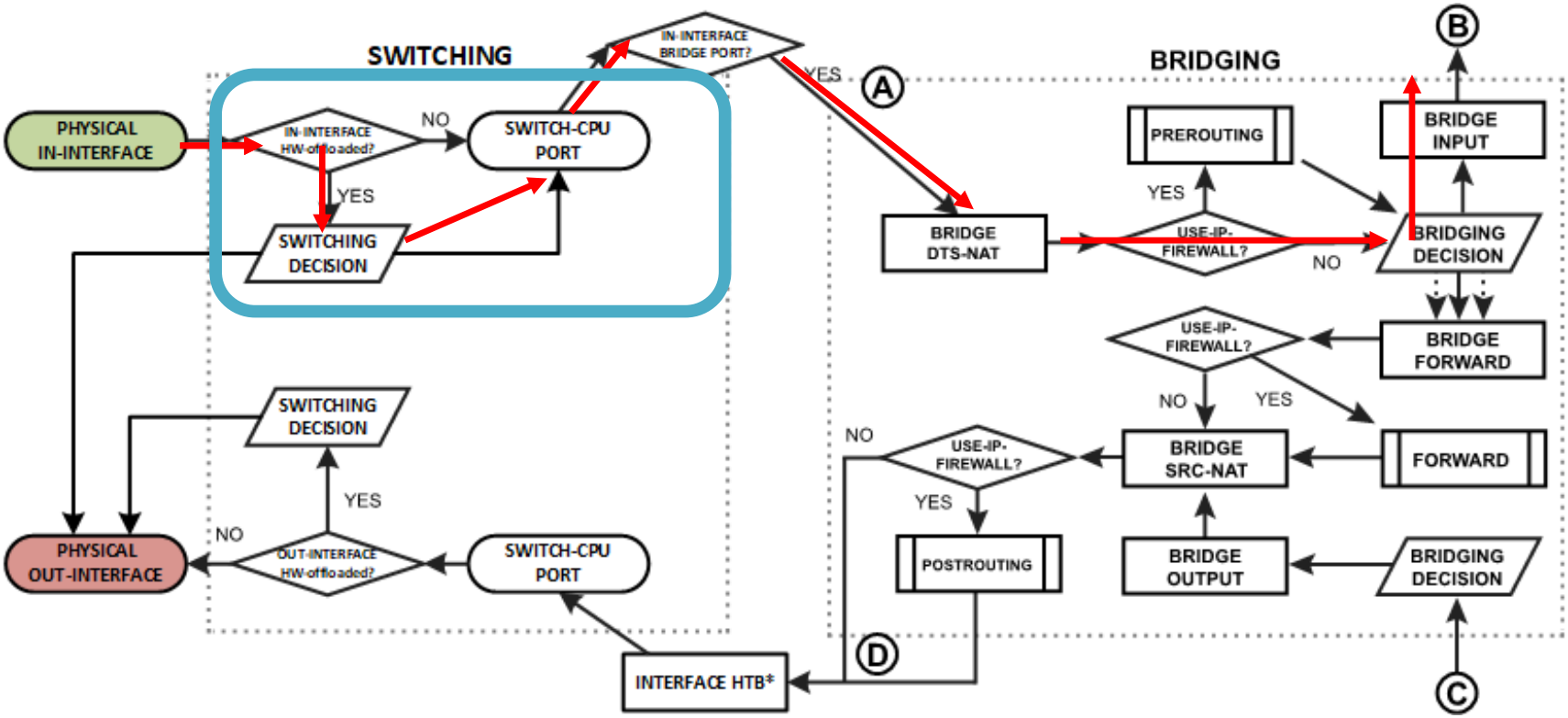

- Switch rules will apply here as the packet passes both switching function and bridge forward flow

# Switch Rules – Webfig on Router

- Can we get to the webfig on router?
- PC1 (ether1) is HW offloaded



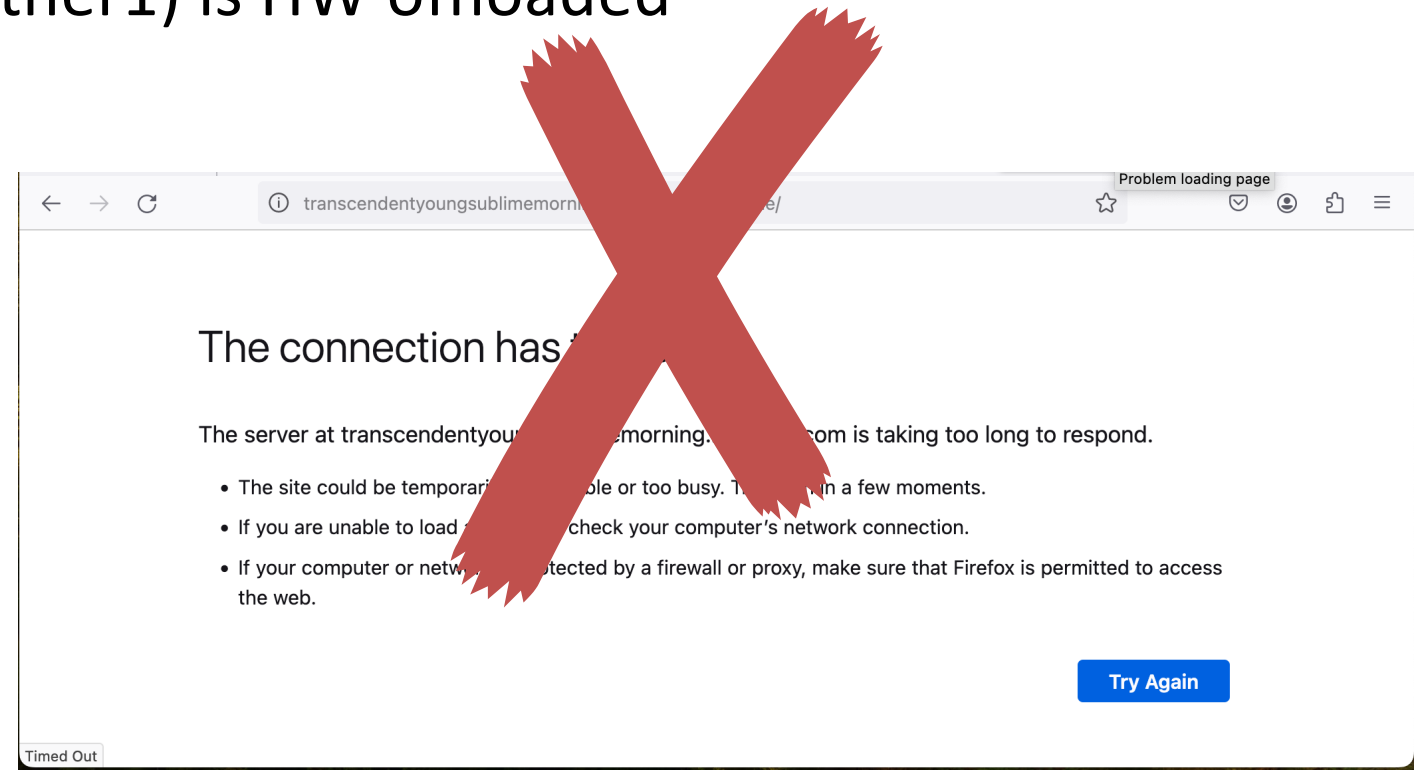
# Switch Rules – Webfig on Router



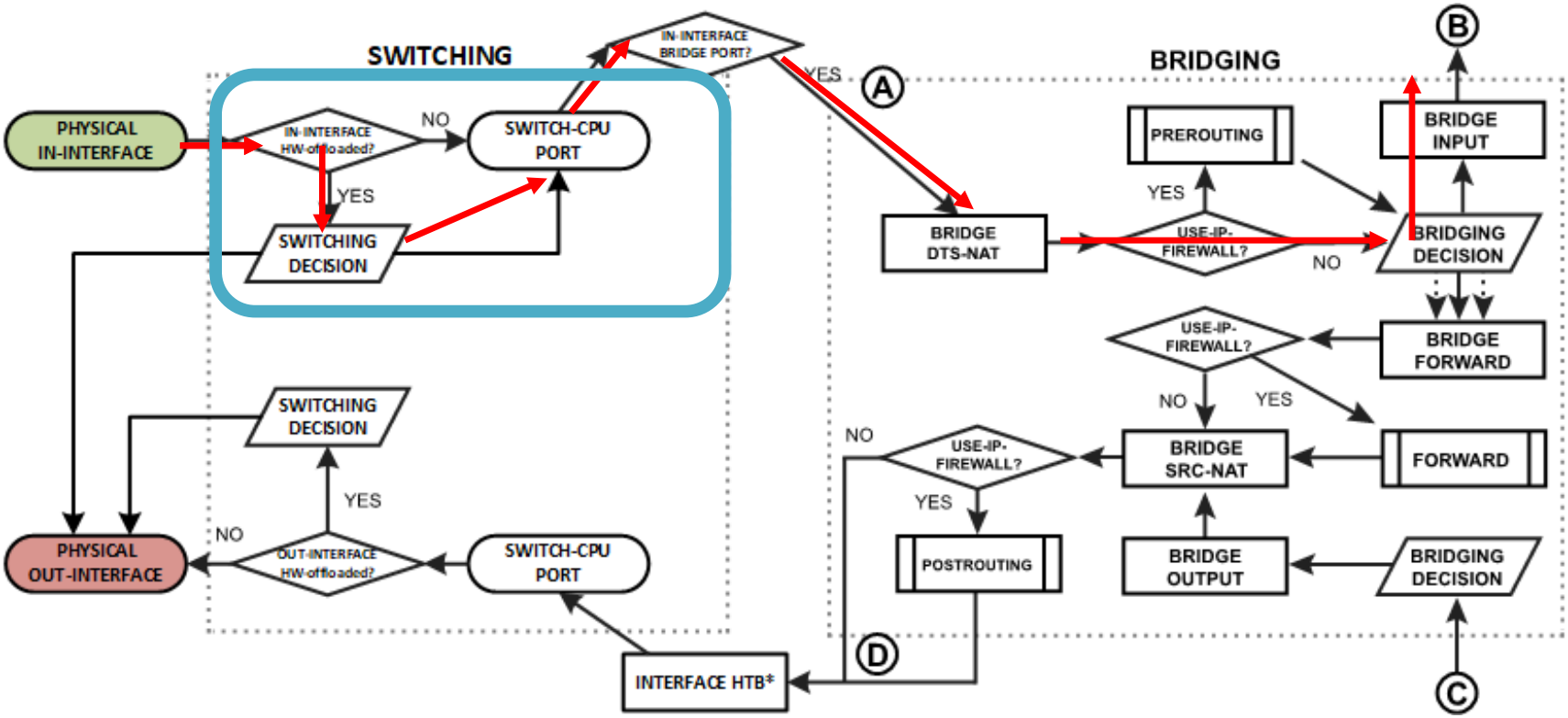

- Switch rules will apply here even though this is bridge input traffic and not forward.

# Switch Rules – http websites on internet

- Can we get to the http websites on the internet?
- PC1 (ether1) is HW offloaded

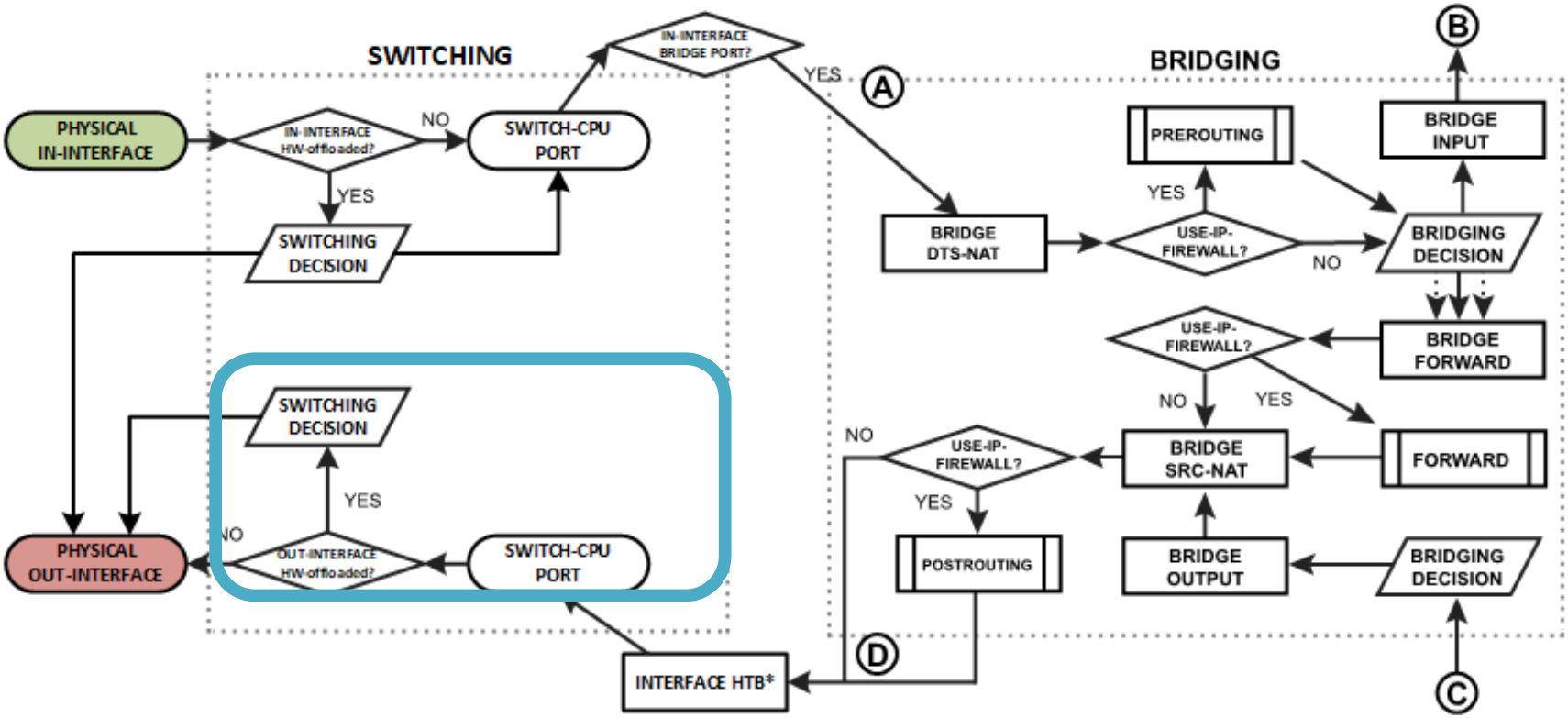


# Switch Rules – http websites on internet



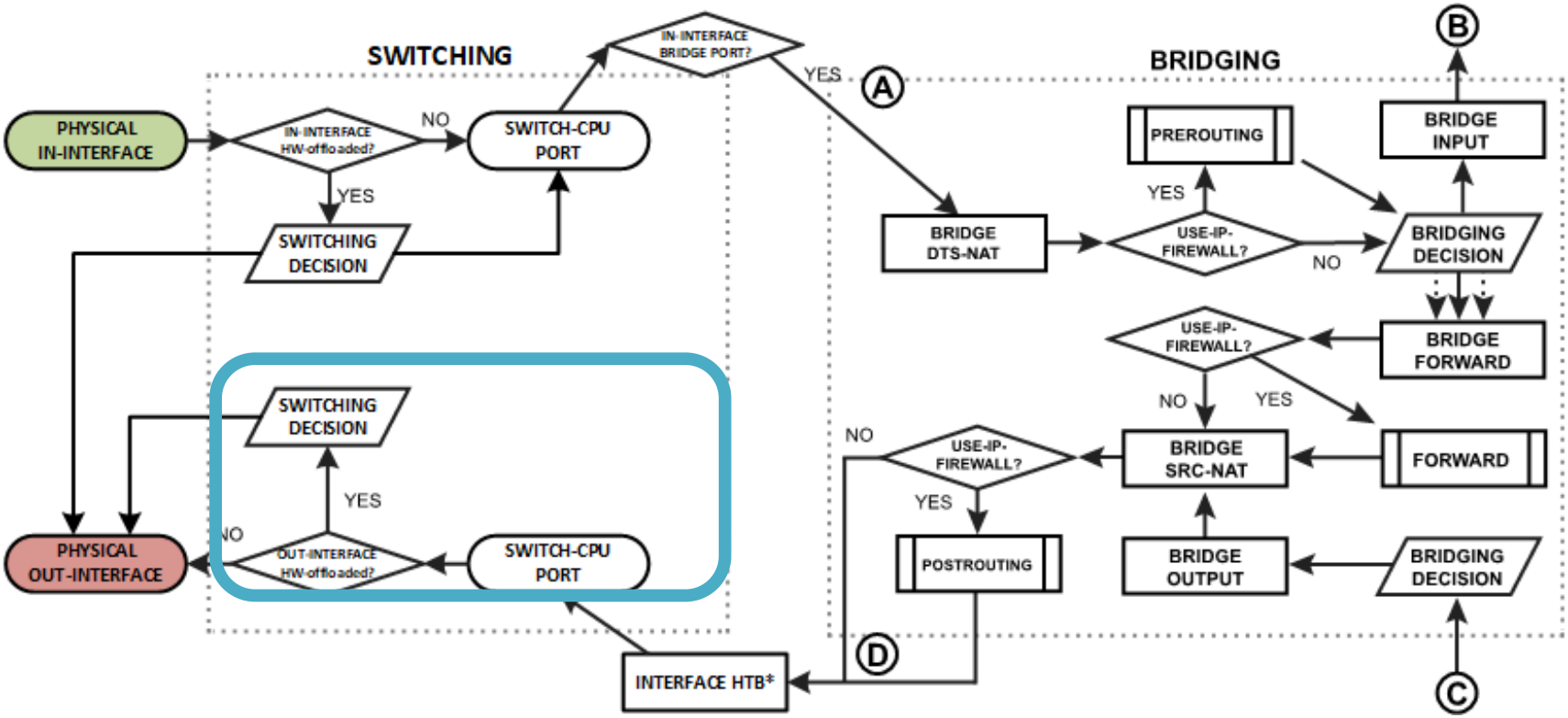

- Switch rules will apply here even though this is bridge input traffic and not forward.

# Switch Rules (ACL) switch-cpu




- Switch rules (ACL) apply in the switching function and on ingress. Switch-cpu port can be used in Switch Rules

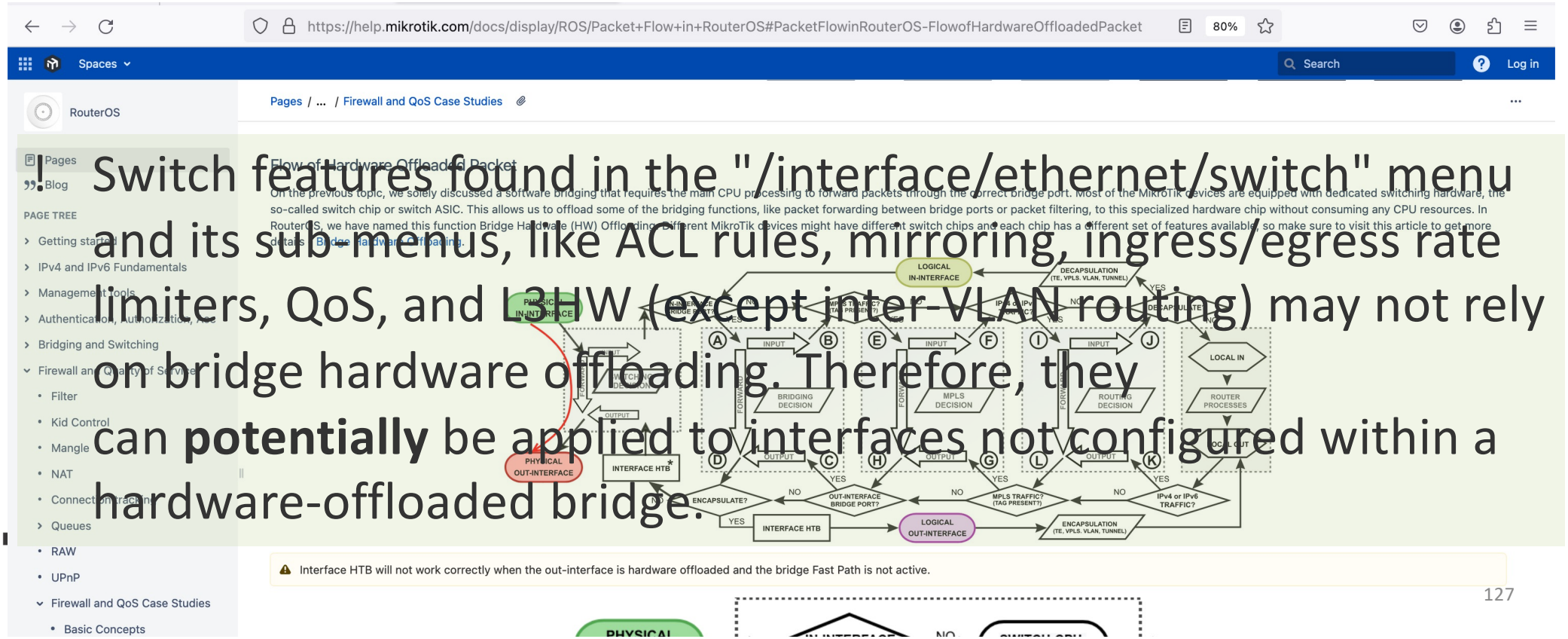
# Switch Rules (ACL) - switch-cpu




```
/interface ethernet switch rule
add dst-port=80 new-dst-ports="" ports=switch-cpu protocol=tcp switch=switch1
```

# Switch Rules – none HW Offloading interfaces

- Where are switch rules applied?
- https://help.mikrotik.com/docs/display/ROS/Packet+Flow+in+RouterOS#PacketFlowinRouterOS-FlowofHardwareOffloadedPacket




! Switch features found in the "/interface/ethernet/switch" menu and its sub-menus, like ACL rules, mirroring, ingress/egress rate limiters, QoS, and L3HW (except inter-VLAN routing) may not rely on bridge hardware offloading. Therefore, they can **potentially** be applied to interfaces not configured within a hardware-offloaded bridge.

# More Packet flow

- There are so many more combinations we could have looked at here.
- Want to know more about Layer2 Packet flow and the CRS3xx/6xx devices?

MTCSWE

# MTCSWE Course

- www.linitx.com/training

https://linitx.com/training


**LinITX MikroTik CWE151024 MTCSWE Training Course - 15-17 October 2024 - Sheffield AMP**

MikroTik Certified Switching Engineer (MTCSWE) - Sheffield AMPDate: 15th-17th October 2024Location: Sheffield AMP The Course The MTCSWE course content was led by Jono Thompson in co-operation with MikroTik. The course is based on the latest v6.x RouterOS available...

8 In Stock

# Thank you for Listening

# References

- https://help.mikrotik.com/docs/display/ROS/Packet+Flow+in+RouterOS