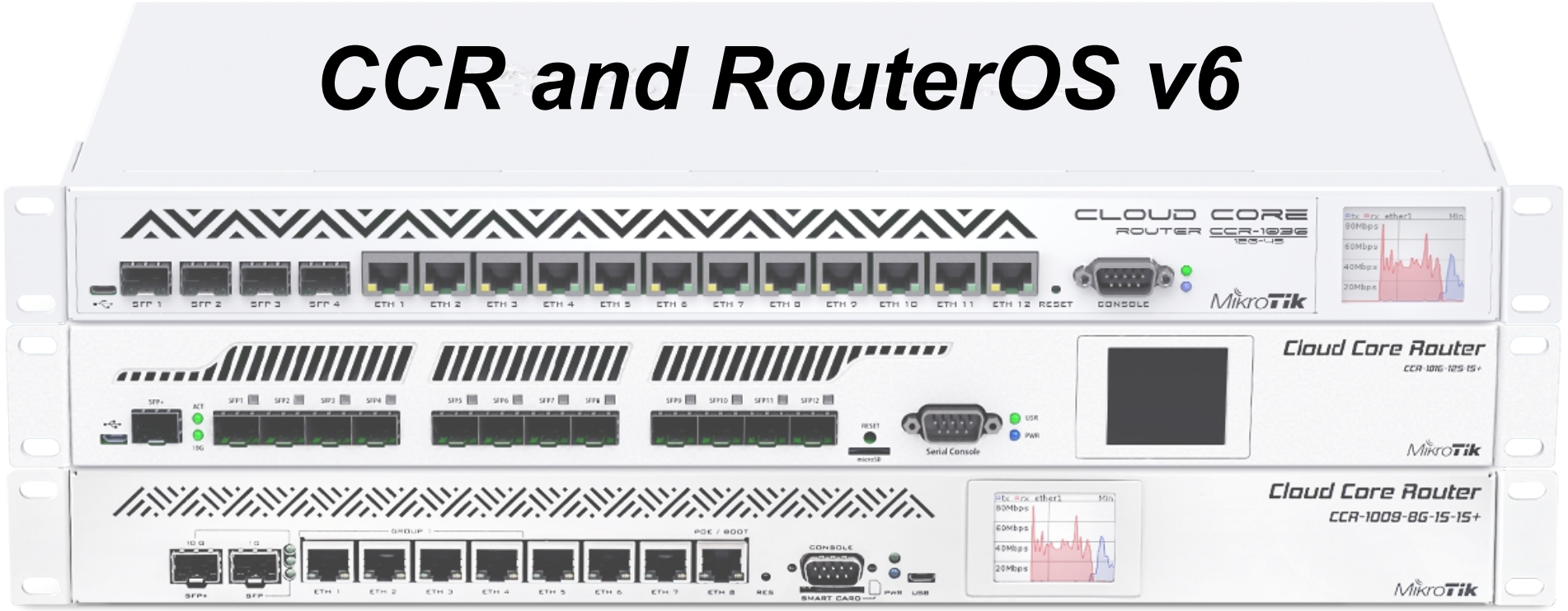
Status update

CCR and RouterOS v6



MUM Mexico and USA, 2014

Current Selection of CCRs

CCR1009-8G-1S

1U rackmount, 8x Gigabit Ethernet, 1xSFP cage, 9 cores x 1.2GHz CPU, 1GB RAM, RouterOS L6

$425.00

CCR1009-8G-1S-1S+

1U rackmount, 8x Gigabit Ethernet, 1xSFP cage, 1xSFP+ cage, 9 cores x 1.2GHz CPU, 2GB RAM, LCD panel, Dual Power supplies, SmartCard slot, RouterOS L6

$495.00

CCR1016-12G

1U rackmount, 12x Gigabit Ethernet, LCD 16 cores x 1.2GHz CPU, 2GB RAM, 17.8mpps fastpath, Up to 12Gbit/s throughput, RouterOS L6

$645.00

CCR1016-12S-1S+

1U rackmount, 12xSFP cage, 1xSFP+ cage, 16 cores x 1.2GHz CPU, 2GB RAM, LCD panel, Dual Power supplies, RouterOS L6

$745.00

CCR1036-12G-4S

1U rackmount, 12x Gigabit Ethernet, 4xSFP cages, LCD 36 cores x 1.2GHz CPU, 4GB RAM, 24 mpps fastpath, Up to 16Gbit/s throughput, RouterOS L6

$995.00

CCR1036-8G-2S+

1U rackmount, 8x Gigabit Ethernet, 2xSFP+ cages, LCD 36 cores x 1.2GHz CPU, 4GB RAM, 41.5mpps fastpath, Up to 28Gbit/s throughput, RouterOS L6

$1,095.00

CCR1036-12G-4S-EM

1U rackmount, 12x Gigabit Ethernet, 4xSFP cages, LCD 36 cores x 1.2GHz CPU, 16GB RAM, 24 mpps fastpath, Up to 16Gbit/s throughput, RouterOS L6

$1,195.00

CCR1036-8G-2S+EM

1U rackmount, 8x Gigabit Ethernet, 2xSFP+ cages, LCD 36 cores x 1.2GHz CPU, 16GB RAM, 41.5mpps fastpath, Up to 28Gbit/s throughput, RouterOS L6

$1,295.00

TILE Architecture

- RouterOS supports TILE-Gx processors with 9, 16, 36, or 72 identical processor cores (tiles) **ONLY** on CCR devices
- Tile architecture provide hardware accelerated AES encryption, automatic RPS and IRQ management
- Complete set of DDR3 memory and I/O controllers integrated into CPU.
- Runs on the first ever 64-bit RouterOS v6 (tile)

Multi-Core Packet Processing

- After Ethernet frame is successfully received by the Interface driver and each frame gets assigned by the Linux Kernel to a specific core
  - Re-assigning frames to a different core is very “expensive” process and should be avoided as much as possible
  - Processing frames on different cores might take different amount of time – packet order might change during the parallel processing
  - Some processes/features requires a set of frames to be under control of the same core

Fast Path

- Fast Path allows the router to forward packets without additional processing in the Linux Kernel. It reveals hardware's true potential.
- Fast Path requirements
  - Fast Path should be allowed in the configuration
  - The interface driver must have support
  - Specific configuration conditions
- Currently RouterOS has Fast Path handlers for: ipv4 routing, traffic generator, mpls and bridge
- More handlers will be added in future

Fast Path Throughput

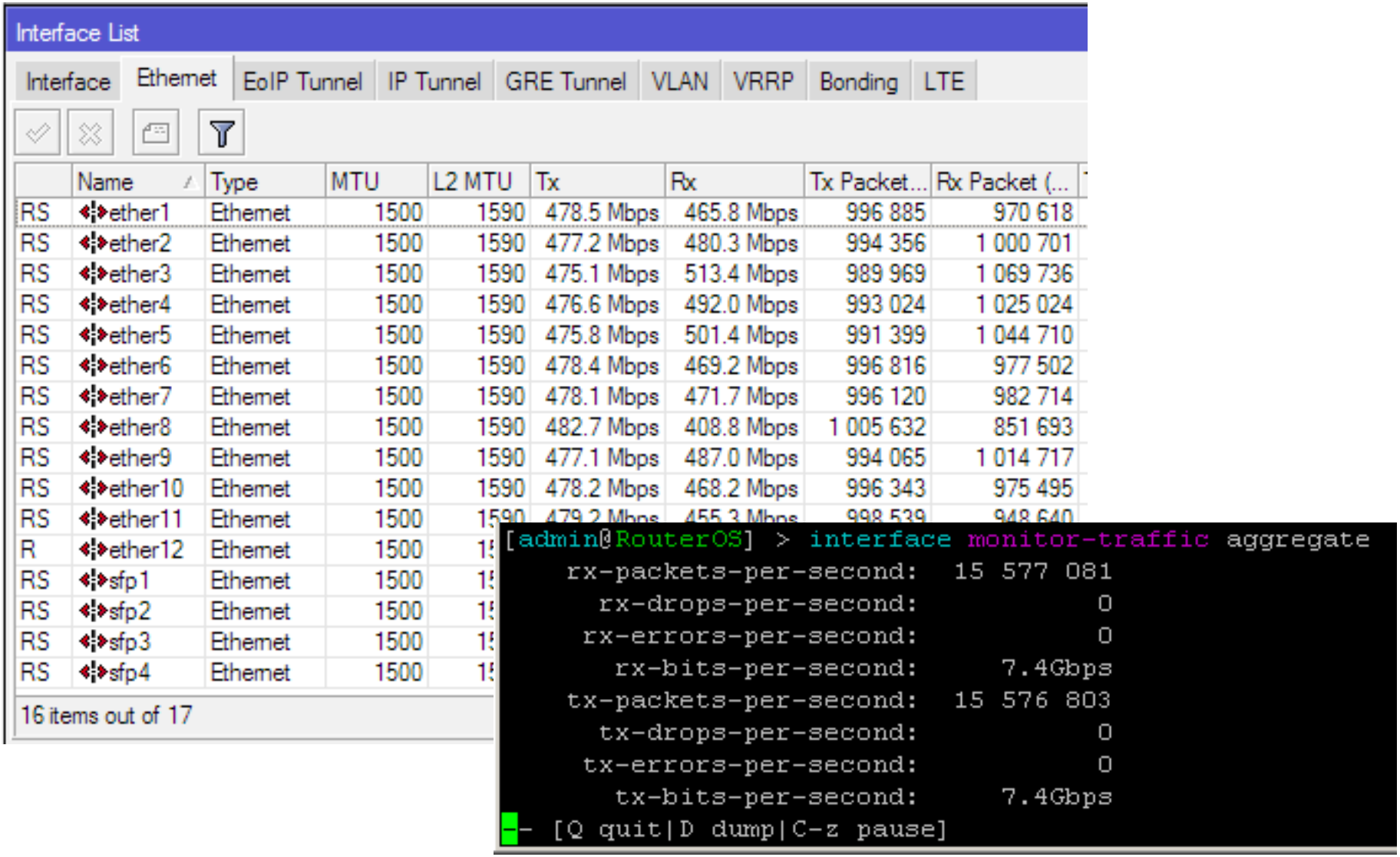
CCR1072-1G-8S+ (1200Mhz)	RouterOS v6.19rc6					
Mode	**64 byte**		**512 byte**		**1518 byte**	
	kpps	**Mbps**	**kpps**	**Mbps**	**kpps**	**Mbps**
Bridging	69,777.7	45,774.2	18,800.0	79,712.0	6,502.0	79,896.6
Routing	55,641.0	36,500.5	18,800.0	79,712.0	6,502.0	79,896.6
CCR1036-8G-2S+ (1200Mhz)						
Bridging	39,764.6	26,085.6	6,579.0	27,895.0	2,274.0	27,942.9
Routing	34,086.1	22,360.5	6,579.0	27,895.0	2,274.0	27,942.9
CCR1016-12S-1S+ (1200Mhz)						
Bridging	15,244.5	10,000.4	4,695.0	19,906.8	1,624.0	19,955.7
Routing	13,186.3	8,650.2	4,695.0	19,906.8	1,624.0	19,955.7
CCR1009-8G-1S-1S+ (1200Mhz)						
Bridging	10,491.1	6,882.2	2,817.0	11,944.1	974.0	11,968.5
Routing	8,217.1	5,390.4	2,817.0	11,944.1	974.0	11,968.5

- Underlined results – max wire-speed reached

Traffic Generator Tool

- The Traffic Generator is a bandwidth-tool evolution
- The Traffic Generator can:
  - Determine transfer rates, packet loss
  - Detect out-of-order packets
  - Collect latency and jitter values
  - Inject and replay *.pcap file
  - Working on TCP protocol emulation
- “Quick” mode
- Full Winbox support (coming soon)

Throughput in millions pps



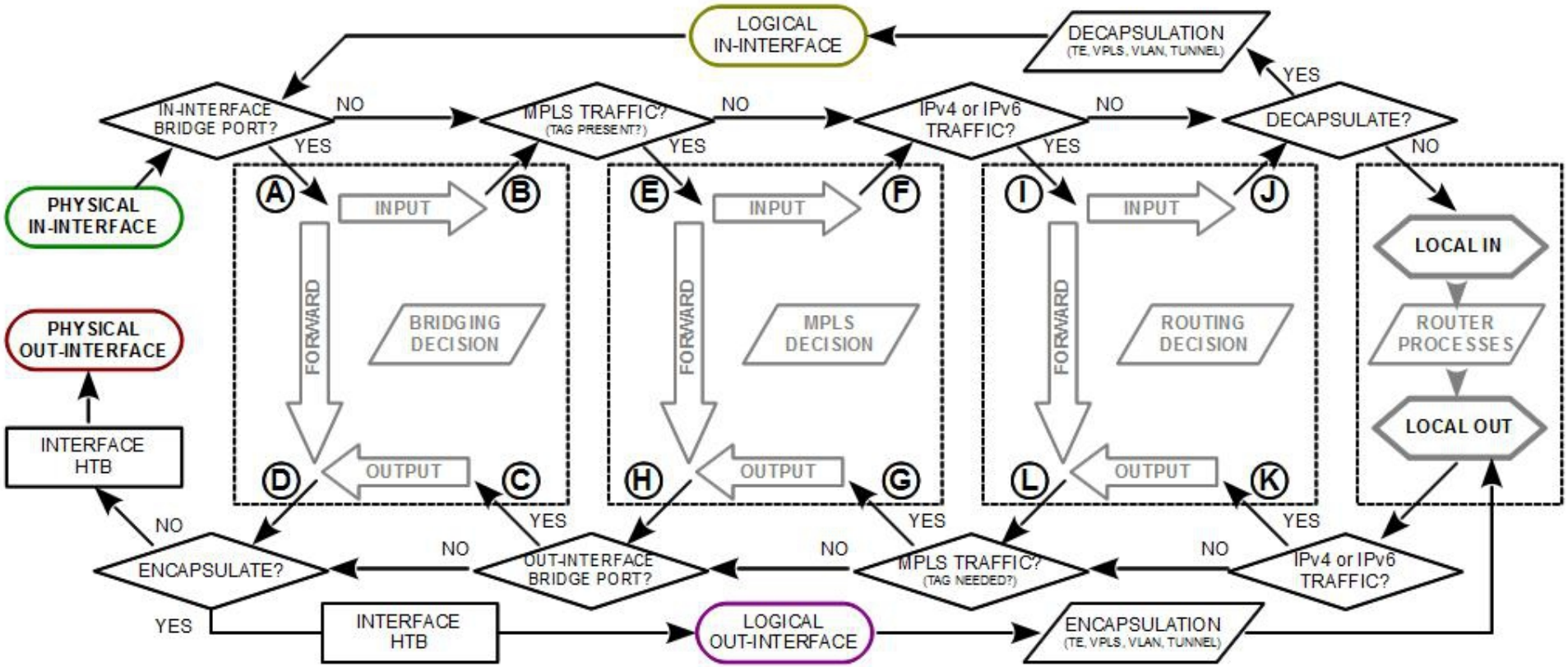
RouterOS on CCR

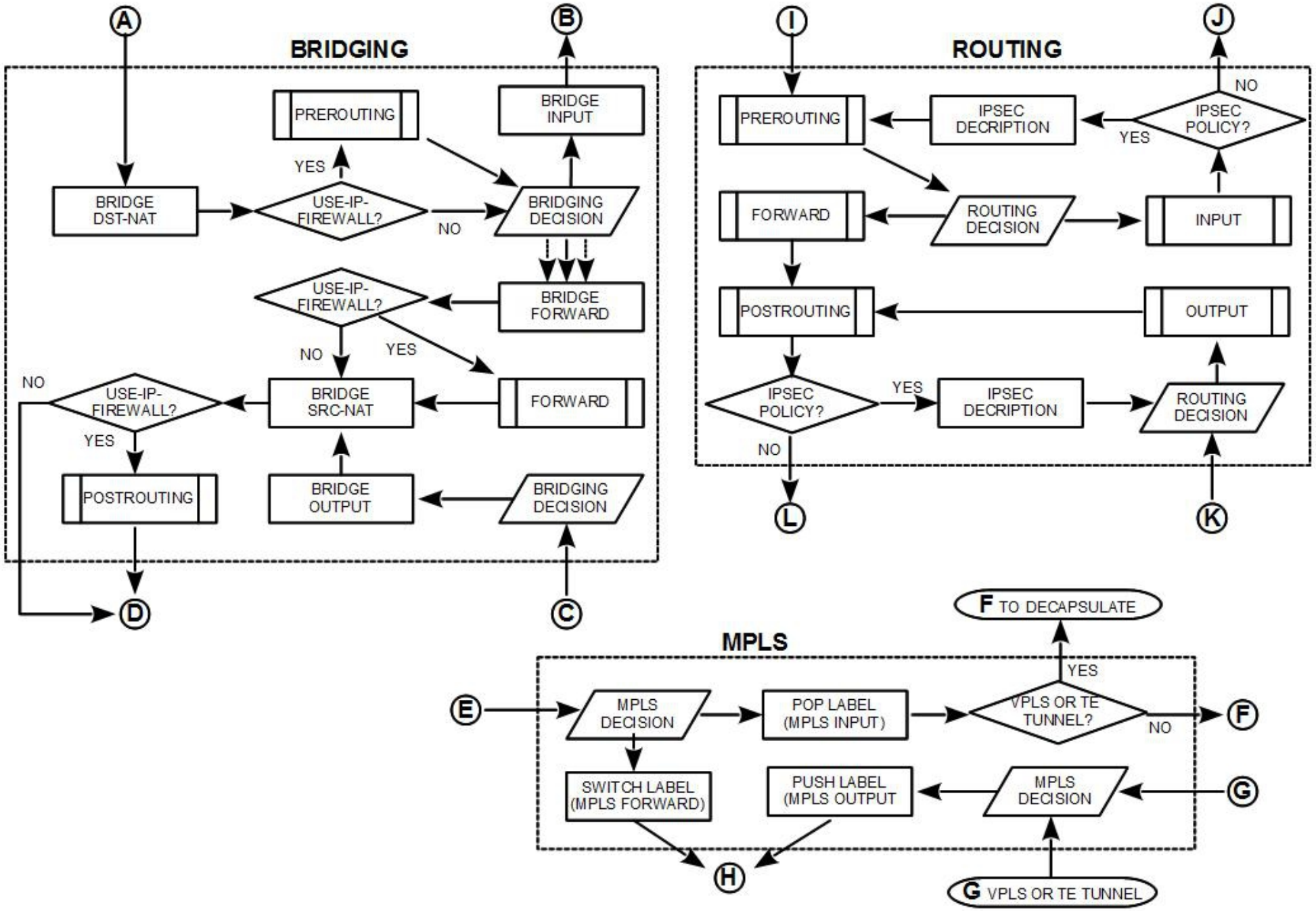
- Previous slides indicate that hardware is very, very fast.
- What about when we start using RouterOS features?
- Answer:

CCR1036-8G-2S+ (1200Mhz) **All port test** **RouterOS v6.19rc6**

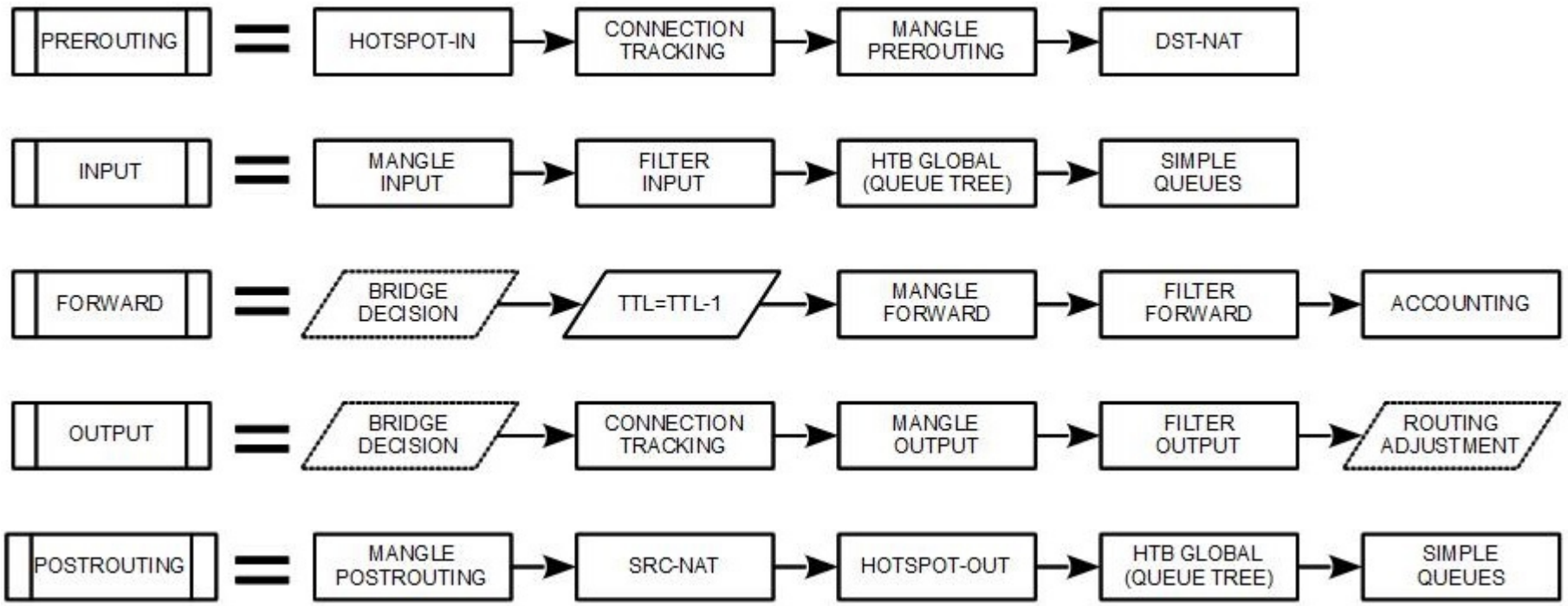
Mode	Configuration	64 byte		512 byte		1518 byte	
		kpps	Mbps	kpps	Mbps	kpps	Mbps
Bridging	none (fast path)	39,764.6	26,085.6	6,579.0	27,895.0	2,274.0	27,942.9
Bridging	25 bridge filter rules	5,704.8	3,742.3	5,451.4	23,113.9	2,274.0	27,942.9
Routing	none (fast path)	34,086.1	22,360.5	6,579.0	27,895.0	2,274.0	27,942.9
Routing	25 simple queues	8,353.1	5,479.6	5,243.2	22,231.2	2,274.0	27,942.9
Routing	25 ip filter rules	3,481.8	2,284.1	3,392.9	14,385.9	1,956.7	24,043.9

MikroTik RouterOS Packet Flow Diagram for version 6.x

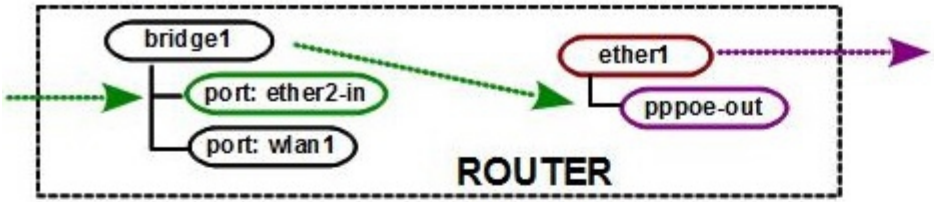



A
B
BRIDGING
PREROUTING
BRIDGE INPUT
BRIDGE DST-NAT
USE-IP-FIREWALL?
YES
NO
BRIDGING DECISION
USE-IP-FIREWALL?
BRIDGE FORWARD
NO
YES
NO
USE-IP-FIREWALL?
BRIDGE SRC-NAT
FORWARD
YES
POSTROUTING
BRIDGE OUTPUT
BRIDGING DECISION
D
C
I
J
ROUTING
PREROUTING
IPSEC DECRIPTION
NO
IPSEC POLICY?
YES
FORWARD
ROUTING DECISION
INPUT
POSTROUTING
OUTPUT
IPSEC POLICY?
YES
IPSEC DECRIPTION
ROUTING DECISION
NO
L
K
F TO DECAPSULATE
MPLS
YES
E
MPLS DECISION
POP LABEL (MPLS INPUT)
VPLS OR TE TUNNEL?
NO
F
SWITCH LABEL (MPLS FORWARD)
PUSH LABEL (MPLS OUTPUT
MPLS DECISION
G
H
G VPLS OR TE TUNNEL

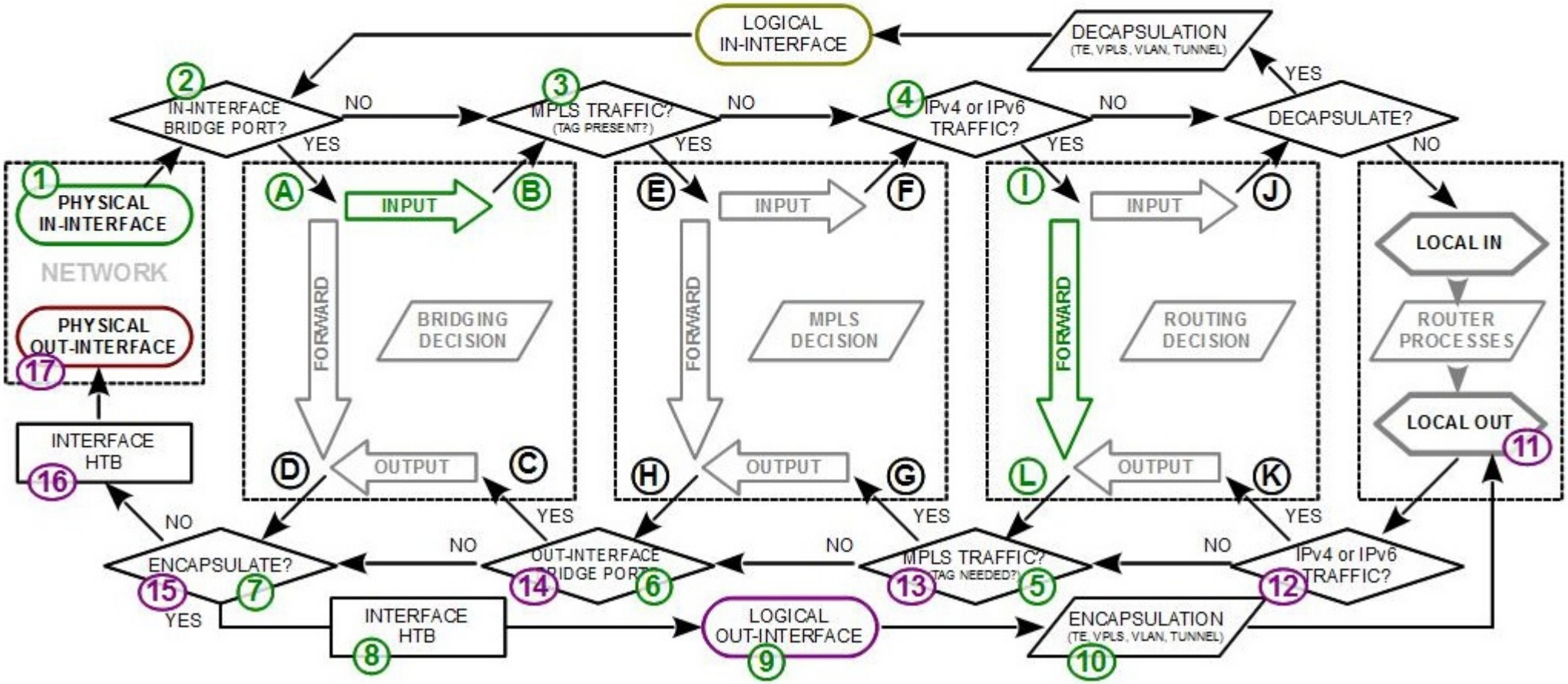
Yes, still - Packet Flow Diagram (page 3)



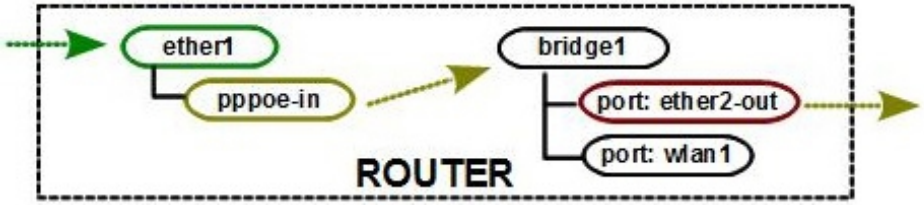
Packet Flow Scenario:



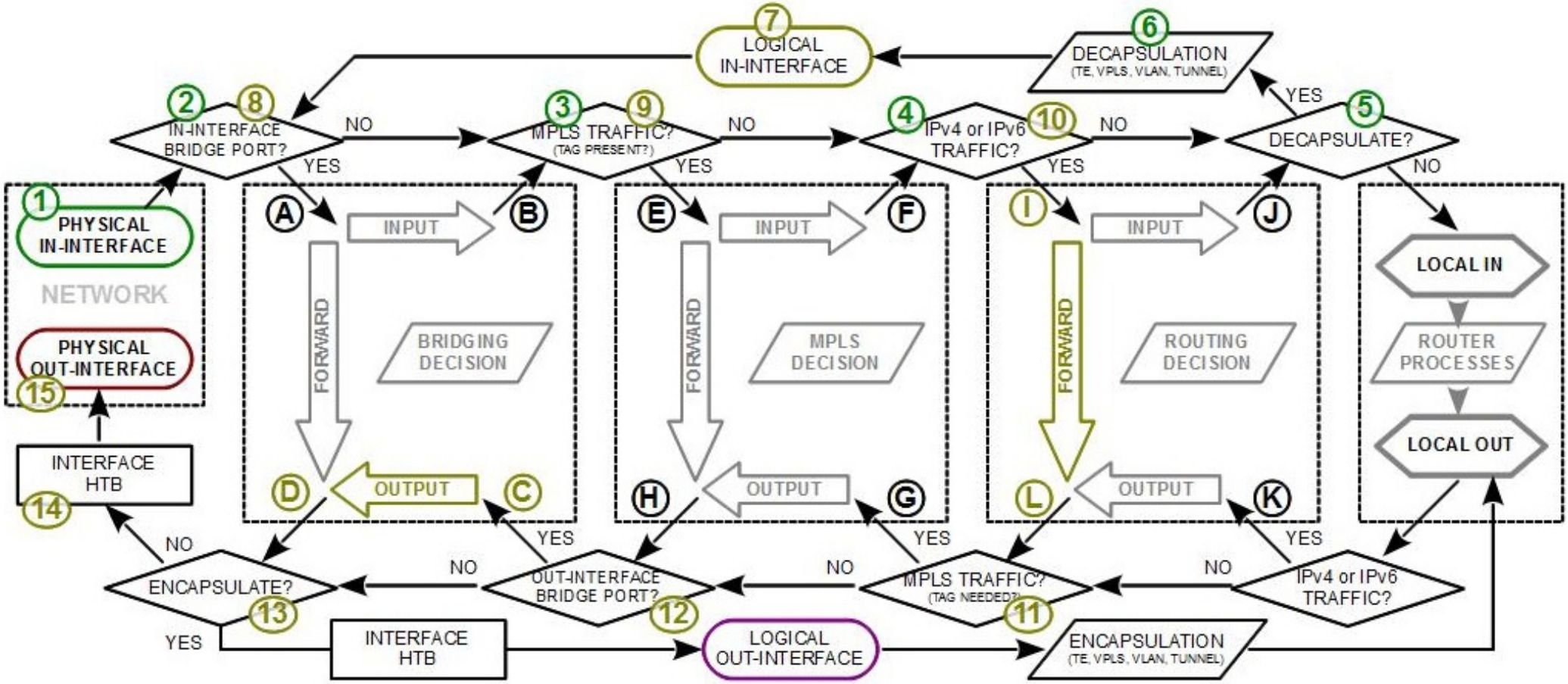
This Scenario in Packet Flow Diagram:



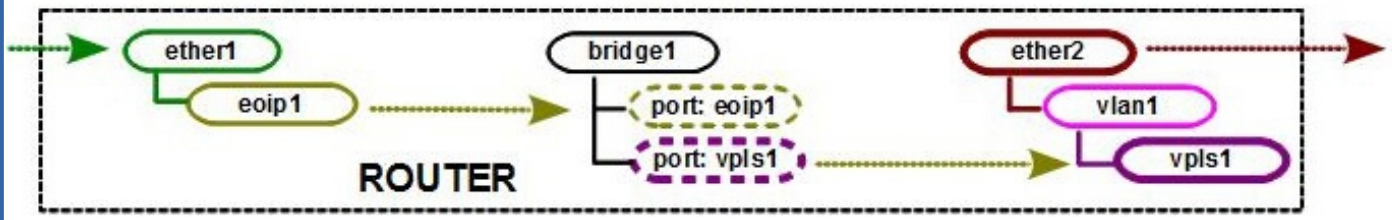
Packet Flow Scenario:



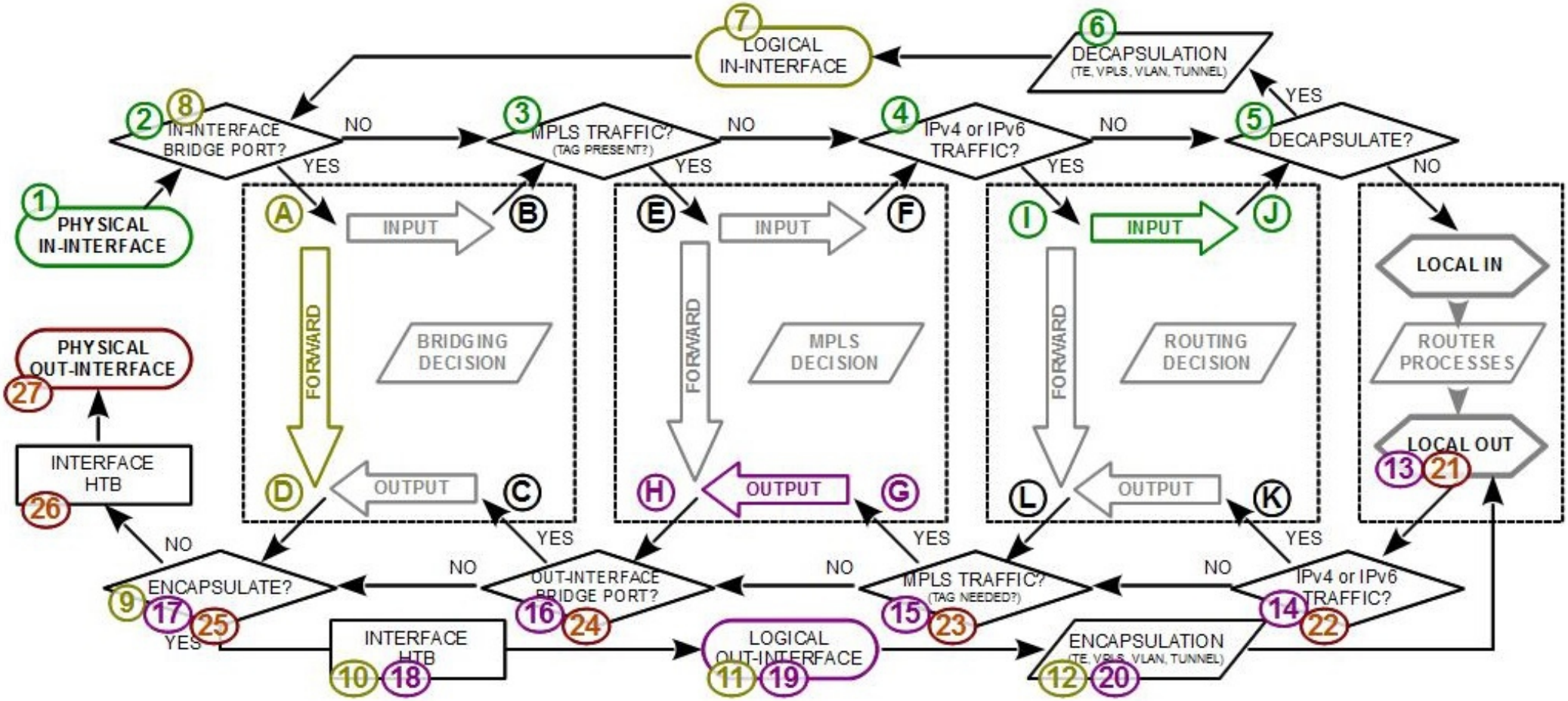
This Scenario in Packet Flow Diagram:



Packet Flow Scenario:



This Scenario in Packet Flow Diagram:



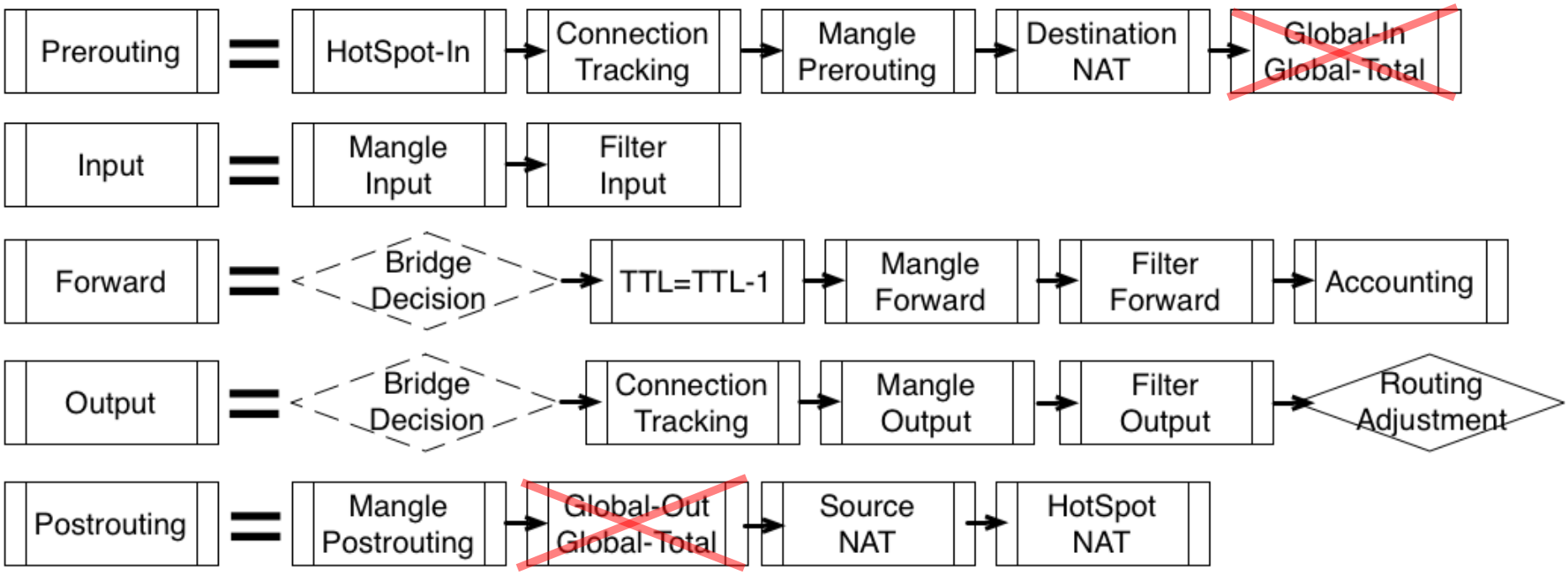
Queues and Multi-Core Processing

- Packets spend the greatest part of the processing time waiting in queues.
- In order not to waste CPU core cycles on waiting, the current core will just leave packet in the queue.
- The packet will be taken out of the queue by a random CPU core, that works on that queue at the time.
- In short: queues shuffle packet assignments to CPU cores.

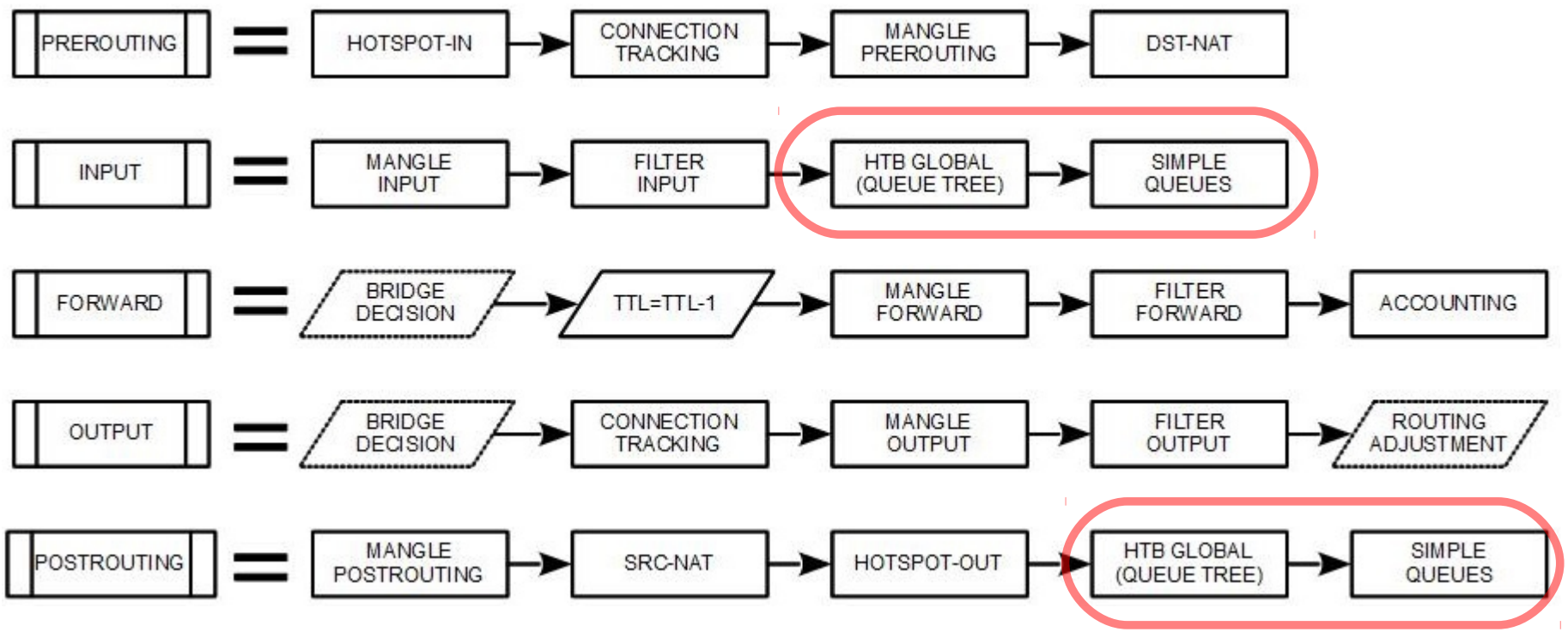
Packet Flow Changes

- In RouterOS v5.x there were several places where packets were queued, so CPU core assignments shuffle happened several times
- In RouterOS v6.x QoS system was redesigned so that queuing happens in the same place respectively to other processes in the router – at the end.

HTB in RouterOS v5

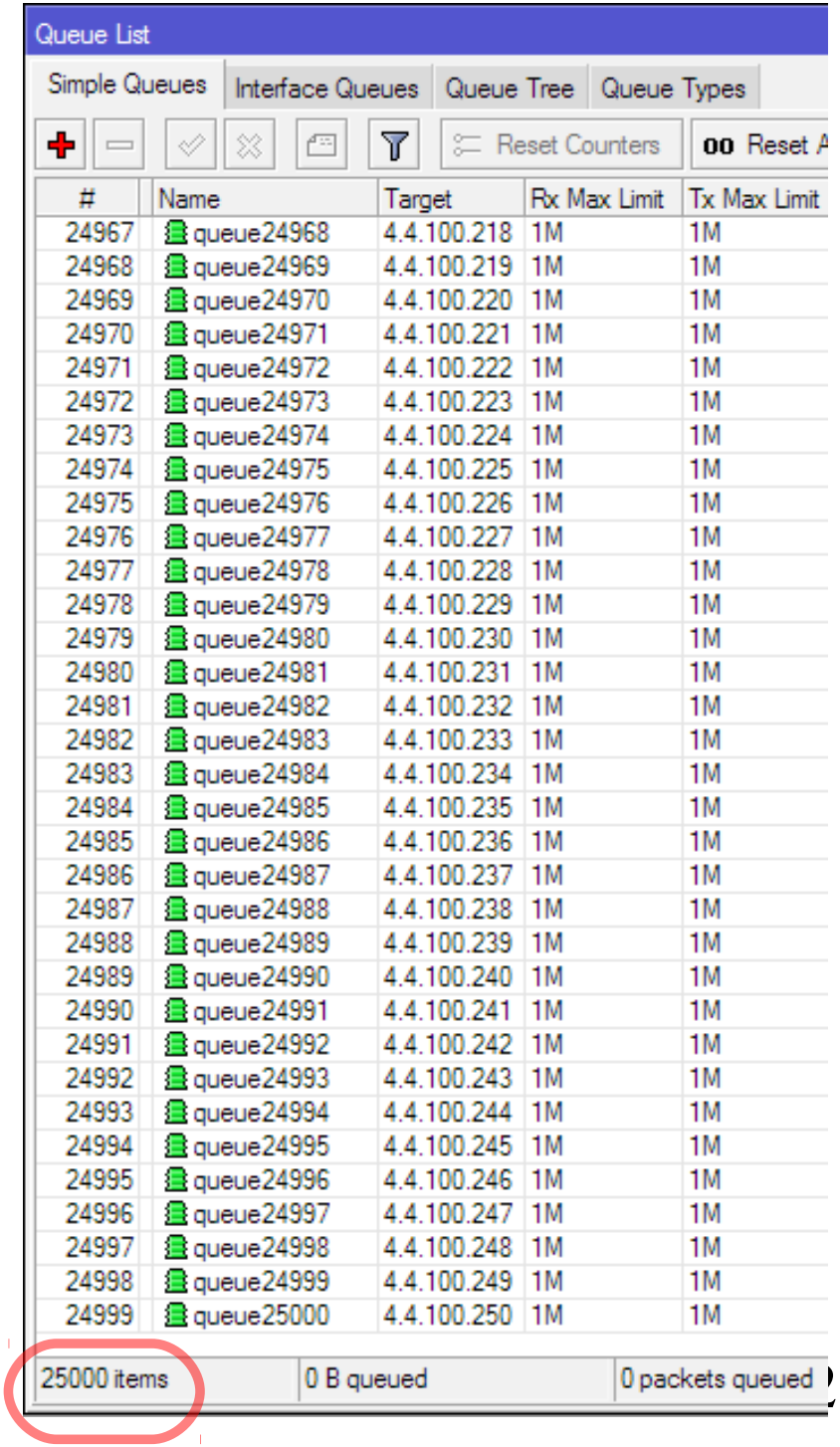


HTB in RouterOS v6



Simple Queues

- The matching algorithm has been updated
  - based on hash
  - faster miss-matches
- Optimal performance on Multi-core devices requires at least 32 top level simple queues, so that queuing process can be distributed properly



Queue Tree and CCR

- The whole HTB tree from Kernel perspective is and will be one queue so only one CPU core can work on HTB at the same time.
- The same optimization as in simple queues (at least 32 top-level queues, faster matching) will come to queue tree in one of the next versions.
- Suggestions:
  - Use Interface HTB as much as possible to offload traffic from HTB “global”
  - Use simple queues

Queue Changes in v6.19

- In RouterOS v6.19 we introduced a software patch to improve queue performance
  - Before CPU core just left packets in the queue and a random other core was taking them out “later”.
  - Now CPU core that leaves packets in the queue will have to take some packets out, at the same time.
  - In a case when queue limit is not reached, the same packets will be left in and taken out of the queue by the same CPU core, making this process seamless

PPTP,L2TP and PPPoE on CCR

- Changes introduced in v6.8:
  - kernel drivers for PPP, PPPoE, PPTP, L2TP now are lock-less on transmit & receive
  - all ppp packets (except discovery packets) now can be handled by multiple cores
  - MPPE driver now can handle up to 256 out-of-order packets (Previously even single out-of-order packet was dropped)
  - roughly doubled the MPPE driver encryption performance

Single PPTP Tunnel Performance on CCR1036

in packets per second with 0,01% loss tolerance

Encryption	Conntrack	Version	64 byte	512 byte	1024 byte	1280 byte	1518 byte
No	No	6.7	135,788	143,454	148,728	147,046	71,400
No	No	6.8rc1	2,123,150	2,336,314	2,305,772	1,865,984	958,942
No	Yes	6.7	98,818	88,294	89,442	100,090	36,826
No	Yes	6.8rc1	1,191,870	1,215,884	1,201,644	1,179,104	32,738
Yes	No	6.7	47,432	38,180	30,830	27,042	13,206
Yes	No	6.8rc1	190,894	233,676	239,462	190,820	80,828
Yes	Yes	6.7	39,062	29,884	22,560	21,062	12,826
Yes	Yes	6.8rc1	189,266	233,676	239,462	190,820	14,192

Single L2TP Tunnel Performance on CCR1036

in packets per second with 0,01% loss tolerance

Encryption	Conntrack	Version	64 byte	512 byte	1024 byte	1280 byte	1518 byte
No	No	6.7	120,906	123,428	197,486	197,846	84,290
No	No	6.8rc1	3,708,612	3,522,140	2,312,554	1,868,990	1,214,036
No	Yes	6.7	98,120	105,438	103,230	101,506	50,160
No	Yes	6.8rc1	1,687,126	1,580,948	1,382,294	1,302,770	53,170
Yes	No	6.7	62,894	47,598	35,628	32,038	19,238
Yes	No	6.8rc1	212,052	234,962	226,366	215,688	91,830
Yes	Yes	6.7	47,432	37,134	29,408	26,614	16,464
Yes	Yes	6.8rc1	212,052	231,364	206,372	208,552	22,782

Single PPPoE Tunnel Performance on CCR1036

in packets per second with 0,01% loss tolerance

Encryption	Conntrack	Version	64 byte	512 byte	1024 byte	1280 byte	1518 byte
No	No	6.7	294,828	305,358	302,276	305,326	N/A
No	No	6.8rc1	5,519,320	4,633,852	2,376,862	1,912,372	N/A
No	Yes	6.7	277,156	260,386	192,272	183,856	83,844
No	Yes	6.8rc1	2,730,596	2,462,744	2,103,608	1,910,588	80,028
Yes	No	6.7	71,730	51,084	37,940	33,894	N/A
Yes	No	6.8rc1	212,052	239,322	228,588	208,552	N/A
Yes	Yes	6.7	56,286	43,412	33,318	29,754	19,358
Yes	Yes	6.8rc1	234,840	234,962	226,366	208,552	29,004

CCR and Packet Fragments

- Currently Connection Tracking requires packet to be re-assembled before further processing
- It is impossible to ensure that all fragments of the packet are received by the same CPU core
- We plan to
  - Make automatic MTU calculation for all tunnels (even tunnel-over-tunnels)
  - add full support to Path MTU Discovery to all tunnels and interfaces
  - Update Connection Tracking to handle fragments.

IP Firewall and CCR

- Each Firewall rule takes a dedicated place in system memory (RAM)
- CPU core need to process packets through whole list of firewall rules before they are captured
- In RouterOS v6.19 we introduced a Linux Kernel patch that allow us to fully utilize unique Tilera TileGx processor cache and memory management features
- All memory heavy processes now should work faster (especially firewall)

Changes in the Firewall

- Firewall now has “all-ether”,”all-wireless”,”all-vlan”,”all-ppp” as possibilities in interface matching (only 2 dynamic “change-mss” rules)
- In RouterOS v6.19 we introduced a patch that:
  - optimized option matching order with-in a rule – simple options will be matched before getting to more complex options
  - Rules that can't possibly match any packets because of current configuration, will be marked as invalid and skipped (with comment that explains why)

Firewall Rule Matching Order

1)in-interface, 2)out-interface, 3)protocol, 4)fragment, 5)src-mac-address, 6)in-bridge-port, 7)out-bridge-port, 8)src-address, 9)dst-address,10)src-address-type,11)dst-address-type,12)dst-address-list, 13)src-address-list,14)ttl,15)dscp, 16)packet-size,17)ipv4-options,18)dst-port, 19)src-port, 20)port, 21)tcp-flags, 22)tcp-mss, 23)icmp-options, 24)ingress-priority, 25)priority, 26)packet-mark, 27)routing-mark, 28)hotspot, 29)connection-mark, 30)connection-state, 31)connection-bytes, 32)connection-limit, 33)connection-rate, 34)connection-type, 35)random, 36)psd, 37)nth, 38)limit, 39)dst-limit, 40)per-connection-classifier, 41)p2p, 42)content, **43)layer7-protocol**

Layer-7

- Layer-7 is the most “expensive” firewall option, it takes a lot of memory and processing power to match each connection to regexp string.
- Layer-7 should be used only as trigger – to add a connection-mark or address-list entry
- Direct action (like accept, drop) should be done by rules that work with those connection marks or address-lists, not by layer-7 rule
- Make sure that if a layer-7 rule is matched once then same connections will not get to the layer-7 rule.

Routing in RouterOS v6

- Packet routing can utilize all cores
- All dynamic routing protocols (more precisely - routing table updates and protocol calculations) are limited to a single core.
  - One BGP full feed will take <3min to load on CCR
  - Two BGP full feeds will take <8min to load on CCR
- Currently board with 256Mb RAM can only hold 2 BGP full feeds
- Searching a specific set of routes takes too long.

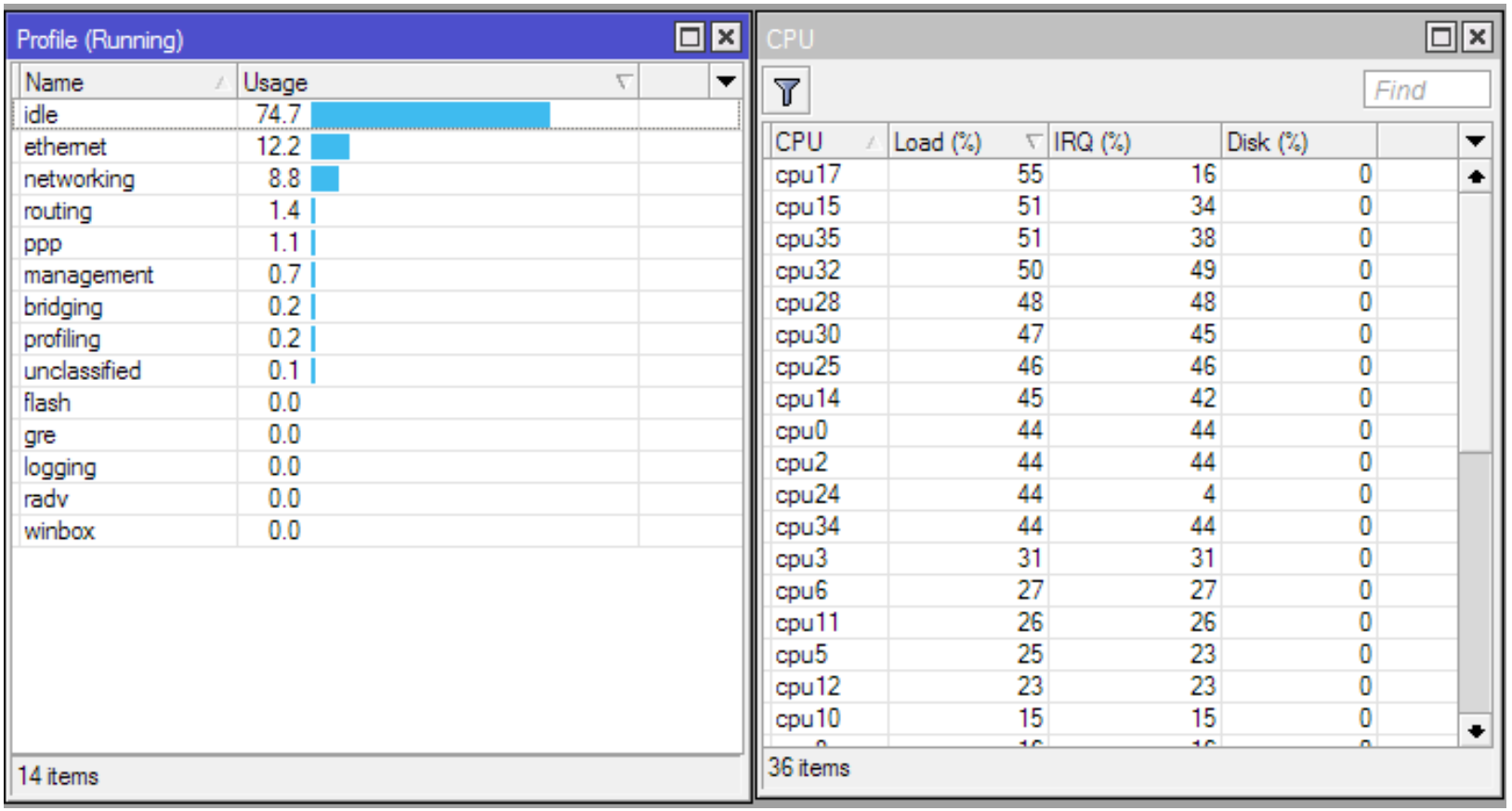
Routing in RouterOS v7

- More that 10x faster routing table calculations
  - One BGP full feed will take ~1.5min to load on CCR
  - Two BGP full feeds will take <2min to load on CCR
- Much more scalable (60 BGP full feeds no problems)
- Uses much less RAM
  - full BGP feed takes ~36MB RAM
  - 12 million routes takes ~500MB RAM
- Completely isolated VRFs
- Much faster route table searches

IPSec and CCR

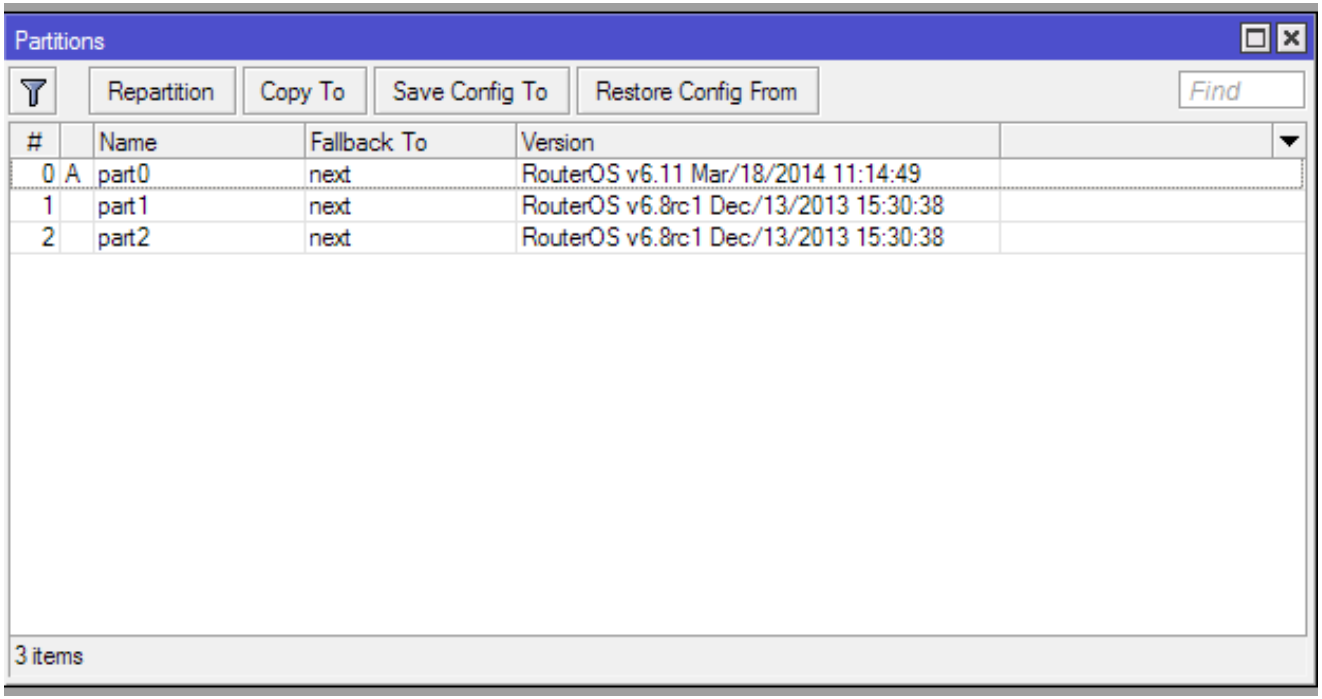
- Hardware acceleration support for aes-cbc + md5|sha1|sha256 Authenticated Encryption with Associated Data (AEAD) was added on CCR in RouterOS v6.8
- Now the CCR1036 can handle 3.2Gbps encrypted IPSec traffic
  - Maintaining ~80% CPU load
  - No fragmentation (1470byte packets)
  - Many peers (100 separate tunnels)
  - AES128 was used

Tools



- /system resources cpu
- /tool profile

Partitions



- Partition will always allow you to keep one working copy of RouterOS just one reboot away and backup configuration before major changes

So what's next for CCR??

- Fix all issues that might be caused by all the patches/improvements implemented into v6.19
- Implement automatic MTU calculation for all tunnel interfaces, add MTU path discovery support
- Apply all changes that are now currently only in simple queues to queue tree.
- Few more patches for general performance improvement.

Questions!!!