Welcome Back

10 Years On!

Quality of Service & Bandwidth Management in RouterOS v.6

Oluyinka Thomas
thomolar@yahoo.com

MikroTik User Meeting

Lagos, Nigeria

28th November, 2017

About Me

- Electrical/Electronics Engineering graduate from University of Ibadan, Nigeria
- Worked briefly with Procter & Gamble
- Worked mostly in the Oil & Gas upstream service industry
  - Coiled Tubing Services

About Me

- Passionate about IT & Telecommunication
- VSAT installer partner for Taide Network AS
  - Taide became Vizada then became part of Astrium
- Love MikroTik RouterOS to bits
- Started using RouterOS from v.2.9
- Certified in MTCNA, MTCRE, MTCINE, Certified Trainer

Agenda

- Define Quality of Service
- Define Bandwidth Management
- Highlight Benefits of Both
- Discuss Implementation Tools for Both
- Examine RouterOS Screenshots on Winbox
- Implementation Examples
- Summary & Conclusion

What is Quality of Service (QoS)?

- Refers to traffic prioritization and resource reservation control mechanisms
- Ability to provide different priorities to different applications, users or data flows
- Guarantee a certain level of performance to a **data flow**

Objective of QoS

- Anybody can deploy internet services
- Identify what affects overall satisfaction of the client
- Capture traffic usage patterns & customize router to dynamically work for them
- Key objective of QoS is differentiation

Bandwidth Management

- The process of measuring and controlling the communications (traffic, packets) on a network link
- Objective is to avoid filling the link to capacity or overfilling the link
- Results in network congestion and poor performance of the network if not done

Benefits to ISP's

- High-cost traffic networks are major assets for ISP's
- Gives the intangible yet significant benefit of seeing what internet traffic is flowing through the network
- Allows ISPs to tier their services to guarantee particular QoS
- Reduces costs and increases the menu of products offered

Benefits to Enterprises

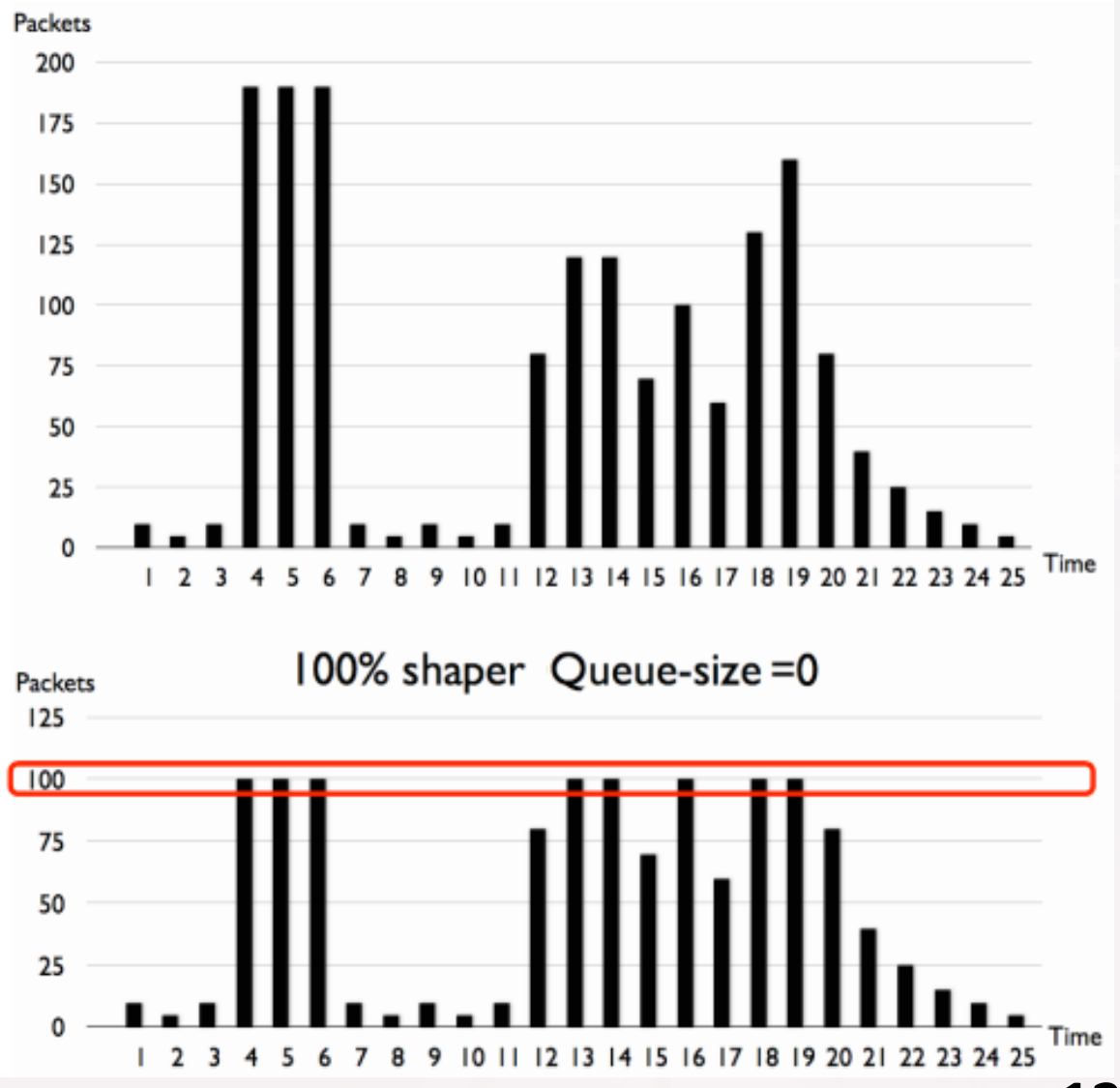
- Applications are centrally hosted at the head office
- Remote offices are expected to pull data from central databases and server farms
- Ensures business-oriented traffic gets priority over best-effort non-critical traffic
- A good means for companies to avoid purchasing additional bandwidth, while properly managing existing resources

Bandwidth Management in ROS

- MikroTik RouterOS is one of the most advanced and easy to configure operating system for bandwidth management
- Traffic shaping (Rate Limiting)
  - HTB and PCQ
- Traffic equalizing (Rate Scheduler)
  - RED, FIFO, SFQ

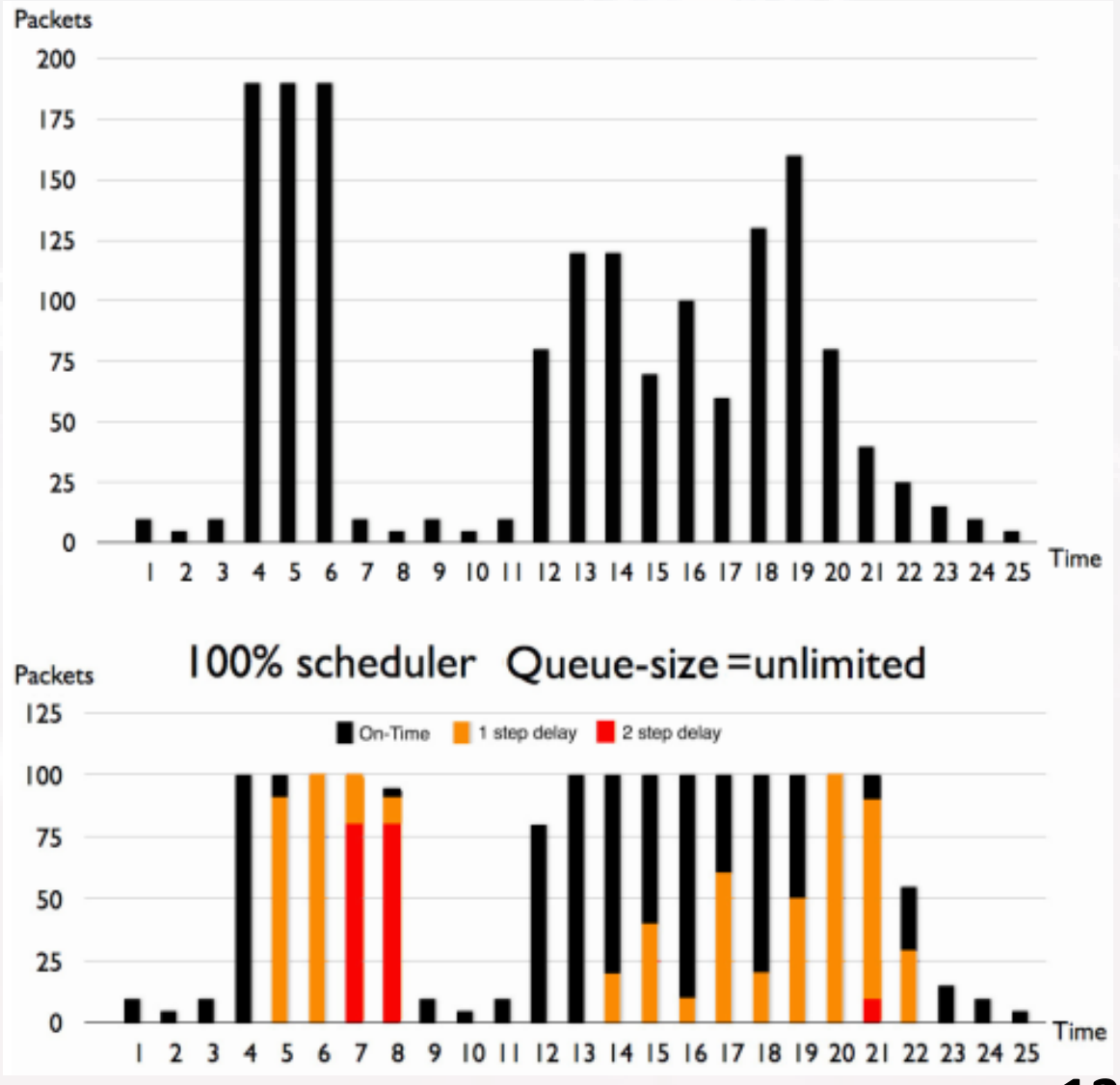
Rate Limiting

- Assume max-limit is ‘100’
- 100% shaper has no queue size
- Therefore packets are dropped when it reaches 100
- In this example about 22% is dropped
- Result: Latency is low

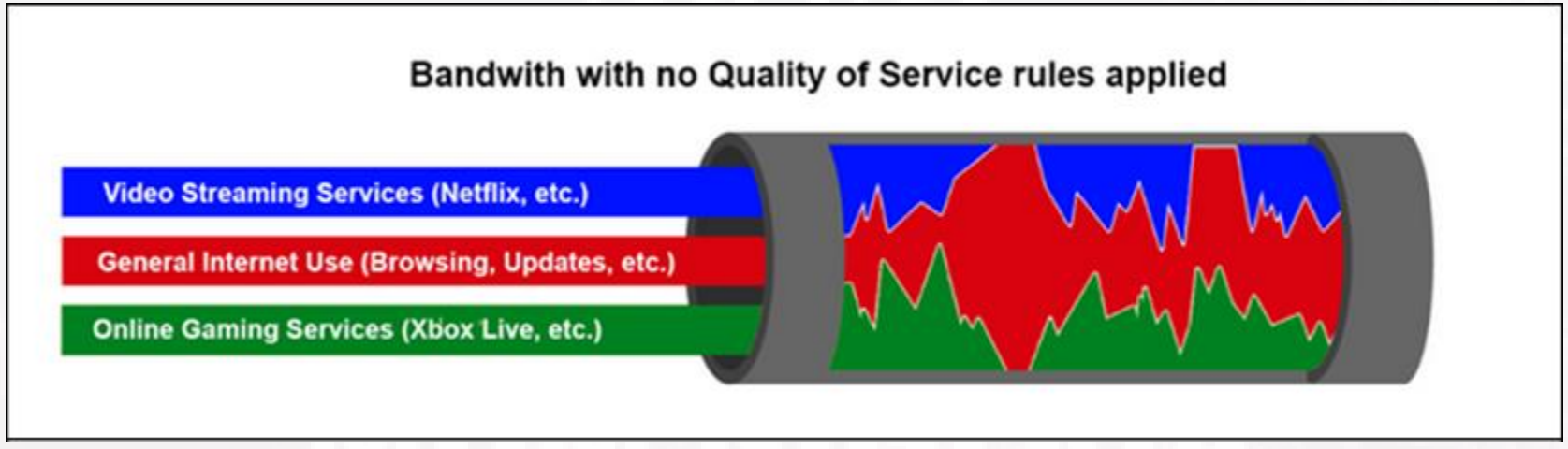


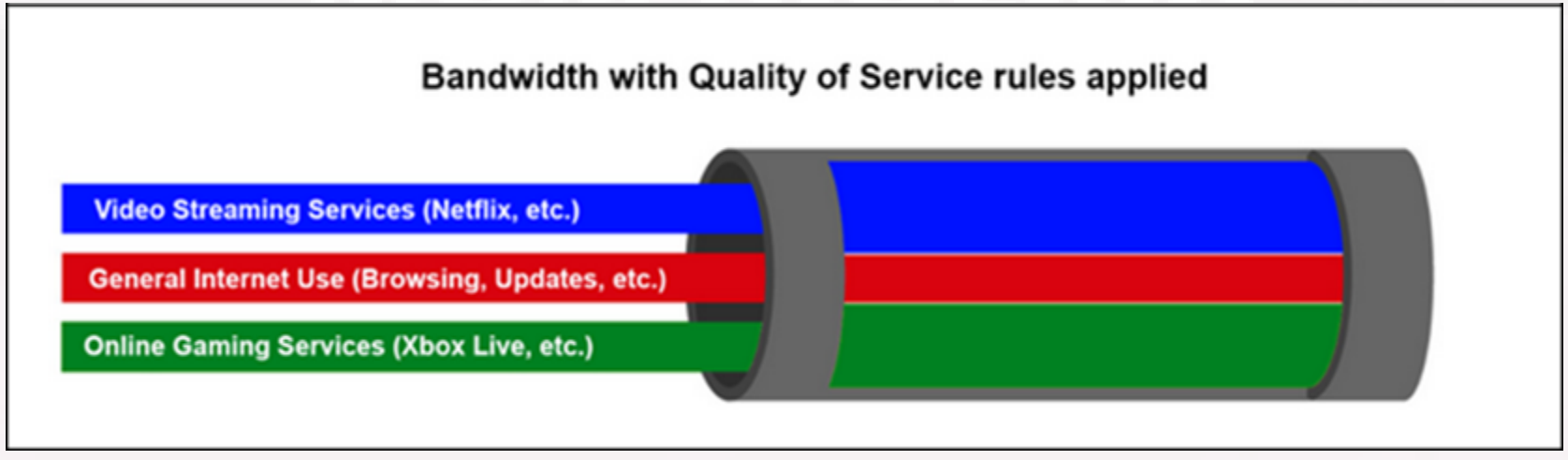
Rate Scheduler

- Assume max-limit is ‘100’
- Queue size is unlimited
- Therefore no packets are dropped when it reaches 100
- In this example 39% are delayed once, 11% delayed twice
- Result: Latency is high



Bandwidth with & without QoS





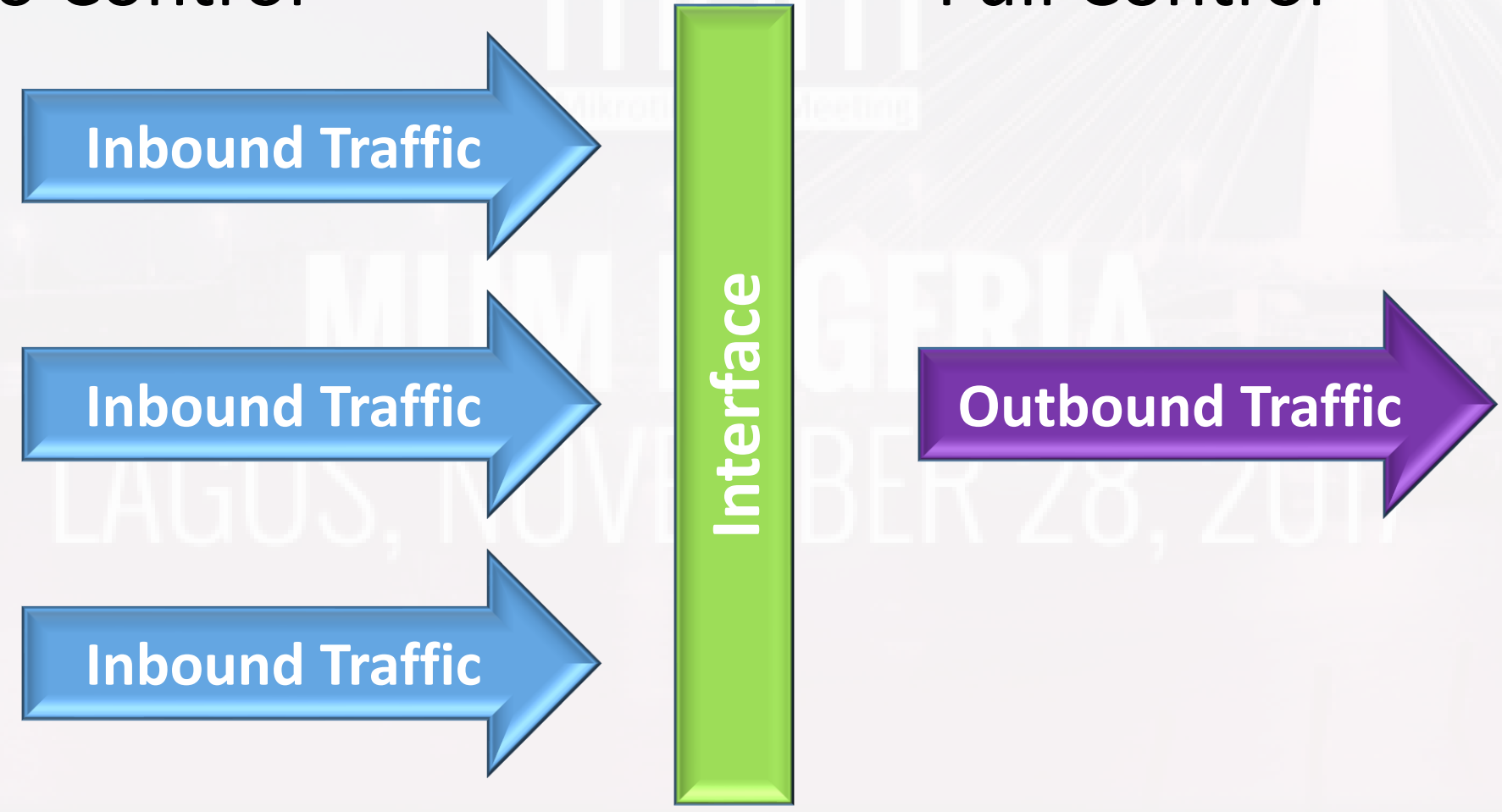
Fundamental Concepts

- We have no control on how much traffic is being sent to an interface
- Traffic control can only be done as the traffic leaves from the interface
- Hence, all control is done on the outbound interface irrespective of upload or download

Traffic Control to an Interface

No Control

Full Control



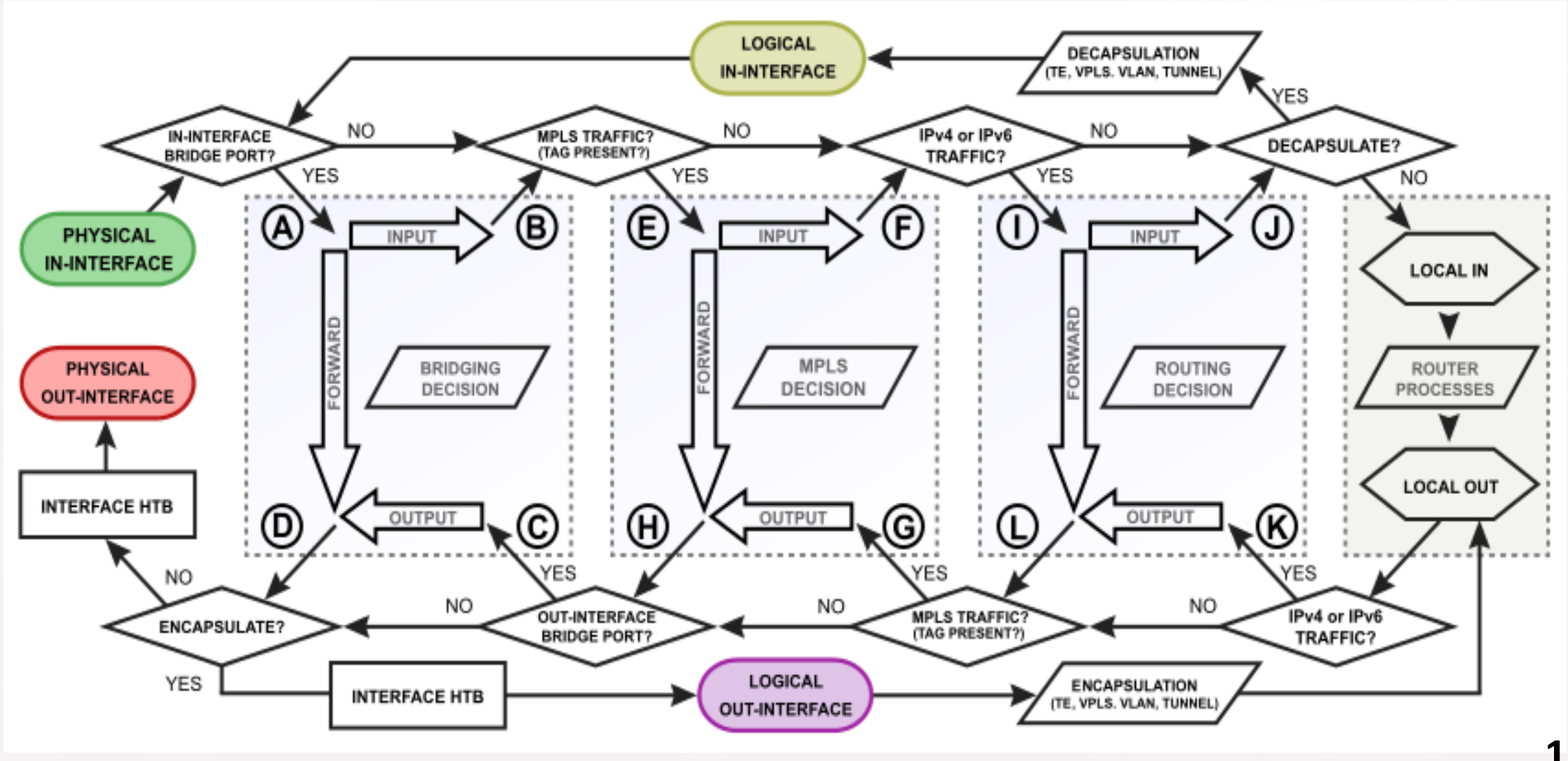
Tools for Implementing BWM & QoS

- Packet Flow Diagram (ROS v.6)
- Mangle
- Address List
- Simple Queues
- Hierarchical Token Bucket, HTB
- Queue Tree
- Per Connection Queue, PCQ

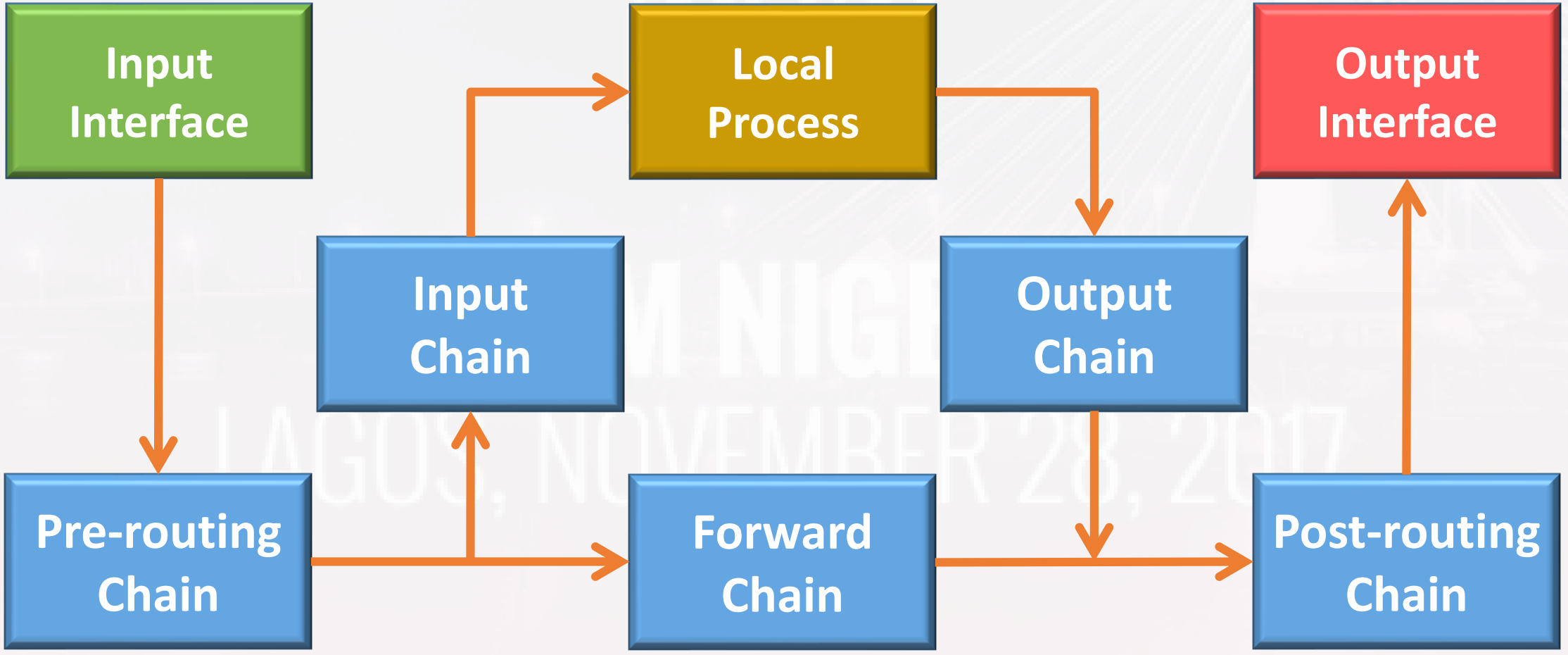
Packet Flow Diagram (ROS v.6)

- Created to form a basis of understanding how packets flow through MikroTik router
- Used to determine where, when and what actions can be taken at any given point
- Knowledge helps to simplify complicated tasks within the RouterOS facilities

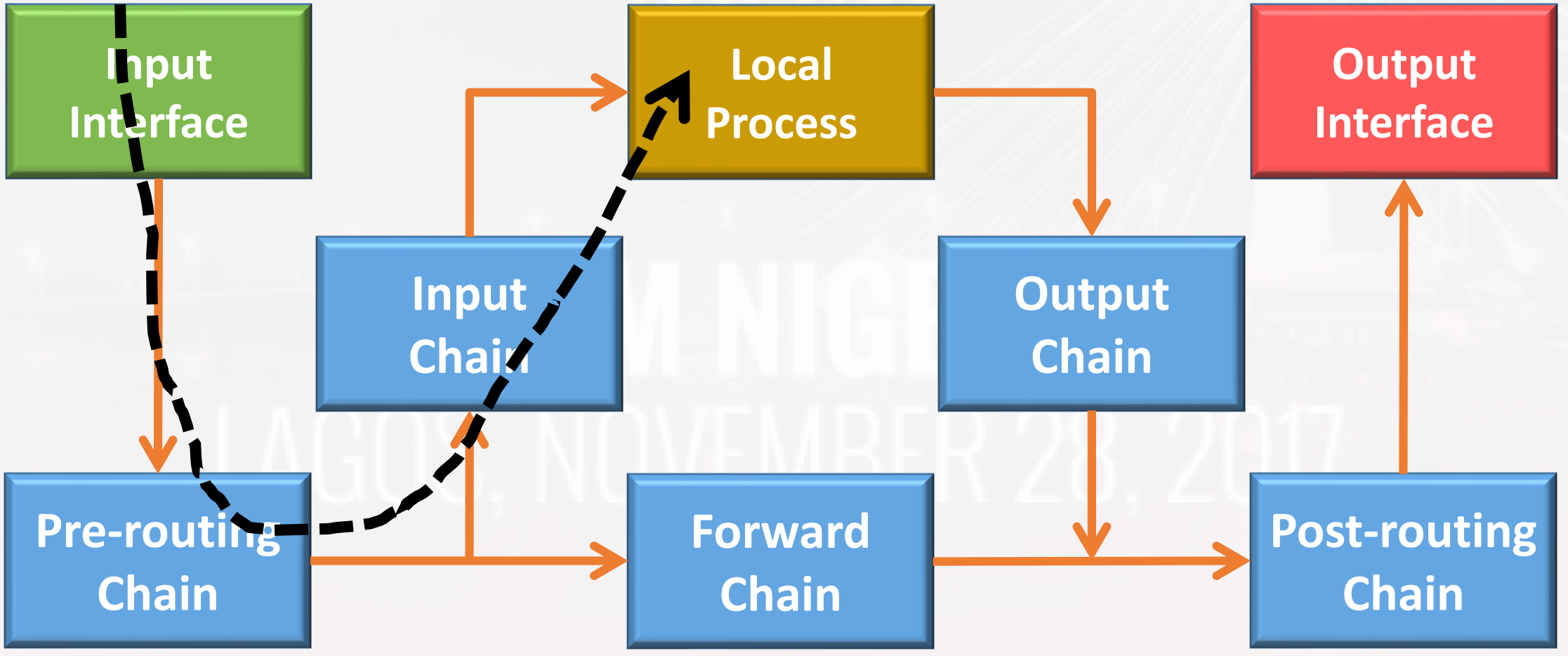
Packet Flow Diagram (ROS v.6)



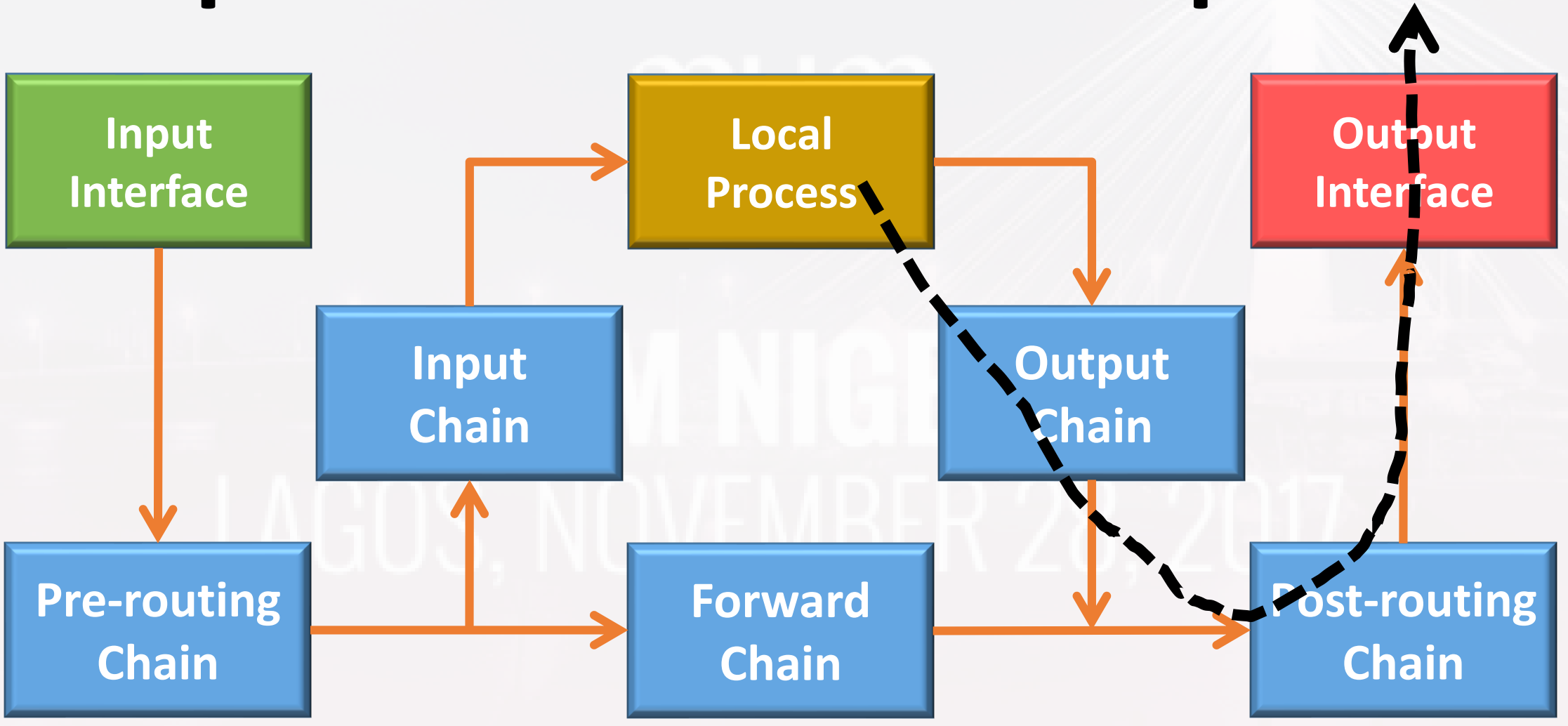
Packet Flow Simplified Diagram



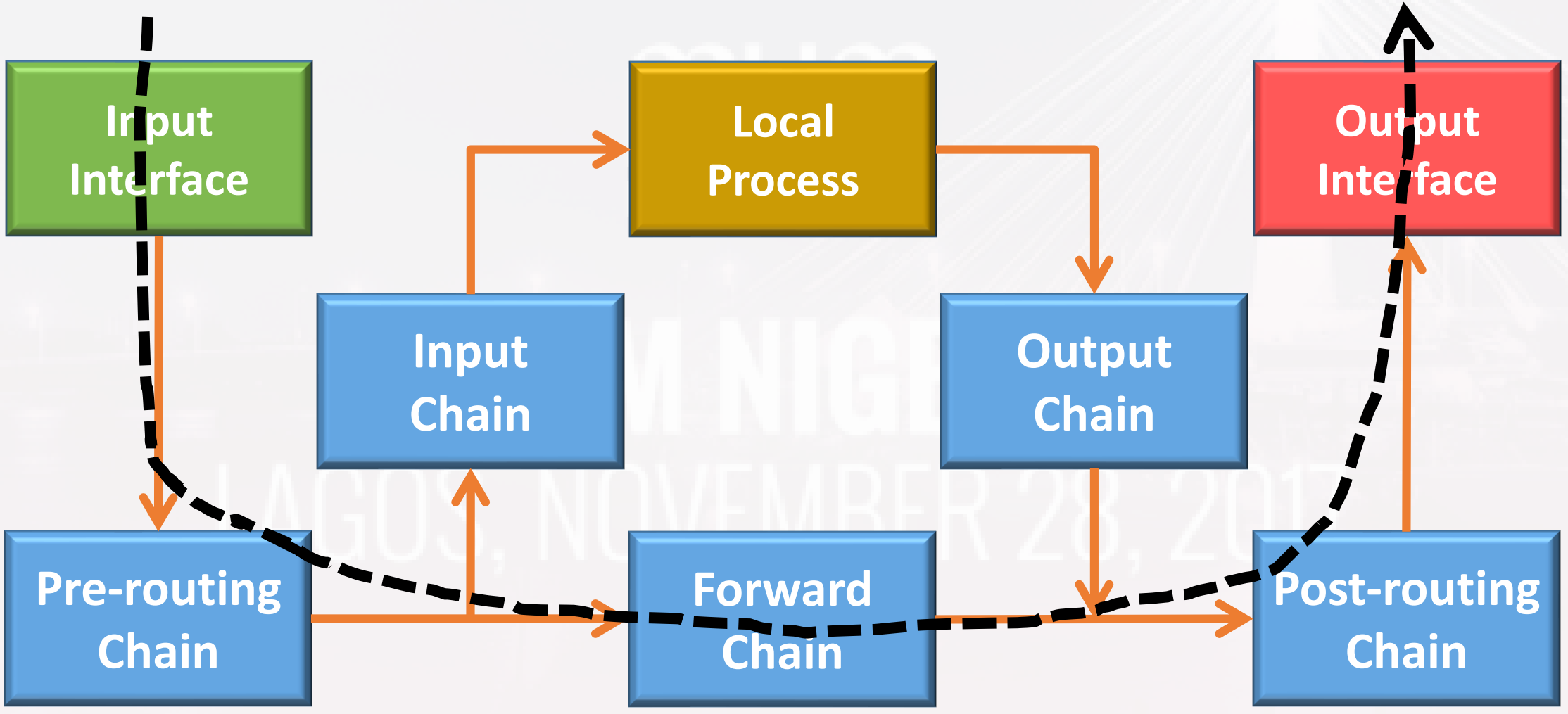
Input Traffic in PFD Simplified



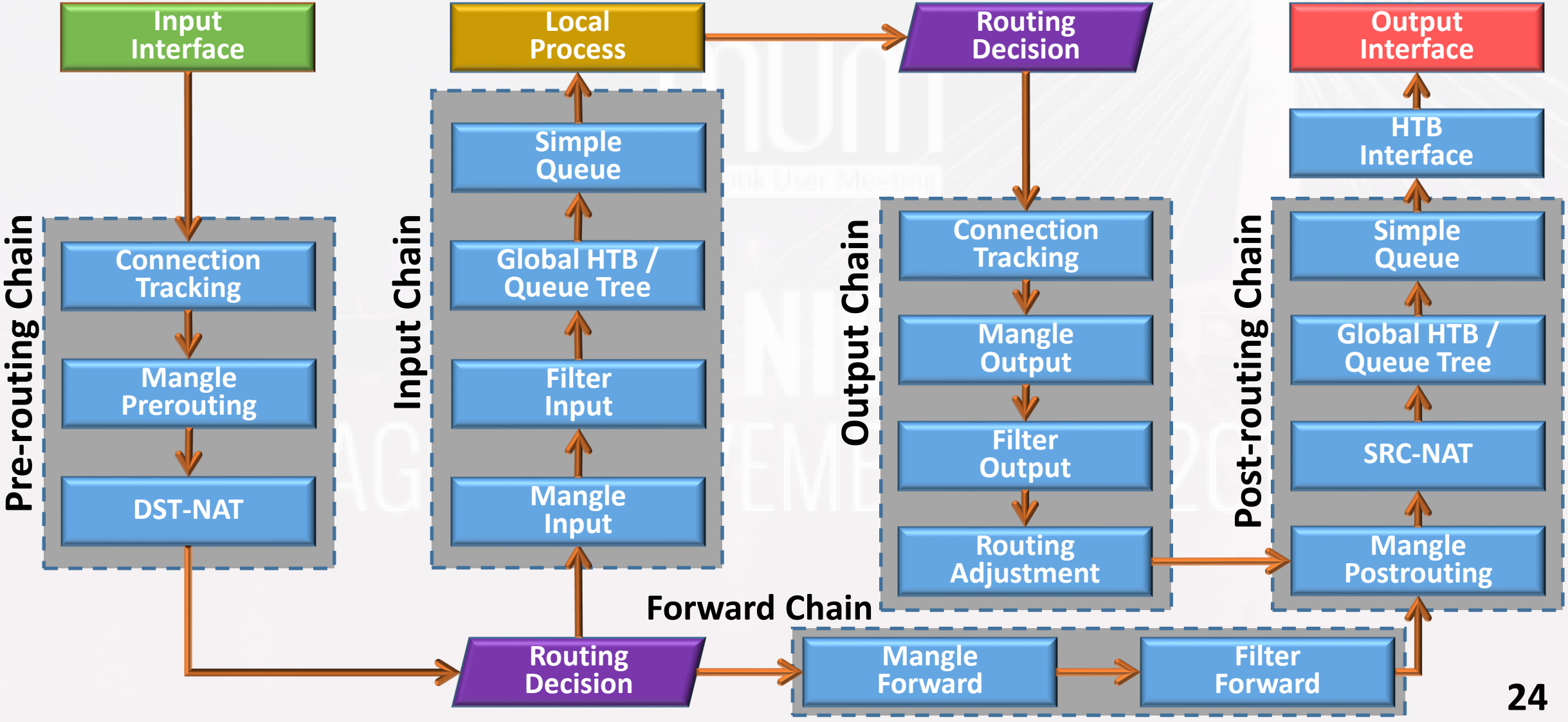
Output Traffic in PFD Simplified



Forward Traffic in PFD Simplified



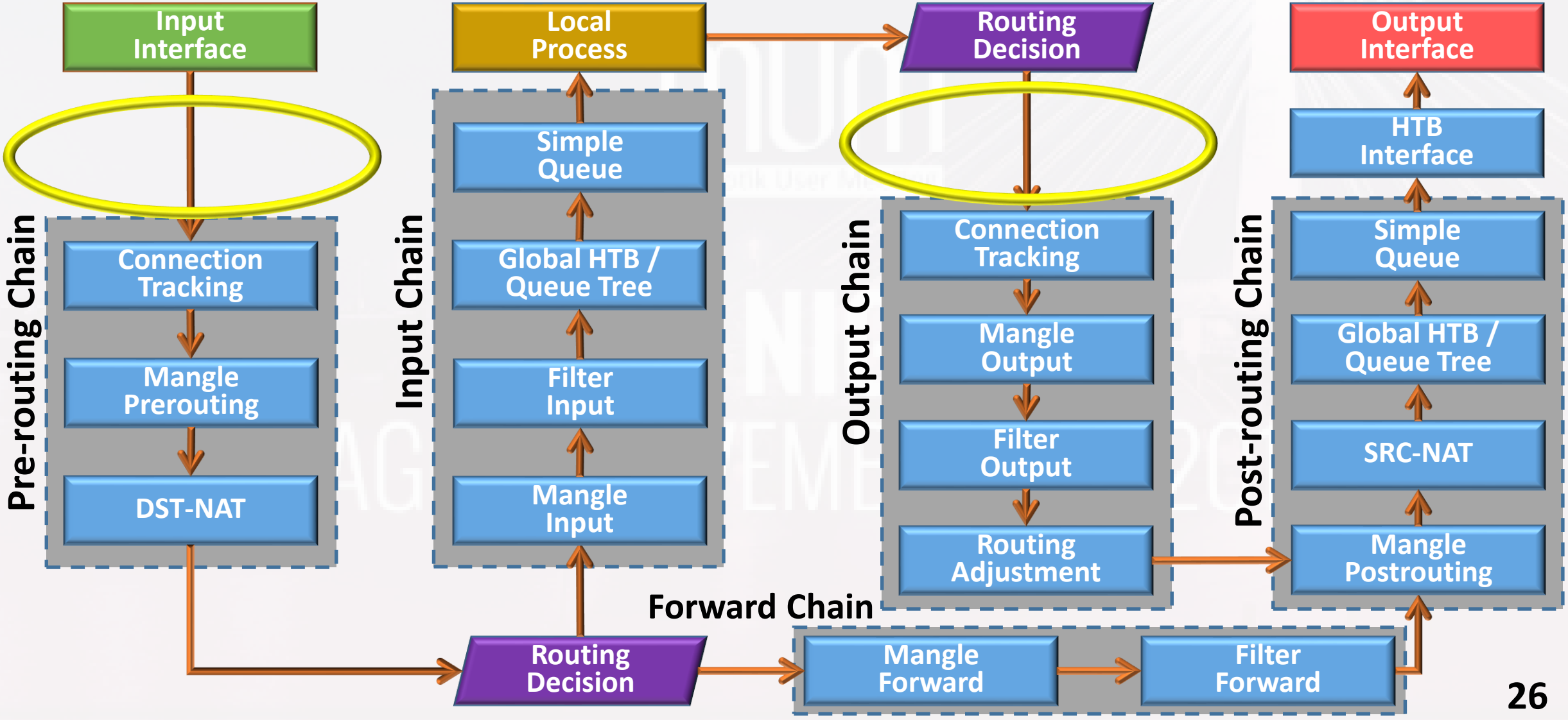
Packet Flow Diagram (ROS v.6)



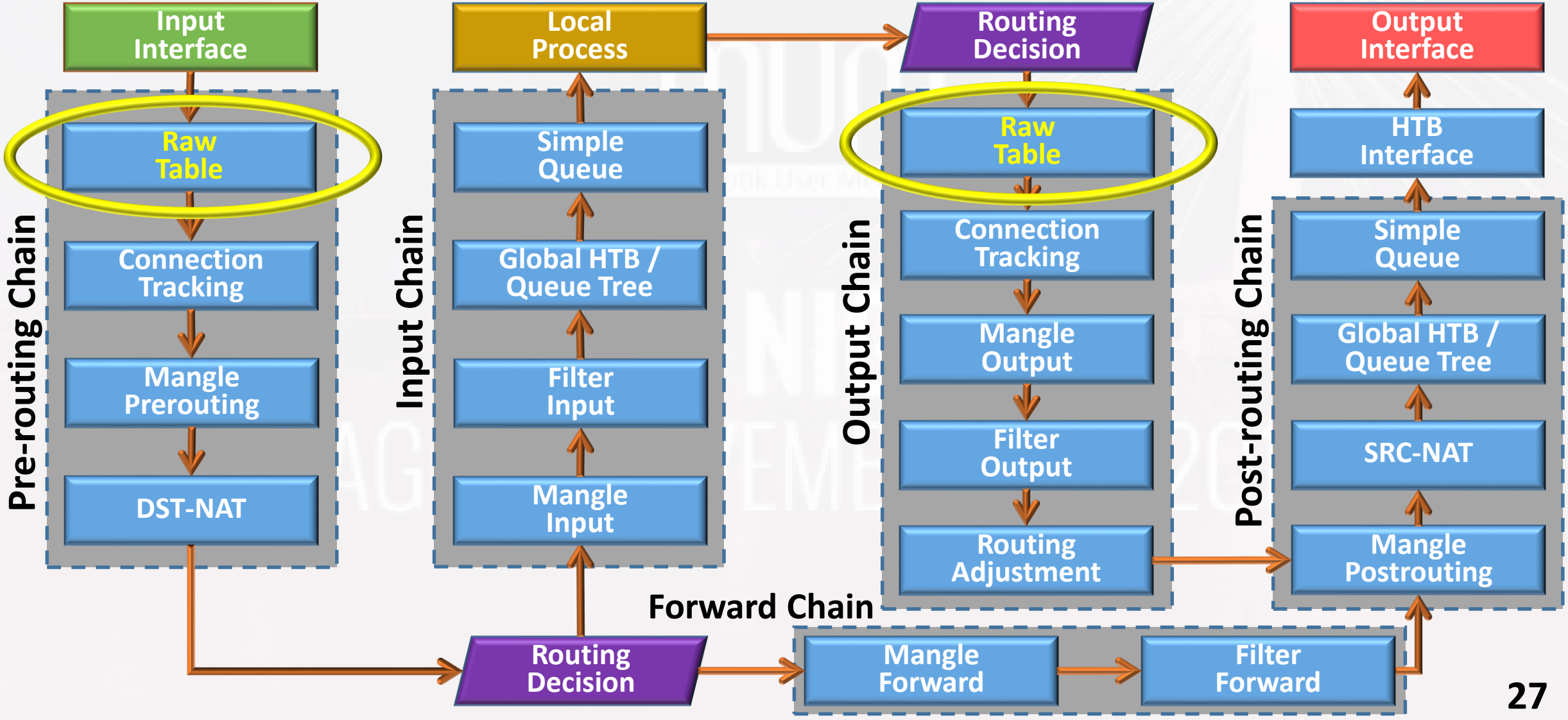
Distributed Denial-of-Service (DDoS)

- A DDoS attack occurs when multiple systems flood the bandwidth or resources of a targeted system
- The incoming disruptive traffic comes from different IP addresses
- This effectively makes it impossible to stop the attack simply by using ingress filtering
  - Connection tracking, a CPU intensive process, already engaged

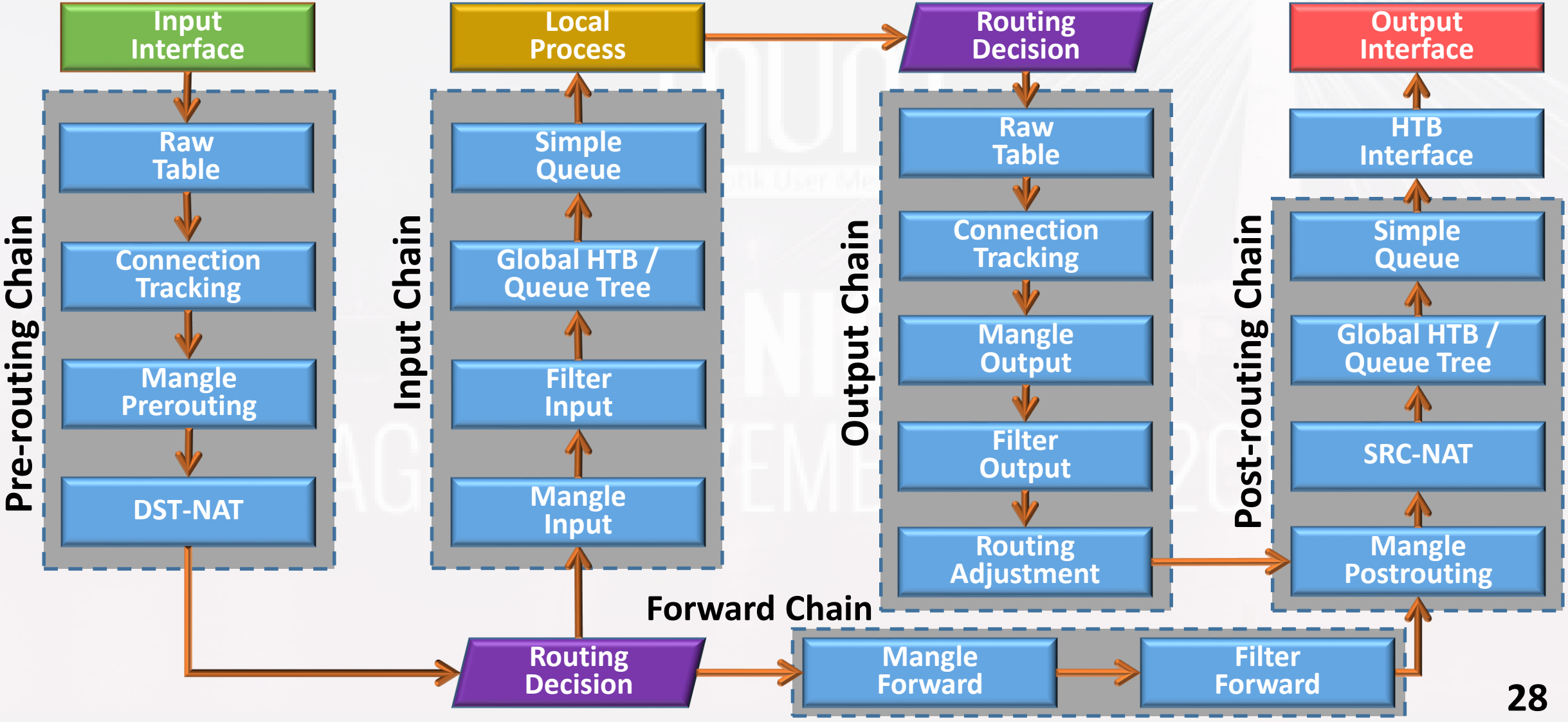
Packet Flow Diagram (ROS v.6)



Packet Flow Diagram (ROS v.6)



Packet Flow Diagram (ROS v.6)

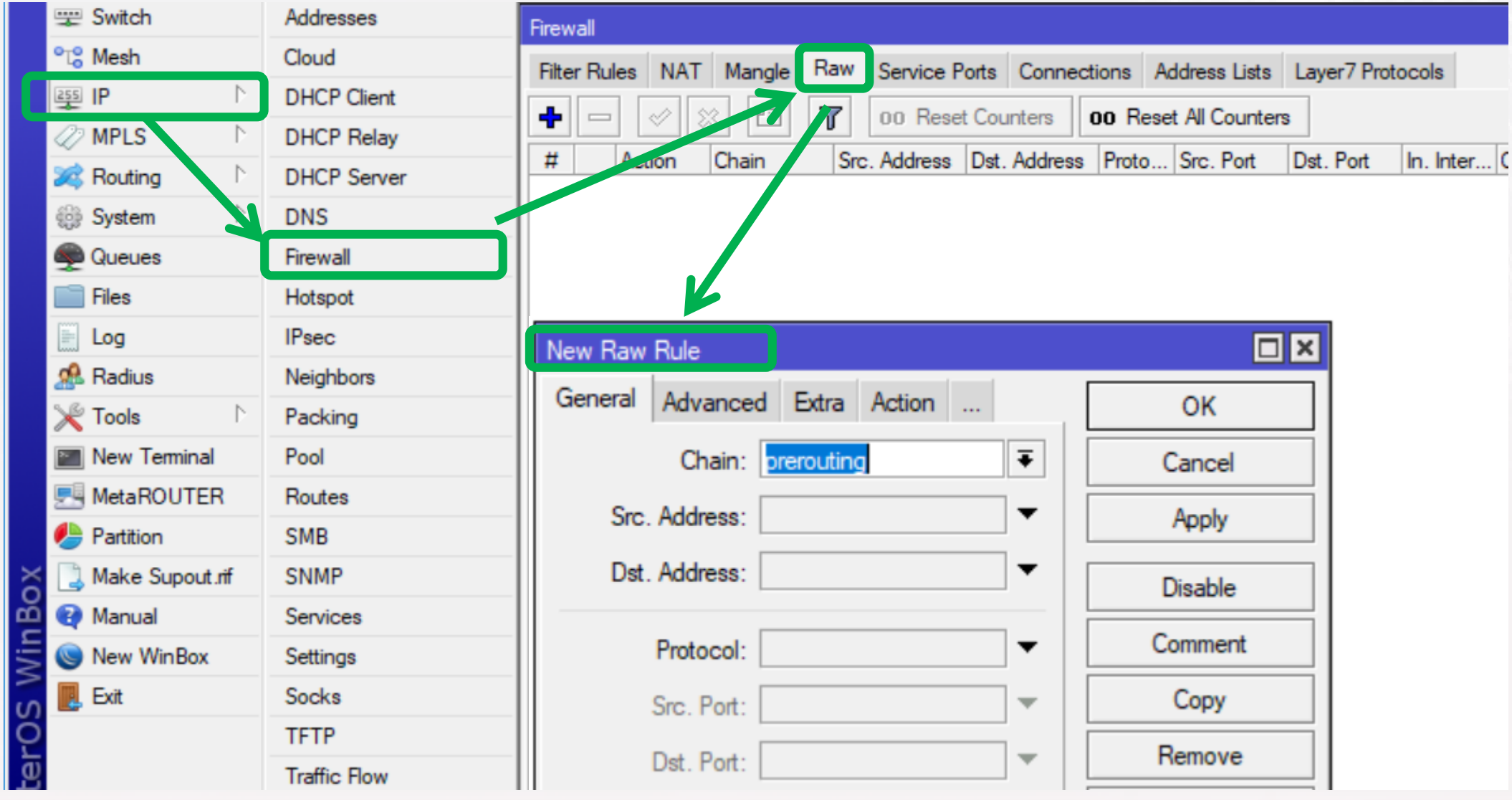


Raw Table

- Very useful tool for DDOS attack mitigation
- Allows to selectively bypass or drop packets before connection tracking
  - Significantly reducing load on CPU
- Does not have matchers that depend on connection tracking (like connection-state, L-7, etc)
- If packet is marked to bypass connection tracking packet de-fragmentation will not occur

IP → Firewall → Raw → +

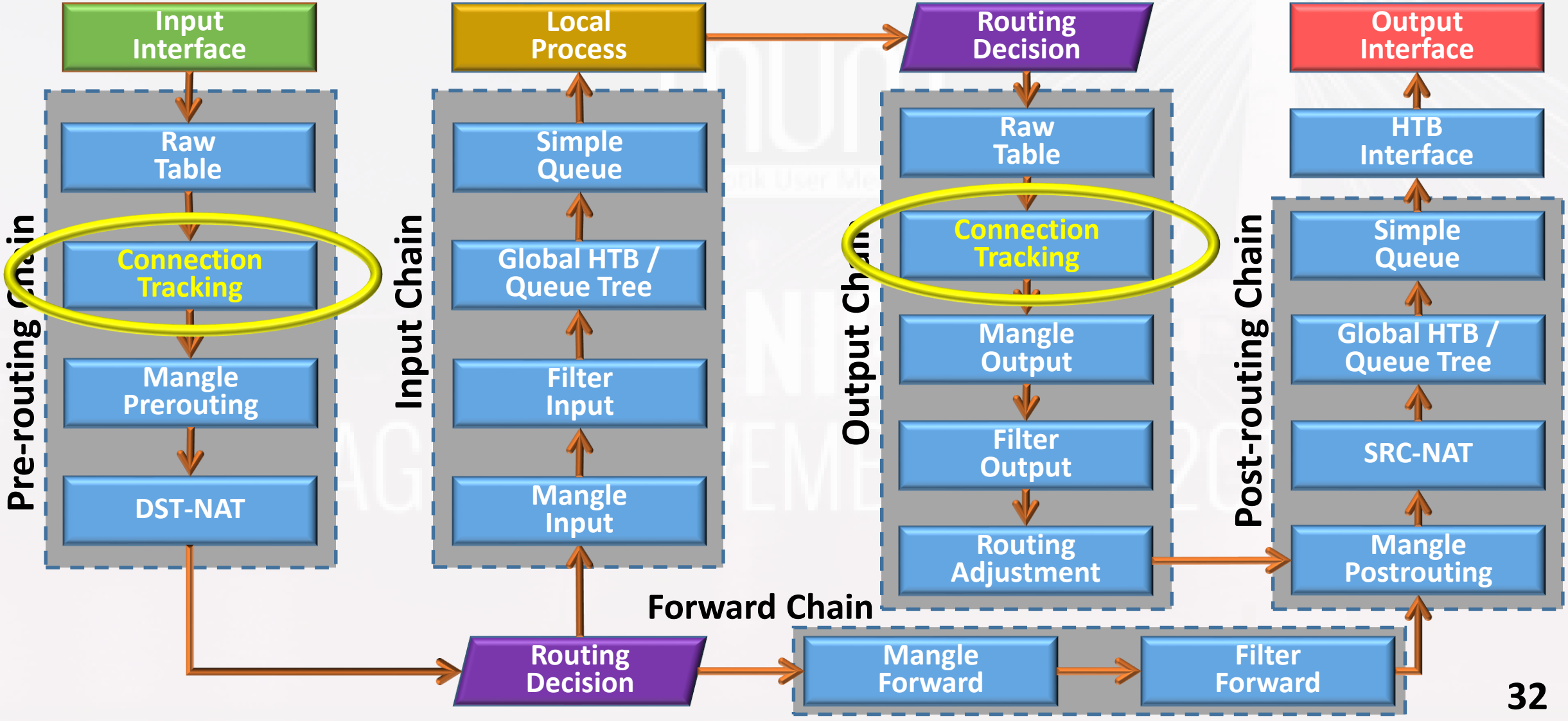
Winbox



Connection Tracking

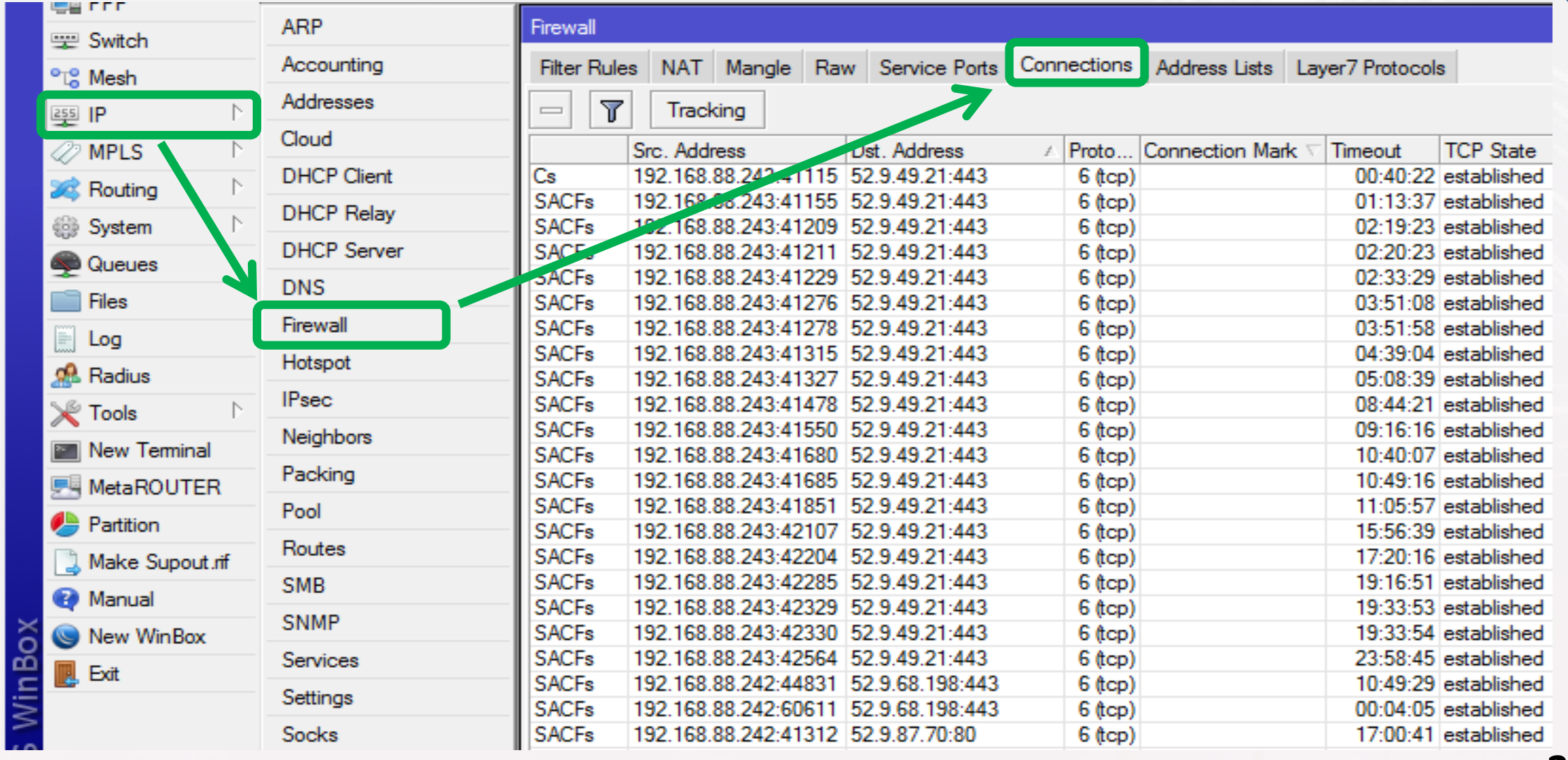
- A way to see what connections are making their way to, from & though the router
- Required for several Firewall facilities in the router to function
  - NAT, Mangle, Filter, etc will stop working if disabled
- Displays source & destination IP addresses and ports associated with a specific connection
- A CPU intensive feature

Connection Tracking in PFD (ROS v.6)



IP → Firewall → Connections

Winbox



	Src. Address	Dst. Address	Proto...	Connection Mark	Timeout	TCP State
Cs	192.168.88.242:41115	52.9.49.21:443	6 (tcp)		00:40:22	established
SACFs	192.168.88.243:41155	52.9.49.21:443	6 (tcp)		01:13:37	established
SACFs	192.168.88.243:41209	52.9.49.21:443	6 (tcp)		02:19:23	established
SACFs	192.168.88.243:41211	52.9.49.21:443	6 (tcp)		02:20:23	established
SACFs	192.168.88.243:41229	52.9.49.21:443	6 (tcp)		02:33:29	established
SACFs	192.168.88.243:41276	52.9.49.21:443	6 (tcp)		03:51:08	established
SACFs	192.168.88.243:41278	52.9.49.21:443	6 (tcp)		03:51:58	established
SACFs	192.168.88.243:41315	52.9.49.21:443	6 (tcp)		04:39:04	established
SACFs	192.168.88.243:41327	52.9.49.21:443	6 (tcp)		05:08:39	established
SACFs	192.168.88.243:41478	52.9.49.21:443	6 (tcp)		08:44:21	established
SACFs	192.168.88.243:41550	52.9.49.21:443	6 (tcp)		09:16:16	established
SACFs	192.168.88.243:41680	52.9.49.21:443	6 (tcp)		10:40:07	established
SACFs	192.168.88.243:41685	52.9.49.21:443	6 (tcp)		10:49:16	established
SACFs	192.168.88.243:41851	52.9.49.21:443	6 (tcp)		11:05:57	established
SACFs	192.168.88.243:42107	52.9.49.21:443	6 (tcp)		15:56:39	established
SACFs	192.168.88.243:42204	52.9.49.21:443	6 (tcp)		17:20:16	established
SACFs	192.168.88.243:42285	52.9.49.21:443	6 (tcp)		19:16:51	established
SACFs	192.168.88.243:42329	52.9.49.21:443	6 (tcp)		19:33:53	established
SACFs	192.168.88.243:42330	52.9.49.21:443	6 (tcp)		19:33:54	established
SACFs	192.168.88.243:42564	52.9.49.21:443	6 (tcp)		23:58:45	established
SACFs	192.168.88.242:44831	52.9.68.198:443	6 (tcp)		10:49:29	established
SACFs	192.168.88.242:60611	52.9.68.198:443	6 (tcp)		00:04:05	established
SACFs	192.168.88.242:41312	52.9.87.70:80	6 (tcp)		17:00:41	established

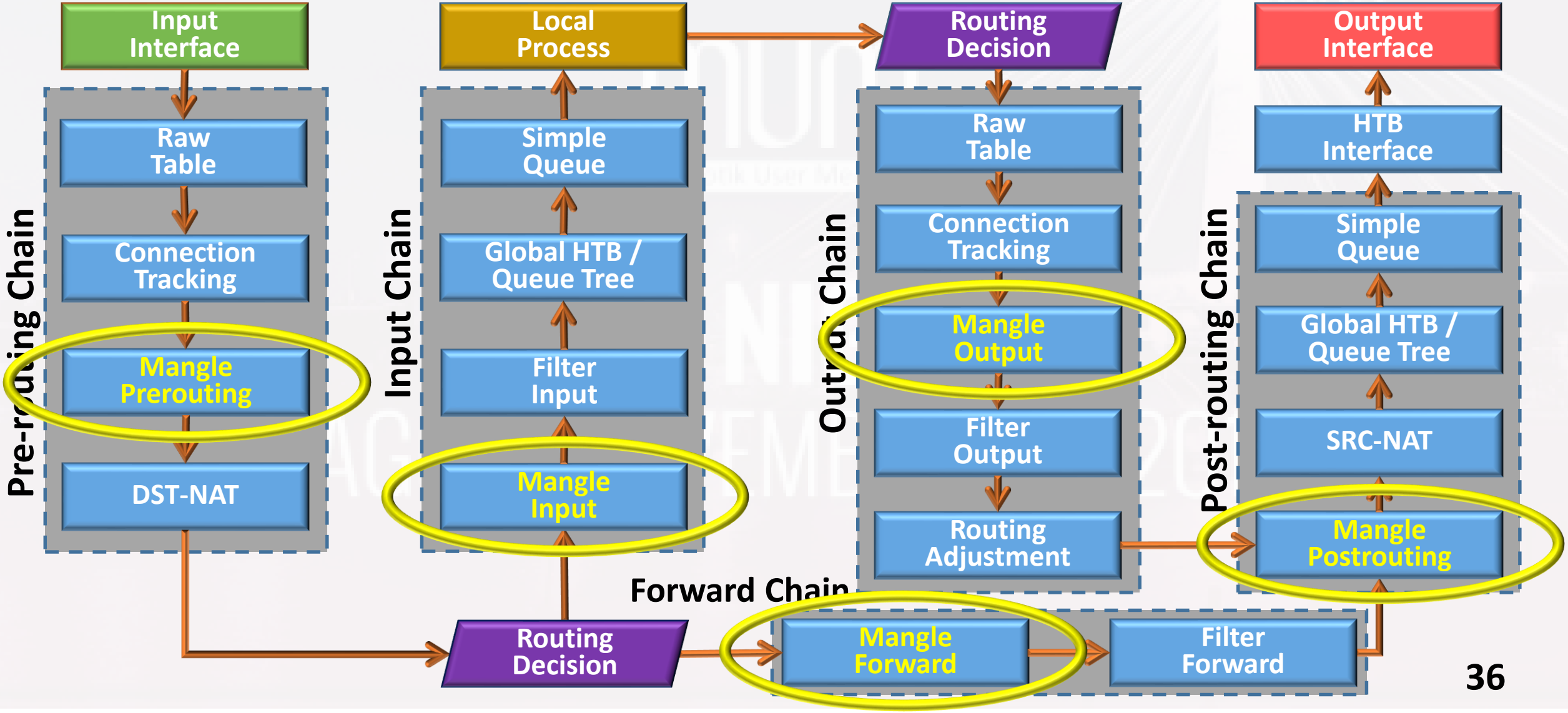
Mangle

- Marks IP connections (bidirectional) and packets (unidirectional) with special marks
- These marks are used for future processing within the same router
  - Filter, Queue and Routing facilities use these marks
- Used to modify some fields in the IP header
  - DSCP (TOS), TTL and MSS fields can be changed

Mangle

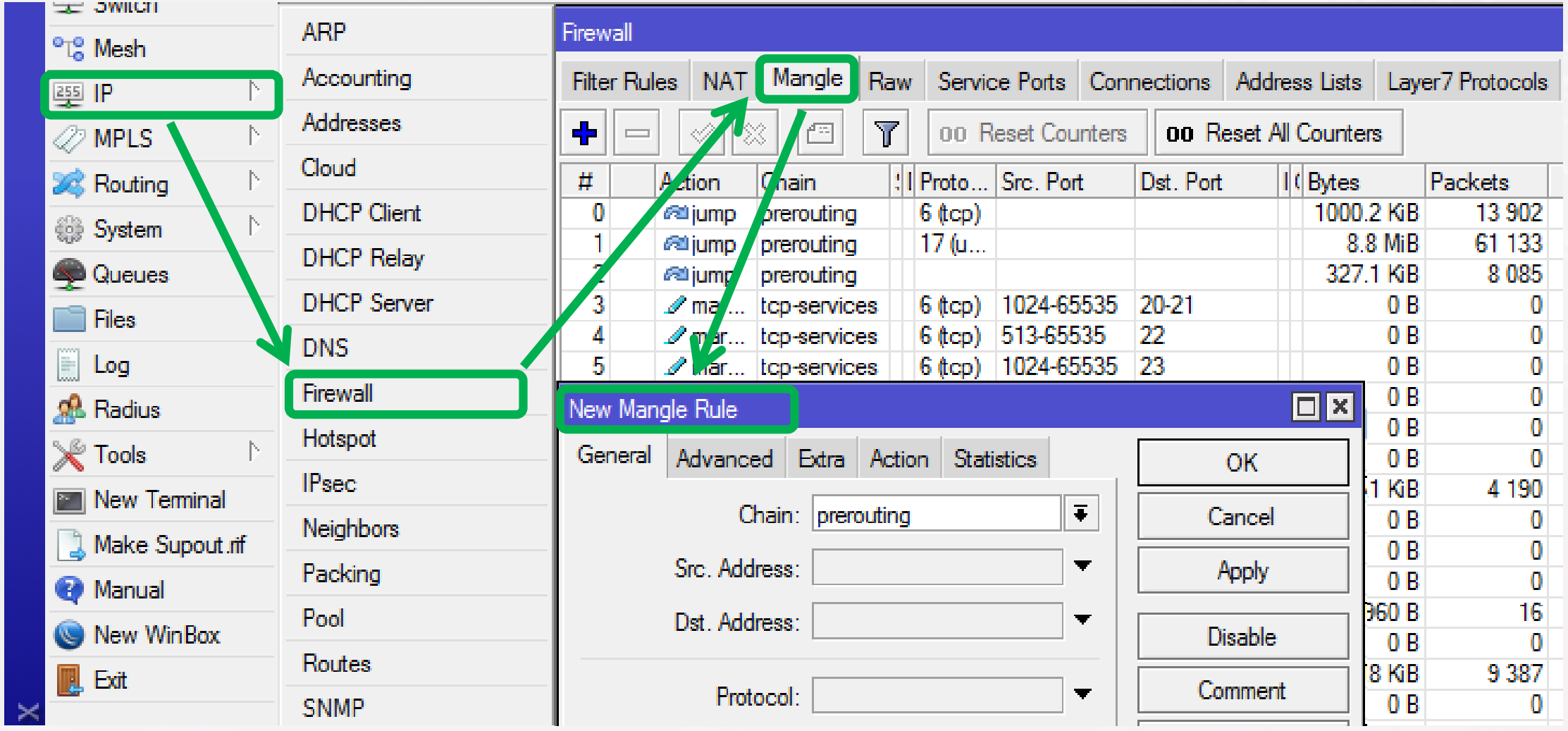
- Mangle rules are organised in 5 default chains
  - Prerouting
  - Input
  - Forward
  - Output
  - Postrouting
- Custom user-defined chains can be added

Mangle Chains in PFD (ROS v.6)



IP → Firewall → Mangle → +

Winbox

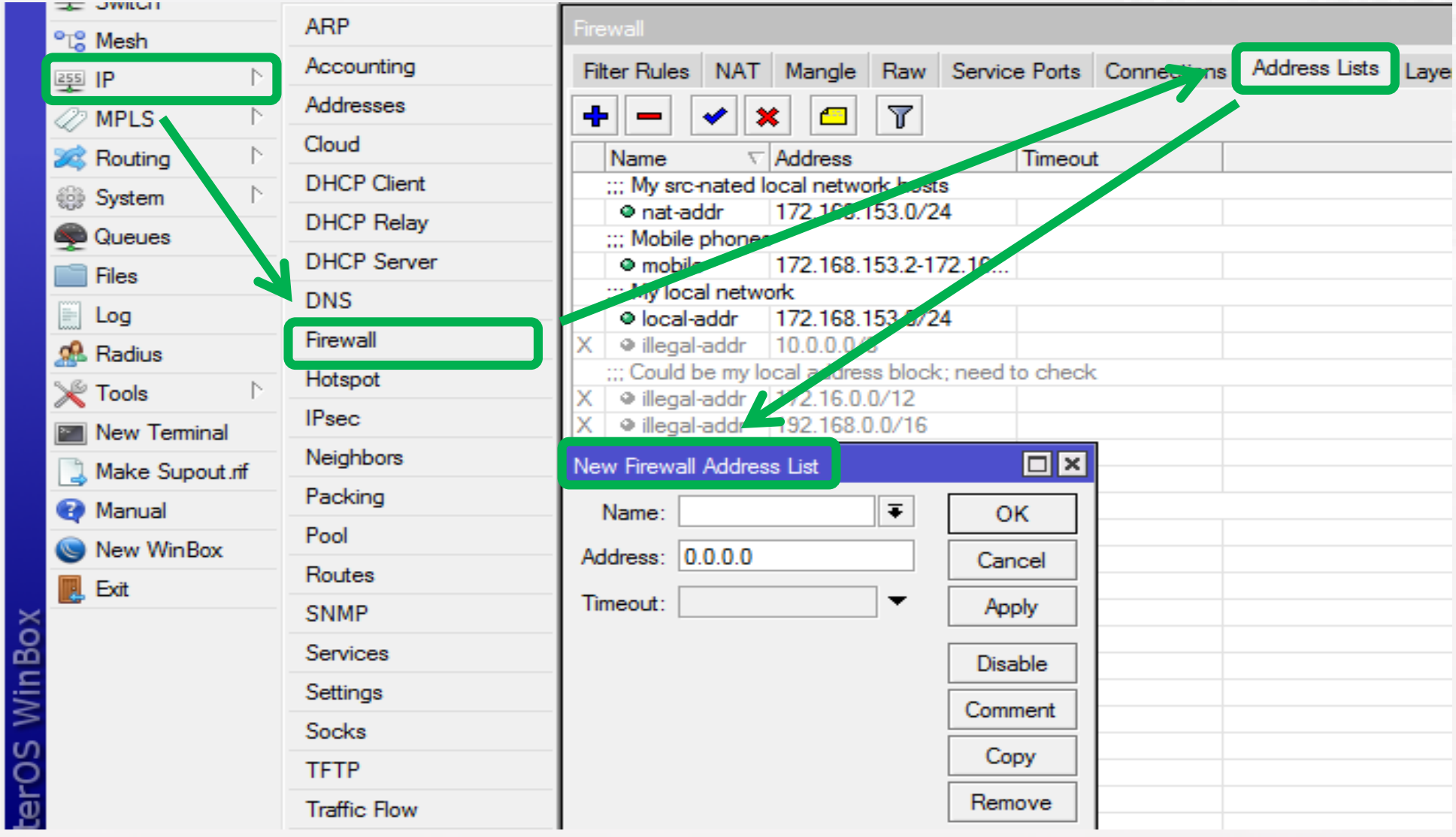


Address List

- Allows to create action for multiple IPs at once
- Possible to automatically add an IP address to the address list
- IP address can be added to the address list permanently or for a defined timeout period
- Address list can contain one IP address, an IP range or a whole subnet

IP → Firewall → Address Lists → +

Winbox



Queues

- Limit data rate for IP addresses, subnets, protocols, ports, interfaces and other parameters
- Prioritize some packet flows over others
- Configure traffic bursts for faster web browsing
- Share available traffic among users equally, depending on the load
- Limit peer-to-peer traffic

Queue Properties

- **limit-at**: Normal data rate that is guaranteed to a target
- **max-limit**: Maximal data rate that is allowed for a target to reach, if available
- **burst-threshold**: Basically, this is burst on/off switch
- **burst-limit**: Maximal data rate which can be reached while the burst is active
- **burst-time**: Period of time over which the average data rate is calculated (not the time of actual burst)

Queue Properties

- **parent**: Top queue in the HTB that assigns bandwidth to the child queues below
- **priority**: Responsible for distribution of remaining parent queue's traffic to child queues so that they are able to reach max-limit
- **packet marks**: Use marked packets from Firewall → Mangle
- **queue type**: Choose queue type created Queue → Type
- **bucket size**: A function of token bucket & max-limit

Simple Queues

- The simplest way to limit data rate for specific IP addresses, interfaces and/or subnets
  - Limit client's download (↓) speed, upload (↑) speed and total speed (↓ + ↑) independently
- All rules are processed sequentially from top

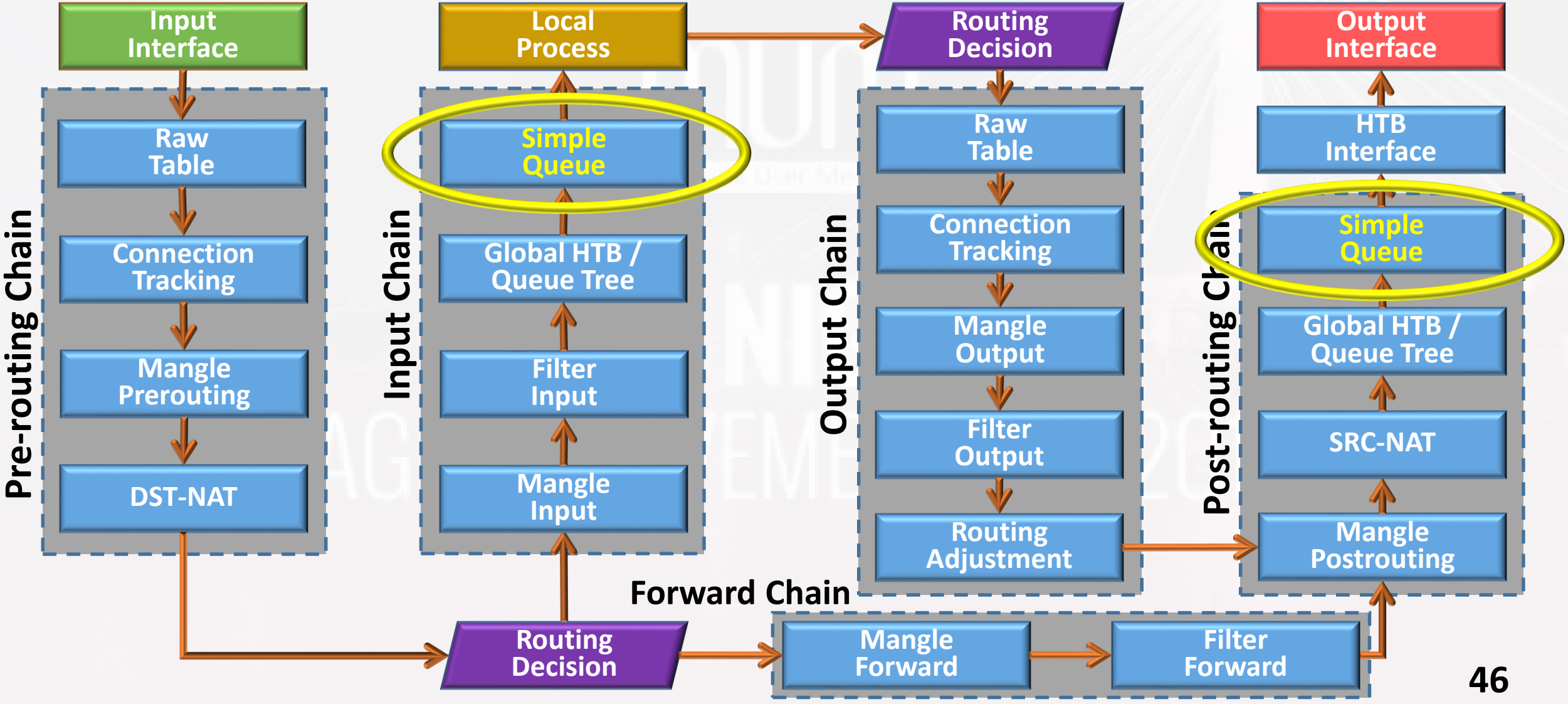
Simple Queues (ROS v.6)

- Not only for “simple” tasks anymore
- Traffic identified based on src-address, interface, dst-address, etc, hence no need to mark packets
- Fast hash algorithm, especially on multicore hardware
  - Number of simple queues no more relevant
  - Can have thousands of them without heavily loading CPU
- Can be easily created dynamically or by scripting

Simple Queues Caution

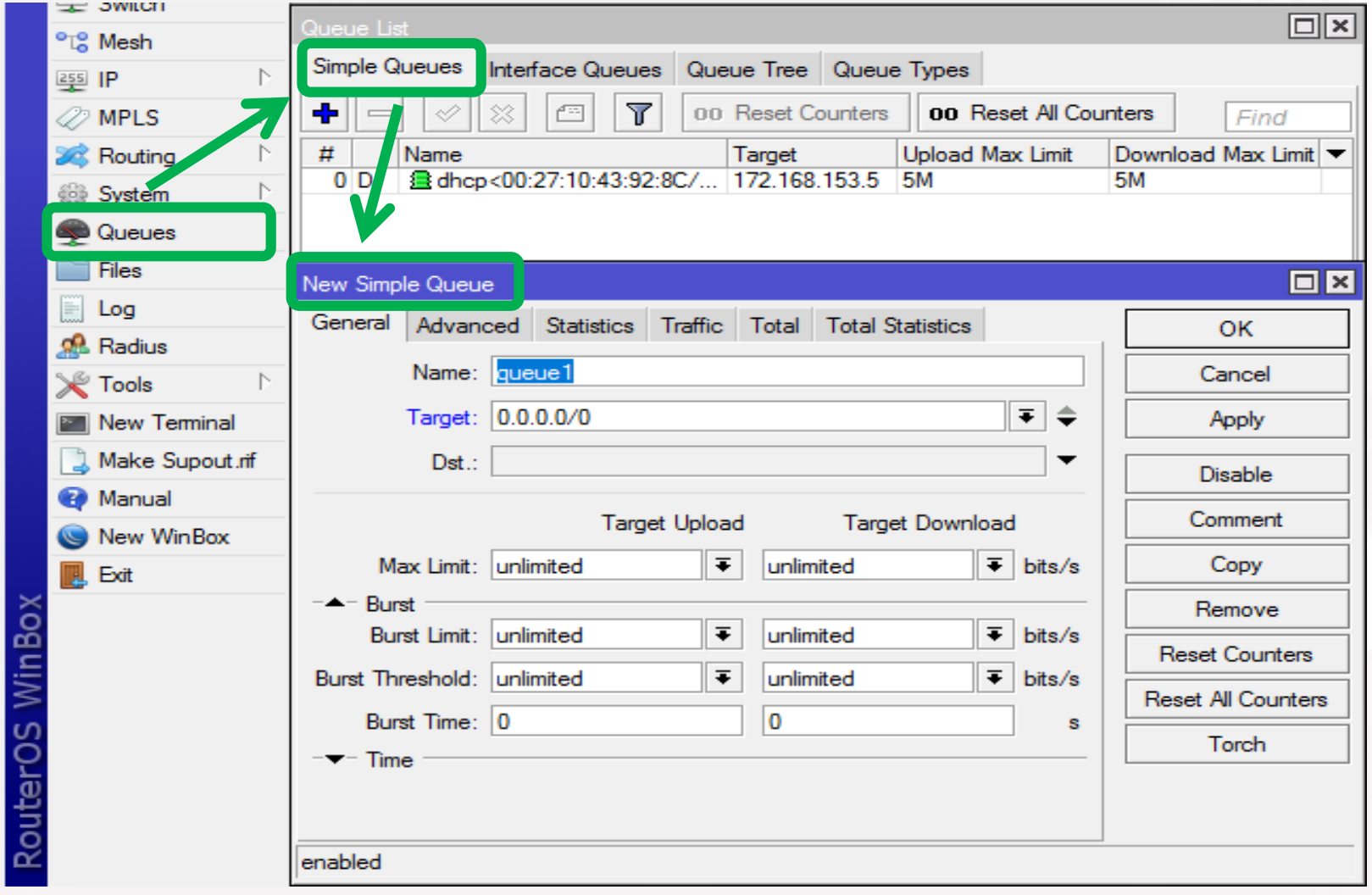
- FastTrack rule in Firewall → Filter needs to be disabled for Simple Queues to work

Simple Queue in PFD (ROS v.6)



Queues → Simple Queues → +

Winbox



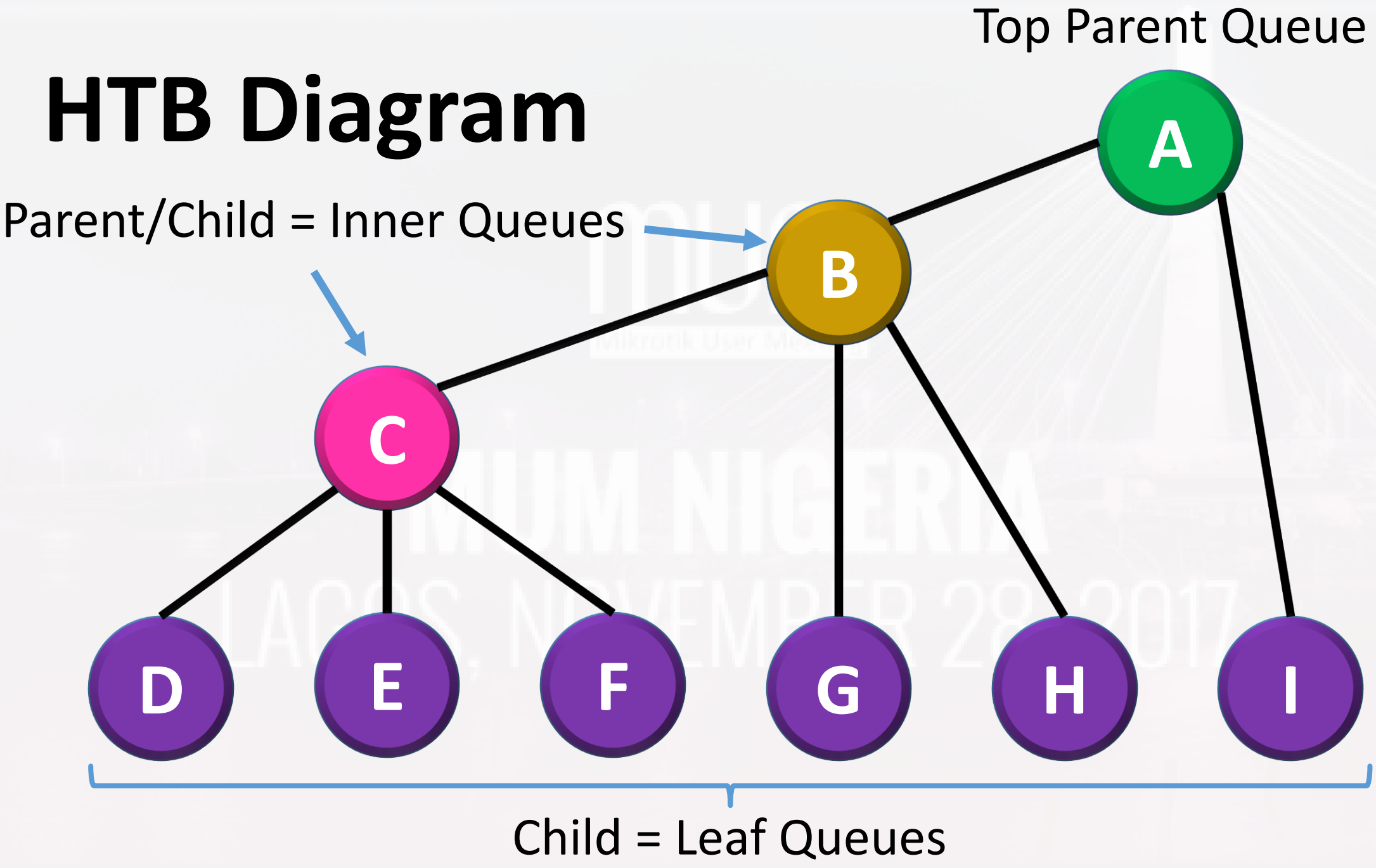
Hierarchical Token Bucket (HTB)

- Classful queuing method that is useful for handling different kinds of traffic
- Allows to create a hierarchical queue structure
- Determines relations between queues
  - Like "parent-child" or "child-child“
  - Priority, burst possibility, etc

HTB in RouterOS

- Bandwidth management implementation in RouterOS is mostly based on HTB
- Three basic steps required to create HTB:
  - Match and mark traffic
  - Create rules/policies to mark traffic
  - Attach policy for specific interface(s)

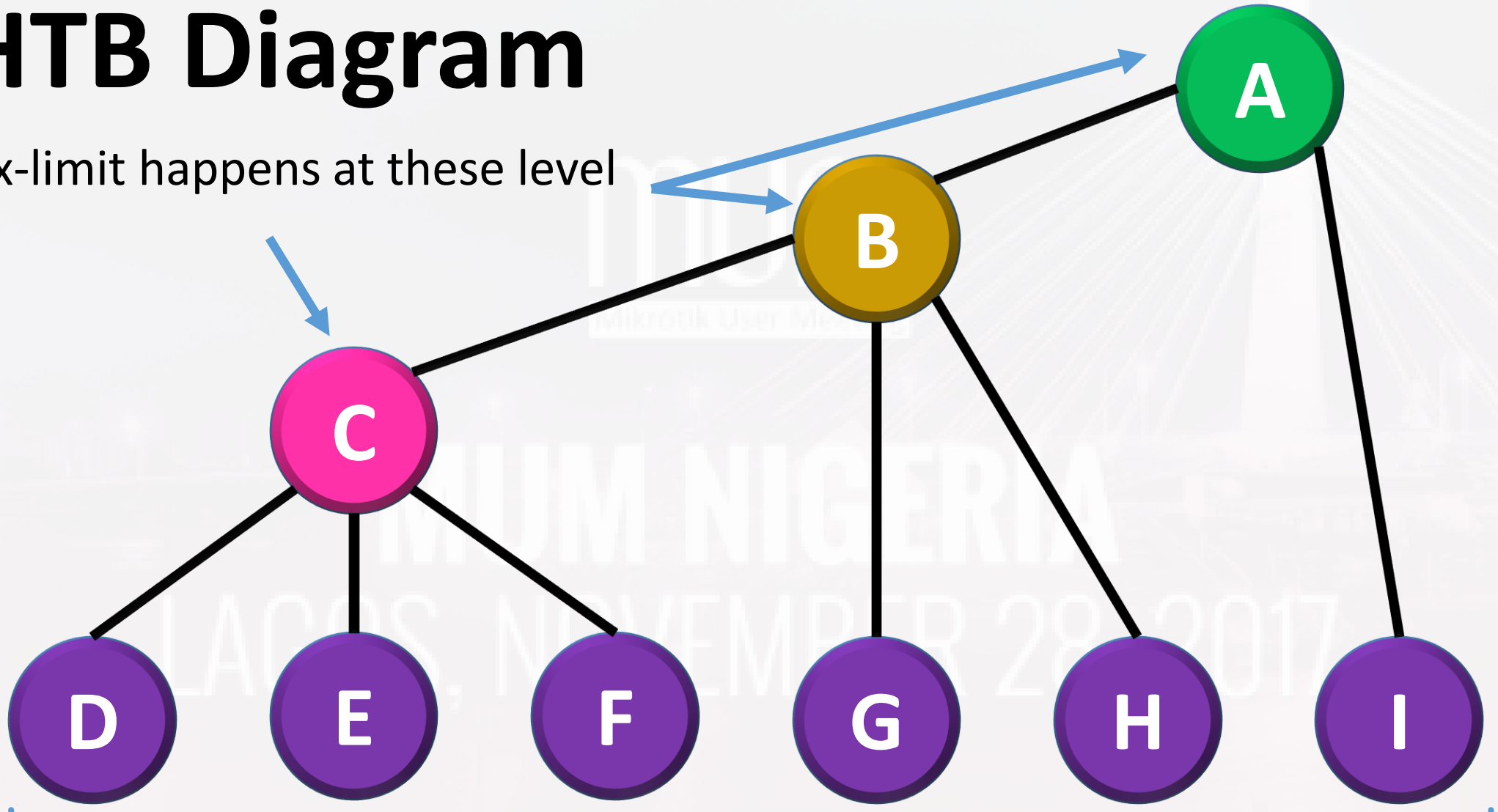
HTB Diagram


Top Parent Queue
Parent/Child = Inner Queues
A
B
C
D
E
F
G
H
I
Child = Leaf Queues

HTB Diagram

Top Parent Queue

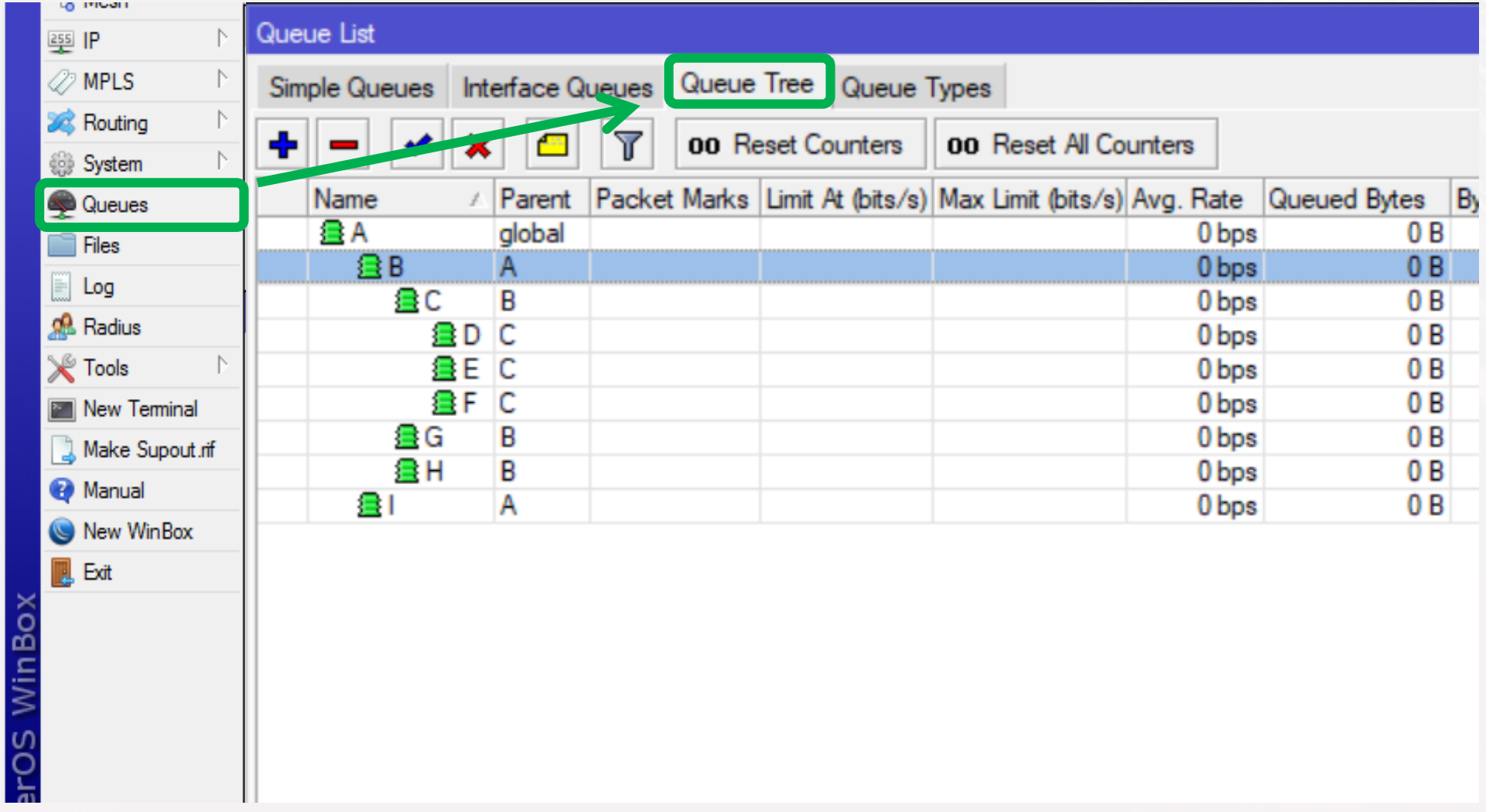
Max-limit happens at these level



Limit-at and priority happen at this level

Queues → Queue Tree → +

Winbox



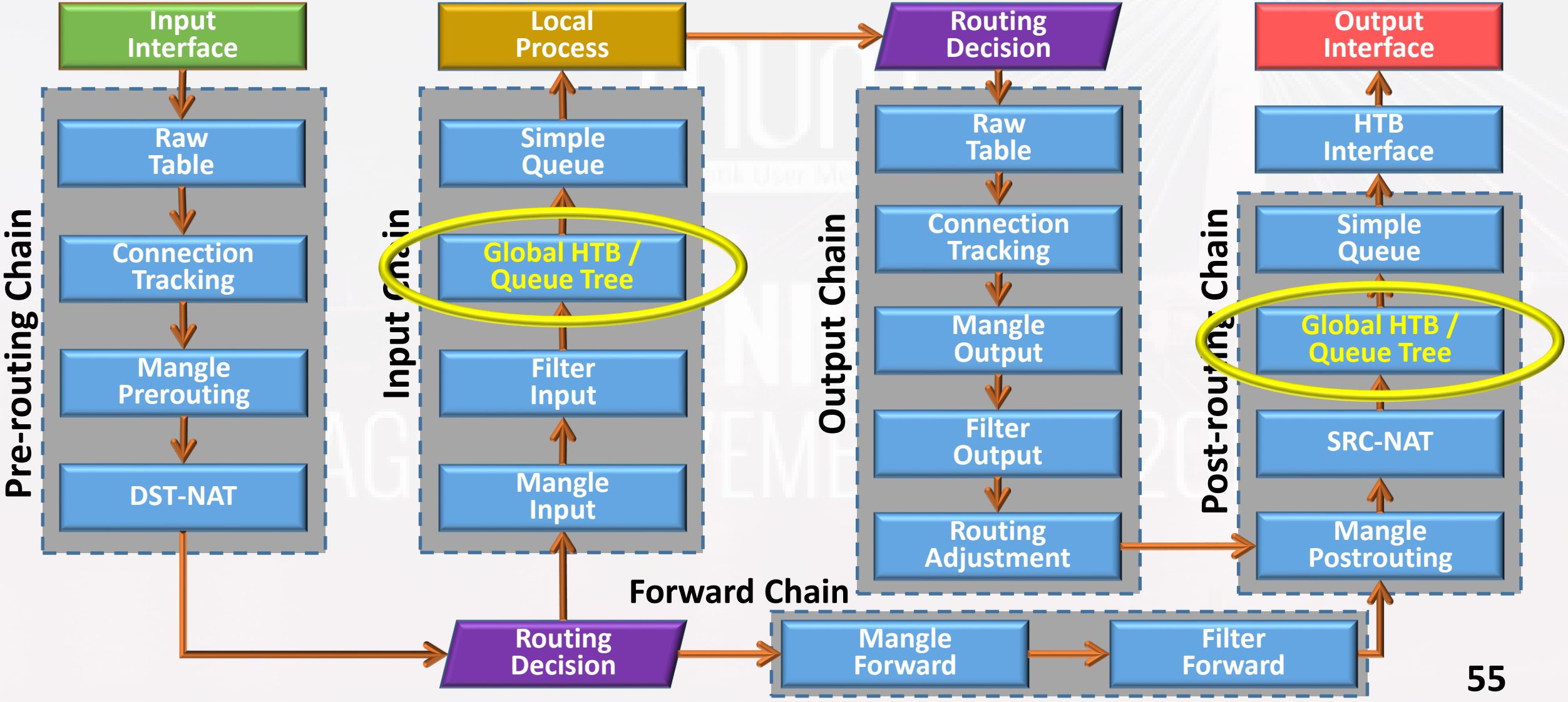
HTB Configuration Caution

- Limit-at and priority will not work if there is no parent in the hierarchy
- Limit-at and priority happen at the leaf queues

Queue Trees

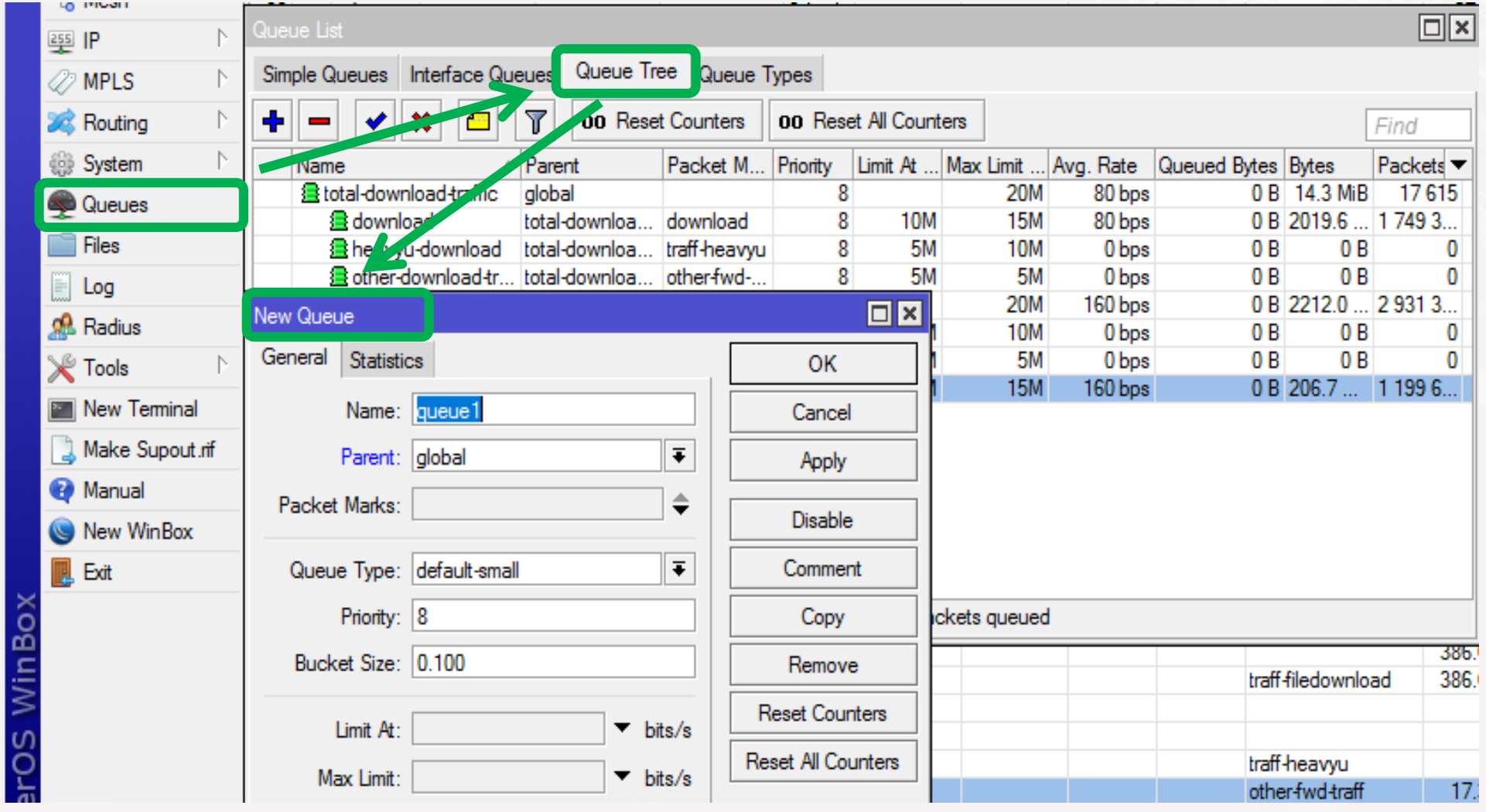
- Unidirectional queue in one of the HTBs
- Requires 2 rules per full duplex traffic control
- All rules are processed simultaneously
  - Highly efficient on CPU load
- Parent, priority & packet mark are very important for efficient operation

Queue Tree in PFD (ROS v.6)



Queues → Queue Tree → +

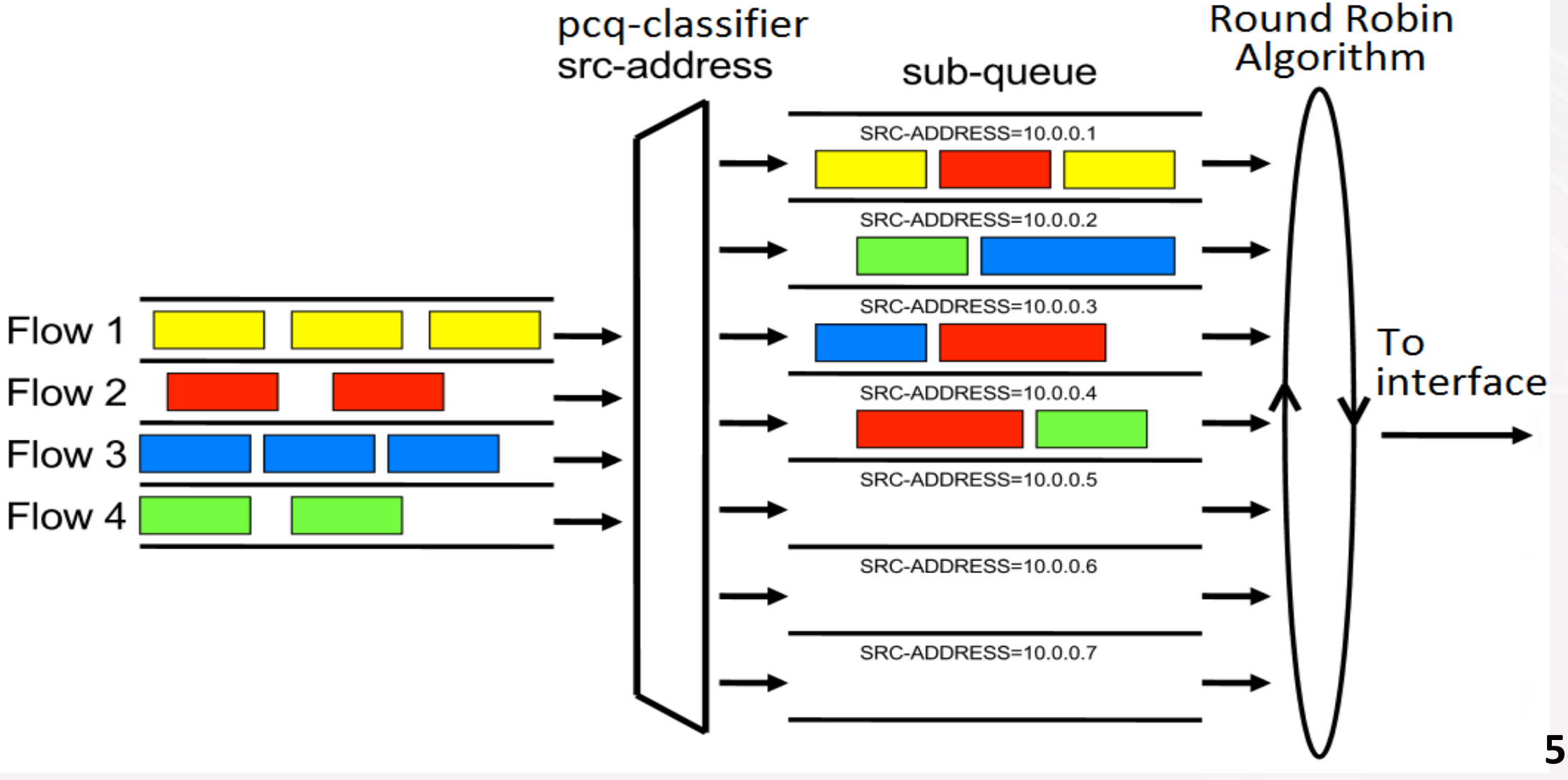
Winbox



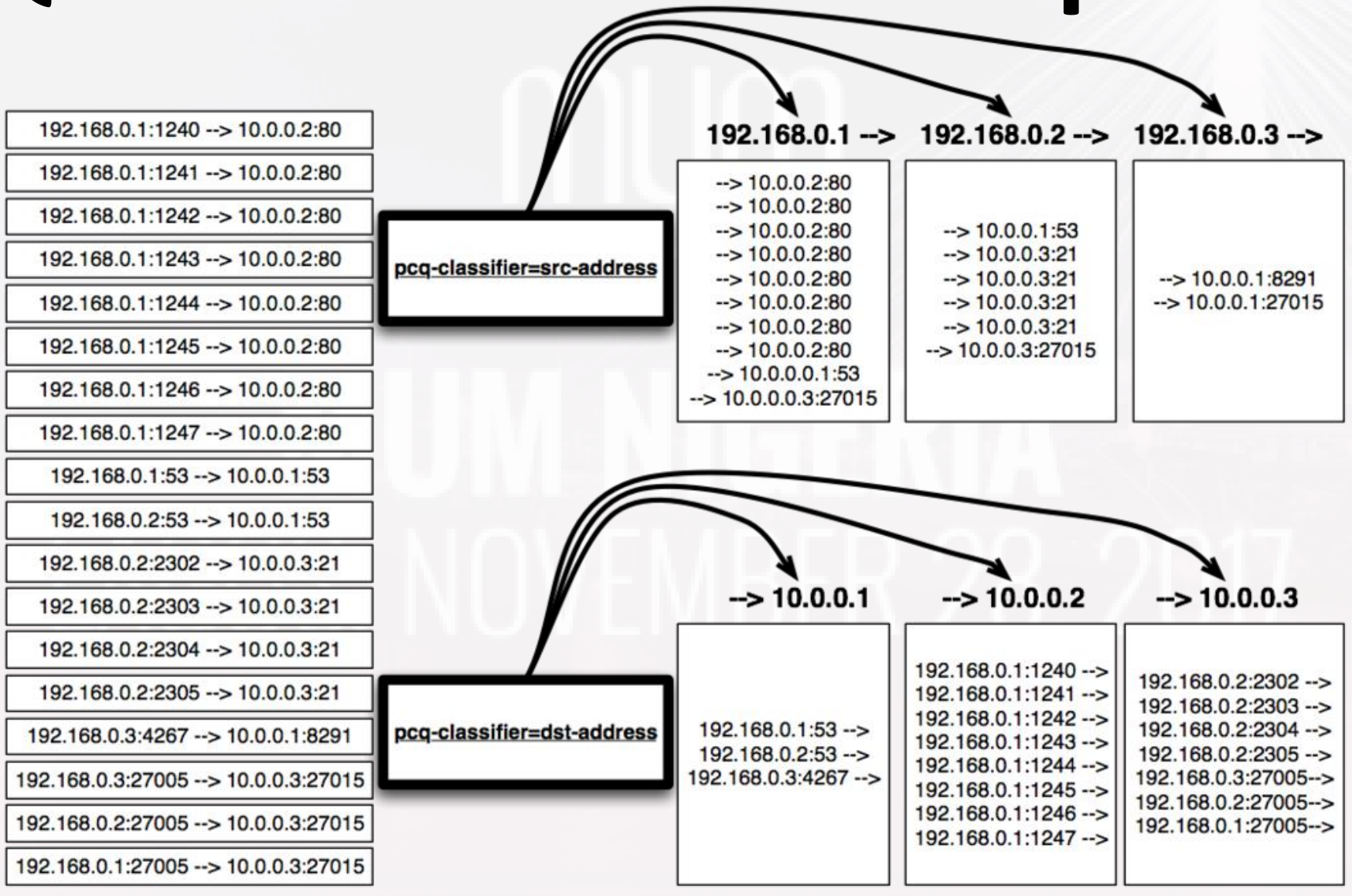
Per Connection Queue (PCQ)

- Queue type for optimizing large QoS deployments by limiting per ‘sub-stream’
- Substitute multiple queues with one
- Using flow identifiers to differentiate traffic
- Several classifiers can be used:
  - Source/destination IP address
  - Source/destination port

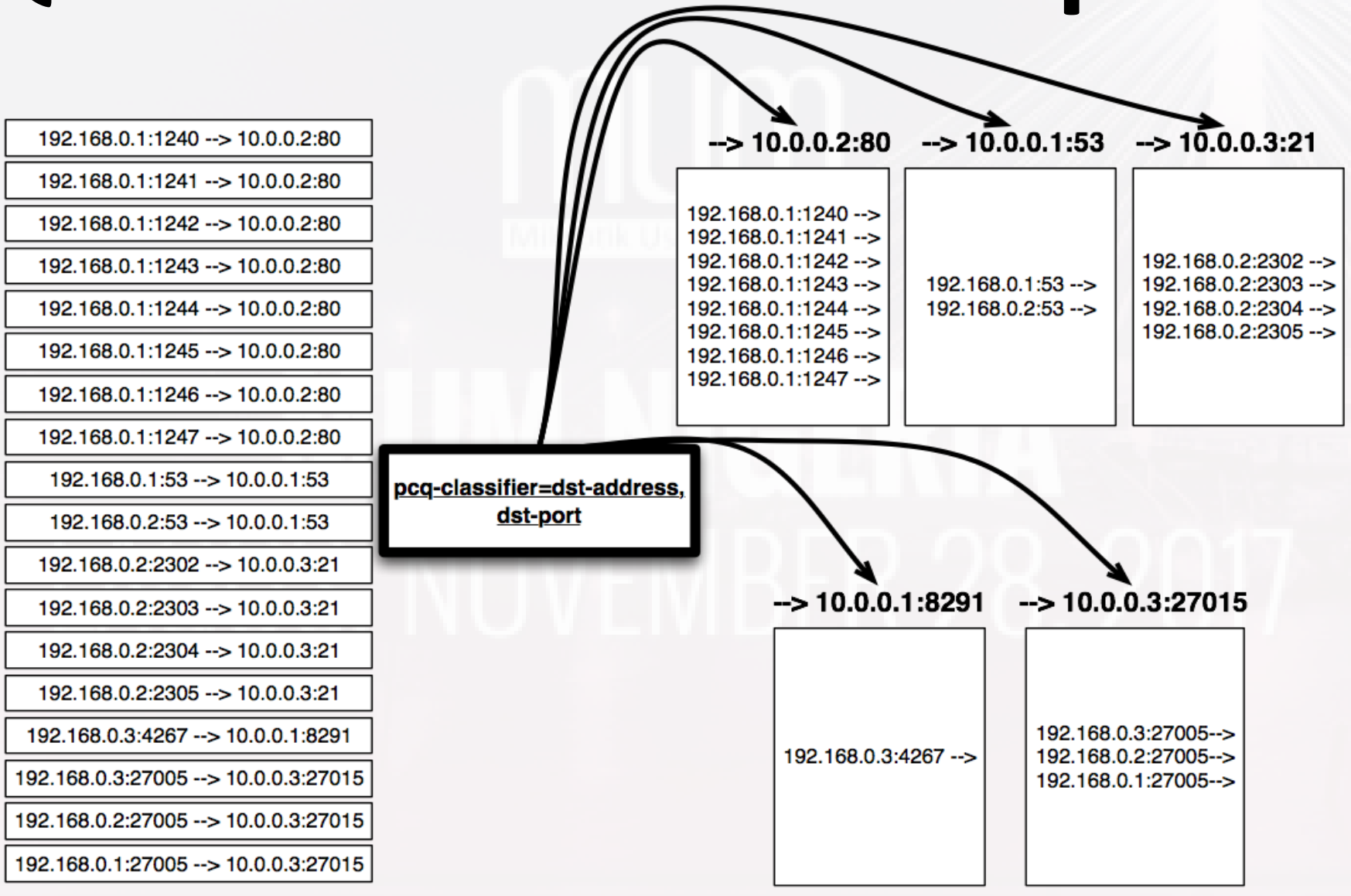
PCQ Flow


pcq-classifier
src-address
sub-queue
Round Robin
Algorithm
Flow 1
Flow 2
Flow 3
Flow 4
SRC-ADDRESS=10.0.0.1
SRC-ADDRESS=10.0.0.2
SRC-ADDRESS=10.0.0.3
SRC-ADDRESS=10.0.0.4
SRC-ADDRESS=10.0.0.5
SRC-ADDRESS=10.0.0.6
SRC-ADDRESS=10.0.0.7
To
interface

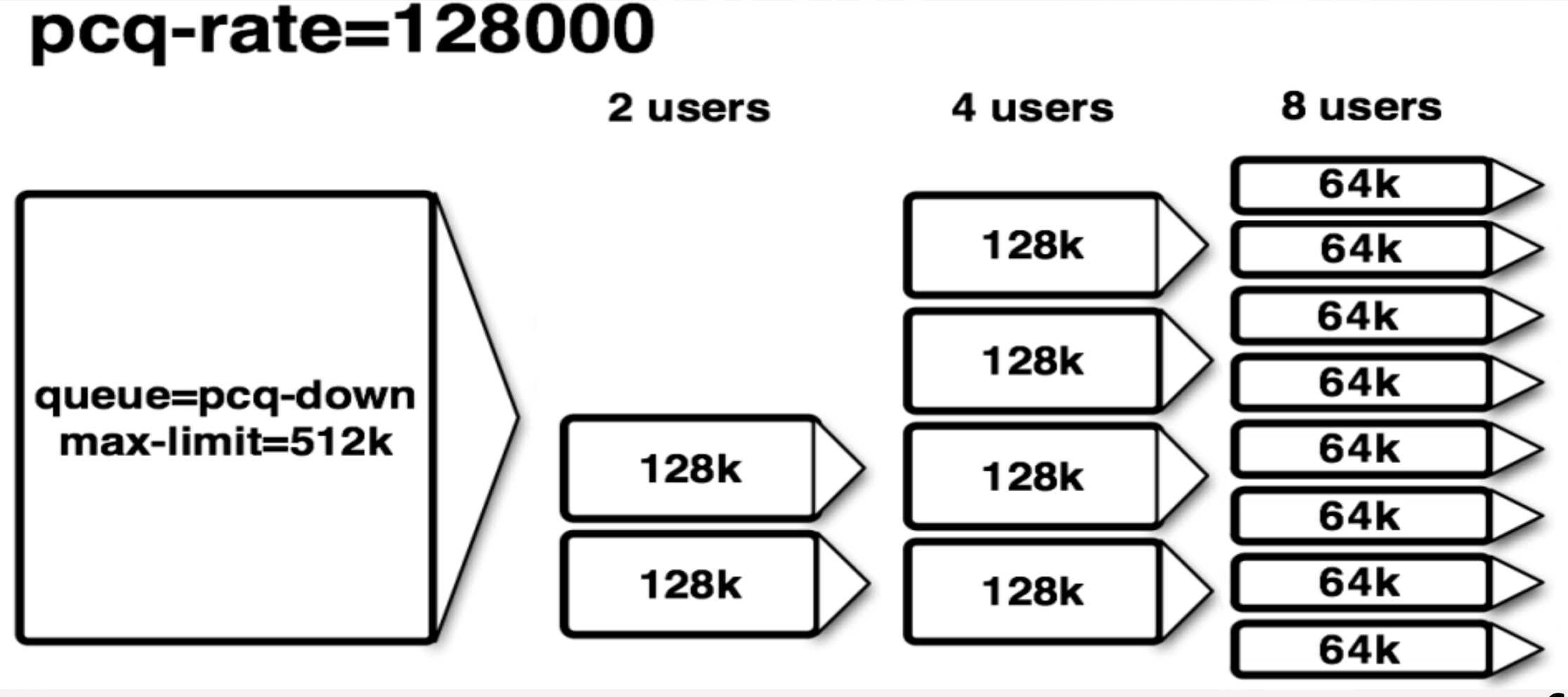
PCQ Classification Example



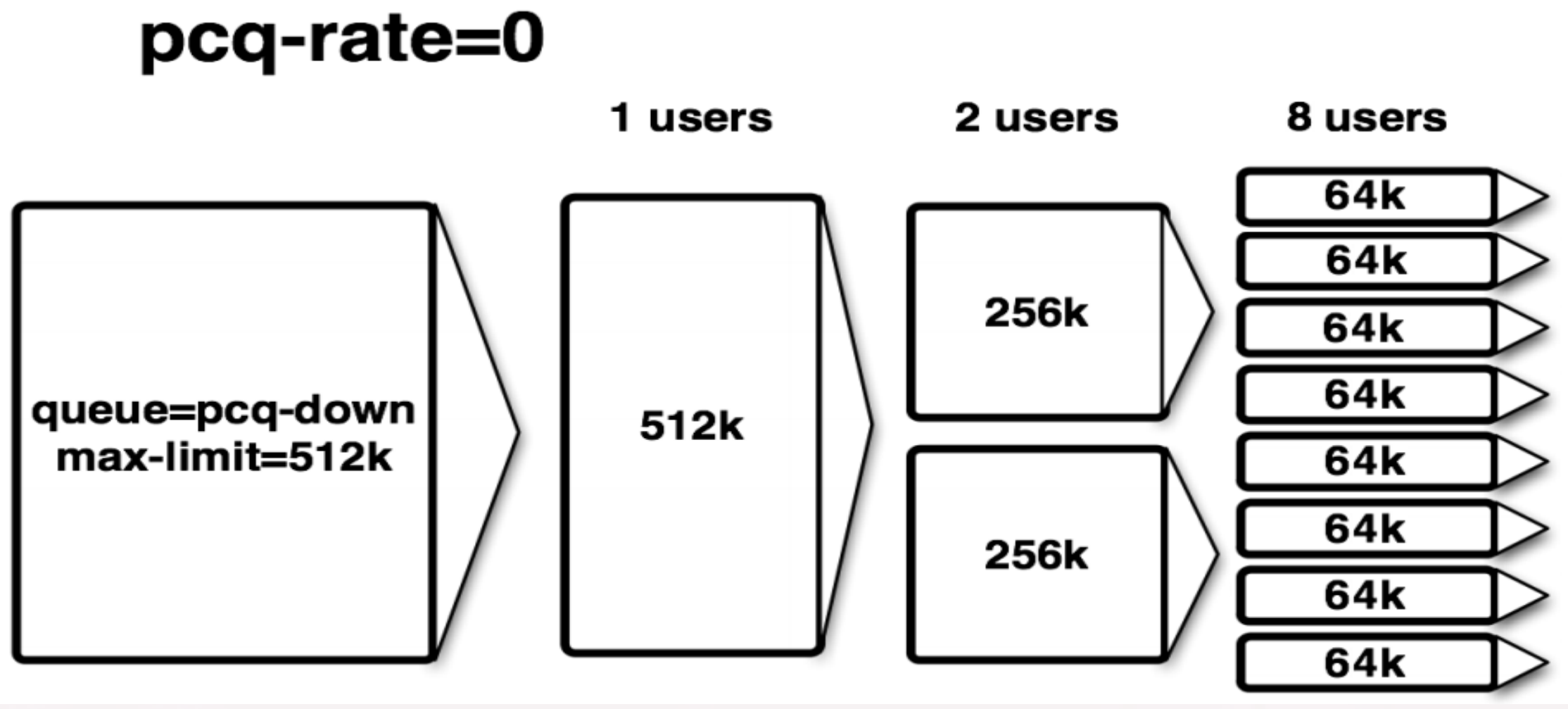
PCQ Classification Example



PCQ in Action

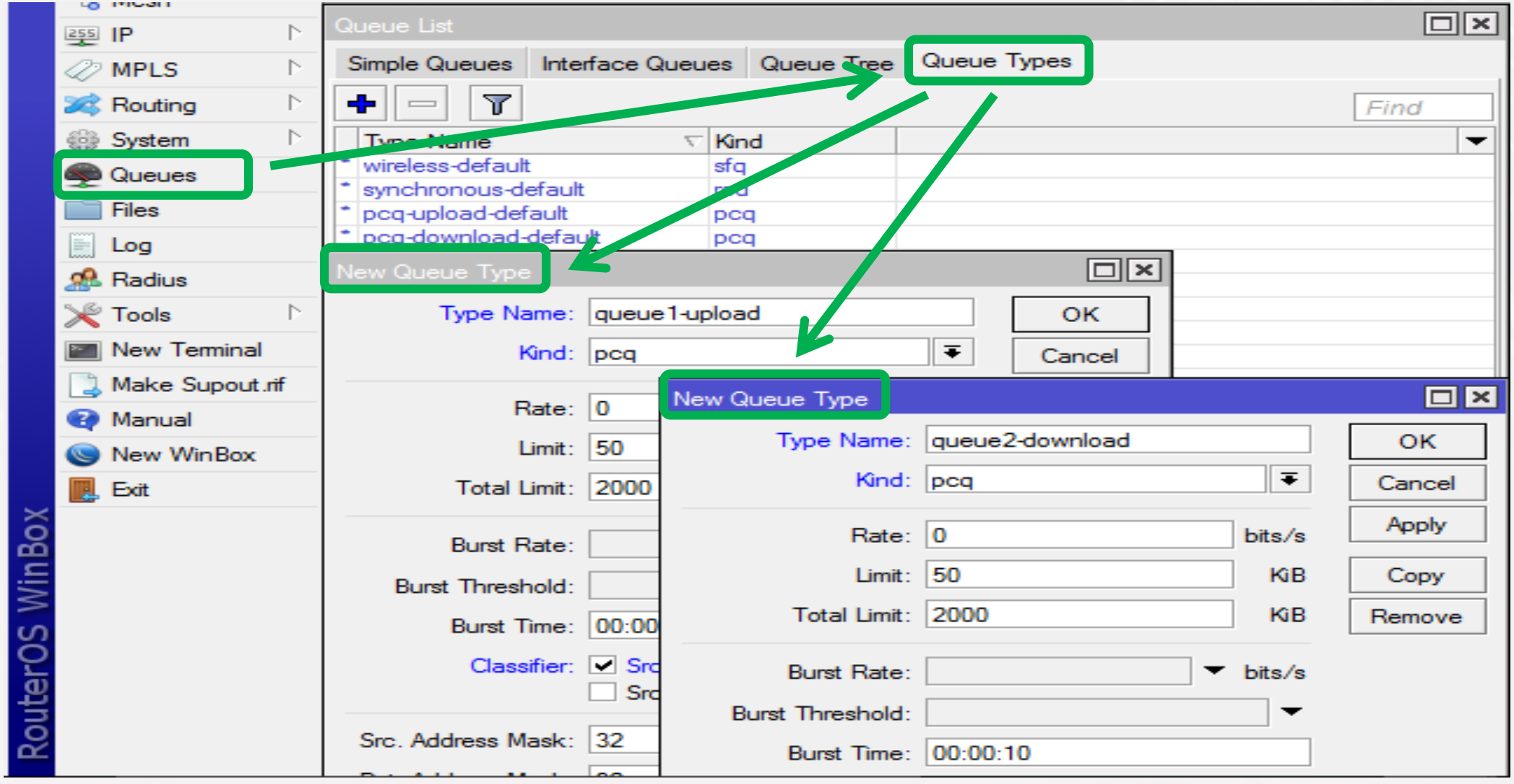


PCQ in Action



Queues → Queue Types → +

Winbox



PCQ Configuration Caution

- PCQ takes its data from Connection Tracking
  - Connection Tracking must be enabled
- If both limits (pcq-rate and max-limit) are unspecified, queue behavior can be imprecise
- So it is strongly suggested to have at least one of these options set

Simple Queues & Queue Tree Combo

- A major aspect of QoS is traffic prioritization
- A major aspect of Bandwidth Management is client limitation
- Prioritization can be done on Forward Mangle & Queue Tree
- Limitation can be done with Simple Queue by targeting IP address

Implementation Summary

- We used Mangle, Address List, Simple Queues and Queue Tree:
  - Mark packets by traffic type in Mangle Chain
  - Classify clients by IP address in Address List
  - Prioritize and limit traffic by type in Global HTB
  - Limit traffic per client in Simple Queue
  - Utilize PCQ to ensure a good Quality of Experience

Conclusion

- MikroTik RouterOS is one of the most advanced (and easy to configure) OS for bandwidth management and QoS
- Understanding the packet flow diagram is required to set the ball rolling
- A knowledgeable combination of the right tools (mangle, simple queue, queue tree, etc) will bring about the desired results

Acknowledgements

- This presentation would not have been possible without the vision behind TikTube
  - Thank you MikroTik & the Latvia crew
  - Please, kindly rename your YouTube Channel to TikTube
- Valens Riyadi
  - King of QoS
  - He has several related video presentations & PDF slides online

Thank you

Questions