

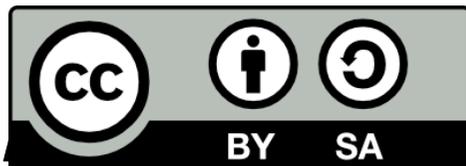


HTB vs PCQ

by: **Valens Riyadi**

Citraweb Nusa Infomedia, Indonesia

www.mikrotik.co.id



Introduction



- **Valens Riyadi**
- Work for Citraweb/Citranet
 - Mikrotik distributor, training partner
 - ISP, web developer
- Using Mikrotik since 2.3.15 (2001)
- MTCNA, MTCTCE, MTCUME, MTCRE, MTCWE, MTCINE, Certified Trainer

Remote Access

- Remote Access to my router:
 - SSID : MUM-QOS
 - IP Address : 10.3.2.1
 - Username : demo
 - Password : [empty]

Bandwidth Management

- MikroTik RouterOS is one of the most advanced (and easy to configure) OS/application for bandwidth management.
- Bandwidth management done by utilize shaper and scheduler
 - Shaper : HTB and PCQ
 - Scheduler : FIFO, RED, SFQ

Question

- Which one is better, HTB or PCQ?
- When we need to use HTB, or PCQ?

Hierarchical Token Bucket (HTB)

- Hierarchical Token Bucket (HTB) allows to create a hierarchical queue structure and determine relations between queues, like "parent-child" or "child-child".

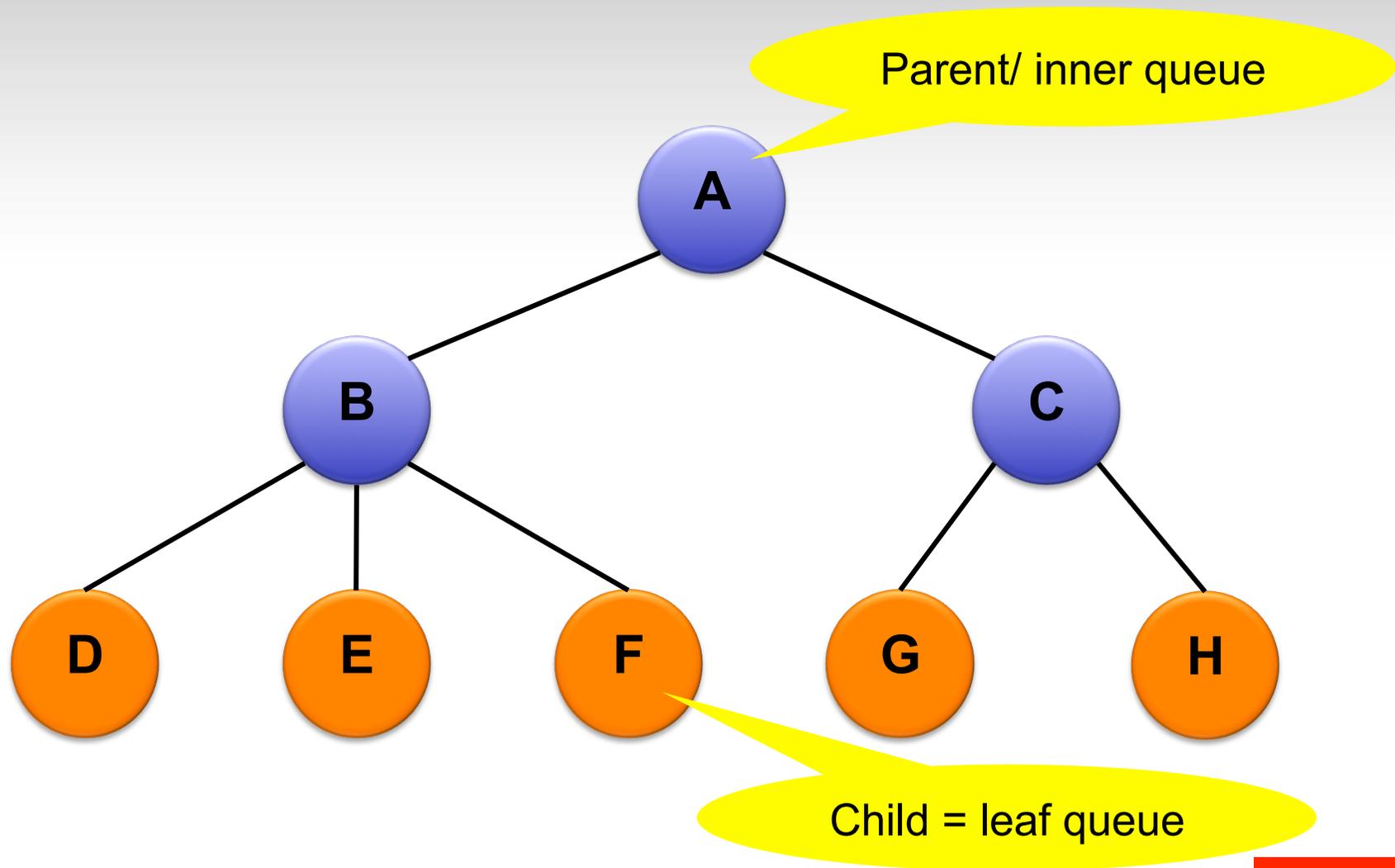
Basic Concept

- HTB (Hierarchical Token Bucket) is part of QoS, to make a hierarchical queue structure and determine relations between queues (priority, burst possibility, etc)
- HTB is meant as a more understandable, intuitive and faster replacement for the CBQ qdisc in Linux.
- HTB assigned to any physical interface or virtual interface (global-in, global-out, global-total)

HTB Features

- Hierarchy
 - Almost no hierarchy limit, the limit is your imagination
- Grouping
 - We can group several clients, into one parent
 - One client can borrow bandwidth from another client in same group, if needed.
- Independent setting for each leaf queue

HTB Sample



HTB Sample

Queue List

Simple Queues Interface Queues Queue Tree Queue Types

Reset Counters Reset All Counters Find

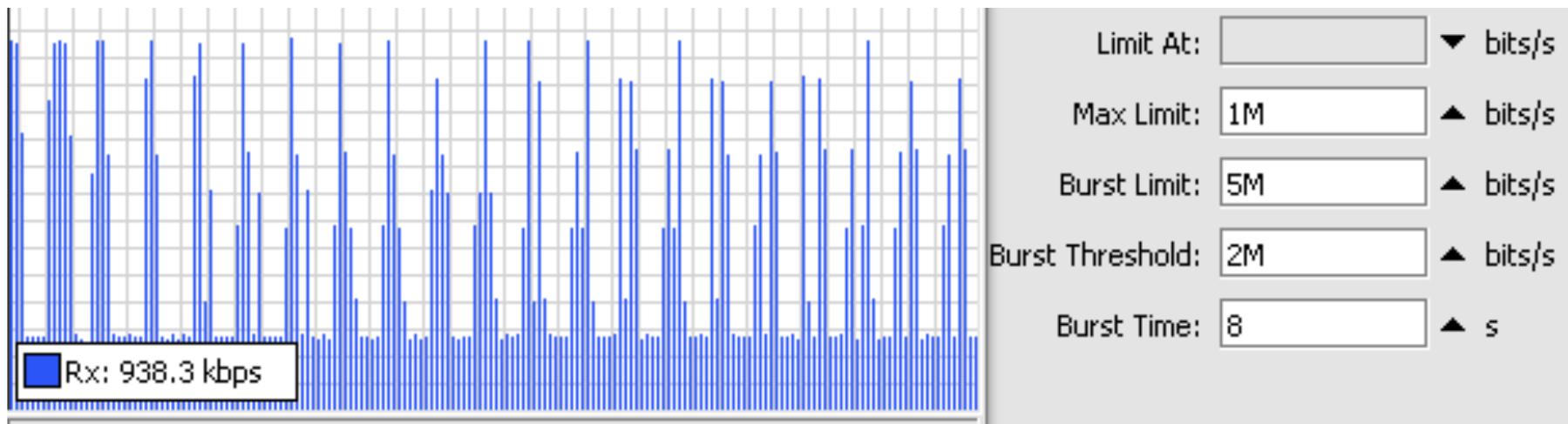
Name	Parent	Packet Marks	Limit At (bits/s)	Max Limit (bits/s)	Avg. R...	Queued Bytes	Bytes	Packets
queue_0	ether2			10M	0 bps	0 B	0 B	0
queue_1	queue_0	packet_1551	468904	9800k	0 bps	0 B	0 B	0
queue_2	queue_1	packet_9220	268289	9600k	0 bps	0 B	0 B	0
queue_17	queue_2	packet_8507	613074	6600k	0 bps	0 B	0 B	0
queue_41	queue_17	packet_8440	371117	1800k	0 bps	0 B	0 B	0
queue_6	queue_2	packet_2299	687353	8800k	0 bps	0 B	0 B	0
queue_18	queue_6	packet_1165	366627	6400k	0 bps	0 B	0 B	0
queue_23	queue_18	packet_1093	538294	5400k	0 bps	0 B	0 B	0
queue_3	queue_0	packet_3333	166813	9400k	0 bps	0 B	0 B	0
queue_16	queue_3	packet_6309	529294	6800k	0 bps	0 B	0 B	0
queue_27	queue_16	packet_8970	562428	4600k	0 bps	0 B	0 B	0
queue_46	queue_27	packet_1154	420425	800k	0 bps	0 B	0 B	0
queue_31	queue_16	packet_8523	563538	3800k	0 bps	0 B	0 B	0
queue_37	queue_16	packet_8389	376173	2600k	0 bps	0 B	0 B	0
queue_40	queue_37	packet_8521	704484	2M	0 bps	0 B	0 B	0
queue_42	queue_40	packet_2889	430111	1600k	0 bps	0 B	0 B	0
queue_39	queue_16	packet_8281	227458	2200k	0 bps	0 B	0 B	0
queue_22	queue_3	packet_9689	370291	5600k	0 bps	0 B	0 B	0
queue_43	queue_22	packet_9101	607074	1400k	0 bps	0 B	0 B	0
queue_8	queue_3	packet_3057	644987	8400k	0 bps	0 B	0 B	0
queue_9	queue_3	packet_9444	433143	8200k	0 bps	0 B	0 B	0
queue_35	queue_9	packet_6885	149412	3M	0 bps	0 B	0 B	0
queue_44	queue_9	packet_6940	508058	1200k	0 bps	0 B	0 B	0
queue_4	queue_0	packet_1485	587640	9200k	0 bps	0 B	0 B	0
queue_5	queue_4	packet_8908	661059	9M	0 bps	0 B	0 B	0
queue_13	queue_5	packet_8132	746955	7400k	0 bps	0 B	0 B	0
queue_26	queue_13	packet_8397	692964	4800k	0 bps	0 B	0 B	0
queue_34	queue_13	packet_1227	483167	3200k	0 bps	0 B	0 B	0
queue_36	queue_13	packet_7635	412515	2800k	0 bps	0 B	0 B	0



limit-at and priority
work only if you use
hierarchy (parent)

Independent setting for leaf

- We can set different limit and burst for each leaf.
- Certain burst parameter will make “normal” customer think their bandwidth fast.

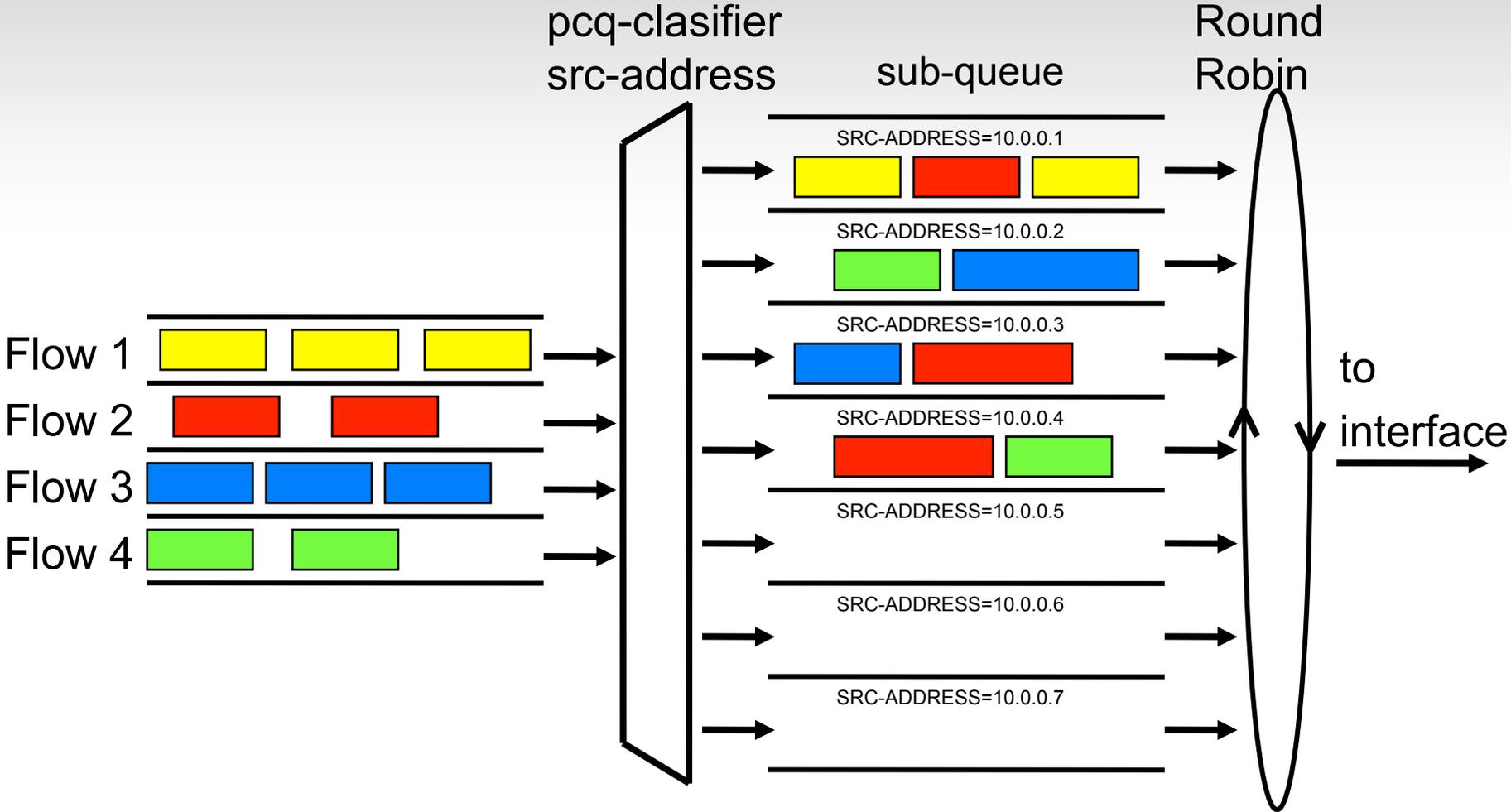


- More detail explanation about HTB :
My presentation MUM USA 2009
 - PDF : <http://bit.ly/aotax9>
 - Video : <http://bit.ly/bFrRiP>

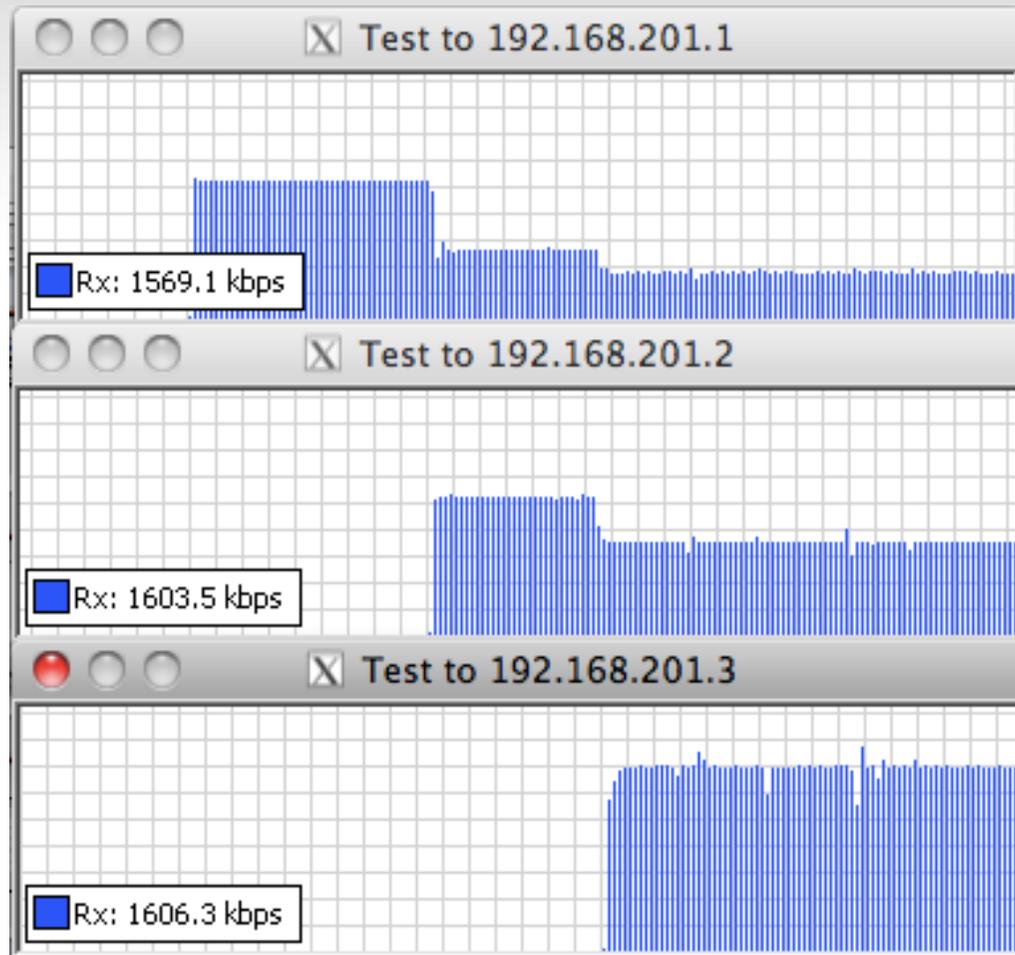
Per Connection Queuing (PCQ)

- Using flow identifiers (dst-address, dst-port, src-address or src-port) to differentiate traffic into sub-streams.
- Introduced to optimize massive QoS systems, where most of the queues are exactly the same for each sub-streams
- 1 rule can handle hundreds customer, and limit them individually

PCQ Flow



PCQ Sample



Queue <queue1>

General Statistics

Name:

Parent:

Packet Marks:

Queue Type:

Priority:

Limit At: bits/s

Max Limit: bits/s

Burst Limit: bits/s

Burst Threshold: bits/s

Queue Type <queue-pcq>

Type Name:

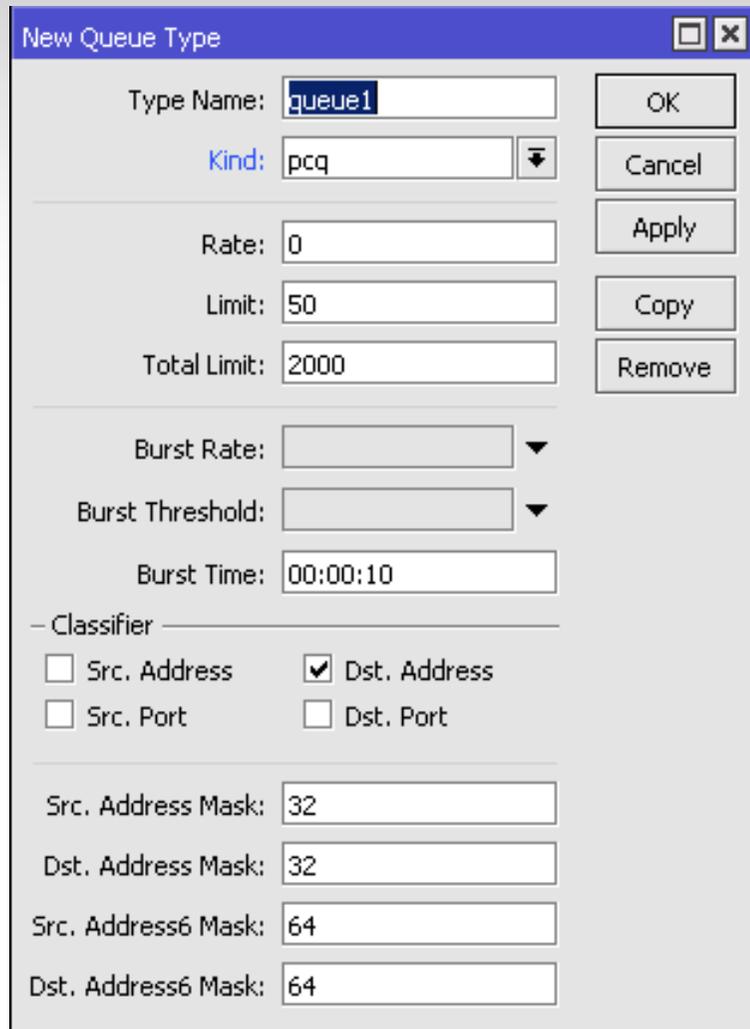
Kind:

Rate:

Limit:

Total Limit:

PCQ Configuration



New Queue Type

Type Name:

Kind:

Rate:

Limit:

Total Limit:

Burst Rate:

Burst Threshold:

Burst Time:

– Classifier –

Src. Address Dst. Address

Src. Port Dst. Port

Src. Address Mask:

Dst. Address Mask:

Src. Address6 Mask:

Dst. Address6 Mask:

Buttons: OK, Cancel, Apply, Copy, Remove

New parameters since
RoS 5rc4

- Burst
- Mask (for IPv4 and IPv6)

PCQ Configuration

Rate:	<input type="text" value="0"/>
Limit:	<input type="text" value="50"/>
Total Limit:	<input type="text" value="2000"/>

- Rate = sub-stream max-limit
- Total-limit (packet in queue) = Limit * number of sub-stream
- Ex: 100 customers, 100 packets per customer (limit), and total-limit=10,000

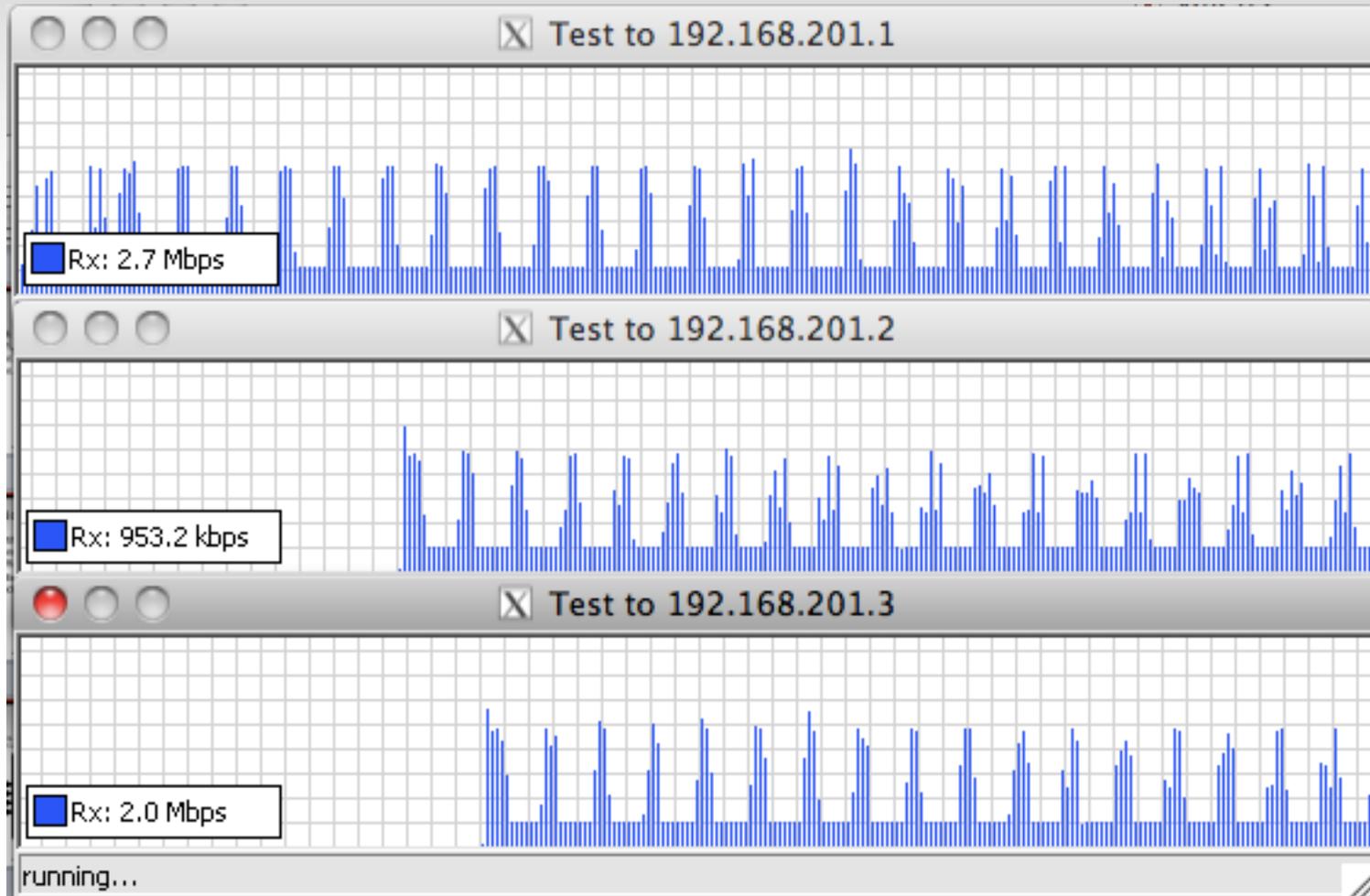


Burst on Sub-Stream

Queue <queue1>		Queue Type <queue-pcq>	
General	Statistics	Type Name:	queue-pcq
Name:	queue1	Kind:	pcq
Parent:	wlan1	Rate:	1M
Packet Marks:	packet-queue	Limit:	50
Queue Type:	queue-pcq	Total Limit:	2000
Priority:	8	Burst Rate:	5M
Limit At:		Burst Threshold:	2M
Max Limit:	20M	Burst Time:	00:00:10
Burst Limit:		- Classifier -	
Burst Threshold:		<input checked="" type="checkbox"/> Src. Address	<input type="checkbox"/> Dst. Address
Burst Time:	8	<input type="checkbox"/> Src. Port	<input type="checkbox"/> Dst. Port

NEW

Burst on Sub-Stream





Burst on Sub-Stream

Src. Address Mask:	<input type="text" value="32"/>
Dst. Address Mask:	<input type="text" value="32"/>
Src. Address6 Mask:	<input type="text" value="64"/>
Dst. Address6 Mask:	<input type="text" value="64"/>

- Now we can group customer that have same subnet size, not only /32
- PCQ now work in IPv6 also, and we can set the subnet

PCQ with Queue Tree

- PCQ (with src-address classifier) and Queue Tree (interface based) on NATed network will not work for uplink traffic.
 - Because interface queue for uplink located after src-nat process → all src-addresses become same
- Suggestions:
 - change the interface to global-in, or
 - use simple queue and set the interface parameter.

Note:

- In PCQ, 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.

PCQ with HTB

- In HTB, we can not set priority on inner queue (groups of clients)
- PCW rule is a leaf queue, we can set priority as the priority of group of clients
 - One group more prioritized than another

Name	Parent	Queue Type	Priority	Max Li...	Avg. R...
queue3-parent	wlan1	default	8	5M	4.9 Mbps
queue1	queue3-parent	default	8	5M	60.5 k...
queue2-pcq	queue3-parent	queue-pcq	1	5M	4.8 Mbps

Conclusions

- Faster configuration with PCQ, one rule for all clients (with same treatment)
- HTB used if clients have different speed and setting
- Since v5rc4, PCQ have sub-stream's burst parameter, this give similar function as in HTB
- MikroTik is ready to queue IPv4 and IPv6

Thank You!

info@mikrotik.co.id

