MUM-KH-2017
Phnom Penh

QoS on RouterOS with Token Bucket

RouterOS v6.35

Valens Riyadi (Citraweb)
@valensriyadi

Valens Riyadi

- MikroTik Certified Engineer
  (MTCNA, MTCWE, MTCRE, MTCTCE, MTCUME, MTCINE)
- MikroTik Certified Trainer since 2004, Certified Consultant, and Academy Coordinator
- CEO for Citranet (WISP) and Citraweb (Mikrotik Distributor and System Developer)
- Manager for IDNIC (Indonesia National Internet Registry) 2009-2015
- Expert on Cyber Crime, Digital Forensic, IT for Disaster Relief, Live Streaming
- Proud member of “Routed World” community

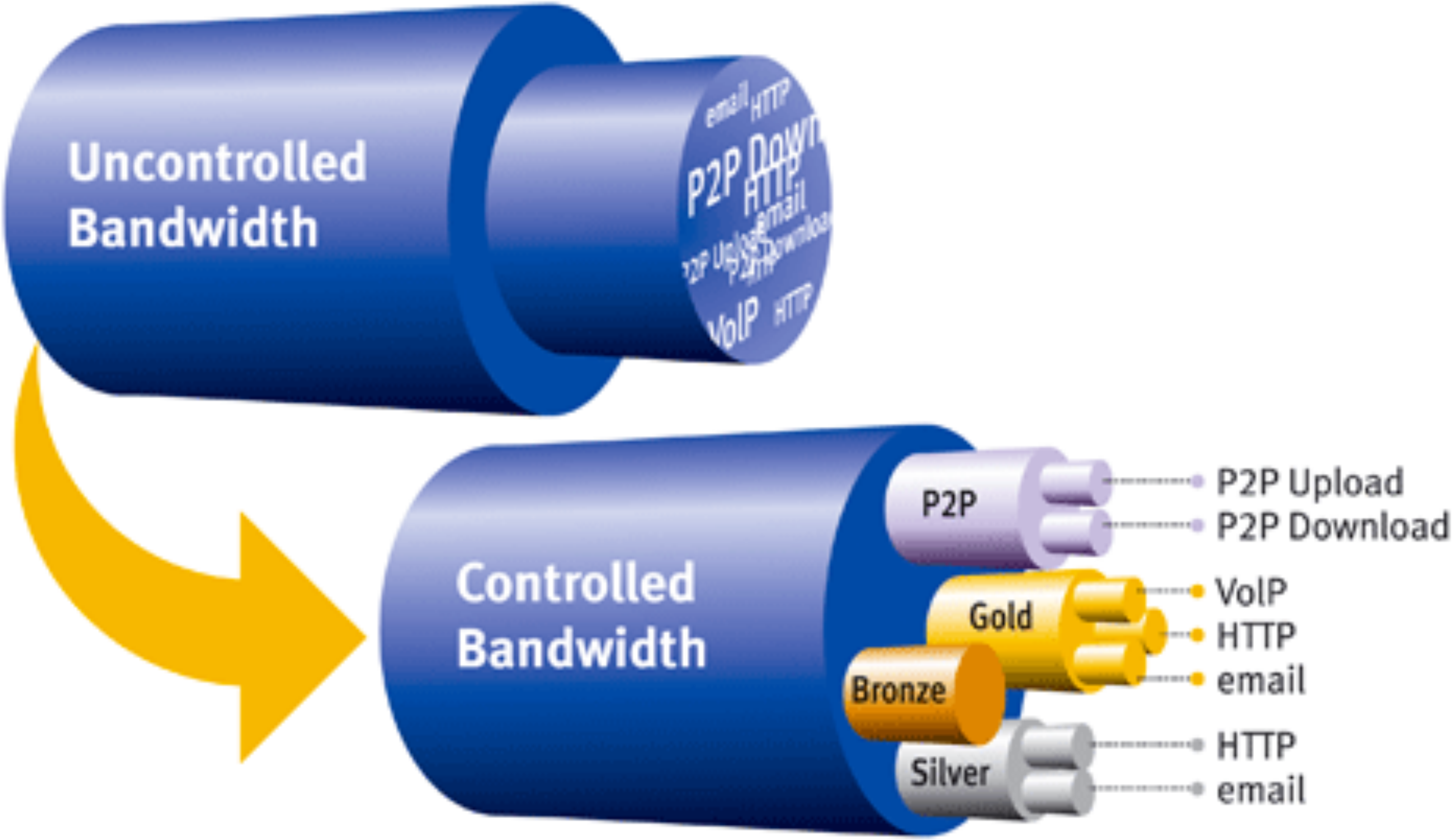
mikrobits
CELOICA 8101
AINOS 2071
AINOS
ANETO
DINARA
CELOICA
MikroTik
mikrobits

MikroTik Training Center

- The first MikroTik Training Center in Asia Pasific since 2004, more then 5500 participants (237 classes).
- Mikrotik Academy Coordinator.

Mikrotik Training for Indonesia Special Forces

Why do we need to manage bandwidth?



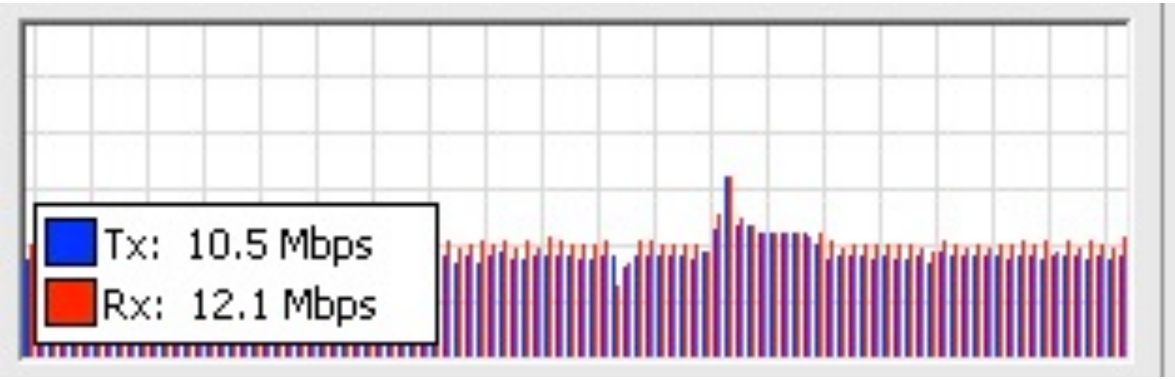
QoS on RouterOS

- MikroTik RouterOS is **one of the most advanced bandwidth management**, compared to any other brand.
- Why?
  - Advanced HTB configuration
  - Double limitation + Burst
  - A lot of option and parameter → packet-mark
    - Grouping, protocol, layer 7, connection size, traffic detection, etc

Burst

A mechanism to be able to provide additional bandwidth to a particular client if:

- Client not always on max-limit
- Additional bandwidth still available on related parent and interface



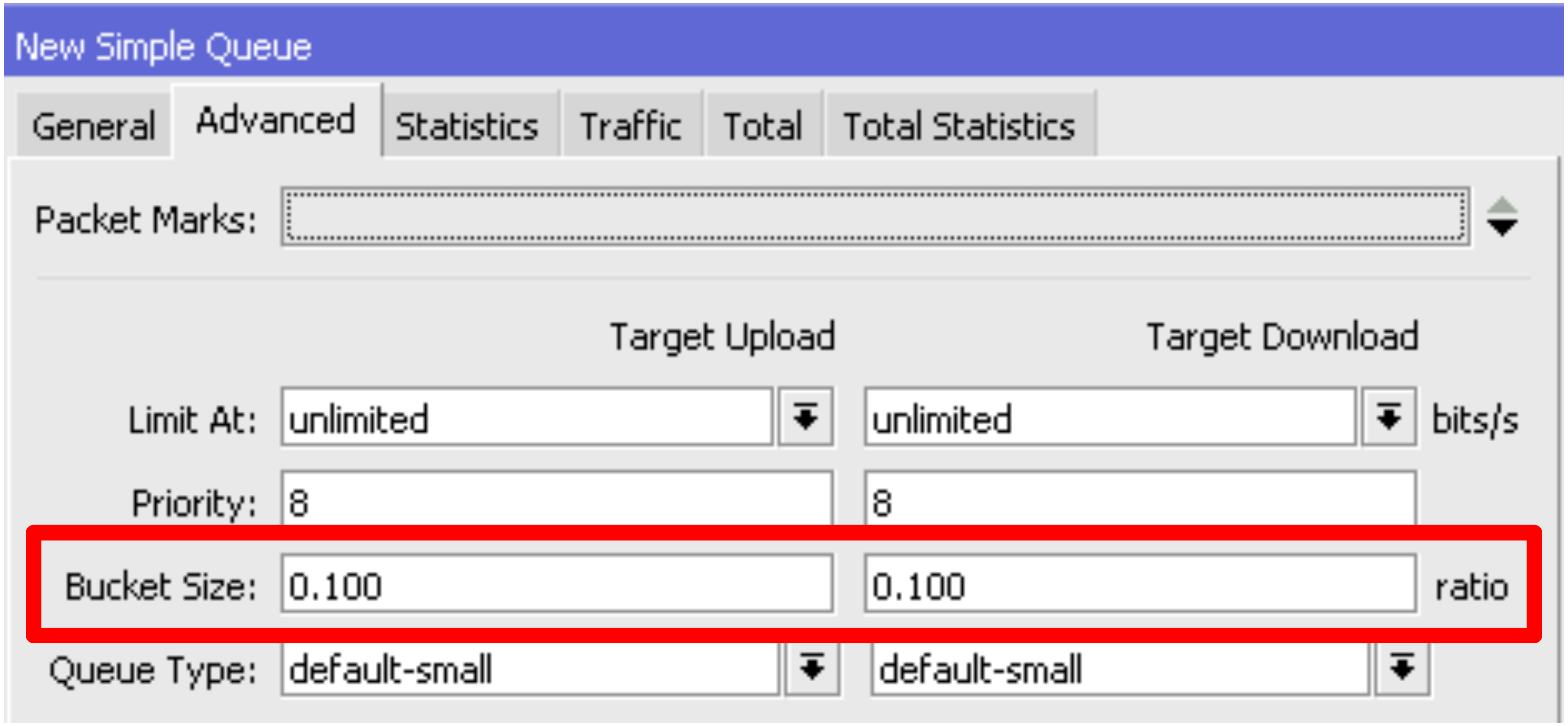
Happy customer

Burst will significantly improve customer experience as they feel the connection is fast

Token Bucket

- Since RouterOS v6.35, MikroTik introduce Token Bucket.
- It’s been on the RouterOS engine for long time, but it’s hidden and statically set to 0.1.
- Concept of Token Bucket also implemented on Linux.
- It works like burst feature, but much simpler.

Winbox Configuration

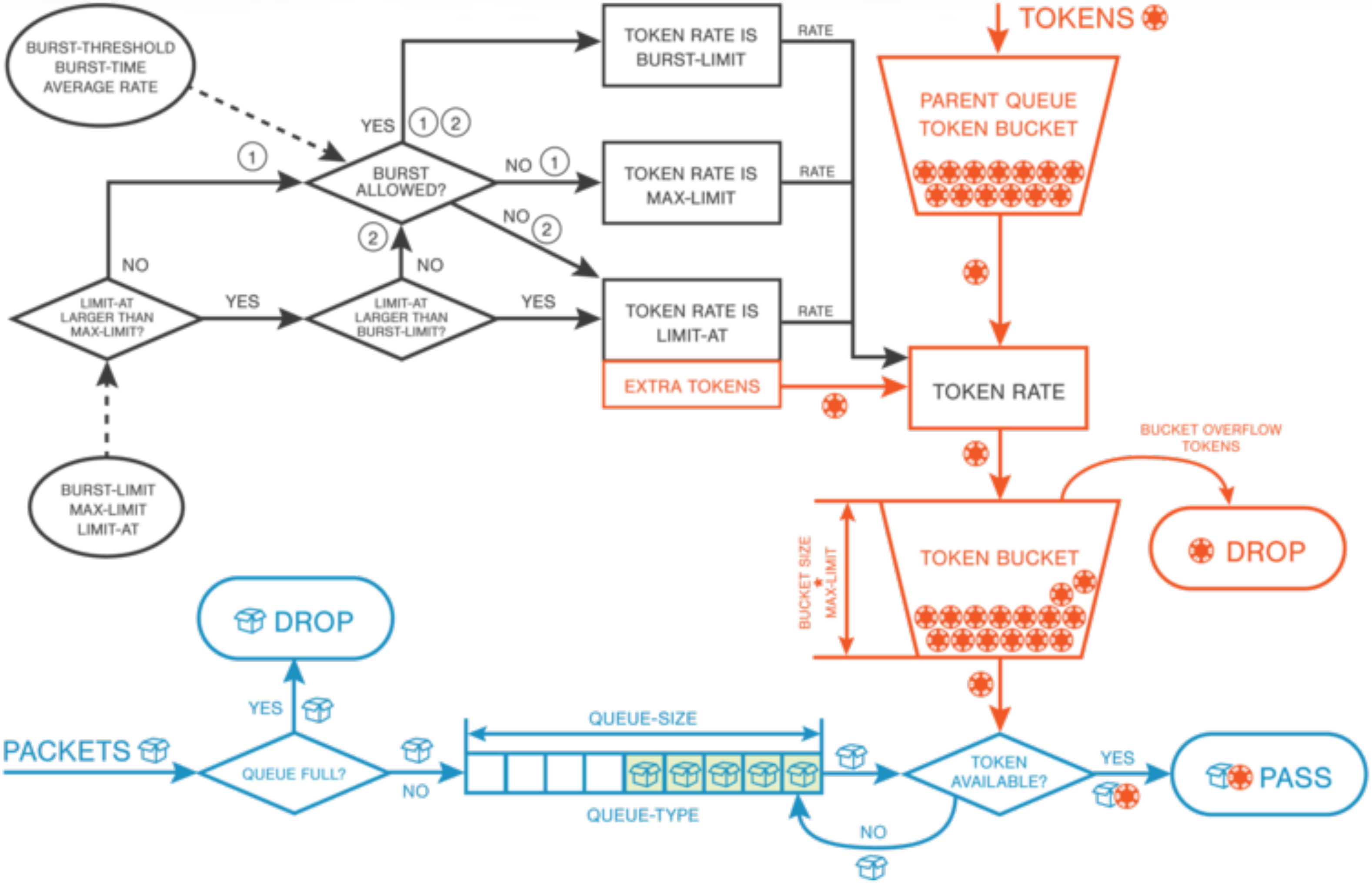


New Simple Queue

Packet Marks:

	Target Upload	Target Download	
Limit At:	unlimited	unlimited	bits/s
Priority:	8	8	
Bucket Size:	0.100	0.100	ratio
Queue Type:	default-small	default-small	

Token Bucket Algoritm



http://wiki.mikrotik.com/wiki/Manual:HTB-Token_Bucket_Algorithm

Bucket

- **Bucket capacity** is **bucket size ratio** compared to **max-limit**.
- Example:
  - – Max-limit = 5Mbps
  - – Bucket size ration = 3 (max 10)
  - – Bucket capacity = 5 * 3 = 15 Mbit
- When the calculation start, bucket is always considered full of token.

Token rate

token-rate = max-limit

token-rate = limit-at

if limit-at on child > max-limit on parent

token-rate = burst-limit

if burst is active and allowed

Token Bucket Algoritm

If traffic = token-rate	All traffic will be delivered
If traffic < token-rate	All traffic will be delivered, and token will be added to the bucket, as much as the differences
If traffic > token-rate	Traffic will be delivered, and token will be taken from the bucket, as much as the differences. If token is not enough/bucket is empty, traffic will be delivered as token-rate.

Token Bucket Example

- Max-limit = 20M, Bucket-size=10
  →Bucket-capacity = 20 * 10 = 200Mbit
- Router will delivered additional (after token-rate) 200Mbit traffic without any limitation.
- 30Mbps will be delivered in ……. seconds:

Bucket Capacity / (Actual Speed - Token Rate)

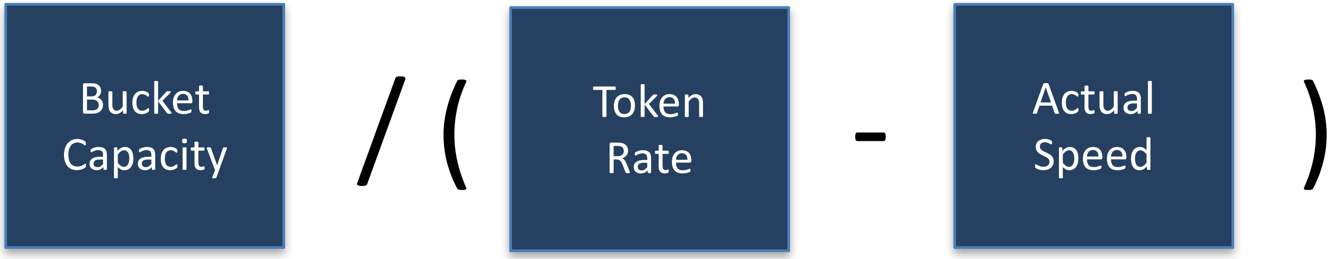
200 / (30 – 20) = 200 / 10 = 20 seconds

40Mbps will be delivered in ……. seconds:

Second(s)	Traffic	Token Deducted/ Speed Exceed	Token left at bucket	Speed Delivered
0	0		200 Mbit	
1	40 Mbps	20 Mbit	180 Mbit	40 Mbps
2	40 Mbps	20 Mbit	160 Mbit	40 Mbps
3	40 Mbps	20 Mbit	140 Mbit	40 Mbps
4	40 Mbps	20 Mbit	120 Mbit	40 Mbps
5	40 Mbps	20 Mbit	100 Mbit	40 Mbps
6	40 Mbps	20 Mbit	80 Mbit	40 Mbps
7	40 Mbps	20 Mbit	60 Mbit	40 Mbps
8	40 Mbps	20 Mbit	40 Mbit	40 Mbps
9	40 Mbps	20 Mbit	20 Mbit	40 Mbps
10	40 Mbps	20 Mbit	0	40 Mbps
11	40 Mbps	0	0	20 Mbps

Token Bucket Example

- After the bucket empty, traffic will be limited to token-rate (20M)
- And if client utilize lower then token-rate, for example only 15Mbps, bucket will be full of tokens after ….. seconds:



Bucket Capacity / (Token Rate - Actual Speed)

200 / (20 – 15) = 200 / 5 = 40 seconds

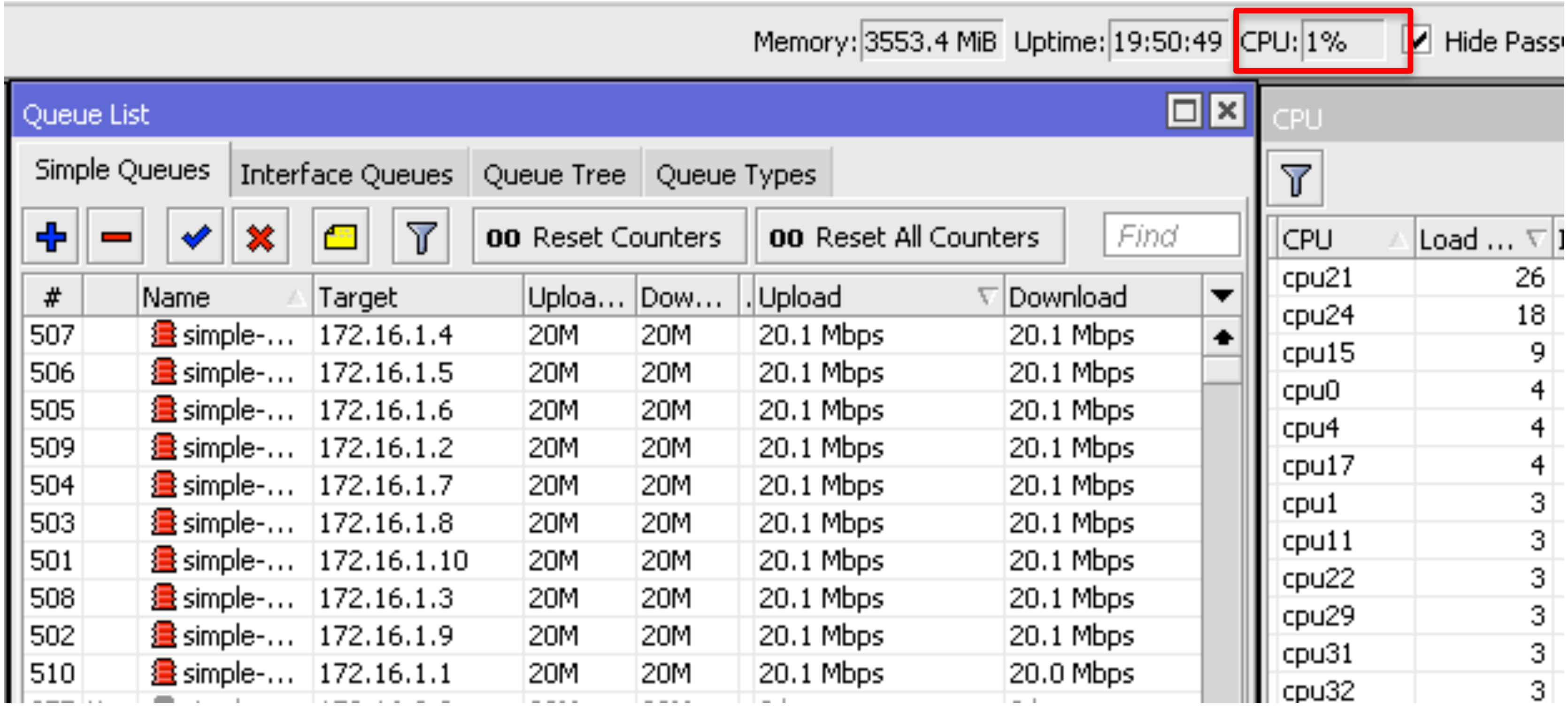
How about burst?

- In burst calculation, router have to remember speed per queue in the last (burst-time) seconds.
- In token-bucket, only token-rate and number of tokens in bucket will be considered.
- In high load application, token bucket might be more efficent then burst. Really?

Comparation

- Is token bucket really more efficient then burst system?
- It’s really difficult to test:
  - CPU load is not only related with queue configuration, but the size of traffic delivered by router
  - Queue is very efficient. 500+ simple queue with 500mbps only make 1% CPU load.

With Simple Queue



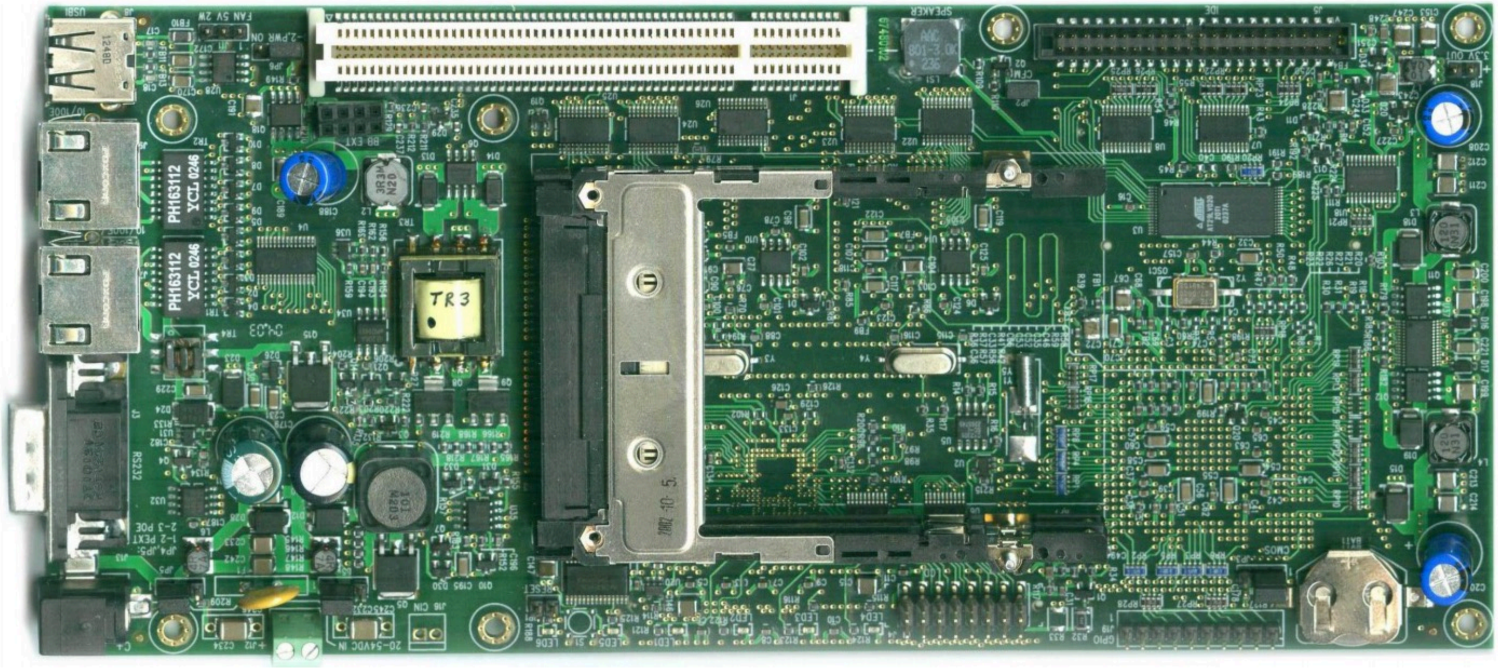
Memory: 3553.4 MiB Uptime: 19:50:49 CPU: 1% ☑ Hide Pass

#		Name	Target	Uploa...	Dow...	Upload	Download
507		simple-...	172.16.1.4	20M	20M	20.1 Mbps	20.1 Mbps
506		simple-...	172.16.1.5	20M	20M	20.1 Mbps	20.1 Mbps
505		simple-...	172.16.1.6	20M	20M	20.1 Mbps	20.1 Mbps
509		simple-...	172.16.1.2	20M	20M	20.1 Mbps	20.1 Mbps
504		simple-...	172.16.1.7	20M	20M	20.1 Mbps	20.1 Mbps
503		simple-...	172.16.1.8	20M	20M	20.1 Mbps	20.1 Mbps
501		simple-...	172.16.1.10	20M	20M	20.1 Mbps	20.1 Mbps
508		simple-...	172.16.1.3	20M	20M	20.1 Mbps	20.1 Mbps
502		simple-...	172.16.1.9	20M	20M	20.1 Mbps	20.1 Mbps
510		simple-...	172.16.1.1	20M	20M	20.1 Mbps	20.0 Mbps

CPU	Load ...
cpu21	26
cpu24	18
cpu15	9
cpu0	4
cpu4	4
cpu17	4
cpu1	3
cpu11	3
cpu22	3
cpu29	3
cpu31	3
cpu32	3

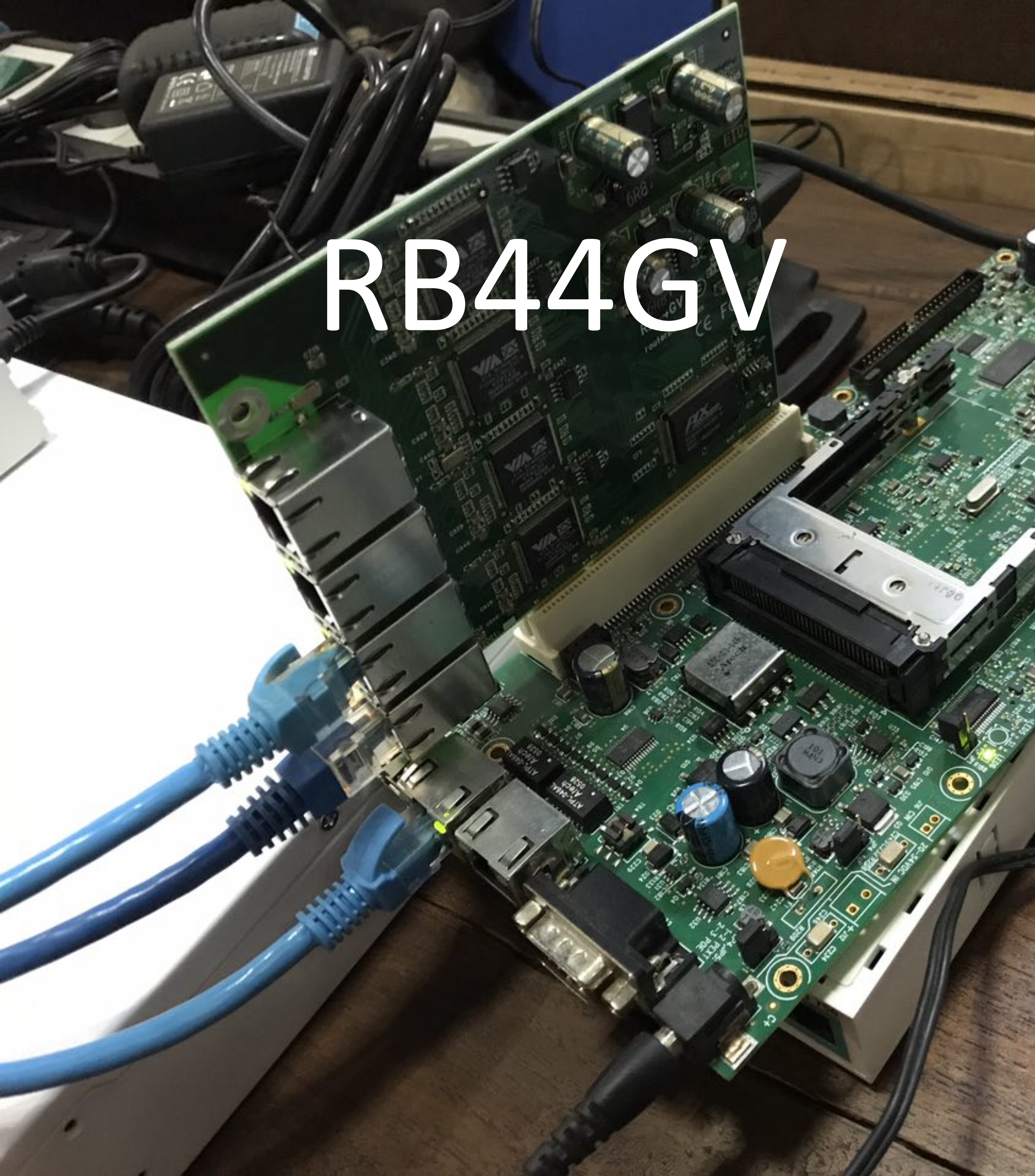
It's only 1% of CPU Load with Simple Queue

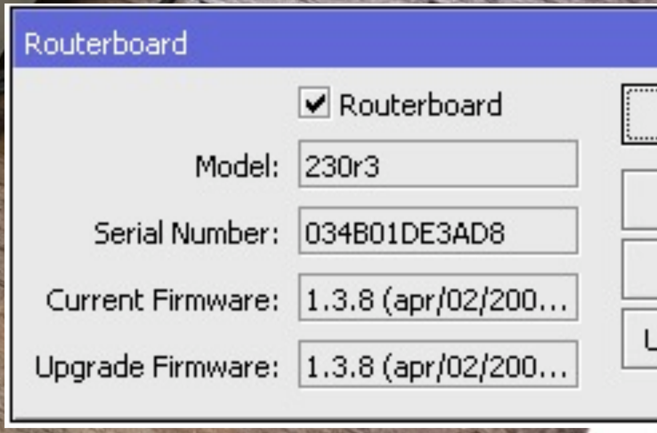
Lets use (not) so fast routerboard!


PH163112
YCL 0246
PH163112
YCL 0246
TR3
SPEAKER
IDE
USB1
RS232

RB230

RB44GV





Routerboard

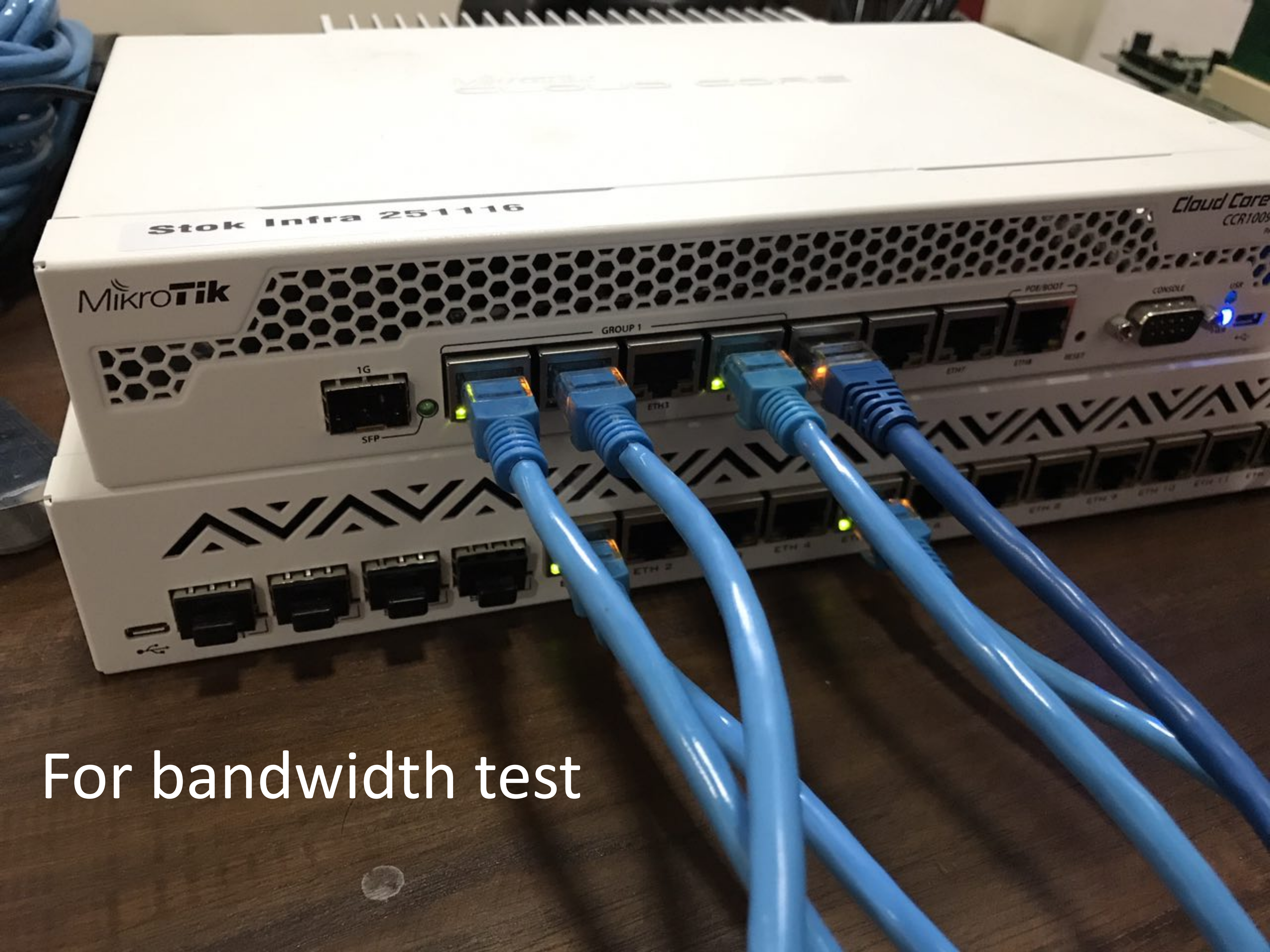
☑ Routerboard

Model: 230r3

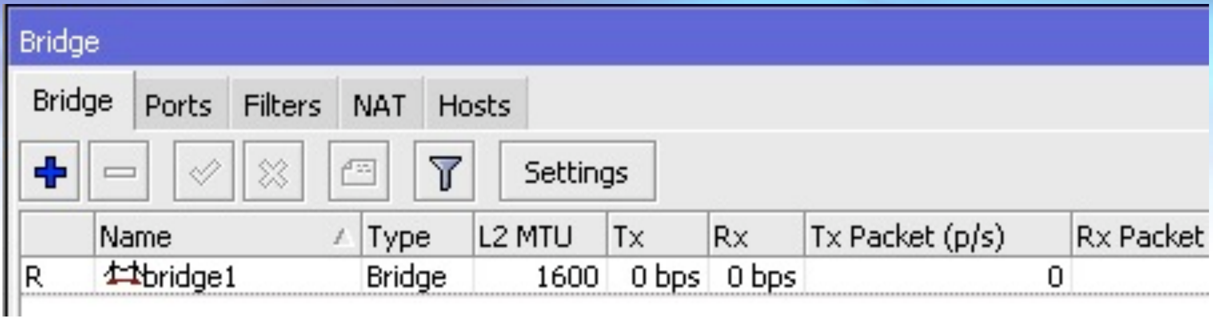
Serial Number: 034B01DE3AD8

Current Firmware: 1.3.8 (apr/02/200...

Upgrade Firmware: 1.3.8 (apr/02/200...



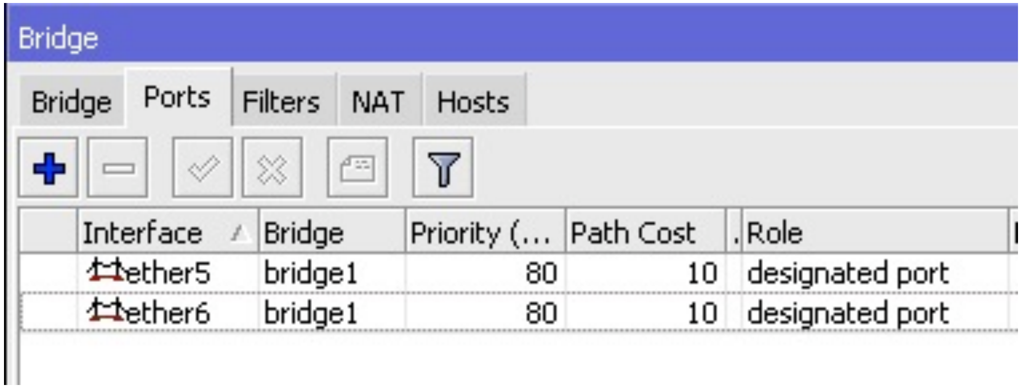
For bandwidth test



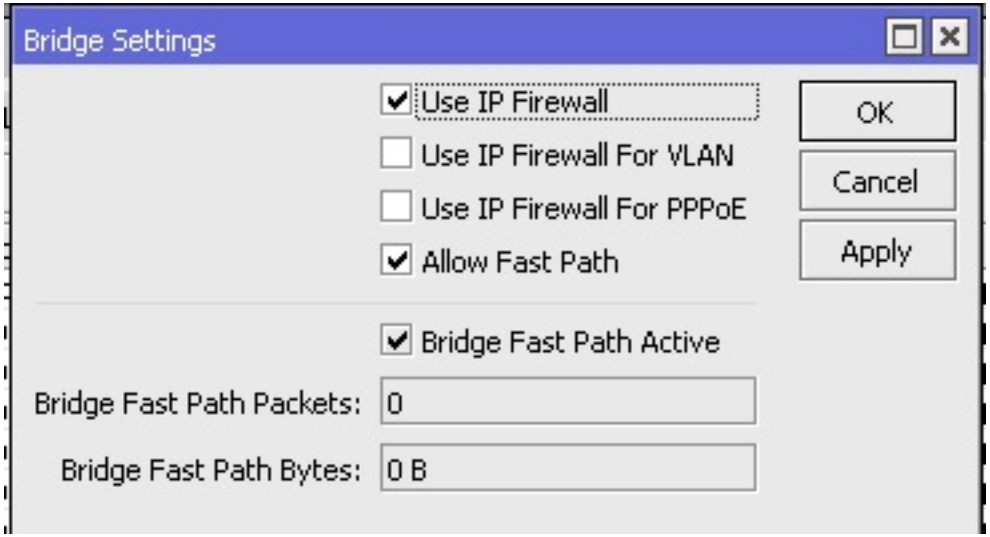
	Name	Type	L2 MTU	Tx	Rx	Tx Packet (p/s)	Rx Packet
R	bridge1	Bridge	1600	0 bps	0 bps	0	

bridge

Bridge ports



	Interface	Bridge	Priority (...	Path Cost	Role
	ether5	bridge1	80	10	designated port
	ether6	bridge1	80	10	designated port



☑ Use IP Firewall

□ Use IP Firewall For VLAN

□ Use IP Firewall For PPPoE

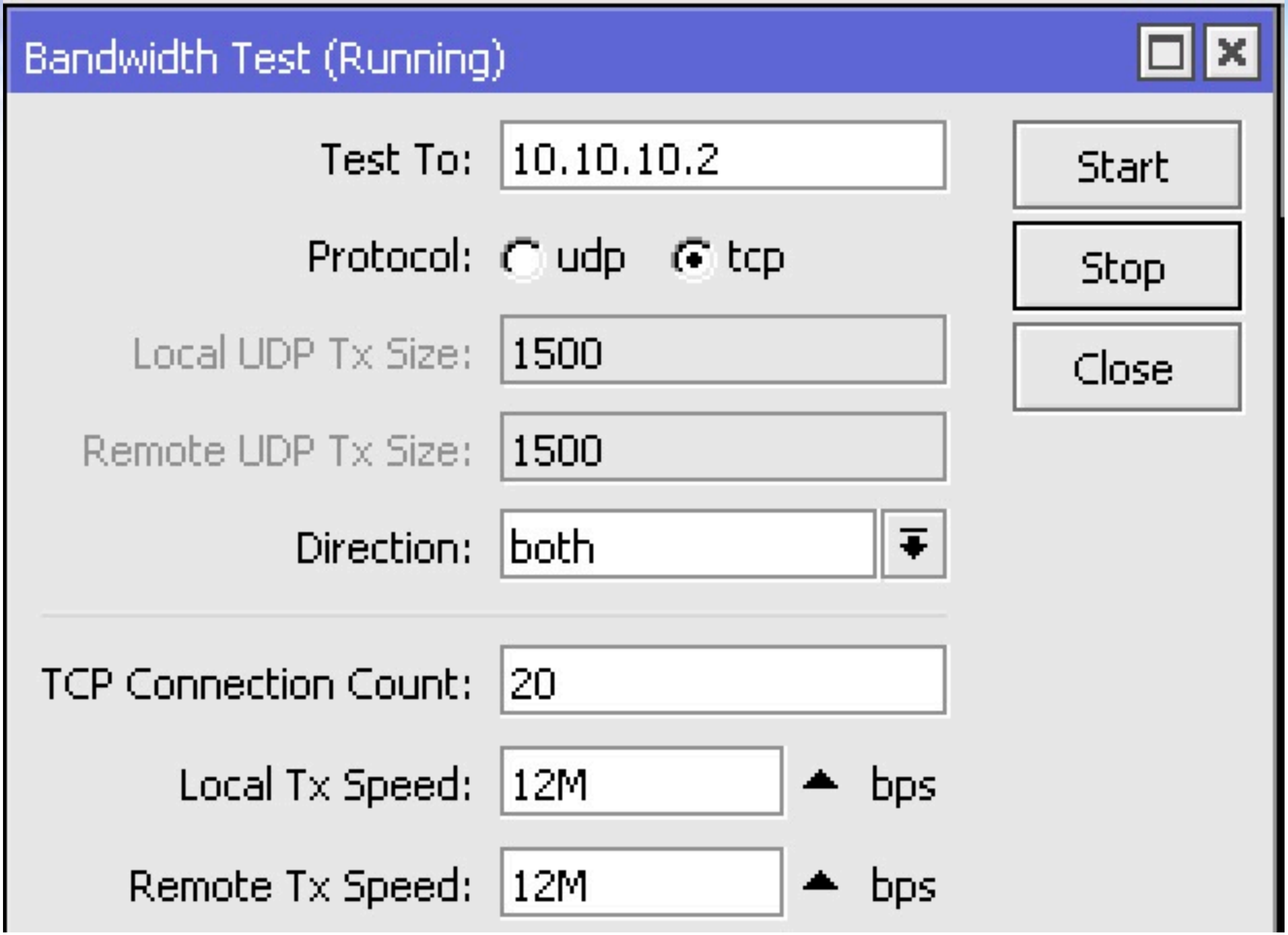
☑ Allow Fast Path

☑ Bridge Fast Path Active

Bridge Fast Path Packets: 0

Bridge Fast Path Bytes: 0 B

Use IP firewall



Bandwidth Test (Running)

Test To: 10.10.10.2

Protocol: ○ udp ● tcp

Local UDP Tx Size: 1500

Remote UDP Tx Size: 1500

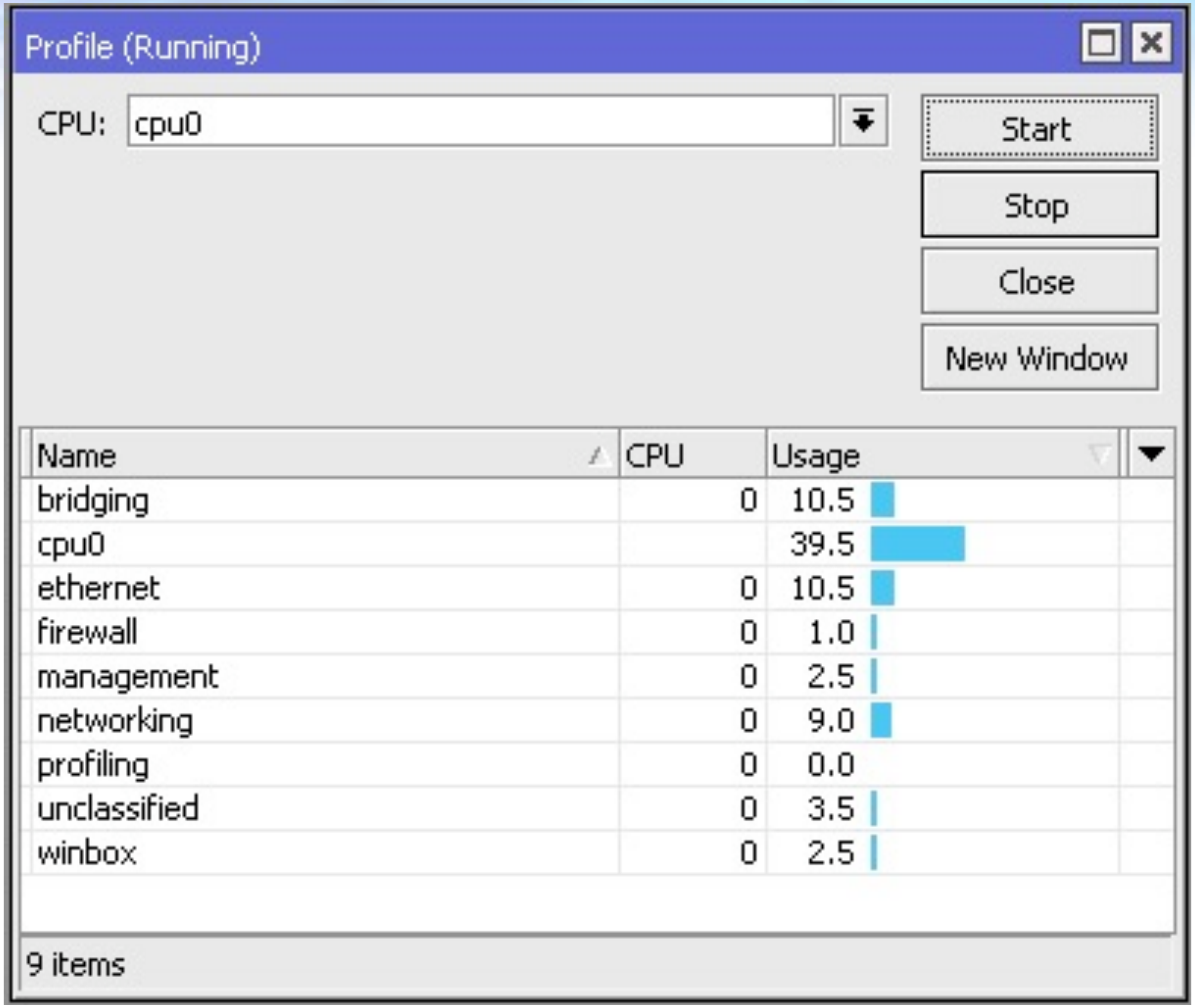
Direction: both

TCP Connection Count: 20

Local Tx Speed: 12M bps

Remote Tx Speed: 12M bps

Bandwidth test on CCR

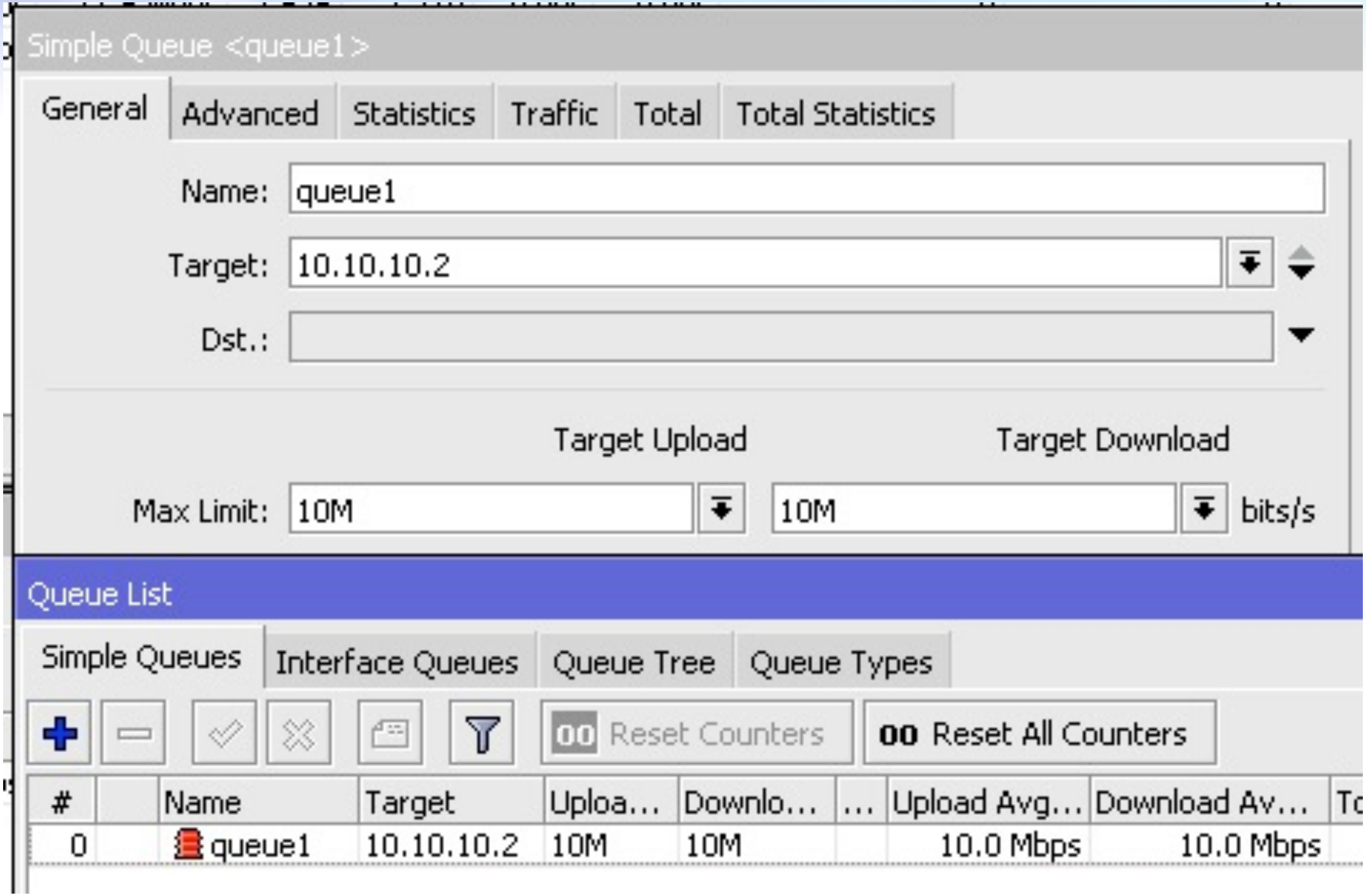


CPU: cpu0

Name	CPU	Usage
bridging	0	10.5
cpu0		39.5
ethernet	0	10.5
firewall	0	1.0
management	0	2.5
networking	0	9.0
profiling	0	0.0
unclassified	0	3.5
winbox	0	2.5

9 items

Without queue



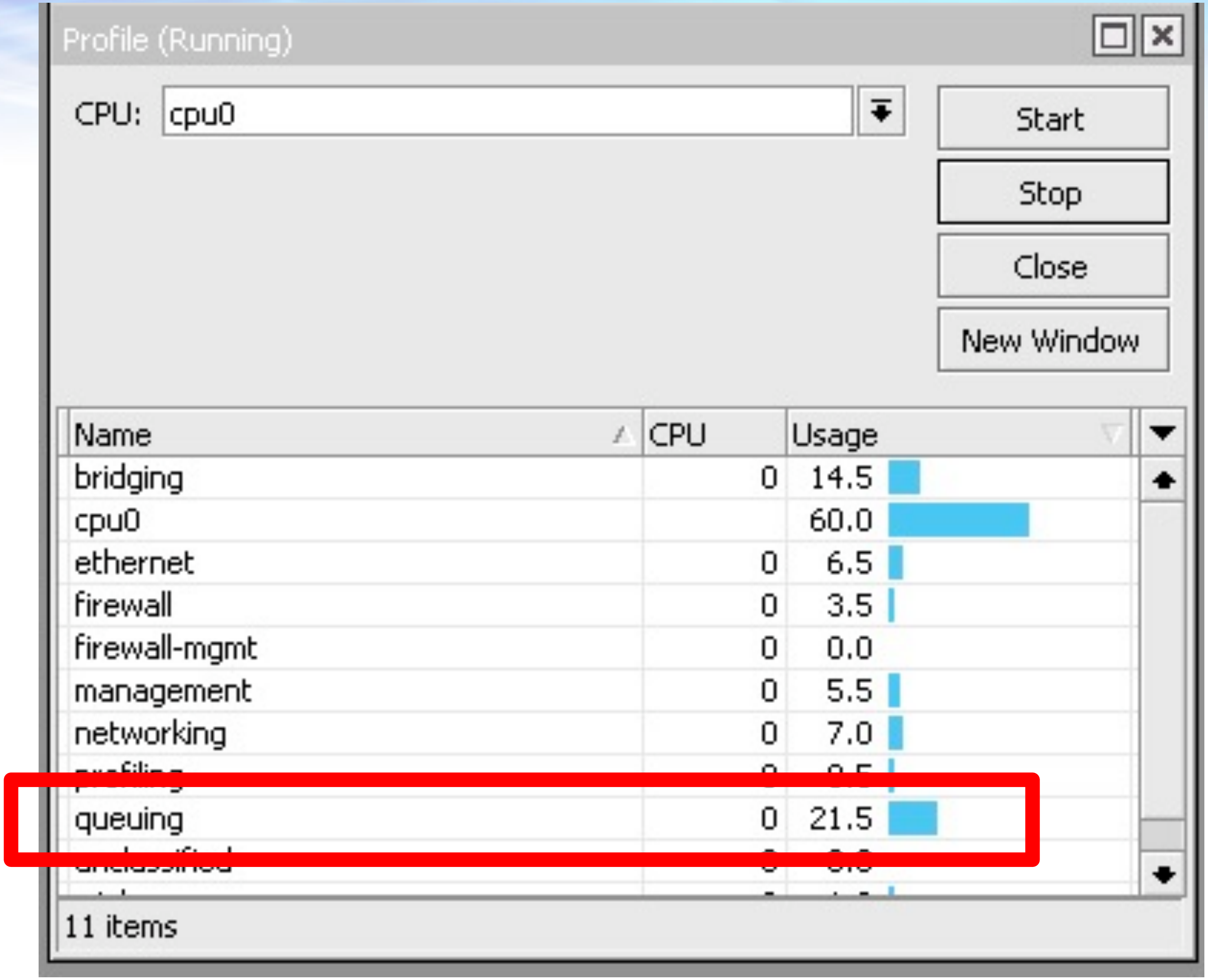
Name: queue1

Target: 10.10.10.2

Dst.:

	Target Upload	Target Download	
Max Limit:	10M	10M	bits/s

#		Name	Target	Uploa…	Downlo…	…	Upload Avg…	Download Av…	To
0		queue1	10.10.10.2	10M	10M		10.0 Mbps	10.0 Mbps	



CPU: cpu0

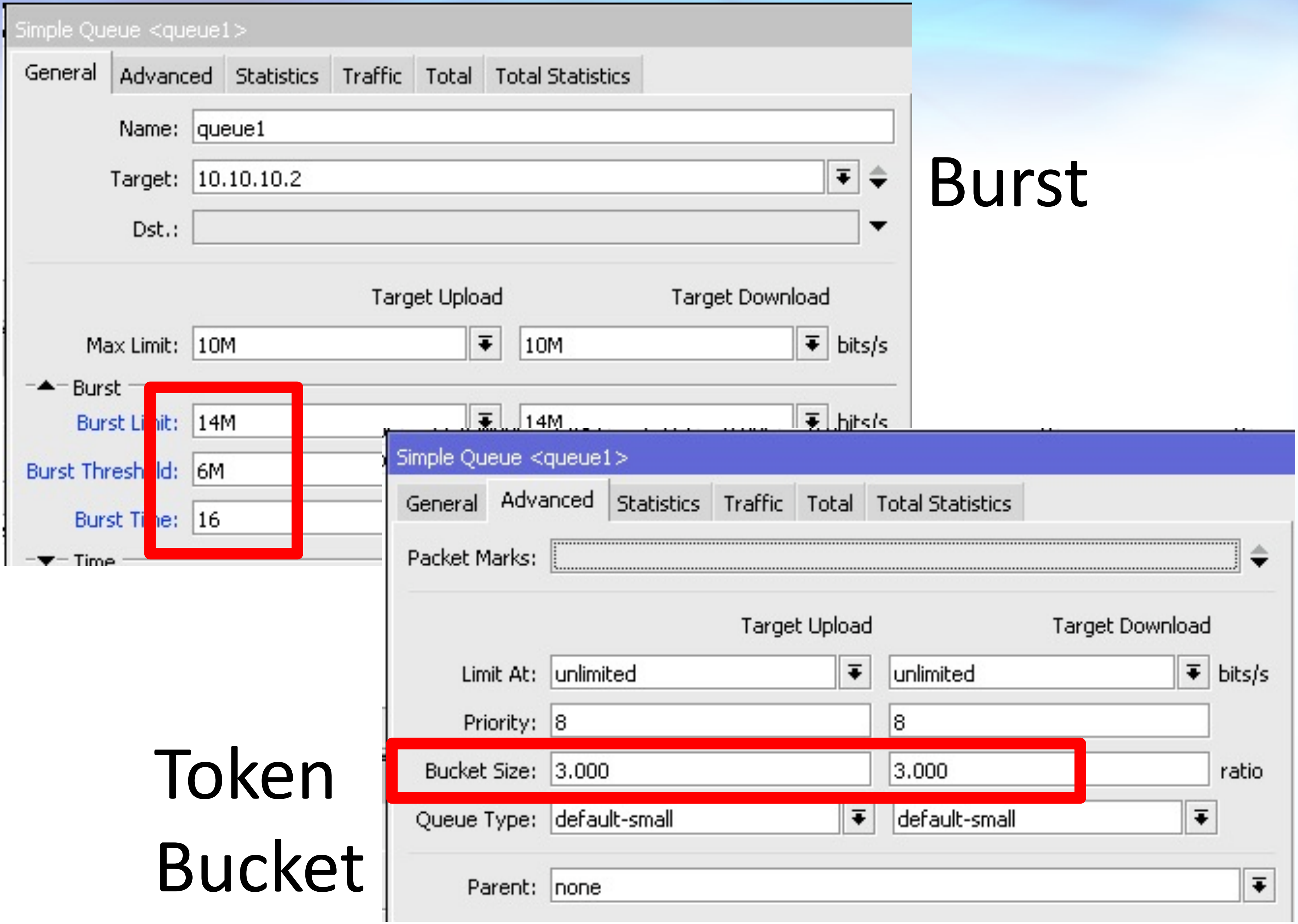
Name	CPU	Usage
bridging	0	14.5
cpu0		60.0
ethernet	0	6.5
firewall	0	3.5
firewall-mgmt	0	0.0
management	0	5.5
networking	0	7.0
profiling	0	0.5
queuing	0	21.5
unclassified	0	0.0

11 items

With queue

Burst

Token Bucket



Simple Queue <queue1>

Name: queue1

Target: 10.10.10.2

Dst.:

	Target Upload	Target Download	
Max Limit:	10M	10M	bits/s

Burst

Burst Limit:	14M	14M	bits/s
Burst Threshold:	6M		
Burst Time:	16		

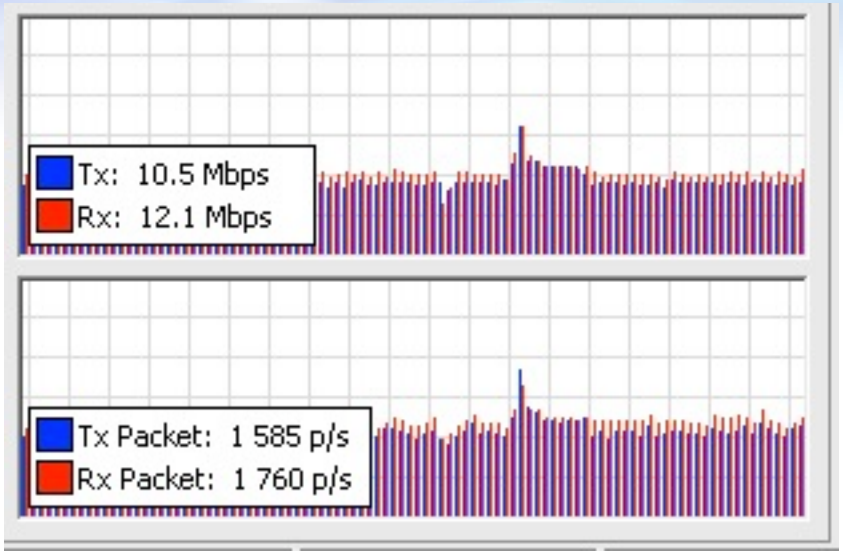
Time

Simple Queue <queue1>

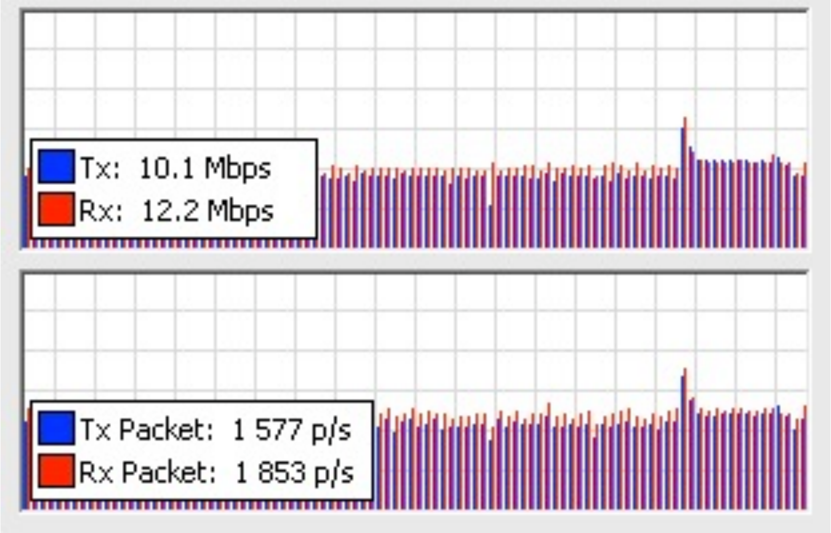
Packet Marks:

	Target Upload	Target Download	
Limit At:	unlimited	unlimited	bits/s
Priority:	8	8	
Bucket Size:	3.000	3.000	ratio
Queue Type:	default-small	default-small	

Parent: none



With burst



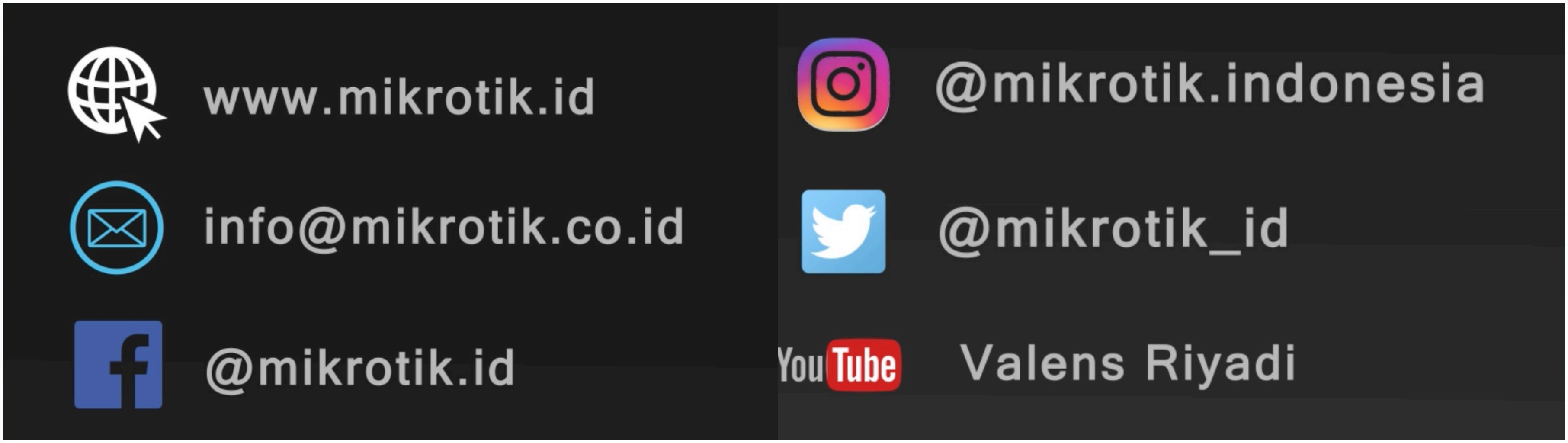
with token bucket

Conclusions

- Token Bucket is alternatives to burst system
- Difference of efficiency between both system is still very small.

Thank you

Comments and suggestions:



www.mikrotik.id

info@mikrotik.co.id

@mikrotik.id

@mikrotik.indonesia

@mikrotik_id

Valens Riyadi