



Mikrotik User Meeting in Indonesia, Yogyakarta, November 29, 2013



UNIVERSITAS ADVENT INDONESIA

Bandwidth Management for University Campus Network

presented by Andrew F. Pakpahan

Self Introduction - *Andrew Pakpahan*

- MTCNA, MTCRE, MTCTCE
- IT Coordinator at Universitas Advent Indonesia (UNAI), Bandung.
- Lecturer at Faculty of Information Technology UNAI.
- You can contact me by email: [andrew\[at\]unai.edu](mailto:andrew[at]unai.edu) or [andrew.pakpahan\[at\]gmail.com](mailto:andrew.pakpahan[at]gmail.com)



UNIVERSITAS ADVENT INDONESIA

<http://www.unai.edu>

Study Case



UNIVERSITAS ADVENT INDONESIA

- Located at Parongpong, Bandung, West Java, Indonesia
- UNAI is a small private boarding University where most of the students stayed in the dormitory and the teachers live in University's provided housing inside the campus.
- The university provide 24 hours internet service for the teachers and students on 25ha campus area.



Our Facts



UNIVERSITAS ADVENT INDONESIA

- We are using various kind of Mikrotik Devices since 2007: RB1000, RB1100 series RB1200, RB2011 series, RB450G, RB750 series, Cloud Core Router
- Currently we have 50mbps dedicated internet (international) connection. We don't have separate connections between IIX (Indonesia's Internet) and International Internet.
- We have around 1800 students, 1200 of them stays in the dormitories.
- We have 80+ Wireless APs, hundreds cable ports, and miles of copper and fibre cables around the campus.
- The system recognised over 2000+ unique MAC addresses had connected to the network on one semester.
- In the peak time we have around 500 concurrent users connected to the network.



Classic Facts

- Bandwidth is limited.
- User's bandwidth demands is always increasing.
- Only 20% of the users takes 80% all the bandwidth.
(20% heavy users)
- Doesn't matter how much bandwidth we throw to the users, they will eat it up!

Bandwidth Management

Bandwidth Management - *What we had tried*

- Linux Based, Squid Pools/HTB scripts queueing.
- Hotspot, Per User Simple Queue.
- Queue Tree + PCQ.
- Prioritising then use PCQ queue (can't be done in Router OS v6.x)

Is it enough?
What's missing?

What's missing?

- With PCQ we can share the bandwidth “**equally**” to all connected users.
- We **can't identify** those heavy users using the existing bandwidth management techniques.

We need to do better bandwidth management!



Mikrotik User Meeting In Indonesia, Yogyakarta November 29, 2013



UNIVERSITAS ADVENT INDONESIA

What we wanted?

- We want to **prioritise users** with less daily usage than those with heavy usage.
- We want users with less internet usage get faster speed than those whose using it a lot.
- User who always want to download something from internet, should be **punished** (we want them to get **slower** speed).

The Strategy

Identify those 'heavy users' and put them in certain group/pool with limited speed.

How we did it?

- Using User based Authentication System (hotspot, PPPoE, etc) with external radius server, custom perl script and Mikrotik's radius attributes.
- Record each client bandwidth usage with the radius server accounting system.
- Create a script running on the server to calculate user's total bandwidth of the day.
 - Then the script will send a radius attributes to Mikrotik to assigns users in different HTB pools according to their usage
 - Run the script every certain time interval (every minutes or every five minutes)
- Create a script that will reset the counters and groups at certain time (we set it at midnight)

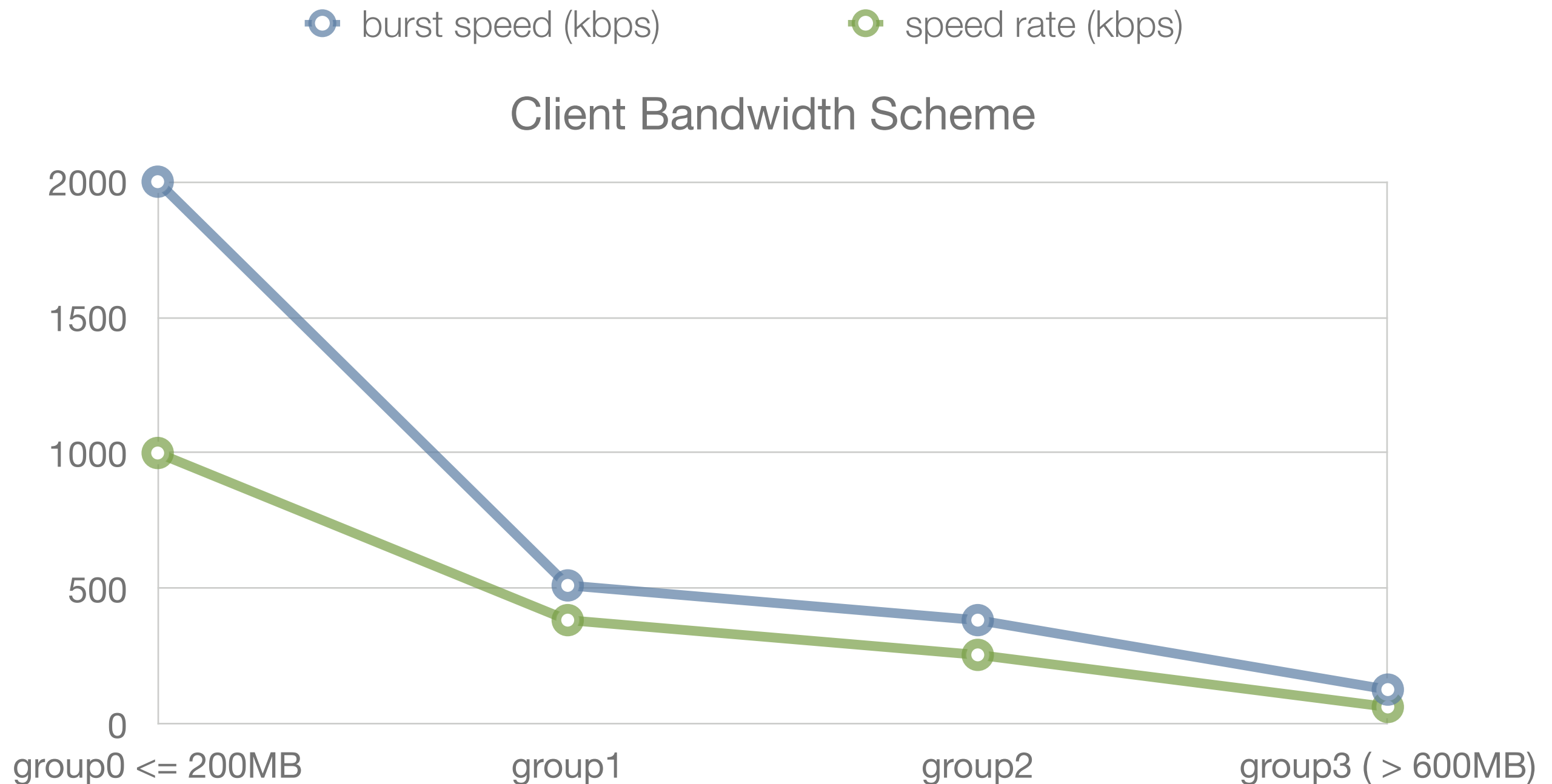
How we did it? - continue

- Identify the users:
 - A: Users that newly connected to the network and use less than X MB of bandwidth.
 - B: Users that already using Y MB of bandwidth.
 - C: Users that already using Z MB of bandwidth.
 - Where ($Z > Y > X$)
- Put the users in certain pools (Using address list features on Mikrotik)
 - Put the A users in group0
 - Put the B users in group1
 - Put the C users in group2
- Put the bandwidth limitations on pools and users.
- The Queue Tree used for bandwidth allocation for each pool.
- The PCQ used for bandwidth allocation for each users on each pool.

Bandwidth Management Design - Per User

Group	BW Usage	Rate	Burst
group0	less than 200MB	1 Mbps	2 Mbps
group1	200MB - 400MB	384 Kbps	512 Kbps
group2	400MB - 600MB	256 Kbps	384 Kbps
group3	more than 600MB	64 Kbps	128 Kbps

Bandwidth Management Design in Graph

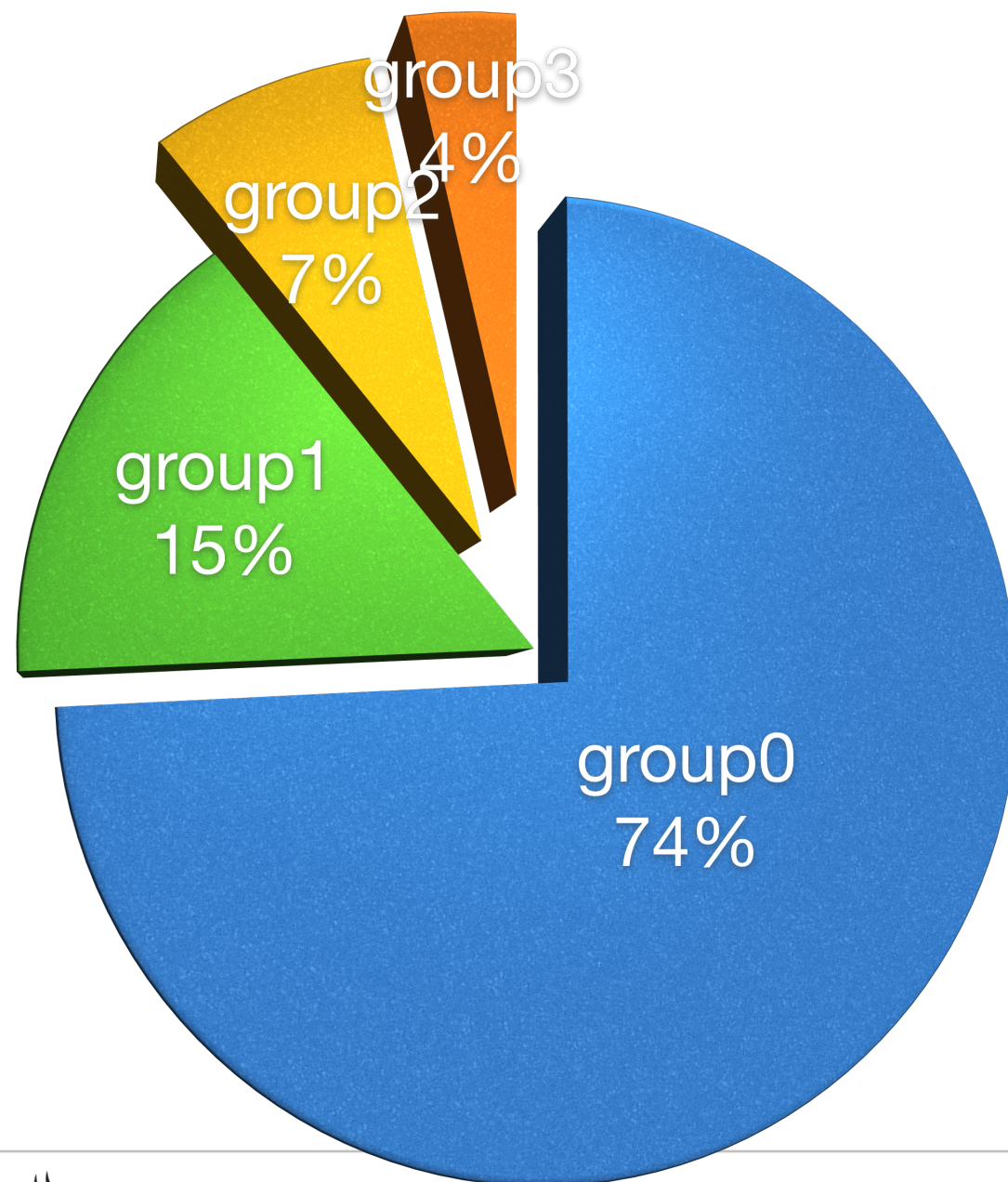


Bandwidth Management Design - Pools

Group	BW Usage	Limit at	Max Limit
group0	less than 200MB	10 Mbps	40 Mbps
group1	200MB - 400MB	4 Mbps	8 Mbps
group2	400MB - 600MB	2 Mbps	4 Mbps
group3	more than 600MB	512 Kbps	1 Mbps

Bandwidth Management Design in Graph

Bandwidth allocation for each group/pool



- Group0 got much large bandwidth than the other groups.
- Group3 got the less bandwidth.

What you need to know before trying this setup

Installing Freeradius + Daloradius for Mikrotik

<http://andrewpakpahan.blogspot.com/2012/08/installing-and-configuring-freeradius.html>



Mikrotik User Meeting In Indonesia, Yogyakarta November 29, 2013



UNIVERSITAS ADVENT INDONESIA

Setting up Hotspot Server & Radius

Hotspot Server Profile <default>

General Login **RADIUS**

☒ Use RADIUS

Default Domain:

Location ID:

Location Name:

MAC Format:

☒ Accounting

Interim Update:

NAS Port Type:

OK Cancel Apply Copy Remove

default

Radius Server <10.1.1.7>

General **Status**

Service

☒ ppp ☒ login

☒ hotspot ☒ wireless

☒ dhcp

Called ID:

Domain:

Address:

Secret:

Authentication Port:

Accounting Port:

Timeout: ms

☐ Accounting Backup

Realm:

Src. Address:

OK Cancel Apply Disable Comment Copy Remove Reset Status

enabled

The bandwidth management script

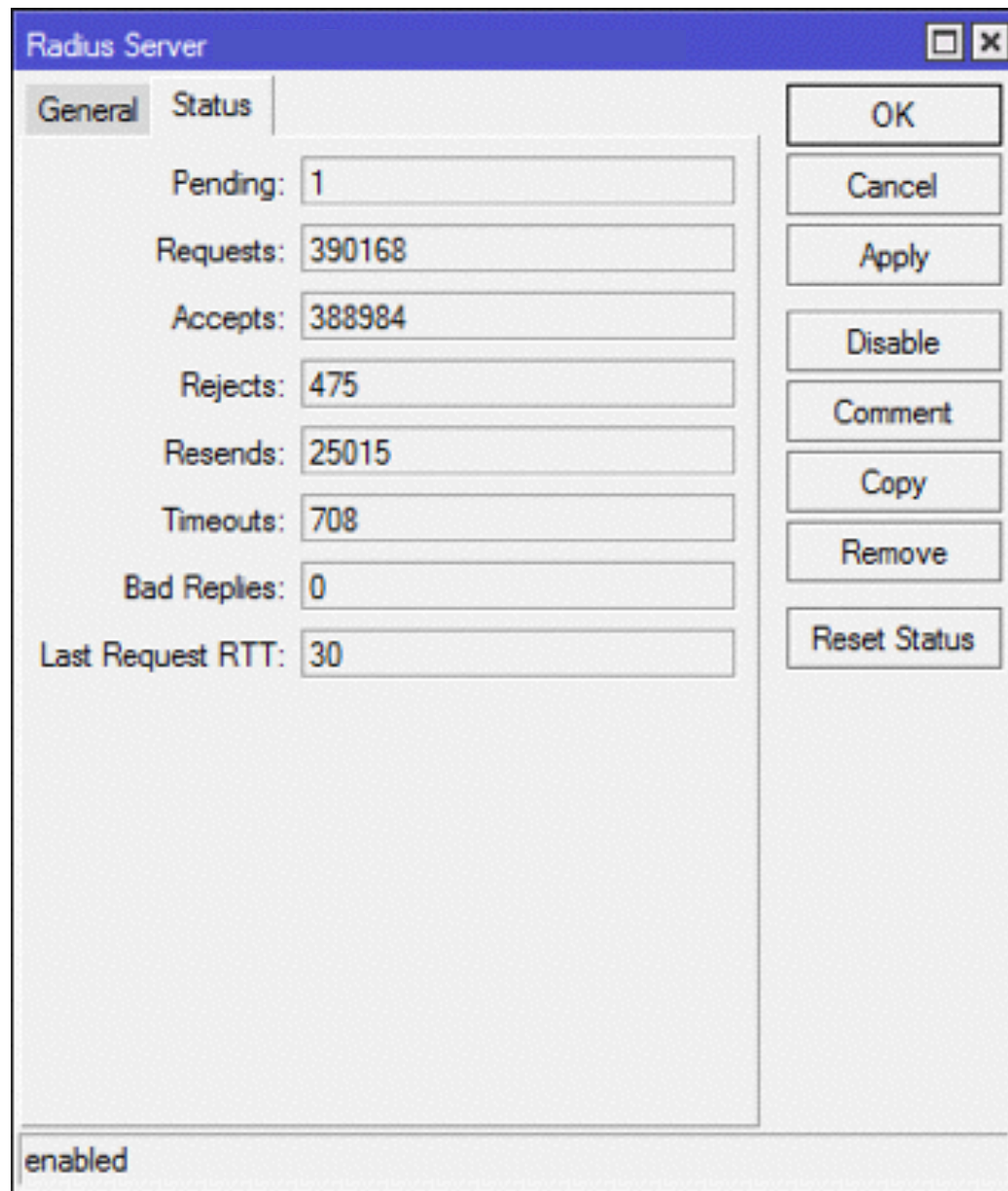
- You can use PHP/Perl or other programming languages.
- In our case we use Perl.
- What the script do?
 - Calculate users usage.
 - Put the users in different group.
 - Send users groups radius updates to Mikrotik.

Radius attribute sends to Hotspot Server

How we send Radius attribute to the router using a perl script

```
system ("echo \"User-Name=[username],Framed-IP-  
Address=[routeripaddress],Acct-Session-  
Id=[acctsessionid],NAS-  
Identifier=[nas_identifier],Mikrotik-  
Group=[group]\" | radclient -x [router_ip]:3799  
coa [password]");
```

Radius Status on Hotspot Server

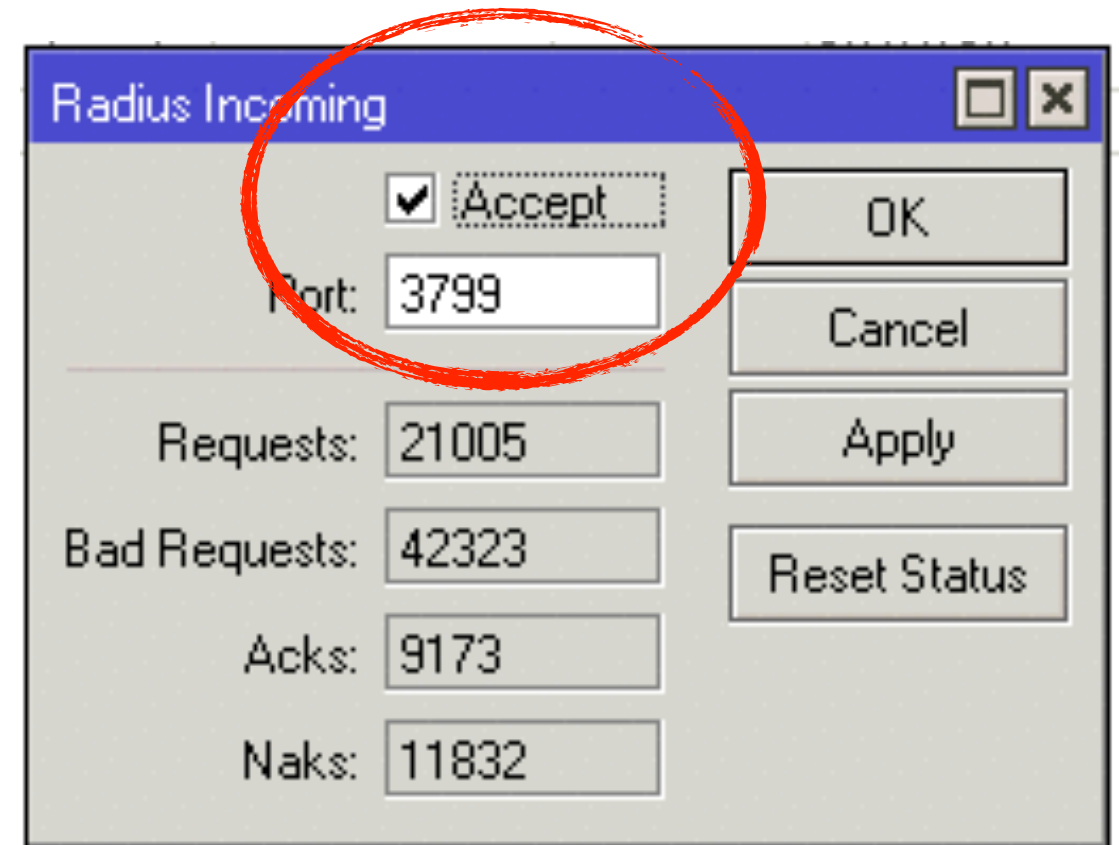


The 'Radius Server' window shows the status of the Radius server. It has two tabs: 'General' and 'Status'. The 'Status' tab is selected, displaying various statistics and control buttons.

Field	Value
Pending:	1
Requests:	390168
Accepts:	388984
Rejects:	475
Resends:	25015
Timeouts:	708
Bad Replies:	0
Last Request RTT:	30

Buttons on the right: OK, Cancel, Apply, Disable, Comment, Copy, Remove, Reset Status.

Bottom status: enabled



The 'Radius Incoming' window shows the configuration for incoming Radius requests. A red circle highlights the 'Accept' checkbox, which is checked, and the 'Port' field, which is set to 3799.

Field	Value
Port:	3799
Requests:	21005
Bad Requests:	42323
Acks:	9173
Naks:	11832

Buttons on the right: OK, Cancel, Apply, Reset Status.

Don't forget to accept incoming radius

Hotspot User Profile Setting

















The image displays three overlapping windows for configuring Hotspot User Profiles. The windows are titled 'Hotspot User Profile <group0>', 'Hotspot User Profile <group1>', and 'Hotspot User Profile <group2>'. Each window has tabs for 'General', 'Advertise', and 'Scripts'. The 'General' tab is active in all three. The settings for each group are as follows:

- group0:** Name: group0, Address Pool: none, Session Timeout: 03:00:00, Idle Timeout: 00:05:00, Keepalive Timeout: 00:10:00, Status Autorefresh: 00:01:00, Shared Users: 1, Rate Limit (x/tx):, Add MAC Cookie: ☒, MAC Cookie Timeout: 3d 00:00:00, Address List: group0.
- group1:** Name: group1, Address Pool: none, Session Timeout: 03:00:00, Idle Timeout: 00:05:00, Keepalive Timeout: 00:10:00, Status Autorefresh: 00:01:00, Shared Users: 1, Rate Limit (x/tx):, Add MAC Cookie: ☒, MAC Cookie Timeout: 3d 00:00:00, Address List: group1.
- group2:** Name: group2, Address Pool: none, Session Timeout: 03:00:00, Idle Timeout: 00:05:00, Keepalive Timeout: 00:10:00, Status Autorefresh: 00:01:00, Shared Users: 1, Rate Limit (x/tx):, Add MAC Cookie: ☒, MAC Cookie Timeout: 3d 00:00:00, Address List: group2.

A detailed inset window for 'group0' shows the following settings:

- Shared Users: 1
- Rate Limit (x/tx):
- Add MAC Cookie: ☒
- MAC Cookie Timeout: 3d 00:00:00
- Address List: group0

Firewall Mangle

::: Group0										
0		mar...	forward					ether3		121.7 GiB 110 368 ...
1		mar...	forward						group0_conn_download	121.7 GiB 110 555 ...
2		mar...	forward					ether2		10.2 GiB 80 949 918
3		mar...	forward						group0_conn_upload	10.2 GiB 80 279 868
::: Group1										
4		mar...	forward					ether3		17.6 GiB 16 587 920
5		mar...	forward						group1_conn_download	17.6 GiB 16 649 087
6		mar...	forward					ether2		1686.9 MiB 12 769 935
7		mar...	forward						group1_conn_upload	1680.4 MiB 12 647 185
::: Group2										
8		mar...	forward					ether3		10.6 GiB 10 074 953
9		mar...	forward						group2_conn_download	10.6 GiB 10 106 359
10		mar...	forward					ether2		1198.8 MiB 7 904 763
11		mar...	forward						group2_conn_upload	1194.1 MiB 7 823 563
::: Group3										
12		mar...	forward					ether3		922.1 MiB 1 057 682
13		mar...	forward						group3_conn_download	922.8 MiB 1 063 966
14		mar...	forward					ether2		131.5 MiB 912 533
15		mar...	forward						group3_conn_upload	128.6 MiB 873 236

We need to do mangle for all our defined groups

Firewall Mangle script

```
/ip firewall mangle
add action=mark-connection chain=forward comment=Group0 dst-address-list=\
group0 in-interface=ether3 new-connection-mark=group0_conn_download \
src-address-list=!localnet
add action=mark-packet chain=forward connection-mark=group0_conn_download \
new-packet-mark=group0_packet_download passthrough=no
add action=mark-connection chain=forward dst-address-list=!localnet \
in-interface=ether2 new-connection-mark=group0_conn_upload \
src-address-list=group0
add action=mark-packet chain=forward connection-mark=group0_conn_upload \
new-packet-mark=group0_packet_upload passthrough=no
add action=mark-connection chain=forward comment=Group1 dst-address-list=\
group1 in-interface=ether3 new-connection-mark=group1_conn_download \
src-address-list=!localnet
```

Firewall Address List

Firewall		
Filter Rules NAT Mangle Service Ports Connections Address Lists Layer7 Protocols		
+ - ✓ ✗ [icon] [icon]		
Name	Address	
D group0	10.4.8.4	
D group2	10.4.8.15	
D group0	10.4.8.16	
D group2	10.4.8.19	
D group0	10.4.8.21	
D group0	10.4.8.23	
D group0	10.4.8.24	
D group1	10.4.8.29	
D group0	10.4.8.33	
D group3	10.4.8.34	
D group0	10.4.8.35	
D group2	10.4.8.37	
D group0	10.4.8.39	
D group0	10.4.8.40	
D group2	10.4.8.45	
D group0	10.4.8.48	
D group0	10.4.8.61	
D group0	10.4.8.62	
D group0	10.4.8.72	
D group0	10.4.8.73	
D group0	10.4.8.76	
D group1	10.4.8.86	
D group0	10.4.8.87	
D group3	10.4.8.88	
D group2	10.4.8.91	
D group1	10.4.8.95	
D group3	10.4.8.97	
D group1	10.4.8.100	
D group3	10.4.8.101	
D group1	10.4.8.105	
D group0	10.4.8.113	
D group0	10.4.8.115	
D group0	10.4.8.116	
D group2	10.4.8.124	
D group2	10.4.8.125	



D	group1	10.4.8.29
D	group0	10.4.8.33
D	group3	10.4.8.34
D	group0	10.4.8.35
D	group2	10.4.8.37
D	group0	10.4.8.39
D	group0	10.4.8.40
D	group2	10.4.8.45
D	group0	10.4.8.48
D	group0	10.4.8.61
D	group0	10.4.8.62
D	group0	10.4.8.72
D	group0	10.4.8.73
D	group0	10.4.8.76
D	group1	10.4.8.86

Users divided to different groups

PCQ Queue Setting

Queue Type <PCQ_down_group0>

Type Name:

Kind:

Rate:

Limit:

Total Limit:

Burst Rate:

Burst Threshold:

Burst Time:

Classifier

☐ Src. Address ☒ Dst. Address

☐ Src. Port ☐ Dst. Port

OK Cancel Apply Copy Remove

Queue Type <PCQ_down_group1>

Type Name:

Kind:

Rate:

Limit:

Total Limit:

Burst Rate:

Burst Threshold:

Burst Time:

Classifier

☐ Src. Address ☒ Dst. Address

☐ Src. Port ☐ Dst. Port

OK Cancel Apply Copy Remove

We must create different PCQ Queue type for each group

The Queue Tree

Total_download	ether2			40M	34.4 Mbps	0 B	152.8 GiB	138 26...	0	
group0_download	Total_downlo...	group0...	10M	40M	22.9 Mbps	346.6 KiB	123.3 GiB	110 48...	35 374	153
group1_download	Total_downlo...	group1...	4M	8M	7.1 Mbps	463.1 KiB	17.9 GiB	16 625 ...	14 556	55
group2_download	Total_downlo...	group2...	2M	4M	3.3 Mbps	670.1 KiB	10.8 GiB	10 101 ...	0	27
group3_download	Total_downlo...	group3...	512k	1M	574.1 kbps	565.0 KiB	942.0 MiB	1 063 5...	0	14
Total_upload	ether3			15M	44.1 kbps	0 B	14.4 GiB	101 13...	0	
group0_upload	Total_upload	group0...			27.5 kbps	0 B	11.2 GiB	79 869 ...	11 829	78
group1_upload	Total_upload	group1...			8.9 kbps	0 B	1844.3 MiB	12 597 ...	1 189	33
group2_upload	Total_upload	group2...			6.2 kbps	0 B	1295.9 MiB	7 802 7...	921	12
group3_upload	Total_upload	group3...			1384 bps	0 B	140.0 MiB	869 757	25	5

Total_download	
group0_download	153
group1_download	55
group2_download	27
group3_download	14
Total_upload	
group0_upload	78
group1_upload	33
group2_upload	12
group3_upload	5

Users are divided to different pool according to their group

Results

- We can **identify users** based on their bandwidth usage.
- Newly connected user will get 'standard' speed and placed in the first group0, after passing a certain limit (usage), the users will be placed in group1 and afterward.
- We can put **users in different group** according to their **internet usage**.
- Those 'heavy users' we targeted will get **slower speed** when they're using the internet connection.

Thank You *MikroTik*

Thank you all for listening