Advanced MikroTik Traffic Control

Prague, June, 2011

Schedule

- 09:00 – 10:30 Morning Session I
  - 10:30 – 11:00 Morning Break
- 11:00 – 12:30 Morning Session II
  - 12:30 – 13:30 Lunch Break
- 13:30 – 15:00 Afternoon Session I
  - 15:00 – 15:30 Afternoon Break
- 15:30 – 17:00 (18.00) Afternoon Session II

Instructors

- Jaromír Čihák, Sys-DataCom
  - Working as Support and Training Engineer at Mikrotikls SIA (MikroTik), SysDataCom CEO.
  - Specialization: Firewall, QoS, Basic, VPN, Large networks, Wireless.
- Jiří Anděl, i4wifi a.s.
  - Working as Support RouterOS Mikrotik.

Housekeeping

- Course materials
- Routers, cables
- Break times and lunch
- Restrooms and smoking area locations

VRRP na dual Wan s aktivnou iface

Course Objective

- Provide thorough knowledge and hands-on training for MikroTik RouterOS basic and advances traffic control capabilities, basic firewalling for networks and multiple networks
- Upon completion of the course you will be able to plan, implement, adjust and debug MikroTik RouterOS network configurations.

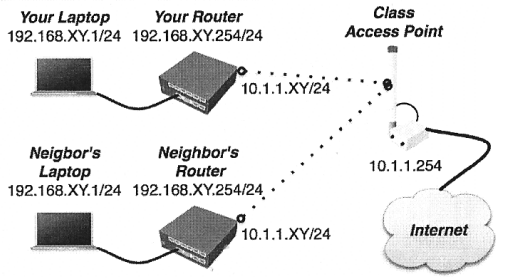
Introduce Yourself

- Please, introduce yourself to the class
  - **Your name**
  - **Your Company**
  - **Your previous knowledge about RouterOS**
  - **Your previous knowledge about networking**
  - **What do you expect from this course?**
- Please, remember your class XY number.
  (X is number of the row, Y is your seat number in the row)
- My number is:_____

Class Setup Lab

- Create an **192.168.XY.0/24** Ethernet network between the laptop (.1) and the router (.254)
- Connect routers to the AP SSID "**AP_RB433**"
- Assign IP address **10.1.1.XY/24** to the wlan1
- Main **GW** and **DNS** address is **10.1.1.254**
- Gain access to the internet from your laptops via local router
- Create new user for your router and change "**admin**" access rights to "**read**"

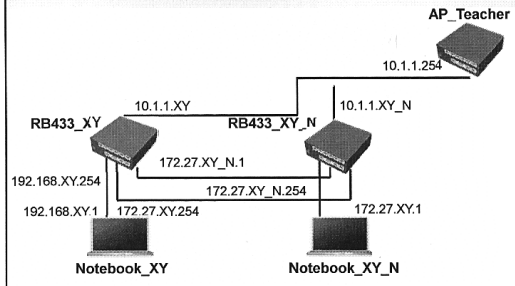
Class Setup



Class setup Lab (cont.)

- Set system identity of the board and wireless radio name to "XY_<your_name>". Example: ***"00_Andel"***
- **Upgrade** your router to the latest Mikrotik RouterOS version **4.x**
- Upgrade your **Winbox** loader version
- Set up **NTP** client – use **10.1.1.254** as server

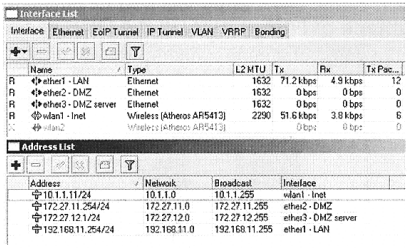
Class setup Lab (cont.)



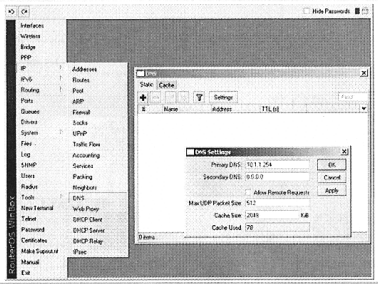
Class setup Lab (cont.)

- Connect **eth2, eth3** LAN cable with your neighbor
- Assign address **172.27.XY.254/24** to eth2
- Assign address 172.27.XY_YOUR_NEIGHBOR.1/24 to eth3
- Create a configuration **backup** and copy it to the laptop (it will be default configuration)

Class setup Lab (cont.)



DNS Client and Cache

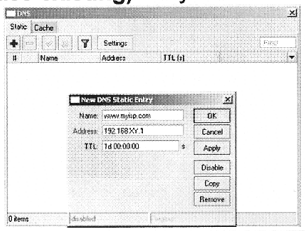


DNS Client and Cache

- DNS client is used only by router in case of web-proxy or hotspot configuration
- Enable "**Allow Remote Requests**" option to transform DNS client into DNS cache
- DNS cache allows to use your router instead of remote DNS server, as all caches - it minimizes resolution time
- DNS cache also can act as DNS server for local area network address resolution

Static DNS Entry

- Each Static DNS entry will add or **override (replace existing)** entry in the DNS cache



DNS Cache Lab

- Configure your router as DNS cache. Use **10.1.1.254** as primary DNS server
- Add static **DNS** entry "www.XY.com" to your router's Local IP address (**10.1.1.XY** – your number)
- Add static DNS entry "www.teacher.com" to **AP_Teacher** router's Public IP address (10.1.1.254)
- Change your laptops DNS server address to your routers address
- Try the configuration and monitor cache list

DHCP

- The Dynamic Host Configuration Protocol is used for dynamic distribution of network setting such as:
  - IP address and netmask
  - Default gateway address
  - DNS and NTP server addresses
  - More than 100 other custom option (supported only by specific DHCP clients)
- DHCP is basically insecure and should only be used in trusted networks

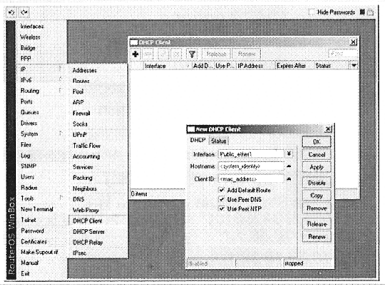
DHCP Comm. scenario

- **DHCP Discovery**
  src-mac=<client>,dst-mac=<broadcast>, protocol=udp, src-ip=0.0.0.0:68, dst-ip=255.255.255.255:67
- **DHCP Offer**
  src-mac=<DHCP-server>, dst-mac=<broadcast>, protocol=udp, src-ip=<DHCP-server>:67, dst-ip=255.255.255.255:67
- **DHCP Request**
  src-mac=<client>, dst-mac=<broadcast>, protocol=udp, src-ip=0.0.0.0:68, dst-ip=255.255.255.255:67
- **DHCP Acknowledgement**
  src-mac=<DHCP-server>, dst-mac=<broadcast>, protocol=udp, src-ip=<DHCP-server>:67, dst-ip=255.255.255.255:67

DHCP Client Identification

- DHCP server are able to track lease association with particular client based on identification
- The identification can be achieved in 2 ways
  - Based on "caller-id" option (*dhcp-client-identifier from RFC2132*)
  - Based on MAC address, if "caller-id" option is not specified
- "hostname" option allow RouterOS clients to send **additional** identification to the server, by default it is "system identity" of the router

DHCP Client



DHCP Server

- There can be only one DHCP server per interface/relay combination on the router
- To create DHCP server you **must** have
  - IP address on desired DHCP server interface
  - Address pool for clients or static leases
  - Information about planned DHCP network
- All 3 options must correspond
- "Lease on Disk" should be used to reduce number of writes to the drive (useful with flash drives)

Mikrotik Traffic Control 21

DHCP Networks

- In DHCP Networks menu you can configure specific DHCP options for particular network.
- Same of the options are integrated into RouterOS, others can be assigned in raw form (specified in RFCs)
- Additional information at:
  http://www.iana.org/assignments/bootp-dhcp-parameters
- DHCP server is able to send out any option
- DHCP client can receive only implemented options

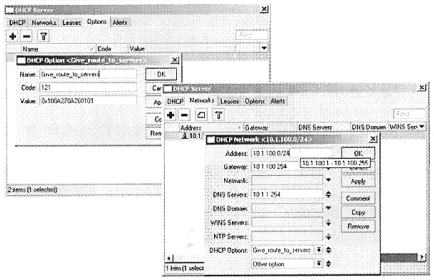
Mikrotik Traffic Control 22

DHCP Options

- Implemented DHCP options
  - Subnet-Mask (option 1) - netmask
  - Router (option 3) - gateway
  - Domain-Server (option 6) - dns-server
  - Domain-Name (option 15) - domain
  - NTP-Servers (option 42) - ntp-server
  - NETBIOS-Name-Server (option 44) - wins-server
- Custom DHCP options (Example:)
  - Classless Static Route (option 121) - *"0x100A270A260101" = "network=10.39.0.0/16 gateway=10.38.1.1"*
  - *- 16 10 39 10 38 1 1*

Mikrotik Traffic Control 23

Custom DHCP Option

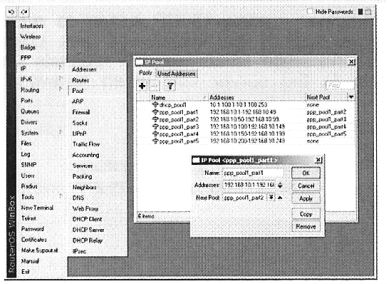


Mikrotik Traffic Control 24

IP Address Pool

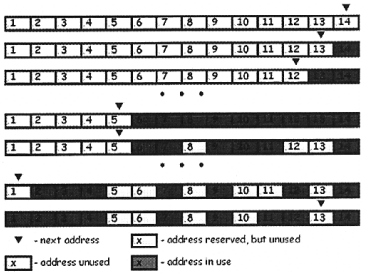
- IP address pools are used to define range of IP addresses for dynamic distribution (DHCP, PPP, Hotspot)
- Address pool must exclude already occupied addresses (such as server or static addresses)
- It is possible to assign more that one range to the pool
- It is possible to chain several pools together by using "Next Pool" option

IP Address Pool



Next pool = až dojdou IP, začne další...

Address Pool in Action



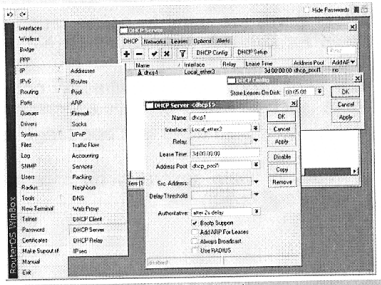
Other DHCP Serv. Settings

- **Src.address** – specifies DHCP servers address if more than one IP on DHCP server's interface
- **Delay Threshold** – prioritize one DHCP server over another (bigger delay less priority)
- **Add ARP For Leases** – allow to add ARP entries for leases if interface ARP=reply-only
- **Always Broadcast** – allow communication with non-standard clients like pseudo-bridges
- **Bootp Support, Use RADIUS** – (obvious)

Authoritative DHCP Server

- Authoritative – allow DHCP server to reply on unknown client's broadcast and ask client to restart the lease (client send broadcasts only if unicast to the server fails when renewing the lease)
- Authoritative allow to:
  - Prevent rouge DHCP server operations
  - Faster network adaptation to DHCP configuration changes

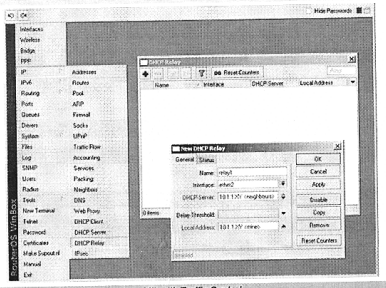
DHCP Server



DHCP Relay

- DHCP Relay is just a proxy that is able to receive a DHCP discovery and request and resend them to the DHCP server
- There can be only one DHCP relay between DHCP server and DHCP client
- DHCP communication with relay does not require IP address on the relay, but relay's "local address" option **must** be the same with server's "relay address" option

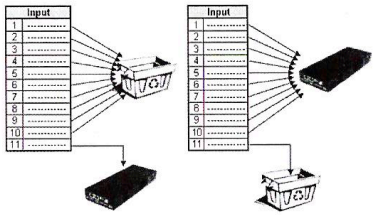
DHCP Relay



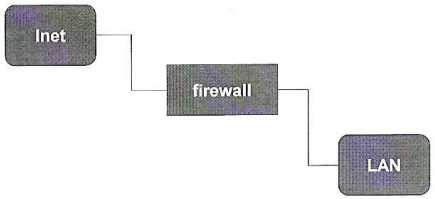
Firewall

Firewall Building Tactics

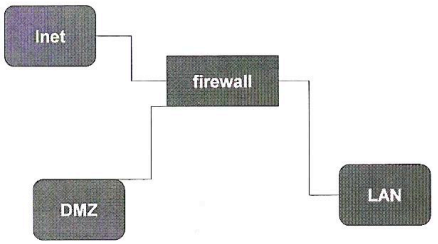
- Drop all unneeded, accept everything else
- Accept only needed, drop everything else



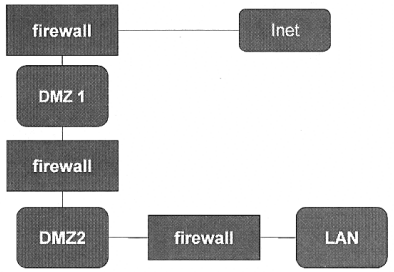
Simple instalation



Firewall Filters Structure



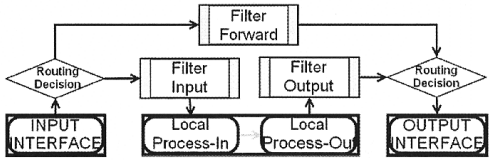
Firewall Filters Structure



Firewall Filters Structure

- Firewall filter rules are organized in chains
- There are default and user-defined chains
- There are three default chains
  - **input** – processes packets sent to the router
  - **output** – processes packets sent by the router
  - **forward** – processes packets sent through the router
- Every user-defined chain should subordinate to at least one of the default chains

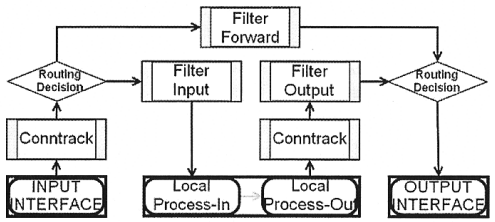
Firewall Filter Str. Diagram



Connection Tracking

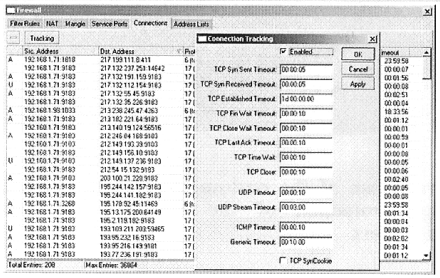
- Connection Tracking (or Conntrack) system is the heart of firewall, it gathers and manages information about **all** active connections.
- By disabling the conntrack system you will lose functionality of the **NAT** and most of the filter and mangle conditions.
- Each conntrack table entry represents bidirectional data exchange
- Conntrack takes a lot of **CPU** resources (disable it, if you don't use firewall)

Connection Placement



Mikrotik Traffic Control 41

Conntrack – Winbox View



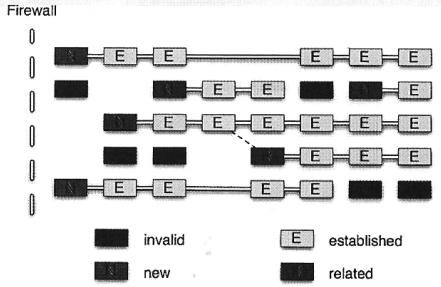
Mikrotik Traffic Control 42

Condition: Conn. State

- Connection state is a status assigned to each packet by conntrack system:
  - **New** – packet is opening a new connection
  - **Established** – packet belongs to already known connection
  - **Invalid** – packet does not belong to any of the known connections
  - **Related** – packet is also opening a new connection, but it is in some kind relation to already known connection
- Connection state ≠ TCP state

Mikrotik Traffic Control 43

Condition: Conn. State



Mikrotik Traffic Control 44

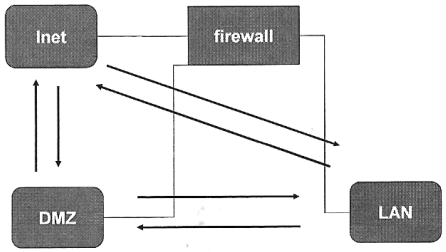
Chain Forward

Protection of the customers from the viruses and protection of the Internet from the customers

Firewall Filter Chains

- The firewall filter facility is a tool for packet filtering
- Firewall filters consist from the sequence of IF-THEN rules
  - 0) IF <condition(s)> THEN <action>
  - 1) IF <condition(s)> THEN <action>
  - 2) IF <condition(s)> THEN <action>
- If a packet doesn't meet all the conditions of the rule, it will be sent on to the next rule.
- If a packet meet all the conditions of the rule, specified action will be performed on it.

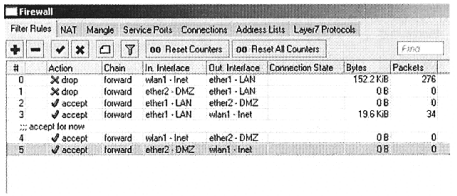
Chain Forward Lab



Chain Forward Lab

- There are followed rules:
  - **Accept** LAN to Internet
  - **Accept** DMZ to Internet
  - **Accept** LAN to DMZ
  - **Drop** Internet to LAN
  - **Drop** DMZ to LAN
  - **Selected** Internet to DMZ

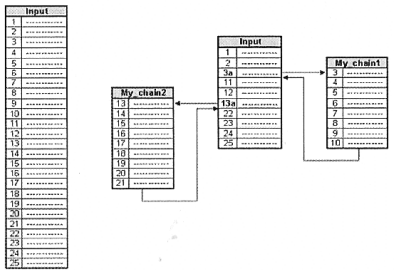
Chain Forward Lab



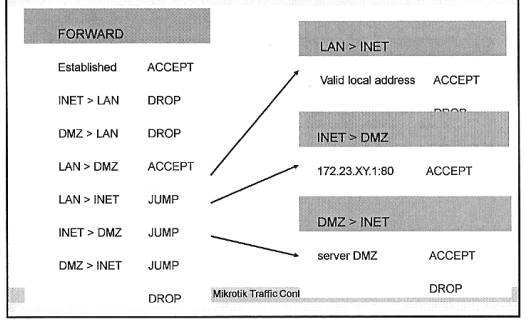
Firewall Filter Chains

- You can reroute traffic to user-defined chains using action jump (and reroute it back to the default chain using action return)
- Users can add any number of chains
- User-defined chains are used to optimize the firewall structure and make it more readable and manageable
- User-defined chains help to improve performance by reducing the average number of processed rules per packet

User-Defined Chains



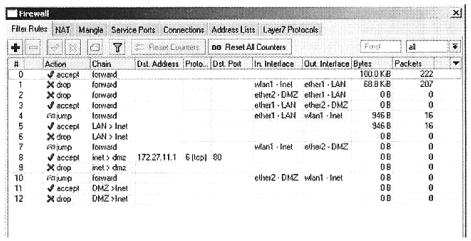
Jump scheme



Chain Forward Lab

- Create 3 jump rules to select:
  - Only valid **customers** can **use internet**
  - Only valid **customers** can **access to DMZ**
  - Only valid **servers** can access **Internet**
  - Only valid **services** can be access **from internet**
    - *172.27.XY.1 – port 80*
- Create accept established packet rule to first position – thing about functionality – check settings with neighbor

Chain Forward Lab

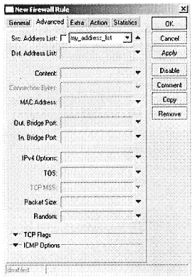


Bogon IPs

- There are ~4,3 billion IPv4 addresses
- There are several IP ranges restricted in public network
- There are several of IP ranges reserved (not used at the moment) for specific purposes
- There are lots of unused IP ranges!!!

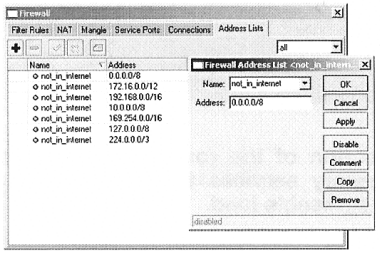
Address List Options

- Instead of creating one filter rule for each IP network address, you can create only one rule for IP address list.
- Use "Src./Dst. Address List" options
- Create an address list in "/ip firewall address-list" menu



Address List Lab

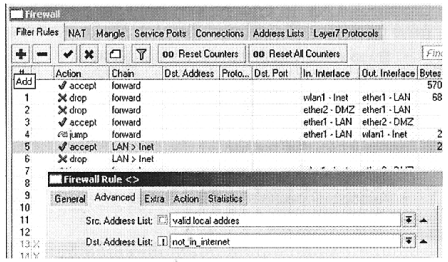
- Make an address list of most common bogon IPs



Adv. Address Filtering Lab

- Allow packets to leave your network only from the valid customers addresses
- Allow packets to leave your network only to the valid Internet addresses

Adv. Address Filtering Lab



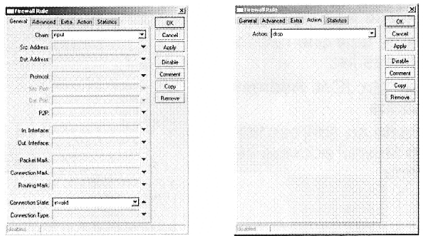
Chain Input

Protection of the router – allowing only necessary services from reliable source with agreeable load.

chráněný RB

Invalid Rule Example

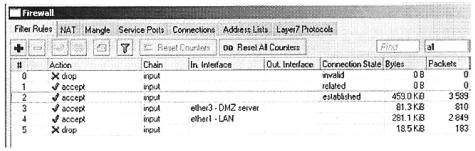
*/ip firewall filter add chain=input connection-state=invalid *
Action=drop comment="Drops invalid packets"



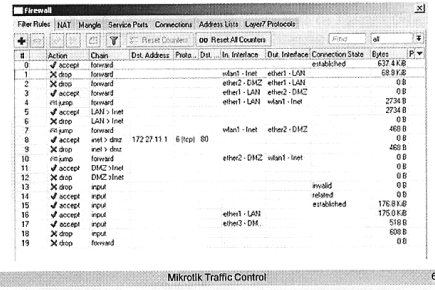
Connection State Lab

- Create 3 rules to ensure that only connectionstate **new** packets will proceed through the input filter
  - **Drop** all connection-state **invalid** packets
  - **Accept** all connection-state **related** packets
  - **Accept** all connection-state **established** packets
- Create 3 rules to ensure that only you will be able to connect to the router
  - **Accept** all packets from your local network
  - **Accept** all packets as DMZ server
  - **Drop** everything else

Connection State Lab



Connection State Lab



RouterOS Services

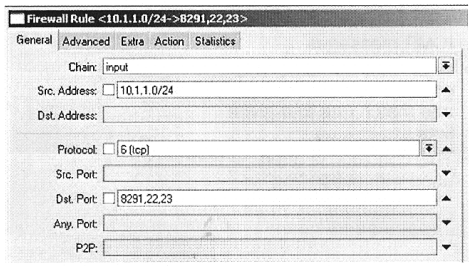
Nr	Port	Protocol	Comments
1	20	tcp	FTP
2	21	tcp	FTP
3	22	tcp	SSH,SFTP
4	23	tcp	Telnet
5	53	tcp	DNS
6	80	tcp	HTTP
7	179	tcp	BGP
8	443	tcp	SHTTP (Hotspot)
9	646	tcp	LDP (MPLS)
10	1080	tcp	SoCKS (Hotspot)
11	1723	tcp	PPTP
12	1968	tcp	MME
13	2000	tcp	Bandwidth server
14	2210	tcp	Dude server
15	2211	tcp	Dude server
16	2828	tcp	uPnP
17	3128	tcp	WEB Proxy
18	8291	tcp	Winbox
19	8728	tcp	API
20	.....	/1	ICMP
21	.....	/2	IGMP (Multicast)
22	.....	/4	IPIP

Nr	Port	Protocol	Comments
23	53	udp	DNS
24	123	udp	NTP
25	161	udp	SNMP
26	500	udp	IPSec
27	520	udp	RIP
28	521	udp	RIP
29	646	udp	LDP (MPLS)
30	1698	udp	RSVP (MPLS)
31	1699	udp	RSVP (MPLS)
32	1701	udp	L2TP
33	1812	udp	User-manager
34	1813	udp	User-manager
35	1900	udp	uPnP
36	1966	udp	MME
37	5678	udp	Neighbour Discovery
38	.....	/46	RSVP (MPLS)
39	.....	/47	PPTP EoIP
40	.....	/50	IPSec
41	.....	/51	IPSec
42	.....	/89	OSPF
43	.....	/103	PIM (Multicast)
44	.....	/112	VRRP

RouterOS Services Lab

- Create rules to **allow** only necessary RouterOS **services** to be accessed from the public network
- Use action "**log**" to determine those services
- Create rule to allow **winbox, ssh** and **telnet** connection from teacher's network (**10.1.1.0/24**)
- **Arrange** rules accordingly
- Write **comment** for each firewall rule

RouterOS Services Lab



Important Issue

- Firewall filters do not filter **MAC** level communications
- You should turn off **MAC-telnet** and **MAC-Winbox** features at least on the public interface
- You should disable network discovery feature and router would not reveal itself anymore
- ("**/ip neighbor discovery**" menu)

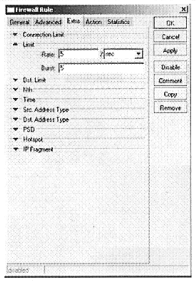
Tools → mac server

Network Intrusion Types

- Network intrusion is a serious security risk that could result in not only the temporal denial, but also in total refusal of network service
- We can point out 4 major network intrusion types:
  - Ping flood
  - Port scan
  - DoS attack
  - DDoS attack

Ping Flood

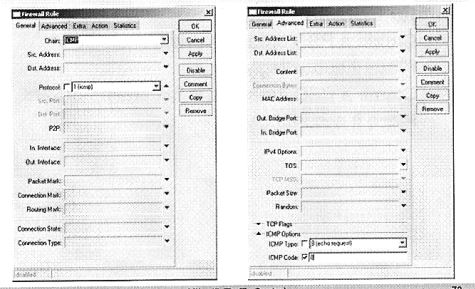
- Ping flood usually consist from volumes of random ICMP messages
- With "limit" condition it is possible to bound the rule match rate to a given limit
- This condition is often used with action "log"



ICMP Message Types

- Typical IP router uses only five types of ICMP messages (type:code)
  - For PING - messages 0:0 and 8:0
  - For TRACEROUTE – messages 11:0 and 3:3
  - For Path MTU discovery – message 3:4
- Other types of ICMP messages should be blocked

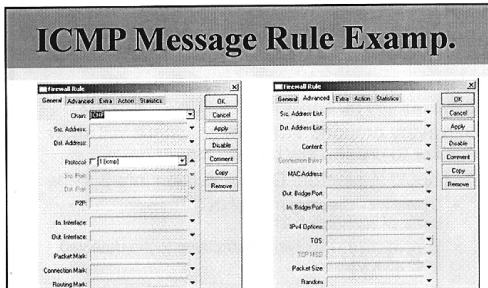
ICMP Message Rule Examp.



ICMP Flood Lab

- Make the new chain – ICMP
  - **Accept 5** necessary ICMP messages
  - **Set** match rate to 5 pps with 5 packet burst possibility
  - **Drop** all other ICMP packets
- Move all ICMP packets to ICMP chain
  - Create an action "**jump**" rule in the chain Input
  - Place it accordingly
  - Create an action "**jump**" rule in the chain Forward
  - Place it accordingly

ICMP Message Rule Examp.

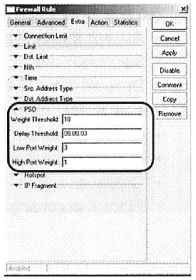


ICMP Flood Lab

- Make the new chain – ICMP
  - **Accept 5** necessary ICMP messages
  - **Set** match rate to 5 pps with 5 packet burst possibility
  - **Drop** all other ICMP packets
- Move all ICMP packets to ICMP chain
  - Create an action "**jump**" rule in the chain Input
  - Place it accordingly
  - Create an action "**jump**" rule in the chain Forward
  - Place it accordingly

Port Scan

- Port Scan is sequential TCP (UPD) port probing
- PSD (Port scan detection) is possible only for TCP protocol
- Low ports
  - From 0 to 1023
- High ports
  - From 1024 to 65535



PSD Lab

- Create PSD protection
  - Create a PSD drop rule in the chain Input
  - Place it accordingly
  - Create a PSD drop rule in the chain Forward
  - Place it accordingly

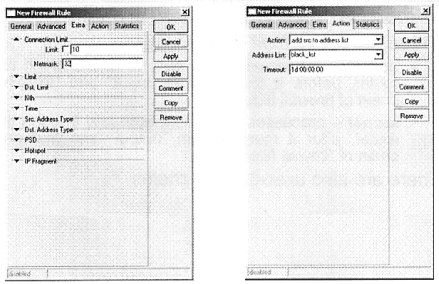
DoS Attacks

- Main target for DoS attacks is consumption of resources, such as CPU time or bandwidth, so the standard services will get Denial of Service (DoS)
- Usually router is flooded with TCP/SYN (connection request) packets. Causing the server to respond with a TCP/SYN-ACK packet, and waiting for a TCP/ACK packet.
- Mostly DoS attackers are virus infected customers

DoS Attack Protection

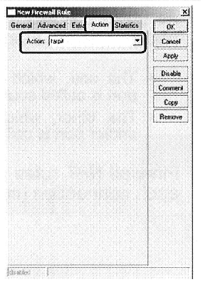
- All IP's with more than 10 connections to the router should be considered as DoS attackers
- With every dropped TCP connection we will allow attacker to create new connection
- We should implement DoS protection into 2 steps:
  - Detection - Creating a list of DoS attackers on the basis of connection-limit
  - Suppression – applying restrictions to the detected DoS attackers

DoS Attack Detection



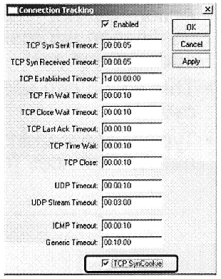
DoS Attack Suppression

- To stop the attacker from creating new connections, we will use action "tarpit"
- We must place this rule before the detection rule or else address-list entry will rewrite all the time



DDoS attacks

- A Distributed Denial of Service attack is very similar to DoS attack only it occurs from **multiple** compromised systems
- Only thing that could help is "TCPSyn Cookie" option in conntrack system



Network Address Translation (NAT)

Destination NAT, Source NAT, NAT traversal

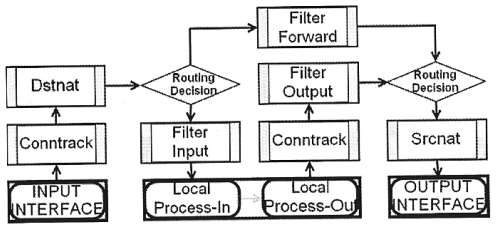
NAT Types

- As there are two IP addresses and ports in an IP packet header, there are two types of NAT
  - The one, which rewrites source IP address and/or port is called source NAT (src-nat)
  - The other, which rewrites destination IP address and/or port is called destination NAT (dst-nat)
- Firewall NAT rules process only the first packet of each connection (connection state "new" packets)

Firewall NAT Structure

- Firewall NAT rules are organized in chains
- There are two default chains
  - **dstnat** – processes traffic sent to and through the router, before it divides in to "input" and "forward" chain of firewall filter.
  - **srcnat** – processes traffic sent from and through the router, after it merges from "output" and "forward" chain of firewall filter.
- There are also user-defined chains

IP Firewall Diagram

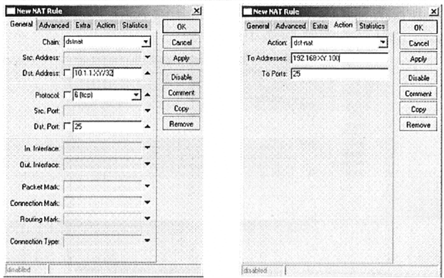


Dst-nat

- Action "dst-nat" changes packet's destination address and port to specified address and port
- This action can take place only in chain dstnat
- Typical application: ensure access to local network services from public network

Masquerade ports

Dst-nat Rule Example



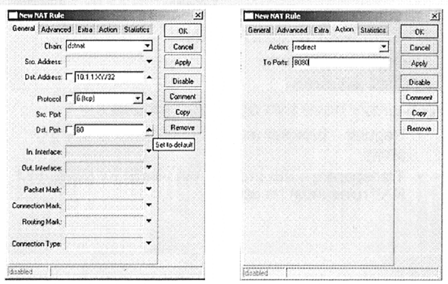
Mikrotik Traffic Control 89

Redirect

- Action "redirect" changes packet's destination address to router's address and specified port
- This action can take place only in chain dstnat
- Typical application: transparent proxying of
- network services (DNS,HTTP)

Mikrotik Traffic Control 90

Redirect Rule Example



Mikrotik Traffic Control 91

Redirect Lab

- Capture all TCP and UDP port 53 packets originated from your private network 192.168.XY.0/24 and redirect them to the router itself.
- Set your laptops DNS server to the random IP address
- Clear your router's and your browser's DNS
- cache
- Try browsing the Internet
- Take a look at DNS cache of the router

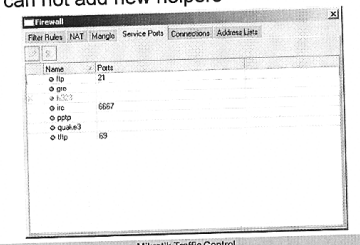
Mikrotik Traffic Control 92

Source NAT Drawbacks

- Hosts behind a NAT-enabled router do not have true end-to-end connectivity:
  - connection initiation from outside is not possible
  - some TCP services will work in "passive" mode
  - src-nat behind several IP addresses is unpredictable
  - same protocols will require so-called NAT helpers to to work correctly (NAT traversal)

NAT Helpers

- You can specify ports for existing NAT helpers, but you can not add new helpers



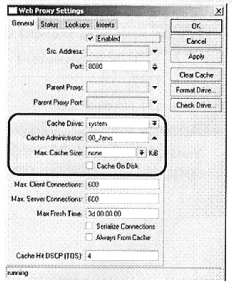
Web Proxy

Web-Proxy

- Web-proxy have 3 mayor features
  - HTTP and FTP traffic caching
  - DNS name filtering
  - DNS redirection
- Web-proxy have two operation modes
  - Regular – browser must be configured to use this proxy
  - Transparent – this proxy is not visible for customers NAT rules must be applied

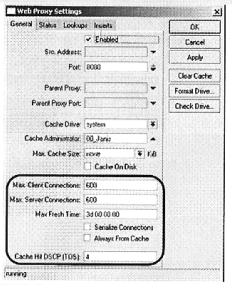
Web-Proxy Caching

- No caching
  - Max-cache-size = none
- Cache to RAM
  - Max-cache-size ≠ none
  - Cache-on-disk = no
- Cache to HDD
  - Max-cache-size ≠ none
  - Cache-on-disk = yes
- Cache drive



Web-Proxy Options

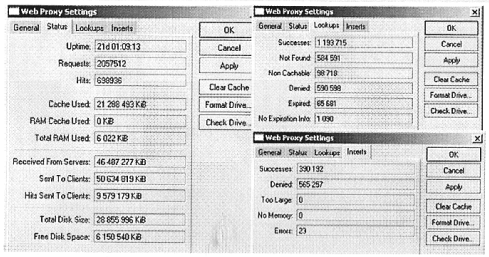
- **Maximal-client-connections** - number of connections accepted from clients
- **Maximal-server-connections** - number of connections made by server



Web-Proxy Options

- **Serialize-connections** – use only one connection for proxy and server communication (if server supports persistent HTTP connection)
- **Always-from-cache** - ignore client refresh requests if the cache content is considered fresh
  - **Max-fresh-time** - specifies how long objects without an explicit expiry time will be considered fresh
- **Cache-hit-DSCP** – specify DSCP value for all packets generated from the web-proxy cache

Web-Proxy Statistics

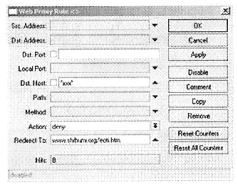


Proxy Rule Lists

- Web-proxy supports 3 sets of rules for HTTP request filtering
  - **Access List –** dictates policy whether to allow specific HTTP request or not
  - **Direct Access List** – list works only if parent-proxy is specified – dictates policy whether to bypass parent proxy for specific HTTP request or not.
  - **Cache List** – dictates policy whether to allow specific HTTP request be cached or not

Proxy Rules

- It is possible to intercept HTTP request based on:
  - TCP/IP information
  - URL
  - HTTP method
- Access list also allow you to redirect denied request to specific page



URL Filtering

http://www.mikrotik.com/docs/ros/2.9/graphics:packet_flow31.jpg

Destination host Destination path

- Special characters
  - "*" - any number of any characters
  - "?" - any character
    - www.mi?roti?.com
    - www.mikrotik*
    - * mikrotik*

Web-Proxy Lab

- Teacher will have proxy, that redirects all requests to separate web-page on 10.1.1.254
- Enable transparent web-proxy on your router with caching to the memory
- Create rules in access list to check its functionality
- Create rules in direct access list to check its functionality
- Create rules in Cache list to check its functionality

Firewall Mangle

IP packet marking and IP header fields adjustment

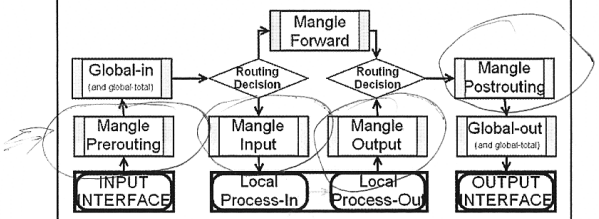
What is Mangle?

- The mangle facility allows to mark IP packets with special marks.
- These marks are used by other router facilities like routing and bandwidth management to identify the packets.
- Additionally, the mangle facility is used to modify some fields in the IP header, like TOS (DSCP) and TTL fields.

Mangle Structure

- Mangle rules are organized in chains
- There are five built-in chains:
  - **Prerouting**- making a mark before Global-In queue
  - **Postrouting** - making a mark before Global-Out queue
  - **Input** - making a mark before Input filter
  - **Output** - making a mark before Output filter
  - **Forward** - making a mark before Forward filter
- New user-defined chains can be added, as necessary

Mangle - Queue Diag. (S)



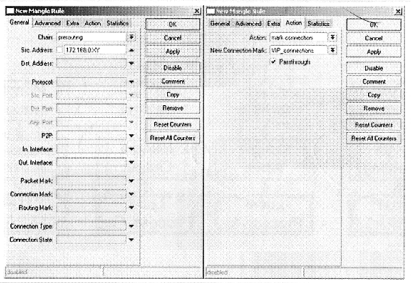
Mangle actions

- There are 7 more actions in the mangle:
  - **mark-connection** – mark connection (**only first packet**)
  - **mark-packet** – mark a flow (all packets)
  - **mark-routing** - mark packets for policy routing
  - **change MSS** - change maximum segment size of the packet
  - **change TOS** - change type of service
  - **change TTL** - change time to live
  - strip IPv4 **options**

Marking Connections

- Use mark connection to identify one or group of connections with the specific connection mark
- Connection marks are stored in the connection tracking table
- There can be only one connection mark for one connection.
- Connection tracking helps to associate each packet to a specific connection (connection mark)

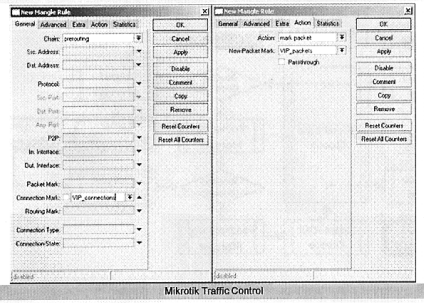
Mark Connection Rule



Marking Packets

- Packets can be marked
  - Indirectly. Using the **connection tracking** facility, based on previously created **connection marks** (faster)
  - Directly. Without the **connection tracking** – no connection marks necessary, router will compare **each** packet to a given conditions (this process imitates some of the connection tracking features)

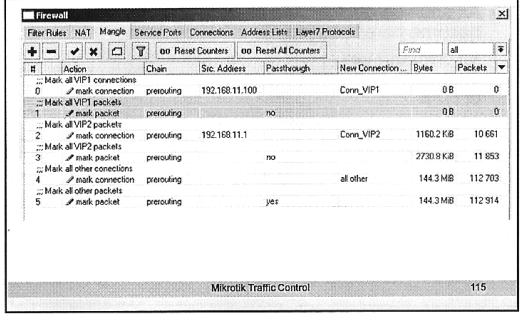
Mark Packet Rule



Mangle Packet Mark Lab

- Mark all connections from 192.168.XY.100 address (imaginary VIP 1)
- Mark all packets from VIP 1 connections
- Mark all connections from 192.168.XY.200 address (imaginary VIP 2)
- Mark all packets from VIP 2 connections
- Mark all other connections
- Mark packets from all other connections

Mangle View



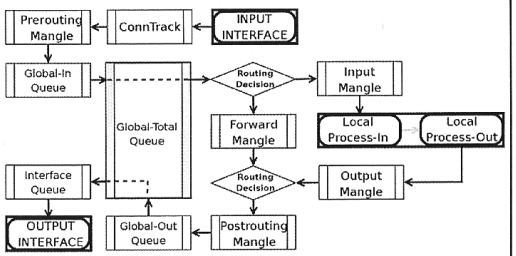
HTB

Hierarchical Token Bucket

HTB

- All Quality of Service implementation in RouterOS is based on Hierarchical Token Bucket
- HTB allows to create hierarchical queue structure and determine relations between parent and child queues and relation between child queues
- RouterOS support 3 virtual HTBs (global-in, global-total, global-out) and one more just before every interface

Mangle and HTBs



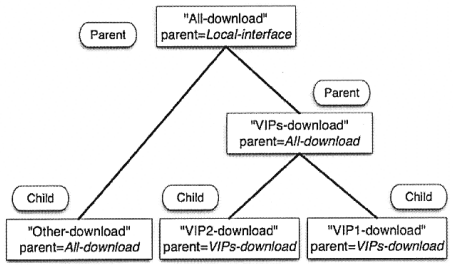
HTB (cont.)

- When packet travels **through** the router, it passes all 4 HTB trees
- When packet travels **to** the router, it passes only global-in and global-total HTB.
- When packet travels **from** the router, it passes global-out, global-total and interface HTB.

HTB Features - Structure

- As soon as queue have at least one child it become parent queue
- All child queues (don't matter how many levels of parents they have) are on the same bottom level of HTB
- Child queues make actual traffic consumption, parent queues are responsible only for traffic distribution
- Child queues are not able to get more traffic than parent has

HTB Features - Structure



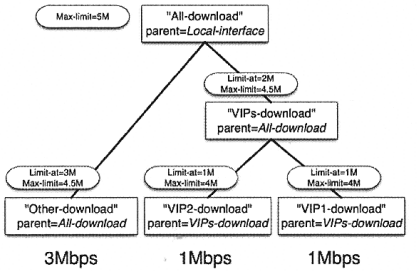
HTB Feat. – Dual Limitation

- HTB has two rate limits:
  - **CIR (Committed Information Rate)** – in worst case scenario flow will get its **limit-at** no matter what (assuming we can actually send so much data)
  - **MIR (Maximal Information Rate)** – in best case scenario a flow can get up to **max-limit** if there is spare bandwidth
- At first HTB will try to satisfy every child queue's CIR **(limit-at)** – only then it will try to reach MIR **(max-limit)**

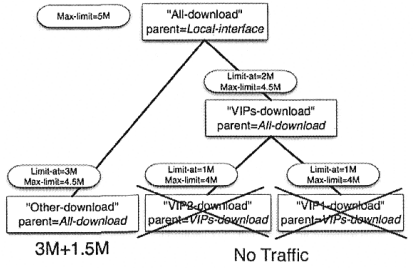
Dual Limitation

- Maximal rate of the parent must be equal or bigger than sum of committed rates of the children
  - MIR (parent) ≥ CIR(child1) +...+ CIR(childN)
- Maximal rate of any child must be less or equal to maximal rate of the parent
  - MIR (parent) ≥ MIR(child1)
  - MIR (parent) ≥ MIR(child2)
  - MIR (parent) ≥ MIR(childN)

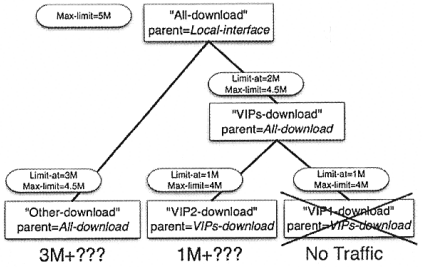
HTB Distribution (limit-at)



HTB Distribution (max-limit)



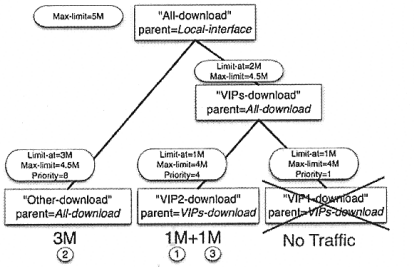
HTB Distribution



HTB Features - Priority

- Work only for child queues to arrange them
- 8 is the lowest priority, 1 is the highest
- Queue with higher priority will reach its CIR before the queue with lower priority
- Queue with higher priority will reach its MIR before the queue with lower priority
- Actual traffic prioritization will work only if limitat and max-limit is specified (not 0)

HTB Distribution (priority)

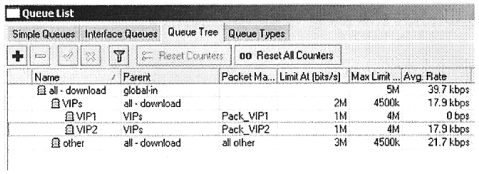


Queue Tree

Advanced queue structures

Queue Tree

- Queue tree is direct implementation of HTB
- Each queue in queue tree can be assigned only in one HTB
- Each child queue must have packet mark assigned to it



Queue Tree and Simple

- Tree queue can be placed in 4 different places:
  - Global-in ("direct" part of simple queues are placed here automatically)
  - Global-out ("reverse" part of simple queues are placed here automatically)
  - Global-total ("total" part simple queues are placed here automatically)
  - Interface queue
- If placed in same place Simple queue will take traffic before Queue Tree

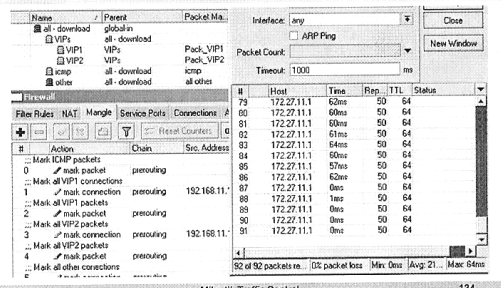
HTB Lab

- Create Queue tree from the example
- Extend mangle and queue tree configuration to prioritize ICMP and HTTP traffic over all other traffic **only** for regular clients
  - Replace regular client packet mark with 3 traffic type specific marks
  - Create 3 child queues for regular client queue in queue tree
  - Assign packet marks to queues

HTB Lab (cont.)

- Consume all the available traffic using bandwidth-test (through the router) and check the ping response times
- Set highest priority to ICMP
- Check the ping response times

HTB Lab (cont.)



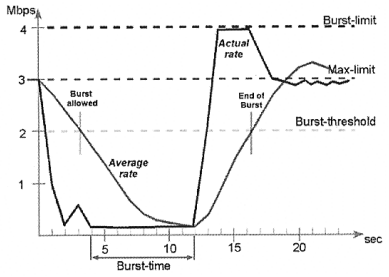
QoS Feature "Burst"

- Burst is one of the best ways to increase HTTP performance
- Bursts are used to allow higher data rates for a short period of time
- If an *average data rate* is less than **burst-threshold**, burst could be used(*actual data rate* can reach **burst-limit**)
- *Average data rate* is calculated from the last **burst-time** seconds

Burst - Average Data Rate

- Average data rate is calculated as follows:
  - **burst-time** is being divided into 16 periods
  - router calculates the **average data rate** of each class over these small periods
- Note, that the **actual burst period** is not equal to the burst-time. It can be several times shorter than the burst-time depending on the max-limit, burst-limit, burst-threshold, and actual data rate history (see the graph example on the next slide)

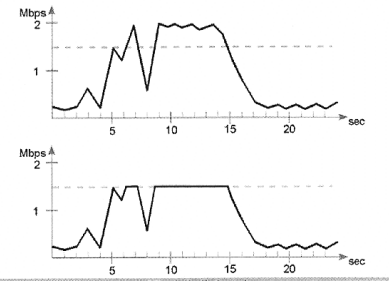
Limitation with Burst



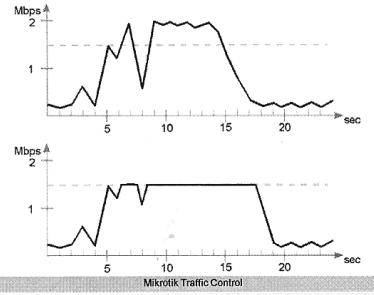
Queue Types

- RouterOS have 4 queue types:
  - **FIFO** – First In First Out (for Bytes or for Packets)
  - **RED** – Random Early Detect (or Drop)
  - **SFQ** – Stochastic Fairness Queuing
  - **PCQ** – Per Connection Queuing (MikroTik Proprietary)
- Each queue type have 2 aspects:
  - Aspect of the **Scheduler**
  - Aspect of the **Shaper**

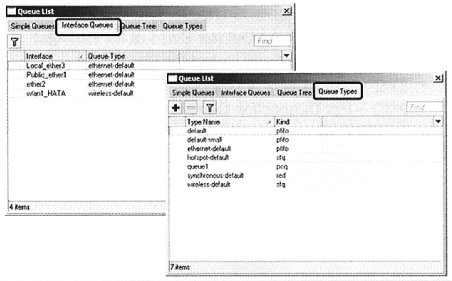
100% Shaper



100% Scheduler

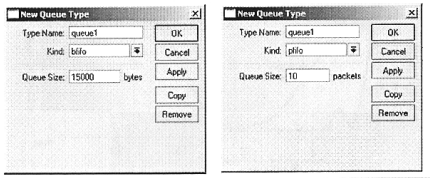


Default Queue Types

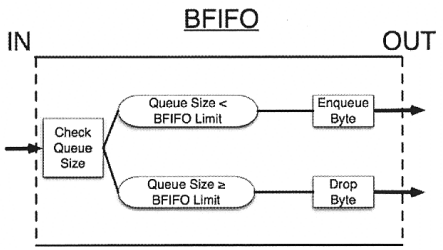


FIFO

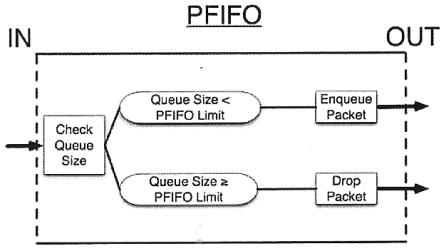
- Behaviour:
  - What comes in first is handled first, what comes in next waits until the first is finished. Number of waiting units (Packets or Bytes) is limited by "queue size" option. If queue "is full" next units are dropped



BFIFO

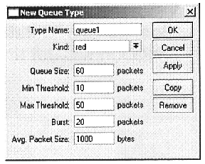


PFIFO

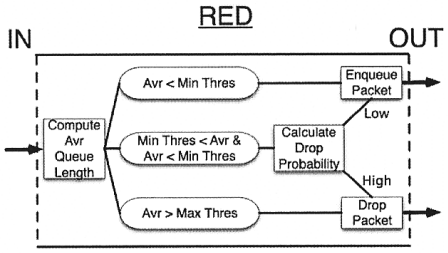


RED

- Behaviour:
  - Same as FIFO with feature – additional drop probability even if queue is not full.
- This probability is based oncomparison of average queue length over some period of time to minimal and maximal threshold – closer to maximal threshold bigger the chance of drop.

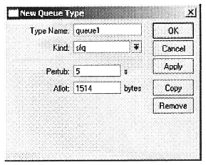


RED



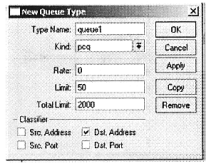
SFQ

- Behaviour:
  - Based on hash value from source and destination address SFQ divides traffic into 1024 sub-streams
- Then Round Robin algorithm will distribute equal amount of traffic to each sub-stream

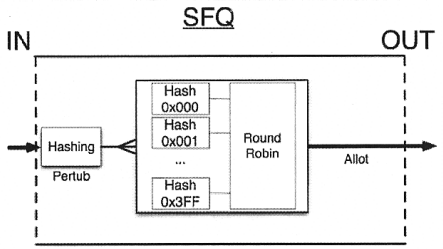


PCQ

- Behaviour:
  - Based on classifier PCQ divides traffic into substreams. Each sub-stream can be considered as FIFO queue with queue size specified by "limit" option
- After this PCQ can be considered as FIFO queue where queue size is specified by "total-limit" option.



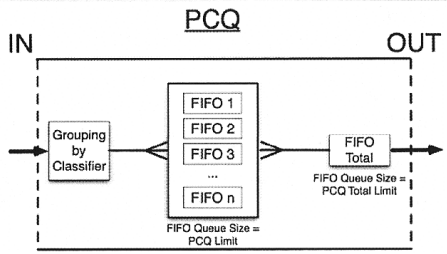
SFQ



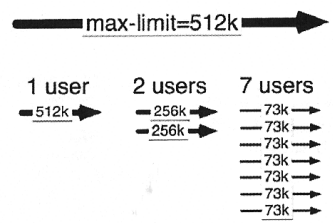
SFQ Example

- SFQ should be used for equalizing similar connection
- Usually used to manage information flow to or from the servers, so it can offer services to every customer
- Ideal for p2p limitation, it is possible to place strict limitation without dropping connections,

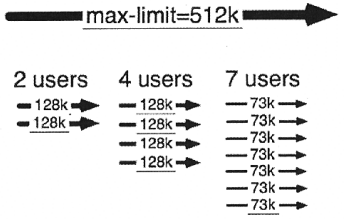
PCQ



pcq-rate = 0



pcq-rate = 128 000



Queue Type Lab

- Try all queue types on "Other-download" queue in your queue tree. Use band-width test to check it.
- Adjust your QoS structure with proper queue type
  - Create a packet mark for all p2p traffic and create SFQ queue for it
  - Change HTTP queue type to PCQ