MikroTik RouterOS (v6) Training
Traffic Control

Vahid Shahbazian fard jahromy www.LearnMikroTik.ir

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

Instructor

- Vahid Shahbazian fard jahromy
  - Training, Support & Consultant
  - Specialization: Wireless, Firewall, The Dude, Routing

Housekeeping

- Course materials
- Routers, cables
- Break times and lunch

Course Objective

- Provide knowledge and hands-on training for MikroTik RouterOS basic and advanced traffic control capabilities for any size networks
- Upon completion of the course you will be able to plan, implement, adjust and debug traffic control configurations implemented by MikroTik RouterOS.

Introduce Yourself

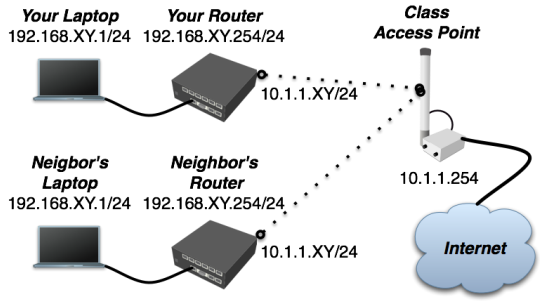
- Please, introduce yourself to the class
  - Your name
  - Your Company
  - Your previous knowledge about RouterOS
  - Your previous knowledge about networking
  - What do you expect from this course?
- Please, remember your class XY number.
  (X is number of the row, Y is your seat number in the row)

  My number is: X=_____ Y=_____

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 "MTCTCEclass"
- Assign IP address 10.1.1.XY/24 to the wlan1
- Router's main GW and DNS address is 10.1.1.254
- Gain access to the internet from your laptops via local router
- Create new full access user for your router and change "admin" access rights to "read"

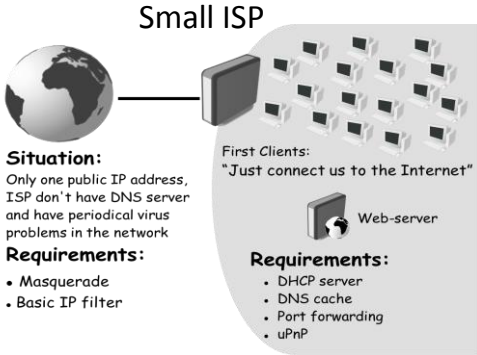
Class Setup



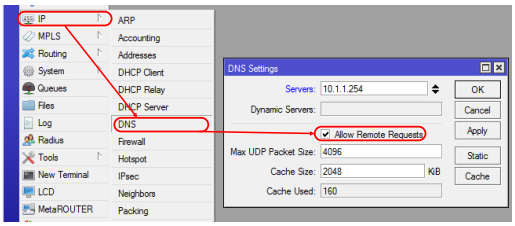
Class setup Lab (cont.)

- Set system identity of the board and wireless radio name to "XY_<your_name>". Example: "88_Shahbazian"
- Upgrade your router to the latest Mikrotik RouterOS version
- Upgrade your Winbox loader version
- Set up NTP client – use 10.1.1.254 as server
- Create a configuration backup and copy it to the laptop (it will be default configuration)

Small ISP



DNS Client and Cache

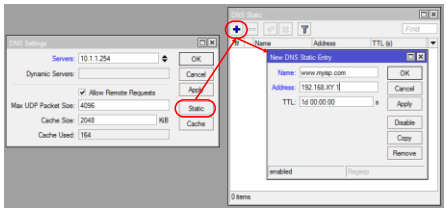


DNS Client and Cache

- 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 server
- Add static DNS entry “www.XY.com” to your router's Local IP address (XY – your number)
- Add static DNS entry “www.XY.com” to neighbour router's Public IP address (XY – your neighbours number)
- 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

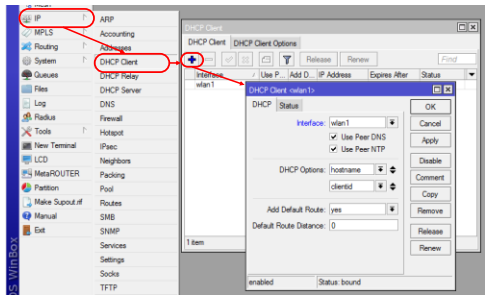
DHCP Communication scenario

- 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
  - 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)

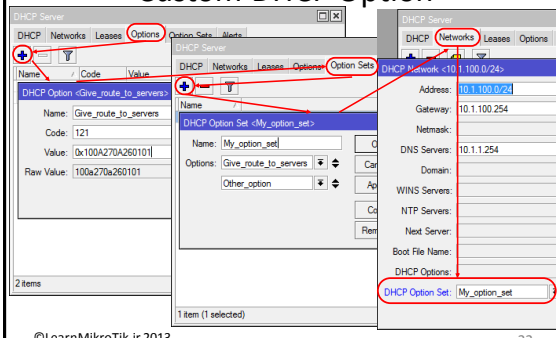
DHCP Networks

- 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

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"

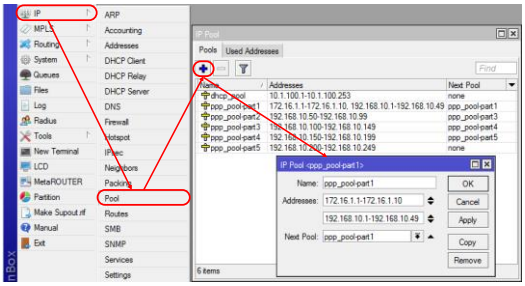
Custom DHCP Option



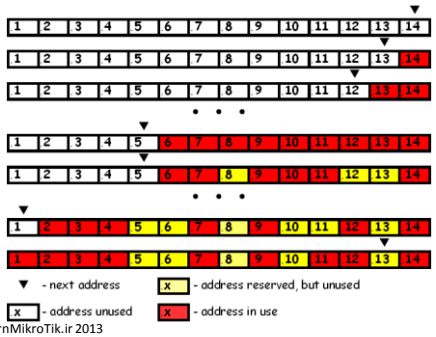
IP Address Pool

- 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 Pools



Address Pool in Action



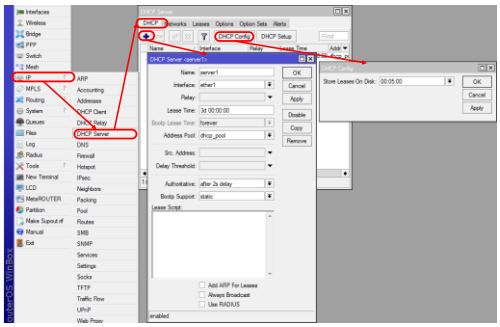
Other DHCP Server Settings

- **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 Server Leases

- **address:** Specify ip address (or ip pool) for static lease. If set to 0.0.0.0 - pool from server will be used
- **mac-address**: If specified, must match the MAC address of the client
- **client-id:** If specified, must match DHCP 'client identifier' option of the request
- **server:** Server name which serves this client

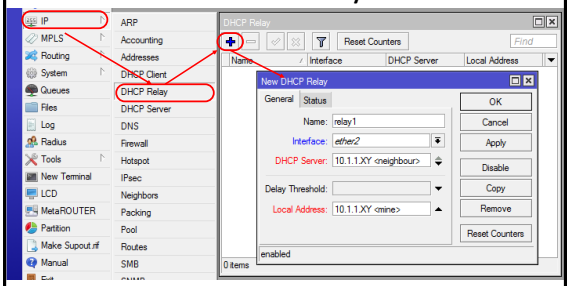
DHCP Server Leases (cont.)

- **block-access:** Block access for this client
- **always-broadcast:** Send all replies as broadcasts
- **rate-limit:** Sets rate limit for active lease. Format is: rx-rate[/tx-rate] [rx-burst-rate[/tx-burst-rate] [rx-burst-threshold[/tx-burst-threshold] [rx-burst-time[/tx-burst-time]]]]
- **address-list:** Address list to which address will be added if lease is bound.

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

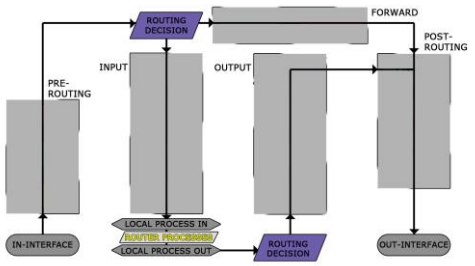
DHCP Relay



DHCP Lab

- Interconnect with your neighbour using Ethernet cable
- Create 3 independent setups:
  - Create DHCP server for your laptop
  - Create DHCP server and relay for your neighbour laptop (use relay option)
  - Create a bridged network with 2 DHCP servers and 2 DHCP clients (laptops) and try out "authoritative" and "delay threshold" options

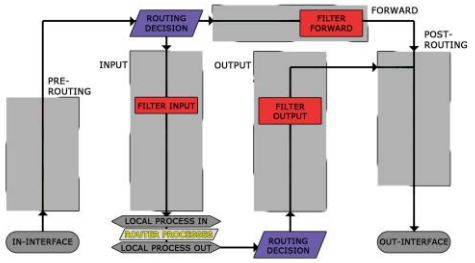
Simple Packet Flow structure



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

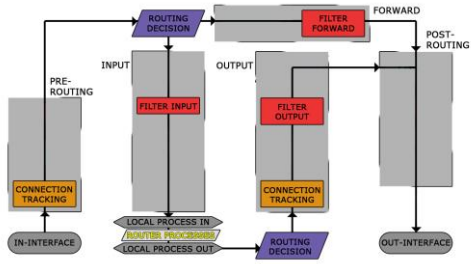
Firewall Filter Structure Diagram



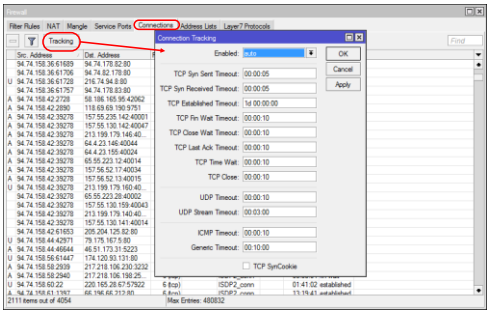
Connection Tracking

- 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)

Conntrack Placement



Conntrack – Winbox View

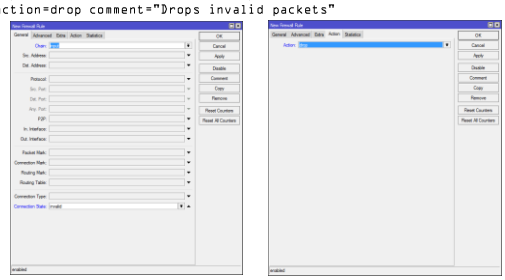


Condition: Connection State

- 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

First Rule Example

```
/ip firewall filter add chain=input connection-state=invalid\
action=drop comment="Drops invalid packets"
```



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

CHAIN INPUT

Connection State Lab

- Create 3 rules to ensure that only connection-state **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 2 rules to ensure that only you will be able to connect to the router
  - **Accept** all packets from your local network
  - **Drop** everything else

RouterOS Services

Nr.	Port	Protocol	Comment
1	20	TCP	FTP data connection
2	21	TCP	FTP control connection
3	22	TCP	Secure Shell (SSH)
4	23	TCP	Telnet protocol
5	53	TCP	DNS
6	80	TCP	World Wide Web HTTP
7	179	TCP	Border Gateway Protocol
8	443	TCP	Secure Socket Layer (SSL)
9	646	TCP	LDP transport session
10	1080	TCP	SOCKS proxy protocol
11	1723	TCP	PPTP
12	2828	TCP	Universal Plug and Play
13	2000	TCP	Bandwidth test server
14	8080	TCP	Web Proxy
15	8291	TCP	Winbox
16	8728	TCP	API
17	8729	TCP	API-SSL
18	---	/1	ICMP
19	---	/2	Multicast | IGMP
20	---	/4	IPIP encapsulation

Nr.	Port	Protocol	Comment
21	53	UDP	DNS
22	67	UDP	BootP or DHCP Server
23	68	UDP	BootP or DHCP Client
24	123	UDP	Network Time Protocol
25	161	UDP	SNMP
26	500	UDP	Internet Key Exchange (IPSec)
27	520	UDP	RIP routing protocol
28	521	UDP	RIP routing protocol
29	646	UDP	LDP hello protocol
30	1701	UDP	Layer 2 Tunnel Protocol
31	1900	UDP	Universal Plug and Play
32	5678	UDP	MNDP
33	20561	UDP	MAC winbox
34	---	/41	IPv6 (encapsulation)
35	---	/47	GRE (PPTP, EOIP)
36	---	/50	ESP (IPSec)
37	---	/51	AH (IPSec)
38	---	/89	OSPF routing protocol
39	---	/103	Multicast | PIM
40	---	/112	VRRP

RouterOS Service Lab

- Create a chain “services”
- Create rules to allow necessary RouterOS services to be accessed from the public network
- Create a “jump” rule from the chain “input” to the chain “services”
- Place a “jump” rule accordingly
- Write comment for each firewall rule
- Ask your neighbour to check the setup

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

CHAIN FORWARD

Virus Port Filter

- At the moment the are few hundreds active trojans and less than 50 active worms
- You can download the complete “virus port blocker” chain (~330 drop rules with ~500 blocked virus ports) from ftp://test@10.1.1.254
- Some viruses and trojans use standard services ports and can not be blocked.

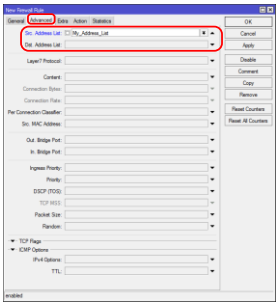
Chain Forward Lab

- Create 3 rules to ensure that only connection-state 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
- Import the virus.rsc file into the router
- Create a jump rule to the chain “virus”

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!!!
- You can find information about all unused IP ranges – judy google for “bogon IPs”

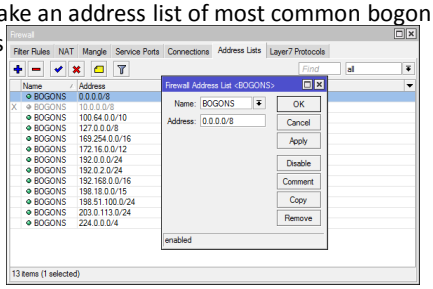
Address List Options



- 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 enter your network only from the valid Internet addresses
- Allow packets to enter your network only to the valid customer addresses
- Allow packets to leave your network only from the valid customers addresses
- Allow packets to leave your network only to the valid Internet addresses
- Place the rules accordingly

Destination NAT, Source NAT, NAT traversal

NETWORK ADDRESS TRANSLATION (NAT)

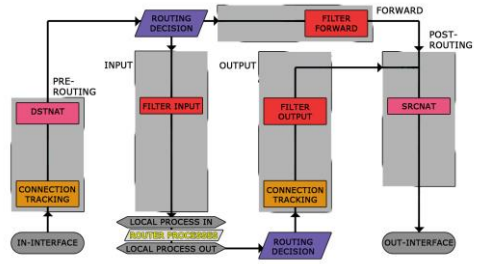
NAT Types

- 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

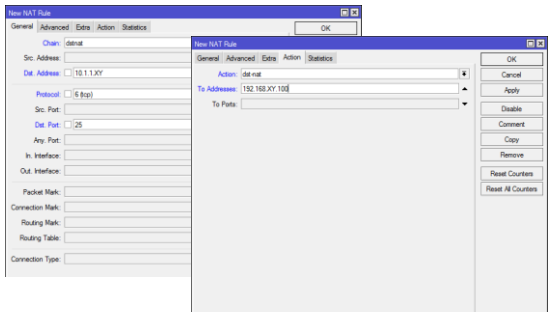
IP Firewall Diagram



Dst-nat Action

- 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

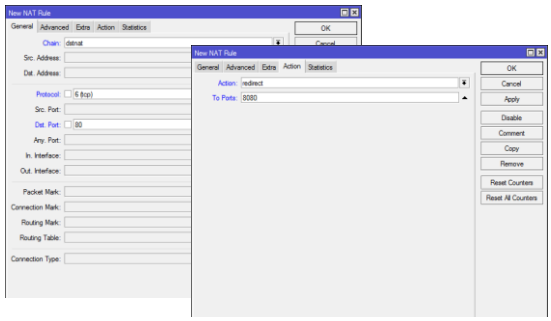
Dst-nat Rule Example



Redirect

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

Redirect Rule Example



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

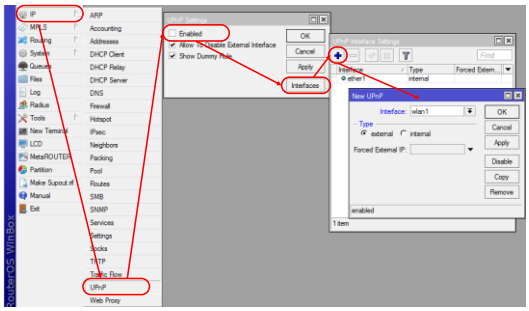
Dst-nat Lab

- Capture all TCP port 80 (HTTP) packets originated from your private network 192.168.XY.0/24 and change destination address to 10.1.2.1 using dst-nat rule
- Clear your browser's cache on the laptop
- Try browsing the Internet

Universal Plug-and-Play

- RouterOS allow to enable uPnP support for the router.
- UPnP allow to establish both-directional connectivity even if client is behind the NAT, client must have uPnP support
- There are two interface types for UPnP-enabled router: internal (the one local clients are connected to) and external (the one the Internet is connected to)

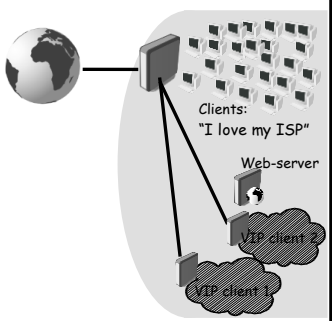
UPnP



First VIP clients

Situation:
You have public IP address and /30 subnet of public addresses,
You sometimes reach ISP speed limitation (5Mbps/5Mbps)

Requirements:
- Public IP address for VIP clients
- Guaranteed speed for VIP clients

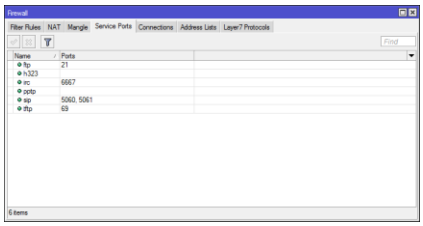


Source NAT Drawbacks

- 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



Src-nat Lab

- You have been assigned one “public” IP address 172.16.0.XY/32
- Assign it to the wireless interface
- Add src-nat rule to “hide” your private network 192.168.XY.0/24 behind the “public” address
- Connect from your laptop using winbox, ssh, or telnet via your router to the main gateway 10.1.1.254
- Check the IP address you are connecting from (use “/user active print” on the main gateway)

IP packet marking and IP header fields adjustment

FIREWALL MANGLE

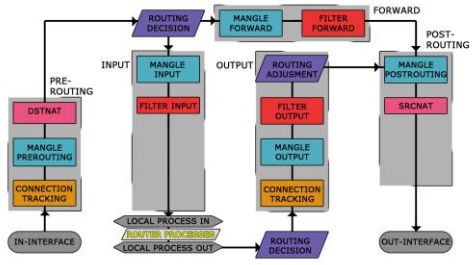
What is Mangle?

- 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 dstnat
  - Postrouting: making a mark before srcnat
  - 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

Firewall Diagram (Filter, NAT and Mangle)



Mangle actions

- There are 9 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

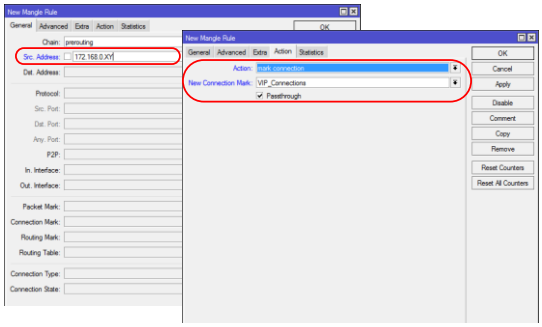
Mangle actions (cont.)

- change-dscp - change Differentiated Services Code Point (DSCP) field value
- set-priority - set priority on the packets sent out through a link that is capable of transporting priority (VLAN or WMM-enabled wireless interface).

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

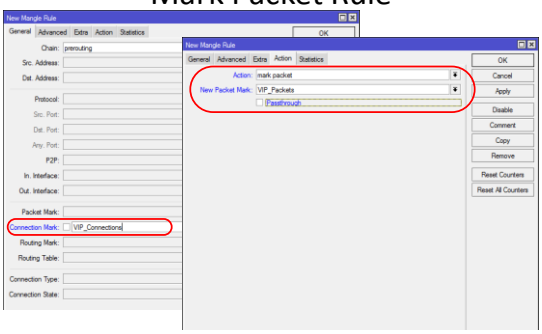


Marking Packets

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)

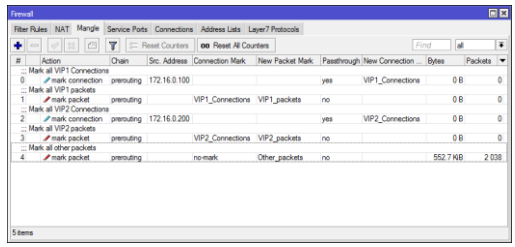
Mark Packet Rule



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



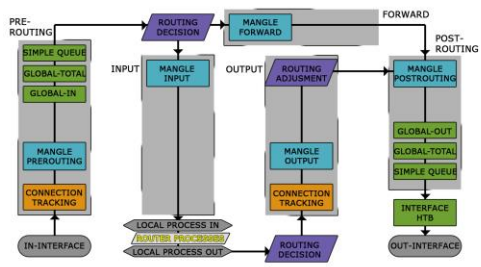
Hierarchical Token Bucket

HTB

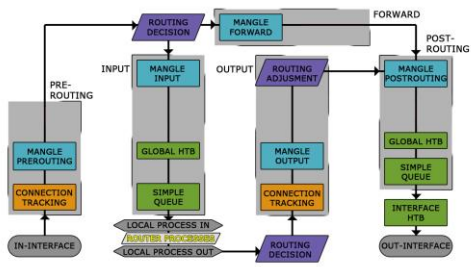
HTB

- 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 v5 or older versions support 3 virtual HTBs (global-in, global-total, global-out) and one more just before every interface
- RouterOS v6 support 1 virtual HTB (global) and one more just before every interface

Mangle and HTBs in RouterOS v5 or older versions



Mangle and HTBs in RouterOS v6



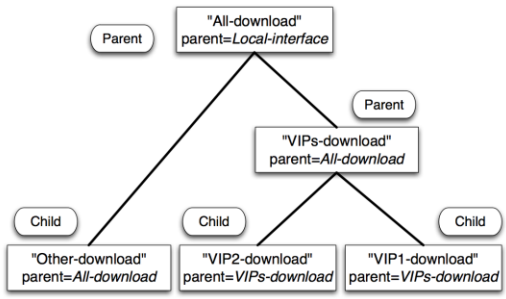
HTB (cont.)

- In RouterOS v5 or older versions when packet travels through the router, it passes all 4 HTB trees
- In RouterOS v5 or older versions when packet travels to the router, it passes only global-in and global-total HTB.
- In RouterOS v5 or older versions 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 will get limit-at first and then rest of the traffic will distributed by parents

HTB Features - Structure



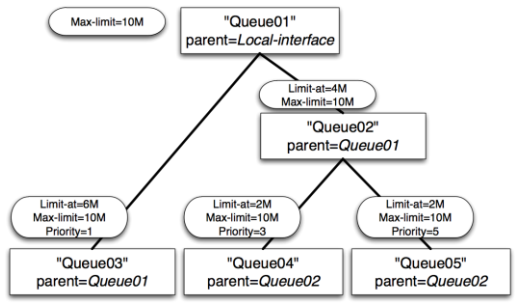
HTB Features – Dual Limitation

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

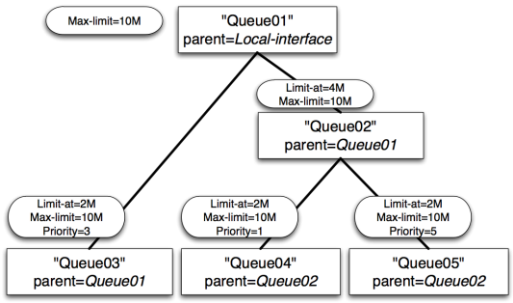
Dual Limitation

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

HTB - limit-at



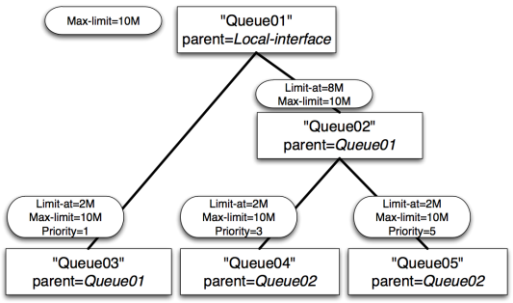
HTB - max-limit



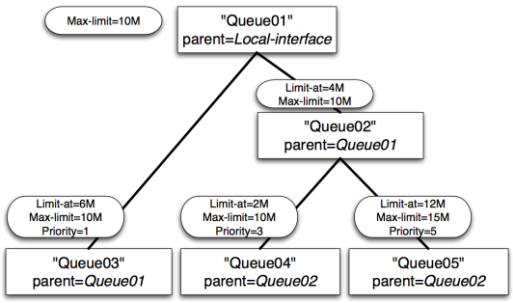
HTB Features - Priority

- Work only for child queues to arrange them
- 8 is the lowest priority, 1 is the highest
- Queue with higher priority will get chance to satisfy its max-limit before other queues
- Actual traffic prioritization will work only if limits are specified. Queue without limits will not prioritize anything

HTB – limit-at of the Parent



HTB – limit-at > parent's max-limit

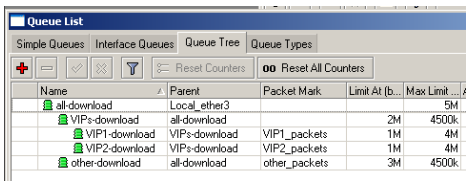


Advanced queue structures

QUEUE TREE

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 Queues

- Tree queue in RouterOS v5 or older versions 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
- In RouterOS v6 can be placed in **global** or **interface queue**
- In RouterOS v5 or older versions if placed in same place Simple queue will take traffic before Queue Tree
- In RouterOS v6 if placed in same place Simple queue will take traffic after Queue Tree

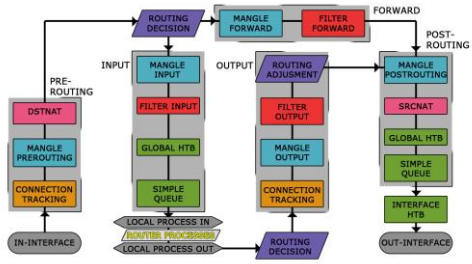
HTB Lab

- 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
- (optional) Create the same queue tree for client upload

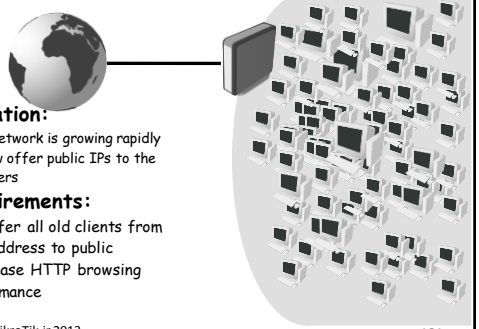
HTB Lab (cont.)

- 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

Simple Packet Flow in RouterOS v6



Medium Size ISP



Situation:

Your network is growing rapidly and now offer public IPs to the customers

Requirements:

- Transfer all old clients from local address to public
- Increase HTTP browsing performance

NAT Action "Netmap"

- Can be used in both (srcnat and dstnat) chains
- Allows to create address range to address range NATing only with one rule
- It is possible to masquerade 192.168.88.0/24 to 88.188.32.0/24 only with one rule
- It is possible to redirect 88.188.32.0/24 to 192.168.88.0/24 with the second rule

NAT Action “same”

- Can be used in both (srcnat and dstnat) chains
- Ensures that client will be NAT'ed to the same address from the specified range every time it tries to communicate with destination that was used before
- If client got 88.188.32.104 from the range when it communicated to the particular server – every next time communicating with this server it will use the same address

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)

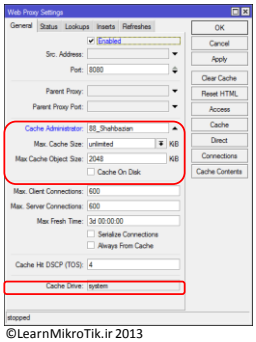
Burst

Burst (Part 2)

Web-Proxy

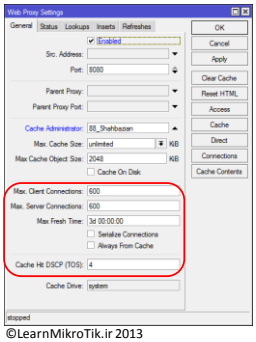
- Web-proxy have 3 mayor features
  - HTTP and FTP traffic caching
  - DNS name filtering
  - DNS redirection
- Web-proxy have two operation modes
  - Regular – browser must be configured to use this proxy
  - Transparent – this proxy is not visible for customers NAT rules must be applied

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
- Max cache object size
- Cache drive

Web-Proxy Options

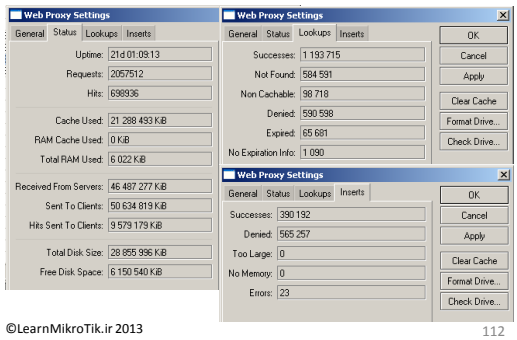


- Maximal-client-connections: number of connections accepted from clients
- Maximal-server-connections: number of connections made by server

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

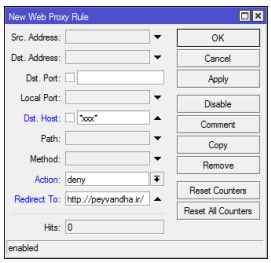
Web-Proxy Statistics



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*

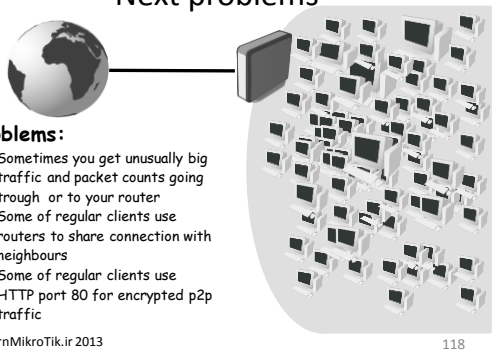
Regular Expressions

- Place ":" at the beginning to enable regular expression mode
  - "^" - show that no symbols are allowed before the given pattern
  - "$" - show that no symbols are allowed after the given pattern
  - "[....]" - A character class matches a single character out of all the possibilities offered by the character class
  - \ (backslash) followed by any of [\^$.|?*+() suppress their special meaning.

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

Next problems



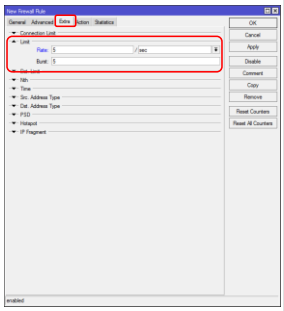
Problems:

- Sometimes you get unusually big traffic and packet counts going trough or to your router
- Some of regular clients use routers to share connection with neighbours
- Some of regular clients use HTTP port 80 for encrypted p2p traffic

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

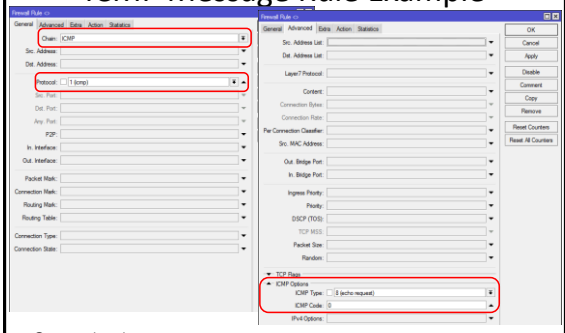


- Ping flood usually consist from volumes of random ICMP messages
- With "limit" condition it is possible to bound the rule match rate to a given limit
- This condition is often used with action "log"

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

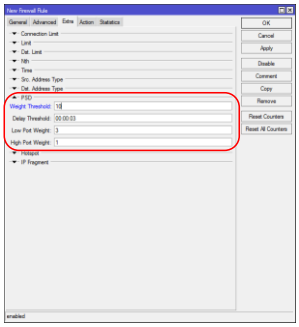
ICMP Message Rule Example



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

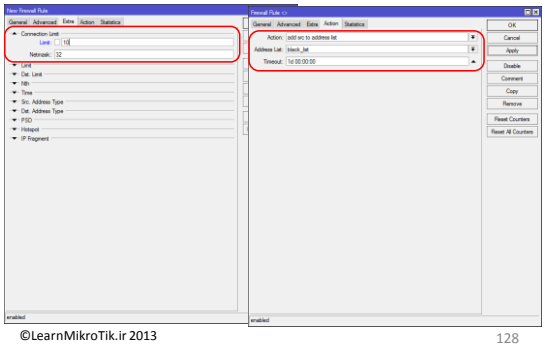
DoS Attacks

- 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

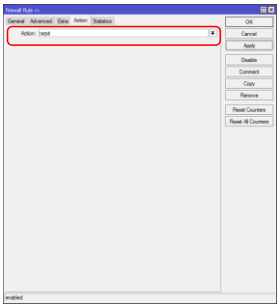
- All IP's with more than 10 connections to the router should be considered as DoS attackers
- With every dropped TCP connection we will allow attacker to create new connection
- We should implement DoS protection into 2 steps:
- Detection - Creating a list of DoS attackers on the basis of connection-limit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

DoS Attack Detection



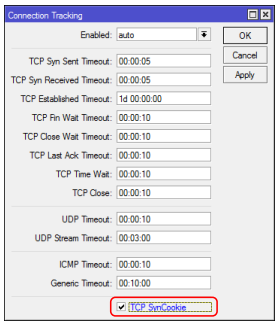
DoS Attack Suppression

- To stop the attacker from creating new connections, we will use action “tarpit”
- You must set protocol=tcp for tarpit rule
- We must place this rule before the detection rule or else address-list entry will rewrite all the time



DDoS attacks

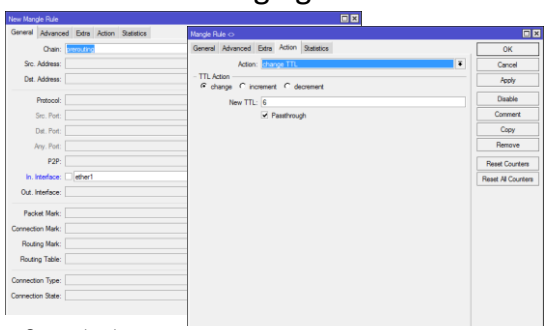
- A Distributed Denial of Service attack is very similar to DoS attack only it occurs from multiple compromised systems
- Only thing that could help is “TCPSyn Cookie” option in conntrack system



Mangle Action “change-ttl”

- TTL is a limit of Layer3 devices that IP packet can experience before it should be discarded
- TTL default value is 64 and each router reduce value by one just before forwarding decision
- Router will not pass traffic to the next device if it receives IP packet with TTL=1
- Useful application: eliminate possibility for clients to create masqueraded networks

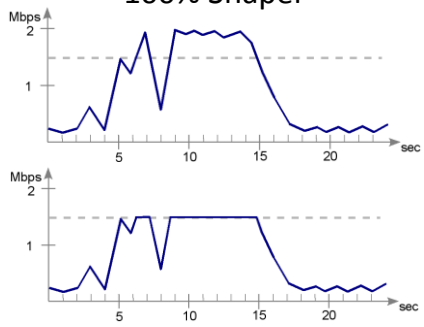
Changing TTL



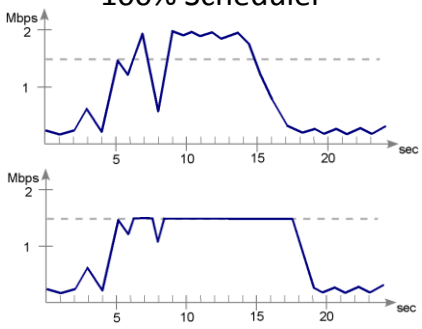
Queue Types

- 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

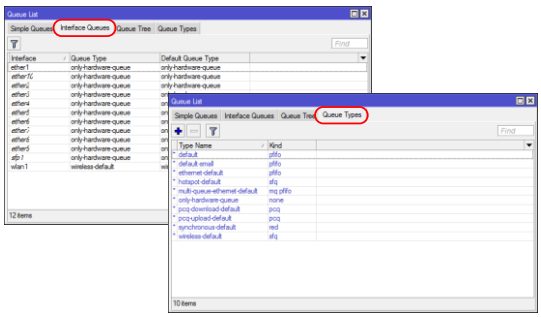
100% Shaper



100% Scheduler



Default Queue Types



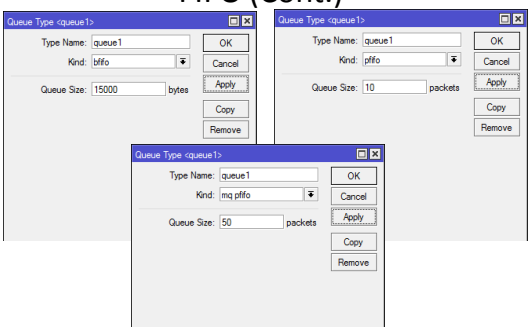
FIFO

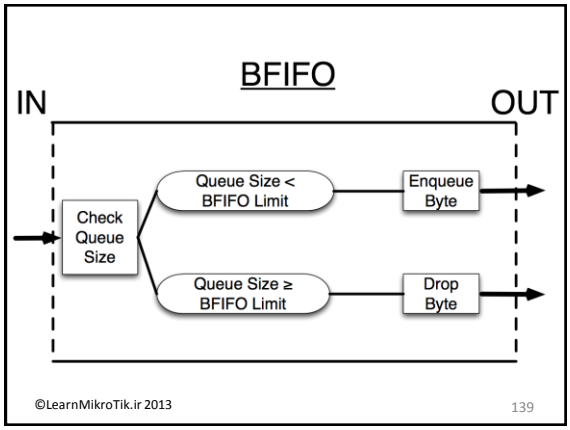
Behaviour:

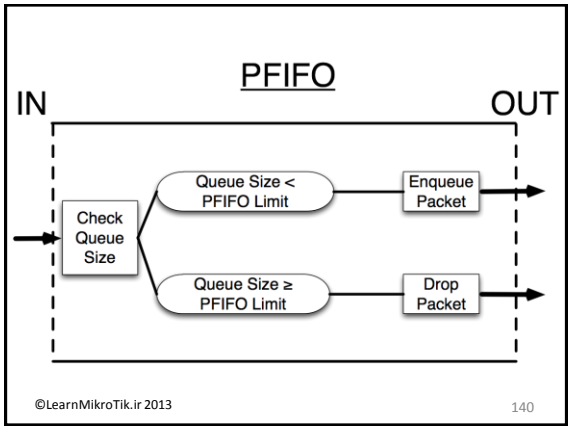
What comes in first is handled first, what comes in next waits until the first is finished. Number of waiting units (Packets or Bytes) is limited by "queue size" option. If queue "is full" next units are dropped

mq-pfifo is **pfifo** with support for multiple transmit queues.

FIFO (Cont.)





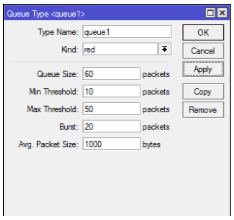


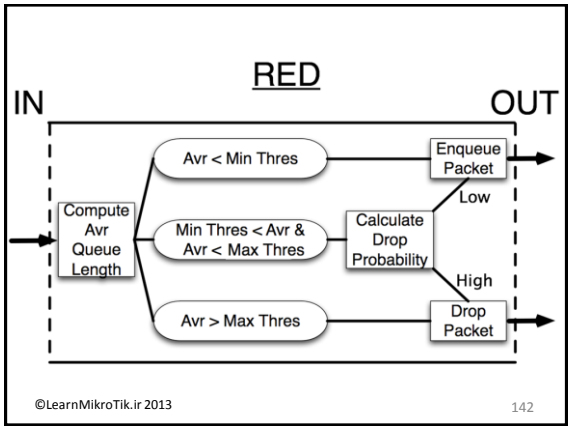
RED

Behaviour:

Same as FIFO with feature – additional drop probability even if queue is not full.

This probability is based on comparison of average queue length over some period of time to minimal and maximal threshold – closer to maximal threshold bigger the chance of drop.



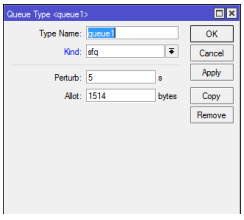


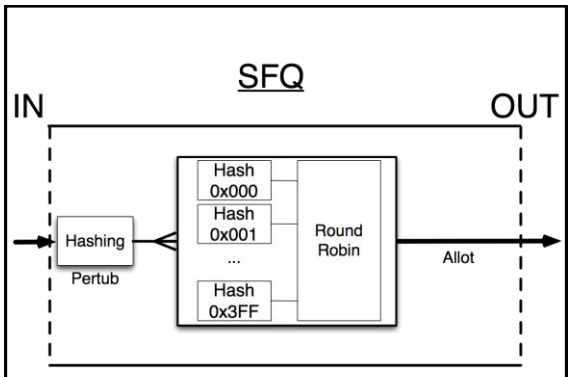
SFQ

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





SFQ Example

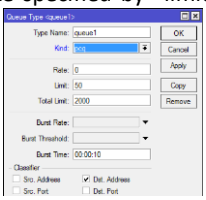
- SFQ should be used for equalizing similar connection
- Usually used to manage information flow to or from the servers, so it can offer services to every customer
- Ideal for p2p limitation, it is possible to place strict limitation without dropping connections.

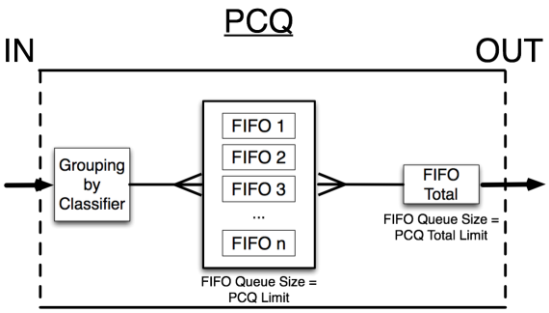
PCQ

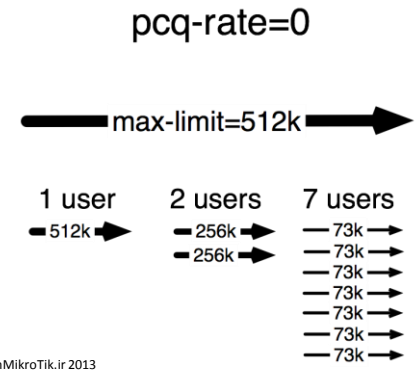
Behaviour:

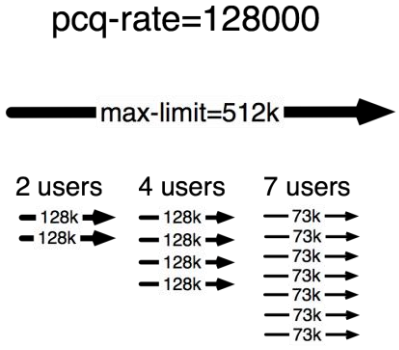
Based on classifier PCQ divides traffic into sub-streams. Each sub-stream can be considered as FIFO queue with queue size specified by "limit" option

After this PCQ can be considered as FIFO queue where queue size is specified by "total-limit" option.









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

Packet Sniffer

- Packet sniffer is a tool that can capture and analyze packets that are going to, leaving or going through the router
- Packet sniffer can store result in router's memory, to a file or send sniffed packets to streaming server
- The filter can be used to curb the packets

NTH

- Matches every nth packet
- It has only two parameters 'every' and 'packet'.
- Every rule has its own counter. When rule receives packet counter for current rule is increased by one.
- If counter matches value of 'every' packet will be matched and counter will be set to zero.

Per Connection Classifier

- PCC matcher allows to divide traffic into equal streams with ability to keep packets with specific set of options in one particular stream.
- PCC is introduced to address configuration issues with load balancing over multiple gateways with masquerade