Understanding throughput

OR

Common misconceptions on what is ’real’ device throughput

MUM, Europe 2019
Objectives

● To help you understand how to properly test and improve the throughput results on RouterOS devices

● To allow MikroTik equipment do more

● Encourage not only to update RouterOS version but also update existing configurations to use the latest features

● Reduce the amount of throughput related emails to support@mikrotik.com!
What is throughput?

- Throughput is a measure of how many units of information a system can process, in a given amount of time.

- In data transmission, network throughput is the amount of data transferred successfully from one place to another in a given time period and typically measured in bits per second (bps).
Router throughput

- Successfully transferred data through the router is equal to sum of all data that is leaving the router (not dropped in the router)
Maximum router throughput

- 1500Mbps
- 750Mbps
- 950Mbps
- 948Mbps

<table>
<thead>
<tr>
<th>Mode</th>
<th>Configuration</th>
<th>QCA9558 1G all port test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1518 byte</td>
<td>512 byte</td>
</tr>
<tr>
<td></td>
<td>kpps</td>
<td>Mbps</td>
</tr>
<tr>
<td>Bridging</td>
<td>none (fast path)</td>
<td>161.9</td>
</tr>
<tr>
<td>Bridging</td>
<td>25 bridge filter rules</td>
<td>125.5</td>
</tr>
<tr>
<td>Routing</td>
<td>none (fast path)</td>
<td>161.9</td>
</tr>
<tr>
<td>Routing</td>
<td>25 simple queues</td>
<td>161.9</td>
</tr>
<tr>
<td>Routing</td>
<td>25 ip filter rules</td>
<td>78.1</td>
</tr>
</tbody>
</table>
Link or wire speed

- Link or Wire speed refers to the rate of data transfer a given telecommunication technology provides at the physical wire level
  - Ethernet - 100Mbps/1Gbps/2.5Gbps/5Gbps/10Gbps
  - SFP – 1Gbps
  - SFP+ - 10Gbps
  - SFP28 - 25Gbps
  - QSFP+ - 40Gbps
  - QSFP28 - 100Gbps
Wire-speed

• Wire-speed, as an adjective, describes any hardware or function that supports data transfer rate without slowing it down

• Functions embedded in microchips (ASIC) are more likely to run at wire-speed than functions that are implemented in software

• Currently in MikroTik hardware all devices with a switch chip are capable of transferring data at wire-speed while using an impressive set of features
Wireless "wire speed" (Data rates)

- Maximum *theoretical* wireless speed is determined by wireless protocol, number of streams, modulation and channel width

<table>
<thead>
<tr>
<th>802.11n</th>
<th>Streams</th>
<th>Modulation</th>
<th>20MHz</th>
<th>40MHz</th>
<th>80MHz</th>
<th>160MHz</th>
<th>802.11ac</th>
<th>VHT MCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HT MCS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>64-QAM 5/6</td>
<td>72.2</td>
<td>150</td>
<td>325</td>
<td>650</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>256-QAM 5/6</td>
<td>n/a</td>
<td>200</td>
<td>433.3</td>
<td>866.7</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>64-QAM 5/6</td>
<td>144.4</td>
<td>300</td>
<td>650</td>
<td>1300</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>256-QAM 5/6</td>
<td>n/a</td>
<td>400</td>
<td>866.7</td>
<td>1733.3</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>64-QAM 5/6</td>
<td>216.7</td>
<td>450</td>
<td>975</td>
<td>1950</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>256-QAM 5/6</td>
<td>288.9</td>
<td>600</td>
<td>1300</td>
<td>n/a</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>4</td>
<td>64-QAM 5/6</td>
<td>288.9</td>
<td>600</td>
<td>1300</td>
<td>2600</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>256-QAM 5/6</td>
<td>n/a</td>
<td>800</td>
<td>1733.3</td>
<td>3466.7</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>
# 802.11ad Data rates

<table>
<thead>
<tr>
<th>MCS</th>
<th>Modulation</th>
<th>NCBPS</th>
<th>Repetition</th>
<th>Code rate</th>
<th>Data rate (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DBPSK</td>
<td></td>
<td></td>
<td>1/2</td>
<td>27.5</td>
</tr>
<tr>
<td>1</td>
<td>π/2 BPSK</td>
<td>1</td>
<td>2</td>
<td>1/2</td>
<td>385</td>
</tr>
<tr>
<td>2</td>
<td>π/2 BPSK</td>
<td>1</td>
<td>1</td>
<td>1/2</td>
<td>770</td>
</tr>
<tr>
<td>3</td>
<td>π/2 BPSK</td>
<td>1</td>
<td>1</td>
<td>5/8</td>
<td>962.5</td>
</tr>
<tr>
<td>4</td>
<td>π/2 BPSK</td>
<td>1</td>
<td>1</td>
<td>3/4</td>
<td>1155</td>
</tr>
<tr>
<td>5</td>
<td>π/2 BPSK</td>
<td>1</td>
<td>1</td>
<td>13/16</td>
<td>1251.25</td>
</tr>
<tr>
<td>6</td>
<td>π/2 QPSK</td>
<td>2</td>
<td>1</td>
<td>1/2</td>
<td>1540</td>
</tr>
<tr>
<td>7</td>
<td>π/2 QPSK</td>
<td>2</td>
<td>1</td>
<td>5/8</td>
<td>1925</td>
</tr>
<tr>
<td>8</td>
<td>π/2 QPSK</td>
<td>2</td>
<td>1</td>
<td>3/4</td>
<td><strong>2310</strong></td>
</tr>
<tr>
<td>9</td>
<td>π/2 QPSK</td>
<td>2</td>
<td>1</td>
<td>13/16</td>
<td>2502.5</td>
</tr>
<tr>
<td>10</td>
<td>π/2 16QAM</td>
<td>4</td>
<td>1</td>
<td>1/2</td>
<td>3080</td>
</tr>
<tr>
<td>11</td>
<td>π/2 16QAM</td>
<td>4</td>
<td>1</td>
<td>5/8</td>
<td>3850</td>
</tr>
<tr>
<td>12</td>
<td>π/2 16QAM</td>
<td>4</td>
<td>1</td>
<td>3/4</td>
<td>4620</td>
</tr>
</tbody>
</table>
Marketing numbers

802.11ac Quad Stream plus 802.11ad WiFi
Up to 4600+1733+800 Mbps† wireless speed

Performance

208 Gbps on 24-port Gigabit Ethernet model
256 Gbps on 48-port Gigabit Ethernet model
640 Gbps on 24-port Multigigabit Ethernet model
580 Gbps on 48-port 2.5G (12 Multigigabit) Ethernet model
640 Gbps on 48-port 5G Ethernet model
All models are wire-speed nonblocking performance

• Implicit/Explicit Beamforming for 2.4 & 5 GHz bands (1,733 + 866 + 400 Mbps)†
Capacity

- Network device capacity is a measure of the device structure’s bandwidth
  - Example: **CRS309-1G-8S+IN** – has a structure of one Gigabit Ethernet and 8x 10Gbps SFP+ ports
    
    \[(8\times10G + 1\times1G) \times 2 \text{ (for duplex)} = 162\text{Gbps}\]
  
  - Example: **RBwAPG-5HacT2HnD** – has a structure of one Gigabit Ethernet and one triple stream 802.11ac wireless and one dual stream 802.11n wireless
    
    \[1G \times 2\text{ (for duplex)} + 1.3G + 0.3G = 4.6\text{Gbps}\]
Ethernet preamble and interframe gap

- Ethernet is a self-clocked digital protocol. The clock is synchronized from preamble field that provides a predictable **7 bytes long** signal for Ethernet receiver, followed by **1 byte** Start-of-Frame Delimiter

- Ethernet specifies minimum idle period between transmission of Ethernet frames known as the interframe gap – the time it takes to send **12 bytes**

---

<table>
<thead>
<tr>
<th>7 Bytes</th>
<th>1 Byte</th>
<th>14 Bytes</th>
<th>46 to 1500 Bytes</th>
<th>4 Bytes</th>
<th>12 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preamble</strong></td>
<td><strong>SFD</strong></td>
<td><strong>Ethernet Header</strong></td>
<td><strong>Layer3 Data</strong></td>
<td><strong>Frame Check Sequence</strong></td>
<td><strong>Inter Frame Gap</strong></td>
</tr>
</tbody>
</table>
Theoretical Layer 2 throughput

- 64 byte Layer 2 packet
  - actually consumes 102 bytes on the wire
  - \( \frac{64}{102} = 62.45\% \)
Theoretical Layer2 throughput

- 1500 byte Layer2 packet
  - actually consumes 1538 bytes on the wire
  - $\frac{1500}{1538} = 97.53\%$
# Switching results

<table>
<thead>
<tr>
<th>Mode</th>
<th>Configuration</th>
<th>1518 byte</th>
<th></th>
<th>512 byte</th>
<th></th>
<th>64 byte</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>kpps</td>
<td>Mbps</td>
<td>kpps</td>
<td>Mbps</td>
<td>kpps</td>
<td>Mbps</td>
</tr>
<tr>
<td>Switching</td>
<td>Non blocking Layer 2 throughput</td>
<td>6,583.2</td>
<td>79,946.7</td>
<td>19,032.0</td>
<td>77,954.9</td>
<td>120,535</td>
<td>61,714.3</td>
</tr>
<tr>
<td>Switching</td>
<td>Non blocking Layer 2 capacity</td>
<td>6,583.2</td>
<td>159,893.4</td>
<td>19,032.0</td>
<td>155,909.8</td>
<td>120,535</td>
<td>123,428.6</td>
</tr>
<tr>
<td>Switching</td>
<td>Non blocking Layer 1 throughput</td>
<td>6,583.2</td>
<td>81,000.0</td>
<td>19,032.0</td>
<td>81,000.0</td>
<td>120,535</td>
<td>81,000.0</td>
</tr>
<tr>
<td>Switching</td>
<td>Non blocking Layer 1 capacity</td>
<td>6,583.2</td>
<td>162,000.0</td>
<td>19,032.0</td>
<td>162,000.0</td>
<td>120,535</td>
<td>162,000.0</td>
</tr>
</tbody>
</table>
“Real” network speed
How does speedtest.net work

• speedtest.net will use up to four HTTP threads
  • After the pre-test, if the connection speed is at least 4 megabits per second then speedtest.net will use four threads. Otherwise, it will default to two

• One side sends an initial chunk of data, the other side calculates the real-time speed of the transfers and adjusts the chunk size along with buffer size
  • All samples are sorted by speed. The two fastest results are removed and the bottom 1/4 which is then left. Everything else is then averaged to get the result

*https://support.speedtest.net/hc/en-us/articles/203845400-How-does-the-test-itself-work-How-is-the-result-calculated-
TCP theoretical throughput

TCP throughput is limited by two windows

- **Congestion window** - sender side
  - the amount of unacknowledged packets that may be in transit
  - it auto-tunes based on congestion control algorithms
  - impacted by packet loss and delays

- **Receive window** - receiver side
  - the amount of received data not processed yet by the application
  - it auto-tunes based on receive buffer size and its level of fullness
TCP theoretical throughput

- Sender and receiver continuously negotiate common transmission window (for both directions)
- Maximal single TCP stream throughput is limited to not more than one full transmission window within one round-trip time (RTT) period
- Middle devices, like routers, have only indirect impact on congestion algorithm used - by impacting RTT, jitter, packet loss
TCP theoretical throughput

- If window size is 4,225,024 bytes, and RTT is 60ms
  - \( \frac{4,225,024 \times 8}{60} = 563,336 \text{ bits/ms} = 563 \text{ Mbps} \)
- If window size is 18,944 bytes, and RTT is 6ms
  - \( \frac{18,944 \times 8}{6} = 25,259 \text{ bits/ms} = 25,26 \text{ Mbps} \)
TCP and multi-threading

- Multi-threading introduces out-of-order packets
- Usually TCP can handle out-of-order packets within a single transmission window
- Depending on the congestion algorithm used, out-of-order packets might be considered as loss and introduce delays (increase calculated RTT)
Single TCP stream and multi-threading

Order of arrival

A_{core 1}
B_{core 2}
C_{core 3}
D_{core 4}

Time

\( t_A \) \( t_D \) \( t_C \) \( t_B \)
Flow/Packet steering

- RouterOS uses Receive Flow/Packet steering to assign incoming traffic to a specific CPU core/thread, based on the hash value.

- The hashing process can be:
  - Hard-coded in the hardware
  - Configurable in the hardware
  - Implemented in the interface driver

- Receive flow/packet steering will try to keep single TCP stream bound to single CPU core/thread as long as possible.
* Switch chip 2 (Eth6 - Eth10) has 2Gb/s aggregated lane to CPU until SFP module is inserted in the SFP1
What can and can’t we do...?

- We can’t choose the congestion control algorithm
- We can’t determine the congestion and the receive window size of the endpoints

• We can change MSS
• We can minimize impact on RTT by reducing packet processing time
• We can impact packet loss
Routing forwarding
Routing forwarding

1. PREROUTING
2. ROUTING DECISION
3. FORWARD
4. POSTROUTING
5. IPSEC POLICY?
6. IPSEC DECRYPTION
7. NO
8. INPUT
9. OUTPUT
10. IPSEC ENCRYPTION
11. ROUTING DECISION
12. YES
13. NO
14. IPSEC POLICY?
Routing forwarding

1. PREROUTING
   - HOTSPOT-IN
   - CONNECTION TRACKING
   - MANGLE PREROUTING
   - DST-NAT

2. INPUT
   - MANGLE INPUT
   - FILTER INPUT
   - HTB GLOBAL (QUEUE TREE)
   - SIMPLE QUEUES

3. FORWARD
   - BRIDGE DECISION
   - TTL=TTL-1
   - MANGLE FORWARD
   - FILTER FORWARD
   - ACCOUNTING

4. OUTPUT
   - BRIDGE DECISION
   - CONNECTION TRACKING
   - MANGLE OUTPUT
   - FILTER OUTPUT
   - ROUTING ADJUSTMENT

5. POSTROUTING
   - MANGLE POSTROUTING
   - SRC-NAT
   - HOTSPOT-OUT
   - HTB GLOBAL (QUEUE TREE)
   - SIMPLE QUEUES
Initial FastPath implementation

- FastPath is an interface driver extension, that allows you to receive/process/send traffic without unnecessary processing.
- Interface driver can now talk directly to specific RouterOS facilities - skipping others.
- FastPath requirements:
  - Interface driver support
  - FastPath should be allowed in configuration
  - No configuration in specific facilities
Routing forwarding FastPath

1. PREROUTING → HOTSPOT-IN → CONNECTION TRACKING → MANGLE PREROUTING → DST-NAT
2. INPUT → MANGLE INPUT → FILTER INPUT → HTB GLOBAL (QUEUE TREE) → SIMPLE QUENUES
3. FORWARD → BRIDGE DECISION → TTL=TTL-1 → MANGLE FORWARD → FILTER FORWARD → ACCOUNTING
4. OUTPUT → BRIDGE DECISION → CONNECTION TRACKING → MANGLE OUTPUT → FILTER OUTPUT → ROUTING
   ADJUSTMENT
5. POSTROUTING → MANGLE POSTROUTING → SRC-NAT → HOTSPOT-OUT → HTB GLOBAL (QUEUE TREE) → SIMPLE QUENUES

FASTPATH → ANY CONFIG? → NO
FastPath + Conntrack

- Implemented as “fasttrack-connection” action for firewall filter/mangle that adds “Fasttracked” flag to connection
- Packets from “Fasttracked” connections are allowed to travel in FastPath
- Works only with IPv4/TCP and IPv4/UDP
- Some packets will still follow the regular path to maintain timeouts in conntrack entries
FastPath + Conntrack = FastTrack

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Timeout</th>
<th>TCP State</th>
<th>Orig./Repl. Rate</th>
<th>Orig./Repl. Bytes</th>
<th>Orig./Repl. Packets</th>
<th>Orig./Repl. Fasttrack Bytes</th>
<th>Orig./Repl. Fasttrack Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>SACFs</td>
<td>6/tcp</td>
<td>1d 00:04:02</td>
<td>54.4 kbps/1546.4 kbps</td>
<td>141.0 MB/3662.3 MB</td>
<td>2737 217/2717</td>
<td>141.0 MB/3662.1 MB</td>
<td>2737 213/2716 883</td>
</tr>
<tr>
<td>SACFd</td>
<td>17 (udp)</td>
<td>00:05:01</td>
<td>1984 bps/34.6 kbps</td>
<td>3107.7 KB/6.5 MB</td>
<td>9 070/10 870</td>
<td>3107.1 KB/6.5 MB</td>
<td>9 068/10 869</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:04:33</td>
<td>0 bps/0 bps</td>
<td>2653.7 KB/3491.0 KB</td>
<td>6 630/5 828</td>
<td>2653.3 KB/3490.9 KB</td>
<td>6 628/5 826</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:04:51</td>
<td>0 bps/0 bps</td>
<td>445.5 KB/50.6 KB</td>
<td>4 842/477</td>
<td>445.0 KB/50.2 KB</td>
<td>4 836/474</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:04:55</td>
<td>0 bps/0 bps</td>
<td>858.6 KB/3085.5 KB</td>
<td>4 711/4 608</td>
<td>858.3 KB/3085.4 KB</td>
<td>4 709/4 607</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:05:03</td>
<td>39.7 kbps/3.6 kbps</td>
<td>2856.8 KB/507.5 KB</td>
<td>4 566/3 922</td>
<td>2856.3 KB/507.4 KB</td>
<td>4 564/3 921</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:01:52</td>
<td>1997.0 KB/2866.6 KB</td>
<td>4 536/3 754</td>
<td>1996.3 KB/2866.6 KB</td>
<td>4 534/3 753</td>
<td>1996.3 KB/2866.6 KB</td>
</tr>
<tr>
<td>SACFs</td>
<td>6 (tcp)</td>
<td>1d 00:03:32</td>
<td>0 bps/0 bps</td>
<td>922.7 KB/367.4 KB</td>
<td>4 406/4 659</td>
<td>920.3 KB/366.9 KB</td>
<td>4 399/4 649</td>
</tr>
<tr>
<td>SACFd</td>
<td>17 (udp)</td>
<td>00:01:43</td>
<td>0 bps/0 bps</td>
<td>262.7 KB/1607.1 KB</td>
<td>4 260/2 618</td>
<td>262.3 KB/1607.1 KB</td>
<td>4 258/2 617</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:05:02</td>
<td>0 bps/0 bps</td>
<td>518.4 KB/188.6 KB</td>
<td>4 254/1 632</td>
<td>517.8 KB/187.8 KB</td>
<td>4 248/1 622</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:05:02</td>
<td>0 bps/0 bps</td>
<td>3.1 kbps/39.5 kbps</td>
<td>3 977/5 265</td>
<td>1066.3 KB/3245.0 KB</td>
<td>3 975/5 264</td>
</tr>
<tr>
<td>SACFd</td>
<td>6 (tcp)</td>
<td>00:00:00</td>
<td>time wait</td>
<td>0 bps/0 bps</td>
<td>232.7 KB/2113.2 KB</td>
<td>232.5 KB/2113.1 KB</td>
<td>232.5 KB/2113.1 KB</td>
</tr>
<tr>
<td>SACFd</td>
<td>17 (udp)</td>
<td>00:02:15</td>
<td>0 bps/0 bps</td>
<td>121.9 KB/1922.1 KB</td>
<td>212.7 KB/1921.8 KB</td>
<td>212.7 KB/1921.8 KB</td>
<td>212.7 KB/1921.8 KB</td>
</tr>
<tr>
<td>SACFs</td>
<td>6 (tcp)</td>
<td>1d 23:59:02</td>
<td>6.6 kbps/38.0 kbps</td>
<td>217.6 KB/1869.3 KB</td>
<td>3 103/4 144</td>
<td>217.5 KB/1869.3 KB</td>
<td>3 101/4 143</td>
</tr>
<tr>
<td>SACFs</td>
<td>6 (tcp)</td>
<td>1d 23:59:03</td>
<td>37.0 kbps/3.4 kbps</td>
<td>1093.6 KB/75.3 KB</td>
<td>2 614/1 111</td>
<td>1093.5 KB/75.2 KB</td>
<td>2 611/1 110</td>
</tr>
<tr>
<td>SACFd</td>
<td>17 (udp)</td>
<td>00:00:48</td>
<td>0 bps/0 bps</td>
<td>162.5 KB/1670.8 KB</td>
<td>2 483/2 732</td>
<td>162.0 KB/1670.7 KB</td>
<td>2 480/2 730</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:05:00</td>
<td>2.3 kbps/45.6 kbps</td>
<td>153.6 KB/1617.9 KB</td>
<td>2 436/2 701</td>
<td>153.3 KB/1617.8 KB</td>
<td>2 434/2 700</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:05:02</td>
<td>992 bps/32.9 kbps</td>
<td>220.2 KB/1548.0 KB</td>
<td>2 217/2 623</td>
<td>221.7 KB/1547.9 KB</td>
<td>2 213/2 620</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:03:13</td>
<td>0 bps/0 bps</td>
<td>134.3 KB/1451.4 KB</td>
<td>134.0 KB/1451.3 KB</td>
<td>134.0 KB/1451.3 KB</td>
<td>134.0 KB/1451.3 KB</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:00:31</td>
<td>0 bps/0 bps</td>
<td>119.3 KB/1259.9 KB</td>
<td>138.2 KB/1259.8 KB</td>
<td>138.2 KB/1259.8 KB</td>
<td>138.2 KB/1259.8 KB</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:05:01</td>
<td>3.2 kbps/39.5 kbps</td>
<td>121.1 KB/1547.2 KB</td>
<td>1872/2 379</td>
<td>120.6 KB/1547.2 KB</td>
<td>1872/2 378</td>
</tr>
<tr>
<td>SACFs</td>
<td>17 (udp)</td>
<td>00:01:01</td>
<td>1984 bps/34.3 kbps</td>
<td>119.3 KB/1259.9 KB</td>
<td>138.2 KB/1259.8 KB</td>
<td>138.2 KB/1259.8 KB</td>
<td>138.2 KB/1259.8 KB</td>
</tr>
<tr>
<td>SACFs</td>
<td>6 (tcp)</td>
<td>1d 23:59:02</td>
<td>34.0 kbps/4.2 kbps</td>
<td>1156.8 KB/108.4 KB</td>
<td>1824/2 777</td>
<td>1156.8 KB/108.4 KB</td>
<td>1822/2 776</td>
</tr>
<tr>
<td>SACFs</td>
<td>6 (tcp)</td>
<td>00:00:00</td>
<td>time wait</td>
<td>0 bps/0 bps</td>
<td>113.1 KB/1859.6 KB</td>
<td>112.9 KB/1859.5 KB</td>
<td>1810/2 086</td>
</tr>
</tbody>
</table>

991 items out of 978 (1 selected)  Max Entries: 218032
Fasttrack-connection

- “fasttrack-connection” action works similar to “mark-connection” action
- “fasttrack-connection” rule is usually followed by identical “accept” rule
- Most common Fasttrack implementations:
  - Fasttrack if connection reach connection-state=established and related
  - Fasttrack to exclude some specific connections from the queues
  - Fasttrack all local connections
Without Fasttrack

- Board: RB2011UiAS-2HnD
- Configuration: default Home AP
- Single TCP connection throughput: 358Mbps
- CPU load: 100%
- Firewall CPU load: 44%
With Fasttrack

- Board: RB2011UiAS-2HnD
- Configuration: default Home AP
- Single TCP connection throughput: 890Mbps
- CPU load: 86%
- Firewall CPU load: 6%
Full queues and multi-core processing

• Packets are spending most part of the processing time waiting in full queues

• In order not to waste CPU core cycles on waiting, current core will just leave the packets in the queue and take already processed packets out of the same queue

• Queued packets can be taken out of the queue randomly by the CPU core, that works on that queue at the time

• In short: full queues can shuffle packet assignments to CPU cores
Empty queues and multi-core processing

- In order not to waste CPU core cycles on waiting, current core will just leave packets in the queue and take already processed packets out of the same queue.
- In the case when the queue limit is not reached, same packets will be left in and taken out of the queue by the same CPU core, making this process seamless.
Conclusions

- Queues don’t slow down single TCP streams if they are not actually queuing packets (limits are not reached)
- The complexity of the configuration has a minimal impact on a single TCP stream (with the exception of deep packet inspection and queues), if the CPU core/thread doesn’t reach 100%
- Use configuration features like fastpath, fasttrack, ip firewall raw, etc. to reduce overall CPU load
Testing from router to router

- Traffic generation/elimination takes at least the same amount of CPU resources as simple traffic forwarding.
- The router needs to do both generate traffic and then forward it to destination.
- By default the traffic forwarding process has the highest priority in routers (the reason why ping through the router is better than to the router).
Traffic-generator tool

- Traffic Generator can:
  - Determine transfer rates, packet loss
  - Detect out-of-order packets
  - Collect latency and jitter values
  - Inject and replay *.pcap file
- “Quick” mode
- Full Winbox support (coming soon)
- Doesn’t have TCP protocol support
- Scares people
Bandwidth-test tool

- People love it!!!
- Until v6.44 was single threaded, now both UDP and TCP tests support multi-threading
- In v6.44 we added warning message when CPU load exceeds 90% (CLI only), to inform that CPU is bottlenecking results, not the link
- Can do multistream tests (but how many should you do?)
Speed-test tool

- Introduced in RouterOS v6.44 (CLI only)
- It is a simple test tool for measuring ping, jitter, TCP and UDP throughput from one MikroTik device, to another
- It is based on bandwidth-test and ping tool, to use it – the bandwidth-test server needs to be accessible
- It automatically determines optimal number of test streams based on the CPU core/thread count on both devices
Speed-test tool

```
[admin@MikroTik]] > /tool speed-test address=192.168.88.1
  status: done
  time-remaining: 0s
  ping-min-avg-max: 541us / 609us / 3.35ms
  jitter-min-avg-max: 0s / 76us / 2.76ms
  loss: 0% (0/100)
  tcp-download: 921Mbps local-cpu-load:30%
  tcp-upload: 920Mbps local-cpu-load:30% remote-cpu-load:25%
  udp-download: 917Mbps local-cpu-load:6% remote-cpu-load:21%
  udp-upload: 916Mbps local-cpu-load:20% remote-cpu-load:6%

[admin@MikroTik]] > /tool speed-test address=192.168.88.1
  ;; results can be limited by cpu, note that traffic generation/termination
  ;; performance might not be representative of forwarding performance
  status: done
  time-remaining: 0s
  ping-min-avg-max: 541us / 609us / 3.35ms
  jitter-min-avg-max: 0s / 76us / 2.76ms
  loss: 0% (0/100)
  tcp-download: 721Mbps local-cpu-load:78%
  tcp-upload: 829Mbps local-cpu-load:100% remote-cpu-load:84%
  udp-download: 905Mbps local-cpu-load:10% remote-cpu-load:54%
  udp-upload: 895Mbps local-cpu-load:55% remote-cpu-load:12%
```
Questions!!!