Load Balancing Rules/Policies examples

□

Contents

Introduction

The information on this page is updated in accordance with firmware version **00.07.02.7**.

In this example, we will examine Load Balancing rules and policies and we will set up rules where all traffic from one LAN port will be going through the Mobile interface and all the traffic from the second port will be going through the Wireless WAN/Wired WAN or the Second Sim card (if used with RUTX12).

Load Balancing is a method of dividing traffic between multiple WAN interfaces. Load Balancing can be used to share the data load between different interfaces and increase the overall Internet speed for multiple users and connections. It is important to note that <u>Load Balancing does not increase speed for any single connection</u>. For example, if you're downloading a large file, using Load Balancing will not increase the speed of that download.

However, Load Balancing can be used to increase the speed of multiple connections. For example, when used in a WiFi network, Load Balancing would utilize the resources of multiple WAN interfaces to handle the many various requests of different clients.

Both Load Balancing and Failover cannot be used at the same time. If you wish to select Load Balancing, you can do that in the dropdown located in the top-right corner of the page:

FAILOVER / LOAD BALANCING INTERFACES

NAME	TYPE	INTERVAL	STATUS		
wan	Wired	3	-		off on

If you're having trouble finding this page or some of the parameters described here on your device's WebUI, you should **turn on "Advanced WebUI" mode**. You can do that by clicking the "Basic" button under "Mode", which is located at the top-right corner of the WebUI.

	MODE	USER	FW VERSION	
≪TELTONIKA ∣ Networks	BASIC	ADMIN	RUTX_R_00.07.00	LOGOUT ⊟

Prerequisites

- RUTX12 device or any other RUT device
- 2 SIM cards or other WAN alternatives (WiFi WAN or regular LAN cable connected to WAN port)
- At least one end device (PC, Laptop, Tablet, Smartphone) to configure the routers
- At least one VLAN configured on a device

Preparation

As mentioned in Prerequisites, you will need to configure a VLAN on the RUT device, detailed instructions on how to configure it can be found on this page: **VLAN Set Up** Everything else we will configure along the way.

Topology

Configurating RUTX12 device

We will start with the RUTX12 configuration. We will configure **RUTX12** traffic from **LAN** port 1 to go through **SIM1** and traffic from **LAN** port 2 to go through **SIM2**. To complete this configuration you will require to have **one VLAN configured already**. Let's begin by navigating to your device's **WebUI → Network → Failover**. On this page you will need to configure two new rules, to do so at the bottom you can find the **ADD NEW INSTANCE** section, where input the name for a rule and press the button **ADD**.

In the topology, you can see that the second **LAN** port has an IP of **192.168.3.0/24**, we will configure the first rule accordingly for this port.

LAN2 configuration

- **Source address** - input your **VLAN** network address in this example **192.168.3.0/24**
- **Destination address** - leave **0.0.0.0/0** here, since we want to use this rule when navigating **anywhere** from **LAN2** port
- **Policy assigned** - pick the policy on which you have set up for **mob1s2a1**
- **Policy** - here we will need to create two Policies (both of them for Load Balancing), like in the example create one rule for **mob1s1a1** and one for **mob1s2a1**. Once they are created assign **mob1s2a1** to this **Rule**.

After setting everything up Save the configuration.

LAN1 configuration

- **Source address** - input your **LAN** network address in this example **192.168.1.0/24**
- **Destination address** - leave **0.0.0.0/0** here, since we want to use this rule when navigating **anywhere** from **LAN1** port
- **Policy assigned** - pick the policy on which you have set up for **mob1s1a1**

Here we will not be creating new Policies since we already did that on the **LAN2** configuration.

After setting everything up Save the configuration. And after that **Enable** the Load Balancing for **mob1s1a1** and **mob1s2a1** interfaces. For RUTX12 the configuration is finished, if you are not planning to configure another device (with one modem) skip to the **Testing the Configuration** section.

Configurating Device with one modem

In this section, we will configure a device unlike **RUTX12** with one modem. We will configure all of the traffic from the **LAN1** port to go through **SIM1** and all of the traffic from the **LAN2** port will go through WiFi WAN (you can use regular Wired WAN instead). Once again we will start the configuration by **configuring LAN2** port traffic, keep in mind that the **LAN2** port is configurated as **VLAN 192.168.3.0/24**.

LAN2 configuration

- **Source address** - input your **VLAN** network address in this example **192.168.3.0/24**
- **Destination address** - leave **0.0.0.0/0** here, since we want to use this rule when navigating **anywhere** from **LAN2** port
- **Policy assigned** - pick the policy on which you have set up for **Wireless WAN** or **Wired WAN**
- **Policy** - here we will need to create two Policies (both of them for Load Balancing), like in the example create one rule for **mob1s1a1** and one for **Wireless WAN** or **Wired WAN**. Once they are created assign **Wired WAN** to this **Rule**.

After setting everything up Save the configuration.

LAN1 configuration

- **Source address** - input your **LAN** network address in this example **192.168.1.0/24**
- **Destination address** - leave **0.0.0.0/0** here, since we want to use this rule when navigating **anywhere** from **LAN1** port
- **Policy assigned** - pick the policy on which you have set up for **mob1s1a1**

Here we will not be creating new Policies since we already did that on the **LAN2** configuration.

After setting everything up Save the configuration. And after that **Enable** the Load Balancing for **mob1s1a1** and **Wireless WAN/Wired WAN** interfaces.

Testing Configuration

In order to test if the configuration if it's working we will use TCP Dump, TCP Dump will let us capture traffic that goes through each interface of the device.

Installing TCP Dump on RUT

Firstly, you will need to connect to the **RUT** device's **CLI/SSH**. You can connect the **RUT** device CLI via WebUI by navigating to **Services → CLI**. Log in to **CLI** with the user name **root** and the router's admin password.

Now execute these commands one at a time:

opkg update

opkg install tcpdump

To test if TCP Dump has been installed on your device execute the command *tcpdump*

Testing the configuration via TCPDUMP

The simplest way to test the configuration via **tcpdump** would be to use two commands from the same device. The first one would be ping to google DNS (from 192.168.1.1 and 192.168.3.1 IPs) and the other would be to **tcpdump** the correct interface. I will be using the setup of **Configurating RUTX12 device** for this example, testing devices with one modem is almost identical.

We will do this by opening two instances of your device's **CLI/SSH**, to access it, login to one of the routers' WebUI and navigate to **Services → CLI**. Log in to CLI with the user name **root** and the router's admin password (Open two instances of CLI).

In the first instance, we will let the ping command which will send ICMP requests to **8.8.8.8** from **192.168.1.1(LAN IP)**. The command for this will be used: ***ping -I 192.168.1.1 8.8.8.8***

On the second instance, we will run TCPDUMP on interface **wwan0** to capture if packets are coming through **Mobile WAN**. The command for this will be used: ***tcpdump -i wwan0 icmp -n*** We will also only look at ICMP requests since the Mobile will have a lot of different packets coming in and out.

Output of Ping command

```
root@Teltonika-RUTX11:~# ping -I 192.168.1.1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 192.168.1.1: 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=113 time=102.315 ms
64 bytes from 8.8.8.8: seq=1 ttl=113 time=102.522 ms
64 bytes from 8.8.8.8: seq=2 ttl=113 time=118.127 ms
64 bytes from 8.8.8.8: seq=3 ttl=113 time=101.033 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 101.033/105.999/118.127 ms
root@Teltonika-RUTX11:~#
```

Output of TCPDUMP command

```
root@Teltonika-RUTX11:~# tcpdump -i wwan0 icmp -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wwan0, link-type RAW (Raw IP), capture size 262144 bytes
11:13:15.544890 IP 213          > 1.1.1.1: ICMP echo request, id 30183, seq 0, length 64
11:13:15.570254 IP 213          > 8.8.8.8: ICMP echo request, id 29777, seq 82, length 64
```

You should see in the **tcpdump** output that we have ICMP requests going out of the **WWAN0** interface, this means that the configuration is working, and **192.168.1.1/24** interface traffic goes through the correct interface, you should also test this on another SIM card(interface). You can also stop the ping on your device and you should see that **ICMP** requests stop appearing in the **tcpdump**