Automatically Switching a Tunnel to the Primary Interface

When two network connections (primary and backup) are used, a tunnel automatically switches to the backup interface if the primary connection fails.

Once the primary connection is restored, however, the tunnel continues to use the backup interface because the existing tunnel session remains active and is not automatically re-established.

On Donyx routers, this behavior can be addressed by using the **Pinger** service together with the **Events** mechanism to re-establish the tunnel when the primary interface becomes reachable again.

The configuration can be performed either through the dnxOS web interface or the CLI. For clarity, this article uses the CLI.

Example

The following example uses this network configuration:

- *port3* — primary wired interface;
- *wifi1* — backup wireless interface.

Under normal conditions, the tunnel should operate through *port3*.

If connectivity through *port3* is lost, the tunnel should automatically switch to *wifi1*.

Once *port3* becomes available again, the tunnel should automatically switch back to the primary interface.

Step 1. Configure interface priorities

Assign different metric values to the interfaces. The interface with the lower metric has the higher priority.

```
/ip interface port3 metric 203
/ip interface wifi1 metric 206
```

Step 2. Create a connectivity check

Create a **Pinger** service to monitor the availability of the remote L2TP server through the primary interface.

```
/service pinger add interface=port3
host <l2tp_server>
apply
```

Step 3. Create an event

Create an event that is triggered when the primary interface becomes reachable again.

When the **Pinger** service changes to the *success* state, the L2TP tunnel is re-established.

```
/event add trigger=pinger
target port3
value success
action /tunnel l2tp <tunnel_name> apply
apply
```

After the tunnel is re-established, routing is recalculated based on the configured interface metrics. Since *port3* has the lower metric, the tunnel is established through the primary wired interface.

Result

After completing the configuration, the tunnel operates as follows:

1. The tunnel operates through *port3*.
2. If *port3* fails, the tunnel automatically switches to *wifi1*.
3. When *port3* becomes reachable again, the **Pinger** service detects the status change.
4. The event re-establishes the tunnel.
5. The new tunnel is established through *port3* because it has the higher priority.

Recommendations

The event is triggered only once when the **Pinger** service changes to the *success* state.

To avoid false or premature triggers, configure the **Pinger** polling interval and retry count so that its state changes only after the tunnel has been successfully re-established through the backup interface. This helps prevent unnecessary tunnel re-establishment during brief network interruptions.

All modifications are permanently saved to the router configuration only after executing the */system config commit* command or clicking the **commit** button in the web interface.