Mikrotik TTT

Selective Failover & Intentional Service Segmentation
Using MikroTik RouterOS v7 V2

Instructor: Aaron Gustafson

Course Level: MTCRE – MikroTik Certified Routing Engineer
Session Length: 1 Hour (Lecture + Lab)

Intro

How many have implemented Dual-Wan with Automatic Fail-over?

(Raise your hand)

What we will cover

- Real-world need for selective multi-WAN failover
- RouterOS v7 policy routing concepts (Mangle Vs Routing Rules)
- Building routing tables and routing rules
- Creating business-critical vs. non-critical VLAN paths
- Implementing selective failover logic
- Hands-on lab: full configuration + failover test
- Validation & troubleshooting techniques

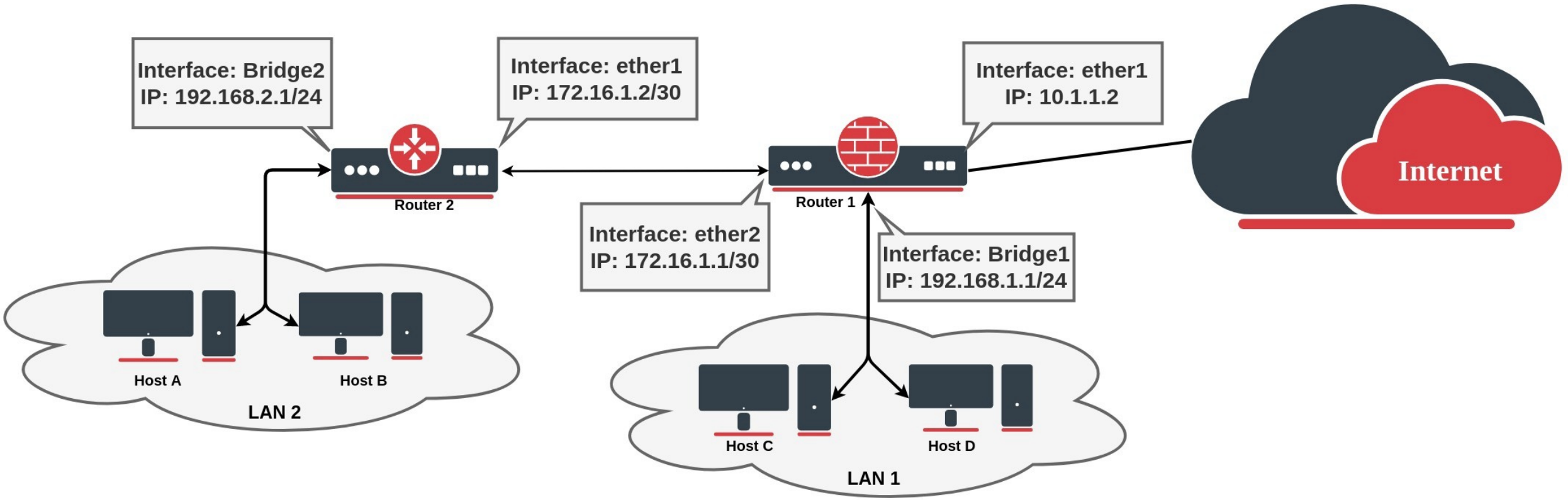
IP Routing

From MikroTik Wiki and the MTCNA course – a brief review

Overview

Routing is the process of selecting paths across the networks to move packets from one host to another.

IP Routing


Interface: Bridge2
IP: 192.168.2.1/24
Interface: ether1
IP: 172.16.1.2/30
Interface: ether1
IP: 10.1.1.2
Internet
Router 2
Router 1
Interface: ether2
IP: 172.16.1.1/30
Interface: Bridge1
IP: 192.168.1.1/24
Host A
Host B
LAN 2
Host C
Host D
LAN 1

FIB and RIB

RouterOS routing information consists of two main parts:

FIB (Forwarding Information Base), is used to make packet forwarding decisions. It contains a copy of the necessary routing information.
RIB (Routing Information Base) contains all learned prefixes from routing protocols (connected, static, BGP, RIP, OSPF).

RIB

Routing Information Base is a database that lists entries for particular network destinations and their *gateways* (address of the next device along the path or simply *next-hop*). One such entry in the routing table is called a *route*.

A *hop* occurs when a packet is passed from one network segment to another. By default, all routes are organized in one "main" routing table. It is possible to make more than one routing table which we will discuss further in this module.

FIB

FIB uses the following information from the packet to determine its destination:

- source address
- destination address
- source interface
- routing mark

Possible routing decisions are:

- receive packet locally
- discard the packet (either silently or by sending an ICMP message to the sender of the packet)
- send the packet to a specific IP address on a specific interface

Why Selective Failover Matters

- Businesses (At least in the US) often have **two WAN connections**
  - ISP1: Primary (Fiber/Coax)
  - ISP2: Backup (LTE)
- Not all traffic should fail over
  - **Business-critical devices** must stay online
  - **Guest Wi-Fi / non-essential services** can be dropped
- Preserves **limited LTE bandwidth** for priority operations
- Used widely in retail and hospitality
  - Example: **Chick-fil-A (Quick Service Restaurant)** POS networks fail over; guest Wi-Fi does not to allow Card Processing
  - Can sometimes save money when dealing with per-usage billing for backup circuits

LAB Topology

ISP 1 - Fiber

ISP 2 - LTE

Eth1

Eth 2

Eth3
VLAN_10
Office WiFi
Failover

Eth4
VLAN_20
Office Wired
Failover

Eth5
VLAN_99
Guest WiFi
No Failover

Policy Goals

Office WiFi (VLAN10) → Use ISP1 → Fail over to ISP2

Office Wired (VLAN20) → Use ISP1 → Fail over to ISP2

Guest WiFi (VLAN99) → Use ISP1 → *Do not* fail over
(Intentionally loses Internet during outage)

Table Name	Purpose
to_isp1	Primary Fiber (ISP1) + Failover to ISP2
guest_isp1	Guest-only → ISP1 only (no fallback)

Initialization

Go to TTT.ByteMeNetworks.com and open both Base Config and Solutions

Factory Reset your router with no-default config and winbox into your router

Copy and paste the Base Config Text into CLI

Configuration

1. Create Routing Tables

- `to_isp1`
- `guest_isp1`

/routing table
add name=to_isp1 fib
add name=guest_isp1 fib

Configuration

2. Add Default Routes per Table

- ISP1 primary (distance 1)
- ISP2 backup (distance 2)
- Guest table ONLY includes ISP1

```
/ip route
add dst-address=0.0.0.0/0 gateway=192.168.1.1 routing-table=to_isp1 distance=1 check-gateway=ping
add dst-address=0.0.0.0/0 gateway=192.168.2.1 routing-table=to_isp1 distance=2 check-gateway=ping
add dst-address=0.0.0.0/0 gateway=192.168.1.1 routing-table=guest_isp1 distance=1 check-gateway=ping
```

Configuration

3. Add Routing Rules

- VLAN10 → `to_isp1`
- VLAN20 → `to_isp1`
- VLAN99 → `guest_isp1` (lookup-only-in-table)

/routing rule
add src-address=192.168.10.0/24 action=lookup-only-in-table table=to_isp1
add src-address=192.168.20.0/24 action=lookup-only-in-table table=to_isp1
add src-address=192.168.99.0/24 action=lookup-only-in-table table=guest_isp1

Configuration

4. Test ISP 1 functional

- Ping from each VLAN
- Simulate ISP1 failure
- Observe selective failover behavior

- From Ether 3 Ping 172.26.2.1
- From Ether 4 Ping 172.26.2.1
- From Ether 5 Ping 172.26.2.1

Configuration

5. Test ISP 1 failure

- From Ether 3 Ping 172.26.2.1 – should work
- From Ether 4 Ping 172.26.2.1 – should work
- From Ether 5 Ping 172.26.2.1 – SHOULD NOT WORK

Testing

1. Test Normal Operation (ISP1 Up)
From all VLANs, test:
ping 172.26.73.1
Expected:
✓ All VLANs reach **172.26.73.1** via ISP1
✓ Guest VLAN works normally
✓ No traffic uses 192.168.2.1 yet
Optional traceroute (if needed):
traceroute 172.26.73.1.

Testing

2. Simulate ISP1 Failure
Instructor disables primary uplink:
/interface disable VLAN_30

Students retest:
Business VLANs (10 & 20):
ping 172.26.73.1
Expected:
✓ Should reach **172.26.73.1**(failover to ISP2)

Guest VLAN (99):
ping 172.26.73.1
Expected:
✗ Should **not** get a reply
✗ Should **not** have a route to ISP2

Summary

What You Learned Today
How RouterOS v7 implements **Policy-Based Routing (PBR)**
Creating and using **multiple routing tables**
Binding VLANs to routing tables with **routing rules**
Using **lookup-only-in-table** to enforce traffic isolation
Implementing **selective failover** between two WANs
Validating behavior by pinging 172.26.73.1

Real-World Applications
Retail (POS continuity, guest WiFi isolation)
Branch offices with LTE or satellite backup
MSP customer segmentation
Bandwidth conservation on metered backup links

Summary cont.

Closing Notes
Full configuration is reproducible on any MikroTik device with RouterOS v7+
This design scales easily with more VLANs or more WANs
You now have a template for real-world selective failover deployments

Tip: You can use CNAMEs for subdomains to make Mikrotik's DDNS look more Branded: Instead of Serial.mynetname.com customer.ByteMeNetworks.com

Handy tool when doing VPNs and Camera Remote access to keep your company Name at the forefront of customer mind