

# **Application notes**







# Table of Contents

Babel dynamic routing protocol	5	
Babel description		
Selected Babel parameters		
Brief introduction to examples	8	
1. Mesh topology	9	
1.1. Description	9	
1.2. RipEX_A configuration	10	
1.3. RipEX_B, RipEX_C and RipEX_D configuration	. 15	
1.4. Diagnostics and Testing	16	
1.4.1. Routing	16	
1.4.2. Tools	18	
2. Mesh topology with Radio and Relay filters	20	
2.1. Radio filters	20	
2.2. Relay filters	25	
3. Two repeaters on the RF channel	29	
3.1. Description and Configuration	29	
3.2. Diagnostics and Testing	31	
3.2.1. Tools	33	
4. Radio channel and Ethernet combination	36	
4.1. Description	36	
4.2. RipEX_A Configuration	37	
4.3. RipEX_B Configuration	39	
4.4. Diagnostics and Testing	41	
4.4.1. Tools and Monitoring	42	
5. Radio channel and Cellular (LTE) combination	45	
5.1. Description	45	
5.2. RipEX_B Configuration	46	
5.3. RipEX_D Configuration	49	
5.4. Diagnostics and Testing	51	
6. Basic Babel and OSPF combination	55	
6.1. Description	55	
6.2. RipEX_A Configuration	56	
6.3. RipEX_D Configuration	61	
6.4. Diagnostics and Testing	66	
7. Advanced Babel and OSPF combination	. 72	
7.1. Description	72	
7.2. RipEX_A Configuration	73	
7.3. RipEX_C Configuration	79	
7.4. RipEX_D Configuration	85	
7.5. Diagnostics and Testing	89	
7.5.1. Checking Routing tables	. 90	
7.5.2. Tools – ICMP ping and Routing	91	
7.5.3. Testing Ethernet failures	93	
7.5.4. Testing Radio failures	94	
8. Hints and Tips	96	
8.1. Throughput, speed	96	
8.2. Static path setup	96	
8.3. Advanced configurations	96	
Revision History	97	

# Babel dynamic routing protocol

This application note will guide you through several **Babel examples** and its typical usage. Examples are based on four RipEX2 units only – feel free to accommodate scenarios to your RipEX2 network.

The application note explains various Babel parameters, but it does not consist all of them. Check the RipEX2 manual or help for further information. On the other hand, this application note can provide a valuable insight into important details or more practical explanation than strict information from the manual.

# **Babel description**

Babel is a dynamic routing protocol working on a link layer. Implementation in RipEX2 (and bird daemon) uses **UDP/IPv6 multicast packets on port 6696**.

Babel is an Interior Gateway Protocol (**IGP**) and works within one autonomous system. Routers with Babel enabled must be directly adjacent to each other (on the same link). The protocol can be configured on multiple interfaces (ETH, radio, ...) of each RipEX2 unit and it seeks for connections on every one of them.

Babel works on both, **wired** and **wireless** connections. In wireless networks, it uses efficient multicast transmission, does not require all participants to "hear" each other and uses a finer link quality evaluation instead of binary decision (good/bad).

Individual protocol instances within the network send and receive Hello packets which map interconnections and their qualities. They also exchange routing update messages regularly.

The protocol tries to find the most convenient path according to the metric and not creating loops. Bellman-Ford algorithm is being used. **The metric** is set on the interface and represents a price for receiving packets. This is an abstract value and can be set to any value (e.g., it can be derived from the particular link speed – Ethernet, LTE, Radio link, …). The lower the value, the more advantageous the link/interface is. The maximum value of the metric is 65535 (infinity) and must be greater than 0.

### Interface types:

- **Wired** assumes a reliable link. The quality is evaluated according to the number of received Hello packets. If more packets are lost than the configured limit, the line is considered down.
- Wireless assumes a variable connection quality. A quality is therefore evaluated by a variant of the ETX (Expected Transmission Count) metric. If Hello packets are lost, the price of the interface gradually increases until the line is declared down.

Bird implementation of Babel does not support any security.

If the protocol is used in the Hot Standby (HS), keep in mind it is completely turned off in the passive unit. Once the passive unit becomes the active one, it must first find out all the routes from the start which may take a while (based on settings). Babel can use the virtual shared HS IP address.

# **Selected Babel parameters**

### Common – Router offering

- Default: "On"
- It turns on/off propagation of routing rules obtained from its own neighbor. If it is turned off, the station behaves as end terminal (Babel paths start and end here, they are not forwarded through).

### Network – Interface

- You must manually set the interface name for each interface you want Babel to be active on. A list of possible interfaces is explained in manual, or see below.
  - LAN "if\_" + interface name defined in GUI (e.g., "if\_bridge")
  - VLAN "if\_" + interface name defined in GUI + '.' dot and VLAN number (e.g., "if\_bridge.29")
  - Radio "radio"
  - Hot Standby "hstdby"
  - $\circ~$  GRE L3 "gre\_tunX" where 'X' is the tunnel number, starting from zero
  - Cellular "aux"

### Network – Hello limit

- Used only for "Wired" interface.
- A limit of received Hello packets for which the link is considered down (from expected 16).

### Network - Advertised next hop

- It is used if multiple IP addresses are set on one interface. Otherwise, use default 0.0.0.0.
- Select which IP address should be used as a "next-hop" IP for our neighbor routing tables.

# **Brief introduction to examples**

In general, Babel is easy to setup. Once you have your RipEX2 units configured by themselves (Radio, ETH IP addresses, Radio protocols, SCADA setup, ...), you just need to:

- · Enable Babel and set its unique Router ID
- Choose RipEX2 interfaces on which Babel should operate
- Select Babel timing (speed of routes propagation, ...)
- Local LANs to be propagated

If you only have two or three RipEX2 units, you can easily accommodate most of the examples as well. You can configure just part of the example, the RipEX2 configuration is mainly the same.

- *Example 1* The first example depicts a network only with RF channel usage and shows "only" dynamically obtained direct (one radio hop) LAN routing.
- *Example 2* The second example shows and explains Radio filters to optimize Babel dynamic routing based on current RF channel conditions and Relay filters to optimize how and where are the routes propagated.
- *Example 3* The third example changes the RF channel setup to simulate using two repeaters for end-to-end communication. The network is not a mesh compared to the first example.
- *Example 4* The fourth example utilizes Ethernet connection as a quick backbone and radio channel only as a backup option. Feel free to utilize just two RipEX2 units primary connection is via Ethernet; backup is via the radio channel.
- *Example 5* The fifth example changes the above Ethernet by LTE and GRE connection.
- Example 6 The sixth example shows a network combining Babel and OSPF. OSPF is configured in RipEX\_A and RipEX\_D units, whereas Babel is configured in RipEX\_A, B and C. These two protocols are neighboring in one Autonomous System Border Router "ASBR" (RipEX\_A) which divides the whole network into two parts. Bandwidth optimized Babel on the Radio segment and standardized, widely-used and fast OSPF on the Ethernet segment.
- Example 7 The seventh example extends the sixth one to show even more from OSPF and Babel combination. There are two ASBR routers between Babel and OSPF network segments. By default, the primary path should go over RipEX\_A and a backup path is over RipEX\_C. Dynamic protocols exchange routing rules and preferences between each other.

## 1. Mesh topology



Fig. 1.1: Example 1 – Mesh topology

## 1.1. Description

The first example shows how Babel can be used to propagate RipEX2's LANs automatically instead of static routes. The topology allows every device to directly see each other resulting in a situation that all communication goes directly between two particular RipEX2 units. There is no dedicated repeater or another technology such as LTE or Fibre optics.

## **1.2. RipEX\_A configuration**

Start with unit's Name and Mode. Go to the SETTINGS – Device – Unit menu. Select the "Router" mode, because it is not possible to configure dynamic routing protocols in the Bridge mode. Set the name to "RipEX\_A".

RipEX2 RipEX_A @10.9.8.	7 I <sup>3)</sup> 'I Remote access
Unit time: 2021-07-16 13:40:15 (UTC+2)	General Service USB Time Hot standby
	Operating mode
O STATUS	Mode Router 🖌
SETTINGS	Linit
Interfaces	Unit name RipEX. A
Routing	Unit note
Firewall	Unit location
VPN	Unit contact
Security	<ol> <li>All information above is used in SNMP device info.</li> </ol>
Device	
Unit	

Fig. 1.2: RipEX\_A – Mode and name

You can also set a correct time in the unit, either via a working NTP server or just manually update it by the time in your browser ("Update in device" button and a check-box "Use browser time"). This is useful in general and mainly for debugging purposes (Statistics, monitoring, ...). It is not required by Babel protocol.

RipEX2 RipEX_A @10.9.8.7	1 <sup>9</sup> 1 Remote access	UNIT
Unit time: 2021-07-16 13:39:20 (UTC+2)	General Service USB Time Hot standby	
STATUS	Status NTP state not synced	
🗘 🛱 settings	Stratum 8 Delay [ms] 0.000	
Interfaces	Dispersion [ms] 0.000	
Routing		
Firewall	Time	
VPN	Change device time manually 2021-07-16 13:39:22 Update in o	device Vse browser time
Security	NTP client synchronization source NTP server	
Device	NTP server minimum polling time 1 min.	
Unit	Time zone Europe/Prague 🗸	
Configuration		
Events	NTP servers	
SNMP	• Table does not contain any data.	
Software keys		
Firmware	+ Add NTP server	

#### Fig. 1.3: RipEX\_A – Time

Configure a correct Ethernet IP address 192.168.1.1/24 (bridged on all ETH ports).

Rip <b>EX2</b> RipEX_A @10.9.8.7	Remote access
Unit time: 2021-07-16 13:43:09 (UTC+2)	ETH1 • ETH2 • ETH3 • ETH4 • ETH5 •
	Status
O STATUS	
SETTINGS	ETH1: • -   Attached   -   Router mode
Interfaces	Network interface
Ethernet	Name bridge bridges: ETH 1, ETH 2, ETH 3, ETH 4
Radio	
COM	Allow unit management
Terminal servers	
Routing	LAN
Firewall	IP / Mask 192.168.1.1/24 Note local
VPN	+ Add IP/Subnet

Fig. 1.4: RipEX\_A – Ethernet IP address

Go to the Radio submenu and configure the Radio interface. You can accommodate most of the parameters to suit your needs, but be consistent throughout this application note and all its examples.

RipEX2 RipEX_A 11/1 Remote	access	RADIO		E Changes	M Nc
Unit time: 2021-07-16 13:49:08 (UTC+2)	Status				
	Radio protocol		Radio parameters		
O STATUS	Radio protocol	Flexible 🗸	TX frequency [Hz]	415500000	$\hat{\mathbf{v}}$
SETTINGS	IP / Mask	10.10.1/24	RX frequency [Hz]	415500000	÷
Interfaces	ACK	On 🗸	Antenna configuration	Single (Tx/Rx)	*
Ethernet	Retries [No]	3	RF power PEP [dBm]	20	*
Radio	Foreign packets RSS threshold	120 🗘	Channel spacing [kHz]	25	*
СОМ	[-dBm]	0."	Occupied bandwidth limit [kHz]	25	*
Terminal servers	Repeat COM broadcast	011	Modulation type	QAM	*
Routing	Management		Modulation	16DEQAM	*
	Allow unit management On	~	FEC	Off	*
VPN	Encryption				
Security	Encryption Off 🗸 🗸				

Fig. 1.5: RipEX\_A – Radio interface settings

Settings:

Radio protocol	Flexible
IP / Mask	10.10.1/24
Radio protocol	Flexible
ACK	On
Retries	3
TX/RX frequencies	415.500.000 MHz
Antenna configuration	Single (Tx/Rx)
RF power PEP	20 dBm (testing on the desk, lowest power)
Channel spacing and OBW	25 kHz
Modulation and its type	QAM, 16DEQAM
FEC	Off

# **(i)**

## Note

Using the Flexible Radio protocol via a simplex channel (half-duplex operation) might result in a FULL MESH organized network. When BDP Radio protocol is set, using the Babel dynamic routing results in a self-configuring network with a STAR topology. There is no Encryption set, neither any Individual link options within these examples. The most important menu is SETTINGS – Routing – Babel. Activate the protocol and set the Router ID to unique 1.1.1.1 and enable Routing offering so that it forwards received routes to other neighbor.

RipEX2 RipEX_A 1 <sup>1</sup> /1 Remote a	access		BABEL	
^				
Unit time: 2021-07-16 13:53:59 (UTC+2)	Common Network	Static rules	Import filter	Export filter
	Common settings			
STATUS	Router ID 1.1.1.1			
🗞 SETTINGS	Routing offering On	~		
Interfaces				
Routing				
Static				
BABEL				

#### Fig. 1.6: RipEX\_A – Babel common settings

Go to the Network panel and create a new Wireless network.

Edit interface	×			
(	Active			
Interface	radio			
Туре	Wireless 🗸			
Rx cost	100 ≎			
Hello interval [s]	30 ≎			
Update interval multiplier	4 ≎			
Advertised next hop	0.0.0.0			
Note				
Confirm and close Close				

Fig. 1.7: RipEX\_A – Babel Network settings

Interface must be "radio" (interface name) and its type "Wireless" (it is a radio channel). Change the Rx cost from default 128 to 100.

Increase the Hello interval to 30 seconds. This is important so that Babel does not send too many overhead packets on the radio channel. Decision on this (and other) parameters is always a "tradeoff". Lower the interval, quicker protocol operations such as topology changes detection or complete routing propagation, but in a cost of higher Radio channel utilization.

Keep in mind that if you run low FSK or even low QAM modulation, the primary goal is still correct SCADA operation and not overhead data.

Another parameter to be set is Update interval multiplier and is set to default 4. This number multiplies the Hello interval (in our example it is  $4 \times 30$  seconds = 2 minutes). Router updates packets are sent in ~ 2 minutes intervals.

This setup, the same in all 4 RipEX2 units, results in approximately 1 Babel packet per 10 second, i.e., 100 bps. The topology change can be detected and spread across a network within quite a long interval, ranging from 30 seconds to several minutes. Optimize these values to suit your topology, used modulation type and SCADA traffic.

Go to another panel "Static rules" and configure advertised network 192.168.1.0/24 with default metric equal to 0.

RIPEX2 RIPEX_A @10.9.8.7	BABEL
Unit time: 2021-07-16 14:17:31	Common Network Static rules Import filter Export filter
(UTC+2)	Static rules
STATUS	Destination IP / Destination mask 192.168.1.0/24 Metric 0
✿。 SETTINGS	Note
Interfaces	+ Add rule
Routing	
Static	
BABEL	

#### Fig. 1.8: RipEX\_A – Static rules settings

Go to another panel "Import filter" and create a new rule. The only non-default parameter we need to set is "Local preferred source address" (LPSA) – set it to 192.168.1.1 (Ethernet IP of RipEX\_A). This is a preferred source IP address for locally generated packets.

Edit rule		×
	Enable rule	
Filter network	Off	~
Action	Accept	~
Set preference	Off	*
Local preferred source address	192.168.1.1	
Note		
Confirm and close	Close	

Fig. 1.9: RipEX\_A – Import filter settings

We do not configure anything in the "Export filter" panel.

The configuration is complete. Go to the "Changes" menu (upper right corner button) and save the changes.

## 1.3. RipEX\_B, RipEX\_C and RipEX\_D configuration

Other units share most of the settings we configured in RipEX\_A. We just highlight the differences compared to RipEX\_A setup.

RipEX_B		
Unit name	RipEX_B	
Ethernet IP	192.168.2.1/24	
Radio IP	10.10.10.2	
Babel routing		
	Router ID	2.2.2.2
	Static rules	192.168.2.0/24
	Import filter LPSA	192.168.2.1
RipEX_C		
Unit name	RipEX_C	
Ethernet IP	192.168.3.1/24	
Radio IP	10.10.10.3	
Babel routing		
	Router ID	3.3.3.3
	Static rules	192.168.3.0/24
	Import filter LPSA	192.168.3.1
RipEX_D		
Unit name	RipEX_D	
Ethernet IP	192.168.4.1/24	
Radio IP	10.10.10.4	
Babel routing		
	Router ID	4.4.4.4
	Static rules	192.168.4.0/24
	Import filter LPSA	192.168.4.1

## 1.4. Diagnostics and Testing

Once configured, you can wait until the network converges to the correct routing, or you can also reboot all the units so the protocol starts from scratch.

### 1.4.1. Routing

Go to the DIAGNOSTICS - Routing menu and check the Babel panel. Let's go through individual tables.

## **BABEL** routing

```
Interfaces
BIRD 2.0.7 ready.
babel1:
Interface State RX cost Nbrs Timer Next hop (v4) Next hop (v6)
radio Up 100 3 0.721 10.10.10.1 fe80::202:a9ff:fe20:6f9
```

Fig. 1.10: Babel diagnostics - Interfaces

In the 1 <sup>st</sup> table, we can see all interfaces on which Babel is either configured or found. If the interface is missing, but should be there, it probably failed while initialization (e.g., because of missing link IPv6 address).

State Interface state, either "up" or "down"

Rx cost It displays configured received cost (for other neighbor)

Nbrs Number of detected neighbors on particular interface

Time Number of seconds until next Babel Hello or Update transmission

Nexthop What "next-hop" address the router offers to neighbors (IPv4 and IPv6)

• Can be useful in case of configured 0.0.0.0 - to check if correct IP is really used

Neighbors					
BIRD 2.0.7 ready.					
babel1:					
IP address	Interface	Metric	Routes	Hellos	Expires
fe80::202:a9ff:fe20:ae3	radio	100	3	16	44.662
fe80::202:a9ff:fe20:531	radio	100	3	16	27.512
fe80::202:a9ff:fe20:789	radio	100	3	16	36.432

#### Fig. 1.11: Babel diagnostics – Neighbors

IP address	Neighbor's IPv6 link address
Interface	Interface on which the neighbor was found
Metric	Current metric for receiving from neighbor
Routes	Number of routes received from neighbor

#### Hellos Number of received Hello packets (from up to 16)

Expires [s] Time until reception of another expected Hello packet from neighbor

Routes						
BIRD 2.0.7 ready.						
babel1:						
Prefix	Nexthop	Interface M	Metric	F	Seqno	Expires
192.168.2.0/24	10.10.10.3	radio	200		2	303.655
192.168.2.0/24	10.10.10.4	radio	200		2	339.506
192.168.2.0/24	10.10.10.2	radio	100	*	2	345.424
192.168.3.0/24	10.10.10.3	radio	100	*	2	303.655
192.168.3.0/24	10.10.10.4	radio	200		2	339.506
192.168.3.0/24	10.10.10.2	radio	200		2	345.424
192.168.4.0/24	10.10.10.3	radio	200		2	303.655
192.168.4.0/24	10.10.10.4	radio	100	*	2	339.506
192.168.4.0/24	10.10.10.2	radio	200		2	345.424

#### Fig. 1.12: Babel diagnostics - Routes

A list of all routes from all neighbors. It may even consist "loops" for routes which are not currently used, but once the protocol switches to them, loops are solved and "fixed". Or in other words, such routes are are not considered as "candidates".

Prefix	Range of destination IP addresses for a particular rule
Nexthop	Next-hop address
Metric	Current cost to the destination
F	* for currently active rule, + for feasible next candidate for active rule
Seqno	Update packet sequence number which announced the rule
Expires [s]	Time until the rule expires

Entries					
BIRD 2.0.7 ready.					
babel1:					
Prefix	Router ID	Metric	Seqno	Routes	Sources
192.168.1.0/24	00:00:00:00:01:01:01:01	0	1	0	0
192.168.2.0/24	00:00:00:00:02:02:02:02	100	2	3	1
192.168.3.0/24	00:00:00:00:03:03:03:03	100	2	3	1
192.168.4.0/24	00:00:00:00:04:04:04:04	100	2	3	1

#### Fig. 1.13: Babel diagnostics - Entries

Prefix	Prefix of routed address range
Router ID	Router ID – first 4 Bytes either zeros, or randomized. Second 4 Bytes equal to Router ID set in the configuration
Metric	Current cost to the destination
Seqno	Number of various routing rules for particular prefix/subnet

#### Routes Time until the rule expires

Sources Number of various routers which export a particular prefix to Babel network

Table babel_ipv4
BIRD 2.0.7 ready.
Table babel_ipv4:
192.168.1.0/24 unreachable [static_babel 11:33:35.727] * (1000)
Type: static univ
Babel.metric: 0
192.168.2.0/24 unicast [babel1 11:50:05.172 from fe80::202:a9ff:fe20:789] (210/100) [00:00:00:00:02:02:02:02]
via 10.10.10.2 on radio
Type: Babel univ
Babel.metric: 100
Babel.router_id: 00:00:00:02:02:02:02
192.168.3.0/24 unicast [babel1 11:48:44.385 from fe80::202:a9ff:fe20:ae3] (210/100) [00:00:00:00:03:03:03:03]
via 10.10.10.3 on radio
Type: Babel univ
Babel.metric: 100
Babel.router_id: 00:00:00:03:03:03:03
192.168.4.0/24 unicast [babel1 11:49:56.247 from fe80::202:a9ff:fe20:531] (210/100) [00:00:00:00:04:04:04:04]
via 10.10.10.4 on radio
Type: Babel univ
Babel.metric: 100
Babel.router_id: 00:00:00:00:04:04:04

Fig. 1.14: Babel diagnostics – Table babel\_ipv4

The table consist of advanced and detailed information of all routes from Babel table.

#### 1.4.2. Tools

Now, check the remote unit's IP accessibility. Go to the DIAGNOSTICS – Tools – ICMP ping menu.

RipEX2 RipEX_A @10.9.8.7	TOOLS 🗏 Changes 🕫 Notifications
Unit time: 2021-07-16 15:37:12 (UTC+2)	ICMP Ping Routing
<ul><li>STATUS</li><li>SETTINGS</li></ul>	Parameters           Length [Bytes]         200         Period [ms]         1000         Timeout [ms]         10000         Count         4           Source IP         Destination IP         192.168.4.1         192.168.4.1         192.168.4.1
Verview	Controls  Start Clear
Events Statistics Monitoring Routing	Output PING 192.168.4.1 (192.168.4.1) from 192.168.1.1 : 200(228) bytes of data. 208 bytes from 192.168.4.1: icmp_seq=1 ttl=64 time=79.3 ms 208 bytes from 192.168.4.1: icmp_seq=2 ttl=64 time=72.7 ms
• Tools Support	208 bytes from 192.168.4.1: icmp_seq=3 ttl=64 time=66.2 ms 208 bytes from 192.168.4.1: icmp_seq=4 ttl=64 time=85.7 ms 192.168.4.1 ping statistics 4 packets transmitted, 4 received, 0% packet loss, time 3004ms rtt min/avg/max/mdev = 66.268/76.029/85.784/7.291 ms

Fig. 1.15: RipEX\_A – Diagnostics – Tools – ICMP Ping

Fill in the Destination IP field with required IP (try all remote LAN IPs). Check if it gets through. If not, check the Routing diagnostics for available routes.

RipEX2 RipEX_A @10.9.8.7 1%1 Remote ad	TOOLS
Unit time: 2021-07-16 15:38:28 (UTC+2)	ICMP Ping Routing
	Parameters
Ø STATUS	Destination IP 192.168.4.1
settings	Controls
	► Run
Overview	Output
Events	102 168 4 1 via 10 10 10 4 dev radio ero 102 168 1 1
Statistics	
Monitoring	
Routing	
• Tools	

You can also check what route is used for particular destination in another panel "Routing".

Fig. 1.16: RipEX\_A – Diagnostics – Tools – Routing

You can debug issues further in Statistics and/or Monitoring menus.

# 2. Mesh topology with Radio and Relay filters

This second example uses the same topology as described within the first example.

Since 2.1.7.0 firmware, Babel can control the path priorities based on current RSS/MSE values (radio filter). This is very handy in case the link qualities differ and some of the links are barely usable. With such filter, we can configure RipEX2 units to ignore such bad quality links.

A second improvement is the Relay filter which selects which of the obtained rules are being forwarded and which are not - i.e. if the unit is a repeater/forward node, or the end station (terminal). It can also be used to alter routing rule metrics and thus, prioritize or discriminate some of the nodes/units.

Within the following example, we will configure both the Radio and Relay filters.

Keep in mind all Babel parameters and its filters are well explained within the manual.

### 2.1. Radio filters

Configure Radio filters in all your networks utilizing Babel protocol via the Radio channel. Why?Neither Babel nor other dynamic routing protocols can take RSS/MSE values into account. In most of the networks, some links have very good RSS and MSE values running high QAM modulations. Links with (very) bad RSS/MSE values should not be used for routing user traffic in case there are links with higher quality. Without the Radio filter, short Babel management/overhead data may be able to successfully go through such links, but important user traffic might be corrupted resulting in unreliable SCADA operation.

Babel itself has a known mechanism of Hello packets and increasing link metrics based on Hello packets success rate of going through particular links. We enhanced this behaviour (proprietary) so that some of the Hello packets are discarded if received with RSS or MSE values below set thresholds.

Default thresholds are:

- RSS
  - Soft: -110 dBm
  - Hard: -130 dBm
- MSE
  - Soft: -10 dB
  - Hard: -5 dB

If the RSS/MSE values of received Hello packet is worse than the "Hard" limit, it is always discarded. In case the RSS/MSE values are better than "Soft" thresholds, they are always accepted.

If the values are between thresholds, Hello packets are discarded randomly with probability increasing linearly between Soft and Hard limits. The probability of a Hello packet being received by the filter is calculated as the product of the probabilities based on RSS and MSE. If at least one of the variables exceeds the Hard limit, the packet is always discarded. The limit setting is either global or individual for each Hello packet source address (radio IP address). The individual setting is only used if there is a translation of the link address to a radio IP address (ARP request/reply for the given IP had to be sent/received so that you don't see 0.0.0.0 Radio IP within the Statistics menu, but a correct IP for a given Link address). Discarding a Hello packet will cause the Babel protocol metric to be increased and the link to be disadvantaged.

Keep in mind each modulation has its own sensitivity levels. If you combine multiple modulations within the network, individual link thresholds are recommended.

Sensitivity levels (RSS) can be read from *the Specification*<sup>1</sup>.

MSE recommendations can be read within the application note on RACOM website<sup>2</sup>.

Let's see one example. In the whole network, we run the QAM modulation  $\pi$ /4-DQPSK, within the 25 kHz channel. The sensitivity level (for BER 10<sup>-6</sup>) is -111 dBm. Recommended MSE is -14 dB.

Default thresholds could be:

- RSS
  - Soft: -91 dBm (20 dBm Fade margin)
  - Hard: -111 dBm (sensitivity level from the Specification)
- MSE
  - Soft: -14 dB (i.e. real value at least equal to the recommended -14 dB)
  - Hard: -10 dB (based on user experience, no precise calculation for this value)

Received Hello packets' RSS/MSE values and its operation:

- RSS = -80 dBm, MSE = -20 dB
  - Packet is passed to Babel successfully
- RSS = -99 dBm, MSE = -15 dB
  - $\circ\,$  RSS value is between the Hard and Soft thresholds and = passing the Hello packet probability = 60 %
  - $\circ$  MSE is over the Soft limit = passing the Hello packet probability = 100 %
  - Resulting probability is 0.6 \* 1 = 60 % (6 out of 10 Hello packets are processed by Babel, 4 our discarded immediately)
- RSS = -105 dBm, MSE = -9 dB
  - MSE is below the Hard threshold = Hello packet is discarded

Let's imagine we have one unit and it has at least two neighbors (10.10.10.2 and 10.10.10.3 are the neighbors' Radio IP addresses). For the neighbor '2' the modulation is set to 64QAM, for the neighbor '3' the modulation is '256QAM'. We set two individual thresholds, e.g.:

• Link to 10.10.10.2

• RSS

- Soft: -72 dBm
- Hard: -92 dBm
- $\circ$  MSE
  - Soft: -27 dB
  - Hard: -22 dB
- Link to 10.10.10.3
  - RSS
    - Soft: -68 dBm
    - Hard: -88 dBm
  - MSE
    - Soft: -33 dB
    - Hard: -28 dB

<sup>&</sup>lt;sup>1</sup> https://www.racom.eu/eng/products/m/ripex2/tech.html#tech-det

<sup>&</sup>lt;sup>2</sup> https://www.racom.eu/download/hw/ripex/free/eng/1\_application/ripex2-app-mse-en.pdf

Further optimization can be discussed with our technical support team via support@racom.eu<sup>3</sup>.

Within the example topology we utilize 25kHz channel and 16DEQAM modulation. We do not set any Individual thresholds, in case you need them, check the information above. Default thresholds in all four units will be set to

 RSS o Soft: -78 dBm • Hard: -98 dBm MSE ○ Soft: -22 dB • Hard: -17 dB RipEX 2 RipEX\_D @10.9.8.7 1<sup>10</sup> Remote access BABEL Unit time: 2024-06-17 15:06:27 (UTC+2) Radio filter Common Static rules Import filter Export filter Relay filter Status STATUS 🕏 🤹 SETTINGS Radio filter Enabled Interfaces Default thresholds Routing Static If a hello packet transmission exceeds the threshold, it is automatically discarded in order to prefer more reliable links Link management RSS threshold (soft) [-dBm] 78 0 Babel RSS threshold (hard) [-dBm] 98 0 OSPF MSE threshold (soft) [-dB] 22 0 RGP MSE threshold (hard) [-dB] 17 \$

Fig. 2.1: Default Radio filter thresholds for 16DEQAM

Save this change in all 4 units.

٠

<sup>&</sup>lt;sup>3</sup>mailto:support@racom.eu

Check the Babel Status after a while. Based on your conditions, you may see similar output.

RipEX2 RipEX_D @10.9.8.7	1 <sup>04</sup> 1 Remote access				BABEL
	<ul> <li>Status</li> </ul>				
Unit time:	Interfaces				
2024-06-18 08:19:03	BTRD 2.13.1 ready.				
(UTC+2)	babel1:				
(0.0.2)	Interface State Auth	RX cost Nbrs Timer N	ext hop (v4)	Next hop (v6)	
	radio Up No	100 3 1.028 10	0.10.10.4	fe80::202:a9ff:fe20:6f9	
	Neighbors				
	BIRD 2 13 1 ready				
(*) STATUS	babel1:				
	TP address	Interface Metric Bout	es Hellos Exp	ires Auth	
	fe80::202:a9ff:fe20:773	radio 100	3 16 37	.702 No	
🗣 SETTINGS	fe80::202:a9ff:fe20:789	radio 100	2 16 41	.574 No	
	fe80::202:a9ff:fe20:531	radio 256	2 10 30	465 No	
Interfaces	Routes				
Interfaces	BIRD 2 13 1 ready				
D	babel1:				
Routing	Prefix	Nexthop	Interface I	Metric E Seano Expires	
	192 168 1 0/24	10.10.10.1	radio	256 * 48 318 465	
Static	192.168.1.0/24	10.10.10.3	radio	313 + 48 364,796	
	192.168.2.0/24	10.10.10.2	radio	100 * 4 379,576	
Link management	192.168.2.0/24	10.10.10.3	radio	300 4 364.796	
	192.168.3.0/24	10.10.10.2	radio	300 18 379,576	
Babel	192.168.3.0/24	10.10.10.1	radio	468 18 318,465	
0.005	192.168.3.0/24	10.10.10.3	radio	100 * 18 364.796	
OSPF	Entries				
DCD	BIRD 2 13 1 ready				
BGP	babel1:				
	Prefix	Bouter ID	Metric Seano	Routes Sources	
Firewall	192,168,1,0/24	00:00:00:00:01:01:01:01	256 48	2 1	
-	192,168,2,0/24	00:00:00:00:01:01:01:01	100 4	2 1	
VPN	192.168.3.0/24	00:00:00:00:02:02:02:02	100 18	3 1	
	192 168 4 0/24	00.00.00.00.03.03.03.03	0 13	0 0	

#### Fig. 2.2: RipEX\_D Babel routing

Within our example output, signals to 10.10.10.2 and 10.10.10.3 are better than configured thresholds, but signals to 10.10.10.1 are between the soft and hard thresholds, i.e. we only got 10 out of 16 Hello packets correctly and the direct metric is now 256. It is still the lowest metric so the direct radio link is used.

Based on Statistics, MSE values are better than the soft threshold, but signals to 10.10.10.1 are in average -83 dBm which is close to soft threshold, but worse. The minimum measured signal is even - 89 dBm. This is the reason why some of the Hello packets are discarded and the metric is higher.

Link address	ID addross	Hoodor count	RSS [c	RSS [dBm]								
	IP auuress		avg	dev	min	max						
:20:05:31	10.10.10.1	167	-83	2	-89	-77						
:20:07:73	10.10.10.3	190	-79	1	-82	-76						
:20:07:89	10.10.10.2	162	-76	0	-78	-76						

#### Radio signal statistics

Fig. 2.3: RipEX\_D Radio signal statistics

Discarding is based on probability so once you check the Status again, it can be different, e.g.:

Neighbors								
BIRD 2.13.1 ready.								
babel1:								
IP address	Interface	Metric	Routes	Hellos	Expires	Aut	h	
fe80::202:a9ff:fe20:773	radio	133	3	12	30.111	No		
fe80::202:a9ff:fe20:789	radio	100	2	16	33.985	No		
fe80::202:a9ff:fe20:531	radio	177	2	12	22.874	No		
Routes								
BIRD 2.13.1 ready.								
babel1:								
Prefix	Nexthop			Interfa	ce Metri	c F	Seqno	Expires
192.168.1.0/24	10.10.10.1			radio	17	7 *	48	310.875
192.168.1.0/24	10.10.10.3			radio	28	5	48	357.206
192.168.2.0/24	10.10.10.2			radio	10	0 *	4	371.985
192.168.2.0/24	10.10.10.3			radio	36	1	4	357.206
192.168.3.0/24	10.10.10.2			radio	32	8	18	371.985
192.168.3.0/24	10.10.10.1			radio	32	9	18	310.875
192.168.3.0/24	10.10.10.3			radio	13	3 *	18	357.206

Fig. 2.4: RipEX\_D Babel routing, #2

# $\triangle$

### Important

Individual thresholds can be taken into account only if the IP address is known and correct Link address and IP address translation has been done. Check your Statistics. For each remote you only see 0.0.0.0 as the IP address, the default thresholds are used. Once there is any communication with a particular IP address, ARP data are exchanged and the IP address is known afterwards. In case there is regular traffic with all the neighbors (SCADA, ...), the IP addresses are always known, but there might be multiple other sites within the radio coverage (visible in Statistics), but with no user traffic resulting in only Link addresses within Statistics and IPs being 0.0.0.

	1 <sup>0</sup> 1 Remote access									STATIS	TICS									<sup>2</sup> ≣ cr	langes	
	Historical	Differential																				
Unit time: 2024-06-14 15:01:07 (UTC+2)	Parameter	s																				
🔊 STATUS	Radio int	erface statistics																				
🍫 SETTINGS	Radio pro	nal statistics	essable statistics																			
Vr DIAGNOSTICS     Overview     Information     Events     Statistics	Radio sig Serial pro Ethernet Cellular is Cellular s Cellular s Measures	nal non-addres btocols statistics statistics interface statistic tate statistics ignal statistics ments 2024-06-14 1	sable statistics	24.06.14	15-01		Now		bour								latt day 10	et hour	Ma	re option		
Monitoring — Tools	Display	2024-00-141		24-00-14	15.01	] ∟			nour								Last Day La	ist Hour	INC	re option	, ,	
III ADVANCED	Data																					
	Radio signal statistics																					
	Link address	IP address	Header count	RSS [d	IBm]			Phy h	ieader M	ISE [dB]		Freq offset [Hz]	a offset (Hz) Att1 (%)		[dB]		- Data count	Data	MSE [dE	MSE [dB]		
	-20-06-E9	0000	150	avg -85	dev	-87	-83	avg -33	dev 1	-39	max -30	175	0	avg	min	max	150	avg -34	dev 1	-37	max -32	
	:20:07:73	0.0.0.0	150	-83	1	-86	-81	-35	1	-40	-32	91	0	0	0	0	150	-36	1	-39	-32	

Fig. 2.5: Unknown IP addresses for Link addresses

## 2.2. Relay filters

Relay filters can be used for

- · Setting any unit to be the end station not forwarding any obtained/received routes from its neighbours
- Increasing metric for routes being propagated i.e. to disadvantage paths leading through this unit within the Babel dynamic routing (higher routing rule metric via this unit)
- Limiting number of propagated routing rules i.e. decrease the number of routes within the network

In case the unit is not supposed to propagate any obtained routes further, so it serves as the end-station (terminal, not repeater), you can reject all rules to be propagated/relayed.

Just go to the SETTINGS > Routing > Babel > Relay filter menu and change the policy to "Reject".

Rip∈×2 RipEX_D @10.9.8.7	Remote access	BABEL
Unit time: 2024-06-18 09:18:04 (UTC+2)	Common Network Static rules Import filter Export filter	Relay filter Radio filter
🕫 STATUS	Status	
SETTINGS	Filter policy Reject 🗸	
Interfaces	Relay filter rules	
<ul> <li>Routing</li> <li>Static</li> <li>Link management</li> </ul>	<ul> <li>Table does not contain any data.</li> <li>Add rule</li> </ul>	
Babel		

Fig. 2.6: RipEX\_D filter policy set to "Reject"

Until this change, e.g. the RipEX\_A could have similar Babel routes.

Routes						
BIRD 2.13.1 ready.						
babel1:						
Prefix	Nexthop	Interface	Metric	F	Seqno	Expires
192.168.3.0/24	10.10.10.3	radio	246	*	20	356.948
192.168.3.0/24	10.10.10.4	radio	383	+	20	324.205
192.168.3.0/24	10.10.10.2	radio	428	+	20	371.729
192.168.4.0/24	10.10.10.3	radio	346	+	14	356.948
192.168.4.0/24	10.10.10.4	radio	283	*	14	324.205
192.168.4.0/24	10.10.10.2	radio	328	+	14	371.729

Fig. 2.7: RipEX\_A Babel routes

We can see RipEX\_D (10.10.10.4) is also used as a gateway for 192.168.3.0/24 (RipEX\_C). If we set that Reject policy in RipEX\_D, no rule will not be propagated from RipEX\_D to other units except its own configured subnets.

Routes					
BIRD 2.13.1 ready.					
babel1:					
Prefix	Nexthop	Interface	Metric	F	Seqno Expires
192.168.3.0/24	10.10.10.3	radio	314	*	21 357.779
192.168.3.0/24	10.10.10.2	radio	580		21 372.560
192.168.4.0/24	10.10.10.3	radio	444	+	14 357.779
192.168.4.0/24	10.10.10.4	radio	141	*	14 325.035
192.168.4.0/24	10.10.10.2	radio	274	+	14 372.560

Fig. 2.8: RipEX\_A Babel routes after setting RipEX\_D to reject relay policy

You can also increase metric for the routes being propagated, e.g. to disadvantage particular repeater. Let's set RipEX\_C (10.10.10.3) to increase metrics obtained from this repeater by 100.

	x_C@10.10.10.3 1 <sup>0</sup> 1	RAR	FI
		Add relay rule	×
	Common Network Static rules Import filter Expo	Enable rule	
(UTC+2)		Filter network	Match 🗸
	Status	Network IP / Network mask	0.0.0/0
😝 STATUS		Mask from	0
		Mask to	32
SETTINGS	Filter policy Accept	Action	Accept 🗸
	Relay filter rules	Filter metric	Off 🖌
Routing		Increase metric	On 🗸
	able does not contain any data.	Added metric	100 🗘
	+ Add rule	Note	
Babel			
		Confirm and close	Close

Fig. 2.9: RipEX\_C Relay filter to increase metric by 100

We can see similar output in neighbouring units, e.g. from RipEX\_D.

Routes						
BIRD 2.13.1 ready.						
babel1:						
Prefix	Nexthop	Interface	Metric	F	Seqno	Expires
192.168.1.0/24	10.10.10.1	radio	100	*	52	311.901
192.168.1.0/24	10.10.10.3	radio	333	+	52	357.797
192.168.2.0/24	10.10.10.2	radio	100	*	6	373.014
192.168.2.0/24	10.10.10.3	radio	300	+	6	357.797
192.168.3.0/24	10.10.10.2	radio	200	+	23	373.014
192.168.3.0/24	10.10.10.1	radio	233	+	23	311.901
192.168.3.0/24	10.10.10.3	radio	100	*	23	357.797

Fig. 2.10: RipEX\_D Babel routes with higher metrics via 10.10.10.3

Routes to 192.168.1.0/24 and 192.168.2.0/24 via 10.10.10.3 have the highest metric. Direct route to its own subnet 192.168.3.0/24 is not disadvantaged this way, as required/configured. If you need to increase the metric of this direct path as well, do it via Static rules tab of Babel menu and increase the "Metric" parameter for the given Destination IP/mask.

Last, but not least, is to limit just some of the propagated routes, not all, via the "Reject" Relay filter policy. Go to the RipEX\_D again and change the Policy to "Accept", but let's reject propagating 192.168.2.0/24 and 192.168.3.0/24 networks. I.e. the unit will only propagate 192.168.1.0/24 (and of course its own subnet, but this not controlled by the Relay filter).

RipEX2 RipEX_D @10.9.8.7	Remote access	RAR	-1 /
		Edit relay rule	×
		Exp: Enable rule	<b>~</b>
(UTC+2)		Filter network	Match 🗸
	Status	Network IP / Network mask	192.168.2.0/23
🚱 STATUS		Mask from	24
		Mask to	24
SETTINGS	Filter policy Accept	Action	Reject 🗸
	Relay filter rules	Filter metric	Off 🗸
Routing		Note	192.168.[2-3].0/24 re
	• 192.168.[2-3].0/24 reject	Confirm and close	Close
	[24,24]		
Babel			
	Added metric: Off		
	Reject 💼		
	+ Add rule		

Fig. 2.11: RipEX\_D Relay filter - rejecting 192.168.2.0/23 subnets with /24 mask

We know we should receive 192.168.2.0/24 and 192.168.3.0/24, i.e. /24 mask in both cases. We can limit the Mask from/to parameters to suit our scenario. We could also have two individual rules or do it via a single rule using /23 mask.

The default policy is "accept" so the 192.168.1.0/24 network will be propagated.

Routes					
BIRD 2.13.1 ready.					
babel1:					
Prefix	Nexthop	Interface	Metric	F	Seqno Expires
192.168.1.0/24	10.10.10.1	radio	200	+	52 361.851
192.168.1.0/24	10.10.10.4	radio	200	*	52 375.410
192.168.2.0/24	10.10.10.2	radio	100	*	6 302.427
192.168.4.0/24	10.10.10.4	radio	100	*	14 375.410
192.168.4.0/24	10.10.10.1	radio	300		14 361.851
192.168.4.0/24	10.10.10.2	radio	200		14 302.427

Fig. 2.12: RipEX\_C Babel routes without 10.10.10.4 for 192.168.[2-3].0/24 networks

RipEX\_C does not use 10.10.10.4 as a gateway to reach 192.168.2.0/24 and 192.168.3.0/24 networks. Obviously, it uses the gateway to reach 192.168.1.0/24 and 192.168.4.0/24.



### Note

If you configure Babel routing for the fist time in your network, it is suggested to reboot each device after final configuration changes so the dynamic protocol is fully restarted in all units with correct and current configuration.

Sure, you can combine all the options in every unit within the whole network to suit your requirements.

Further optimization can be discussed with our technical support team via *support@racom.eu*<sup>4</sup>.

<sup>&</sup>lt;sup>4</sup> mailto:support@racom.eu



## 3. Two repeaters on the RF channel

Fig. 3.1: Example 3 – Two repeaters topology

## 3.1. Description and Configuration

The third example is very similar to the first one, but the difference is that RipEX\_A and RipEX\_D can only communicate via repeaters and repeaters do not "see" each other. This is a typical situation in the field. For the test in an office, we use a frequency pair so that RipEX\_A and RipEX\_D send data on frequency 415.5 MHz and receive on 436 MHz, whereas RipEX units B and C have it vice versa.

Otherwise, the setup is completely the same.

If RipEX\_A and RipEX\_D are about to communicate only to each other, it might be useful to turn off "Routing offering".

RipEX2 RipEX_A @10.9.8.7	ា <sup>រ</sup> ា Remote access
Unit time:	Common
2021-07-19 07:41:48 (UTC+2)	Network Static rules import hiter Export hiter
	Common settings
O STATUS	Active
SETTINGS	Router ID 1.1.1.1
Interfaces	Routing offering Off
Routing	
Static	
BABEL	

#### Fig. 3.2: Routing offering

That would limit forwarding received routes from units A and D. Thus, RipEX\_B and C will not be able to reach each other and there will be less Babel routes in all devices, you can compare it within this example. We will enable full routing so "Routing offering" will be "on" in all four units.

Change the frequencies in all units accordingly. Antenna configuration can be set to Single or Dual, it does not change the behaviour, neither performance while testing with dummy loads on your desk.

RipEX\_A, RipEX\_D

- TX: 415.500.000 MHz
- RX: 436.000.000 MHz

RipEX\_B, RipEX\_C

- TX: 436.000.000 MHz
- RX: 415.500.000 MHz

## 3.2. Diagnostics and Testing

Check the Routing in Diagnostics menu. All three routes should be accessible via both repeaters 10.10.10.2 and 10.10.10.3 with the same price. One route is selected to be used; another one is a failure (candidate) option.

RipEX2 RipEX_A @10.9.8.7	Remote access
Unit time:	Static Dynamic BABEL OSPF BGP
2021-07-19 06:02:20 (01C+2)	BABEL routing
STATUS	Interfaces
🎎 SETTINGS	BIRD 2.0.7 ready. babel1: Interface State RX cost Nbrs Timer Next hop (v4) Next hop (v6) radio Up 100 2 6.431 10.10.1 fe80::202:a9ff:fe20:6f9
V DIAGNOSTICS	Neighbors
Overview — Events	BIRD 2.0.7 ready.           babel1:           IP address           Interface           Metric Routes Hellos           Expires           fe80::202:a9ff:fe20:789           radio         100           100         3           6         33.409           fe80::202:a9ff:fe20:ae3         radio           100         3         6           33.336         33.336
Statistics	Routes BIRD 2.0.7 ready.
Monitoring	Prefix Nexthop Interface Metric F Seque Expires
Routing	192.168.2.0/24         10.10.10.3         radio         300         1 322.337           192.168.3.0/24         10.10.10.3         radio         100 *         1 372.337
	192.168.3.0/24         10.10.10.2         radio         300         1 325.411           192.168.4.0/24         10.10.10.3         radio         200 *         1 372.337           192.168.4.0/24         10.10.10.2         radio         200 +         1 325.411

Fig. 3.3: Babel Routing state

If you want to prioritize one of the repeaters, you can do it by decreasing the cost in this repeater. E.g., decrease it to 10 and check the routing again.

Edit interface		×
(	Active	
Interface	radio	
Туре	Wireless	*
Rx cost	10	÷
Hello interval [s]	30	\$
Update interval multiplier	4	÷
Advertised next hop	0.0.0.0	
Note		
Confirm and clo	Close	

Fig. 3.4: Rx cost decreased in RipEX\_C

Neighbors									
BIRD 2.0.7 ready.									
babel1:									
IP address	Interface	Metric	Route	s Hello	os Exp	pires			
fe80::202:a9ff:fe20:789	radio	100		3 1	16 18	8.264			
fe80::202:a9ff:fe20:ae3	radio	10		3 1	16 17	7.944			
Routes									
BIRD 2.0.7 ready.									
babel1:									
Prefix	Nexthop			Inter	face	Metric	F	Seqno	Expires
192.168.2.0/24	10.10.10.2			radio	o	100	*	1	340.265
192.168.2.0/24	10.10.10.3			radio	C	210		1	386.947
192.168.3.0/24	10.10.10.3			radio	o	10	*	1	386.947
192.168.3.0/24	10.10.10.2			radio	o	300		1	340.265
192.168.4.0/24	10.10.10.3			radio	D C	124	*	1	386.947
192.168.4.0/24	10.10.10.2			radio	c	200	+	1	340.265
Entries									
BIRD 2.0.7 ready.									
babel1:									
Prefix	Router ID			Metric	Seqno	Rout	es	Source	es
192.168.1.0/24	00:00:00:00	:01:01:0	1:01	0	1	1	0		0
192.168.2.0/24	00:00:00:00	:02:02:0	2:02	100	1	1	2		1
192.168.3.0/24	00:00:00:00	:03:03:0	3:03	10	1	1	2		1
192.168.4.0/24	00:00:00:00	:04:04:0	4:04	124	1	1	2		1

#### Fig. 3.5: Changed routing costs

As you can see, route to 192.168.3.0/24 has a metric "10" now. Also, a route to 192.168.4.0/24 is 124, it could be 110, but some Hello packets were lost and it is increased for a while. Original values were 100 and 200.

And of course, Nexthop gateway is set to 10.10.10.3 for both routes.

### 3.2.1. Tools

Unit time: 2021-07-19 08:40:18 (UTC+2)	TOOLS ?
Ø STATUS	ICMP Ping Routing
🏟 SETTINGS	Parameters
	Length [Bytes]         200         Period [ms]         1000         Timeout [ms]         10000
Overview	Count 1000 Source IP 🗎 Destination IP 192.168.4.1
 Events	Controls
 Statistics	Stop
Monitoring	Output
Routing	
Tools	PING 192.168.4.1 (192.168.4.1) from 192.168.1.1 : 200(228) bytes of data.
	200 Syste from 152.100.3.1. Tomp_Seq-1 001-00 Sime-100 MS

Now, start a ping from RipEX\_A to RipEX\_D. Set a count to 1000.

Fig. 3.6: RipEX\_A – Ping to 192.168.4.1 (RipEX\_D)

In another browser panel/window, open RipEX\_A GUI again and check the Routing diagnostics. You can keep clicking on the "Refresh" button to see current data.

Turn off RipEX\_C (currently used RipEX as a repeater for traffic). You will stop seeing successful pings, but after some time, you should start seeing correct replies (via RipEX\_B). You should also see metrics being increased in the Routing diagnostics menu.

Note that even working routes can have higher metric due to propagating forwarded routes from "end devices". If you had disabled RipEX\_A and RipEX\_D "routing offering" option, metrics would be more stable.

The last option is to use a tool called "**RSS ping**" (since RipEX2 firmware 2.0.5.0). RSS ping is a diagnostic tool for the radio performance measurement (RSS and MSE) of the individual radio hops within a RipEX2 network. RSS ping sends UDP data on port 8906. You should see changes in used repeater on the end to end communication path.

Neighbors								
BIRD 2.0.7 ready.								
babel1:								
IP address	Interface	Metric	Routes	Hellos	Expires			
fe80::202:a9ff:fe20:789	radio	100	3	16	28.385			
fe80::202:a9ff:fe20:ae3	radio	65535	3	6	28.078			
Routes								
BIRD 2.0.7 ready.								
babel1:								
Prefix	Nexthop			Interfa	ace Metric	F	Seqno	Expires
192.168.2.0/24	10.10.10.2			radio	100	*	1	320.388
192.168.2.0/24	10.10.10.3			radio	65535		1	7.078
192.168.3.0/24	10.10.10.3			radio	65535	+	1	7.078
192.168.3.0/24	10.10.10.2			radio	213		1	320.388
192.168.4.0/24	10.10.10.3			radio	65535	+	1	7.078
192.168.4.0/24	10.10.10.2			radio	214		1	320.388

Fig. 3.7: RipEX\_C turned off, changes in metrics

```
208 bytes from 192.168.4.1: icmp_seq=20 ttl=63 time=177 ms
208 bytes from 192.168.4.1: icmp_seq=21 ttl=63 time=164 ms
208 bytes from 192.168.4.1: icmp_seq=22 ttl=63 time=171 ms
ping: sendmsg: No route to host
ping: sendmsg: No route to host
208 bytes from 192.168.4.1: icmp_seq=320 ttl=63 time=297 ms
208 bytes from 192.168.4.1: icmp_seq=321 ttl=63 time=158 ms
208 bytes from 192.168.4.1: icmp_seq=322 ttl=63 time=210 ms
```

Fig. 3.8: RipEX\_A – ping

As you can see, it took 5 minutes to Babel to increase the metric high enough so the backup repeater is used. It is mainly due to low metric "10" of RipEX\_C setup. If you change it to 50, it will be quicker (its metric goes up faster).

Turn on RipEX\_C again and check the Routing again. Note you can also check Monitoring menu for live traffic monitoring.

One another option where to check how this works are Statistics. You should see some ICMP data being sent to 10.10.10.3 and some to 10.10.10.2. And there should be more Tx than Rx data for RipEX\_C (10.10.10.3), because it was turned off for a while.

RipEX2 RipEX_A @10.9.8.7	1 <sup>3</sup> 'I Remote access							S1	ATISTIC	cs /			
	Historical Diff	erential											
Init time: 021-07-19 08:52:43 (UTC+2)	Parameters												
<b>⊘</b> STATUS	Radio Interfac	Kabio interrace statistics     Radio protocol non-addressable statistics     Radio climat statistics											
SETTINGS	Radio signal n	on-addressable statistic	s										
CONTRACTION DIAGNOSTICS	Serial protoco	ls statistics stics											
Overview		14 07 40 07 50	2021	07 10 00	.50			1					
Events	Interval 202	1-07-19 07:52	2023	-07-19 08	:52		w >	1 nour					
Statistics	Display 🛓 Do	ownload Selected Data											
Monitoring	Data												
Routing													
Fools	Radio interface s	statistics											
Support	MAC address	IP address		UDP	(D)	ТСР	(0)	ICMP	(D)	ARP	101	Other	(0)
	D TOTAL	0.0.0.0	Rx	Count 0	0	104	29960	164	39688	8	336	291	[B] 30279
ADVANCED			Тх	0	0	127	24742	459	111078	1	42	201	20891
	00.00 0.00 05.04	10.10.10.4	Rx	0	0	0	0	0	0	0	0	11	1249
	00:02:a9:20:05:31		Тх	0	0	0	0	0	0	0	0	0	0
	00.02.29.20.07.00	10 10 10 2	Rx	0	0	22	5396	9 <mark>7</mark>	23474	2	84	146	
	00.02.47.20.07.07	10.10.10.2	Tv	0	0	29	5730	97	23474	0	0		15133
			17	0	0						0	4	15133 336
	00:02:a9:20:0a:e3	10.10.10.3	Rx	0	0	82	24564	67	16214	6	252	4 134	15133 336 13897
	00:02:a9:20:0a:e3	10.10.10.3	Rx Tx	0	0	82 98	24564 19012	67 362	16214 87604	6	252 42	4 134 16	15133 336 13897 1325
	00:02:a9:20:0a:e3 BROADCAST	10.10.10.3	Rx Tx Rx	0 0 0 0	0 0 0	82 98 0	24564 19012 0	67 362 0	16214 87604 0	6 1 0	252 42 0	4 134 16 0	15133 336 13897 1325 0

Fig. 3.9: RipEX\_A – Statistics (Radio interface statistics)

# 4. Radio channel and Ethernet combination



Fig. 4.1: Example 4 – Radio channel and Ethernet combination

## 4.1. Description

The fourth example requires you to have an additional one simple switch and two Ethernet cables. Interconnect RipEX\_A and RipEX\_B via switch – plug the cable into their ETH4 ports.



### Note

It is also possible to interconnect RipEX2 devices directly with a cable only. External switch is not mandatory.

Radio/RF channel cost stay the same for all units, set to 100. The Ethernet path is on 100 Mbps link and Babel setup will utilize a low cost equal to 10 for this link and a faster Babel routes propagation, because the RF channel capacity is not an issue on Ethernet.

This example can show you how to utilize Babel if you have RipEX2 units connected to some fast "backbone" and would like to use RipEX2 link only as a backup; or to route particular radio segment via particular part of the backbone.



## Note

As depicted, example 3 utilizes dual frequency solution from example 2.
## 4.2. RipEX\_A Configuration

Start with ETH4 configuration. Go to the SETTINGS – Interfaces – Ethernet menu. Select ETH4 panel and detach it from a current bridge. Create a new interface called "backbone" and assign 192.168.100.1/24 IP address to it.

RipEX2 RipEX_A @10.9.8.7	Remote access
Unit time: 2021-07-19 09:45:11 (UTC+2)	ETH1 • ETH2 • ETH3 • ETH4 • ETH5 •
	Status
STATUS	
SETTINGS	ETH4: • -   Attached   -   Router mode
Interfaces	Network interface
Ethernet	Name backbone bridges: ETH 4
Radio	
COM	Allow unit management
Terminal servers	
Routing	LAN
Firewall	IP / Mask 192.168.100.1/24 Note
VPN	+ Add IP/Subnet

Fig. 4.2: RipEX\_A – ETH4 configuration

The last menu is SETTINGS – Routing – Babel. Within the "Network" tab, add a new interface.

Edit interface		×
	Active	
Interface	if_backbone	
Туре	Wired	~
Rx cost	10	$\bigcirc$
Hello limit	12	$\bigcirc$
Hello interval [s]	4	$\bigcirc$
Update interval multiplier	4	$\bigcirc$
Advertised next hop	192.168.100.1	
Note		
Confirm and clo	se Close	

Fig. 4.3: RipEX\_A – Babel setup (ETH)

Non-default parameters:

Interface	if_backbone	
Туре	Wired (Ethernet is used, not the RF channel)	
Rx cost	10	
Advertised next hop 192.168.100.1 (our ETH4 IP address)		
Save the current configurat	ion.	

### 4.3. RipEX\_B Configuration

A very similar setup is to be done in RipEX\_B as well. Set the ETH4 IP address in a new "backbone" bridge to 192.168.100.2/24.



### Note

You can optimize IP usage with /30 mask for "backbone" interface for our example.

	EX_B@192.168.2.1 I <sup>0</sup> 1
Unit time: 2021-07-19 09:55:05 (UTC+2)	ETH1 • ETH2 • ETH3 • ETH4 • ETH5 •
	Status
Ø STATUS	
SETTINGS	ETH4: • -   Attached   -   Router mode
Interfaces	Network interface
Ethernet	Name backbone bridges: ETH 4
Radio	Allow unit management
Terminal servers	
Cellular	LAN
Routing	IP / Mask 192.168.100.2/24 Note
Firewall	+ Add IP/Subnet
VPN	

Fig. 4.4: RipEX\_B – ETH4 setup

#### Create a new Babel Network interface.

Edit interface ×		
1	Active	
Interface	if_backbone	
Туре	Wired	~
Rx cost	10	$\bigcirc$
Hello limit	12	$\bigcirc$
Hello interval [s]	4	$\sim$
Update interval multiplier 4		$\bigcirc$
Advertised next hop 192.168.100.2		
Note		
Confirm and close Close		

Fig. 4.5: RipEX\_B – Babel setup (ETH)

Save the changes.



### Important

Do not forget to increase the Cost in RipEX\_C (192.168.3.1) back to 100 for the radio interface.

The configuration should be complete, there is no need to change other settings neither in RipEX\_C, nor RipEX\_D.

### 4.4. Diagnostics and Testing

Babel Routing in RipEX\_A state can be similar to:

BABEL routing         Interfaces         BIRD 2.0.7 ready.         babel1:         Interface State RX cost Nbrs Timer Next hop (v4) Next hop (v6)         radio       Up       100       2       18.930       10.10.1       fe80::202:a9ff:fe20:6f9         if_backbone       Up       10       1       0.279       192.168.100.1       fe80::202:a9ff:fe20:6f9         Neighbors         BIRD 2.0.7 ready.         babel1:       IP       address       Interface Metric Routes Hellos Expires         fe80::202:a9ff:fe20:ae3       radio       100       3       5       26.797         fe80::202:a9ff:fe20:ae3       radio       100       2       4       22.828         fe80::202:a9ff:fe20:789       radio       100       2       4       22.828         fe80::202:a9ff:fe20:788       if_backbone       10       3       16       2.813         Routes         BIRD 2.0.7 ready.         babel1:       Prefix       Nexthop       Interface Metric F Seqno Expires         192.168.2.0/24       192.168.100.2       if_backbone       10*       1       50.810         192.168.2.0/24       10.10.10.3       radio
Interfaces         BIRD 2.0.7 ready.         babel1:         Interface State RX cost Nbrs Timer Next hop (v4) Next hop (v6)         radio       Up       100       2       18.930       10.10.10.1       fe80::202:a9ff:fe20:6f9         if_backbone Up       10       1       0.279       192.168.100.1       fe80::202:a9ff:fe20:6f8         Neighbors         BIRD 2.0.7 ready.       Dabel1:         IP address       Interface Metric Routes Hellos Expires         fe80::202:a9ff:fe20:a83       radio       100       3       5       26.797         fe80::202:a9ff:fe20:789       radio       100       2       4       22.828         fe80::202:a9ff:fe20:788       if_backbone       10       3       16       2.813         Routes       BIRD 2.0.7 ready.       babel1:       Prefix       Nexthop       Interface Metric F Seqno Expires         192.168.2.0/24       192.168.100.2       if_backbone       10 * 1       50.810         192.168.2.0/24       10.10.10.3       radio       100 * 1       36.659         192.168.3.0/24       10.10.10.3       radio       100 * 1       36.659         192.168.3.0/24       10.10.10.3       radio       100 * 1       36.659
fe80::202:a9ff:fe20:788       if_backbone       10       3       16       2.813         Routes         BIRD 2.0.7 ready.         babel1:       Prefix       Nexthop       Interface Metric F Seqno Expires         192.168.2.0/24       192.168.100.2       if_backbone       10 *       1 50.810         192.168.2.0/24       10.10.10.2       radio       100 +       1 327.850         192.168.2.0/24       10.10.10.3       radio       100 *       1 361.659         192.168.3.0/24       10.10.10.3       radio       100 *       1 361.659         192.168.3.0/24       192.168.100.2       if_backbone       120       1 50.810         192.168.4.0/24       192.168.100.2       if_backbone       110 *       1 50.810         192.168.4.0/24       192.168.100.2       if_backbone       110 *       1 50.810
Routes         BIRD 2.0.7 ready.         babel1:         Prefix       Nexthop         192.168.2.0/24       192.168.100.2         192.168.2.0/24       10.10.10.2         radio       100 +         192.168.2.0/24       10.10.10.3         radio       100 *         192.168.3.0/24       10.10.10.3         radio       100 *         192.168.3.0/24       192.168.100.2         if_backbone       10 *         192.168.4.0/24       192.168.100.2         if_backbone       110 *         192.168.4.0/24       192.168.100.2
BIRD 2.0.7 ready.         babel1:         Prefix       Nexthop         192.168.2.0/24       192.168.100.2         192.168.2.0/24       10.10.10.2         radio       100 +         192.168.2.0/24       10.10.10.3         radio       100 *         192.168.3.0/24       10.10.10.3         radio       100 *         192.168.3.0/24       192.168.100.2         if_backbone       120         192.168.3.0/24       192.168.100.2         if_backbone       110 *         192.168.4.0/24       192.168.100.2         if_backbone       110 *         192.168.4.0/24       192.168.100.2
babel1:         Nexthop         Interface Metric F Seqno Expires           192.168.2.0/24         192.168.100.2         if_backbone         10 *         1 50.810           192.168.2.0/24         10.10.10.2         radio         100 +         1 327.850           192.168.2.0/24         10.10.10.3         radio         210         1 361.659           192.168.3.0/24         10.10.10.3         radio         100 *         1 361.659           192.168.3.0/24         192.168.100.2         if_backbone         120         1 50.810           192.168.4.0/24         192.168.100.2         if_backbone         110 *         1 50.810
Prefix         Nexthop         Interface Metric F Seqno Expires           192.168.2.0/24         192.168.100.2         if_backbone         10 *         1 50.810           192.168.2.0/24         10.10.10.2         radio         100 +         1 327.850           192.168.2.0/24         10.10.10.3         radio         210         1 361.659           192.168.3.0/24         10.10.10.3         radio         100 *         1 361.659           192.168.3.0/24         192.168.100.2         if_backbone         120         1 50.810           192.168.4.0/24         192.168.100.2         if_backbone         110 *         1 50.810
192.168.2.0/24       192.168.100.2       if_backbone       10 *       1 50.810         192.168.2.0/24       10.10.10.2       radio       100 +       1 327.850         192.168.2.0/24       10.10.10.3       radio       210       1 361.659         192.168.3.0/24       10.10.10.3       radio       100 *       1 361.659         192.168.3.0/24       192.168.100.2       if_backbone       120       1 50.810         192.168.4.0/24       192.168.100.2       if_backbone       110 *       1 50.810         192.168.4.0/24       192.168.100.2       if_backbone       110 *       1 50.810
192.168.2.0/24       10.10.10.2       radio       100 +       1 327.850         192.168.2.0/24       10.10.10.3       radio       210       1 361.659         192.168.3.0/24       10.10.10.3       radio       100 *       1 361.659         192.168.3.0/24       10.10.10.3       radio       100 *       1 361.659         192.168.3.0/24       192.168.100.2       if_backbone       120       1 50.810         192.168.4.0/24       192.168.100.2       if_backbone       110 *       1 50.810
192.168.2.0/24       10.10.10.3       radio       210       1 361.659         192.168.3.0/24       10.10.10.3       radio       100 *       1 361.659         192.168.3.0/24       10.10.10.3       radio       100 *       1 361.659         192.168.3.0/24       192.168.100.2       if_backbone       120       1 50.810         192.168.4.0/24       192.168.100.2       if_backbone       110 *       1 50.810
192.168.3.0/24     10.10.10.3     radio     100 *     1 361.659       192.168.3.0/24     192.168.100.2     if_backbone     120     1 50.810       192.168.4.0/24     192.168.100.2     if_backbone     110 *     1 50.810
192.168.3.0/24         192.168.100.2         1+ backbone         120         1         50.810           192.168.4.0/24         192.168.100.2         if_backbone         110 *         1         50.810
192.168.4.0/24 192.168.100.2 1t_backbone 110 * 1 50.810
192.108.4.0/24 10.10.2 Faulo 200 + 1.52/.850
Entries
BTRD 2.0.7 ready.
babel1:
Prefix Router ID Metric Seano Routes Sources
192.168.1.0/24 00:00:00:01:01:01:01 0 1 0 0
192.168.2.0/24 00:00:00:02:02:02:02 10 1 3 1
192.168.3.0/24 00:00:00:00:03:03:03 100 1 2 1
192.168.4.0/24 00:00:00:04:04:04:04 110 1 3 1

#### Fig. 4.6: RipEX\_A – Babel routing

Our radio can "see" neighbor on two interfaces, with cost of 10 and 100 and it has three neighbors in total (two neighbors are of RipEX\_B).

To each remote subnet, we have three routes. The best one to 192.168.2.0/24 is, of course, via ETH with metric equal to just 10. We can get to 192.168.3.0/24 via one radio hop – metric is 100. And the last 192.168.4.0/24 subnet is accessible with metric equal to 110 (one hop ETH and one radio).

The last table shows four entries in Babel table – the first one is our RipEX\_A and then, three neighboring units with particular Metric/cost.

### 4.4.1. Tools and Monitoring

As a test, you can (this time) run an ICMP ping test from your laptop connected via ETH1, ETH2 or ETH3 to RipEX\_A. Ping a remote 192.168.4.1 RipEX2 ETH IP address. Do not forget to set a default route to 192.168.1.1 or set a static route to 192.168.4.0/24 via 192.168.1.1 on your laptop.

Your laptop should have any other IP address within 192.168.1.0/24 subnet, except .0 (network, .1 RipEX2 and .255 broadcast).

A command and its output from Windows CMD line can be similar to:

```
C:\Windows\system32>ping 192.168.4.1 -t
Pinging 192.168.4.1 with 32 bytes of data:
Reply from 192.168.4.1: bytes=32 time=61ms TTL=62
Reply from 192.168.4.1: bytes=32 time=48ms TTL=62
Reply from 192.168.4.1: bytes=32 time=42ms TTL=62
Reply from 192.168.4.1: bytes=32 time=48ms TTL=62
```

Verify that data are really sent via Ethernet. Run the Monitoring on ETH4 interface. Enable it, capture both Tx and Rx and limit the output to ICMP data only (All = "Off", ICMP = "On"). Length can be 0 Bytes, because it is not important for us now. You should see similar ICMP output:

#### Output

```
Show output
Show output
Show output
10:07:23.719498 (+969.695ms) [ETH4:phy:tx] IP 192.168.1.254 >
192.168.4.1 ICMP type 8, length:74
10:07:23.760089 (+40.591ms) [ETH4:phy:rx] IP 192.168.4.1 >
192.168.1.254 ICMP type 0, length:74
10:07:24.736638 (+976.549ms) [ETH4:phy:tx] IP 192.168.1.254 >
192.168.4.1 ICMP type 8, length:74
10:07:24.790207 (+53.569ms) [ETH4:phy:rx] IP 192.168.4.1 >
192.168.1.254 ICMP type 0, length:74
```

Fig. 4.7: RipEX\_A – ETH monitoring

At the same time, enable monitoring on the Radio interface as well. Limit the output to ICMP data only (All = "Off", ICMP = "On"). There should not be any data now. Unplug the ETH cable from RipEX\_A or RipEX\_B (the one attached to the switch) and check if data start going through the Radio interface.

15:11:57.168526 (+688.481ms) [RF:phy:tx] IP 192.168.1.254 >
192.168.4.1 ICMP type 8, length:78
15:11:57.248369 (+79.842ms) [RF:phy:rx] IP 192.168.4.1 > 192.168.1.254
ICMP type 0, length:78, rss:81 mse:24
15:11:58.184686 (+936.316ms) [RF:phy:tx] IP 192.168.1.254 >
192.168.4.1 ICMP type 8, length:78
15:11:58.270987 (+86.301ms) [RF:phy:rx] IP 192.168.4.1 > 192.168.1.254
ICMP type 0, length:78, rss:81 mse:29

#### Fig. 4.8: RipEX\_A – Radio monitoring

Possible Babel routing while an Ethernet cable is disconnected:

Neighbors			
BIRD 2.0.7 ready.			
babel1:			
IP address	Interface Metric Routes	Hellos Expires	
fe80::202:a9ff:fe20:ae3	radio 123 3	16 15.357	
fe80::202:a9ff:fe20:789	radio 114 3	16 41.390	
fe80::202:a9ff:fe20:788	if_backbone 65535	3 11 1.374	
Routes			
BIRD 2.0.7 ready.			
babel1:			
Prefix	Nexthop	Interface Metric F Seq	no Expires
192.168.2.0/24	192.168.100.2	if_backbone 65535 +	1 25.369
192.168.2.0/24	10.10.10.2	radio 114 *	1 406.411
192.168.2.0/24	10.10.10.3	radio 266	1 320.208
192.168.3.0/24	10.10.10.3	radio 123 *	6 320.208
192.168.3.0/24	192.168.100.2	if_backbone 65535	6 25.369
192.168.3.0/24	10.10.10.2	radio 328	6 406.411
192.168.4.0/24	192.168.100.2	if_backbone 65535 +	2 25.369
192.168.4.0/24	10.10.10.2	radio 228 +	2 406.411
192.168.4.0/24	10.10.10.3	radio 223 *	2 320.208

Fig. 4.9: RipEX\_A – Babel routing with Ethernet being disconnected

If not completely gone, metric for GRE/ETH routes is quickly 65535 and thus, one of the Radio paths is used.

In the ETH4 Statistics, you should see a lot of IPv6 (Babel) traffic.

IPv4 ot	her	IPv6	
count	[B]	count	[B]
1	46	14	1750
0	0	14	1280
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
936	92481	598	52061
991	96618	553	47687

Fig. 4.10: RipEX\_A – Babel data on ETH4 interface

# 5. Radio channel and Cellular (LTE) combination



Fig. 5.1: Example 5 – Radio channel and Cellular LTE combination

### 5.1. Description

The fifth example depicts a situation with two RipEX2 units equipped with LTE module. Both units are connected to the private APN and GRE tunnel is built between them to enable mutual communication over the cellular network.



### Note

Check MIDGE2 application notes and manual for more details about cellular networks and APN explanations.

This cellular path will be set with Rx cost equal to 50 to be better than RF channel, but worse than Ethernet connection. It can resemble Wired networks, but we will use Wireless. Choose whatever fits you more in your particular installation.

The example uses the third example configuration as a startup point.

### 5.2. RipEX\_B Configuration



#### Important

Keep in mind we set up RACOM private APN. Your APN setup and IPs will be different! Make appropriate changes so that your configuration reflects your APN details.

Enable the cellular interface.

RipEX_2 RipEX_A @192.168.1.1 RipEX_B@192.168.2.1		
Unit time: 2021-07-19 15:51:37 (UTC+2)	Common SIM1 SIM2	
	Status	
STATUS		
🕏 🖏 SETTINGS	Cellular Enabled	
Interfaces	Parameters	
Ethernet	SIM SIM1 ~	
Radio	Preferred service 4G (LTE) first v	
СОМ	Header compression Off v	
Terminal servers	Data compression Off v	
Cellular	MTU [B] 1500 C	
Routing	Masquerade On ~	
Firewall	Allow unit management	

Fig. 5.2: RipEX\_B – Cellular interface

You can keep everything in defaults, but set it as required by you and the cellular operator.

RipEX2 RipEX_A @192.16	8.1.1 RipEX_B@192.168.2.1 1 <sup>30</sup> 1	
Unit time: 2021-07-19 15:53:44 (UTC+2)	Common SIM1 S	ilM2
	SIM1	
Ø STATUS	PIN protection Off	~
🗘 🔅 SETTINGS	Network selection Au	tomatic 🗸 🗸
Interfaces	Access point name (APN)	ate
Ethernet	Authentication No	ne 🗸
Radio		
СОМ		
Terminal servers		
Cellular		

Configure your APN details in SIM1 panel.

#### Fig. 5.3: RipEX\_B - SIM1 setup

Create a GRE tunnel interface.

	×
Active	
10.203.0.29	
172.16.2.0/31	
1476 0	
Key enabled	
<ul> <li>Allow unit management</li> </ul>	
	]
Close	
	Active   10.203.0.29   172.16.2.0/31   1476   Key enabled   Allow unit   management

Fig. 5.4: RipEX\_B – GRE tunnel over cellular network

The Peer IP address is the IP of the RipEX\_D cellular interface. Tunnel address/mask is a new setup for GRE tunnel addresses.

The last step is configuring Babel Network interface again. Add a new interface.

Edit interface ×		
	Active	
Interface	gre_tun0	
Туре	Wireless ~	
Rx cost	50 0	
Hello interval [s]	10 0	
Update interval multiplier	4 0	
Advertised next hop	172.16.2.0	
Note	GRE over LTE	
Confirm and close Close		

Fig. 5.5: RipEX\_B – Babel interface (GRE over LTE)

Note the cellular interface is called "aux", but we need to define a GRE interface. Its name is "gre\_tun0". Change the Type to "Wireless" and set the Rx cost to 50. The Hello interval parameter is increased from 4 to 10. It is still faster than over RF channel (30 seconds), but not so fast as via Ethernet (4 seconds). Optimize this number to suit your needs and possible data usage of your SIM cards.

The Advertised next hop is its local GRE IP address.

Do not forget to set a new static route over cellular/aux network. A new route should either be a default one, or like with our example, we route there APN's subnet 10.203.0.0/17. Go to the SETTINGS – Routing – Static menu and add a new rule.

RipEX2 RipEX_B 10%1 Remote a	CCCESS STATIC
2021-09-22 13:29:54 (UTC+2)	Status
	Static routes
Ø STATUS	Destination IP / Mask 10.203.0.0/17 Mode WWAN (AUX)
🌣 SETTINGS	Gateway 0.0.0.0
Interfaces	+ Add
Routing	
Static	

Fig. 5.6: RipEX\_B – Static routing via cellular (aux) interface

Note that you can make this cellular/LTE routing rule permanent causing that if LTE is down, the rule is still active and traffic for LTE segment is not handled by another Routing rule (unreachable messages). Once LTE comes up again, the routing rule works as expected again.

To do so, check "Routing\_Persistent" parameter in this routing rule. Save the changes.

### 5.3. RipEX\_D Configuration

Common settings are the same. In your APN, you might have a different SIM1 setup compared to RipEX\_B (each SIM can have different Authentication credentials).

Once the cellular interface is ready, create a new GRE tunnel setup again.

Add	×
	Active
Peer address	10.203.0.28
Tunnel address / Tunnel mask	172.16.2.1/31
MTU	1476
	Key enabled
	Allow unit
Note	management
Confirm and close	e Close

Fig. 5.7: RipEX\_D – GRE tunnel setup

Peer address is the cellular IP of RipEX\_B; 172.16.2.1/31 is a new local GRE IP address.

Add a new Babel Network interface.

Edit interface		×
	Active	
Interface	gre_tun0	
Туре	Wireless	~
Rx cost	50	0
Hello interval [s]	10	0
Update interval multiplier	4	0
Advertised next hop	0.0.0.0	
Note	GRE over LTE	
Confirm and clo	ose Close	

Fig. 5.8: RipEX\_D – Babel interface (GRE over LTE)

Add a new static route via a cellular interface.

RipEX2 RipEX_A @192.168.1.1	RipEX_D@192.168.4.1 1 <sup>0</sup> 1 STATIC
Unit time: 2021-07-19 16:11:42 (UTC+2)	Status
STATUS	Static routes
🗘 🏟 SETTINGS	II Constitution IP / Mask 10.203.0.0/17 Mode WWAN (AUX) V Gateway 0.0.0
Interfaces	+ Add
Routing	
Static	

Fig. 5.9: RipEX\_D – Static routing via cellular (aux) interface

You can make the rule persistent again.

Save the changes.

### 5.4. Diagnostics and Testing

Check the Cellular/LTE status in SETTINGS – Interfaces – Cellular menu.

RipEX_2 RipEX_A @192.168.1.1 RipEX_B@192.168.2.1					
Unit time: 2021-07-19 16:19:34 (UTC+2)	Common SIM1	SIM2			
	Status				
Ø STATUS	Actual SIM	SIM1			
	SIM IMSI	230021200276879			
SETTINGS	SIM ID (ICCID)	8942020622802259004			
	SIM phone number	-			
Interfaces	PIN required	no			
Ethernet	Remaining PIN attempts	3			
	Operational status	up			
Radio	Registration status	registered (home network)			
СОМ	Network	0725			
	LAC/ IAC	114:05:			
Terminal servers	Dand	ITE Dand 3			
Cellular	Service type	ITE Daliu S			
	Signal	RSRP: -102 dBm			
Routing	Signal level	weak			
	Link up since	2021-07-19 16:18:59			
Firewall	IP address	10.203.0.28			
VPN	Module type	u-blox: MPCI-L210-03S-00			
	Module FW	15.63			
Security	Module IMEI	352255061921239			

Fig. 5.10: RipEX\_B – Cellular interface status

The status provides with a lot of information about current cellular connection.

Now, go to the Diagnostics and Tools menu. Try the mutual ping between 10.203.0.28 and 10.203.0.29. Also run a ping between 172.16.2.0 and 172.16.2.1 IP addresses.

Timeout [ms] 10000
Destination IP 10.203.0.29
🛍 Clear
203.0.28 : 200(228) bytes
ttl=63 time=848 ms
acket loss, time Oms 848.803/0.000 ms

Fig. 5.11: RipEX\_B – Pings over cellular/LTE

If these pings are not working correctly, check your APN setup and GRE tunnel configuration.

Go to the Diagnostics – Routing menu and check the Babel statistics.

Static	Dynamic	BABEL	OSPF	E	BGP		
BABEL routing							
Interfaces BIRD 2.0. babel1: Interface radio gre_tun0 Neighbors BIRD 2.0. babel1: IP. addres	7 ready. State RX co Up 1 Up 5 7 ready.	st Nbrs Time 30 2 0.30 50 1 5.91 Interface Me	r Next hop 2 10.10.10 9 172.16.2	) (v4) ).2 2.0	Next fe80: fe80:	hop (v6) :202:a9 :2025:d	) ff:fe20:789 33c:faf7:52a9
fe80::202 fe80::202 fe80::94a	s :a9ff:fe20:6f9 :a9ff:fe20:773 c:7df1:f421:20	radio radio 3a gre_tun0	100 50	3 1 3 1 3 1	16 35. 16 34. 16 9.	645 017 398	
Routes BIRD 2.0. babel1:	7 ready.						
Prefix 192.168.1 192.168.1	.0/24 .0/24	Nexthop 10.10.10.1 10.10.10.4		Inter radio radio	face M ) )	letric F 100 * 250	Seqno Expires 2 415.645 2 348.015
192.168.1 192.168.3 192.168.3	.0/24 .0/24 .0/24	172.16.2.1 10.10.10.1 10.10.10.4		gre_t radio radio	cun@ ) )	200 200 + 200 +	2 112.315 4 415.645 4 343.019
192.168.3 192.168.4 192.168.4	.0/24 .0/24 .0/24	172.16.2.1 10.10.10.4 10.10.10.1		gre_t radio radio	cun@ ) )	150 * 100 + 250	4 112.315 11 343.019 11 415.645
192.168.4 Entries BIRD 2.0.	.0/24 7 ready.	172.16.2.1		gre_t	tunØ	50 *	11 112.315
babel1: Prefix 192.168.1	.0/24	Router ID 00:00:00:00:00	:01:01:01	Metric 100	Seqno 2	Routes 3	Sources
192.168.2 192.168.3 192.168.4	.0/24 .0/24 .0/24	00:00:00:00:02 00:00:00:00:00:03 00:00:00:00:00:04	:02:02:02 :03:03:03 :04:04:04	0 150 50	1 4 11	0 3 3	0 1 1

#### Fig. 5.12: RipEX\_B – Babel Routing

There are already 3 routes to each destination in RipEX\_B radio. Two are using RF channel and one the GRE tunnel. Cost/Metric via LTE is set to "50" so you can see a route to 192.168.4.0/24 via "gre\_tun0" with a Metric equal to 50.

You can test the scenario by unplugging the cellular antennas or turning off RipEX\_B completely. Keep the ICMP ping running from your laptop to 192.168.4.1 IP address again.

Check Statistics and Monitoring menus. It is NOT possible to monitor cellular interface in RipEX2 firmware 2.0.5.0 and older. Also note that Monitoring via Remote access is only available since RipEX2 firmware 2.0.5.0.

Neighbors					
BIRD 2.0.7 ready.					
babel1:					
IP address	Interface	Metric	Routes	Hellos	Expires
fe80::202:a9ff:fe20:6f9	radio	100	3	16	40.940
fe80::202:a9ff:fe20:773	radio	114	3	16	39.314
fe80::94ac:7df1:f421:203a	gre_tun0	65535	0	3	4.534

Fig. 5.13: RipEX\_A – neighbors (cellular antenna removed from RipEX\_B)

You can also check if the cellular interface is "up" or "down" in the SETTINGS – Interfaces – Cellular menu.

RipEX2 RipEX_A (@192.168.1.1)	CELLULAR			
Unit time: 2021-07-20 09:08:04 (UTC+2)	Common SIM1	SIM2		
	Status			
STATUS	Actual SIM	SIM1		
🔅 SETTINGS	SIM IMSI SIM ID (ICCID) SIM phone number	230021200276879 8942020622802259004 		
Interfaces	PIN required	no		
Ethernet	Remaining PIN attempts Operational status	3 connecting		
Radio	Registration status	not registered		
СОМ	Network LAC/TAC	_		
Terminal servers	Cell	-		
Cellular	Band Service type	_		
D. 11	Signal	_		
Routing	Signal level	_		
Firewall	Link up since	_		
_	IP address Module type			
VPN	Module FW	15.63		
Security	Module IMEI	352255061921239		

Fig. 5.14: RipEX\_B – Cellular/LTE connection down



## 6. Basic Babel and OSPF combination

Fig. 6.1: Example 6 – Babel and OSPF diagram

### 6.1. Description

The sixth example shows a network combining Babel and OSPF. OSPF is configured in RipEX\_A and RipEX\_D units, whereas Babel is configured in RipEX\_A, B and C. These two protocols are neighboring in one Autonomous System Border Router "ASBR" (RipEX\_A) which divides the whole network into two parts. Bandwidth optimized Babel on the Radio segment and standardized, widely-used and fast OSPF on the Ethernet segment.

These two parts are interconnected together utilizing default gateways. E.g., Radio network is completely routed by Babel and all other routes are routed to ETH4 interface of RipEX\_A via a default gateway rule (0.0.0.0/0). In other words, there is no other export of routing rules.

Without dynamic protocols, all routes would need to be filled manually and statically.

Startup configurations are taken from first example. See the required changes below. It is required to change RipEX\_A and RipEX\_D settings only.

### 6.2. RipEX\_A Configuration

Add a 2<sup>nd</sup> IP/Mask to the "bridge" Ethernet interface. A new 192.168.100.1/24 IP address will be used for Ethernet connection to RipEX\_D and OSPF. You could also detach a particular ETH port and assign this IP to this designated port.

RipEX2 RipEX A (19)1 Remot	e access ETHERNET
Unit time: 2021-07-27 14:35:19 (UTC+2)	ETH1 • ETH2 • ETH3 • ETH4 • ETH5 •
O STATUS	Status
SETTINGS	ETH1: • -   Attached   -   Router mode
Interfaces	Network interface – Detach
Ethernet Radio	Name bridge bridges: ETH 1, ETH 2, ETH 3, ETH 4
	Allow unit management
Routing	LAN
Firewall	II V Mask 192.168.1.1/24 Note LAN
VPN	II V Mask 192.168.100.1/24 Note OSPF
Security	+ Add IP/Subnet

#### Fig. 6.2: RipEX\_A – Ethernet bridge

Add a new "default route" to current Babel Static rules. This default route will be used between OSPF and Babel network segments.

	e access BABEL
Unit time: 2021-07-27 14:36:38 (UTC+2)	Common Network Static rules Import filter Export filter
O STATUS	Static rules
SETTINGS	II     Destination IP / Destination mask     192.168.10/24       Metric     0     Note
Interfaces	Destination IP / Destination mask 0.0.0.0/0
Static BABEL	+ Add rule

Fig. 6.3: RipEX\_A – Babel Static rules (def. gateway)

Go to the OSPF Routing menu and enable OSPF dynamic protocol. The Router ID is shared among all dynamic protocols.

	emote access		OSPF		3 ≣ Changes
Unit time: 2021-07-27 14:37:29 (UTC+2)	Common	Network	Static rules	Import filter	Export filter
	Commo	n settings			
STATUS		Active			
🔅 🗱 SETTINGS	Router ID	1.1.1.1			
Interfaces	Instance ID	0	0		
Routing					
Static					
BABEL					
OSPF					

#### Fig. 6.4: RipEX\_A – OSPF activation

Create a new OSPF Area ID 0.0.0.0 (so called "backbone area" which has to be used in OSPF and all other areas must be directly connected to it).

RipEX2 RipEX_A 01 Remote	access		OSPF		<sup>4</sup> ≣ Changes
Unit time: 2021-07-27 14:39:37 (UTC+2)	Common	Network	Static rules	Import filter	Export filter
	Areas and i	interfaces	;		
	Area	ID 0.0.0.0		Stub area 🔽	Stub default GW
Interfaces	+ Add int	terface			
Routing					
Static	+ Add area				
BABEL					
OSPF	Neighbors				

Fig. 6.5: RipEX\_A – New OSPF backbone Area

Add a new interface under this backbone Area. Change the following parameters:

Interface if\_bridge

Network IP/Mask 192.168.100.0/24

Network type Broadcast

Multiple OSPF routers on a link on which all participants can hear each other. The link allows both broadcast and multicast for OSPF mechanism. All participants vote for one Designated Router (DR) and one Backup Designated Router (BDR) which are responsible for resending routing updates among other routers (to limit protocol overhead data).

Hello interval

Ethernet is a fast link so the interval can be low (i.e., fast OSPF processes).

Other parameters stay in default values.

2

Add interface	×
	Active
Interface	if_bridge
Network IP / Network mask	192.168.100.0/24
Network type	Broadcast ~
Cost	10 🗘
Hello interval [s]	2
Retransmit interval [s]	5
Dead count	4 0
TTL security	Off ~
Authentication	None ~
Password	
Priority	1 0
Use broadcast	Off ~
Note	
Confirm and clos	e Close

Fig. 6.6: RipEX\_A – New OSPF interface

Go to the Static rules tab and add two rules.

- 192.168.1.0/24
  - Metric type 2
  - Metric 1000
- 0.0.0.0/0
  - Metric type 2
  - $\circ$  Metric 1000

It is not important if the metric type is "1" or "2" in this example, but it is used to distinguish rules which came to OSPF from Babel (and cannot be exported back to Babel).

RipEX2 RipEXA 191 Remote a	OSPF	2  ☐ Changes
Unit time: 2021-07-27 14:40:42 (UTC+2)	Common Network Static rules Import filter	Export filter
STATUS	Static rules	
to SETTINGS	Image: Second dor in the second	
Routing Static	Destination IP / Destination mask 0.0.0.0/0	0
BABEL OSPF	+ Add rule	

Fig. 6.7: RipEX\_A – OSPF static rules

Go to the last required tab and add an Import filter with the following settings. Only the "Local preferred source address" is to be filled – with local ETH IP 192.168.1.1 so that packets generated in this RipEX2 unit use 192.168.1.1 as their Source IP.

Edit rule		×
	Active	
Filter network	Off	~
Filter source	Off	~
Filter OSPF tag	Off	~
Action	Accept	~
Set preference	Off	~
Local preferred source address	192.168.1.1	
Note		
Confirm and close	Close	

Fig. 6.8: RipEX\_A – OSPF Import filter

There is no change required in **RipEX\_B** and **RipEX\_C** units.

## 6.3. RipEX\_D Configuration

Add a 2<sup>nd</sup> IP/Mask to the "bridge" Ethernet interface. A new 192.168.100.2/24 IP address will be used for Ethernet connection to RipEX\_A and OSPF. You could also detach a particular ETH port and assign this IP to this designated port.

RipEX2 RipEX_A @10.9.8.7 @192.168.4.1	<sup>19</sup> <sup>1</sup> ETHERNET <sup>1</sup> ≣ Change
Unit time: 2021-07-27 14:46:12 (UTC+2)	ETH1 • ETH2 • ETH3 • ETH4 • ETH5 •
	Status
Ø STATUS	
🔅 settings	ETH1: • -   Attached   -   Router mode
Interfaces	Network interface
Ethernet	Name bridge bridges: ETH 1, ETH 2, ETH 3, ETH 4
Radio	
СОМ	Allow unit management
Terminal servers	
Routing	LAN
Firewall	II V IP / Mask 192.168.4.1/24 Note LAN
	IP / Mask 192.168.100.2/24 Note OSPF

Fig. 6.9: RipEX\_D – Ethernet configuration

Disable the Radio protocol so there is no Radio/RF communication, only Ethernet connection to RipEX\_A.

RipEX2 RipEX_A @102.168.4.1	t <sup>io</sup> t RADIO
Unit time: 2021-07-27 14:46:42 (UTC+2)	Status
STATUS	Radio protocol
SETTINGS	Radio protocol None 🗸
Interfaces	Management
Ethernet	Allow unit management On 🗸
Radio	

Fig. 6.10: RipEX\_D – Radio communication disabled

Disable Babel protocol.

RipEX2 RipEX_A @10.9.8.7 RipEX_D @192.168	.4.1 <sup>191</sup> 1		BABEL		I Changes
		_		-	
Unit time: 2021-07-27 14:47:25 (UTC+2)	Common	Network	Static rules	Import filter	Export filter
	Commor	n settings			
STATUS		Active	3		
🕸 settings	Router	ID 4.4.4.4			
Interfaces	Routing offer	ng On	~		
Routing					
Static					
BABEL					

Fig. 6.11: RipEX\_D – Babel protocol disabled

Enable OSPF with the same Router ID 4.4.4.4.

RipEX2 RipEX_A RipEX_A @10.9.8.7	D 168.4.1 I <sup>01</sup> I		OSPF		4
				-	
Unit time: 2021-07-27 14:47:51 (UTC+2)	Common	Network	Static rules	Import filter	Export filter
	Commo	n settings			
Ø STATUS	1	Active			
🔅 SETTINGS	Router ID	4.4.4.4			
Interfaces	Instance ID	0	0		
Routing					
Static					
BABEL					
OSPF					

### Fig. 6.12: RipEX\_D – OSPF protocol enabled

Add one OSPF interface within a new backbone Area ID 0.0.0.0.

Interface	if_bridge
Network IP/Mask	192.168.100.0/24
Network type	Broadcast
Hello interval	2

RipEX 2 RipEX_A @10.9.8.7 RipEX_D @192.168.4.1	1941	OSPF	7 III Changes
Unit time: 2021-07-27 14:48:55 (UTC+2)	Common Network	Static rules Import	t filter Export filter
STATUS	Areas and interfaces	Chub a	rea Etub dafault CM
🕨 🏟 SETTINGS	II Note		Tea Stub default Gvv
Interfaces	Interface if bridge	Network	type Broadcast V
Routing	Edit Interface Configur	ration Note	
Static	1. Add interface		
BABEL	+ Add Interface		
OSPF			

Fig. 6.13: RipEX\_D – New OSPF Area and Interface

#### Details:



Fig. 6.14: RipEX\_D – OSPF interface details

Add one Static rule so that local 192.168.4.0/24 is propagated through OSPF protocol. Set the same Metric type (2) and Metric (1000) as in RipEX\_A.

RipEX2 RipEX_A @10.9.8.7 RipEX_D @192.168.4.1	t <sup>®</sup> ा OSPF <sup>®</sup> ≣ Changes
Unit time: 2021-07-27 14:50:08 (UTC+2)	Common Network Static rules Import filter Export filter
	Static rules
C STATUS	Destination IP / Destination mask 192.168.4.0/24
Interfaces	Metric type 1    OSPF tag      O      Note
Routing	+ Add rule
Static	
BABEL	
OSPF	

Fig. 6.15: RipEX\_D – OSPF Static rules

Finally, add one Import filter with Local preferred source address equal to 192.168.4.1.

Add rule		×
	Active	
Filter network	Off	~
Filter source	Off	~
Filter OSPF tag	Off	~
Action	Accept	~
Set preference	Off	~
Local preferred source address	192.168.4.1	
Note		
Confirm and close	Close	

Fig. 6.16: RipEX\_D – OSPF Import filter

### 6.4. Diagnostics and Testing

You can attach your laptops to RipEX2 units and do some PING tests. You can also configure some SCADA protocol traffic simulation.

The "issue" with testing this scenario on the desk is that each radio can hear each other (except RipEX\_D). Thus, it is not possible to force Babel and Radio channel to send traffic via a repeater by default and only if this fails, sending data directly. The traffic always uses a direct connection. Only if you had some attenuators or you use it in the field, there should be situations in which traffic goes via two radio hops.

Also, the OSPF and Babel are neighboring in one ASBR router (RipEX\_A) – so if you turn off RipEX\_A, then RipEX\_D loses its connection to the radio segment (and vice versa). There is no other way.

So, check the correct routing is set and used in all routers and also check the ping works among all units. Typical outputs you should see are:

RipEX2 RipEX_A @10.9.8.7	<sup>ມ</sup> າ Remote access				
Unit time: 2021-07-28 07:57:46 (UTC+2)	Static Dynamic BABEL OSPF BGP				
<ul><li>STATUS</li><li>SETTINGS</li></ul>	System routing System routing table unreachable default proto bird metric 32 10.9.8.0/28 dev service proto kernel scope link src 10.9.8.7 10.10.10.0/24 dev radio proto kernel scope link src 10.10.10.1				
DIAGNOSTICS Overview Events Statistics	192.0.2.232/30 dev agl0 proto kernel scope link src 192.0.2.233 192.168.1.0/24 dev if_bridge proto kernel scope link src 192.168.1.1 unreachable 192.168.1.0/24 proto bird metric 32 192.168.2.0/24 via 10.10.10.2 dev radio proto bird src 192.168.1.1 metric 32 192.168.3.0/24 via 10.10.10.3 dev radio proto bird src 192.168.1.1 metric 32 192.168.4.0/24 via 192.168.100.2 dev if_bridge proto bird src 192.168.1.1 metric 32 192.168.100.0/24 dev if_bridge proto kernel scope link src 192.168.1.1 metric 32				
Monitoring Routing					

Fig. 6.17: RipEX\_A – System routing

RipEX\_A should have /24 routes to all LANs behind every other RipEX2 in this example.

	<b>RipEX2</b> RipEX_A @10.9.8.7 ا <sup>ψ</sup>	Remote access				
_	Unit time: 2021-07-28 07:59:07 (UTC+2)	Static Dynamic	BABEL OSPF	BGP		
		BABEL routing				
	Ø STATUS	Interfaces				
	♣, SETTINGS	BIRD 2.0.7 ready. babel1: Interface State RX cos radio Up 10	t Nbrs Timer Next hop 0 2 11.260 10.10.10	) (v4) Ne: 0.1 fe	kt hop (v6 80::202:a9	) ff:fe20:6f9
		Neighbors				
	Overview	BIRD 2.0.7 ready. babel1: IP address fe80::202:a9ff:fe20:ae3	Interface Metric Route	es Hellos E	xpires	
	Events	fe80::202:a9ff:fe20:789	radio 100	2 16	26.326	
	Statistics	Routes BIRD 2.0.7 ready. babel1:				
	Monitoring	Prefix 192.168.2.0/24	Nexthop 10.10.10.2	Interfac radio	e Metric F 100 *	Seqno Expires
	Routing	192.168.2.0/24	10.10.10.3	radio	200	1 322.309
	Routing	192.168.3.0/24	10.10.10.3	radio	100 *	1 322.309
	Tools	Entries	10.10.10.2	1 8010	200	1 5/4.521
	Support	BIRD 2.0.7 ready. babel1:			-	-
		0.0.0/0	ROUTER ID 00:00:00:00:01:01:01:01	Metric Seq 0	1 0	8 ources
	ADVANCED	192.168.1.0/24	00:00:00:00:01:01:01:01	0	1 0	0
		192.168.2.0/24	00:00:00:00:02:02:02:02	100	1 2	1
		192.168.3.0/24	00:00:00:00:03:03:03:03	100	1 2	1

Fig. 6.18: RipEX\_A – Babel routing

In the Babel output, you should see two neighbors and a direct metric to each with a cost of 100.

RipEX2 RipEX_A @10.9.8.7 Ⅰ <sup>№</sup> 1	Remote access
Unit time: 2021-07-28 08:00:40 (UTC+2)	Static Dynamic BABEL OSPF BGP
	OSPF routing
Ø STATUS	Neighbors
🏟 SETTINGS	BIRD 2.0.7 ready. ospf1: Router ID Pri State DTime Interface Router IP
Contraction of the second seco	4.4.4.4 1 Full/DR 7.766 if_bridge 192.168.100.2 State & Topology BIRD 2.0.7 ready.
Overview	area 0.0.0.0
Events	router 1.1.1.1
Statistics	network 192.168.100.0/24 metric 10 external 0.0.0.0/0 metric2 1000
Monitoring	external 192.168.1.0/24 metric2 1000
Routing	router 4.4.4.4 distance 10
Tools	network 192.168.100.0/24 metric 10 external 192.168.4.0/24 metric2 1000
Support	network 192.168.100.0/24 dr 4.4.4.4
## ADVANCED	distance 10 router 4.4.4.4 router 1.1.1.1

Fig. 6.19: RipEX\_A – OSPF output

There should be one 4.4.4.4 neighbor with state either Full/DR or Full/BDR.

	RipEX2 RipEX_A @10.9.8.7	RipEX_B@192.168.2.1 1 <sup>0</sup> 1
	Unit time: 2021-07-28 08:13:00 (UTC+2)	Static Dynamic BABEL OSPF BGP
		BABEL routing
	Ø STATUS	Interfaces
	🏟 SETTINGS	BIRD 2.0.7 ready. babel1: Interface State RX cost Nbrs Timer Next hop (v4) Next hop (v6) radio Up 100 2 13.478 10.10.10.2 fe80::202:a9ff:fe20:789
)		Neighbors
	Overview	BIRD 2.0.7 ready. babel1: IP address Interface Metric Routes Hellos Expires fe80::202:a9ff:fe20:ae3 radio 100 3 16 16.918
	Events	Routes
		BIRD 2.0.7 ready. babel1:
	Monitoring	Prefix Nexthop Interface Metric F Seqno Expires
		0.0.0.0/0 10.10.10.1 radio 200 1 324.918
1	Routing	192.168.1.0/24 10.10.10.1 radio 100 * 1 411.205
	Taala	192.168.1.0/24 10.10.10.3 radio 200 1 324.918
	IDOIS	192.168.3.0/24 10.10.10.3 radio 100 * 1 324.918
	Support	192.168.3.0/24 10.10.10.1 radio 200 1 411.205
	Support	Entries
1		babel1:
	ADVANCED	Prefix Router ID Metric Segno Routes Sources
		0.0.0/0 00:00:00:01:01:01 100 1 2 1
1		192.168.1.0/24 00:00:00:01:01:01:01 100 1 2 1
		192.168.2.0/24 00:00:00:02:02:02:02 0 1 0 0
		192.168.3.0/24 00:00:00:03:03:03 100 1 2 1

Fig. 6.20: RipEX\_B – Babel routing

RipEX\_B should have routes to its Radio neighbors (192.168.1.0/24 and 192.168.3.0/24) with Metric equal to 100 and one default gateway rule back to RipEX\_A (also with a Metric equal to 100). The same is for RipEX\_C.

	RipEX_D@192.168.4.1
Unit time: 2021-07-28 08:24:04 (UTC+2)	Static Dynamic BABEL OSPF BGP
	System routing
STATUS	System routing table
🏟 SETTINGS	<pre>default via 192.168.100.1 dev if_bridge proto bird src 192.168.4.1 metric 32 10.10.00/24 dev radio proto kernel scope link src 10.10.10.4 192.0.2.232/30 dev agl0 proto kernel scope link src 192.0.2.233 102.101.101.101.101.101.101.101.101.101.</pre>
V DIAGNOSTICS	192.168.1.0/24 Via 192.168.100.1 dev 11_0ridge proto bird src 192.168.4.1 metric 32 192.168.4.0/24 dev if_bridge proto kernel scope link src 192.168.4.1 unreachable 192.168.4.0/24 proto bird metric 32
Overview	192.168.100.0/24 dev ir_bridge proto kernel scope link src 192.168.100.2 192.168.100.0/24 dev if_bridge proto bird scope link src 192.168.4.1 metric 32
Events	
Statistics	
Monitoring	
Routing	

Fig. 6.21: RipEX\_D – System routing

RipEX\_D should have a /24 link to 192.168.1.0 subnet via its if\_bridge interface and gateway 192.168.100.1 (RipEX\_A) and a default gateway (also 192.168.100.1) for the whole radio segment.

RipEX2 RipEX\_A @10.9.8.7 RipEX\_D@192.168.4.1 Unit time: OSPF Static Dynamic BABEL BGP 2021-07-28 08:29:46 (UTC+2) **OSPF** routing O STATUS Neighbors BIRD 2.0.7 ready. ospf1: 🏟 SETTINGS Pri 1 Router ID State DTime Interface Router IP 1.1.1.1 Full/BDR 7.591 if\_bridge 192.168.100.1 DIAGNOSTICS State & Topology BIRD 2.0.7 ready. Overview area 0.0.0.0 Events router 1.1.1.1 distance 10 network 192.168.100.0/24 metric 10 Statistics external 0.0.0.0/0 metric2 1000 external 192.168.1.0/24 metric2 1000 Monitoring router 4.4.4.4 Routing distance 0 network 192.168.100.0/24 metric 10 external 192.168.4.0/24 metric2 1000 Tools network 192.168.100.0/24 Support dr 4.4.4.4 distance 10 router 4.4.4.4 ADVANCED router 1.1.1.1

All the routes are obtained via OSPF. There is no Babel configured.

Fig. 6.22: RipEX\_D – OSPF routing

# 7. Advanced Babel and OSPF combination



Fig. 7.1: Example 7 Babel and OSPF diagram 3

### 7.1. Description

The seventh example extends the sixth one to show even more from OSPF and Babel combination. There are two ASBR routers between Babel and OSPF network segments. By default, the primary path should go over RipEX\_A and a backup path is over RipEX\_C.

We set the static preferences of routes to be propagated so that RipEX\_A is a preferred ASBR. Routing rules are being exchanged between mentioned segments, not just default gateways.

This example continues from the sixth example and only changes are explained below.
# 7.2. RipEX\_A Configuration

First of all, let's divide one common bridge interface on all Ethernet ports. This is not mandatory, but shows a different approach compared to the previous example, i.e., one IP address/subnet on each individual ETH port.

To do so, go to the SETTINGS – Interfaces – Ethernet menu. Delete the 2<sup>nd</sup> subnet from the primary bridge interface.



Fig. 7.2: RipEX\_A – ETH1 configuration

Go to the ETH4 tab. Detach it from the current bridge and add a new interface called "ospf". Add IP/subnet equal to 192.168.100.1/24.

RipEX2 RipEX_A @192.168.1.1	1 <sup>0</sup> 1 Remote access
Unit time: 2021-07-28 13:20:52 (UTC+2)	ETH1 • ETH2 • ETH3 • ETH4 • ETH5 •
	Status
STATUS	
🗴 🎝 SETTINGS	ETH4: • -   Attached   -   Router mode
Interfaces	Network interface
Ethernet	Name ospf bridges: ETH 4
Radio	
СОМ	Allow unit management
Terminal servers	
Routing	LAN
Firewall	IP / Mask         192.168.100.1/24         Note
VPN	+ Add IP/Subnet

Fig. 7.3: RipEX\_A – ETH4 configuration

We need to change OSPF setup now. Go to the Routing – OSPF menu and select a Network tab. Change the Interface name to "if\_ospf" to reflect ETH4 change and a new interface name.

RipEX2 RipEX_A @192.168.1	.1 Ι <sup>ψ</sup> Ι Remote access
Unit time: 2021-07-28 13:21:08 (UTC+2)	Common Network Static rules Import filter Export filter
	Areas and interfaces
O STATUS	Area ID 0.000 Stub area Stub default GW Note
SETTINGS	
Interfaces	Interface if_ospf Network type Broadcast v Edit Interface Configuration
Routing	+ Add interface
Static	
BABEL	+ Add area
OSPF	



Change the tab to "Static rules". We only keep one rule 192.168.1.0/24. Delete the default route rule, because particular routes will be propagated, not a default gateway.

RipEX2 RipEX A @192.168.1.1	1 <sup>0</sup> 1 Remote access	OSPF	🛛 🗮 Changes 🛛 ៧ Notifici	ations
	^			
Unit time: 2021-07-28 14:33:48	Common Network	Static rules Import filter	Export filter	
(UTC+2)	Static rules			<b>∷</b> ≡
O STATUS	Destination IP / D	estination mask 192.168.1.0/24	Metric type Type 2	
🔅 🔅 SETTINGS	Metric 1000	OSPF tag 0	○ Note	
Interfaces	+ Add rule			
Routing				
Static				
BABEL				
OSPF				

Fig. 7.5: RipEX\_A – OSPF Static rules

Go to the "Import filter" tab and create two rules. The first one is to import external routes (EXT2) with preference 150 (lower/worse than Babel which has 210 by default). Set the Local preferred source address to local ETH1 IP 192.168.1.1.

These are the rules which were imported from Babel to OSPF in a different ASBR. We could also use OSPF tag filter instead of source filter.

Edit rule		×
1	Active	
Filter network	Off	*
Filter source	Match	*
Source	External type 2	*
Filter OSPF tag	Off	*
Action	Accept	*
Set preference	On	*
Preference	150	0
Local preferred source address	192.168.1.1	
Note		
Confirm and close	Close	

Fig. 7.6: RipEX\_A – Import filter, 1<sup>st</sup> rule

Import all other rules and set a preference to 250 (higher/better than Babel). Make sure the order of rules is met, **the order is important**!

Edit rule		×
	Active	
Filter network	Off	~
Filter source	Off	~
Filter OSPF tag	Off	~
Action	Accept	~
Set preference	On	~
Preference	250	$\bigcirc$
Local preferred source address	192.168.1.1	
Note		
Confirm and close	Close	

Fig. 7.7: RipEX\_A – Import filter, 2<sup>nd</sup> rule

We also need to set Export filter. OSPF export filter rules define set of routing rules to be exported from the unit into the OSPF area.

Due to this rule, OSPF export rules from Babel with Metric type 2 (EXT2) and Metric equal to 1000 (RipEX\_A is a preferred ASBR/router).

Edit rule		×
1	Active	
Filter network	Off	~
Filter protocol	Match	~
Protocol	Babel	~
Action	Accept	*
Metric type	Type 2	*
Metric	1000	0
OspfExFRule_SetMetricFromOtherProt	Off	*
Set OSPF tag	Off	*
Note		
Confirm and close	Close	

#### Fig. 7.8: RipEX\_A – Export filter

Now, select the Routing – Babel menu. First of all, keep only one Static rule (192.168.1.0/24, metric 0).

RipEX2 RipEX_D @192.168.4.1 (#192.168.3	1.1 <sup>(3)</sup> T	BABEL		≣ C
Unit time: 2021-07-29 09:47:57 (UTC+2)	Common Network	Static rules	Import filter	Export 1
STATUS	Static rules	Destination mask	192 168 1 0/24	
SETTINGS	Metric 0		lote	
Interfaces	+ Add rule			
Routing				
Static				
BABEL				



The last step in RipEX\_A setup is to create a new Babel Export filter rule.

This rule exports rules from OSPF with EXT1 type and sets a metric to 1000 (preferred ASBR/router, higher number than any sum of metrics in Babel network/segment).



# Note

EXT2 rules mark rules originally from Babel network and they are not exported back. EXT1 rules mark rules from OSPF network which are forwarded to Babel. It is also possible to use OSPF tag filter – marking rules from Babel which are exported to OSPF with this tag.

Add rule		×
	Enable rule	
Filter network	Off	*
Filter protocol	Match	~
Protocol	OSPF	~
Filter OSPF source	Match	~
OSPF source	External type 1	*
Filter OSPF tag	Off	~
Action	Accept	*
Metric from other protocol	Off	~
Metric	1000	0
Note		
Confirm and close	se Close	

Fig. 7.10: RipEX\_A – Babel export rule

Save all the changes. RipEX\_B does not require any change, continue with RipEX\_C. If the remote access is not possible currently, we recommend using the management USB/ETH access.

# 7.3. RipEX\_C Configuration

RipEX\_C is a 2<sup>nd</sup> ASBR (router) on the OSPF/Babel border and is used as a backup OSPF router. Most of the required configuration steps are the same as in RipEX\_A, but we make sure to set various metrics/preferences worse (so RipEX\_A is a primary ASBR).

Start with detaching ETH4 port and setting its new name "ospf" and IP/Mask equal to 192.168.200.1/24. Do not forget to delete this IP from ETH1 settings.



Fig. 7.11: RipEX\_C – ETH4 configuration

Go to the Routing – OSPF menu and activate its functionality. The Router ID is shared among all dynamic protocols so keep it 3.3.3.3.

RipEX2 RipEX_A @192.168	1 RipEX_C@192.168.3.1 ( <sup>0</sup> )
Unit time: 2021-07-28 13:58:08 (UTC+2)	Common Network Static rules Import filter Export filter
	Common settings
STATUS	Active
SETTINGS	Router ID 3.3.3.3
Interfaces	Instance ID 0
Routing	
Static	
BABEL	
OSPF	

Fig. 7.12: RipEX\_C – OSPF activation

Go to next tab "Network" and create a new backbone area 0.0.0.0 with the following new interface. The interface must be named per the ETH4 name, so "if\_ospf". Set the correct network on this port 192.168.200.0/24 and decrease the Hello interval to 2 seconds, because Ethernet is a fast interface. Other parameters can stay in default values (e.g., "broadcast" type, or cost equaling 10).

Add interface	×
	Active
Interface	if_ospf
Network IP / Network mask	192.168.200.0/24
Network type	Broadcast 🗸
Cost	10 0
Hello interval [s]	2
Retransmit interval [s]	5 0
Dead count	4 0
TTL security	Off 🗸
Authentication	None 🗸
Password	
Priority	1
Use broadcast	Off 🗸
Note	
Confirm and clos	e Close

Fig. 7.13: RipEX\_C – New OSPF interface under backbone 0.0.0.0 area

After the changes, the setup should look like this:

RipEX2 RipEX_A @192.168.1.1	RipEX_C@192.168.3.1
Unit time: 2021-07-28 13:59:18 (UTC+2)	Common Network Static rules Import filter Export filter
STATUS	Areas and interfaces
🗢 🚓 SETTINGS	Area ID 00000 Stub area V Stub derauit GW Note
Interfaces	Interface if_ospf     Network type     Broadcast     Configuration
Routing	+ Add interface
Static	
BABEL	+ Add area
OSPF	

Fig. 7.14: RipEX\_C – OSPF network

Configure "Static rules" with the local LAN IP subnet. Set the Metric type to "2" (EXT2) and its metric to 1000. We do not use "OSPF tag" parameter.

EXT2 is used because unit's LAN is considered to be part of the Babel network.

	EX C 92.168.3.1	OSP	F		10 E Changes	M No
Unit time: 2021-07-28 14:32:52	Common Netwo	ork Static rules	Import filter	Export filter		
(UTC+2)	Static rules					
② STATUS	Destinatio	IP / Destination mask	92.168.3.0/24	Metric type	Type 2	<b>~</b>
SETTINGS	Metric 1	00 01 01	SPF tag 0	0 No	te	
Interfaces	+ Add rule					
Routing						
Static						
BABEL						
OSPF						

Fig. 7.15: RipEX\_C – OSPF Static rules

We need to configure two Import filter rules, the same way as in RipEX\_A.

The first one for rules matching EXT2 and setting the Preference to 150. Keep the order of the rules.

Edit rule		×
	Active	
Filter network	Off	*
Filter source	Match	*
Source	External type 2	*
Filter OSPF tag	Off	*
Action	Accept	~
Set preference	On	*
Preference	150	0
Local preferred source address	192.168.3.1	
Note		
Confirm and close	Close	

Fig. 7.16: RipEX\_C – 1<sup>st</sup> OSPF Import filter rule

A 2<sup>nd</sup> rule is for other traffic and setting the Preference to 250 (higher/better than Babel).

Edit rule		×
	Active	
Filter network	Off	~
Filter source	Off	~
Filter OSPF tag	Off	~
Action	Accept	~
Set preference	On	~
Preference	250	0
Local preferred source address	192.168.3.1	
Note		
Confirm and close	Close	

Fig. 7.17: RipEX\_C – 2<sup>nd</sup> OSPF Import filter rule

And the last OSPF setting is Export filter rule. It is the same as in RipEX\_A, but we set the Metric to be 2000 so this ASBR/Router is considered as a backup one (RipEX\_A is a preferred ASBR).

Add rule		×
	Active	
Filter network	Off	*
Filter protocol	Match	*
Protocol	Babel	*
Action	Accept	*
Metric type	Type 2	*
Metric	2000	\$
OspfExFRule_SetMetricFromOtherProt	Off	*
Set OSPF tag	Off	*
Note		
Confirm and close	Close	

Fig. 7.18: RipEX\_C – OSPF external filter rule

Go to the Routing – Babel menu and select an Export filter tab. Add a new rule which exports rules from OSPF with EXT1 type and sets a metric to 2000 (backup router, higher than any sum of metrics in Babel network).

Edit rule		×
	Enable rule	
Filter network	Off	~
Filter protocol	Match	*
Protocol	OSPF	~
Filter OSPF source	Match	*
OSPF source	External type 1	~
Filter OSPF tag	Off	*
Action	Accept	~
Metric from other protocol	Off	~
Metric	2000	÷
Note		
Confirm and clo	se Close	

Fig. 7.19: RipEX\_C – Babel Export filter rule

Save the changes and configure the last unit – RipEX\_D.

## 7.4. RipEX\_D Configuration

Again, we need to start in Interfaces – Ethernet menu. Configure three network interfaces.

ETH1 + ETH2 bridge - 192.168.4.1/24 ETH3 ospf1 - 192.168.100.2/24

ETH4 ospf2 - 192.168.200.2/24

If not yet done, connect ETH3 port to RipEX\_A ETH4; and connect ETH4 to RipEX\_C ETH4.

RipEX2 RipEX_A @192.168.1.1	RipEX_D@192.168.4.1 [ <sup>0</sup> ]
Unit time: 2021-07-28 13:08:52 (UTC+2)	ETH1 • ETH2 • ETH3 • ETH4 • ETH5 •
	Status
★ SETTINGS	ETH1: • -   Attached   -   Router mode
Interfaces	Network interface
Ethernet	Name bridge bridges: ETH 1, ETH 2, ETH 3
Radio	
СОМ	Allow unit management
Terminal servers	
Routing	LAN
Firewall	V IP / Mask 192.168.4.1/24 Note LAN

Fig. 7.20: RipEX\_D – ETH1 configuration

RipEX2 RipEX_A @192.168.1.1	RipEX_D@192.168.4.1 1 <sup>9</sup> 1
Unit time: 2021-07-28 13:54:25 (UTC+2)	ETH1 • ETH2 • ETH3 • ETH4 • ETH5 •
	Status
O STATUS	
🔅 SETTINGS	ETH3: • -   Attached   -   Router mode
Interfaces	Network interface
Ethernet	Name ospf1 bridges: ETH 3
Radio	
СОМ	Allow unit management
Terminal servers	
Routing	LAN
Firewall	✓ IP / Mask 192.168.100.2/24 Note

Fig. 7.21: RipEX\_D – ETH3 "ospf1" interface

RipEX2 RipEX_A @192.168.1.1	RipEX_D@192.168.4.1 1 <sup>0</sup> 1
Unit time	
2021-07-28 13:54:51 (UTC+2)	ETH1 • ETH2 • ETH3 • ETH4 • ETH5 •
	Status
Ø STATUS	ETHA: • - LAttachad L - Douter mode
SETTINGS	
Interfaces	Network interface
Ethernet	Name ospf2 bridges: ETH 4
Radio	
СОМ	Allow unit management
Terminal servers	
Routing	LAN
Firewall	P / Mask 192.168.200.2/24 Note

Fig. 7.22: RipEX\_D – ETH4 "ospf2" interface

Go to the Routing – OSPF menu and go to the Network tab. Edit the 1<sup>st</sup> interface and add a 2<sup>nd</sup> interface. See the non-default values.

if\_ospf1

•

Туре	broadcast
Network IP/mask	192.168.100.0/24
Hello interval [s]	2
if_ospf2	
Туре	broadcast
Network IP/mask	192.168.200.0/24
Hello interval [s]	2

RipEX2 RipEX\_A @192.168.1.1 RipEX\_D@192.168.4.1 Unit time: 2021-07-28 13:55:44 (UTC+2) Network Static rules Import filter Export filter Common Areas and interfaces O STATUS Area ID 0.0.0.0 🗌 Stub area 🔽 Stub default GW 🛛 Note ♦ SETTINGS ✓ Interface if\_ospf1 Network type Broadcast Edit Interface Configuration ~ Interfaces ✓ Interface if\_ospf2 Network type Broadcast Edit Interface Configuration ¥ Routing + Add interface Static BABEL OSPF + Add area

Fig. 7.23: RipEX\_D – OSPF Network configuration

Edit interface		×
(	Active	
Interface	if_ospf1	
Network IP / Network mask	192.168.100.0/24	
Network type	Broadcast	~
Cost	10	0
Hello interval [s]	2	0
Retransmit interval [s]	5	0
Dead count	4	0
TTL security	Off	~
Authentication	None	~
Password		
Priority	1	0
Use broadcast	Off	~
Note		
Confirm and clos	close	

Fig. 7.24: RipEX\_D – OSPF interface (if\_ospf1)



Fig. 7.25: RipEX\_D – OSPF interface (if\_ospf2)

Change the tab to "Static rules" and make sure there is just one rule for 192.168.4.0/24, Type 1, Metric 1000.

RipEX 2 RipEX A @192.168.1.1 RipEX D @192.168	8.4.1 <sup>19</sup> 8	OSPF	o 🏾 🕮 Char	nges M Notifications O
Unit time: 2021-07-28 14:37:18	Common Network	Static rules Import filter	Export filter	
	Static rules			. =
STATUS	Destination IP / D	Destination mask 192.168.4.0/24	Metric type Type 1	<u>ب</u>
SETTINGS	Metric 1000	OSPF tag 0	Note	
Interfaces	+ Add rule			
Routing				
Static				
BABEL				
OSPF				

Fig. 7.26: RipEX\_D – OSPF Static rules

There are no special changes in other OSPF tabs. Babel is not configured in RipEX\_D. Save the changes. We recommend rebooting all RipEX2 units after final configuration to speed things up.

### 7.5. Diagnostics and Testing

Compared to previous example, communication from RipEX\_D can use either a link via RipEX\_A (primary ASBR), or via RipEX\_C (backup ASBR).

From the other end, RipEX\_B can communicate with RipEX\_D via RipEX\_A (primary), or via RipEX\_C (backup).

So, first of all, check the DIAGNOSTICS – Routing menu in all the units. Afterwards, check if you can reach every LAN IP addresses from each unit. Eventually, remove Ethernet cable between RipEX\_A and RipEX\_D and check if the communication starts via RipEX\_C. Connect the cable back and check if/when it goes back to primary link. Ping can be running from RipEX\_B to RipEX\_D, or vice versa (for example).

You can also run ping between RipEX\_A and RipEX\_C – it uses the Radio link. Try to remove antennas and check if the ping starts to use an Ethernet link via RipEX\_D.

#### 7.5.1. Checking Routing tables

Check RipEX\_D if OSPF is established successfully (State: Full) with both other RipEX2 units, Router IDs 1.1.1.1 and 3.3.3.3 with correct IP address. Note external metrics for 192.168.1.0/24 and 192.168.2.0/24, i.e., metric 1000 for routes via RipEX\_A and metric 2000 for routes via RipEX\_C.

	RipEX2 RipEX_D @192.168.4.1	n Remote access ROUTING I≣ Changes	< Notificati
	Unit time:		
ļ	2021-07-29 14:42:08 (UTC+2)	Static Dynamic BABEL OSPF BGP	
1		OSPF routing	O Refresh
	Ø STATUS	Neighbors	
	🍫 SETTINGS	BIRD 2.0.7 ready. ospf1: Router ID Pri State DTime Interface Router IP	
•		S.S.S.S.S         I         Full/BDR         7.510         IT_03P12         122.108.000.1           1.1.1.1         1         Full/BDR         7.564         if_0spf1         192.168.100.1           State & Topology         1         Full/BDR         7.564         if_0spf1         192.168.100.1	
	Overview	BIRD 2.0.7 ready.	
	Events	area 0.0.0.0	
	Statistics	router 1.1.1.1 distance 10	
		network 192.168.100.0/24 metric 10 external 192.168.1.0/24 metric2 1000	
	Ivionitoring	external 192.168.2.0/24 metric2 1000	
•	Routing	CALCINGT 152,108,5,0,24 metric2 1000	
	Tools	router 3.3.3.3 distance 10	
	_	network 192.168.200.0/24 metric 10	
	Support	external 192.168.1.0/24 metric2 2000 external 192.168.2.0/24 metric2 2000	
1		external 192.168.3.0/24 metric2 1000	

Fig. 7.27: RipEX\_D – OSPF Routing diagnostics

Check Babel routes in RipEX\_B. Especially check a route 192.168.4.0/24. If all is configured and connected physically correctly, primary link should go via RipEX\_A with Metric equal to 1100. A candidate route should be routable via RipEX\_C with Metric equal to 2100 (100 = one radio hop, 2000 = exported EXT2 Babel metric from RipEX\_C).

If both units have correct routes, it is highly probable that routes in ASBR routers are also correct.

You can e.g., check that both routers have a route to each other primarily via Radio link and a backup link via RipEX\_D (over Ethernet).

Fig. 7.28: RipEX\_C – Dynamic routing diagnostics

Note that Babel Preference to 192.168.1.0/24 via 10.10.10.1 is 210, whereas OSPF Preference is 150. Higher the Preference, more preferred route. The OSPF Metric1 is 20 ( $2 \times 10$  for OSPF) and Metric2 is 1000 (EXT2 – i.e., it does not sum metrics on the path compared to a sum for EXT1 metrics).

#### 7.5.2. Tools – ICMP ping and Routing

Choose from what device you ping what IP addresses. We check pings from RipEX\_D to

- 192.168.1.1
- 192.168.2.1
- 192.168.3.1

Go to the DIAGNOSTICS – Tools – Routing. Check what gateway is used for a particular IP. We should see 192.168.100.1 for 192.168.1.1 and 192.168.2.1. A gateway 192.168.200.2 should be seen for 192.168.3.1.

RipEX2 RipEX_D @192168.4.1	te access TOOLS 🗮 Changes
Unit time: 2021-07-29 15:01:21 (UTC+2)	ICMP Ping Routing
<ul><li>STATUS</li><li>SETTINGS</li></ul>	Parameters Destination IP 192.168.2.1 Controls
	► Run
Overview Events Statistics Monitoring Routing	Output
Tools	

Fig. 7.29: RipEX\_D – Routing check to remote networks

Select ICMP ping tab and ping the mentioned IP addresses.

RipEX_D         RipEX_D         I <sup>№</sup> 1         Ren	note access TOOLS 🗮 Changes Motifications C
Unit time: 2021-07-29 15:02:53 (UTC+2)	ICMP Ping Routing
<ul><li>STATUS</li><li>SETTINGS</li></ul>	Parameters           Length [Bytes]         200         Period [ms]         1000         Timeout [ms]         10000         Count         4           Source IP         I         Destination IP         192.168.3.1         192.168.3.1
	Controls
Overview	Start
Events Statistics Monitoring Routing Tools Support MOVANCED	Output
	208 bytes from 192.168.3.1: icmp_seq=1 ttl=64 time=0.333 ms 208 bytes from 192.168.3.1: icmp_seq=2 ttl=64 time=0.214 ms 208 bytes from 192.168.3.1: icmp_seq=3 ttl=64 time=0.279 ms

Fig. 7.30: RipEX\_D – ICMP ping to remote IP addresses

#### 7.5.3. Testing Ethernet failures

You can also ping from connected PC/laptop to remote IP addresses rather than a ping started in web interface. As the 1<sup>st</sup> test, we start a ping to 192.168.2.1 from PC 192.168.4.254 connected locally to RipEX\_D.

Rip∈×2 <sup>RipEX_D</sup> @192.168.4.1 №1 Remot	te access	ROUTING	≣ Changes				
Unit time: 2021-07-29 15:11:54 (UTC+2)	Static Dynamic	BABEL OSPF	BGP				
O STATUS	System routing						
SETTINGS	System routing table           10.10.10.0/24 dev radio proto kernel scope link src 10.10.10.4           192.0.2.232/30 dev agl0 proto kernel scope link src 192.0.2.233           192.168.1.0/24 via 192.168.100.1 dev if_ospf1 proto bird src 192.168.4.1 metric 32           192.168.2.0/24 via 192.168.100.1 dev if_ospf1 proto bird src 192.168.4.1 metric 32           192.168.3.0/24 via 192.168.200.1 dev if_ospf1 proto bird src 192.168.4.1 metric 32           192.168.4.0/24 dev if_bridge proto kernel scope link src 192.168.4.1 metric 32           192.168.100.4/24 dev if_ospf1 proto kernel scope link src 192.168.4.1           unreachable 192.168.4.0/24 proto bird metric 32           192.168.100.0/24 dev if_ospf1 proto kernel scope link src 192.168.4.1 metric 32           192.168.200.0/24 dev if_ospf1 proto bird scope link src 192.168.4.1 metric 32           192.168.200.0/24 dev if_ospf1 proto bird scope link src 192.168.4.1 metric 32           192.168.200.0/24 dev if_ospf1 proto bird scope link src 192.168.4.1 metric 32           192.168.200.0/24 dev if_ospf2 proto kernel scope link src 192.168.4.1 metric 32           192.168.200.0/24 dev if_ospf2 proto bird scope link src 192.168.4.1 metric 32           192.168.200.0/24 dev if_ospf2 proto bird scope link src 192.168.4.1 metric 32           192.168.200.0/24 dev if_ospf2 proto bird scope link src 192.168.4.1 metric 32           192.168.200.0/24 dev if_ospf2 proto bird scope link src 192.168.4.1 metric 32						
& DIAGNOSTICS							
Overview Events							
Statistics							
Monitoring							
Routing							

Fig. 7.31: RipEX\_D – Highlighted dynamic route to 192.168.2.0/24 via RipEX\_A

Once the ping is running successfully, disconnect the Ethernet cable between RipEX\_D and RipEX\_A. Keep checking current dynamic routing rules continuously and wait for ICMP ping to start working again (via RipEX\_C).

Rip∈X2 <sup>RipEX_D</sup> (№1 Remot	e access ROUTING	<ul> <li>Notifications</li> </ul>						
Unit time: 2021-07-29 15:13:08 (UTC+2)	Static Dynamic BABEL OSPF BGP							
<ul><li>STATUS</li><li>SETTINGS</li></ul>	System routing System routing table 10.10.10.0/24 dev radio proto kernel scope link src 10.10.10.4 192.0.2.232/30 dev agl0 proto kernel scope link src 192.0.2.233 192.168.1.0/24 via 192.168.200.1 dev if_ospf2 proto bird src 192.168.4.1 metric 32	C Refresh						
V DIAGNOSTICS Overview Events Statistics	192.168.2.0/24 via 192.168.200.1 dev if_ospf2 proto bird src 192.168.4.1 metric 32 192.168.3.0/24 dev if_bridge proto kernel scope link src 192.168.4.1 metric 32 192.168.4.0/24 dev if_bridge proto kernel scope link src 192.168.4.1 unreachable 192.168.4.0/24 proto bird metric 32 192.168.100.0/24 dev if_ospf1 proto kernel scope link src 192.168.100.2 linkdown 192.168.100.2 dev if_ospf1 proto bird scope link src 192.168.4.1 metric 32 linkdown 192.168.200.0/24 dev if_ospf2 proto kernel scope link src 192.168.4.1 metric 32 linkdown 192.168.200.0/24 dev if_ospf2 proto bird scope link src 192.168.4.1 metric 32 linkdown 192.168.200.0/24 dev if_ospf2 proto bird scope link src 192.168.4.1 metric 32							
Monitoring Routing								

Fig. 7.32: RipEX\_D – Highlighted dynamic route to 192.168.2.0/24 via RipEX\_C

Because the OSPF timing is set to be very fast, you should also see a rapid (within several seconds) change in the mentioned routing.

Connect the Ethernet cable back and see when the communication comes back via primary ASBR RipEX\_A.

#### 7.5.4. Testing Radio failures

Connect locally to RipEX\_A, e.g., using 192.168.1.254/24 IP address and static routes to remote networks.

Start a ping to 192.168.3.1 (RipEX\_C). The ping should be running over the Radio channel and RTT should be in tens/hundreds of milliseconds (based on selected modulation).

```
C:\Windows\system32>ping 192.168.3.1 -t

Pinging 192.168.3.1 with 32 bytes of data:

Reply from 192.168.3.1: bytes=32 time=40ms TTL=63

Reply from 192.168.3.1: bytes=32 time=47ms TTL=63

Reply from 192.168.3.1: bytes=32 time=40ms TTL=63

Reply from 192.168.3.1: bytes=32 time=47ms TTL=63

Reply from 192.168.3.1: bytes=32 time=47ms TTL=63

Reply from 192.168.3.1: bytes=32 time=47ms TTL=63
```

Fig. 7.33: RipEX\_A – Ping from laptop to RipEX\_C

Remove RipEX\_A and RipEX\_C antennas to stop Radio traffic communication and keep checking Dynamic routing in RipEX\_A diagnostics.

Neighbors								
BIRD 2.0.7 ready.								
babel1:								
IP address	Interface	Metric	Routes	Hellos	Expires			
fe80::202:a9ff:fe20:ae3	radio	160	3	10	1.164			
fe80::202:a9ff:fe20:789	radio	177	3	9	18.274			
Routes								
BIRD 2.0.7 ready.								
babel1:								
Prefix	Nexthop			Interfa	ace Metric	F	Seqno	Expires
192.168.2.0/24	10.10.10.2			radio	177	*	3	147.276
192.168.2.0/24	10.10.10.3			radio	260		3	181.690
192.168.3.0/24	10.10.10.3			radio	160	*	3	181.690
192.168.3.0/24	10.10.10.2			radio	277		3	147.276
192.168.4.0/24	10.10.10.3			radio	2160	*	3	181.690
192.168.4.0/24	10.10.10.2			radio	65535		3	147.986

Fig. 7.34: RipEX\_A – Babel metrics go down for the Radio channel

Also keep an eye on running ping. The communication should stop working until a route via Ethernet (RipEX\_D) is used an RTT drops to units of milliseconds. This process is much slower due to Babel settings on the Radio channel (30 seconds Hello intervals, ...) and big difference in metrics (100 for one Babel hop, whereas 1000 or 2000 metrics for OSPF).



Fig. 7.35: RipEX\_A – Dynamic routing using Ethernet (OSPF) connection to RipEX\_C

Ping starts working via Ethernet.

Reply	from	192.168.3.1:	bytes=32	time<1ms	TTL=62
Reply	from	192.168.3.1:	bytes=32	time<1ms	TTL=62
Reply	from	192.168.3.1:	bytes=32	time<1ms	TTL=62
Reply	from	192.168.3.1:	bytes=32	time=4ms	TTL=62
Reply	from	192.168.3.1:	bytes=32	time<1ms	TTL=62
Reply	from	192.168.3.1:	bytes=32	time<1ms	TTL=62

Fig. 7.36: RipEX\_A – RTT drops down

OSPF works fine, whereas Babel has no route to the destination.

Fig. 7.37: RipEX\_A – Babel route is unreachable

Connect antennas back and see if and when it comes back. This operation should take less time than going from primary to backup route.

For the whole process, you can also check Monitoring and Statistics pages of RipEX2 Diagnostics.

# 8. Hints and Tips

### 8.1. Throughput, speed

As already mentioned within examples, consider Hello intervals together with SCADA protocol requirements. Using 4 RipEX2 units, these were measured values on the RF channel (just several examples):

- Hello interval: 5 seconds, Multiplier: 2, ~ 1 pps, 792 bps
  - $\circ\,$  Path switching usually within several tens of seconds, up to 2 minutes
- Hello interval: 15 seconds, Multiplier: 4, ~ 0.3 pps, 224 bps
- Hello interval: 20 seconds, Multiplier: 3, ~ 0.2 pps, 168 bps
  - Path switching usually 1-3 minutes
- Hello interval: 30 seconds, Multiplier: 4, ~ 0.1 pps, 102 bps
  - $\circ\,$  Path switching between approx. 30 seconds and 5 minutes

### 8.2. Static path setup

It might be beneficial to statically define those packets should primarily go over repeater "A" and only if this link fails, use repeater "B", or even direct link. Babel (and OSPF/BGP) work in background and logic is based on metrics and particular algorithms. It is not possible to manually define it this way as with RipEX Backup paths. Only the received cost can be set to higher or lower values, but per unit/interface, not per neighboring IP. E.g., one repeater can have Rx cost set to 100, whereas second one 200. This can prioritize the first one.

It is also possible that while topology changes, the metric can change up and down for multiple paths. It can switch to turned off path again, because even other metrics are being increased. Eventually, the path over turned off repeater gets a metric of 65535 and then disappears completely.

### 8.3. Advanced configurations

For any other advanced setup, contact our technical support *support@racom.eu*<sup>1</sup>.

<sup>&</sup>lt;sup>1</sup> mailto:support@racom.eu

# **Revision History**

Revision 1.0 First issue 2021-10-01

Revision 1.1

2024-08-28

Added Chapter 2, Mesh topology with Radio and Relay filters