BIRD 1 End of Life

Dear Users, BIRD 1 last version is 1.6.8, released in 2019. BIRD 2 has been stable for a long time and the BIRD 1 codebase knowledge is slowly waning out throughout BIRD Team. It takes us more and more time to support BIRD 1, especially to check whether bugs found in BIRD 2 also apply to 1.6.8.

We have announced **BIRD 1 End of Life** to happen on **Dec 31, 2023**. If you are still running BIRD 1, you should upgrade to BIRD 2 as soon as possible.

BIRD customer support by CZ.NIC

We offer Support Packages for everyone who needs to upgrade their BIRD 1 configuration to BIRD 2 and learn everything about BIRD 2 and upcoming BIRD 3 options. Please contact us at bird-support@network.cz for more.

Two daemons become one

BIRD 2 supports both IPv6 and legacy IPv4 in one daemon, allowing to have one single configuration. This brings some necessary changes inside configuration. Routing table declaration is now typed: instead of `table foo;`, you write `ipv6 table foo;` and also `master` table is now `master6` or `master4`.

All protocols (except for Pipe) now have an explicit **channel** configuration:
Let’s consider an old BIRD 1 configuration of Static:

```
protocol static {
 import filter { bgp_med = 42; };
 table special_routes;
 route 2001:db8:beef::/48 unreachable;
 route 2001:db8:f000::/40 via 2001:db8::baa3;
}
```

For BIRD 2, you have to explicitly specify the channel:

```
protocol static {
 ipv6 {
       import filter { bgp_med = 42; };
       table special_routes;
 };
 route 2001:db8:beef::/48 unreachable;
 route 2001:db8:f000::/40 via 2001:db8::baa3;
}
```

Multiprotocol BGP

BGP now supports multiple address families inside one connection. By specifying multiple channels, you can exchange IPv6, IPv4 (both unicast and multicast SAFI) and also VPNv6, VPNv4 and Flowspec which is not supported in BIRD 1. The configuration then can look like this:

```
protocol bgp bgp_client42 {
 ipv6 {
       import where check_import_ipv6();
       export where check_export_ipv6();
       table tab6_client42;
 };
 flow6 {
       import where check_import_flow6();
       export where check_export_flow6();
       table flow6_client42;
 };
 ipv4 {
       import where check_import_ipv4();
       export where check_export_ipv4();
       table tab4_client42;
 };
 neighbor 2001:db8::4e14 as 65500;
 local as 65501;
}
```

RPKI-To-Router Protocol

Manipulating ROAs is entirely different in BIRD 1 and BIRD 2. Definition of static ROA tables in configuration are done by the Static protocol in BIRD 2. **Dynamic updates** are implemented by RPKI-To-Router (RTR) protocol, **allowing to load ROAs from public caches** like Cloudflare **or locally** running Routinator or StayRTR. You configure your RPKI connection as follows:

```
protocol rpki {
 roa6 { table r6; import all; };
 roa4 { table r4; import all; };
 remote 2001:db8::33aa port 8282;
 transport tcp;
}
```

The CLI commands for dynamically manipulating ROA tables have been removed.
You can either inject such ROA into your RTR cache or into the static configuration and reload.

Other changes

Many other minor changes have happened which can't be covered by this simple cheatsheet.
For more information, check also https://gitlab.nic.cz/labs/bird/-/wikis/transition-notes-to-bird-2
If you come across anything strange or different, you may ask in our mailing-list bird-users@network.cz.