



BIRD Internet Routing Daemon

Ondřej Zajíček

CZ.NIC z.s.p.o.

2015-02-16



BIRD overview



- ▶ BIRD Internet Routing Daemon
- ▶ Routing protocols BGP, OSPF, RIP and BFD
- ▶ IPv4 and IPv6 support
- ▶ Linux and BSD kernel support
- ▶ Free and open source software (GPL)



BIRD features

- ▶ Programmable filters
- ▶ Clear and structured config files
- ▶ Multiple protocol instances
- ▶ Multiple routing tables
- ▶ Automatic reconfiguration on the fly
- ▶ Extensive documentation



Typical applications

- ▶ OSPF routers in enterprise or small ISP networks
- ▶ BGP for external routing or route reflectors
- ▶ Route servers in internet exchange points

BGP Route server:

- ▶ Brokering of routing information in IXPs
- ▶ Only distribution of routing information
- ▶ Task not suited for dedicated hardware routers
- ▶ Requirements for many tables and flexible filtering

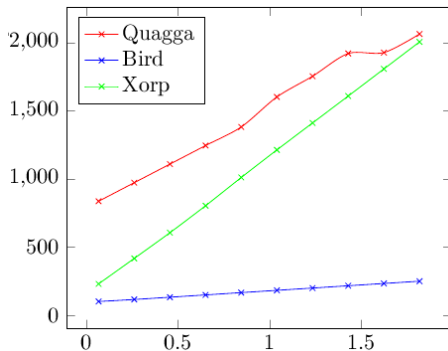


BIRD deployments

Euro-IX 2013: BIRD most popular route server in IXPs



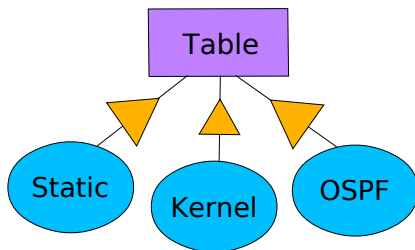
Memory usage comparison



Memory usage [MB] / number of paths in RIB [millions].
Results and the graph taken from independent comparison at
http://wh.cs.vsb.cz/sps/images/b/ba/BGP_route_server.pdf

BIRD Concepts

- ▶ Routes
- ▶ Protocols
- ▶ Tables
- ▶ Filters



Protocols

- ▶ Represent routing protocols (BGP, OSPF, RIP)
- ▶ Or other route sources (static, kernel, direct)
- ▶ Generate and receive routes
- ▶ Are connected to routing tables
- ▶ Protocols may have more instances



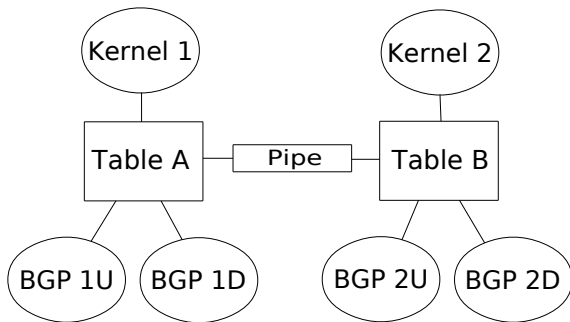
Tables

- ▶ Import and accumulate routes from protocols
- ▶ Preferred route is selected for each destination
- ▶ Then exported to attached protocols
- ▶ BIRD supports any number of tables
- ▶ Two tables can be connected through pipe



More routing tables

- ▶ Usually just one routing table
- ▶ Linux kernel supports more tables
- ▶ Can be used for e.g. policy routing
- ▶ Also for route servers



Filters

- ▶ Stand between protocol and table
- ▶ May modify, reject or accept routes
- ▶ Scripting language for route filtering
- ▶ Filter may access all route attributes
- ▶ Also for filtering of listings of routing tables



Filters – example

```
define martians = [ 10.0.0.0/8+, 172.16.0.0/12+  
    192.168.0.0/16+, 169.254.0.0/16+, 224.0.0.0/4+,  
    240.0.0.0/4+, 0.0.0.0/32-, 0.0.0.0/0{25,32} ];
```

```
filter bgp_in  
{  
    if net ~ martians then reject;  
    if bgp_path.first != 1234 then reject;  
    if bgp_path.len > 64 then reject;  
  
    if net ~ [120.10.0.0/16+, 120.20.0.0/16+]  
    then bgp_local_pref = 500;  
    else bgp_local_pref = 100;  
  
    bgp_med = 0;  
    accept;  
}
```

Essentials

- ▶ Separation of IPv4 and IPv6
- ▶ Config file and reconfiguration
- ▶ Control socket, birdc shell and commands
- ▶ Logging

- ▶ show route [all]
- ▶ show protocols [all]
- ▶ show interfaces
- ▶ show ospf ...



Basic configuration

```
router id 192.168.1.1;
log syslog all;

protocol device {
}

protocol static {
    import all;
    route 192.168.10.0/24 via 192.168.1.2;
    route 192.168.20.0/24 unreachable;
}

protocol kernel {
    export all;
    scan time 10;
}
```



Commands – examples

- ▶ show route 192.168.1.0/24
- ▶ show route for 192.168.1.10
- ▶ show route protocol ospf1
- ▶ show route where gw \sim 192.168.0.0/16
- ▶ show route where net.len \sim [16..24, 32]
- ▶ show route where bgp_path.len $>$ 4
- ▶ show route where proto \sim "bpg*"
- ▶ show route where ifname = "eth0"
- ▶ show route filter myfilter

- ▶ show protocols
- ▶ enable | disable | restart ospf1
- ▶ configure [timeout | undo | confirm]
- ▶ down



OSPF – Open Shortest Path First

- ▶ Popular protocol for internal routing
- ▶ OSPFv2 for IPv4 (RFC 2328)
- ▶ OSPFv3 for IPv6 (RFC 5340)
- ▶ Router monitors reachability of its neighbors
- ▶ Local network topology is distributed to neighbors (LSA - Link State Advertisement)
- ▶ Every router gets complete map of network
- ▶ And computes shortest paths to all destinations



OSPF configuration

```
protocol ospf {  
    import all;  
    export filter {  
        ospf_metric1 = 1000;  
        if source = RTS_STATIC then accept; else reject;  
    };  
  
    area 0 {  
        interface "eth0" {  
            cost 5; hello 5; wait 10; dead 60;  
        };  
        interface "eth*" {  
            cost 100; type pointopoint;  
        };  
    };  
};
```



BGP – Border Gateway Protocol

- ▶ Standard protocol for internet routing
- ▶ BGPv4 (RFC 4271)
- ▶ Router receives routes from its neighbors
- ▶ Chooses preferred routes by local policy
- ▶ Preferred routes are used for forwarding
- ▶ And possibly propagated to other neighbors
- ▶ Forwarded routes contain many additional route attributes



BIRD as BGP router

```
protocol static {  
    import all;  
  
    route 10.10.0.0/16 reject;  
    route 10.20.0.0/16 reject;  
}  
  
protocol bgp {  
    import all;  
    export where source = RTS_STATIC;  
  
    local 192.168.1.1 as 65100;  
    neighbor 192.168.1.2 as 65200;  
}
```



BFD – Bidirectional Forwarding Detection

- ▶ Protocol for neighbor reachability and liveness testing
- ▶ Supplementary protocol to OSPF, BGP, ...
- ▶ Reaction time in tens to hundreds of ms
- ▶ Command `show bfd sessions`



BFD – Bidirectional Forwarding Detection

```
protocol bfd {  
    interface "eth*" {  
        interval 50 ms;  
        multiplier 4;  
    };  
}  
  
protocol bgp {  
    . . .  
  
    local 192.168.1.1 as 65100;  
    neighbor 192.168.1.2 as 65200;  
    bfd;  
}
```



IPv6 router advertisements

- ▶ For IPv6 stateless address autoconfiguration
- ▶ Easy way to generate IPv6 RAs from BIRD
- ▶ Support of RDNSS a DNSSL in RAs
- ▶ Dynamic IPv6 router advertisements

```
protocol radv {  
    interface "eth*";  
    rdnss 2001:0DB8:1234::10;  
    dnssl "domain.cz";  
    trigger 2000::/3;  
}
```



Future plans

- ▶ Integrated multiprotocol design
- ▶ MPLS/VPN support
- ▶ Ethernet AF / bridge FDB support
- ▶ IS-IS protocol



Pitfalls

Sockets API

- ▶ Nice for simple TCP sockets
- ▶ Not so nice for raw or multicast sockets
- ▶ Sending packets with specified src addr and iface
- ▶ bind() overloaded / useless for multicast
- ▶ On Linux, at least we have SO_BINDTTOIFACE
- ▶ IP_PKTINFO vs IP_SENDSRCADDR vs IP_HDRINCL
- ▶ For IPv6, IPV6_PKTINFO works well



Pitfalls

Ephemeral Source Port Selection

- ▶ IANA, RFC 6335 – range 49152–65535 should be used
- ▶ Linux – by default 32768–61000
- ▶ Tunable by `net.ipv4.ip_local_port_range`
- ▶ FreeBSD – by default 10000–65535, also tunable
- ▶ In FreeBSD, we have `IP_PORTRANGE_HIGH` cmsg
- ▶ Some BFD implementations reject packets with *sport* < 49152



Pitfalls

FIB, Netlink

- ▶ Multipath routes in IPv4 and IPv6
- ▶ Missing RTM_DELROUTE notifications for removed routes due to iface down
- ▶ IPv6 kernel device routes did not use RTPROT_KERNEL
- ▶ IPv6 routes did not support RTN_BLACKHOLE, RTN_PROHIBIT
- ▶ net.ipv6.route.max_size limit





Questions?

<http://labs.nic.cz/>
<http://bird.network.cz/>