

# BIRD Internet Routing Daemon

Introduction, version 2.0.x



Ondřej Filip • [ondrej.filip@nic.cz](mailto:ondrej.filip@nic.cz) • 11 Jun 2018 • Brno  
• CSNOG 1



# Project history



- Project started in 1998
  - Seminar project – Charles University Prague
  - Project slept for a while
  - Small reincarnation in 2003 and 2006
  - Project fully renewed since Q4 2008 – part of CZ.NIC Labs - <https://labs.nic.cz>



# Features

- Multiple routing table - RIBs (internal and also synchronization with OS)
- Protocol PIPE
- Multiple routers, route reflectors on a single system
- Powerful configuration
- Very powerful filtering language
- Command line interface (show, restart, ...)
- Automatic reconfiguration
- Latency tracking & internal watchdog

: :



# Features

- Portable – Linux, FreeBSD, NetBSD, OpenBSD
  - IPv4/IPv6 support, IPv6 RA
  - Static routing, BFD
  - RIP, RIPv2, RIPng
  - OSPFv2, OSPFv3
  - Babel
  - BGP
  - RPKI
  - MRTdump logging
- :

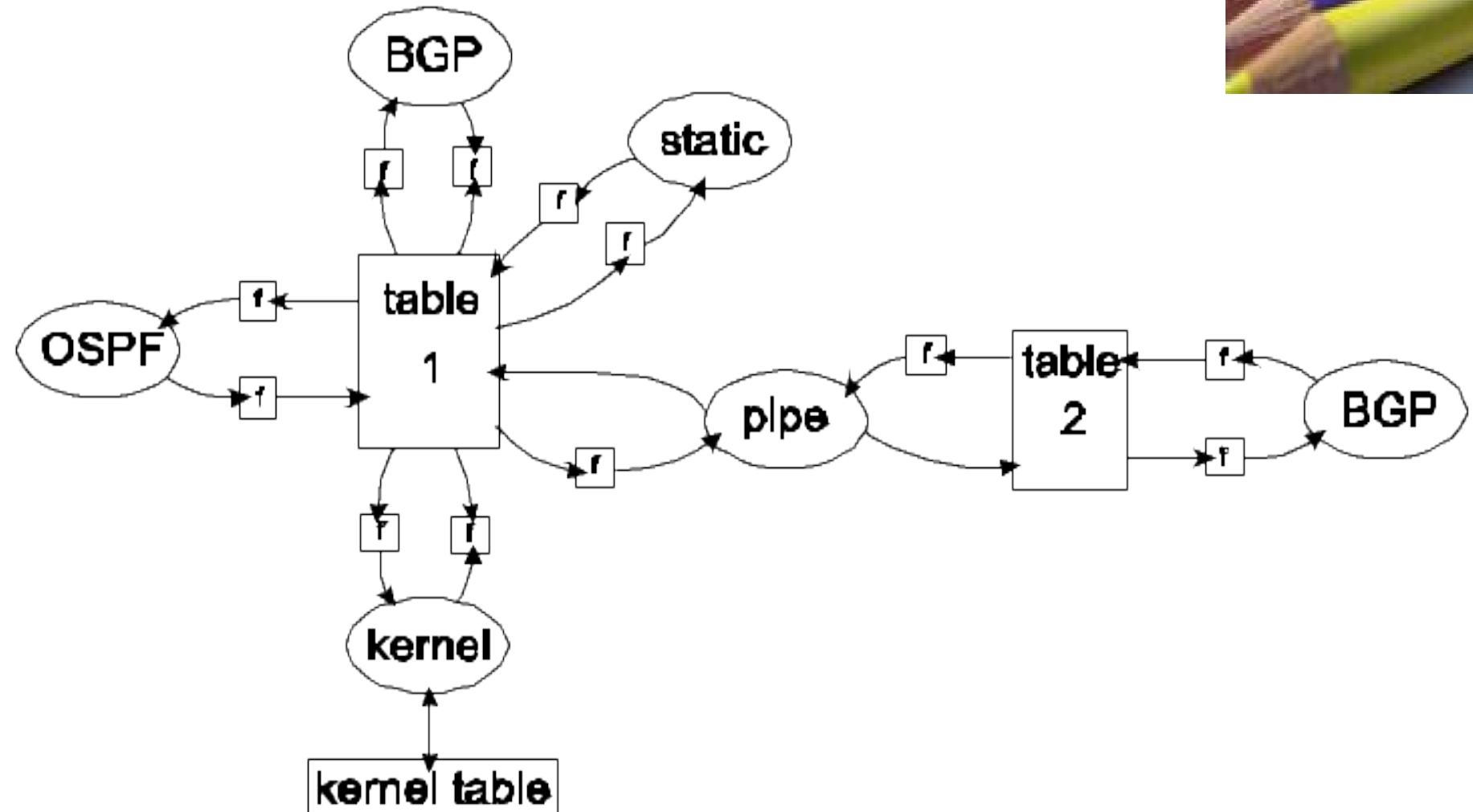


# BGP features

- BGP community, extended, large
- Capability negotiations
- Graceful restart
- Route reflector, Route server
- Add-path, BGP multipath
- ASN32, RFC6286 - BGP AS-wide unique rtr ID
- RFC7313 - BGP enhanced route refresh
- Link state support in BGP

: :

# Design



# Configuration example

```
log "/var/log/bird.log" all;  
  
router id 193.51.100.238;  
  
protocol static {  
    route 10.0.0.0/8 drop;  
    route 172.16.0.0/12 drop;  
    route 192.168.0.0/16 drop;  
}  
filter bgp_out {  
    if (net = 192.175.48.0/24 ) &&  
        (source = RTS_DEVICE) then accept;  
    else reject;  
}  
protocol bgp NIX_1 {  
    local as 112;  
    neighbor 193.51.100.235 as 6981;  
    import all;  
    export filter bgp_out;  
}
```



# CLI example



```
bird> show protocols
name      proto    table   state  since   info
direct1   Direct   master  up     Apr11
kernel1   Kernel   master  up     Apr11
device1   Device   master  up     Apr11
static1   Static   master  up     Apr11
NIX_2     BGP      master  up     Apr11   Established
NIX_1     BGP      master  up     Apr25   Established
ospf1     OSPF     master  up     Apr11   Running
bird>
bird> show status
BIRD 1.6.3
Current server time is 06-08-2017 22:01:06
Last reboot on 11-07-2017 22:54:12
Last reconfiguration on 30-07-2017 06:25:25
Daemon is up and running
bird>
```





# CLI example

```
bird> show route
10.0.0.0/8      via 200.30.10.3 on eth2 [ospf1 13:10] E2 (150/5/1000)
127.0.0.0/8     dev lo [direct1 13:09] (240)
200.30.20.0/24   via 200.30.10.3 on eth2 [ospf1 13:10] I (150/10)
200.30.10.0/24   dev eth2 [direct1 13:09] (240)
                  dev eth2 [ospf1 13:10] I (150/5)
200.0.10.0/24    dev eth0 [direct1 13:09] (240)
                  dev eth0 [ospf1 13:09] I (150/5)
172.16.0.0/16    via 200.30.10.3 on eth2 [ospf1 13:10] E2 (150/5/1000)
195.47.235.0/24  via 194.50.100.246 on eth1 [NIX2 Apr11] (100)[AS688i]
                  via 194.50.100.245 on eth1 [NIX1 Apr25] (100)[AS688i]
bird>
bird> show route protocol ospf1
10.0.0.0/8      via 200.30.10.3 on eth2 [ospf1 13:10] E2 (150/5/1000)
200.30.20.0/24   via 200.30.10.3 on eth2 [ospf1 13:10] I (150/10)
200.30.10.0/24   dev eth2 [ospf1 13:10] I (150/5)
200.0.10.0/24    dev eth0 [ospf1 13:09] I (150/5)
172.16.0.0/16    via 200.30.10.3 on eth2 [ospf1 13:10] E2 (150/5/1000)
```





# CLI example

```
bird> show route for 127.0.0.1  
127.0.0.0/8      dev lo [direct1 13:09] (240)
```

```
bird> show route filter bgp_out  
192.175.48.0/24    dev dummy0 [direct1 Apr1] (240)
```

```
bird> show route count  
1469 of 1469 routes for 849 networks
```

```
bird> show route export NIX_1  
192.175.48.0/24    dev dummy0 [direct1 Apr1] (240)
```

```
bird> show route where 127.0.0.5 ~ net  
0.0.0.0/0      via 195.47.235.1 on eth0 [static1 Apr1](200)  
127.0.0.0/8    dev lo [direct1 Apr1] (240)
```

```
bird> show route filter {if 127.0.0.5 ~ net then accept;}  
0.0.0.0/0      via 195.47.235.1 on eth0 [static1 Apr1](200)  
127.0.0.0/8    dev lo [direct1 Apr1] (240)
```

• •

# Filter example – route servers

- Route server policy - NIX.CZ

Evaluation order	Community	Action
1	0:<peer-as>	Do not advertise to <peer-as>
2	47200:<peer-as>	Advertise to <peer-as>
3	0:47200	Do not advertise to any peer
4	47200:47200	Advertise to all peers



# Filter example (ASN16 only)

```
define myas = 47200;

function bgp_out(int peeras)
{
    if ! (source = RTS_BGP ) then return false;
    if (0,peeras) ~ bgp_community then return false;
    if (myas,peeras) ~ bgp_community then return true;
    if (0, myas) ~ bgp_community then return false;
    return true;
}

protocol bgp R25192x1 {
    local as myas;
    neighbor 194.50.100.13 as 25192;
    import where bgp_in(25192);
    export where bgp_out(25192);
    rs client;
}
: : : : : : : : : : : : : : : : : : : : : : : : : : : :
```

# Filter example

```
function asmatch()
int set asnums;
{
    asnums = [ 11111, 22222, 33333, 44444, 55555,
               66666, 77777, 88888, 99999, 100..200 ];

    # Check originating AS number
    if bgp_path.last ~ asnums then return true;

    return false;
}
```

# Filter example

```
function avoid_martians()
prefix set martians;
{
    martians = [ 169.254.0.0/16+, 172.16.0.0/12+,
        192.168.0.0/16+, 10.0.0.0/8+, 224.0.0.0/4+,
        240.0.0.0/4+, 0.0.0.0/32-, 0.0.0.0/0{25,32},
        0.0.0.0/0{0,7} ];

    # Avoid RFC1918 networks
    if net ~ martians then return false;

    return true;
}
```

# Filters

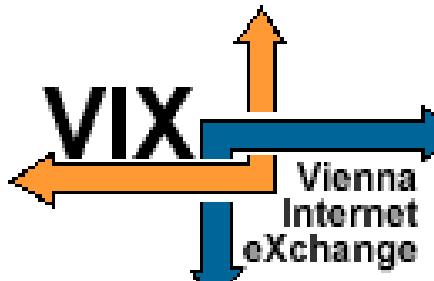
- Filters compiled into bytecode
- Variables, sets, constants
- Data types: bool, int, ip, prefix, enum, quad, string, bgppath, bgpmask, clist, eclist, lclist
- Operators: +, -, \*, /, comparisons, logical, element\_of\_set (~), roa\_check()
- Control structures – if/else, case, functions
- Set implemented by Weight-balanced tree (or similar structures) – logarithmic time complexity

# Protocol templates

```
template bgp NIXPEERS {
    local as 112;
    export filter bgp_out;
    start delay time 120;
    mrtdump all;
    import limit 50000 action warn;
}

protocol bgp NIXRS1 from NIXPEERS {
    neighbor 91.210.16.1 as 47200;
    import limit 60000 action block;
}
```

# Deployed at ... (and much more)



# Prize – LINX & Charles University



# Current development

- New releases 4 months ago - two branches
- 1.6.x - stable
  - Currently 1.6.4
  - LTS branch
- 2.0.x - „experimental“
  - Major redesign – IPv4 & IPv6 integration
  - Slight changes in configuration





# Current version – 1.6.4

- Basic VRF support
- RAdv: Support for more specific routes (RFC 4191)
- BGP: Shutdown communication (RFC 8203)
- BGP: Exchanging LOCAL\_PREF with eBGP peers
- BGP: Allow to specify interface for regular sessions
- BGP: New option 'disable after cease'

: :

# New version 2.0.2



- BGP multicast support (SAFI 2)
- BGP flowspec support (RFC 5575)
- New RPKI-Router protocol
- BGP with MPLS labels (RFC 3107)
- BGP MPLS/VPN support (RFC 4364)
- BGP New option 'disable after cease'



# New version 2.0.2



- VPNv4 and VPNv6 network types
- BGP 6PE - IPv6 NLRI over IPv4 MPLS (RFC 4798)
- BGP IPv4 NLRI with an IPv6 Next Hop (RFC 5549)
- BGP Confederations (RFC 5065)
- Default EBGP Route Propagation Behavior (RFC 8212)





# New version family 2.0.x - config

```
protocol bgp example_bgp {  
    local 192.168.11.1 as 1000;  
    neighbor 192.168.11.2 as 2000;  
    ipv4 {  
        import filter avoid_martians;  
        export where source ~ [ RTS_STATIC, RTS_BGP ];  
    };  
    ipv6 {  
        import all;  
        export where source ~ [ RTS_STATIC ];  
        next hop address 2001:db8:1:1::1;  
    };  
}
```

• •



# New version family 2.0.x - config

```
protocol bgp example_bgp {  
    local 192.168.11.1 as 1000;  
    neighbor 192.168.11.2 as 2000;  
    ipv4 mpls { #IPv4 with MPLS labels  
        table mtab4; import all; export all;  
    };  
    vpn6 multicast { #VPNv6 multicast topology  
        table vpn6mc; import all; export all;  
    };  
    flow6 { #IPv6 Flowspec  
        table flowtab6; import all; export all;  
    };  
}
```

# New version family 2.0.x - config



```
protocol rpk {
    roa4 { table r4; };    roa6 { table r6; };
    remote 192.168.1.1 port 2345;
    transport ssh {
        bird private key "/home/birdgeek/.ssh/id_rsa";
        remote public key "/home/birdgeek/.ssh/known_hosts";
        user "birdgeek";
    };
}
filter peer_in_v6 {
    if (roa_check(r6, net, bgp_path.last) = ROA_INVALID) then
    {
        reject;
    }
    accept;
}
```





# New version family 2.0.x - config

```
protocol static {  
    flow4;  
  
    route flow4 {  
        dst 10.0.0.0/8;  
        port > 24 && < 30 || 40..50,80 && >= 90;  
        tcp flags 0x03/0x0f;  
        length > 1024;  
        dscp = 63;  
        fragment dont_fragment;  
    };  
};  
: : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : : :
```

# Future development (mainly 2.0.x)

- Depends on our supporters!
  - Filter re-evaluation if ROA changes
  - Filter improvement – set/list iterator
    - e.g. `foreach x (bgp_community) { execute; }`
  - BGP convergency time and responsiveness
  - Filter optimization
  - ISIS
  - ...



# Conclusion



- New version 1.6.4 - LTS
- Many new BIRD features in version 2
  - Please help us testing!
  - Feedback needed!



# Thank You!



Ondrej Filip • [ondrej.filip@nic.cz](mailto:ondrej.filip@nic.cz) • <http://bird.network.cz>