

BIRD 2 Upgrade Tutorial

Artificial Ornithology Lab

Maria Matějka • 27 November 2023

cz.nic | CZ DOMAIN
REGISTRY

BIRD 1 is EOL

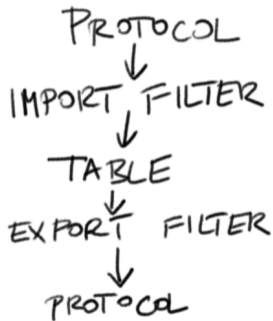
Upgrade to BIRD 2 now

Don't wait for BIRD 3

Agenda

- High-level concepts of BIRD 1, 2 and 3
- What is same and what is changing
- Table configuration
- Protocol walkthrough
- Filter development
- Future plans and thoughts
- Do not do this, please

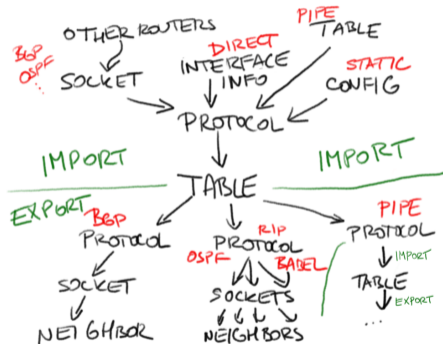
BIRD architecture overview



BIRD architecture overview

Protocol

- gathers routes
- imports into the table
- exports from the table
- sends updates around



Major architectural changes between versions

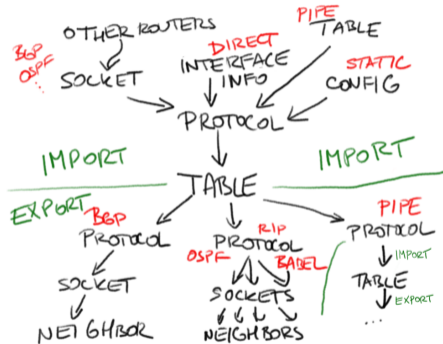
BIRD 2

explicit channel definitions

BIRD 3

asynchronous route propagation

BIRD	Tables per protocol	Order of propagation
1	one	sync
2	multiple	sync
3	multiple	decoupled



Major user-level changes between versions

BIRD 2

legacy IP build integrated into IPv6
automatic ROA loading and evaluation
multiple address family support
different show-protocol output
default tables: `master4`, `master6`

BIRD 3

multithreaded execution
different show-route output
full YANG/CBOR API (in future)

What stayed (almost) the same

- Protocol-specific options
- Protocol algorithms and packet parsers
- Filter language syntax and principles
- Best route selection algorithms
- Basic operational behavior

Q&A #1

Table definition in BIRD 2

```
ipv6 table meow;  
ipv4 table meow_legacy;
```

```
vpn6 table woof;  
roa6 table eek;  
flow6 table clog;
```

```
mpls table stickers;
```

Babel

- All channels in one protocol
- Easy to setup, easy to run
- BIRD 2+: also RTT metric available

```
protocol babel {  
  ipv6 { ... };  
  ipv4 { ... };  
  interface "my-wg-vpn-*";  
  ...  
}
```

BGP multiprotocol

- Multiple channels for BGP
- VPN (RFC 4364), FlowSpec (RFC 8955)
- Mixed nexthops (RFC 8950)

- Confederation support
- BGP Roles (RFC 9234)
- Explicit Adj-RIB-In and Adj-RIB-Out

BGP channel config in BIRD 2+

- Filters and limits
- Export filter executes **before** final attribute adjustments
- Nexthop and Gateway settings
- Add Paths
- GR and LLGR
- Table connections
- new: Adj-RIB-In (before import filter)
- new: Adj-RIB-Out (right after export filter)

BGP Config for BIRD 1

```
protocol bgp {  
    local 2001:db8::1 as 64500;  
    neighbor 2001:db8::2 as 64501;  
    import where bgp_path.len < 42;  
    add paths;  
    ...  
}
```


BGP Config for BIRD 2+

```
protocol bgp {  
  local 2001:db8::1 as 64500;  
  neighbor 2001:db8::2 as 64501;  
  ipv6 {  
    import where bgp_path.len > 42;  
    add paths;  
    ...  
  };  
  ...  
}
```

BGP Config for BIRD 2+

```
protocol bgp {  
    local 2001:db8::1 as 64500;  
    neighbor 2001:db8::2 as 64500;  
    ipv6 { table master6; igp table igp6; ... };  
    ipv4 { table master4; igp table igp4; ... };  
    flow6 { table fs6; base table master6; ... };  
    flow4 { table fs4; base table master4; ... };  
    ...  
}
```

BGP Config for BIRD 2+

```
protocol bgp {  
  ipv6 {  
    table master6; import table; export table;  
    ... };  
  ipv4 {  
    table master4; import table; export table;  
    extended next hop; ... };  
  ...  
}
```

BGP in BIRD 3

- runs in parallel in worker threads
- Adj-RIB-In reimplemented (stored in main table)
- Adj-RIB-Out stores on-wire route state (after final adjustments)

BMP Config for BIRD 2+

- not available in BIRD 1
- always monitoring all BGP's
- experimental, everything may change

```
protocol bmp {  
    station address ip 2001:db8::1 port 6543;  
    system name "cats";  
    system description "meow";  
    monitoring rib in pre_policy;  
    monitoring rib in post_policy;  
}
```

L3VPN Config for BIRD 2+

- not available in BIRD 1
- translating IP routes to VPN and back

```
protocol l3vpn meow {  
  vrf "cats";  
  ipv6 { table freecat6; }; vpn6 { table catbox6; };  
  ipv4 { table freecat4; }; vpn4 { table catbox4; };  
  mpls { label policy aggregate; label range vrfrange; };  
  rd 64500:11; route target (rt, 64500, 2);  
}
```

MRT

- BIRD 2+ allows direct routing table specification for monitoring
- no other changes

OSPF in BIRD 2+

- default (shame on us!) OSPF v2 for only legacy IP
- `protocol ospf v3 <name>` for OSPF v3
- `ipv6 { ... }`; channel setting as in BGP
- table and filters → channel
- one channel per protocol

Pipe in BIRD 2+

- removed opaque mode
- no other changes

RAdv in BIRD 2+

- `ipv6 { ... };` channel setting as in BGP
- table and filters → channel
- minor improvements

RIP in BIRD 2+

- default (shame on us!) RIP v2 for only legacy IP
- `protocol rip ng <name>` for RIP NG
- `ipv6 { ... }`; channel setting as in BGP
- table and filters → channel
- one channel per protocol
- added: demand circuit mode

Protocol RPKI in BIRD 2+

- replaces CLI interface present in BIRD 1
- gathering ROAs over RTR protocol from a trusted cache

```
protocol rpki vet {  
  roa6 { table rt6; import all; };  
  roa4 { table rt4; import all; };  
  remote 2001:db8::4e1 port 8282;  
  transport ssh { user "bird"; bird private key "beep";  
    remote public key "known_vets";  
};  
}
```

Static Route Config for BIRD 2+

```
protocol static cat_statue {  
    ipv6 { table meow6; };  
    route 2001:db8:bad::/48 unreachable;  
    ...  
}
```

Static ROA Config for BIRD 2

```
protocol static vet_statue {  
    roa6 { table rt6; };  
    route 2001:db8:bad0::/44 max 48 as 64500;  
    ...  
}
```

Q&A #2

Filters: Variables inside blocks

```
filter cat {  
    int paw;  
    ...  
    if igp_metric < 175 then {  
        int hair;  
        ...  
    } else reject;  
    ...  
}
```


Custom route attributes

- Available since 2.0.3 (2019)
- Limited number of attribute types
- Ignored by all protocols but pipes
- Handy for keeping intermediate results

Unknown BGP attributes

- Available since 2.14 (2023)
- Always a bytestring
- Always transitive
- Set, print or delete

Filters: Custom attributes

```
attribute int meow;  
attribute bgp 29 bytestring paws;  
...  
filter cat {  
    case roa_check(rt6) {  
        ROA_VALID:    meow = 2;  
        ROA_UNKNOWN: meow = 1;  
        ROA_INVALID: meow = 0;  
    }  
    ...  
}
```

Filters: Community list preprocessing

```
if (64500, 42) ~ bgp_community
then meow = 42;
else meow = 0;

# Accessing meow is faster than
# checking the whole community list.
```

Filters: Loops over iterables

- Available since 2.14 (2023)
- Run a code block for every item
- Works on AS Path and Communities

Filters: Loops over iterables

```
bool stuffed = false; int n = 0;
int prev = 0;
for int asn in bgp_path do {
    if prev = asn then { stuffed = true; n = n + 1; }
    else if n > 0 then {
        print "ASN ", prev, " stuffed ", n, " times"; n = 0; }
    prev = asn;
}
if n > 0 then print "ASN ", prev, " stuffed ", n, " times";
```

Filters: Community list sweep

```
for lc comm in bgp_large_community do {  
  if lc.asn = 64500 then  
    case lc.data1 {  
      ...  
    }  
  }  
}
```

Filters: Object-like calls

- dot-notation
- possible on any term, not modifying if assigned
- if on an lvalue **and** not assigned, modifying

Filters: Object-like calls

```
# old
clist n = filter(bgp_community, [(0,*)]);

# new
clist n = bgp_community.filter([(0,*)]);
```

Filters: Object-like calls

```
# old
bgp_community = add(bgp_community, (0,42));

# new
bgp_community.add((0,42));
```

Filters: Object-like calls

```
# old
bgppath mice = lure_mice();
int mouse = mice.last;

# new
int mouse = lure_mice().last;
```

Filters: Function return values

```
# This function returns an integer
function meow() -> int
{
  # This throws a syntax error
  return -empty-;
}
```

Filters: Function return values

- Explicit return type declarations
- Return type inference
- No multi-type-return functions

Filters: Interpreter performance

- Complete interpreter rework in BIRD 2
- Constant pre-computation
- Strict type checking
- More magical source code

```
# THIS IS A M4 MACRO FILE GENERATING 3 FILES ALTOGETHER.  
# KEEP YOUR HANDS OFF UNLESS YOU KNOW WHAT YOU'RE DOING.  
# EDITING AND DEBUGGING THIS FILE MAY DAMAGE YOUR BRAIN SERIOUSLY.
```

Filter / Channel: Automatic reloads

- ROA checks depend on ROA table contents
- routes reloaded automatically if possible
- selective reload in BIRD 3 only

Features in progress

- EVPN support
- ASPA checks
- Route aggregation
- Nexthop filtering
- CBOR/YANG API

Features expected only for v3

- Accessing protocol information from filters
- Dependent routes
- Complete CBOR/YANG API
- Mutable static routes
- BGP MRAI, BGPsec
- Multicast routing (?)

Please (don't) sort your AS Paths!

```
function as_path_sort(bgppath input) -> bgppath {
  bool sorted = true; bgppath temp = +empty+; int prev = 0;
  for int asn in input do {
    if prev > asn then { temp.prepend(asn); sorted = false; }
    else if prev != 0 then { temp.prepend(prev); prev = asn; }
    else { prev = asn; }
  }
  if prev != 0 then temp.prepend(prev);
  if sorted then return input;
  sorted = true; bgppath output = +empty+; prev = 0;
  for int asn in temp do {
    if prev = 0 then { prev = asn; }
    else if prev < asn then { output.prepend(asn); sorted = false; }
    else { output.prepend(prev); prev = asn; }
  }
  if prev != 0 then output.prepend(prev);
  if sorted then return output; else return as_path_sort(output);
}
```

QED

<https://bird.nic.cz/>
maria.matejka@nic.cz