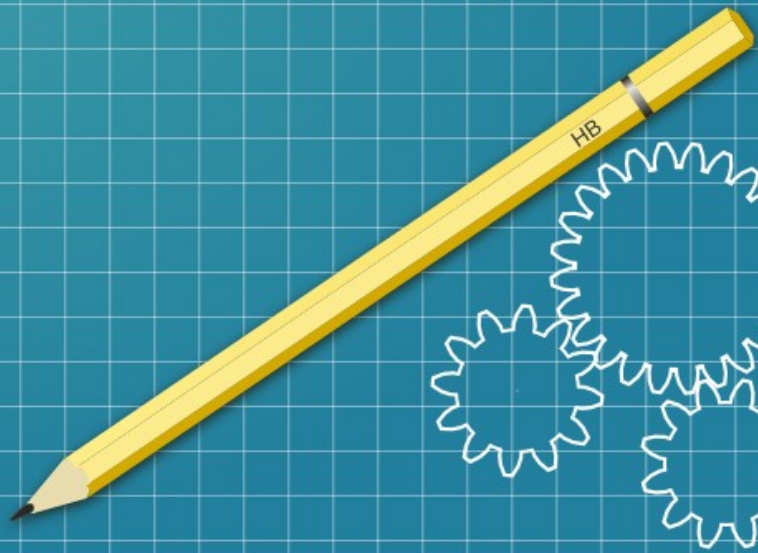
Dynamic Routing with BIRD (no, not the former UCC president)

B(erkley)
I(nternet)
R(outing)
D(aemon)



Quick revision (callback to week 2)

- BGP is a protocol for exchanging dynamic routes between routers
  - Designed for use between different *administrative domains*
- To exchange data with neighbouring routers, you need an existing connection
  - BGP doesn’t help there
  - We’re going to stick with static routes

Quick revision (callback to week 2)

- Revision: setting a single link between peers
  - Peer 1$ ip addr add dev eth0 10.1.2.1/24
  - Peer 2$ ip addr add dev eth0 10.1.2.2/24
    - Shared link and common subnet, so “directly connected”
- Revision: attaching “floating” addresses/ranges to current router
  - Just add to loopback
  - Router$ ip addr add dev lo 1.0.0.0/8

BGP

- Goal: each router knows “where to send next” for a packet with any valid address
  - Called “next hop”
- Information about paths exchanged
  - If I am “4” and have a route somewhere via “3 → 2 → 1”, then tell people about “4 → 3 → 2 → 1”
- Best path selected dynamically

BIRD

- Multi-protocol dynamic routing software
  - Doesn’t actually route anything, just sets up the Linux kernel’s own mechanisms
- Modular “protocol” plugins
  - Some are (almost) always enabled
  - Others are specific to the scenario
- Configured in /etc/bird/bird.conf
  - Each router has an “ID”, usually just an IP address of the router
  - Config option: router id 1.2.3.4;

BIRD

- Protocol “device”
- Allows learning connected interfaces
- Usually used

```
protocol device {
        scan time 10;
}
```

BIRD

- Protocol “direct”
- Learn the addresses/subnets directly connected to this router
- Often used, though usually limited to specific interfaces

```
protocol direct {
    interface “lo”;
}
```

BIRD

- Protocol “kernel”
- Bidirectional, can learn routes from the kernel (maybe static, or from another routing protocol?) or pass them back
- Can’t do much useful without this on a router

```
protocol kernel {
    persist;
    scan time 60;
    export all; # Use every route that other routers tell us about
    import none; # We’re only going to export routes we learned via direct
}
```

BIRD

- Protocol “bgp”
- One for each “peer”/neighbouring router
- Does the actual exchanging and building of routes
  - Uses AS numbers to work out where routes came from/via

```
protocol bgp neighbour_<n> {
    local as <insert_my_as_number_here>;
    direct; # we have a direct link to them
    neighbour 10.1.2.2 as <insert_their_as_number_here>;
}
```