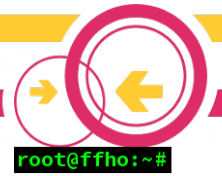
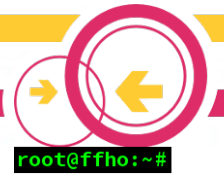


Contemporary Linux Networking

Confessions of a Professional Freifunker

DENOG9

Maximilian Wilhelm
<max@sdn.clinic>
@BarbarossaTM

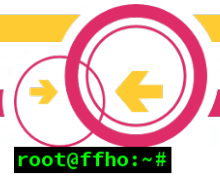


Who am I?

- Maximilian Wilhelm
  - @BarbarossaTM
- Senior Infrastructure Architect, Uni Paderborn
- Infrastructure Archmage, Freifunk Hochstift
- Fanboy of
  - (Debian) Linux
  - ifupdown2
- Networker
- OpenSource Hacker

Agenda

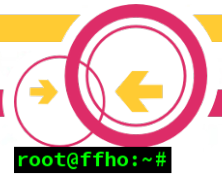
- The Old Gods and the New
- ifupdown2
- VRFs
- VXLAN
- VLAN-aware bridges
- MPLS
- DIY-SDN



THE OLD GODS

```
root@Stretch:~$ ifconfig
-bash: ifconfig: command not found
root@Stretch:~$ route
-bash: route: command not found
root@Stretch:~$ arp
-bash: arp: command not found
```

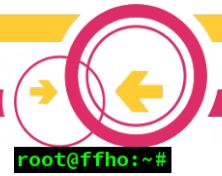
- vconfig
- ifenslave



AND THE NEW

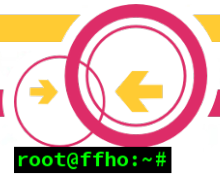
- iproute2 – Swiss Army knife for networkers
- Functions cleanly separated into subcommands

- `ip link` – L2: MTU, VLANs, LAGs, bridges, ...
- `ip addr` – L3 addresses
- `ip neigh` – ARP/ND
- `ip route` – Routing + MPLS



Old vs. New

```
vconfig add eth0 42

→ ip link add vlan42 link eth0
                     type vlan id 42

ifenslave bond0 eth0

→ ip link add bond0 type bond mode 4
→ ip link set master bond0 dev eth0

arp

→ ip -4 neigh
```



Old vs. New - Bridges

```
brctl addbr br0
```

```
➔ ip link add br0 type bridge
    [ forward_delay FORWARD_DELAY ]
    …
    [ vlan_filtering VLAN_FILTERING ]
    [ vlan_default_pvid VLAN_D_PVID ]
    …
    [ nf_call_iptables NF_CALL_IPT ]
    …
```

```
brctl addif br0 eth0
```

```
➔ ip link set eth0 master br0
```

Network interface configuration

- Classic `ifupdown` not easily automated
- Generating /etc/network/interfaces simple
- How to reload?
  - »service networking restart« disruptive
  - No tool for “reload” present
  - Isn't trivial to build
- ➔ CumulusNetworks Ifupdown2
  - Rewrite of ifupdown in Python
  - https://github.com/CumulusNetworks/ifupdown2

ifupdown2

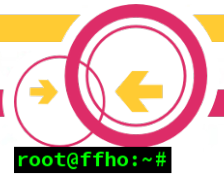
- No full feature parity with ifupdown (yet?)
- Shipped with batteries included
  - dependency resolution
  - `ifreload`
  - VRFs
  - VXLAN
  - VLAN-aware bridges
- Not (yet) ~~supported~~merged:
  - ppp

ifupdown2 Patches

- Easy to extend, thanks to Python
- Upstream open for ideas (Hi Julien & Roopa)
- Added support for
  - B.A.T.M.A.N. interfaces
  - Tunnel (GRE, SIT, IPIP, GRETAP)
- Open Pull-Requests for
  - Condoning bridge interfaces for configuration
  - Setting phys-dev for VXLAN
  - Setting vEth peer name

VRFs

- Independent routing instances
  - L3-VPNs
  - Usually in combination with MPLS
- Related features
  - Policy-Routing (since Kernel 2.2)
    - Old and busted
    - Management headache
  - Network Namespaces (Kernel 2.6.24++)
    - Sometimes “too much” separation



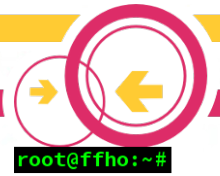
VRFs on Linux

- Separation for Layer3 communication
- VRF interface is master for “real” interfaces
  - Defines routing table for VRF
- Since Kernel 4.[345] (use >= 4.9)

`https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/networking/vrf.txt`

`https://cumulusnetworks.com/blog/vrf-for-linux/`

`https://de.slideshare.net/CumulusNetworks/operationalizing-vrf-in-the-data-center`



VRFs on Linux

```
ip link add VRF_DEVICE type vrf
            table ID

ip link set dev DEVICE
            master VRF_DEVICE
```

Note:

- Device routes move from table `main` and `local` to table `$ID`

VRFs with ifupdown2

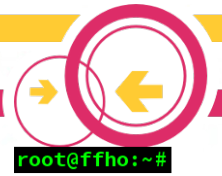
```
auto eth0
iface eth0
        address 185.46.137.163/25
        address 2a00:13c8:1000:2::163/64
        gateway 185.46.137.129
        gateway 2a00:13c8:1000:2::1
        vrf vrf_external


auto vrf_external
iface vrf_external
       vrf-table 1023
```

inter VRF Communication

- Requires vEth pair
  - Like a virtual network cable within the box
- A end in main VRF, Z end in VRF “foo”
- Usual routing
  - Static
  - Bird talking BGP to itself
  - ...

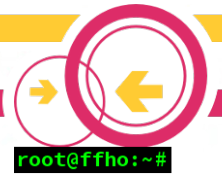
vEth interfaces w/o + w/ ifupdown2

```
ip link add VETH_END1 type veth
            peer name VETH_END2

iface veth_ext2int
    link-type veth
    veth-peer-name veth_int2ext
    vrf vrf_external

iface veth_int2ext
    link-type veth
    veth-peer-name veth_ext2int
```



VXLAN

- “Ethernet over UDP”
  - Or: “Poor mans approach to MPLS”
- Designed as Layer2 overlay for DCs
  - Multi-tenant Overlay over IP-Fabric
  - 24Bit VNI => 16M Instances
  - Unicast/Multicast communication
    - Read: VLL / VPLS
  - Endpoints = VTEP (VXLAN Tunnel End Point)
- RFC7348

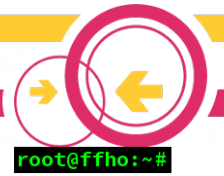


VTEPs on Linux

```
ip link add DEVICE type vxlan id ID
[ dev PHYS_DEV ]
[ { group | remote } IPADDR ]
[ local { IPADDR | any } ]
[ … ]

bridge fdb show [ brport DEVICE ]
```

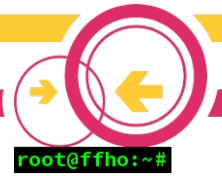
VTEPs with ifupdown2

```
# vx_v2001_padcty

auto vx_v2001_padcty
iface vx_v2001_padcty
        vxlan-id          1310977
        vxlan-physdev     vlan2001
        vxlan-svcnodeip 225.20.1.1
        #
        hwaddress f2:00:c1:01:20:01
        mtu 1560
```



VLAN-aware bridges

- VLANs and bridges have been a challenge
- That ain't true no more
  - `echo 1 > /sys/class/net/br0/bridge/vlan_filtering`
  - Now it's a “regular switch”
- Configured with `bridge` utility from `iproute`
- Simple KVM/Qemu hook for VLAN assignment
  - https://github.com/FreifunkHochstift/ffho-salt-public/blob/master/kvm/qemu-hook



Bridge utility

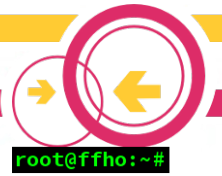
```
bridge vlan { add | del }
       vid VLAN_ID dev DEV
       [ pvid ] [ untagged ]
       [ self ] [ master ]


bridge vlan show [ dev DEV ]
                 [ vid VLAN_ID ]
```

- Related:

  `bridge fdb [...]`

VLAN-aware bridges w/ ifupdown2

```
iface br0
    bridge-ports bond0
    bridge-vlan-aware yes
    bridge-vids 1013 4002

iface bond0
    bridge-vids 100 101 200 201 1013 2000 [..]

iface cr02_eth1
    bridge-vids 1013 2000 2004 2006 3002

iface br0.1013
    address 10.132.252.22/28
[…]
```



MPLS

- Forwarding path available in vanilla kernel
  - Use >= 4.9
- Requires iproute >= 4.3

  `ip -f mpls` or `ip -M`
- Enable use of labels up to `n`

  `sysctl -w net.mpls.platform_labels=n`
- Enable MPLS decap on `$iface`

  `sysctl -w net.mpls.conf.$iface.input=1`

MPLS

- Push

```
ip route add 10.23.42.0/24 encap
mpls 100 via inet 192.168.42.23
```

- Swap (100 → 200)

```
ip -f mpls route add 100 as 200 via
inet 192.168.47.11
```

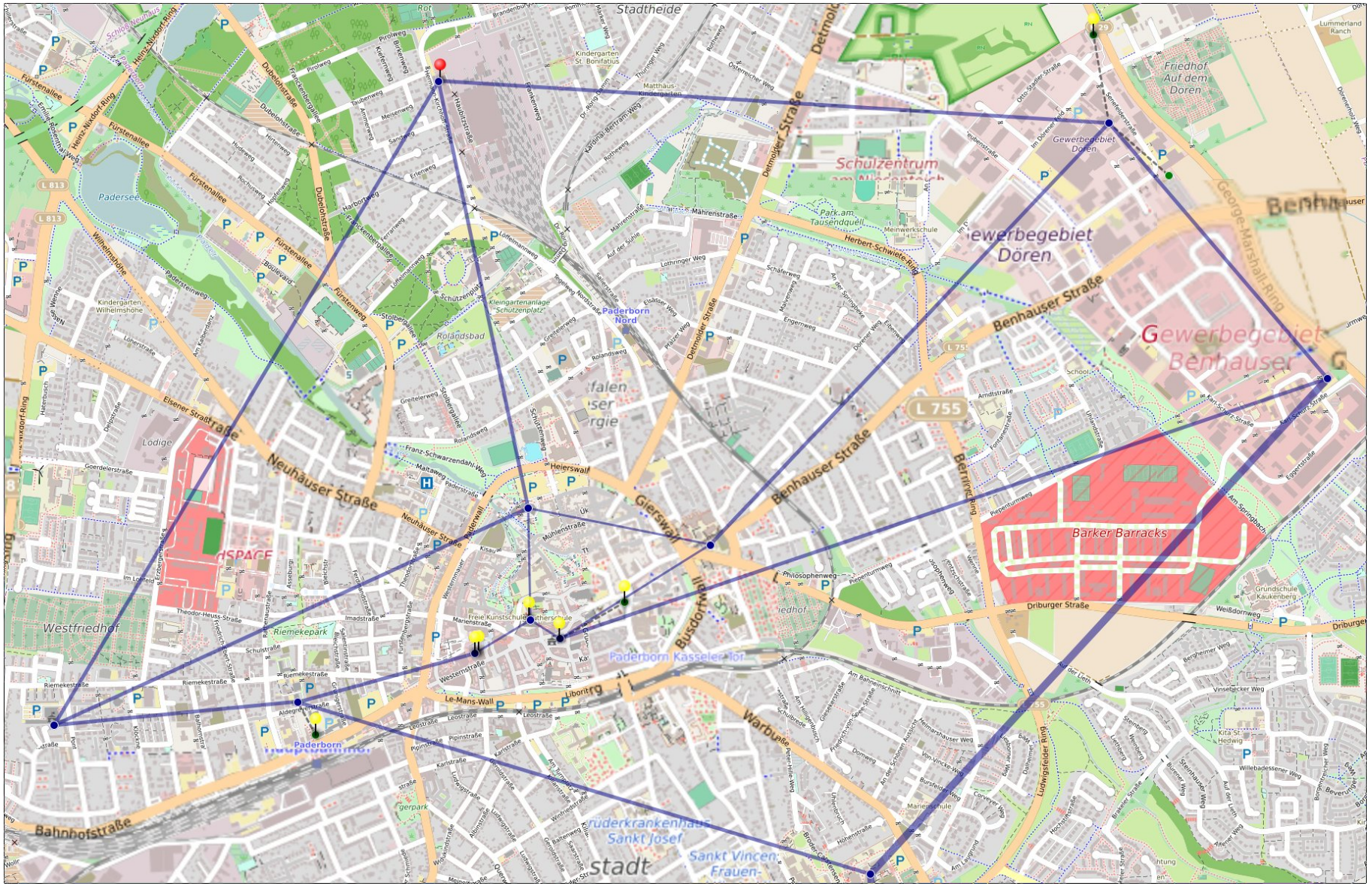
- Pop

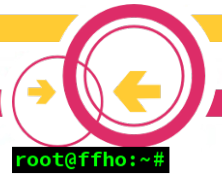
```
ip -f mpls route add 300 dev lo
```

The SDN part

Disclaimer: Font on special request of AbraXXL

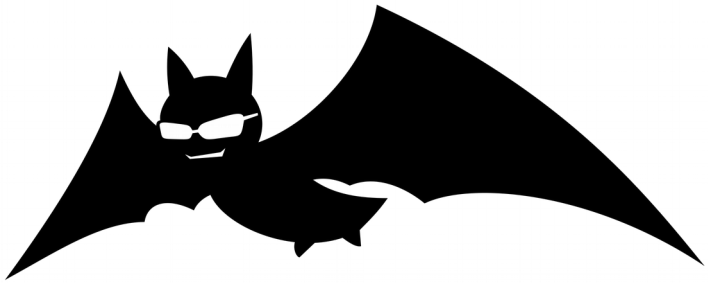
Wireless Backbone (planned)





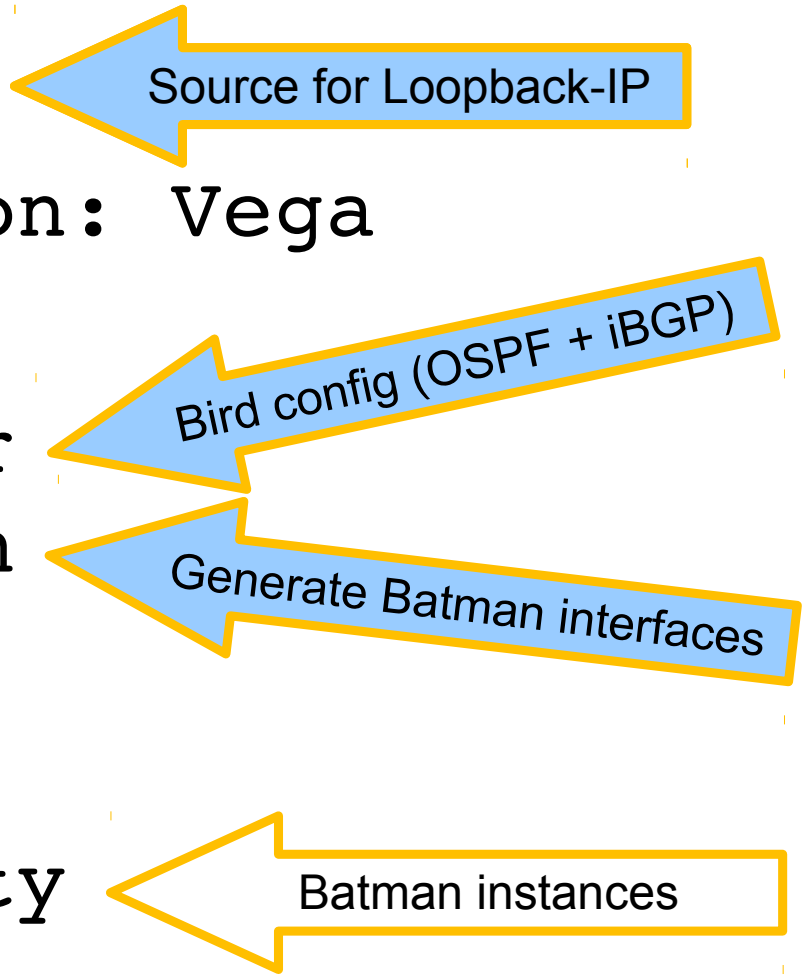
~~Cyber Supply Chain~~

SDN ingredients

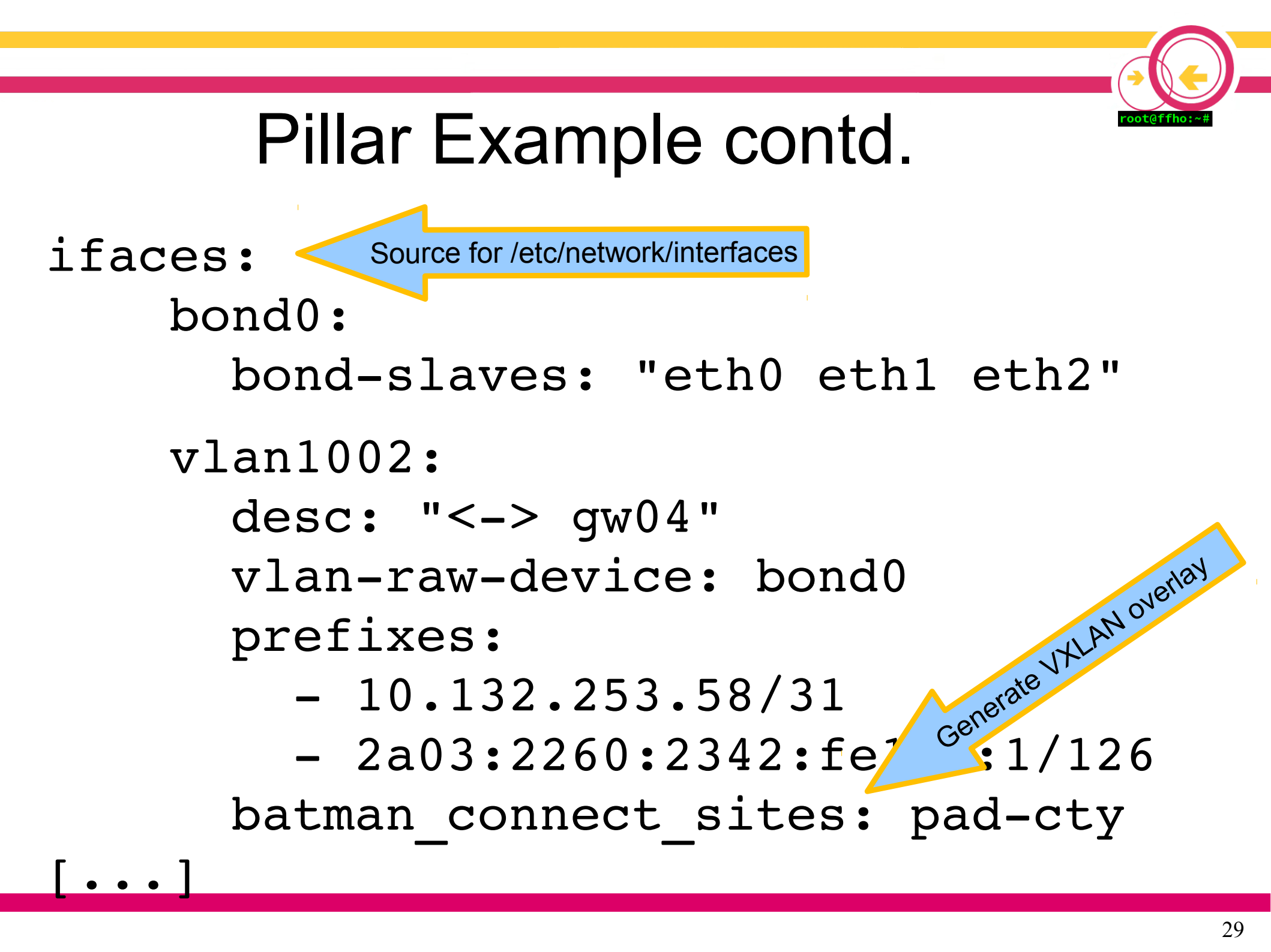


Pillar Example

```
bbr-vega.in.ffho.net:
  id: 198
  sysLocation: Vega
  roles:
    - router
    - batman
    - bbr
  sites:
    - pad-cty
```



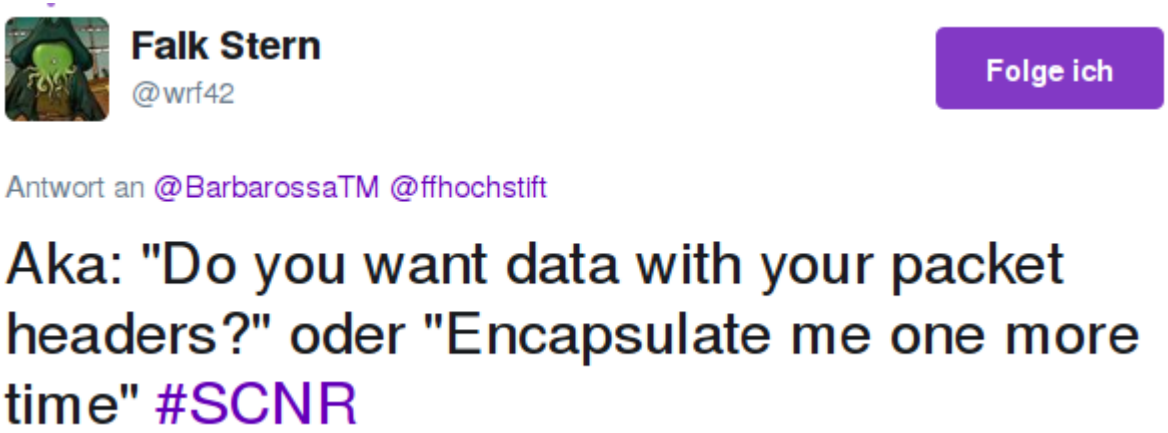
Pillar Example contd.




```
ifaces:
    bond0:
      bond-slaves: "eth0 eth1 eth2"

    vlan1002:
      desc: "<-> gw04"
      vlan-raw-device: bond0
      prefixes:
        - 10.132.253.58/31
        - 2a03:2260:2342:fe1   :1/126
      batman_connect_sites: pad-cty
[...]
```

IPoBATMANoVXLANoIPoVLANoRF

- Wait, what?



Lessons Learned

Offloading

- Difference between 4KB/s and 40MB/s...

```
for iface in eth0 eth1; do
    for feature in sg gro gso tso; do
        ethtool --offload ${iface}
                  ${feature} off
        done
done
```

https://downloadmirror.intel.com/22919/eng/README.txt

OpenVPN vs. VRFs

- Lots of OpenVPN tunnels
- OpenVPN tunnel should use VRF “external”
- Needed a small patch

```
setsockopt (sd, SOL_SOCKET,
SO_BINDTODEVICE, dev, strlen(dev)
+1);
```

- https://github.com/OpenVPN/openvpn/pull/65
  - Hi Gert, are you here?

Systemd + OpenVPN vs. ifup

- Lots of OpenVPN instances
- `up /etc/openvpn/ifup`
  - `ifup “$1”`
- Thanks to systemd all starting in parallel
  - Some ifup calls in parallel
  - Nearly no IPs configured anywhere
  - Damn

→ `flock –exclusive –wait 30`

Further Reading

- Salt-Orchestrated Software Defined (Freifunk) Network (german)
  - https://www.slideshare.net/BarbarossaTM/software-defined-freifunk-backbones-78288014
- Blog series DIY-SDN with OSS
  - https://blog.sdn.clinic/2017/09/building-your-own-software-defined-network-with-linux-and-open-source-tools/
- #routingdays – Learn to build the Internet
  - https://blog.sdn.clinic/2017/09/ffrl-routingdays-learn-to-build-the-internet/

Questions? Remarks?

Tell me:

Maximilian Wilhelm
<max@elitepeer.de>
@BarbarossaTM

Freifunkromantik