VANDERVECKEN

joshb@google.com

What is VANDERVECKEN?

VANDERVECKEN is an integrated, fully functional OpenFlow controlled LSR/router - boot up an ISO on a PC, supply OpenFlow compatible switch hardware - and you have a hardware forwarding LSR/router that can speak BGP, IS-IS and LDP. VANDERVECKEN is entirely open source - rather than spend time putting together an entire software stack, you can proceed immediately to experimenting with new features and changing the code.

VANDERVECKEN is currently based on RouteFlow (https://sites.google.com/site/routeflow/) using the python based Ryu stack. It has been most tested with Pica8 (http://pica8.org/) 3290 and 3780 switches, running PicOS 1.6.1, using OpenFlow 1.2. Please see also the FAQ section at the end of this document.

Control PC requirements

- 4GB RAM
- DVD drive
- Hard disk (if you want to keep changes across reboots)
- Internet access (helpful but not required)

Current Features

- Entirely open source controller stack (as of October 2013, https://github.com/routeflow/RouteFlow vandervecken branch).
- Live CD implementation (http://www.vandervecken.com/vandervecken.iso)
- Hardware forwarding (via an external OpenFlow switch)
- Supports BGP, IS-IS, and more (works with Quagga, though other E/IGP implementations are compatible)
- More than one switch per controller (distributed router, AKA CARDIGAN), with dedicated links between switches

Future Features

- IPv6 support (via OpenFlow 1.2 support) - soon, being integrated
- MPLS/LSR support (via OpenFlow 1.2 support and Quagga/LDP) - soon, being integrated

FAQ

What OpenFlow versions does VANDERVECKEN support?

As of October 2013, OF1.3 only.

How does VANDERVECKEN differ from RouteFlow?

VANDERVECKEN has a few features, such as distributed router support and some scalability/ performance changes that are not yet in the main RouteFlow code base. Those changes will likely be integrated into RouteFlow over time.

How do I configure BGP, etc?

rfvm1's route and ARP tables are effectively replicated in the OpenFlow switch. So, any means of adding or removing a route (whether via "ip route" static routes, or via Quagga) will be replicated. Quagga is included on the ISO - see http://www.nongnu.org/quagga/ for configuration instructions.

How do I view flows on the OpenFlow switch?

It depends what kind of switch you are using; on the Pica8 3290, you can type (assuming you have used the default "switch1" configuration below) **`ovs-ofctl dump-flows switch1.`**

How do I change the IP addresses assigned to each switch port?

At the moment IP addresses are automatically assigned by **`projectw.sh`**. **`projectw.sh`** can be modified not to do this. A configuration option to do this better will be added.

How do I install the liveCD on a hard disk?

You will need to, as root, install X and run ubiquity (requires root access). The install option from the liveCD boot menu does not currently work.

- **`sudo bash`**
- **`apt-get install xserver xdm ubiquity`**
- **`/etc/init.d/xdm start`**
- login as root
- start an xterm, run **`ubiquity`**

Alternatively you can use ssh X forwarding to install remotely.

- **`sudo bash`**
- **`apt-get install ubiquity openssh-server`**
- start an xterm locally
- ssh -X root@<ip_address_of_cardigan>
- run **`ubiquity`**

Logging into VANDERVECKEN

- User is always projectw, password projectw - root password is projectw
- When logging into rfvm1, b1, or b2 virtual machines - user is root, password is root

How to run VANDERVECKEN as a standalone demonstration

You can run VANDERVECKEN as a standalone, single PC demonstration system (Eg, for teaching) - you don't need an external switch or even Internet access.

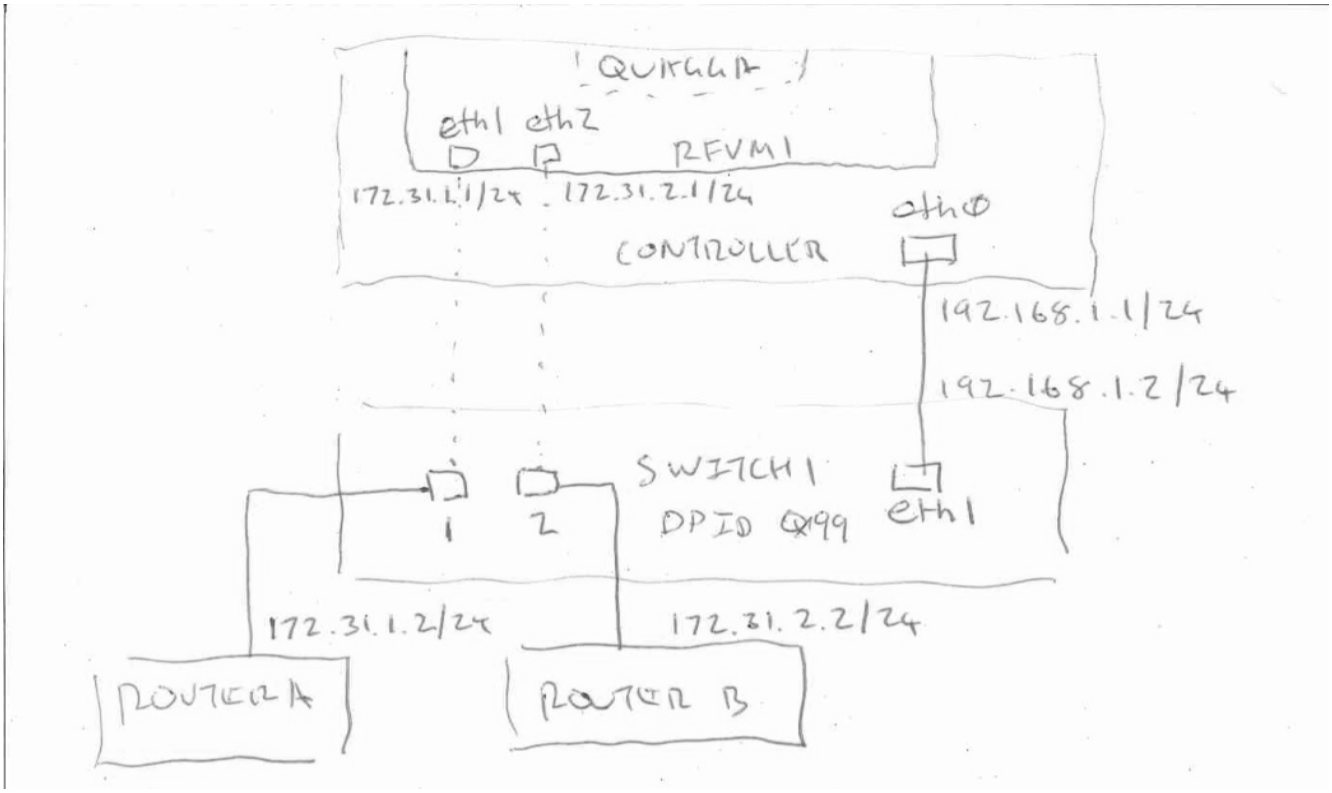
1. Boot ISO on PC
2. On first console, **`sudo bash`**
3. **`./bootstrap-lxc.sh`**
4. **`./project --ryu`** (wait for `INFO:rfserver:Mapping client-datapath association (vm_id=0x12a0a0a0a0a0, vm_port=1, dp_id=0x99, dp_port=1, vs_id=0x7266767372667673, vs_port=1`)
5. On second console, **`./b1.sh`**, log in as root.
6. **`ping 172.31.2.2`**

You have effectively set up the RouteFlow tutorial described here: https://sites.google.com/site/routeflow/documents/first-routeflow.

How to configure VANDERVECKEN as a simple router

This example uses a Pica8 3290 switch - substitute configuration per your switch. The system provides connectivity between two router/hosts, A and B.

1. Cable your PC and switch together (Eg, from a dedicated Ethernet port on the PC, to the “eth1” port on the Pica8 3290).
2. Boot ISO on PC
3. Login as projectw, then **`sudo bash`**
4. **`./bootstrap-lxc.sh`**
5. **`vi projectw.sh`**
    a. set **STARTBVMS** to 0.
    b. set **SWITCH1DPID** to the OpenFlow datapath ID of your switch (or configure switch to use the default, which for VANDERVECKEN is 0x99)
    c. set **DPPORTS** to the number of ports you will use on the switch (default is the first 2)
6. **`./projectw.sh --ryu`** (the controller is now waiting for the switch to connect).
7. Assign an IP address to the PC and to the switch, and confirm that the PC can reach the switch (Eg, 192.168.1.1 to the PC, 192.168.1.2 to the switch).
8. Configure the switch to connect to the PC (see example config for 3290).



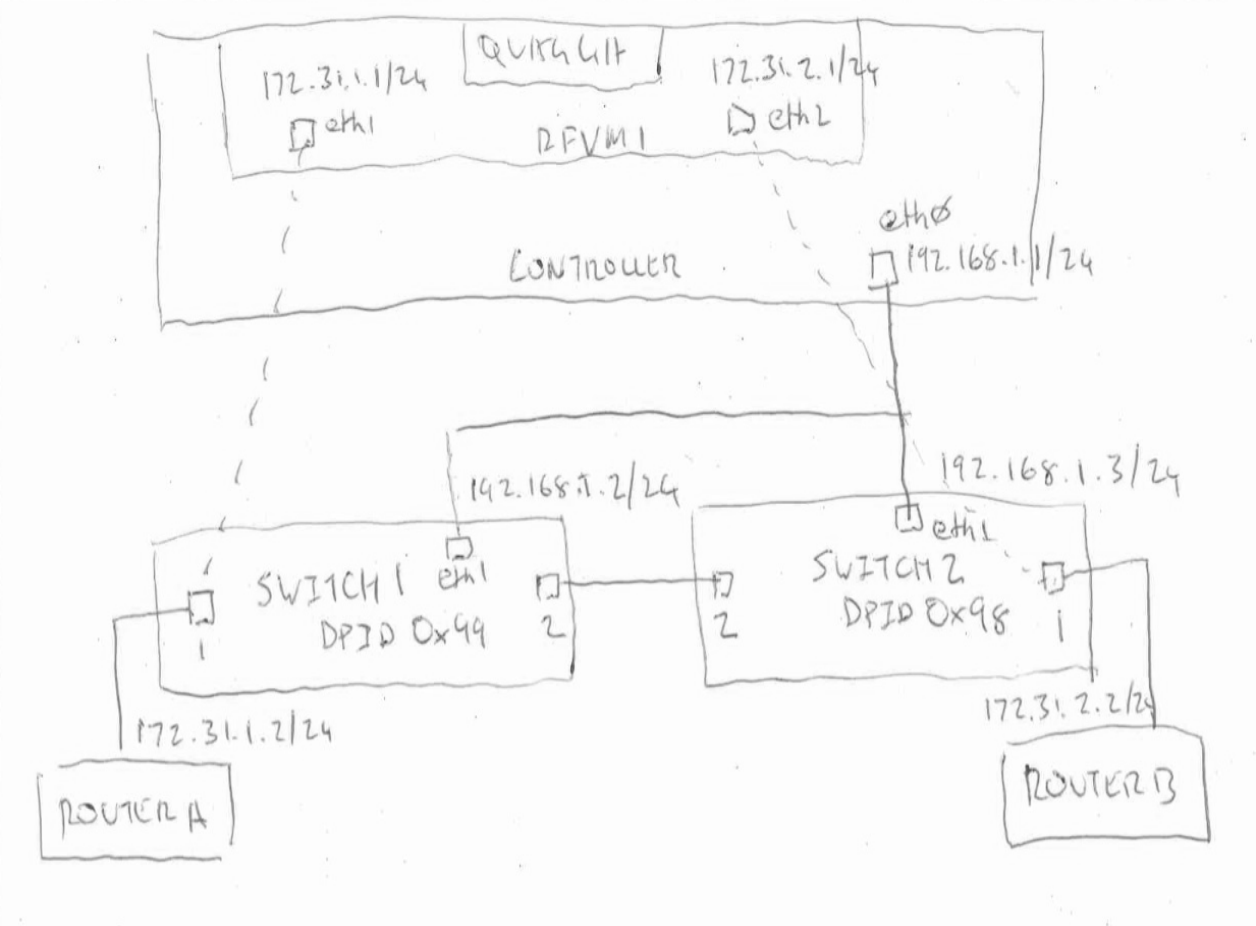
How to configure VANDERVECKEN as a distributed router

In this example we are configuring two switches with a single interswitch link (ISL).

Use the same instructions as for a single switch (above), but make an additional change to **projectw.sh** (adding the line highlighted in bold):

```
echo "vm_id,ct_id,dp_id,dp_port,eth_addr,rem_ct,rem_id,rem_port,rem_eth_addr" > $RFSERVERINTERNAL
echo 12A0A0A0A0A0,0,$SWITCH1DPID,2,12:a1:a1:a1:a2:01,0,$SWITCH2DPID,2,12:a1:a1:a1:a2:02
>>$RFSERVERINTERNAL
```

You will also have to change DPID of switch2 to 0x98.



Configuring the Pica8 3290

This config is suitable for the above examples, using the first two ports. The PC controller should be connected to the Pica8 port labelled “eth1”.

- Select “system shell” from the Pica8 boot menu option.
- **`vi /usr/bin/startapp.sh`,** add `exec /bootsw.sh` after `#!/bin/sh`.
- **`vi /bootsw.sh`**, add the following:

```
#!/bin/sh
/sbin/ifconfig eth0 192.168.1.2/24
/usr/sbin/sshd
```

- **`chmod +x /bootsw.sh`**
- **`vi /bootovs-3290.sh`**, add the following:

```
#!/bin/sh
MYIP=192.168.1.2
CTLIP=192.168.1.1
GWIP=192.168.1.1
DPPORTS=2
EP=ge
OPTS=""
VSCTL="ovs-vsctl --db=tcp:$MYIP:9999"
cd /
ifconfig eth0 $MYIP netmask 255.255.255.0
route add -net default gw $GWIP
rm -f /ovs/ovs-vswitchd.conf.db
ovsdb-tool create /ovs/ovs-vswitchd.conf.db /ovs/bin/vswitch.ovsschema
ovsdb-server /ovs/ovs-vswitchd.conf.db --remote=ptcp:9999:$MYIP &
ovs-vswitchd tcp:$MYIP:9999 --pidfile=pica8 -- &
$VSCTL add-br switch1
$VSCTL set bridge switch1 datapath_type=pronto
for i in `seq 1 $DPPORTS` ; do
    PORT=$EP-1/1/$i
    $VSCTL add-port switch1 $PORT -- set Interface $PORT type=pronto
$OPTS
done
$VSCTL set Bridge switch1 other-config:datapath-id=0000000000000099
$VSCTL set-controller switch1 tcp:$CTLIP:6633
```

- **`chmod +x /bootovs-3290.sh`**
- reboot the switch via **`reboot`** - switch should come up with assigned IP address
- **`/bootovs-3290.sh`** (switch should start trying to connect to controller)

Configuring the 3780

See instructions for 3290, but set `EP=te`. For 1G SFPs, set `OPTS="options:link_speed=1G"`