Testing Plan

Test plan includes testing of three sections:

1. Test of the switch/contoller API
2. Test of the NDDI(?) under no errors.
    1. Back-end
    2. Front-end
3. Test of Failure handling
    1. controller failure
    2. app failure (HA)
    3. network failure (backup path).

Test of the switch/controller API

Tests to be done:

1. OpenFlow synchronous messages
    1. Do they propagate?
    2. Do the propagate in time (latency < 50ms).
    3. Can it handle error conditions gracefully
    4. Error reporting is sane
2. OpenFlow asynchronous messages
    1. Do they propagate?
    2. Do they propagate in time?
    3. What are the performance limits on this processing.
3. OpenFlow table consultation (including caching if applicable)
4. Setting up QoS settings
    1. Can you access the CLI of the switches via the api?
    2. Can you transform the QoS non-openflow settings into a same API?
5. Support of plurality of connections
    1. All connections serviced in time?
    2. Performance impact on the controller,switches and API when under load of multple users.
6. Graceful client handling.
    1. Support fast and slow users of the api,without affecting the performance of the fast ones.
    2. Support for ungraceful disconnection of clients
7. Support of ungraceful degradation of controller/switches.
8. Has a call to get current switch topology.

Test harness

Most of these tests can be done without access to specialized hardware, the exception are the QoS settings as they are hardware dependent. All the non-QoS test can be scriptable and testing automated.

Test of OE-SS under no errors (Backend)

1. Can create a a two point vlan
    1. same vlan tag externaly
    2. tag translation
    3. QinQ (if Possible)
2. Can create a multipoint vlan
    1. same vlan externally
    2. tag translation
    3. all QinQ
3. single external IDC (stage 2)
    1. two point vlan with external idc
    2. multipoint vlan with extternal idc
4. multiple external IDC
    1. two point vlan with external idc (network path)
    2. multipoing vlan with internal and external drops
    3. mutipoint vlan to three external idc
5. Accounting data generation

All of these are system tests as they report the interaction of the software and hardware components. The testing of the system can be automated, with the interactions being: the OE-SS API, the test endpoints the controller/switch API.

Test of NDDI under no errors (front end)

Same as back-end and in addition:

1. Scheduding
2. creation of backup paths
3. Accounting(?)

Test of Failure OE-SS under failure:

These are system tests, which could be scriptable by using glimmerglass equipment to simlate conection errors on the

Test of controller failure

1. behavior on primrary/backup unplanned shutdown
2. Effect of degraded network communications.
3. Bahavior on failure on conectivity between switch an one of the two controllers
4. behavior on management network conectivity on muliple swithces

Test of APP behavior

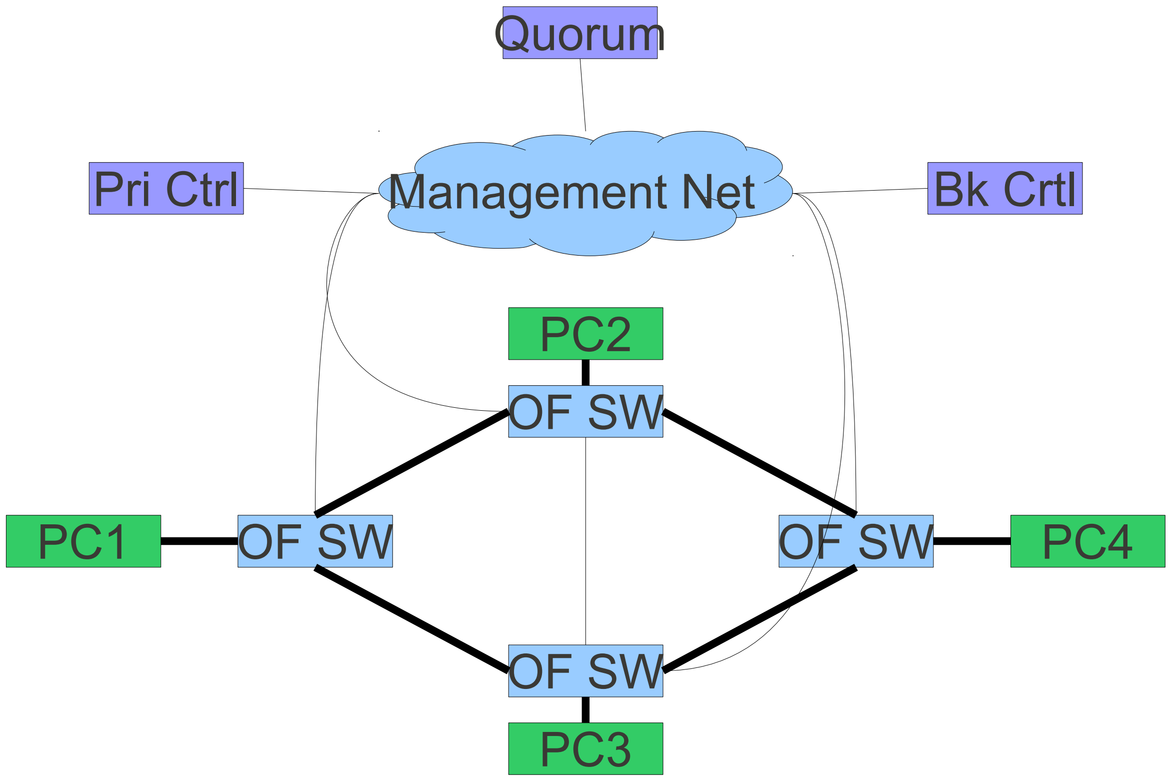
1. behavior on primary/backup contoller planned shutdown
2. behavior on primrary/backup unplanned shutdown
3. behavior communication failure between primary and secondary (split brain).
4. Effect of degraded network communications.
5. Bahavior on failure on conectivity between switch an one of the two controllers
6. behavior on management network conectivity on muliple swithces

Test of non-management network failures

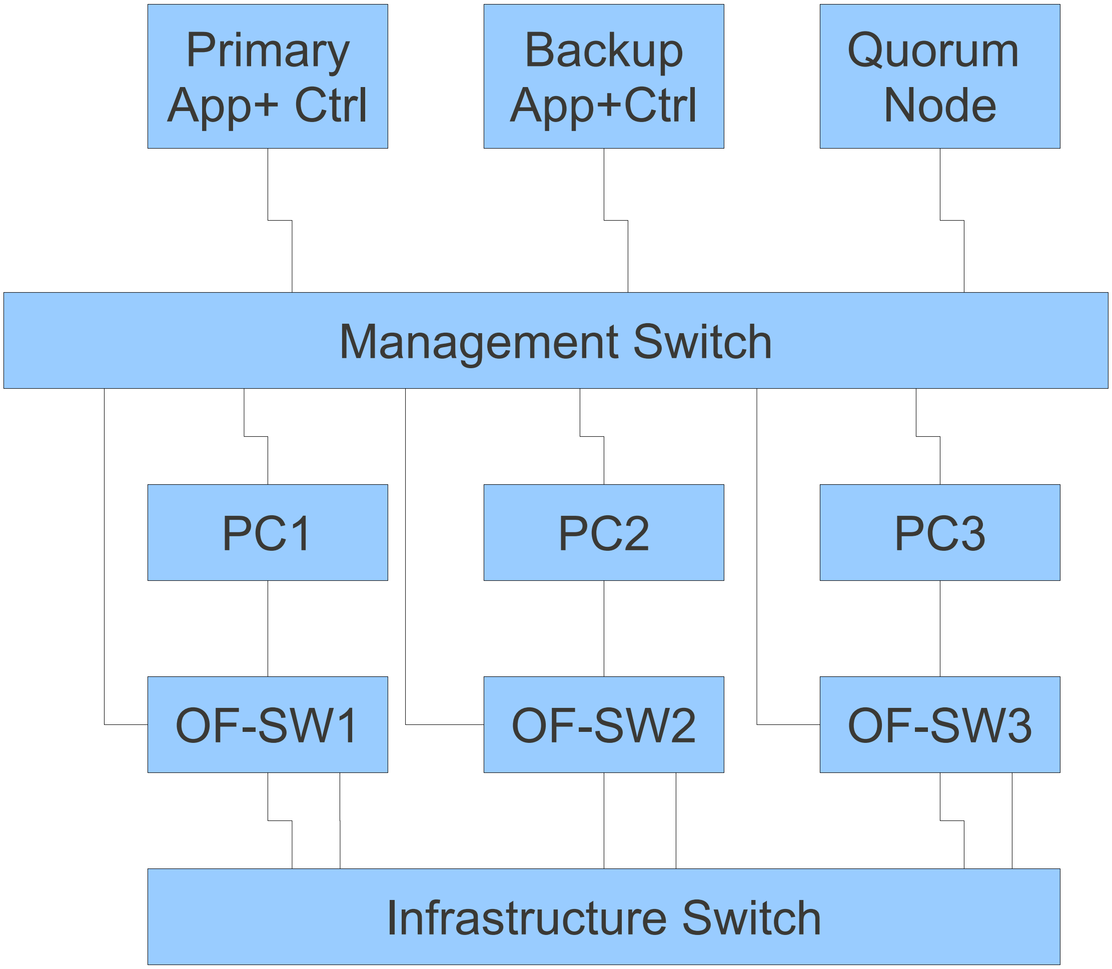
1. detection of network failures
2. backup path restoration
   1. primary path recovery
3. automatic path restoration
   1. primary path recovery

Test Environment

It would be extremely useful to do as close as possible of a full system test before the system is deployed on the field. The following is a logical diagram for full system test. In the picture the purple boxes are the controller/API/Fronteed system, the green boxes represent test PCs and the blue boxes openflow switches.



Another view would be:



Each of the test PCs, must have a direct connection to at least one switch. Furthermore, each of the test PC should have a 10GB interface to the switch (This is for the case of using the pronto switch).

I can see several options to do this (in order or preference).

Number	Option	Possible Now	Number of HW PC	Price (US$)
1	Single PC to virtualize all	NO	1	4628.00[1] + optics
2	Termination PC in duples and controller ina	Yes[2]	3	(+900?)
3	Real hardware for each	Maybe	5(4)	
4	Real Hardware for each component.	Maybe	7	

[1] For a DELL 510 with two dual 10GB cars and 12GB of RAM. No optics are included on the price. We could by the cards/optics from other network and probably drop about US$ 500 from the price.

[2] Currently we have two machines one with 2 10GB cards and one with 10GB cards. However: there are only two optics, the cards are from NLR, the PC are not redundant (no raid) and the PC are out of warranty.

The reason of the for prefering option 1 is that the other options are not long term options as the hardware has been borrowed from other networks and is mostly out of warrranty. Once there are no switches on the testlab, it would be beneficial to have some way to test the system using software switches, a single PC with four nics for the testing would be sufficient.

Short term

This is what we can do now with the current hardware available. This assumes we will be using the IBM switches.
For the test PC, we can continue using nddi-test1 and nddi-test2(or another box with a 10GB card). NDDI-test1 can be reconfigured to have four(4) copper gig interface cards (three connected to the switch, one for management) and one 10GB fiber card. The primary limitations of NDDI-test1 for virtualization is the amount of memory, followed by its lack of redundancy. We will use NEC Switches (both of them)

The mapping of PC to devices will be:

Device in diagram	Hardware Device
Pri Crtl	Nddi-test1 (host OS)
PC1	Nddi-test1 (vm), using 10G
PC2	Nddi-test1 (vm)
PC3	Nddi-test1(vm)
Bk Crtl	Nddi-test2 (vm)
Quorum	Nddi-test2(vm)
PC4	Nddi-test2(vm)
OF-SW1	NEC Switch1
OF-SW2	NEC Switch2
OF-SW3	Not mapped, pc connection mapped to pc3
OF-SW4	Pronto switch(?)

This setup will allow us to test most of the setup without backup path and multiple location multi-path.

Long term

We need to have at least three switches for the complete testing of the failover and backup plans. I think at least 2 PC are needed.
PC 1.a → host for all the controller setup including the quorum
PC 1.b → host for the load PC
PC 2 → host to simulate the switches (when there are no switches on the lab).

Requirements:
PC1a → at least 5GB of ram (1 for host, 512 for quorum, 1.5G for each controller), harware

virtualization (ie at least a dell 410).
PC 1b → at least 4GB (8B preferred) of RAM (512MB for each vm), hardware virtualization, (at least 2 10GB connections (4 preferred), at least 5 network interfaces total.
PC2 → at least 5 nics.