How about physical_network in TOSCA for SR-IOV

Problem Description:

In a nutshell, **How can we provide the physical_netwrok information** (which is required by OpenStack when to use SR-IOV) **in TOSCA VNFD?**

In OpenStack, when to use SR-IOV, physical_network is needed both by the administrator and the users.

First the administrator need to do some configurations for SR-IOV for various openstack services, e.g.

nova configurations (i.e. nova.conf):

```
pci_passthrough_whitelist={"address":"0000:08:00.0", "physical_network":"physnetNFV"}

scheduler_default_filters = <defaultlist>, PciPassthroughFilter
```

neutron ml2 configurations: (ml2.conf)

```
[ml2_sriov]
supported_pci_vendor_devs = 8086:10fb
agent_required = False
[sriov_nic]
physical_device_mappings = physnetNFV:eth1
```

To use SR-IOV, users first need to create a network using the appropriate physical_network according the the administrator's configuration, then create a port and launch a VM by that port, e.g.

```
openstack network create --provider-network-type flat --provider-physical-network physnetNFV sriov-net

... ...

openstack port create --vnic-type direct --network sriov-net ... sriov-port

openstack server create ... ... --nic --port-id=<uuid of port sriov-port created above> ...
```

In Heat, we could provide the physical_network information in the template as:

```
sriov_net:
  type: OS::Neutron::Net
  properties:
    name: sriov_network
    network_type: flat
    physical_network: physnetNFV
/** Subnet details are not provided for brevity**/
sriov_port:
   type: OS::Neutron::Port
   properties:
     name: sriov_port
     network_id : { get_resource: sriov_network}
     binding:vnic_type: direct
virtual_firewall_appliance:
   type: OS::Nova::Server
   image: ....
   networks:
      - port: {get_resource: sriov_port}
```

Now the question is how about the TOSCA VNFD.

Proposals:

Here we came up with 3 proposals about how to provide such **physical_netwrok** information in tosca VNFD.

1. Using physicalNetwork property specific in ONAP R2 DM

`In ONAP R2 DM, we have the physicalNetwork property in virtual link profile data type, and we can use that.`

```
   VL_mux_gw_private_net:
      type: tosca.nodes.nfv.VnfVirtualLink
      properties:
        connectivity_type:
          layer_protocol: ipv4
        vl_profile:
          ... ...
          networkName: sriov_net
          cidr: ......
          dhcpEnabled: false
          physicalNetwork: physnetNFV
```

Pros:

- no need to change the tosca work flow within ONAP for policy/oof

Cons:

- VNF provider should NOT know the underlying cloud configuration information, such as physical_network information(Yes, heat template has the same problem), before they create their own VNFD.
- The physicalNetwork property above is not in IFA011 or SOL001, so it might be removed in the future.
- SR-IOV requirement is specified as HPA features in CP node in tosca VNFD, ONAP policy generator needs to back reference from CP to corresponding VL to get the relevant information.

2. Put the information in SRIOV specific HPA feature

Since physical_network information is for SR-IOV only in our case, we could add this information into the HPA feature's configurationValue, in the format of <interface_type>:<physical_network>

```
  Cp_SRIOV:
      type: tosca.nodes.nfv.VduCp
      properties:
        ... ...
        virtual_network_interface_requirements:
          - name: sriov
            support_mandatory: true
            network_interface_requirements:
              interfaceType: '{"schemaVersion": "0","schemaSelector": "","hardwarePlatform": "generic","
mandatory": "true", "configurationValue": "SR-IOV:physnetNFV"}'
            nic_io_requirements:
              pciVendorId: '{"schemaVersion": "0","schemaSelector": "","hardwarePlatform": "generic","
mandatory": "true", "configurationValue": "8086"}'
              pciDeviceId: '{"schemaVersion": "0","schemaSelector": "","hardwarePlatform": "generic","
mandatory": "true", "configurationValue": "10fb"}'
```

Pros:

- no need to change the tosca work flow within ONAP for policy/oof
- No inconsistency to IFA011 or SOL001.
- policy generator can just parse the VNFD Cp definition and generate the corresponding rules for OOF.

Cons:

- VNF provider should NOT know the underlying cloud configuration information, such as physical_network information(Yes, heat template has the same problem), before they create their own VNFD.

3. Put the information as a runtime property/input in HPA feature

Ideally, the VNF provider should NOT know the underlying cloud configuration information when he/she create the onboarding VNFD. So we can put the physical_network information as a runtime property/input by using tosca function.

```
inputs:
    sriov_physical_net :
      type: string
      description: physical network for sriov
   ......
node_templates:
    Cp_SRIOV:
      type: tosca.nodes.nfv.VduCp
      properties:
        ... ...
        virtual_network_interface_requirements:
          - name: sriov
            support_mandatory: true
            network_interface_requirements:
              interfaceType: { concat : [ '{"schemaVersion": "0","schemaSelector": "","hardwarePlatform":
"generic","mandatory": "true", "configurationValue": ',
                                          '"SR-IOV:', { get_input: sriov_physical_net }, '"}'
                                        ]
            nic_io_requirements:
              pciVendorId: '{"schemaVersion": "0","schemaSelector": "","hardwarePlatform": "generic","
mandatory": "true", "configurationValue": "8086"}'
              pciDeviceId: '{"schemaVersion": "0","schemaSelector": "","hardwarePlatform": "generic","
mandatory": "true", "configurationValue": "10fb"}'
```

Pros:

- VNF provider doesn't need to know the underlying cloud configuration information
- No inconsistency to IFA011 or SOL001.

Cons:

- Need to change the ONAP internal work flow. Policy can not generate the corresponding policy rules for OOF until SDNC has finished its operations and provide the tosca input, or unless the service designer can provide the default value of the input parameter.
- I don't believe we have time to finish this in C release.