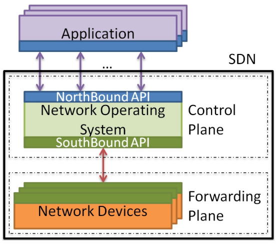
Software Defined Networking (SDN)

Software-Defined Networking or SDN is a relevant new term for an older paradigm of programmable networks. In summary SDN refers to the ability of writing applications to program the behavior of the network and the network devices.



SDN framework

Key elements of the SDN architecture are the separation of control and forwarding planes and open interfaces to allow programmability. On the northbound side, an interface (the northbound API) is provided to applications controlling the network devices; and on the southbound an interface (the southbound API) is provided for controllers to communicate to the network devices. Underneath the hood, the southbound interface is achieved by a definition of a common model and a protocol. The northbound interface is still under discussion.

The SDN architecture is comprised of a centralized controller with full visibility of the underlying network and network devices.

Our research can be classified into the following areas:

1. Control and forwarding plane separation.

One of the SDN requirements is the creation of an abstraction model and a protocol to control the devices. Current protocols and models that satisfy the criteria for SDN and we're focused on are:

a. ForCES - Forwarding and Control Element ForCES Separation.

ForCES [1]is an IETF working group that defines an architecture which includes a protocol, transport and model. Forces was motivated by work that the Network Processing Forum(NPF). NPF was later merged into Optical Internetworking Forum [2] in June 2006. ForCES mainly addresses the open API/protocol that provides a clear separation between control and forwarding plane. The real strength of ForCES lies with its model that enables description of new datapath functionality without changing the protocol between control and datapath.

b. OpenFlow

OpenFlow [3] is a relative new framework developed by Standford and now managed by the ONF - Open Networking Foundation [4] initially to provide a way for researchers to run experimental protocols in the network. OpenFlow provides a protocol with which a controller may manage a static model of an OpenFlow switch.

Both OpenFlow and ForCES have an abstraction model for the forwarding plane albeit a different one. We are investigating the relationship between these two protocols and models, for instance, how they relate to each other and if they need to co-exist. We have begun to develop an OpenFlow testbed [5]for testing as well as an open distributed network element [6] based on ForCES.

Our analysis motivated us to use ForCEs model [7] to describe an Openflow switch thus unifying the two approaches.

2. New Functionality Deployment.

One of the advantages having a forwarding plane abstraction model is the ability to deploy and publish new functionalities hosted by the forwarding devices. We are interested in how ForCES and OpenFlow handles such a static or even dynamic deployment of new functionality, what is the overhead and how fast can they adapt to these changes while publishing them to the applications.

3. SDN Northbound API.

The northbound API for SDN is still under discussion as the applications that take advantage of SDN vary (load-balancing, security, routing etc). One such effort is IETF's I2RS [8] (Interface to Routing System) and another could be Netconf [9]. ForCES through its expressiveness and flexibility of its model can also be such a candidate.

Our research effort focuses initially on a definition for a flexible SDN API framework able to adapt to each application's need.

4. Network Virtualization.

Network Virtualization and SDN are not synonyms terms. Network Virtualization has existed a long time before SDN in various forms such as VLANs and Tunnels. However with the increased virtualization environments in data centers, network virtualization plays and will continue to play a key role in solving various problems of the data centers, such as VM mobility, isolation, security etc. However currently creating, maintaining and updating virtual networks in such volatile environments such as a fully virtualized data center tends to be cumbersome. New network virtualization solutions have been proposed such as VxLAN [10], VGNRE [11], STT [12] and NVO3 [13].

We are interested in how SDN and Network Virtualization interact with each other in order to provide a programmatical interface to setup and maintain virtual networks from an application.

Our initial efforts have led us to a prelimenary definition of a network hypervisor in the form of ForCES elements that can be found here [14].

For our implementation and experiments we are using a ForCES SDK (Software Developement Kit) created by Mojatatu Networks [15]. Inquiries regarding availability of the SDK software can be made at the following email: sdk[at]mojatatu[dot]com.

Additionally we have initated the developement of a ForCES DSL (Domain Specific Language) tool based on the ForCES model to allow developers to define in a more, familiar to programmers, code style ForCES libraries, output their code in the ForCES schema. The output can be used with the ForCES SDK tool and output code.

- ForCES
- OpenFlow
- SDN
- Software Defined Networking
- Virtualization

```
var _gaq = _gaq || [];   _gaq.push(['_setAccount', 'UA-35307132-1']);
```

```
_gaq.push(['_trackPageview']);    (function() {     var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;     ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';     var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);   })();
```

Source URL: http://nam.ece.upatras.gr/index.php?q=node/59

Links:
[1] http://datatracker.ietf.org/wg/forces/charter/
[2] http://www.oiforum.com/
[3] http://www.openflow.com
[4] https://www.opennetworking.org/
[5] http://nam.ece.upatras.gr/index.php?q=node/46
[6] http://nam.ece.upatras.gr/index.php?q=node/43
[7] http://datatracker.ietf.org/doc/draft-haleplidis-forces-openflow-lib/
[8] http://datatracker.ietf.org/wg/i2rs/charter/
[9] http://datatracker.ietf.org/wg/netconf/charter/
[10] http://tools.ietf.org/html/draft-mahalingam-dutt-dcops-vxlan-02
[11] http://tools.ietf.org/html/draft-sridharan-virtualization-nvgre-00
[12] http://tools.ietf.org/html/draft-davie-stt-02
[13] http://datatracker.ietf.org/wg/nvo3/charter/
[14] http://datatracker.ietf.org/doc/draft-haleplidis-forces-virtualization/
[15] http://www.mojatatu.info/