Report

On

Performance Analysis of Various Hybrid SDN Models

In partial fulfilment of the

First Degree Thesis (BITSF421T)

Under the guidance of Hari Babu,Bits Pilani

Submitted by

Kanwardeep 2012C6PS643P

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE,PILANI

Contents

INTRODUCTION

The main purpose of the communication network is the transfer of information from one place to another. Within the network the data moves across various nodes. Efficient and effective forwarding is done by the control plane in the respective nodes.

Traditional Networks

In the traditional networks as given in Fig. 1 a, the control

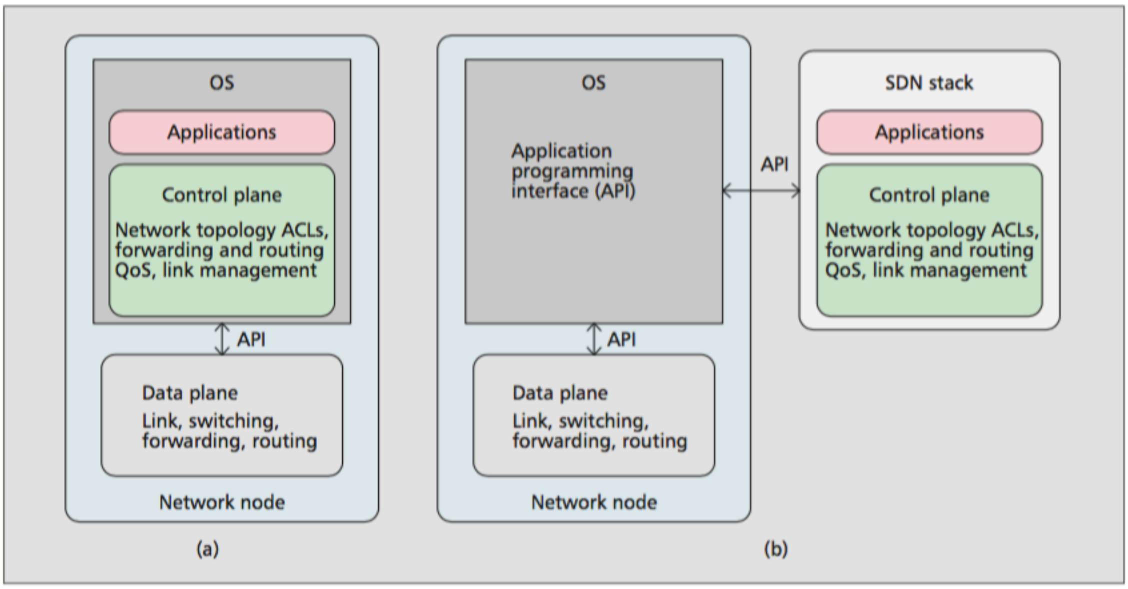

Figure 1. Traditional network view against SDN network view a)Traditional Approach ; b) SDN approach

the control plane and data plane are together in a network node. The control plane is responsible for configuration of the node and the paths to be used for the flow of data. Once these paths have been calculated, they are pushed onto the data plane. Data forwarding at the hardware level is based on this information forwarded by the control plane.

In the traditional approach, once the forwarding policy has been defined, the only way we can make adjustments to the policy is by changing the configuration of the devices. This has restricted the network operators who are keep to scale their networks in response to the changing demands in the traffic, increasing use of mobiles and the impact of big data.

SDN Networks

In SDN the control plane has moved out of the individual network nodes and placed on to a separate, centralized controller. SDN switches are operated by a network operating system (NOS) that gathers information using the API shown in Fig 1 b and changes their forwarding plane, providing an abstract model of the network topology to the SDN controller which is hosting various applications.

The controller can exploit the complete knowledge of the network to fully optimize flow manipulation and provide support for user requirements of flexibility and scalability.

In Fig 3 once the first packet of data of a new flow comes at the switch from the sender (as indicated by step 1), the switch checks for a rule for this packet in the SDN cache (indicated by step 2) . If a matching entry is found, then the steps associated with the specific flow entry are executed (i.e. updating or matching fields or metadata). Then the packets are forwarded to the receiver (as indicated in step 5).

If no match is found in the flow table, the packet may be forwarded to the controller over a channel (indicated as step 3). Using the southbound API (i.e. OpenFlow, PCEP) the controller can then update, add and delete flow entries. The controller executes the routing algorithm and then adds a new forwarding entry to the flow table in the switch and also to each of the relevant switches along the flow path (indicated in step 4). The switch then forwards the packet to the appropriate port to send the packet to the receiver (indicated in step 5).

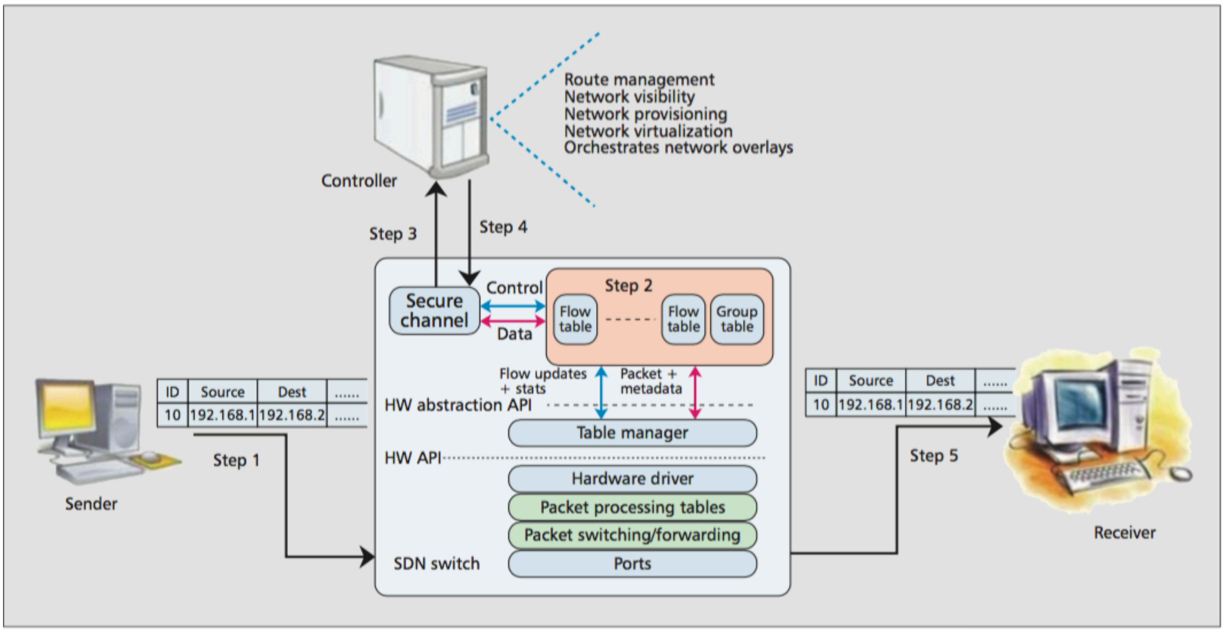

Figure 3 The working of SDN (controller - switch).

Key Challenges

Performance vs Flexibility

One key challenge of SDN is how to handle high security and high performance packet processing flows in an efficient manner. Performance means processing speed of the network node considering both the latency and the throughput. Programmability means the capability to change and accept new rules in order to change the functional behavior. Flexibility is the ability to adapt systems to support new unforeseen features.

Figure 4 outlines the main technologies that can be used for network processing in terms of their relationship between programmability/flexibility and performance.

General purpose processors (CPU,GPP) provide highest flexibility but limited in performance and power dissipation constrained by the general purpose architecture.

Network flow processors are optimized processor for network processing with dedicated hardware accelerators and various technologies are used to reduce power dissipation but this comes at the cost of flexibility as the knowledge of the device is required in this case.

Programmable logic devices (PLDs) or field programmable gate arrays (FPGAs) have high parallel and pipelined data paths that are required for the individual network processing functions.

Last is the Application – specific standard products (ASSPs) are the cornerstone of high performance networks. They are specifically designed for high volume and for a device.

Looking at all this suggests us that the this trade-off can be solved by a hybrid approach.

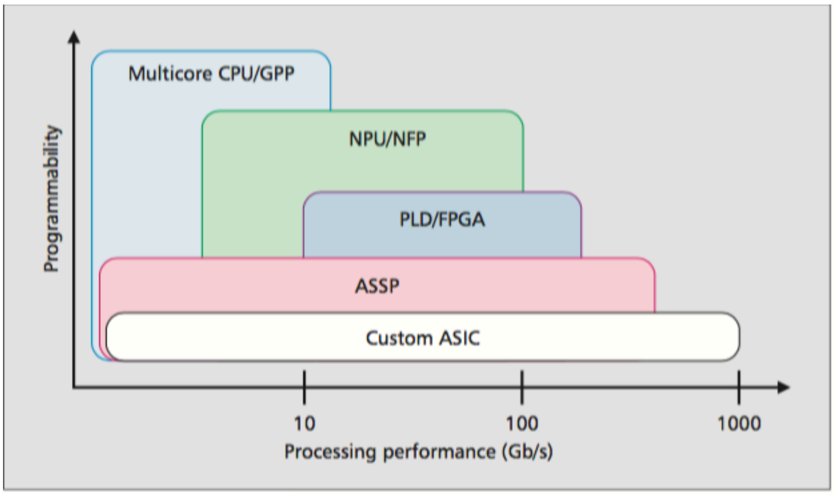

Figure 4 Network processing performance vs programmability

Security

The controllers are an attractive target for attackers in SDN architecture open to unauthorized access and exploitation. Furthermore if the attacker takes control of the controller he can carry out malicious activities as he can control all the nodes which are under that controller.

One potential attack is the denial of service attack (DOS). Within the operation of SDN in Fig 3, there are two options for the handling of a new flow when no flow match exists in the flow table. If only the header information is transmitted to the controller, the packet itself must be stored in the node memory in which case it would be easy for the attacker to execute DOS attack. If whole of the packet is sent to the controller then it will absorb high bandwidth.

Interoperability

It is not possible to simply swap out the whole of the legacy system and introduce the SDN system as the cost involved with it would be extremely high and also the SDN model would not be as secure and reliable as the legacy system is currently present and also the protocols for SDN model have not been formulated. So the transition to SDN requires simultaneous support of SDN and legacy equipment. The IETF path computation equipment (PCE) could help with partial migration to SDN. With this there is a need for a protocol for communication between legacy and the SDN network elements.

Further development is required to achieve a hybrid SDN infrastructure in which traditional and SDN infrastructure are both working together. Such a solution will reduce the cost, risk and disruption for the enterprise and carrier networks transitioning to SDN.

Further work would require consideration of standardization of protocols mechanisms and interfaces.

Hybrid SDN Models

Topology – Based Hybrid SDN

In the topology based hybrid SDN (lets call it TB hSDN) model relies on a topological separation of the nodes under each paradigm. The network is partitioned in zones, so that each node belongs to only one zone. A zone is a set of interconnected nodes controlled by the same paradigm. Example of TB hSDN is depicted in Fig 5

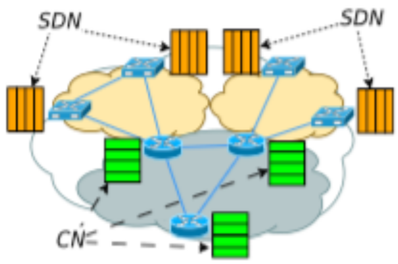

Figure 5 topology based

The figure represents a network divided in 3 zones i.e. two SDN zones (orange color) and 1 legacy zone (grey color).
During transition TB hSDN fits a transition strategy in which SDN is introduced region wise i.e. first introducing in some countries and then progressively increased along with

- Maturity of the technology
- SDN node deployment
- Acquisition of expertise from operators

In the long run as many networks especially in enterprises, are already divided in many routing domains. Reasons vary from business and organizational to technical ones. By making sure SDN and legacy infrastructure in different zones we generalize this design.

Service Based Hybrid SDN

In this the idea is that service based hybrid SDN (SB hSDN) model and the legacy model provide different services. For example network wide forwarding, the two paradigms can span an overlapping set of nodes, controlling a different part of the Forwarding Information Base (FIB) of each node. Some nodes can be controlled by a single paradigm for example to realize load balancing etc.

As shown in Fig 6. In this example SDN fills most of FIB entries of nodes at the border of the network (indicated by Orange color), while legacy system (indicated by green color) has a has an exclusive control of the FIB of internal nodes.

Figure 6 service-based

Here we are indicating cases in which forwarding is delegated to legacy system and traffic engineering and access policies are guided by SDN.

The idea here is that in transition from the legacy system the SDN system takes on some services in the starting and as SDN technology improves we keep on adding more services and nodes controlled by SDN. In the long run we can hope that SDN can support new network services like Network Function

Virtualization.

Class Based Hybrid SDN

In class based hybrid SDN (CB hSDN) model we partition the traffic in classes and delegate them to legacy system and SDN system. In this SDN and legacy span all the nodes controlling a disjoin set of FIB entries. As shown in Fig 7.

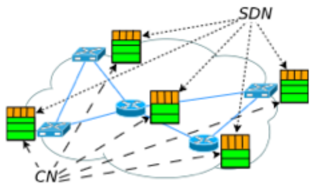

Figure 7 Class based

In figure 7 each node SDN fills the FIB entries to control a small portion of traffic whereas legacy system fills FIB entries to control huge portion of traffic. For example SDN controlled classes can be defined for TCP flows or some applications.

In this idea we delegate more and more traffic to the SDN system over time. This will help us to test SDN on non critical data and help to figure out flaws if there are any and also help us to figure out the solutions to the problem of scalability. In the long run this could a good solution as a few flows typically attract most of the traffic classes those flows can be grouped in SDN controller to flexibly control load balancing and change forwarding paths without the need to tweaking legacy system. The classes can be assigned by specifying values of packet fields (like all traffic towards TCP port 80) .

Integrated Hybrid SDN

This is different from all the hybrid models described above in that in integrated hybrid SDN (integrated hSDN) is responsible for all the network services and uses legacy system protocols as an interface to node FIBs as shown in Figure 8.

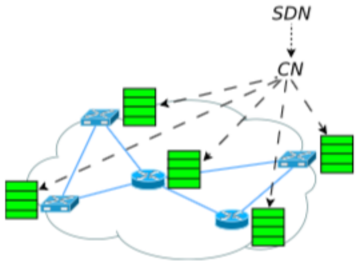

Figure 8 Integrated

In the starting only the control plane is moved from legacy system to the SDN hence reducing the costs and the disruption risks involved with the SDN model. So operators can acquire confidence in SDN while relying on traditional well known protocols. After that is done we can start adding SDN nodes and updating SDN controller in parallel to proper SDN deployment.

Tradeoff Analysis

Dimensions used for comparing different models

- Expressiveness and management simplicity
- Robustness and scalability
- Deployment costs
- Flexibility and paradigm complexity

The tradeoff comparison is summarized in Table 1

	CN	TB hSDN	SB hSDN	CB hSDN	Integrated	SDN
non IP-based forwarding	hard, complex configuration	programmable in SDN zones	programmable for SDN services	programmable for SDN traffic	very hard (e.g., BGP FlowSpec)	programmable
traffic steering, middleboxing	hard (e.g., box replication)	programmable in SDN zones	programmable for SDN services	programmable for SDN traffic	programmable by the controller	programmable
scalability and robustness	by CN protocols	by CN protocols in CN zones	by CN protocols for CN services	by CN protocols for CN traffic	possibly, by CN protocols	SDN controller concern
required custom software	none	controllers of SDN zones	controllers for SDN services	controllers for SDN flows	SDN controller	SDN controller
upgrade costs (hw, sw, expert)	none	partial, progressive renovation	partial, progressive renovation	partial, progressive renovation	none	global renovation
paradigm interaction	none	control-plane collaboration	data-plane visibility	control-plane coordination	control-plane integration	none

Table 1

Hybridization drawbacks

The drawbacks are rather low in CB hSDN in which the interaction is restricted to coordination for specific operations and for SB hSDN where each paradigm only needs to have a vision on the FIB entries configured by the other. In TB hSDN and integrated hSDN quite strong interaction is needed i.e. control plane collaboration, which may not be an easy task.
These drawbacks can be overcome with the hybridization of the various hybrid models. For example the there can be a TB hSDN network that is split into a legacy zone and a SB hSDN zone and a pure SDN zone.

Mininet

Mininet is a network emulator. It runs a collection of end-hosts, switches, routers and links on a single Linux kernel. It uses virtualization to make a single machine seem like a complete network running the same kernel and user code. A mininet host behaves like a real machine. We can ssh into it and run arbitrary programs. Packets are processed by real seeming switch and router. We can also include delays and decide the sped of links.

Miniedit

Miniedit is a graphical user interface of mininet. In this we can design topology by dragging and dropping routers, switches, hosts, SDN switches and controllers. We can also set almost all the setting of the topology

Experiment

The experiment involved 2 routers at the highest level followed by 3 switches of each router which in turn have 2 switches each and which in turn has 3 hosts. The network is set in Miniedit on a machine running Ubuntu 16.04 LTS.
In one experiment both the topologies under each of the routers are using legacy networking and under the next experiment it one tree under one router is using legacy networking and the other is using SDN as indicated in topology based hybrid SDN
We set the link bandwidth as 10Mbps and link delay as 2 ms and packet loss as 0. Then we use the ping command to see the response time. In case of Hybrid SDN it is averaged to about 70ms and for legacy it is close to 100ms. This shows the advantage of Hybrid SDN as compared to legacy network.

The utilization of links is also more in SDN as compared to legacy networks as more links are used optimally in Hybrid SDN. But in smaller networks the legacy networks perform better than Hybrid SDN. This shows that the time taken to query of routing table have a great impact on the forwarding speed of the node.

References

1. Are we ready for SDN? Implementation Challenges for SDN by Sakir, Sandra Scott and pushpinder kaur, CSIT Queen’s University Belfast, Barbara Fraser and david lake, Cisco Systems Jim Finnegan and Niel Viljoen, Netronome
2. Repeaing the benefits of incremental SDN Deployment in Enterprize networks by Dan Levin, Marco Canini and Anja Feldmann
3. Opportunities and research challenges of Hybrid SDN by Stefano, Oliver and Laurent
4. Open networking foundation
5. The future of networking and the past of the protocols
6. Introduction to SDN by Blue elephant
7. How networking will shape Networking by Nick Mckeown