# 

# An incubated co-working space for technology innovation

## Dasein

**Software Defined Networking**

**Table of Contents**

[1](#_Toc419738684)

[An incubated co-working space for technology innovation 1](#_Toc419738685)

[Dasein 1](#_Toc419738686)

[Document Change Log ii](#_Toc419738687)

[Definition of terms iii](#_Toc419738688)

[Team Register iii](#_Toc419738689)

[Description of Acronyms iv](#_Toc419738690)

[1. Proposal Overview 6](#_Toc419738691)

[1.1 Background 6](#_Toc419738692)

[2. Scope 6](#_Toc419738693)

[3. Problem Statement 6](#_Toc419738694)

4. Solution…………………………………………………………………………………………………………………………7

5. Architecture…………………………………………………………………………………………………………………8

6. Hardware………………………………………………………………………………………………………………………9

7. Benefits………………………………………………………………………………………………………………………….10

8. Limitations……………………………………………………………………………………………………………………..11

9. Significance……………………………………………………………………………………………………………………12

# Document Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 0.01 | 21 March 2015 | Malusi Gcakasi | Created initial template document. |
| 0.02 | 18 May 2015 | Dasein | Upgrading the information in the document. |

# Definition of terms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **SDN** | An approach to computer networking that allows network administrators to manage network services through abstraction of lower-level functionality. |
| **QOS** | It is the overall performance of a telephony or computer network, particularly the performance seen by the users of the network. |
| **OS** | It is a software that manages computer hardware and software resources and provides common services for computer programs. |
| **CLI** | It is a means of interacting with a computer program where the [user](http://en.wikipedia.org/wiki/User_(computing)) (or [client](http://en.wikipedia.org/wiki/Client_(computing))) issues [commands](http://en.wikipedia.org/wiki/Command_(computing)) to the program in the form of successive lines of text (command lines). |
| **VLAN** | A **VLAN** is a group of end stations with a common set of requirements, independent of physical location |
| **IP** | An **Internet Protocol address** (**IP address**) is a numerical label assigned to each device (e.g., computer, printer) participating in a [computer network](http://en.wikipedia.org/wiki/Computer_network) that uses the [Internet Protocol](http://en.wikipedia.org/wiki/Internet_Protocol) for communication |
| **IO** | In [computing](http://en.wikipedia.org/wiki/Computing), **input/output** or **I/O** (or informally, **io** or **IO**) is the communication between an [information processing system](http://en.wikipedia.org/wiki/Information_processing_system) (such as a [computer](http://en.wikipedia.org/wiki/Computer)) and the outside world, possibly a human or another information processing system |
| **CPU** | A **central processing unit** (**CPU**) is the [electronic circuitry](http://en.wikipedia.org/wiki/Electronic_circuit) within a [computer](http://en.wikipedia.org/wiki/Computer) that carries out the [instructions](http://en.wikipedia.org/wiki/Instruction_(computing)) of a [computer program](http://en.wikipedia.org/wiki/Computer_program) by performing the basic arithmetic, logical, control and [input/output](http://en.wikipedia.org/wiki/Input/output) (I/O) operations specified by the instructions |
| **SoC** | A **system** on a **chip** or **system on chip** (**SoC** or **SOC**) is an integrated circuit (IC) that integrates all components of a computer or other electronic **system** into a single **chip**. |

# Team Register

|  |  |
| --- | --- |
| **Member Name** | **Role Description** |
| Zimkhitha Matshangaza |  |
| Zimasa Bulo |  |
| Lerato Selemela |  |

# 

# Description of Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| SDN | Software Defined Networking |
| QOS | Quality of service |
| OS | Operating System |
| CLI | Command Line Interface |
| VLAN | Virtual Local Area Network |
| LAN | Local Area Network |
| IP | Internet Protocol |
| IO | Input/output |
| CPU | Central Processing Unit |
| SoC | System on a Chip |

# Proposal Overview

# Background

Traditional networking is an infrastructure that many companies has been using to increase productivity in their work environment. This infrastructure has been helpful it allowed companies to enhance connectivity, share hardware and data management.

However the technology is advancing the physical hardware use is being minimized and everything is being controlled by software. Another thing the use of data is increasing which gives the traditional infrastructure performance challenge.

This document will show how Software defined Networking is the future in networking.

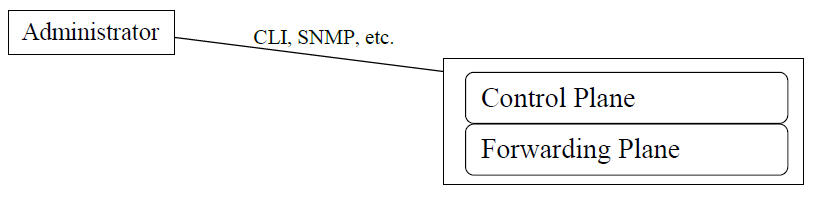
# Scope

This document aims to be an initial overview description of the project. It aims to explain the business sense for the new network infrastructure and the problem it solves.

# Problem Statement

The current situation in networking is if you want to affect your network’s knowledge or operating status, you have to configure each device in your network. Adding in new hardware, new programs, and new QOS or security policies can sometimes require an administrator configuring numerous individual devices. Those devices could be from multiple vendors, which will require the administrator to have working knowledge of the operating system (OS) and configuration methods for each device. As your network scales, it becomes more and more inflexible as more and more devices have to be configured for every change. Eventually it can stagnate and you will need a new solution, or your administrators won’t want to change anything because it would simply be such a huge undertaking.

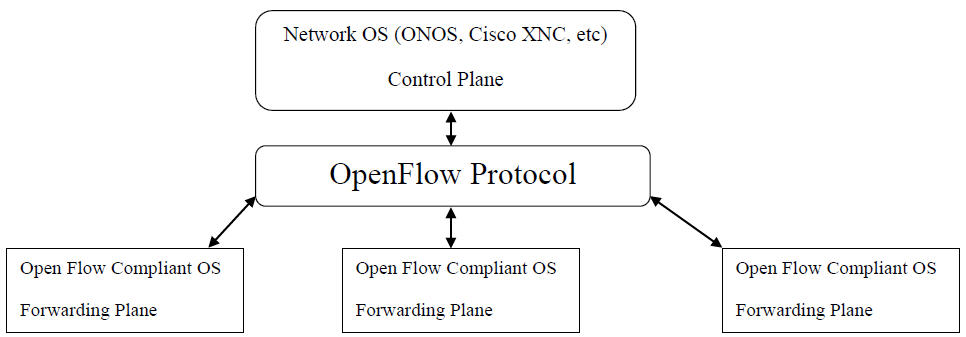
Another aspect of our current model is that the control and forwarding planes of devices are usually in the same box. The control plane is the part of the device that makes decisions about how it will handle traffic, the forwarding plane is the part that actually processes and forwards traffic based on how the control plane is configured.



An administrator has to access into the device, using CLI or SNMP or other access methods, and configure the control plane in order to influence new behaviour in the forwarding plane. Whether that means modifying an ACL, turning on a routing protocol, adding new HSRP neighbours, really anything, it must be configured on the device. So once your network scales as your business grows, every change and addition will require more and more configuration tasks on existing network devices, in order to guarantee uniform performance. And with more requirements for flexible networking for server provisioning, as well as cloud usage, this can quickly get out of hand.

1. **Solution**

The big strategy behind SDN is that it decouples the control plane from the forwarding plane. No longer will the decisions for how to handle traffic be made on the same device that is actually forwarding it. One of the most popular tools for this is the OpenFlow protocol.



It will periodically gather information from network devices concerning their status as well as issue commands concerning how to handle traffic. The information it gathers will be passed in an abstract format to a network controller OS, examples of which are ONOS or Cisco’s XNC. On the controller an administrator can programmatically define how programs and applications traffic should be handled. This control information is then pushed out to the network devices through OpenFlow. A great way to think of this is like having a large number of static routes on all your devices, defining traffic forwarding. But instead of the inflexibility of traditional static routes, these SDN routes can auto-adjust and reshape network flow based on what information OpenFlow is receiving from the devices. This also gives administrators a centralized control over their networks. No longer do you have to configure multiple devices when you use a new application. Now you will be able to program how that application should be handled by the network only once, and no matter where it goes in your network, it will be treated uniformly.

There are two major reasons that makes SDN a best solution. The first is cloud usage. With increased use of cloud computing and storage, there is a concern that your programs and traffic will not be handled the same way “out there” as it is “in here,” or in your local network. What SDN does is view cloud based services as just more network devices. So when you program how an application should be handled, that control will follow the program out to the cloud, or back to your local network, or wherever your devices exist.

The other major reason is user programmability. OpenFlow currently works with Python and other programming languages, so an individual administrator can define their own tools and protocols for how their network behaves and responds to changes. Sure there will be some standardization, but if there is that One Thing that you wish your network could do, you can now program your network to do it, without waiting for a manufacturers OS update to bring you that functionality.

SDN is the answer to some major speed bumps in the world of networking. ACLs, routing protocols, QOS….they all do their job, but each one must be configured individually and on every device. With SDN you get complete control of your network through a single programmable interface, regardless of whether your network is physically present or in the cloud.

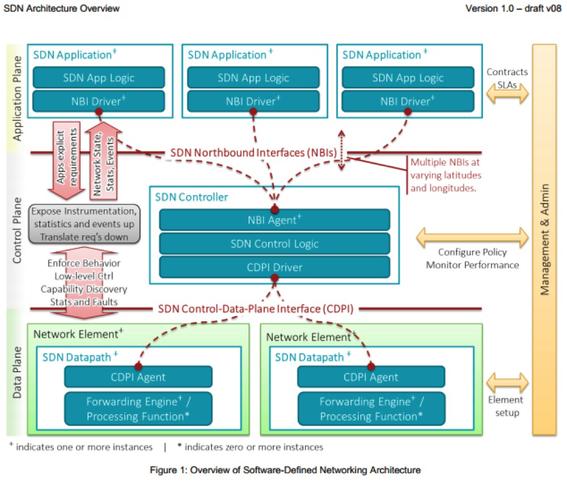
1. **Architecture**

The SDN operates in three layers

* **Application (application plane)**: Network services, utilities and applications which interface with the control level to specify needs and requirements. These may be what is known as “network aware.”

The network hardware you’re probably familiar with performs much or all these functions; the goal of SDN is to offload the handling of traffic and the way it meets the needs of the applications involved. For instance, the control layer might reside on a server and the application layer as a software-based application programming interface (API). This means the hardware which handles the network traffic doesn’t need to direct it or deal with management, making the environment more flexible and adaptable

* **Control (control plane)**: The intelligence in devices which works in true “middle-man” fashion, determining how traffic should flow based on the status of the infrastructure layer and the requirements specified by the application layer. The model here is centralized (either physical or logical) rather than distributed across various and possibly disparate devices
* **Infrastructure (data plane)**: Network switches and routers and the data itself as well as the process of forwarding data to the appropriate destination.



1. **Hardware**

* Switch
* Operating system(OS) Linux
* Beacon controller
* Raspberry Pie

1. **Benefits**

**Centralized network provisioning.**

Software defined networks provide a centralized view of the entire network, making it easier to centralize enterprise management and provisioning. For example, more VLANs are becoming part of physical LANs, creating a Gordian knot of links and dependencies. By abstracting the control and data planes, SDN can accelerate service delivery and provide more agility in provisioning both virtual and physical network devices from a central location.

**2. Holistic enterprise management.**

Enterprise networks have to set up new applications and virtual machines on demand to accommodate new processing requests such as those for big data. SDN allows IT managers to experiment with network configuration without impacting the network. SDN also supports management of both physical and virtual switches and network devices from a central controller; something you can’t do with SNMP. SDN provides a single set of APIs to create a single management console for physical and virtual devices.

**3. More granular security.**

One of the advantages of security defined networking that appeals most to IT managers is centralized [security](http://www.im-techsolutions.com/security?__hstc=62151795.3959f0dc380e8e0be211e54361daf52d.1431945354353.1431945354353.1431945354353.1&__hssc=62151795.1.1431945354354&__hsfp=101159394).  Virtualization has made network management more challenging. With virtual machines coming and going as part of physical systems, it’s more difficult to consistently apply firewall and content filtering polices. When you add in complexities such as securing BYOD devices, the security problem is compounded.

The SDN Controller provides a central point of control to distribute security and policy information consistently throughout the enterprise. Centralizing security control into one entity, like the SDN Controller, has the disadvantage of creating a central point of attack, but SDN can effectively be used to manage security throughout the enterprise if it is implemented securely and properly.

**4**. **Lower operating costs.**

Administrative efficiency, improvements in server utilization, better control of virtualization, and other benefits should result in operational savings. Although it is still early to show real proof of savings, SDN should lower overall operating costs and result in administrative savings since many of the routine network administration issues can be centralized and automated.

**5. Hardware savings and reduced capital expenditures.**

Adopting SDN also gives new life to existing network devices. SDN makes it easier to optimize commoditized hardware. Existing hardware can be repurposed using instructions from the SDN controller and less expensive hardware can be deployed to greater effect since new devices essentially become “white box” switches with all the intelligence centered at the SDN controller.

**6. Cloud abstraction.**

Cloud computing is here to stay and it is evolving into a unified infrastructure. By abstracting cloud resources using software defined networking, it’s easier to unify cloud resources. The networking components that make up [massive data center](http://www.im-techsolutions.com/green-grid-ebay-data-center-efficiency-case-study?__hstc=62151795.3959f0dc380e8e0be211e54361daf52d.1431945354353.1431945354353.1431945354353.1&__hssc=62151795.1.1431945354354&__hsfp=101159394) platforms can all be managed from the SDN controller.

**7. Guaranteed content delivery.**

The ability to shape and control data traffic is one of the primary advantages of software defined networking. Being able to direct and automate data traffic makes it easier to implement quality of services (QoS) for voice over IP and multimedia transmissions. Streaming high quality video is easier because SDN improves network responsiveness to ensure a flawless user experience.

1. **Limitations**

SDN limitations

Software-defined networking is touted as a panacea for data centres, but the combination of faster networking speeds and virtualization restricts what it can do.

Software-defined technology will solve all data center problems and allow your computing gear to be an undifferentiated mass of identical nodes.

**But will a cluster of identical machines handling networks, servers and storage solve everything? Or do we need a set of task-specific systems?**

We think the latter, with the move to software-defined technologies more evolutionary. For example, we see network switching chips sticking around for the foreseeable future. Here's why.

Moving the smarts of switching to virtual machines sounds like an answer to many issues, such as adding compute power during peak loads to keep latencies down, which would help the spread of in-transit encryption, firewalling and filtering and access control systems.

1. **Significance**

**Flexibility**. The centralized “controller” model offers a single point of control for the network and can support rapid changes in the network. This is especially important in large Cloud Data Centers which host thousands of users and where the workloads are highly dynamic. Software control over the network can be used to tailor the network to the usage / workload pattern.

**Performance and Network Optimization**. The operation of traditional network switches is governed by distributed network protocols, such as the IEEE 802.11 protocols for Ethernet switches. Although the spanning tree protocols used in Ethernet switches are robust and simple, they are most efficient for communication between the leaves to the root of the tree but not efficient for communication between leaves. Patterns of communication in Data Centers is now weighted towards “leaf-to-leaf”, which means between servers, or “East-West” traffic. The SDN model is well suited for rapid introduction of new control protocols since these would be deployed as software modules on the SDN controller.

**New network applications**. The SDN model supports the creation of virtual networks in which each network user can be provided full control over their virtual network. The SDN controller also supports APIs that will foster the creation of new network applications.