



Universiteit Leiden

ICT in Business

Investigating network automation at Capgemini

Name: Huihang He
Student-no: s1372866

Date: 21/08/2015

1st supervisor: Luuk Groenewegen

2nd supervisor: Aske Plaat

External Supervisor: ir. Wim van der Bijl

External 2nd Supervisor: Marco van der Pal

MASTER' S THESIS

Investigating network automation at Capgemini

Huihang He

hehuihang@yahoo.com

S1372866

In partial fulfillment of the requirements for the degree of

Master of Science (M.SC.) of ICT in Business

Graduation: August 2015

1st supervisor: Luuk Groenewegen, Leiden University

2nd supervisor: Aske Plaat, Leiden University

External supervisor: ir. Wim van der Bijl, Capgemini

External 2nd supervisor: Marco van der Pal, Capgemini

Leiden Institute of Advanced Computer
Science (LIACS)

Leiden University

Leiden, The Netherlands

<http://liacs.leidenuniv.nl/en/>

Infrastructure Outsourcing

Infrastructure Services

Capgemini Nederland

Utrecht, The Netherlands

<https://www.nl.capgemini.com/>

Table of Contents

List of figures	v
List of Tables.....	v
Abstract.....	vi
Chapter 1 - Introduction	1
1.1 Problem Statement	1
1.2 Statement of research.....	4
1.3 Thesis outline	6
Chapter 2. Analysis of current processes using UML.....	7
2.1 The reasons to choose subset of UML analysis.....	7
2.2 Use case analysis overview	8
2.3 Business view of current processes and the chosen scope.....	10
2.4 Actors description	14
2.5 Technical view in scope of current provisioning procedures	15
2.6 Use case diagram	17
Chapter 3. Analyzing “to be” design in NetMRI.....	23
3.1 Reasons to choose flowchart	24
3.2 From “as is” process to “to be” design using flowchart	25
3.2.1 “to be” process overview in NetMRI	25
3.2.2 Initialization the “to be” process in NetMRI	27
Chapter 4. Empirical study	28
4.1 Define main goal and subgoals for the test lab.....	29
4.2 Issues between “to be” design and implementation	30
4.3 Set-up of the test lab.....	40
4.4 Test lab realization	43
4.5 Conclusion on the test lab results and future improvements.....	50
Chapter 5. Software-Defined Network (SDN) roadmap	51
5.1 SDN Brief review	51
5.2 What is the roadmap	53
5.3 Creation of the roadmap.....	54
5.3.1 Top layer	55
5.3.2 Middle layer and Bottom layer	56
5.4 Conclusion for roadmap.....	56
Chapter 6. Discussion and Conclusion	57
Bibliography	60
Appendix	62
1. OSI Layer 2 and Layer 3 Functional Summary	62
2. LAN vs. WAN	63
3. VLAN and Trunking.....	64
4. CLI.....	65

5.	PKI	66
6.	Ping	68
7.	Background knowledge for DMVPN.....	68
8.	SNMP	72
9.	DHCP	72
10.	VRF	72
11.	CIDR.....	72
12.	Telnet	72
13.	Putty.....	72
14.	Preparation for the test lab	73
15.	SDDC.....	73
16.	Control plane	74
17.	Data Plane (Forwarding Plane)	74

List of figures

Figure 1 Solution design logical view (Capgemini, 2014)	9
Figure 2 Overview use case diagram for DMVPN.....	9
Figure 3 Working process of standard DMVPN service (Capgemini, 2014)	10
Figure 4 Sequences for building the tunnels.....	12
Figure 5 Use Case Diagram.....	20
Figure 6 Changes in use case between "as is" and "to be"	24
Figure 7 "to be" process overview flowchart in NetMRI.....	26
Figure 8 Initialization of "to be" process in NetMRI.....	27
Figure 9 Define "custom issue"	39
Figure 10 SNMP community string list in NetMRI.....	42
Figure 11 SNMP enabled in stepping stone (pre-install) hub router	43
Figure 12 SNMP community string in stepping stone (pre-install) hub router	43
Figure 13 Credential status to access a device.....	45
Figure 14 CLI credential list in NetMRI	45
Figure 15 NetMRI accessible for stepping stone (pre-install) hub router	46
Figure 16 Detailed flowchart for subgoal 3	47
Figure 17 Perl script in NetMRI	48
Figure 18 Script Result.....	49
Figure 19 CLI results after script executed	49
Figure 20 T-plan fast approach template	54
Figure 21 Using OSI Layers for Referencing Other Protocols (CCNA Intro Exam Certification Guide)..	62
Figure 22 Example LAN, Two Buildings (CCNA Intro Exam Certification Guide)	64
Figure 23 Point-to-point WAN (CCNA Intro Exam Certification Guide)	64
Figure 24 VLAN Trunking Between Two Switches (CCNA Intro Exam Certification Guide)	65
Figure 25 CLI Access (CCNA Intro Exam Certification Guide)	66
Figure 26 DMVPN Two Deployment Design(Cisco Services and Technologies).....	69
Figure 27 Spoke registers public IP address	70

List of Tables

Table 1 Main reasons for Capgemini to use software NetMRI	3
Table 2 Actors for use case analysis	15
Table 3 Main goal and sub-goals for the test lab	30
Table 4 Main goal and subgoals	43
Table 5 Result of the subgoals.....	50
Table 6 Purpose of adopt SDN	56
Table 7 OSI Layer2 and Layer3 Functional Summary	62
Table 8 Comparison between Routers and Switches.....	Error! Bookmark not defined.

Abstract

As the current network provisioning process in Capgemini can't meet the clients' growing requirements, a desirable network automation solution is needed. Besides, new technology software-defined network (SDN) is being developed to make current network architecture dynamic and more flexible. Thus, testing whether the chosen network automation software NetMRI is a desirable working tool, and the relationship between the chosen software and software-defined network, are analyzed in Capgemini Netherlands branch.

A specific network service, Dynamic Multipoint Virtual Private Network (DMVPN), is analyzed and a test lab is built in the chosen software. To figure out how this network automation product NetMRI fits into the new technology software-defined network (SDN), a high level implementation roadmap for software-defined network is created.

The result showed that the network automation product NetMRI can send provisioning command automatically, but there are lots of changes compared with the original provisioning procedures. Due to the limitation of time, a high level software-defined roadmap is created. NetMRI could be regarded as a preparation for adopting new technologies, like SDN, but there is no concrete answer.

It can be concluded that the management of Capgemini needs to decide whether to continue further investigating in NetMRI. More research should be conducted regarding the initiatives of adopting software-defined network and what concrete functions software-defined network can provide, as well as what are the current available resources in Capgemini and what resources for adopting the software-defined network, are necessary in order to meet the customers' growing requirements.

This paper contributes to a process of analyzing current "as is" provisioning procedures, of designing "to be" procedures, and of implementing the design in a test lab environment. Besides, it provides an overview for further research about adopting software-defined network.

1 Chapter 1 - Introduction

In this master thesis, I apply to practice what I have learned from the ICT in Business program aiming to automate network provisioning by linking the gap between management and technology. Conducted inside the Dutch branch of Capgemini, this thesis research has two main practical goals. One goal is to introduce network automation using traditional product NetMRI, the other goal is to clarify whether using the traditional network automation product NetMRI is a good step on the road to software-defined network (SDN) for Capgemini. NetMRI being a traditional network automation product means it has not been designed according to software-defined network (SDN) principles and protocols. Therefore, our first goal is to automate network configuration and provisioning for the concrete and already existing service Dynamic Multipoint Virtual Private Network (DMVPN) and to build a test lab on the product NetMRI to test whether it is a suitable working tool. Our second goal is to create a roadmap for software-defined network (SDN) and figure out to what extent network automation using NetMRI can fit into the road to software-defined network (SDN).

This introduction chapter has three parts: problem statement, statement of research, and thesis outline. In problem statement part, it will first discuss why NetMRI as network automation tool is a good choice and the needs to create a roadmap for software-defined network (SDN) following the statement of research. The last part is a brief summary of all the chapters in this thesis.

1.1 Problem Statement

The problem statement addresses, in general terms, the reasons to use product NetMRI for network automation and the reason to explore the relationship between NetMRI and software-defined network.

There are three main reasons to use software NetMRI for automating the DMVPN provisioning, including current problems in provisioning, possibility for automation, and currently available resources inside Capgemini. Dynamic Multipoint Virtual Private Network (DMVPN) is one kind of Virtual Private Network, which refers to the secure communication between a set of sites and a closed user group (Zhensheng Zhang, 2004).

The first main reason to use software NetMRI for network automation is because of current problems in provisioning. Network operations still involve lots of manual processes, including provisioning, configuration, ongoing maintenance (Jones, 2006). The detailed reasoning is as follows. Firstly, clients or customers want to shorten the time used for building the connection between the branch office and Capgemini data center. Secondly, although the Dynamic Multipoint Virtual Private

Network (DMVPN) connection will run without human intervention after the network connection established, there are a lot of manual configurations must be carried out using the current procedures. The network administrator has to configure the initial Virtual Private Network connections. That is to say, when there is a client request, engineers have to repeat a similar DMVPN provisioning, which is one kind of VPN, especially for the standardized service, which has fixed parameters. Thirdly, as all the provisioning steps need to be finished manually, lots of mistakes occur during manual configuration. Last but not the least, there are problems due to collaborative work between Capgemini Netherlands and Capgemini India. Capgemini Netherlands is responsible for checking the provisioning and Capgemini India is responsible for realizing the detailed provisioning procedures. This situation makes it difficult to solve problems, as communicating is not so easy and time consuming.

The second main reason to use software NetMRI is the particular characters of this type of automation. First of all, the DMVPN provisioning procedures are concrete and very detailed. Network automation aims to automate firstly the standard network provisioning, which is easy to apply with fixed parameters. When a client requests the standard Dynamic Multipoint Virtual Private Network (DMVPN) services, there are predefined options and parameters. For example, the parameters could be a network component, a firewall, a source, a destination, or an application. The procedure is detailed and engineers have to repeat similar provisioning all the time. Secondly, using software to automate procedures could reduce mistakes as it is predefined and predictable what should be done in each step. Thirdly, Capgemini is using another product Infoblox DDI, which is from the same company(Infoblox) as NetMRI. There are possibilities to optimize part of the process using the connection between the two software products.

The last main reason to use software NetMRI for automation is: Capgemini bought the software NetMRI a year ago, but due to the complexity of analyzing and lack of employees and time, Capgemini didn't use it yet. So trying to start using it now is most urgent. Besides, there is a master student available for graduate internship.

The table below gives a clear summary of three main reasons for Capgemini to use software NetMRI for automation.

Current problems in provisioning	<ul style="list-style-type: none"> • Client demands to shorten the provisioning time • Engineers have to repeat a DMVPN provisioning when there is a request • Lots of mistakes occur during manually provisioning • Problems arise from collaborative work between Capgemini Netherlands and Capgemini India
Characters of this type of automation	<ul style="list-style-type: none"> • DMVPN provisioning procedures are detailed enough • The result of using software to automate provisioning is predictable • Possibilities to optimize and integrate processes using the two products from same company (Infoblox)
Current available resources	<ul style="list-style-type: none"> • Capgemini Netherlands bought the product NetMRI a year ago • Lack of employees and time • Internship Master student available

Table 1 Main reasons for Capgemini to use software NetMRI

SDN The other goal is to clarify whether using the traditional network automation product NetMRI is a good step on the road to software-defined network (SDN) for Capgemini. The definition below of SDN is quoted from the white paper of the Open Networking Foundation:

“Software-defined network is the physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices. This architecture decouples the network control and forwarding functions, thus enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services” (Open Networking Foundation, 2015).

The Open Networking Foundation (ONF), which is a nonprofit organization dedicated to the open standards development of Software-Defined Networking with currently 138 members, was founded in 2011 by several companies including: Google, Yahoo!, Deutsche Telekom, Verizon, Facebook, and Microsoft.

The software-defined network roadmap is created at the data center of Capgemini Netherlands branch. The reasons to explore the relationship between NetMRI and software-defined network are as follows. Firstly, the current initiatives for using NetMRI and software-defined network are the same: clients of the Capgemini data center require both stable network service and adapting network service requests quickly, especially to shorten the provisioning time. As mentioned early, NetMRI is worth to try and Capgemini’s generic service team is trying to automate network

provisioning by using NetMRI. The Capgemini Netherlands branch is seeking the desirable technology to meet clients' changing requirements. Secondly, there is a thesis research in Capgemini about software-defined network value creation, which arises the awareness to have more research on the software-defined network. Besides, software-defined network is a hot topic nowadays, for example, Gartner mentions the possibility to change the current static network architecture in a dynamic and more flexible architecture (Gartner, 2015). These are the reasons why the management of Capgemini wants to know whether using the traditional network automation product NetMRI is a good step on the road to software-defined network (SDN) for Capgemini.

SDN roadmap A roadmap is most helpful to answer the second question: whether using the traditional network automation product NetMRI is a good step on the road to SDN. In Capgemini Netherlands infrastructure generic service team, a roadmap for future planning is commonly used. A roadmap can help make better technology investment decisions (Olin H. Bray, 1997). Therefore, the management of Capgemini wants to have a roadmap for software-defined network. To be specific, the reason to create a software-defined network roadmap is to figure out to what extent network automation using NetMRI can fit into the road to software-defined network.

1.2 Statement of research

The research approach of this thesis is the integration of different components in an unusual way. These components are from two sides. One side is the network technologies, the other side is business analysis. Nowadays, there are needs for network automation. Besides, new network technologies, like software-defined network, aim to enable programming in network. From the business analysis side, there are books about using UML analysis for business analysis. Moreover, Capgemini has its own business analysis method using UML. This thesis combines these components together by using the expertise knowledge in both fields of business analysis and networking technologies. To be specific, this chapter will briefly introduce the method for researching including literature review, the reason and purpose to choose a subset of UML for analyzing, and purpose of building Test Lab.

Literature review The book "Research Methods for Business Students" mentions the importance to have a critical literature review and different choices for structure of a critical literature review. In this thesis, we use literature review throughout the thesis (Saunders, 2009), because this thesis includes two main goals. The first part is to automate a small set of current Dynamic Multipoint Virtual Private Network (DMVPN) provisioning procedures with traditional network automation software NetMRI. The first goal is to figure out whether this software is a suitable working tool for network automation. The second goal is to create a SDN roadmap and figure out whether using this

software is a good step on the road to SDN for Capgemini. Although we regard the software-defined network roadmap as a future implication of using NetMRI, these two goals are focusing on different topics. Therefore, it's a better choice using literature review throughout the thesis. As this master graduation thesis is conducted at Capgemini Netherlands branch, this thesis tries to solve the practical problem in a Capgemini way. For instance, using the tools and methods that are commonly used in Capgemini. Therefore, the literature review includes both of primary literature and secondary literature. Most of primary literature consists of reports available inside Capgemini, including some reports from other organizations like Gartner, Forrester, Oracle, and Cisco.

UML and flowchart analysis In this thesis, the high level design is using a subset of UML analysis and the detailed design is using a activity-diagram-like flow chart. The reasons to choose a subset of UML analysis, which is use case diagrams, are as follows. Firstly, for this case, UML analysis is relatively simple and there are no other sub languages needed. Besides, a high level design could be analyzed by using UML analysis, especially use case diagram, which is commonly used in Capgemini. The use case diagram provides an overview of usage from the outside perspective, which makes it convenient to communicate with people not sharing the expertise knowledge. As it turns out, use case diagram is good enough. Although using other available modeling tools, for example Archimate, could be helpful too, it's not a must, mainly because there is no real collaboration and only linear and local sequence of work.

Use case diagram is for high level overview usage of the chosen provisioning procedures, a detailed visualization tool is needed to compare the current "as is" process and "to be" process. The "to be" process refers to the process of using the network automation software NetMRI. This thesis chooses flow chart for detailed analysis. The reasons are as follows. Flowchart is simple and commonly used. Besides, flow chart fits the contents of detailed provisioning procedures as it shows clearly different choices and corresponding results. An official activity diagram can also be used for detailed design. But as the flow chart already achieves the goal of simple analysis and efficient communication, there is no need to use an activity diagram.

The detailed reasoning for using use case diagrams and flow chart are in Chapter 2 and Chapter 3, addressing as-is and to-be respectively. The reason to have a separate "to be" design chapter is to make it clear what should be changed in NetMRI, compared to the current procedures.

Building Test lab The first goal of this thesis is to automate network configuration and provisioning for the DMVPN service and to build a test lab on the software NetMRI to test whether it is a suitable working tool. The test lab will result in hands-on experience of how to use NetMRI for Dynamic

Multipoint Virtual Private Network (DMVPN) provisioning. Besides, it's a valuable test before implementing it in the production environment. Currently, available documents for Dynamic Multipoint Virtual Private Network (DMVPN) are high level design documents and detailed procedures documents. There are no documents that aim for communicating and explaining, which makes it difficult for people who are not familiar with DMVPN provisioning to understand the processes. Besides, a document to fill the gap between the high level design and detailed procedures is needed because it shows to people how NetMRI can automate the network provisioning in the content of Dynamic Multiple Virtual Network (DMVPN). As there are many procedures in the Dynamic Multipoint Virtual Private Network (DMVPN) provisioning, a modeling tool is needed for analyzing. Furthermore, a clearly defined scope is also needed for the test lab due to the limitation of time. A clearly defined scope of the test lab will also help to achieve the goal of the test lab efficiently.

For the scope of this test lab, it only focuses on the starting point of provisioning DMVPN spoke router inside Capgemini. This part comes after successful provisioning by the on-site engineer from the third party. The detailed reasons of why choosing this set of procedures will be explained in Chapter2.

1.3 Thesis outline

As mentioned at the beginning, this thesis research has two main practical goals. One goal is to introduce network automation using traditional product NetMRI, the other goal is to clarify whether using the traditional network automation product NetMRI is a good step on the road to software-defined network (SDN) for Capgemini. To investigate the first goal, the first step is analyzing the current network processes, which is Chapter 2. Analysis of current processes using UML. This chapter analyzes the current processes from a business perspective, to be specific, the function of the current processes. The next Chapter 3. Analyzing "to be" design in NetMRI is the design for the "to be" processes in NetMRI. Chapter 4. Empirical study, then discusses the implementation of the "to be" design from Chapter 3. The second goal is to figure out whether using the traditional network automation product NetMRI is a good step on the road to software-defined network (SDN). To this aim, Chapter 5. Software-Defined Network (SDN) roadmap, then introduces the chosen roadmap for SDN and discusses to what extent the second goal has been achieved. The last Chapter 6 is: Conclusion and Discussion, which summaries the results and future improvements for this thesis.

2 Chapter 2. Analysis of current processes using UML

This chapter analyzes the current provisioning process by clarifying reasons to choose a subset of UML analysis, following the detailed use case modeling. As the whole process is very long and as limitation of time is very strict, this thesis needs to define a clear scope and focuses on only a small set of procedures, which are helpful to make a decision whether NetMRI is a suitable working tool for DMVPN provisioning automation.

This chapter includes reasons to choose a subset of UML analysis, use case analysis overview, business view of current procedures and chosen scope, actor description, technical view of partial current provisioning procedures, and use case diagram.

2.1 The reasons to choose subset of UML analysis

This thesis aims to figure out whether NetMRI is a suitable working software for network automation. Within the defined scope, a visualization tool is useful to compare the current “as is” processes with the automated “to be” processes after using the software NetMRI.

As the aim of analyzing the process is to compare the current “as is” process with the “to be” process and answer the practical goal mentioned in introduction, it’s crucial to understand the current usage overview first, which gives initial insights. This overview “as is” process is analyzed from the business usage. L.P.J.Groenewegen, A.W.Stam, P.J.Toussaint, E.P.de Vink (2005) believe that “surrounding business can be modeled in the same object-oriented modeling language with enough expressiveness for coordination” (p. 14). UML modeling fits for an object-oriented language (Object Management Group, 2005). Thus, UML modeling can specify such “as is” business usage as the UML modeling fits for object- oriented language, which can model surrounding business.

The detailed reasons to choose a subset of UML are as follows. This subset is use case diagrams. Firstly, UML analysis then remains simple and there are no other sub languages needed. Besides, as a use case diagram represents goals that the user wants to achieve with a system (Object Management Group, 2014), a high level design could be analyzed by using UML analysis, especially use case diagram, which is commonly used in Capgemini. A use case diagram provides an overview of usage from the outside perspective. In the case of this thesis, Capgemini wants to analyze the network automation software NetMRI as a working tool: using a use case diagram makes it convenient to communicate with people not sharing expertise knowledge. Besides, use case analysis models the dialog between the users and the system and provides an external usage viewpoint (Visual Paradigm, 2011). As it turns out, use case diagram is good enough.

2.2 Use case analysis overview

This chapter gives an overview of the use case modeling, including the main concepts in the use case diagram and an usage overview of the DMVPN procedure by using the use case diagram. The details of modeling are described in sections 2.2 to 2.6.

The main concepts in the use cases diagram include actor and use case. An Actor is a role of an object(s) outside of a system that interacts directly with the system as part of a coherent work unit (Object Management Group, 2014). The use case analysis is the main service offered by the system, that is to say, main things the system is supposed to do. One use case is one main thing the system is supposed to do. As such, use case addresses a concrete objective that users want to achieve with a system (Visual Paradigm, 2011).

The structure of use case modeling is based on the template inside Capgemini: Use Case Modeling Guidelines. In these guidelines, the first part is the purpose of the use case, including: identifying requirements, boundaries and communication with customers. The next paragraph addresses requirements identified. Section 2.3 is about boundaries. Communications with customers are throughout the analysis session to ensure the analysis result fulfills customers' requirements.

"Dynamic Multipoint Virtual Private Network (DMVPN) is a connection service that interconnects multiple branch office local networks (spokes) to the central company network at the data center (hub) by means of encrypted VPNs" (Capgemini, 2014). The detailed information for DMVPN can be found in the appendix, which is in part 7. As the whole DMVPN processes are very broad, this thesis only builds the test lab based on part of the process and tries to figure out whether the traditional network automation tool NetMRI could be the suitable working tool. The two figures below show the overall usage of DMVPN. Figure 1 is from Capgemini product description document (Capgemini, 2014), which includes technical details. Comparing with Figure 1, Figure 2 shows a DMVPN usage overview in a use case diagram without showing the details of routers. There is detailed use case diagram in Section 2.6. Figure 2 aims to show the general usage of DMVPN. Once set up, the data center hub router has a static IP address and the spoke router from customer branch office are added to the DMVPN without having to change the initial DMVPN configuration of the hub router (Capgemini, 2014). The spoke router is assigned a dynamic IP address. As Figure 2 shows, each spoke router will connect to two data centers, which are the main data center and the backup data center at different locations. The reason to have connections in two data centers are to ensure the network can work continuously. That is to say, one of the VPN connections is standby.

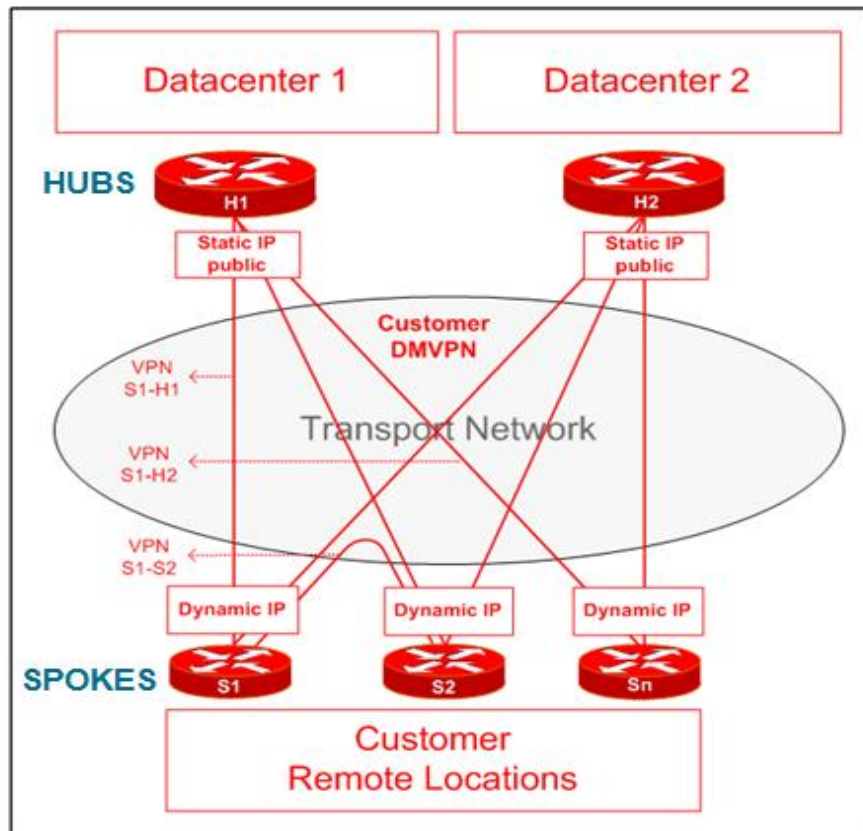


Figure 1 Solution design logical view (Capgemini, 2014)

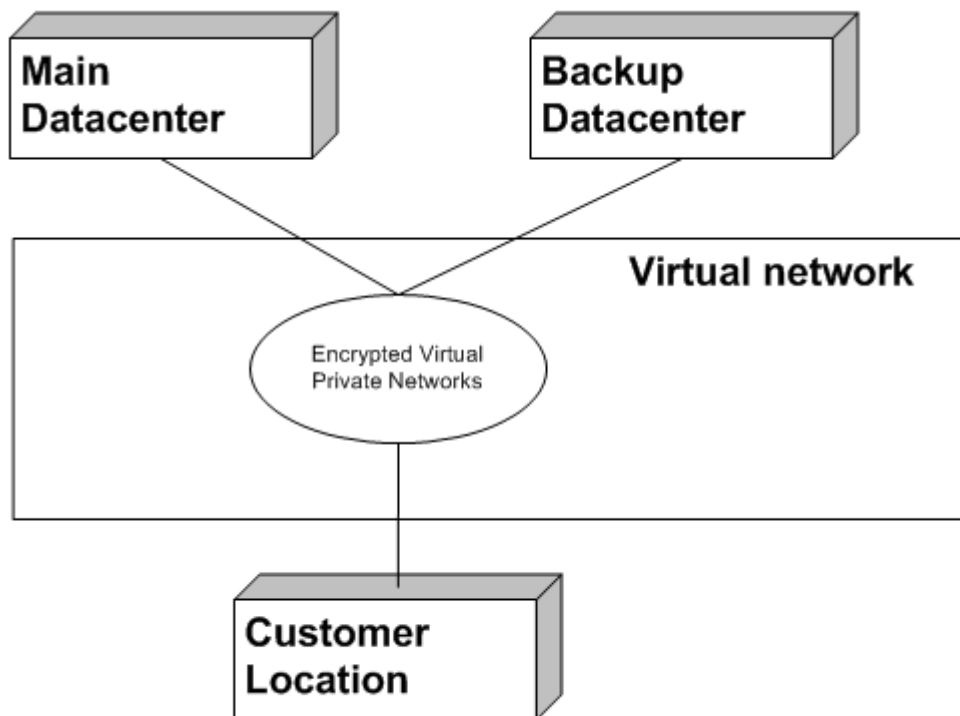


Figure 2 Overview use case diagram for DMVPN

2.3 Business view of current processes and the chosen scope

This section will explain the reasons to choose the specific subset of the whole DMVPN provisioning service and to define the assumptions for the test lab.

Business view of current processes provisioning DMVPN

At Capgemini, the DMVPN services include standard services and nonstandard services. In general, there are three different roles involved in the processes: client or customer, third party on-site engineer, and Capgemini engineers.

Currently, the working process of standard service is shown in the Figure 3 below.

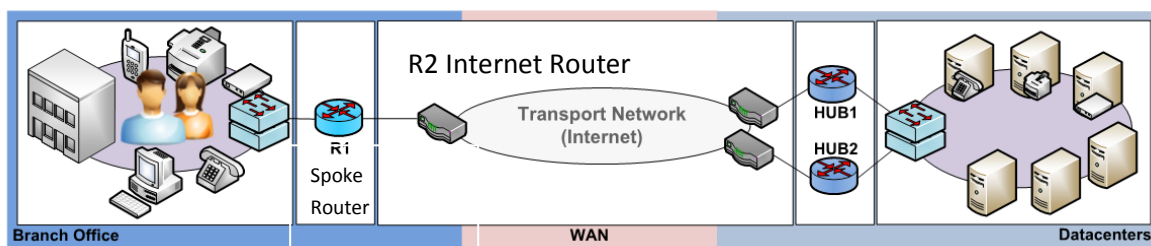


Figure 3 Working process of standard DMVPN service (Capgemini, 2014)

As the figure shows, the client or customer, shown at the left side of the figure, needs a new branch office to access the DMVPN hub routers at the data centers. The branch office is the customer location mentioned in Figure 2. The data centers at the right side of Figure 3 are the main data center and the backup data center mentioned in Figure 2. Between the left side and the right side of Figure 3, which is shown with a pink border, is the encrypted virtual network.

Current Processes for standard DMVPN services consist of 3 roles:

- Client: requests network connection for a new branch office
- Third party: puts the spoke router, which is shown as R1 in the figure, in the remote office location of client and builds the basic connection to the data center
- Capgemini: configures provisioning procedures manually

The three steps below show the detailed tasks finished by these three roles.

1. Client requests a network connection for a branch office.

Client requests for the DMVPN connection for its branch office with basic requirements filled on a web portal. There are three connection options, including wire, wireless, wire + wireless.

2. Third party pre installation.

2.1. Capgemini sends the request to the third party.

Capgemini engineers fill the standard information form in the portal website to order the routers, referring to the spoke router R1. This device has an Ethernet connection both to the inside LAN and the internet router.

2.2. Third party puts the router in the location and builds the basic connection to the data center router, which is a virtual router acting as a stepping stone and just receiving data. The third party may also provide an internet router, which is shown in the figure above as R2. This router acts as an interface router to connect different Internet connections. The partner field engineer from a third party will build the VPN tunnel to the staging hub. Staging routers means basic and standardized configuration finished by the third party on-site engineer. This standardized configuration could be seen as a preparation step before the provisioning steps in Capgemini.

After provisioning from the third party on-site engineer, a pre-install tunnel is built between each spoke router and the stepping stone (pre-install) hub router.

3. Manual provisioning procedures by Capgemini engineer

Then the Capgemini engineer will finish manually a series of provisioning procedures with lots of configure documents. Below is the description of the document in the scope of this thesis, reflecting the reason why Capgemini wants to automate the provisioning. This document is the work instruction to prepare a spoke router.

Steps of the spoke router configuration

- Configure the LAN and WAN interfaces
- Build the tunnel by using certificates

In this document, there are three routers involved, spoke router, pre-install hub router, and hub router. Spoke router is the branch office router that client requests for DMVPN connection. The pre-install hub router acts as a stepping stone. The hub router is the aimed router at the data center that the branch office router aimed to connect. To build the tunnel between spoke router and the hub router, the spoke router needs connect with the management network and PKI infrastructure, which will be provided by the pre-install hub router. For security reasons, the pre-install tunnel between the spoke router and the stepping stone (pre-install) hub router will be terminated after the tunnel between spoke router and the hub router is built.

The Figure 4 below shows the sequence of building the tunnel. The stepping stone (pre-install) hub router and the hub router are connected to management Network and PKI Infrastructure, which are shown as tunnel 0 in the figure. The pre-install tunnel between spoke router and the stepping stone hub router is the tunnel 1, which is a temporary VPN tunnel based on pre-shared key. The aimed tunnel to build is the tunnel 2 in the figure, which is a VPN tunnel based on PKI infrastructure. The stepping stone (pre-install) hub router will only give the spoke router access to the PKI infrastructure. The Management Network can access to spoke router via the stepping stone (pre-install) hub router, but not the other way around .

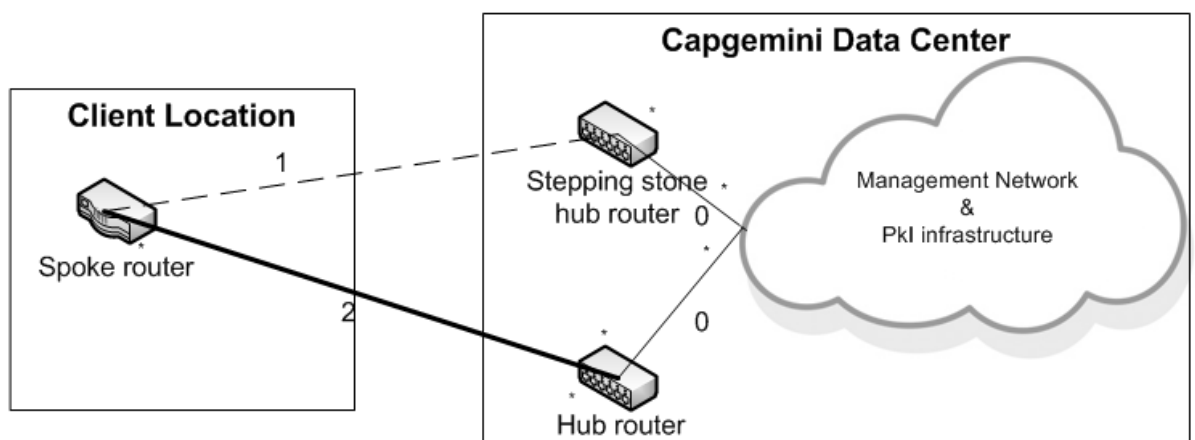


Figure 4 Sequences for building the tunnels

There are also some other documents involved in the complete version of DMVPN installation. The function of these documents is out of the scope of this thesis. The reasons why these documents are out of scope of this thesis are given in the next part "chosen scope". The document "Basic description of installation flow.txt" gives an overview and general description of all the documents involved.

Chosen scope

Due to limitation of time, the chosen scope of this thesis is a subset of steps that are supposed to be finished by Capgemini engineer. This set of steps aims to figure out whether NetMRI is the desirable network automation tool. The reasoning is as follows.

Procedures for the on-site engineers from the third party are not included in the scope, because the procedure has been finished by the third party without problem by now. After on-site engineers' configuration from the third party, the spoke router has a basic configuration and a temporary tunnel

to the pre-install (stepping stone) hub router at the data center. This tunnel is the pre-install (Virtual Private Network) VPN tunnel. Besides, the problems for Capgemini are occurring during the manual provisioning, which is the main reason to conduct this thesis. But there are possibilities for Capgemini to automate the pre-install procedures finished by the on-site engineers from third party engineers in the future.

As observed in the previous part of this session, even inside Capgemini, there are lots of procedures and different documents involved. The starting point of provisioning inside Capgemini is to build up the tunnel between spoke router and the hub router via stepping stone (pre-install) hub router and dismiss the tunnel between spoke router and the stepping stone hub router, which is the pre-install tunnel. Based on the use case overview in Figure 2, there are connections to the main data center and the backup data center. As the procedures to build the connection are similar, we only focus on building the connection to the main data center. As the procedure to terminate the pre-install tunnel is similar to the procedures to build it, the procedures to build the tunnel between stepping stone (pre-install) hub router and the spoke router are representative. If the procedures to build the tunnel can be implemented successfully in NetMRI, it is most likely the remaining procedures could also be implemented successfully in NetMRI. Besides, the spoke router is supposed to connect two hub routers in different data centers as shown in Figure 2. But in the provisioning procedures for these two hub routers are exactly the same. Besides, there is only one aimed hub router in the test lab. So we only build the connection to one hub router in the test lab.

To sum up and clarify all the assumptions for the chosen scope, the list below shows all the basic assumptions. Our analyzing and modeling of the current processes are based on the following basic assumptions.

- This thesis only focuses on the starting procedures of standardized DMVPN services inside Capgemini.
- The DMVPN hub routers, including the pre-install hub router and the hub router aimed to connect with the spoke router, have been configured and are operational.
- The stepping stone (pre-install) hub router and the hub router have been connected to the management network and the PKI infrastructure, located at the data centers.
- The pre-install tunnel, which is a VPN connection based on pre-shared key, between the spoke router and stepping stone (pre-install) hub router has been installed by the on-site engineers from the third party.
- Spoke router has a basic configuration, which has been installed correctly by the on-site engineers from the third party.

2.4 Actors description

The use case is created by identifying all the actors in the defined scope. The actors include administrator, pre-install hub router, spoke router, aimed hub router, Infoblox network services software, certificate server, Remote Desktop Protocol-client (RDP-client), and Microsoft Management Console (MMC). This section clarifies choice and definition of actors in our use case analysis, by indicating how these actors arise from the provisioning procedures.

Based on the Oracle White Paper, an actor is anyone or anything outside the system, with behavior comprising some usage as offered by the system. Thus, an actor can be a person, a piece of hardware or a different system. That is to say, hardware could also be an actor, for instance, a server outside the system communicating with the system. There are different levels of actors: primary actor and supporting actor. The primary actor has an interest in the delivery of a specific goal of a use case and interacts with the system under discussion. Often it is the one who triggers the use case, but there are some exceptions, when the use case is triggered by time or triggered by an actor who does this on behalf of the real primary actors. Supporting actor is an (external) actor who needs to provide a service to the system. An actor can be the primary actor for one use case and the supporting actor for another. In addition, it is important to discover these “hidden” actors at an early stage as it may reveal requirements that may be missed otherwise (Oracle, 2005).

If we keep track of all the detailed provisioning steps, there is only one actor from first sight. This actor is the administrator, who is carrying out these provisioning procedures. As administrator has interests of a specific goal of the use case and the one who triggers the use case, so the administrator is the primary actor. As a piece of hardware or a different system could also be an actor (Oracle, 2005), these include routers, Infoblox network service software (DDI), Remote Desktop Protocol – client (RDP-client), Microsoft Management Console (MMC) tool, and certificate server. RDP–client is to make a remote connection to a server and MMC is a management control tool. These actors are selected by identifying the subject who initiates the action of each step. There are many interactions between different routers. Routers with different functions are involved in the defined scope, including the pre-install hub router, spoke router, the two hub routers at the data center. We regard the specific router, which is hardware, as having behaviors, for instance, they receive a request, execute the request and provide output if needed. As spoke router and pre-install hub router (stepping stone hub router) are acting on behalf of the primary actor being the administrator, which is an exception for primary actor (Oracle, 2005), these two routers are primary actors too. Aimed hub routers at the data center are not initiating any behavior, so this router is a supporting actor. Infoblox network services software (DDI) and the certificate server are the external actor providing service to the whole provisioning procedures. Certificate server, Remote Desktop Protocol – client (RDP- client)

and the Microsoft Management Console (MMC) tool are also involved. But these steps need checks and approval from an administrator, so these actors are supporting actors. To sum up, Table 2 Actors for use case analysis below shows the actors in the boundaries of this use case analysis and their types.

Actors	Type
<i>Administrator</i>	Primary
<i>Stepping stone hub router (pre-install hub router)</i>	Primary
<i>Spoke router</i>	Primary
Aimed hub routers in data center	Supporting
Infoblox network services software (DDI)	Supporting
Certificate server	Supporting
Remote Desktop Protocol – client (RDP-client)	Supporting
Microsoft Management Console (MMC)	Supporting

Table 2 Actors for use case analysis

2.5 Technical view in scope of current provisioning procedures

This part presents my analysis of the current detailed installation procedures in the chosen scope for the use case diagram, by first giving the normal sequence flow of detailed original procedures. The normal sequence flow of detailed procedures is based on document “Capgemini DMVPN Spoke Installation Procedure v2.3” (Arendsen, 2014), which is the technical provisioning procedure document using Cisco Command Line Interface (CLI). As mentioned in the beginning of this chapter, the scope of use case analysis is the first 29 steps. The provisioning procedures in this chapter focus on the function of each step, instead of showing the original commands in the installation document. But there is still a lot of network terminology for people who don’t have network knowledge. More information about network terminology can be found in the Appendix. Not up to now, there is no use case diagram available in Capgemini. So another reason to represent the detailed procedures by use cases is that it can give people who don’t have networking experiences a rather thorough idea of provisioning. Moreover, it will help people to understand how a use case analysis is conducted.

1. Prepare for the certificate issuing

As the starting point of the procedure, this step aims to connect the Stepping stone hub router at the data center to the other system (RDP-client) and prepare for Private Key Infrastructure (PKI) in MMC. RDP refers to Remote Desktop Protocol. MMC refers to Microsoft Management Console tool.

2. Telnet to the pre-install hub router

The pre-install hub router used for stepping stone is connected to the new to be installed locations. “To Telnet” means: connect to the hub router through the desktop. Via Telnet, the user can send a command line to the hub router.

3. Locate the router

To get spoke router’s IP address from stepping stone (pre-install) hub router, this step uses the Cisco Discovery Protocol (CDP) to discover basic information of neighboring routers.

4. Connect to and login at the spoke router

Using the pre-install hub router to Telnet the spoke router on the basis of its IP address got from the last step. From this step, the engineer starts to use the spoke router.

5. Check required settings

This step aims to check software versions and the hostname of the spoke router, which has been installed by the third party on-site engineer.

6. Configure to see the output of PKI-commands

The result of this step will be demonstrated in later steps to show all messages in real time.

7. Check time synchronization

8. Configure certificate server hostname and IP address for PKI

Turn to configure mode and configure with the certificate server hostname and IP address.

9. Configure root certificate

In this step, it uses command “crypto pki trustpoint”, which allows to declare the root certificate authority (CA) and to specify characteristics for the CA.

10. Authenticate the trustpoint and install the root certificate

11. Copy certificate

12. For checking purpose, display the imported certificate

13. Configure Private Key Infrastructure (PKI) certificate trustpoint

14. Configure certificate subject name with the correct parameters.

15. Authenticate the trustpoint and request the device certificate

16. Wait for spoke router to report back

17. For checking purpose, display the certificate request. Check if spoke router has sent the certificate request to the CA.

18. Open prepared RDP-session to the stepping-stone server for PKI management

19. In MMC with Certificate Authority Plug-In select Pending Requests
20. Select the certificate request, right click the request, select All Tasks, and select Issue
21. Wait for the spoke router to receive the certificate
22. Exit configure
23. Copy the config-file for the chosen router type. In this test lab, we use router type: Cisco 88x.
24. Copy tunnel configuration for router type Cisco 88x
25. Get IP address from Infoblox

Configure relevant management interfaces with IP address getting from Infoblox.

26. Set up 2 new EIGRP tunnels

Enhanced Interior Gateway Routing Protocol (EIGRP) uses bandwidth to calculate a metric for routing protocol and discover a neighbor before sending routing information. This step aims to wait until tunnel established and EIGRP routing activated.

27. Verify the connection with two new tunnels' state.
28. Save configuration and exit the session with the spoke router (customer router)

This step is the end of the configuration in the spoke router. In step 4, the engineer starts to use the spoke router. This step logs out from the spoke router. The engineer is on the behavior of the pre-install hub router again.

29. Exit the Telnet session with the stepping stone (pre-install) hub router.

This is the end point of the whole chosen procedure. Step 2 start a Telnet session to the pre-install hub router. This step is the end of the Telnet session. As these 29 steps are part of the whole provisioning procedure, the tunnel termination is not covered here.

2.6 Use case diagram

This section describes general scenarios for six use cases based on the defined scope. The reasons to have these general scenarios are as follows. As shown in the previous section about the Technical view in scope of current provisioning procedures, the 29 procedural steps are separated based on technical perspective. That is to say, these procedural steps are not separated from a more functional perspective, which makes it difficult to create a use case diagram as some steps can combine into one use case. To have a better overview of which actor is doing what, the more general scenario below groups these detailed procedural steps according to which actor is finishing this function. The actors were defined in chapter 2.4: Actor description above, including administrator, pre-install hub router, spoke router, aimed hub router, Infoblox network services software, certificate server, Remote Desktop Protocol-client (RDP-client), and Microsoft Management Console

(MMC). The general scenario actually shows the reason for constructing a use case diagram based on the current detailed technical procedural steps. After the general scenario, this section will present the use case diagram and the various use case description for each use case.

There are actually two versions of the general scenario due to the improvements after test lab implementation. Analysis of constructing the general scenario is carried out as follows. The same name of both general scenario versions reflects the goal of the chosen scope: Set up a tunnel between the spoke router and aimed hub router at the data center. The statement in each step consists of who or what is doing what. The first version of the a general scenario is as follows. The first step in general scenario is based on step 1 in Technical view in scope of current provisioning procedures, which are finished by the administrator. From the second step, the administrator login the pre-install hub router (stepping stone hub router). As the pre-install hub router (stepping stone hub router) is on behalf of the primary actor administrator (Oracle, 2005), which is an exception for primary actor, thus it is a primary actor. The second step in the general scenario is based on steps 2 to 4 on Technical view in scope of current provisioning procedures. After step 2, the provisioning begins on spoke router. As the first version shows, the spoke router starts installing the certificate. But from this step, all the rest procedures aims to build the tunnel between the spoke router and aimed hub router at the data center. Step 3 in first version only shows the starting point of building the tunnel. But the order of steps is not correct if we separate the steps as a first version: step 5 should be between step 3 and step 4. Besides, the administrator needs to approve the certificate request via other systems, which are Remote Desktop Protocol–client (RDP-client) and Microsoft Management Console (MMC). As the step 4 in the first version should be moved after step 5, the last step 6: verifies the connection in first version is actually part of the first version step 4, which means these two steps can be combined together. Thus the improved second version is shown after the first, more preliminary version.

First version: Set up a tunnel between the spoke router and aimed hub router at the data center

1. The admin: prepares for PKI management
2. The pre-install hub router: Telnets to the spoke router and find its IP address
3. The spoke router: installs PKI certificate
4. The spoke router: configures the tunnel to the hub routers by using configuration files
5. The admin: gets an IP address from Infoblox software
6. The admin: verifies the connection

Second version: Set up a tunnel between the spoke router and aimed hub router at the data center

1. The admin: prepares for PKI management
2. The pre-install hub router: Telnets to the spoke router and find its IP address

3. The spoke router: installs PKI certificate
4. The admin: approve certificate request
5. The admin: gets an IP address from Infoblox software
6. The spoke router: configures the tunnel to aimed hub router in data center

Based on the general scenario listed above, Figure 5 Use Case Diagram consists of actors and use cases initiated by actors, which are mentioned in the Chapter 2.4 Actors description. This use case diagram is part of the way to provision the virtual network mentioned in Figure 2. Some use cases have two actors involved. One is primary actor, the other is supporting actor. This use case diagram uses different figures to represent different actors. The reason to use a human figure for the administrator is that the administrator is a human. As for the pre-install (stepping stone) hub router and spoke router, they are hardware. It's better not to use the human figure to represent these actors as it doesn't show clearly these are hardware instead of human. But there is no figure of a router in UML and particularly not in use case diagrams. But in deployment view of UML, we find "a node is a run-time computational resource, such as computers or other device" (Rumbaugh, Jacobson, & Booch, 2004). So we could use the figure "node" in UML to represent the routers. As the remaining actors are supporting software, including a certificate server, RDP-client, MMC, and Infoblox network service software (DDI), we use a component visualization from UML: "components are considered autonomous, encapsulated units within a system or subsystem that provides one or more interfaces" (Bell, 2004). As we only use part of encapsulated units within these systems, for example certificate server, we can use a component visualization represent to these actors.

To clarify the details of each use case, this thesis uses use case description template, according to "Use Case Modeling Guidelines" from Capgemini (Borao, 1999) and An Oracle White Paper: Getting started with use case modeling (Oracle, 2005), which is based on the book "Writing Effective Use Cases" from Alistair Cockburn. A use case description starts with a general description of the usage of this use case, to clarify the correspondence between the use case and the various step X, with $X \in \{1, \dots, 29\}$, from the current provisioning procedures mentioned in Section 2.5 to the reader. The description for each use case then includes the steps involved in current provisioning procedures. Primary actor initiates the use case to achieve a goal. Preconditions clarify the conditions that must hold true before the main flow of the use case starts. Alternate flows addresses error conditions that could prevent successfully achieving a specific step in the main flow. Use case names refer to the six grouping from the second general scenario version on page 24.

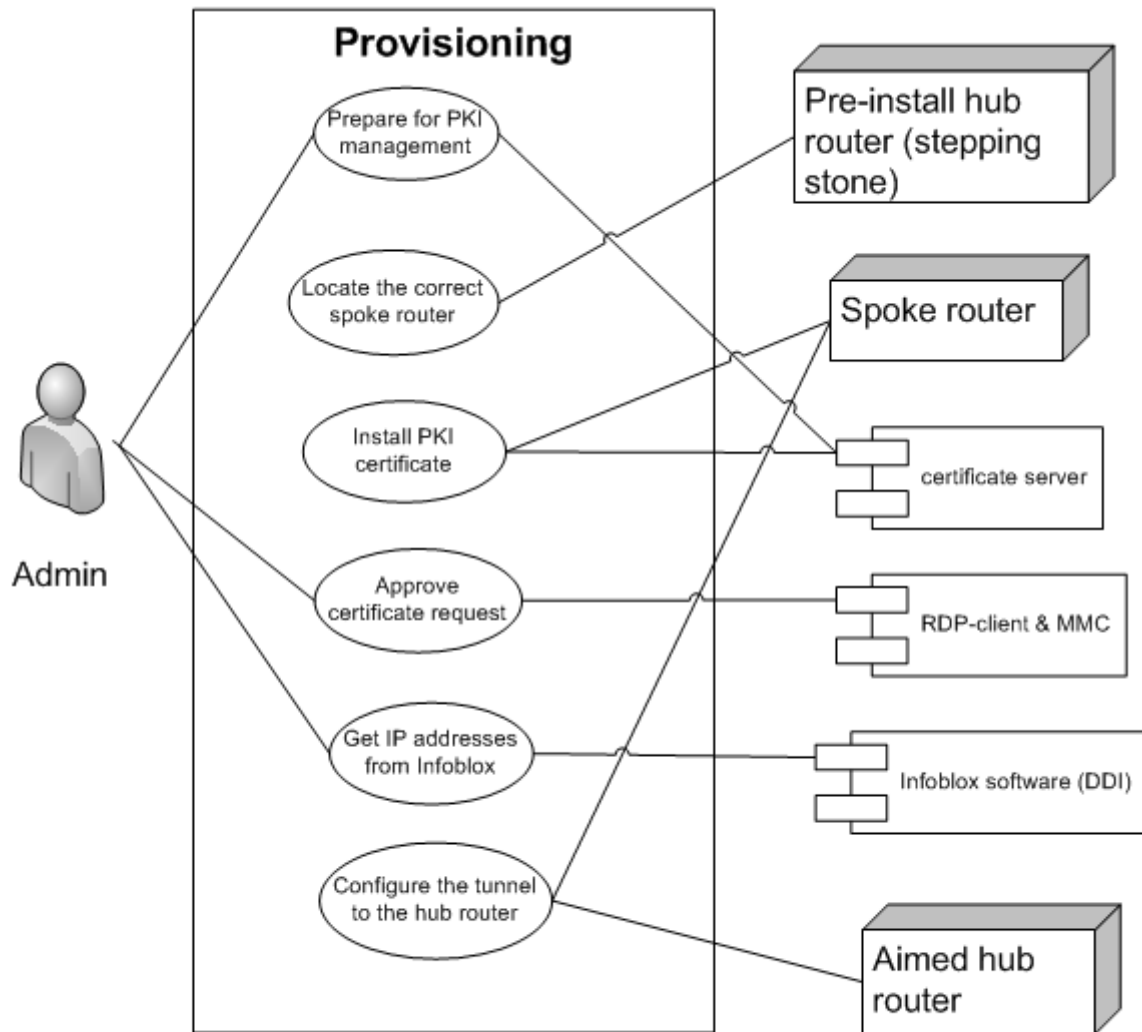


Figure 5 Use Case Diagram

Use case: <<Prepare for PKI management>>

Description: This use case is initiated by the admin. It prepares for the certificate issuing by connecting to different management tools, including RDP-client and MMC. This use case is according to step1 in Section 2.5.

Primary Actors: Admin

Preconditions: RDP-client, certificate server, and MCC are operational and accessible.

Main flow:

1. Connecting with an RDP-client to the certificate server
2. Open MMC for certificate management

Alternate Flow:

- 1.A If connection failed. Troubleshooting and get connected
- 2.A If can't open MMC, trouble shooting for reasons

Use case: <<Locate the correct spoke router>>

Description: This use case is initiated by the pre-install hub router (stepping stone hub router). The goal of this use case is to Telnet to the spoke router, which means the first step is to find the spoke router's IP address. The original name of this use case is "Telnet to the spoke router", but this use case name is too narrow. With the use case name "locate the correct spoke router", it could include the use case "find the corresponding IP address" into one use case. This use case is based on step2 to 4 in Section 2.5.

Primary Actors: pre-install hub router

Preconditions: Successful provisioning by the on-site engineer from the third party.

Main flow:

1. Telnet to the pre-install hub router
2. Find the right spoke router IP address
3. Telnet from the pre-install hub router to spoke router

Alternate Flows:

- 1.A If can't Telnet to the pre-install hub router, troubleshooting to solve the problem
- 2.A If the IP address is not in expected format, try to find out which IP is the correct one
- 3.A If can't Telnet, ensure the password for Telnet is correct first
- 3.B If the password is correct, try to trouble shooting

Use case: <<Install PKI certificate>>

Description: This use case is initiated by the spoke router and the goal is to start configuring the PKI certificate installation. The result should be sending certificate request successful. This use case is based on step 5 to step 17 in Section 2.5. It gives various differences. Originally, this use case lists all the detailed technical procedures, but it is more clarifying to group these procedural steps by function.

Primary Actors: spoke router

Preconditions: The pre-install hub router gets access to the spoke router by Telneting.

Main flow:

1. Check the required setting, time synchronization, and PKI-commands
2. Send the certificate request.

Alternate Flows:

- 1.A If check out something is not correct, resend the command to ensure the settings are correct
- 2.A If can't send a certificate request successfully, try again and trouble shooting

Use case: <<Approve the pending request>>

Description: This use case is initiated by the administrator and the goal is to approve the certificate request in other management tool, including RDP-client and MMC. This use case is based on step 18 to step 22 in Section 2.5.

Primary Actors: Administrator

Preconditions: certificate sent successfully to the certificate server by spoke router, RDP-client and MCC are prepared.

Main flow:

1. Open prepared RDP-session to the stepping-stone server for PKI management
2. Approve the certificate request in MMC
3. Close RDP session

Alternate Flows:

- 1.A If can't find the correct certificate request, refresh until finding the correct request.
Otherwise trouble shooting

Use case: <<Get IP addresses from Infoblox>>

Description: This use case is initiated by the administrator and goal is to get the right IP address from Infoblox, which is the mother company of NetMRI. Configure the relevant management interfaces with IP addresses by copying and pasting the correct information from the corresponding file. This use case is based on the step 25 in Section 2.5. But this use case is itself another process related to Infoblox software. So the details of this use case are not included.

Use case: <<Configure the tunnel to the hub router>>

Description: This use case is initiated by the spoke router. It is prepared for the PKI installation. This use case is based on the step 23 to step 29 in Section 2.5, except for step 25, which is mentioned in the last use case.

Primary Actors: spoke router

Preconditions: Spoke router sent the certificate request and get an IP address from Infoblox network services software (DDI). The certificate request is approved.

Main flow:

1. Wait for the router to receive the certificate
2. Continue the rest configuration for setup new tunnels
3. Verify connection

Alternate Flows:

- 1.A If the spoke router didn't receive the certificate, wait for the spoke router to receive the certificate
- 2.A If a tunnel doesn't set up successfully, try to trouble shoot the reasons.

As we can see from analyzing of current provisioning procedures to use case diagram, the use case analysis focuses on the usage of the whole process. Such usage should be achieved as much as possible in the “to be” design in NetMRI, which is explained in the next chapter.

3 Chapter 3. Analyzing “to be” design in NetMRI

This chapter elaborates the “to be” design in NetMRI. Because Capgemini wants to keep the new provisioning procedures as much the same for the current provisioning procedures as possible, the “to be” design aims to follow the current provisioning procedures. The reason to have a separate “to be” design chapter is to make it clear what will be different in NetMRI and because of NetMRI only, compared to the current procedures. In addition, this “to be” design could also give a clear view of what should actually be done, which will help to decide which part should actually be built first in the test lab.

To have a better understanding of the current provisioning procedure, current procedures are realized following the current Telnet method before the “to be” design. That is to say, the current “as is” procedures are tested exactly as the current way of realization. Based on the detailed assumptions, which will be listed in the Section 4.3 Set-up of the test lab, detailed current procedures are tested manually in the terminal emulator software PuTTY with a high level understanding of each step. In other words, it will be tested whether the original procedures could be realized successfully by using Cisco Command Line Interface (CLI) in PuTTY. The detailed procedures can be found in the document: LAB Installation procedure DMVPN-location (Capgemini, 2015) section 2: Procedure.

Figure 6 Changes in use case between “as is” and “to be” shows the differences between “as is” procedures and overview changes in “to be” design. Because certificate request should be approved manually for security reasons, use cases related to certificate approval will not automate in NetMRI. These use cases are initiated by the administrator. That is to say, all the use cases initiated by an administrator should be kept and the use cases with underline are aimed to automate in NetMRI. For the use case “get IP address from Infoblox”, this use case related to another process. It’s not included in the scope of this thesis.

Thus we have the following list of use cases that should be automated in NetMRI.

- a. <<Locate the correct spoke router>>
- b. <<Install PKI certificate>>
- c. <<Configure the tunnel to the hub router>>

This means, from step 1 – step 29 (in Section 2.5) exactly step 2 – step 17, step 23 – step 24, step 26 – step 29 are to be automated in NetMRI.

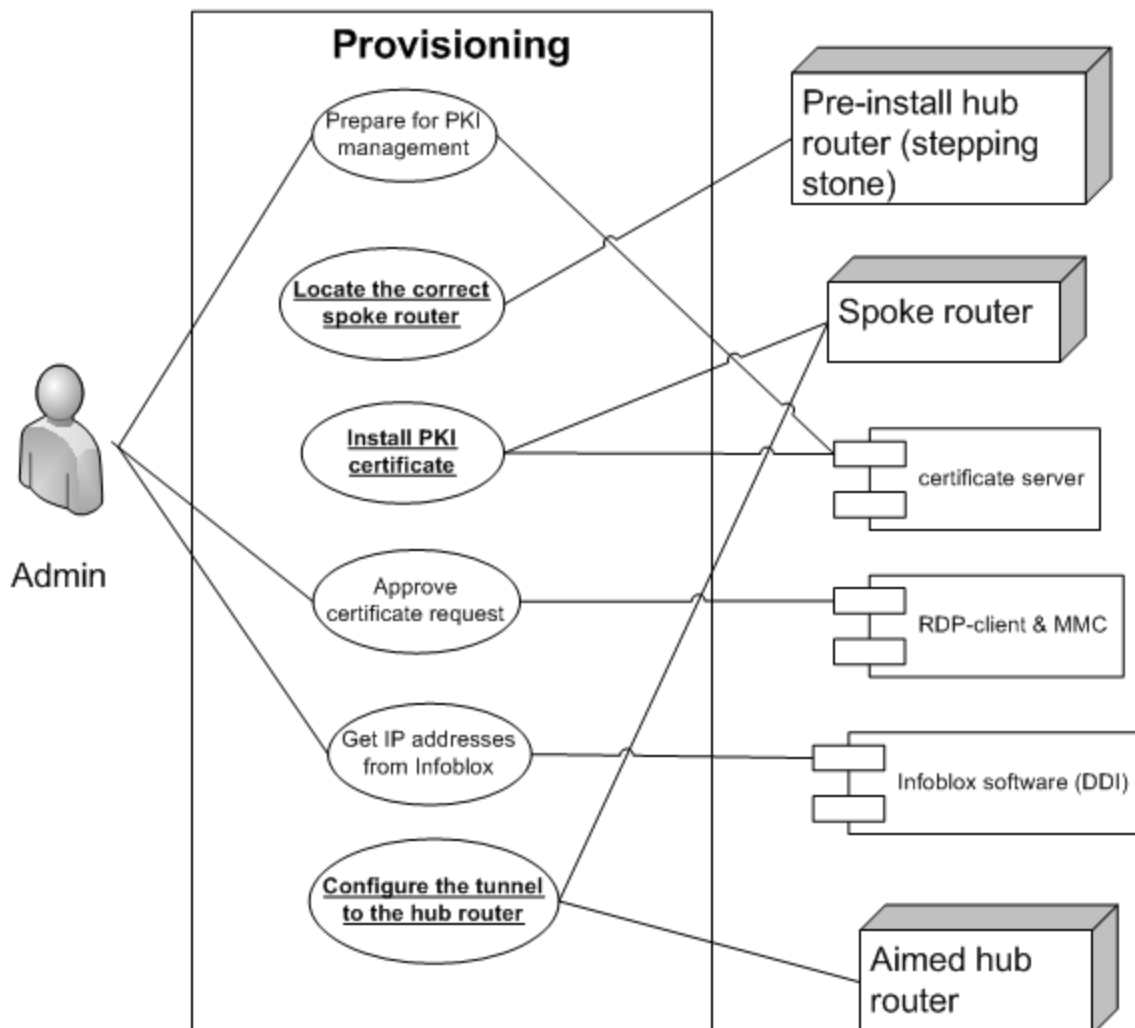


Figure 6 Changes in use case between "as is" and "to be"

This chapter will explain the creation of “to be” design by using flowchart in NetMRI context by clarifying the reasons to choose flowchart, following analysis from “as is” process to “to be” design using flow charts.

3.1 Reasons to choose flowchart

As a use case diagram only provides a high level view of what are the functions, a visualization tool is needed to analyze the detailed “to be” design. Besides, detailed procedures in NetMRI need to be analyzed as preparation for implementation of the test lab.

The reasons to choose a flow chart are as follows. As mentioned in the introduction, flow charts are simple and commonly used. Moreover, flow chart fits the contents of detailed provisioning

procedures as it shows clearly different choices and corresponding results. An official activity diagram could also be used for detailed design. But as the flow chart already achieves the goal of simple analysis and efficient communication, there is no need to change to use an activity diagram.

3.2 From “as is” process to “to be” design using flowchart

As NetMRI is a software system with lots of functions. To make it easy to analyze the “to be” design, the analysis will be divided into two parts: the overview of “to be” process in NetMRI and the initialization of “to be” process in NetMRI. The first part is “to be” process overview in NetMRI, which will come into execution only after the initialization step. The “to be” process overview in NetMRI shows the working processes in NetMRI if everything has been successfully implemented, which is based on “initialization process” part. To be specific, initialization of “to be” process in NetMRI means things need to be prepared and finished getting the same result as the current provisioning procedure. The same result in this thesis means to finish 29 the steps shown in chapter 2, which is to build the tunnel between spoke router and the hub router at the data center. To sum up, the part of the initialization in NetMRI is the preparation and the part of “to be” process overview in NetMRI is a high level design of how to realize the same result as current process in NetMRI.

The design starts with the overview “to be” process in NetMRI is designed. The reasons are as follows. The “to be” process overview in NetMRI should be as much same as possible compared to the current procedures. The “initialization process” is about how to achieve the goal of building the tunnel between spoke router and the hub router at the data center successfully. Besides, the use case analysis already gives a good understanding of what should be achieved in “to be” process overview in NetMRI. So the “to be” process overview in NetMRI is easier to start with compared to the “initialization process”. It will also ensure to achieve the goal of building the tunnel between the spoke router and the stepping stone (pre-install) hub router. This goal is crucial for investigating whether NetMRI is a suitable working tool for network automation provisioning. These are the reasons why this thesis will first focus on the “to be” process overview in NetMRI. The details of these two parts will be explained in the following two subsections.

3.2.1 “to be” process overview in NetMRI

To realize the goal of building up the tunnel between the spoke router and the stepping stone (pre-install) hub router, this part will first give an overview of what need to be finished in the context of NetMRI. The Figure 7 “to be” process overview flowchart in NetMRI shows the overview. The reason to call this “to be” process overview in NetMRI is that this flowchart is just an overview design considering the basic function in NetMRI.

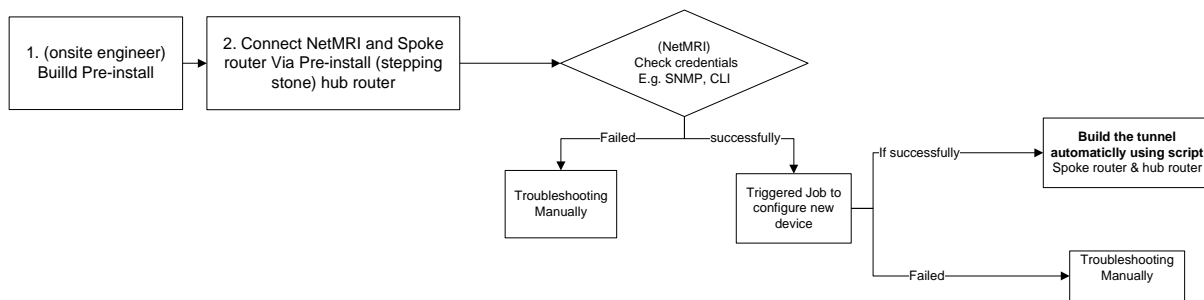


Figure 7 “to be” process overview flowchart in NetMRI

The starting point of “to be” process overview in NetMRI is after the on-site engineer finishing the pre-install tunnel between the spoke router and the stepping stone (pre-install) hub router. The detailed descriptions of the function of this step could be found in chapter 2.3 Business view of current processes and the chosen scope. The initial starting point is shown as step 1 in the flow chart. The reason to include the on-site engineer in this overview is to ensure a better use of NetMRI because there might be changes for all the steps. As the figure shows, the next step is: Connect NetMRI and Spoke router via pre-install (stepping stone) hub router. The reason to use a stepping stone (pre-install) hub router to connect spoke router and aimed router at the data center is for security reasons. The remaining steps are supposed to be finished automatically by NetMRI if implemented successfully in NetMRI. These steps include use case “locate the correct spoke router”, “install PKI certificate”, “configure the tunnel to hub router” in the use case diagram. In NetMRI, the first step to add a new router is to check credentials, following trigger job to configure the new device and build the tunnel automatically.

To build the tunnel automatically, the configuration commands should be added in NetMRI in a script, which uses script language Perl. As this “to do” design is finished without hands on experience on NetMRI and the next Chapter 4: Chapter 4. Empirical study addresses the implementation of “to be” design, the “to be” process overview summarizes all these use cases as “build the tunnel automatically using scripts”. NetMRI will first check the credentials of the new-added spoke router. For example, password to login to the router and password for different configuration mode for a router. The detailed description of these credential will be explained in chapter 4. If the credential checking is successful and the pre-defined job is triggered successfully, NetMRI will execute automatically the predefined procedures to build the tunnel. The pre-defined triggered job is a mechanism in NetMRI to trigger the pre-defined script, which contains Command Line Interface (CLI) in current provisioning procedures. This predefined procedure is called Job and Script in NetMRI.

3.2.2 Initialization the “to be” process in NetMRI

To realize “to be” process overview flowchart in NetMRI mentioned in last subsection 3.2.1, lots of detailed procedures need to be finished in different functional areas in NetMRI. This part will give an explanation of how to realize the functions mentioned in the subsection 3.2.1. Figure 8 Initialization of “to be” process in NetMRI below shows realization process in NetMRI.

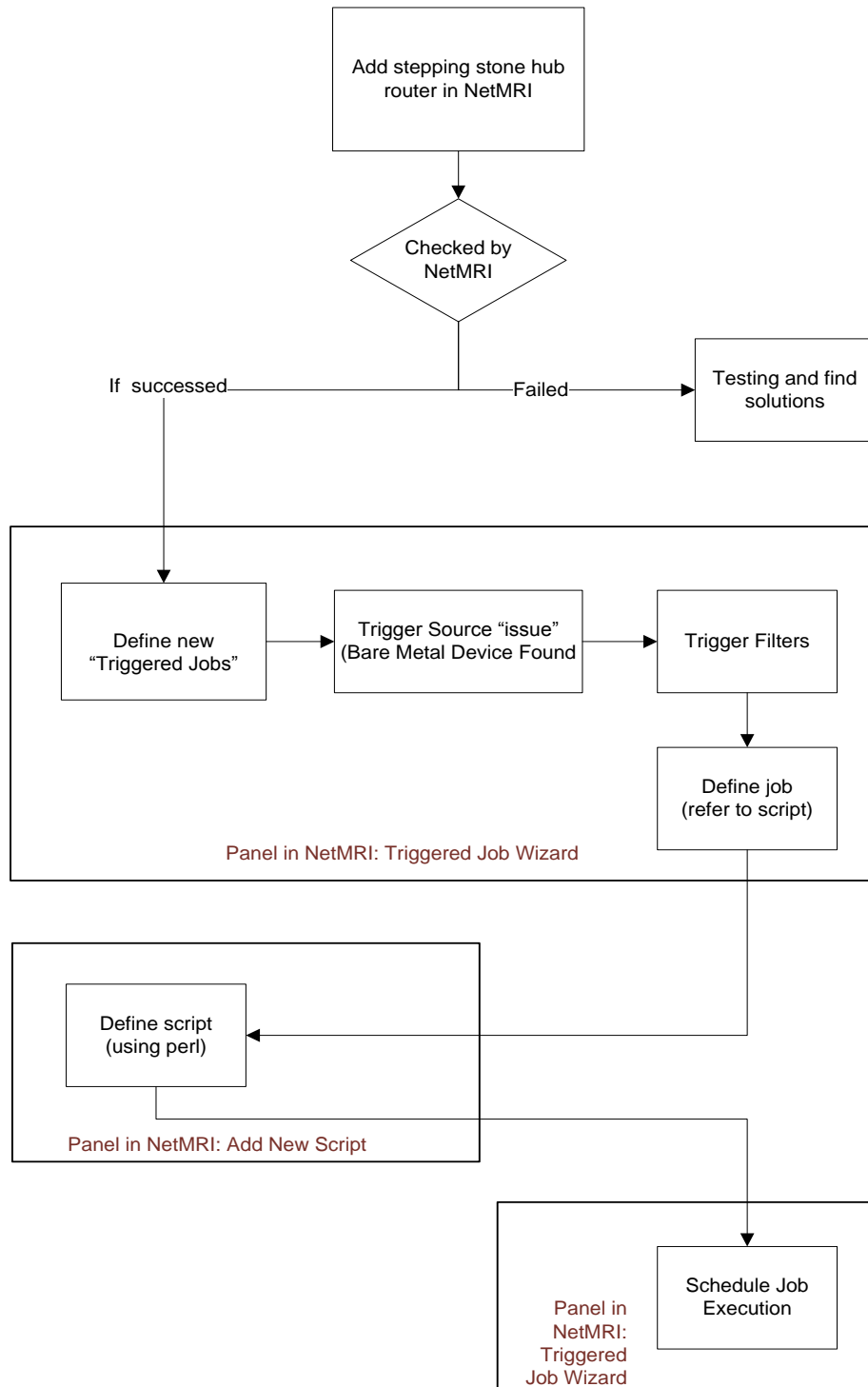


Figure 8 Initialization of “to be” process in NetMRI

Figure 8 Initialization of “to be” process in NetMRI addresses how to realize Figure 7 “to be” process overview flowchart in NetMRI. Because there are different function panels in NetMRI, the goal of this flowchart is to clarify which panel is going to define what part. The boundaries of the flow chart indicate which panel in NetMRI should be looked into to finish the setting. For instance, Triggered Job Wizard and Add New Script. The steps inside the black square are what should be finished in NetMRI. In other words, they are a mechanism in NetMRI and cannot be changed. There are two main panels involved, including Triggered Job Wizard and Add New Script, which are all in Configuration Management function in NetMRI. The panel “Triggered Job Wizard” is to define a job, which consists of trigger source and job script. The trigger source is a mechanism to trigger the script. A script contains Command Line Interface (CLI) using script language Perl, which is defined in Add New Script panel. To define a new job in NetMRI, there are several steps, including: defining new “Triggered Jobs”, choose a trigger source, choose trigger filters, define job and refer to script, and schedule job. The detailed description of what these steps are doing can be found in the document: Infoblox Network Automation Administrator Guide (Infoblox, 2014). As shown in Figure 8 Initialization of “to be” process in NetMRI, defining a script is in a different panel comparing with defining a Job. The black square in the figure shows the different panels.

To sum up for this chapter, this chapter act as a bridge between the “as is” process and the implementation, which is the next chapter: Empirical study.

4 Chapter 4. Empirical study

The goal of this empirical study is to build a test lab based on the former analysis. After having figured out how the steps in Chapter 2 can be carried out through NetMRI in Chapter 3, we here concentrate on what we actually want to achieve by doing that. So here we concentrate on the main goal and subgoals for test lab implementation, following the reasoning of issues comparing with “to be” design. Lots of issues, which are things that need to be changed compared to current provisioning procedures, have been figured out during the implementation of the test lab. These issues will be clarified in chapter 4.2. The next part of this chapter is about the preparations for both NetMRI and Capgemini, reporting on the realization of the test lab. As the main goal of using test lab is to investigate to what extent NetMRI is a suitable working network automation tool, we have chosen to implement part of the former analysis in the test lab and to try to draw conclusions regarding the main goal of this test lab within limited time. The last part of this chapter is the conclusion of the test lab result and future improvements.

4.1 Define main goal and subgoals for the test lab

Based on the “to be” design in chapter 3, which tries to follow the current procedure as closely as possible, this chapter is to define the main goal and subgoals for building the test lab. To be specific, the main goal of the test lab, as part of this thesis study, is to investigate to what extent NetMRI is a suitable working network automation tool. As for the subgoals, they are divided based on the analysis in Chapter 2 and Chapter 3. Due to the limitation of time, the main goal is divided into 6 subgoals, aiming to narrow down the scope and focusing on the smaller scope first. The subgoals are divided based on the sequence analysis in Chapter 2 and Chapter 3. Based on the use case analysis in Chapter 2, the first use case “prepare for PKI management” is combined with use case approval certificate request” because these two use cases need to be finished by the administrator instead of automating in NetMRI. Thus the first use case to be automated in NetMRI is use case “locate the correct spoke router”, which is carried out by pre-install (stepping stone) hub router. To achieve this use case, the first step is adding the pre-install (stepping stone) hub router in NetMRI and the second step is to find the spoke router through pre-install (stepping stone). The reason to divide this use case into two subgoals is that there are a series of steps needed to be finished first before adding a router in NetMRI. Besides, it’s also a good preparation for adding other routers in NetMRI. Based on the analysis in Chapter 3, all remaining use cases are combined as a subgoal: building the connection between the spoke router and the hub router, which is shown in Figure 7 “to be” process overview flowchart in NetMRI. Thus the third subgoal is: Building the DMVPN tunnel between Spoke router and hub router. But there are two key issues with this subgoal: the interaction of input and output and the interaction for the certificate request. As the goal is to investigate to what extent NetMRI is a desirable network automation tool, it’s better to put all the possible subgoals at the beginning. These subgoals are optional and will be investigated if time available. These subgoals include “improve script for checking the desirable output and different options for other service, like 3G and 4G. The reason to include this subgoal is that this thesis only focuses on the basic Ethernet option as mentioned in Chapter 2. If there is time left, it’s good to explore the possibilities for other options like 3G and 4G. Besides, if time available, the possible interaction between NetMRI and Infoblox network service software (DDI) can also be included in the subgoals.

To sum up all the subgoals, there are five sub-goals for the test lab with different priorities using MoSCoW method, which is an acronym represents each of four categories. MoSCoW stands for different prioritization of requirements: Must have, Should have, Could have, and Would like but won’t have (MoSCow Analysis , 2009). The category descriptions are based on book: A Guide to the Business Analysis Body of Knowledge. Requirement labeled with must has the highest priority and it must be satisfied. Should describes a requirement that should be included in the solution if possible.

Could represents a requirement that is desirable but not necessary. Would like, which also called won't, "describes a requirement that stakeholders have agreed will not be implemented in a given release, but may be considered for the future." The reason to use MoSCoW method is that there is not enough time to do everything, but everything needs to be addressed(Kruiswijk, 2013). It's crucial to focus on the key issues first. Defining subgoals will also make it clear what are the most important steps and what should be finished in the first place in NetMRI. But there are lots of issues figured out during the implementation of the test lab. These issues will explain in Chapter 4.2. The result of test lab will also use this main goal and subgoal table.

Main Goal: Investigate to what extent NetMRI is a suitable network automation tool	Priority (MoSCoW)
1. Add stepping stone hub router (also called pre-install hub router) in NetMRI	Must have
2. Find the spoke router through stepping stone hub router by triggering jobs in NetMRI	Must have
3. Build the connection(DMVPN tunnel) between the spoke router and the hub router <ul style="list-style-type: none"> a. Interaction between input and output (optional) b. Interaction between administrator and NetMRI for certificate request 	Should have
4. Improve script for checking desirable output & different options (e.g. 3G/4G)	Could have
5. Interaction between NetMRI & Infoblox DDI (IP address)	Would like

Table 3 Main goal and sub-goals for the test lab

As shown in the table above, subgoals that must be finished are "add stepping stone (pre-install) hub router as a seed router in NetMRI", and "Find the spoke router through the stepping stone (pre-install) hub router by triggering jobs in NetMRI". These two subgoals are the starting point for using NetMRI. The implementation details for the subgoals will be explained in the Chapter 4.4: Test lab realization.

4.2 Issues between "to be" design and implementation

As the "to be" design tries to follow as closely as possible the current provisioning procedures, there are some issues figured out during the realization of the test lab. The reason why "to be" design should follow the current procedures could be found in Chapter 3. This chapter explains about all the issues, which are things needed to be changed compared to current provisioning procedures. The sequence of issues is based on the implementation of the test lab, which is following subgoal priorities mentioned in chapter 4.1. A summary of all the needed issues, which are divided with category solved and category proposed solutions, will be given at the end of this chapter.

Section Structure As is shown from chapter 4.1: the first two subgoals have highest priority. That is to say, the test lab will realize the first two subgoals first. As the test lab implementation starts with these two subgoals, most issues are figured out during the realization of these two subgoals. Below is the list of all the issues, following the detailed description for each issue. The description structure of each issue starts with pointing out why this is an issue and giving the issue category at the end.

The reason to summarize these issues in different categories is to provide different viewpoints for these issues. The issue category includes: issues have to change, issues with proposed solutions, and issues need to be solved. The reason to have this issue category is that the issue list is following a chronological order, a summary is needed to show the result of these issues. The result of some issues has to change means there is a concrete solution for these issues and the solution is valid. The second category is issues with proposed solutions. These issues are provided with proposed solutions, but these solutions have not been tested yet. The last category is issues need to be solved. These are issues without solutions, so they need to be figured out in the future.

Issue list:

- *Issue of building physical connection between the spoke router and NetMRI*
- *Issue of subgoal 1(must have): add the stepping stone (pre-install) hub router in NetMRI*
- *Issue of subgoal 2(must have) change to: Add spoke router directly in NetMRI*
- *Issue of generating message of adding a new spoke router*
- *Issue of how to trigger script containing configurations*
- *Issue of NetMRI credential lists*
- *Issue of sending several commands at one time*
- *Issue of sending the certificate*
- *Issue of string manipulating*
- *Issue of waiting approval from other systems and continue configuration*
- *Issue of troubleshooting*

Issue of building physical connection between the spoke router and NetMRI

Although the spoke router for testing and NetMRI are all in the management network at Capgemini, there is no physical connection between them. NetMRI needs access to the spoke router physically. Currently, the spoke router has a physical connection with pre-install hub router. But there is no physical connection between the routers and NetMRI. For security reason, the VRF (Virtual Routing Forwarding), which contains the pre-install hub router, has only one interface to ensure only one access to the VRF in the current provisioning procedures.

There are two solutions, one is to add another interface in VRF to make NetMRI access to the VRF, another solution is to add the IP address of NetMRI in the routing table of the pre-install hub router, which is in the VRF and connect to the spoke router.

Adding another VRF is not following the original procedures, which is for security reasons. So in this test, we choose the second solution, which is adding the IP address in the routing table, but a lot of firewall changes are involved during the testing.

Issue of subgoal 1(must have): add the stepping stone (pre-install) hub router in NetMRI

The subgoal 1 mentioned in Section 4.1 is: Add stepping stone hub router (also called pre-install hub router) in NetMRI. In the original procedures, the spoke router is configured through stepping stone using Telnet for security reasons. For example, Capgemini doesn't want any management network exposure to the unknown device. Compared to the original procedures, NetMRI has to access spoke router directly in order to configure it because NetMRI is designed to have direct connections with devices. Telnet through the stepping stone (pre-install) hub router to configure the spoke router is not possible in NetMRI, as it will end up in programming another NetMRI. This is based on the discussion with NetMRI engineer. This decision also influence the second subgoal, which is adding the spoke router directly in NetMRI instead of finding spoke router via pre-install hub router.

But it is helpful to add the stepping stone (pre-install) hub router in NetMRI for faster discovering newly added devices instead of the reason in the original procedures, which is to Telnet via stepping stone (pre-install) hub router to the spoke router.

Issue of subgoal 2(must have) change to: Add spoke router directly in NetMRI

Based on the analysis of provisioning procedures "to be" design in Chapter 2.6 and Chapter 3.2.1, the subgoal 2 is supposed to be finished in this way: when a new spoke router is added in Capgemini network, NetMRI discovers the spoke router via stepping stone (pre-install) hub router and runs the script to configure the tunnel between the aimed hub router and the spoke router.

To figure out the solution for subgoal 2: Find the spoke router through stepping stone hub router by triggering jobs in NetMRI, some changes need to be clarified. Firstly, we will not find the spoke router via stepping stone as explained in the last issue. So in the test lab we will add the spoke router directly in NetMRI. But to use NetMRI in the production environment, it's more efficient to let NetMRI find the spoke router automatically.

There are three options for the issue of how NetMRI can add a new spoke router. The first option is waiting for NetMRI discover the newly added device automatically. Another option is adding the static IP address of the newly added spoke router manually in NetMRI. The last option is to use a trigger in NetMRI.

Based on the NetMRI engineer's explanation, the first automation scanning option is supposed to "take up to an hour, but it all depends on the network size and how long till the NetMRI sees the new router via ARP/Router." If a regular automatic scanning is chosen, the provisioning has to wait until NetMRI automatically scans the Capgemini network and discover the newly added router.

The second option needs to be finished by the administrator from Capgemini. The last option is via a trigger when a new router is added. To make NetMRI figure out the newly added spoke router automatically and immediately, a message should be sent to NetMRI, which is to inform NetMRI about the newly added router. The details of how to generate a message for the newly added router will be explained in next issue: *generating message of adding a new spoke router*.

As key issue of automation provisioning is to shorten the provisioning time, which is the clients' requirement, further testing is needed for all these three options to figure out which one fits best.

Issue of generating message of adding a new spoke router

If we choose the third option to trigger NetMRI as mentioned in the last issue, a message, which is a DHCP syslog, needs to be sent to NetMRI when there is a new spoke router added into the network. In the current provisioning procedures, stepping stone (pre-install) hub router can generate a message when a spoke router connects to the stepping stone (pre-install) hub router. This connection is a pre-install VPN tunnel, which is finished with a third party on-site engineer. The stepping stone (pre-install) hub router could send a message to NetMRI with the information of the new-added router. In NetMRI, the discovery of a device can be triggered by sending a syslog message, which is a piece of message to record the action executed by the device. That is to say, the "Discovery" function in NetMRI, which is to scan the whole network, could be triggered by receiving the DHCP syslog containing the IP address of the newly added router.

The following is a simple example of how the message is generated among different network devices. The following is an example aiming to give a general idea how this message is generated. Usually, Infoblox, being the company producing NetMRI, uses the following steps to send the message. There are four devices involved, including a switch, a device needed to be connected, an Infoblox DHCP Appliance, and a system log server. The switch is connected to the Infoblox DHCP Appliance. The

system log server is connected with the Infoblox DHCP Appliance. Once the device is connected to the switch, it requests for an IP address from the DHCP Appliance by broadcasting. After the Infoblox DHCP Appliance has given the IP address to the new device, it generates a message called syslog and sends it to a system log server to record the action.

In NetMRI, this DHCP syslog is usually generated from the new-added device with a pre-defined format:

```
<MessageParsePattern>DHCPACK on [[${ipassigned}] to [[${mac}]] .*via [[${interface}]] relay [[${relay}]]  
lease-duration [[${leaseduration}]]</MessageParsePattern>
```

This is a normal syslog message generated by a DHCP server. When NetMRI receives the DHCP syslog with correct format, the “Discovery” function will be triggered and NetMRI will check the credentials automatically. The engineer from NetMRI proposes a solution. When the new-added spoke router is going to build the pre-install tunnel with pre-install (stepping stone) hub router, the spoke router generates an event instead of a normal syslog message, which could be used to fake a DHCP syslog message to NetMRI.

Issue of how to trigger script containing configurations

As discussed in *Issue of subgoal 2 (must have) change to: Add spoke router directly in NetMRI*, there is a built-in mechanism to trigger the pre-created provisioning procedures. This built-in mechanism called “trigger source” in NetMRI. There are two trigger source in NetMRI. One is issue, the other is policy rule. In this test lab, we use the build-in issue: bare metal device found, because this is the suitable build-in trigger source and the only change is the hostname of the spoke router. Besides, using build-in trigger source will save time comparing with create a issue or a policy rule. But this build-in issue “bare metal device found” is triggered only if the network device has hostname “autoconfig”. That is to say, the spoke router needs to have the hostname “autoconfig”. When NetMRI discover there is a newly added router or other network device, it will check credentials. If the credentials, to be discussed in the next issue of credential lists, are correct and the network device has hostname “autoconfig”, the built-in mechanism called “bare mental device found issue” will be raised. This “bare mental device found issue” is triggered by the device name “autoconfig”. If “bare mental device found issue” is raised, then the pre-created script, which contains the provisioning command, will be triggered and run automatically. The information on how to define a script can be found in the user guide (Infoblox, 2014).

Currently, the spoke router hostname contains information for a specific client. But to speed up testing, we change the hostname into “autoconfig”. There are possibilities to define issue instead of

using the build-in issue “Bare Metal Device Found”. More investigations are needed to solve this issue. The proposed solution is to define a policy rule in NetMRI that checks for another hostname instead of “autoconfig”.

Issue of credential lists

To add a router successfully in NetMRI, there are several credentials need to be added in the credential lists, which is part of the setup in NetMRI. NetMRI will check every credential to these lists if it discovers a new router or other network devices. These credential lists include SNMP, the router login password, the router enable-model password, router configure-model password. The problem is: if every device has its own credentials, the lists will be very long. If many devices use the same credentials, it's not secure.

There are two proposed solutions. Firstly, a local account for NetMRI can be created for the newly added router with credentials known by NetMRI. After the configuration, the local account is removed. The second solution is to use the current temporary local account, which are used by the third party.

Issue of sending several commands at one time

This issue was figured out during testing a group of commands. There are two situations. The following is an example of the first situation. If we send some commands separately, NetMRI cannot send these commands to a router successfully. For example, it is possible to deal with situations where the device requests some feedback. Some commands request responds immediately, if we choose to send these commands via NetMRI in separate built-in command, which is `send_command()`, the router cannot understand it and the execution result is not correct, comparing with the situation we send the command directly to a router. Do not put the answer in a different `send_command` statement because NetMRI will wait for a prompt after a command. The way to solve this is to send the answer together with the command. For example:

```
send_command ("restart\r\n");
```

instead of:

```
send-command ("restart");
```

```
send_command("yes");
```

If we choose to send the command “restart” and “yes” separately in NetMRI, the router cannot understand it. In a word, for the situation required an answer or a feedback command immediately, these commands have to be sent at one time.

The following is the second situation. When sending a configure mode command, there are usually several commands need to be sent continuously without immediately respond from a router. It's better to send these commands at once via NetMRI to a router instead of sending these commands separately. This could be solved by using “\r”. Below a specific example, which uses NetMRI built-in method send_command to send a series command at one time to a router:

```
send_command ("configure terminal\r enrollment terminal pem");
```

Instead of:

```
send_command("configure terminal");
```

```
send_command("enrollment terminal pem");
```

As this is the kind of mechanism adopted for NetMRI to send command, this issue could be solved if follows the solution mentioned above. This issue is in the category need to be changed.

Issue of sending certificates

In the chosen scope of current procedures, there is one command of sending a PKI certificate. But this certificate has a pre-defined format, which cannot use NetMRI built-in command, with multiple lines. Some changes are needed to send the certificate successfully via NetMRI to a router, for instance, below is a simple example of the pre-defined format of the certificate.

-----BEGIN CERTIFICATE-----

CONTENT OF FIRST LINE OF THE CERTIFICATE

CONTENT OF SECOND LINE OF THE CERTIFICATE

.....

-----END CERTIFICATE-----

NetMRI built-in method “send_command” should be able to send this certificate. But further tests of how to use regular expression to send the certificate successfully are needed. Nevertheless, we propose the following preliminary solution:

1. Use the following regular expression to solve the start line and the end line, which can be found in detailed procedural steps (Capgemini, 2015):
-----END CERTIFICATE-----\ny\n
2. For the content of each line in the certificate, using \r to ensure NetMRI knows these lines are together. (See also our solution for the previous issue)

Issue of string manipulating

In current procedures, when a command using Command Line Interface (CLI) is sent to a router, engineers check the output of the CLI manually. When using scripting to send a command automatically, the output of command needs to be checked. This could be achieved by using string manipulating. There are many different options for string manipulating. This test lab uses a regular expression in Perl. The regular expression match, which is “=~”, has been tested. This is only a rough match. Further implementation could be improved by creating more accurate regular expression matching or by other string manipulating methods. This is a programming issue.

Issue of waiting approval from other systems and continues configuration

Current procedures The current procedures of sending certificate are as follows. There are 3 actors involved in total: administrator, spoke router, and other systems, which are Remote Desktop Protocol-client (RDP-client) and Microsoft Management Console (MMC). When the administrator does a login on the spoke router, he sends the certificate request command to the spoke router. Then the spoke router sends the certificate request to another system automatically. In such other systems, being the RDP-client and MMC, the administrator has to approve the certificate request. Thereafter the administrator needs to check whether the approved certificate is received in the spoke router. Together this means, spoke router first needs to wait until it is informed the certificate has been received and approved. The approval is finished in the RDP-client and MMC. To check the certificate has been received, the administrator needs to send check status command to the spoke router.

Procedures in NetMRI The new procedures in NetMRI involves four actors in total: administrator, spoke router, other systems, and NetMRI. The certificate request is sent to the spoke router first via NetMRI instead of sending by administrator in current procedures. If NetMRI sends this request successfully to the spoke router, this request is sent to other systems from the spoke router automatically. This certificate request should be approved by an administrator manually for security reasons. When the request is approved, the spoke router needs to know it by checking the certificate status and continue the rest configuration.

Proposed solution There are two possible solutions. The first one is to create two separate scripts and run the script manually in NetMRI. The other solution is to create three separate scripts and run automatically.

The two separate scripts mentioned in the first solution is divided after the certificate request. The first script contains commands before the certificate request, and the second script contains the rest procedure commands.


The second solution is based on suggestions from NetMRI engineers. Three scripts can be used to solve this issue, including: current script1, script2, and triggered script3. Current script1 stops when manual intervention is required. Another script, scrip2, containing the rest of the configuration, is automatically triggered when the certificate status is okay. That is to say, when the spoke router receives the approval of the certificate, which is finished manually in other systems. Script2 is part of a triggered job, which is a terminology in NetMRI, and is triggered by the “custom issue”.

To automatically trigger script2, a trigger script3 is needed. This trigger script3, which is in a scheduled job and runs several times a day to check the certificate approval status by sending a command to the spoke router. This command is defined in this trigger script3. A scheduled job is a mechanism in NetMRI to define when this script will be executed. If this trigger script3 is not a scheduled job and doesn’t run several times a day, the duration to ensure the spoke router does receive the approval of certificate might be very long. Based on the output of the status checking command, a “custom issue” could be set.

Custom issue The “custom issue” could be defined as the Figure 9 Define "custom issue" shown below. The mechanism of “custom issue” is similar to the issue “Bare Metal Device Found”. The difference is that “Bare Metal Device Found” is a built-in issue and “custom issue” is defined by a user.

Add Command Script Issue

This form is used to define custom issues for use in scripts.

Issue ID: 

Name:

Description:

Component:

Penalty: ☐ Correctness ☐ Stability

Detail Columns:

Figure 9 Define "custom issue"

Issue of trouble shooting

Using NetMRI will make it difficult to do troubleshoot during the transition from current procedures to using NetMRI. For example, the interaction between NetMRI and the controlled device is via predefined script and job. If something unexpected happens, this might be a problem with the device, with the network or with the communication between the device and NetMRI. Besides, due to different functional panels in NetMRI, testing the results of scripts and jobs is not efficient. For instance, to test the script, the script is triggered manual in one panel in NetMRI. But the test result is in another panel. Moreover, to see the command result and detailed information, three or four panels are involved in total. A better solution is to use Perl development environment for checking the correctness of the script.

Summary of all the issues

Issues have to change in the test lab (new status: solved)

- *Issue of building physical connection between the spoke router and NetMRI*

- *Issue of subgoal 1(must have): add the stepping stone (pre-install) hub router in NetMRI*
- *Issue of subgoal 2(must have) change to: Add spoke router directly in NetMRI*
- *Issue of sending several commands at one time*

Issues with proposed solutions

- *Issue of how to add a new router in NetMRI*
 - o *regular automatic scanning in NetMRI*
 - o *Issue of generating message of adding a new spoke router*
- *Issue of credential lists*
- *Issue of how to trigger script containing configurations*
- *Issue of sending certificates*
- *Issue of string manipulating*
- *Issue of waiting approval from other systems and continue configuration*

4.3 Set-up of the test lab

This section is about preparations for setting up the test lab, including scope and assumptions for the test lab, preparation in NetMRI, preparation of hardware and network in Capgemini.

Scope and assumptions for the test lab

The goal of the test lab is to investigate to what extent NetMRI is a suitable network automation tool for DMVPN service provisioning. Given the limitation of time, to figure out the goal of the test lab as soon as possible, this part narrows down the scope by listing all the assumptions for the test lab, following the reasoning for the chosen assumptions.

The assumptions include input parameters and add another VRF. The detailed information is listed below.

- Inputs parameters for testing in NetMRI:
 - o Hostname of Stepping stone (pre-install) hub router
 - o IP address of Stepping stone (pre-install) hub router
 - o Spoke router hostname: "autoconfig"
 - o Spoke router IP
 - o Only focus on the scenario of "Ethernet"
 - o Spoke router type c880 and software version
 - o Certificate server hostname and IP address
- Add another VRF interface for the test lab

The details for each assumption are explained as follows. The detailed information can be found in document: Capgemini LAB DMVPN Spoke Installation Procedure. In the original procedures, the

hostname and IP address for the stepping stone hub router (pre-install) are fixed. Besides, on-site engineers from the third party will configure the hostname based on the service type and the customer. But in this test lab, we use the hostname “autoconfig” because the built-in issue in NetMRI called “Bare Mental Device Found” assumes each new device hostname is “autoconfig”. The detailed reasoning could be found in Section 4.2 Issues between “to be” design and implementation. To speed up the test lab and figure out whether NetMRI is a suitable working tool for provisioning automation, the test lab will use the hostname “autoconfig” for the spoke router.

In the real situation, there are different service options, for example: Ethernet only, 3G, and 4G. Clients could also choose different types of routers. In this test lab, we use only one type of service with Ethernet only, as this option is the basic one and the other options are all based on this one. If the Ethernet-only option is working successfully in the test lab, it is highly probable that other options could also be automated in NetMRI as well. If not, it seems useless to test other options.

Furthermore, clients could also choose different types of spoke routers based on the internet options, which are Ethernet only, 3G, and 4G. In this test lab, we use spoke router type c880 and software version 15.2(4)M2. Because this is the type of the spoke router in the old test lab environment. Using this spoke router can save time. Moreover, if this type of router is working, other types of routers can be used by changing script parameters to the correct ones.

For security reasons, the VRF (Virtual Routing Forwarding) only has one interface to ensure precisely one access to the VRF. But in this test lab, there are two interfaces in VRF to make the NetMRI access the VRF. The detailed reasoning has been given in Section 4.2.

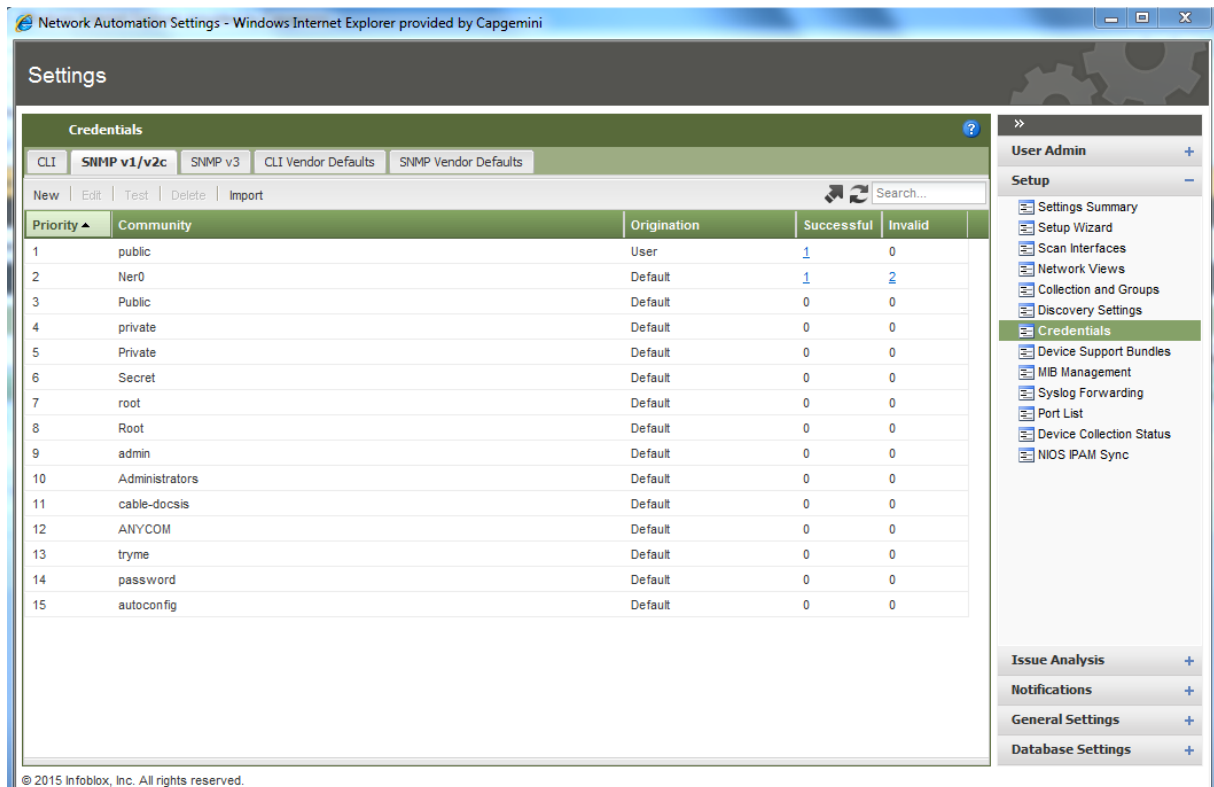
Preparation in NetMRI

The starting point for the test lab is after initial preparations in NetMRI, which means basic settings are finished and operational according to the NetMRI user guide. To be specific, these chapters include “set up” and “Discovery with a New Network Automation Deployment” in the document “Infoblox Network Automation Administrator Guide, Release 6.9”(Infoblox, 2014). NetMRI will run “Discovery” function automatically and so will scan the network. When the new device is accessible by NetMRI physically, NetMRI will check 6 statuses for the new device, including Exists, Reached, SNMP, CLI and Groups. The detailed explanations for these statuses are in the document “Infoblox Network Automation Administrator Guide, Release 6.9” Chapter “Running the Setup Wizard”. The aim of such statuses checking is to ensure NetMRI can access to the device and get the basic information of the device using credentials.

Preparation of hardware and network in Capgemini

To ensure the device is physically reached via NetMRI, which is the second status mentioned above, firewall changes are needed. The explanations of firewall changes are out of the scope of the test lab.

Enable credentials To make the SNMP status enabled, several changes are needed. First of all, the SNMP community string should be added and enabled in the router. Secondly, the SNMP community string should be included the SNMP credential list as is shown by the Figure 10 below. NetMRI will check each of the community strings in the list with the new device. Below is an example of adding the credentials in a router and NetMRI. We use the example of adding the SNMP community string in the stepping stone (pre-install) hub router. The Figure 10 SNMP community string list shows the SNMP credentials list in NetMRI.



The screenshot shows the 'Settings' window in a web browser, specifically the 'Credentials' tab for 'SNMP v1/v2c'. The interface includes a table with columns for Priority, Community, Origination, Successful, and Invalid. The table lists 15 community strings. The first entry, 'public' with priority 1, is highlighted. The right sidebar shows a navigation menu with options like 'User Admin', 'Setup', 'Issue Analysis', 'Notifications', 'General Settings', and 'Database Settings'.

Priority	Community	Origination	Successful	Invalid
1	public	User	1	0
2	Ner0	Default	1	2
3	Public	Default	0	0
4	private	Default	0	0
5	Private	Default	0	0
6	Secret	Default	0	0
7	root	Default	0	0
8	Root	Default	0	0
9	admin	Default	0	0
10	Administrators	Default	0	0
11	cable-docsis	Default	0	0
12	ANYCOM	Default	0	0
13	tryme	Default	0	0
14	password	Default	0	0
15	autoconfig	Default	0	0

Figure 10 SNMP community string list in NetMRI

In this test lab, we use the SNMP community string “public”, which is shown as the Figure 11 below by Telnet to the router.


```

Chassis: FC2170660H2
99986 SNMP packets input
  0 Bad SNMP version errors
  8 Unknown community name
  0 Illegal operation for community name supplied
  0 Encoding errors
  507814 Number of requested variables
  0 Number of altered variables
  31967 Get-request PDUs
  67716 Get-next PDUs
  0 Set-request PDUs
  0 Input queue packet drops (Maximum queue size 1000)
99978 SNMP packets output
  117 Too big errors (Maximum packet size 1500)
  1256 No such name errors
  0 Bad values errors
  0 General errors
  99978 Response PDUs
  0 Trap PDUs
SNMP Dispatcher:
  queue 0/75 (current/max), 0 dropped
SNMP Engine:
  queue 0/1000 (current/max), 0 dropped
SNMP logging: disabled

```

Figure 11 SNMP enabled in stepping stone (pre-install) hub router

```

Community name: public
Community Index: cisco3
Community SecurityName: public
storage-type: nonvolatile      active

```

Figure 12 SNMP community string in stepping stone (pre-install) hub router

4.4 Test lab realization

This part explains the details of what are realized in NetMRI for the test lab. The realization of the test lab is based on the table of subgoals listed at the beginning of this chapter.

Main Goal: Investigate to what extent NetMRI is a desirable network automation tool	Priority (MoSCoW)
1. Add stepping stone hub router (also called pre-install hub router) in NetMRI	Must have
2. Find the spoke router through stepping stone hub router by triggering jobs in NetMRI	Must have
3. Build the connection(DMVPN tunnel) between the spoke router and the hub router <ul style="list-style-type: none"> a. Interaction between input and output (optional) b. Interaction between administrator and NetMRI for certificate request 	Should have
4. Improve script for checking desirable output & different options (e.g. 3G/4G)	Could have
5. Interaction between NetMRI & Infoblox DDI (IP address)	Would like

Table 4 Main goal and subgoals

Subgoal 1 (must have): Add stepping stone (pre-install) as a seed router in NetMRI.

In the production environment, the newly added spoke router should be discovered by NetMRI automatically, the proposed solution can be found in chapter 4.2 Issues between “to be” design and implementation, Issue of how to add a new router in NetMRI.

To clarify how to add a router in NetMRI, this part uses stepping stone router as an example. It is the same procedures for any other router. To ensure NetMRI could login to the router automatically, NetMRI needs to know a set of credentials, including SNMP credentials and CLI credentials as Figure 13 shows. The detailed procedures are in the document “Infoblox Network Automation Administrator Guide, Release 6.9” chapter “Running the Setup Wizard” and “Data Collection Techniques” (Infoblox, 2014).

The Figure 14 CLI credential list in NetMRI below shows the list of CLI credentials in NetMRI. When there is a newly added device sending a syslog to NetMRI, NetMRI will test automatically with each of these credentials. The SNMP credential is explained in chapter 4.3 Set-up of the test lab. CLI credentials are used when NetMRI needs to access the device. For example, these are username and password when logging into the a router via Telnet, which is as shown in the Figure 13 below.

After these changes are implemented successfully, the statuses are all available as Figure 15 NetMRI accessible for stepping stone (pre-install) hub router shows below.

Type: Router (20%) Vendor: Model: SNMP Status: Enabled (Pending)
 O/S Version: Up Time: Last Communication: MAC Address:

Management Status

Current status
 First Seen: 2015-05-12 09:22:18 Last Seen: 2015-05-13 09:17:55
 Last Action: Device Groups: Successfully assigned to device groups at 2015-05-13 09:18:03
 License: Unlicensed

Exists Device exists / Source: Seed
 Last Timestamp: 2015-05-12 09:22:18

Port Scanned

Reached Failed to reach
 Last Timestamp: 2015-05-12 12:27:05

SNMP Failed to authenticate
 Last Timestamp: 2015-05-12 12:27:05

SNMP Collection

CLI Not Applicable

CLI Collection Not Licensed

Groups Successfully assigned to device groups
 Last Timestamp: 2015-05-13 09:18:03

Discover Next Discover Now License Unmanage Delete Export

© 2015 Infoblox, Inc. All rights reserved.

Figure 13 Credential status to access a device

Network Automation Settings - Windows Internet Explorer provided by Capgemini

Settings

Credentials

CLI SNMP v1/v2c SNMP v3 CLI Vendor Defaults SNMP Vendor Defaults

New Edit Test Delete Import Hide Passwords Search...

Priority	Protocol	Origination	Username	Password	Vendor	Successful	Invalid
1	ANY	USER	c	c	ANY	0	0
2	ANY	USER	ENABLE	c	ANY	1	0
3	ANY	NETC	netmri-os	aKLy63w	ANY	0	0
4	ANY	NETC	autoconfig	AutoConfig	ANY	0	0

© 2015 Infoblox, Inc. All rights reserved.

Figure 14 CLI credential list in NetMRI

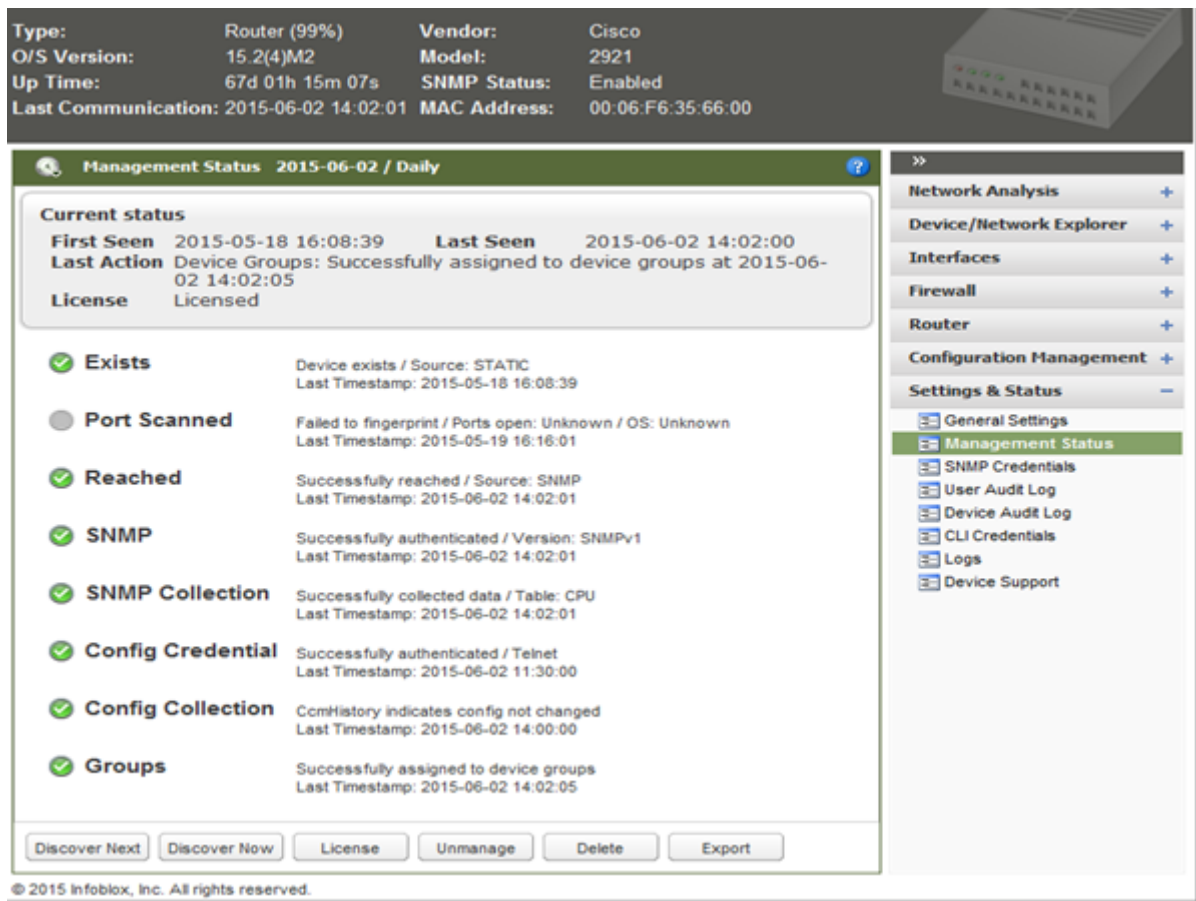


Figure 15 NetMRI accessible for stepping stone (pre-install) hub router

Subgoal 2: Find the spoke router through stepping stone hub router by triggering jobs in NetMRI.

During searching solutions for the second subgoal, the engineer from NetMRI suggests it's unnecessary to find the spoke router through the stepping stone (pre-install) hub router inside NetMRI. According to "to be" design in Chapter3, the first step is to add stepping stone in NetMRI and the second step is to find the spoke through the stepping stone (pre-install) hub router inside NetMRI. The second step will end up with programming NetMRI all over again. So it is really difficult to find the spoke router via stepping stone (pre-install) hub router. An effective solution is to use DHCP syslog as mentioned in Chapter 4.2 Issues between "to be" design and implementation. In a word, there is no need to implement this subgoal.

Subgoal 3 (should have): Build the connection (DMVPN tunnel) between spoke router and hub router using certificates

The procedures to build the connection between spoke router and hub router using the script are as follows. Figure 16 below shows the detailed steps need to be finished in order to build the connection. This flow chart is an extension of Figure 7 "to be" process overview flowchart in NetMRI. In Figure 7, step 5 is: Trigger Job to configure new device. In Figure 17, the detailed steps are shown.

To trigger a job, NetMRI needs to define a trigger source that used to specify the event triggers the execution of the script containing configuration commands. There are two trigger source in NetMRI. One is issue, the other is policy rule. In this test lab, we use the build-in issue: Bare metal device found because this is the suitable build-in trigger source and the only change is the hostname of the spoke router. Besides, using build-in trigger source will save time comparing with create a issue or a policy rule. This issue will be triggered successfully by checking the hostname, which should be “autoconfig”. As mentioned in Chapter 4.2 *Issue of changing router hostname into “autoconfig”*, the hostname should be “autoconfig” in order to trigger the job, which is built-in mechanism in NetMRI. If successfully triggered, the next step in Figure 7 is: Build the tunnel automatically using scripts. In Figure 17, detailed steps are as follows. When the job is triggered as shown in Figure 17 step6, there are two options for the time period when the script must be executed, including: run a script immediately and schedule a time that the script will be executed later.

We should choose to run a script immediately, which is option ‘trigger the job immediately’ in NetMRI, to save time. If needed, it also possible to run a script at a scheduled time. The next step is finished in NetMRI automatically by sending the certificate request via script, which contains configuration commands.

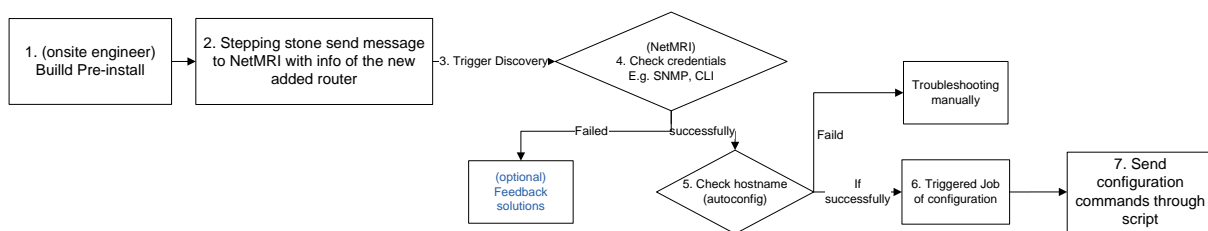
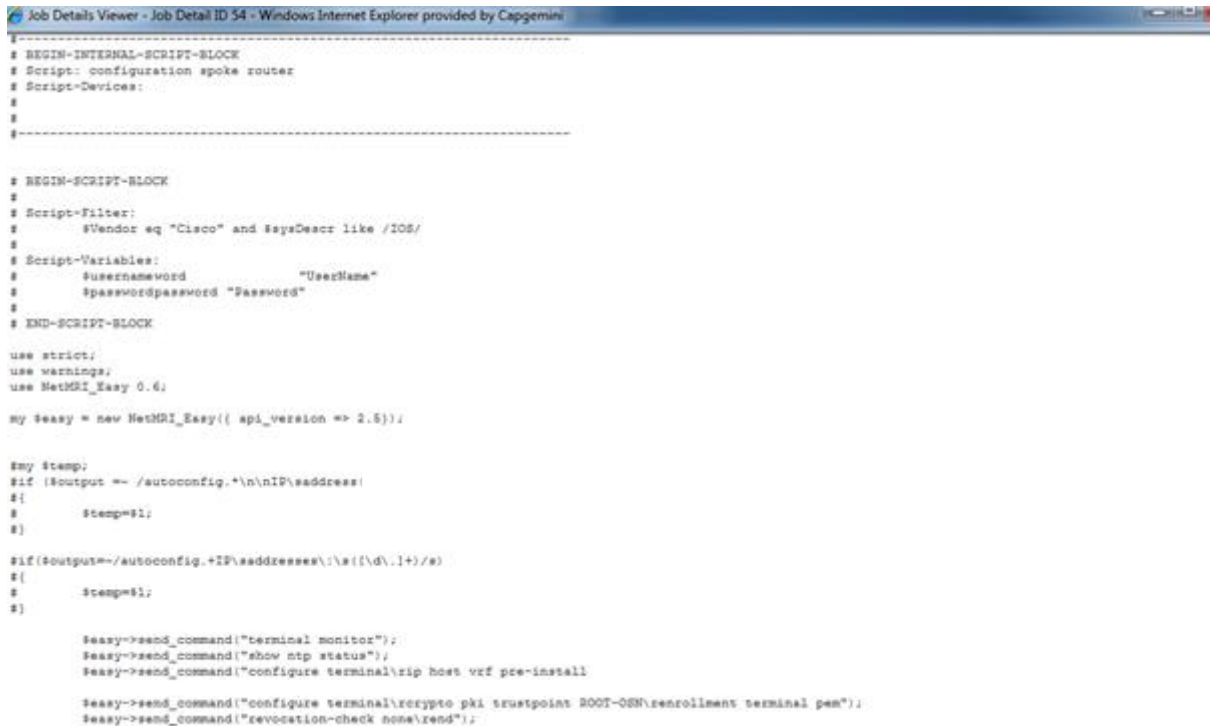


Figure 16 Detailed flowchart for subgoal 3

Sending command To realize this subgoal, which is step 7 in Figure 17, it is sending the Cisco Command Line Interface (CLI) through NetMRI script using Perl language. In NetMRI, there are some built-in methods, which make it easy to send CLI. The detailed information is in the document “Infoblox Network Automation Administrator Guide, Release 6.9” Chapter “Job Scripting”, session “Anatomy of a Perl Script: About NetMRI_Easy.pm”. The build in method used for sending command is: `$easy->send_command("command that needs to be sent");`. The output of the command could be saved if it is assigned to a Perl variable, e.g.: `$output = $easy->send_command ("command that needs to be sent");`. In this Perl sentence, the output of the command is stored in the variable output. In the original procedures, some CLI commands need to check the output of the former step. In Perl, Regular expression is used to match the string. In this test lab, we use the regular expression match,

which is “=~”. This is only a rough match. Further implementation could be improved by creating more accurate regular expression matching or by other string manipulating methods. The Figure 17 below shows the Perl script created in NetMRI and Figure 18 shows the result of this script. The heading of this script is related to NetMRI mechanism. For detailed information, please see the NetMRI user guide (Infoblox, 2014).



```

# BEGIN-INTERVAL-SCRIPT-BLOCK
# Script: configuration spoke router
# Script-Devices: ...
#
#
#
#
# BEGIN-SCRIPT-BLOCK
#
# Script-Filter:
#   $Vendor eq "Cisco" and $sysDescr like /IOS/
#
# Script-Variables:
#   $username $UserName
#   $password $Password
#
# END-SCRIPT-BLOCK

use strict;
use warnings;
use NetMRI_Easy 0.6;

my $easy = new NetMRI_Easy({ api_version => 2.5 });

my $temp;
if ($output =~ /autoconfig.*\n\nIP\address/)
{
    $temp=$1;
}

if ($output =~ /autoconfig.*IP\addresses\:\s({\d\.1+})/)
{
    $temp=$1;
}

$easy->send_command("terminal monitor");
$easy->send_command("show ntp status");
$easy->send_command("configure terminal\nrip boot vrf pre-install");

$easy->send_command("configure terminal\ncrypto pki trustpoint ROOT-08N\nenrollment terminal pem");
$easy->send_command("revocation-check none\nend");

```

Figure 17 Perl script in NetMRI

As the Figure 18 below shows, NetMRI could send Command Line Interface (CLI) successfully to the spoke router.

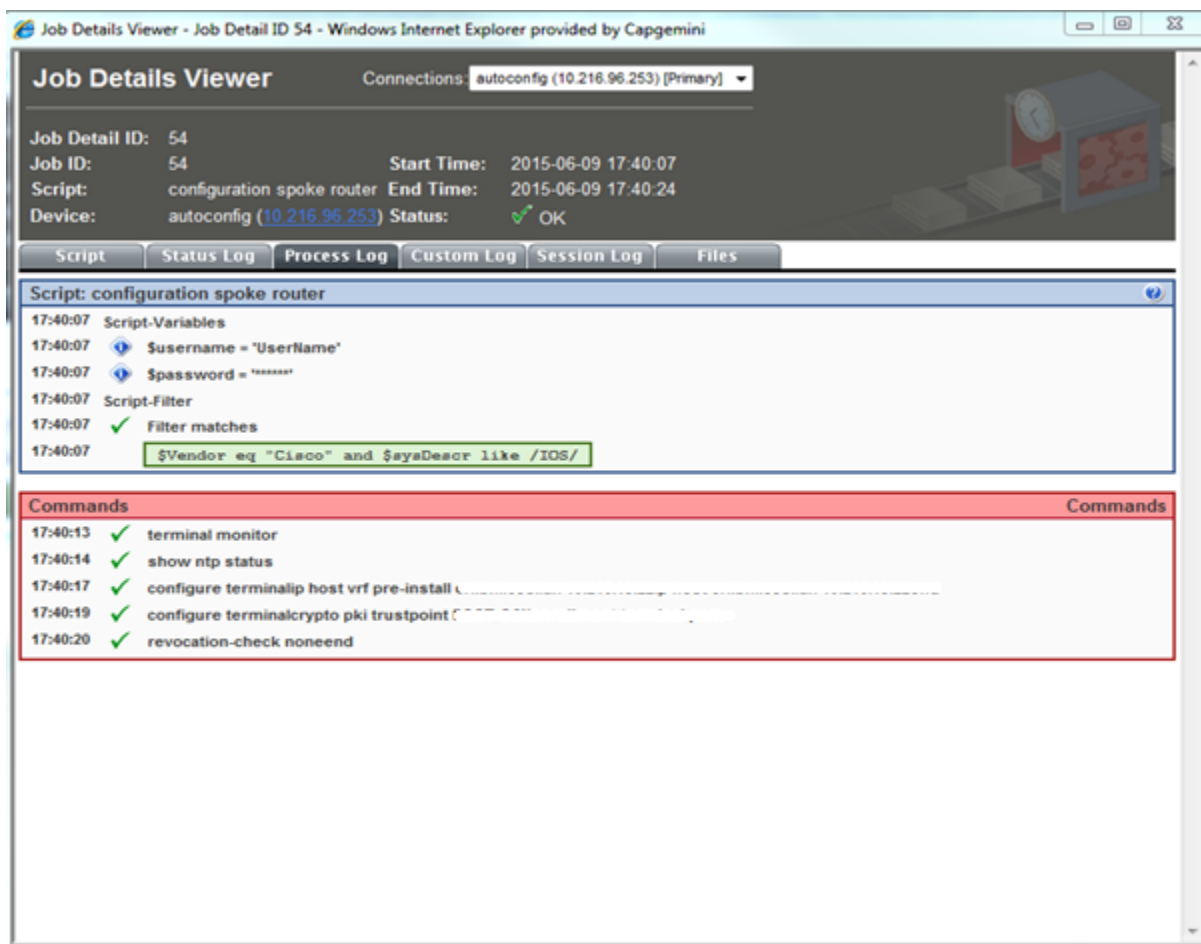


Figure 18 Script Result

NetMRI can also record the result of the Command Line Interface (CLI) commands, see Figure 19 below. But the result of CLI can be checked only after the script has been executed.

```
*****
This System is for a NETMRI POC. Please don't use it.
*****

User Access Verification

Password:
autoconfig>
autoconfig#enable
Password:
autoconfig#
autoconfig#terminal no monitor
autoconfig#terminal length 0
autoconfig#terminal no editing
autoconfig#terminal monitor
autoconfig#show ntp status
Clock is synchronized, stratum 6, reference is [REDACTED]
nominal freq is 280.0000 Hz, actual freq is 249.9909 Hz, precision is 2**20
ntp uptime is 123850500 (1/100 of seconds), resolution is 4016
reference time is D9215423.1F3B9412 (14:01:39.122 CEST Tue Jun 9 2015)
clock offset is -8.8041 msec, root delay is 68.43 msec
root dispersion is 375.76 msec, peer dispersion is 0.10 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000036313 s/s
system poll interval is 1024, last update was 13115 sec ago.
autoconfig#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Figure 19 CLI results after script executed

The interaction between input and output have been mentioned in 4.2 Issues between “to be” design and implementation, Issue of sending continuous CLI and Issue of string manipulating.

4.5 Conclusion on the test lab results and future improvements

To sum up for this chapter, the two tables below shows the result of the test lab. The first table is the same as Table 4, which is the original main goal and subgoals. Table 5 shows the changing of the subgoals and the results. Although the subgoal is not finished yet, but the issues in this subgoal are discussed in 4.2 Issues between “to be” design and implementation.

Main Goal: Investigate to what extent NetMRI is a suitable network automation tool	Priority (MoSCoW)
1. Add stepping stone hub router (also called pre-install hub router) in NetMRI	Must have
2. Find the spoke router through stepping stone hub router by triggering jobs in NetMRI	Must have
3. Build the connection(DMVPN tunnel) between the spoke router and the hub router <ul style="list-style-type: none"> a. Interaction between input and output (optional) b. Interaction between administrator and NetMRI for certificate request 	Should have
4. Improve script for checking desirable output & different options (e.g. 3G/4G)	Could have
5. Interaction between NetMRI & Infoblox DDI (IP address)	Would like

Main Goal: Investigate to what extent NetMRI is a suitable network automation tool	Results
1. Add stepping stone hub router (pre-install) router in NetMRI	solved
2. Change to: Find the spoke router through NetMRI	solved
3. Build the connection (DMVPN tunnel) between the spoke router and the hub router using certificates <ul style="list-style-type: none"> a. Interaction between input and output (optional) 	partially
4. Improve script for checking desirable output & different options (e.g. 3G/4G)	NA
5. Interaction between NetMRI & Infoblox DDI (IP address)	NA

Table 5 Result of the subgoals

Test lab future improvements

Exceptions This test lab focuses on the situation without exceptions. Exception means if the configuration fails and needs some further changes. The reason to focus on the successful scenario first is aiming to figure out whether NetMRI is a suitable working tool as soon as possible. That is to say, the situations of failed procedures are not included in the test lab. But considering exceptions are crucial if Capgemini is going to use NetMRI in the production environment. For example, if a

checking command doesn't get the expected result, how can NetMRI inform the administrator and what should be done next.

As shown in the above chapters, there are lots of assumptions for building the test lab. Besides, the subgoal list is not finished. Below is the summary list of future improvements in the test lab, which are also needed as preparation for the production environment.

- Situations with exceptions
- Interaction between NetMRI & Infoblox DDI for IP address
- Improve script for checking desirable output & different options (e.g. 3G/4G)
- Issue of how to add a new router in NetMRI
- Issue of credential lists
- Issues Issue of how to trigger script containing configurations
- Issue of changing router hostname into "autoconfig"
- Issue of sending certificates
- Issue of string manipulating
- Issue of waiting for approval from other systems and continue configuration

5 Chapter 5. Software-Defined Network (SDN) roadmap

The introduction of this chapter describes the needs to have a roadmap towards software-defined network based on the network automation situation in Capgemini Netherlands. To this aim the chapter has the following structure. In addition, it indicates how such a roadmap can be generated. But for a complete and detailed roadmap more research still has to be carried out and more decisions have to be taken.

Structure To create a roadmap for software-defined network, some terminology needs to be clarified: (1) What is software-defined network, (2) What is a roadmap. This chapter first clarifies these two concepts. To be specific, Section 5.1 addresses what is software-defined network at a high level. In Section 5.2, it is clarified what is a roadmap and why having a roadmap make sense. What elements should be included in the roadmap, is also explained in Section 5.2. Section 5.3 addresses the creation of the roadmap. The last part discusses the conclusion of this chapter and the answer for the question when is the appropriate time to adopt software-defined network.

5.1 SDN Brief review

Based on the SDN definition from Open Network Foundation mentioned in Section 1.1:

“Software-Defined Network is the physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.” (Open Networking Foundation, 2015)

Gartner lists software-defined architecture as top 10 strategic technology trends for 2015 because it will change the current static system into fully configurable and dynamic computing devices (Gartner, 2015). There are many vendors for traditional network services offering solutions for SDN, for instance, HP, Juniper Networks, Nuage networks and VMware.

The concept of software-defined network makes it possible to integrate different perspectives. Software-defined network could provide programming capability to currently network and use Application Programming Interface (API), which has begun another renaissance in the digital age (Gartner, 2014), to interact with network devices (Muhammad H.Raza, 2014).

Furthermore, some vendors for IT automation are involved in software-defined network too. For example, the leader in IT automation Puppet Labs, with industry leading software and hardware vendors, builds joint software-defined infrastructure solutions (Puppet Labs, 2015).

But there is no standard for this new technology yet. Although software-defined network could bring many benefits from business, IT, and end user perspectives, there is no successful implementation use case available of using software-defined network to solve provisioning automation yet. As mentioned in Google’s Globally-Deployed Software Defined WAN, “the future bottlenecks are in bridging protocol packets from the control plane to the data plane and overheads in hardware programming are important areas.” Currently, there are three implementation strategies of SDN: proprietary, open source and hybrid approach.

Apart from network vendors, there are also some competitors in the field of consultant industry moving towards software-defined network. These competitors are just a challenge for Capgemini. For example, Accenture and IBM. Accenture discussed how SDN allows network into a truly dynamic and flexible asset in Technology Vision 2013: Software-Defined Networking (Accenture, 2013). Two years later, Accenture created a two-year roadmap for SDN (Accenture, 2015). IBM published “Implementing IBM Software Defined Network for Virtual Environments” in 2014, which shows the integration between IBM SDN for Virtual Environments within a new or an existing data center (IBM, 2014).

5.2 What is the roadmap

To create a roadmap, the first question is what is a roadmap and how should a roadmap look like. This chapter clarifies what is a roadmap and discuss what kind of roadmap should be developed in this thesis.

Capgemini Generic services roadmap

In Capgemini infrastructure outsourcing generic service group, one kind of roadmap is commonly used. The roadmap has two dimensions, one dimension is time, the other dimension is what goals that generic team wants to achieve in a pre-defined time frame. These goals are usually achieved in the form of projects. That is to say, there are few relationships between different goals and projects, but most goals are well-separated and independent.

But for a roadmap of software-defined network, the relationship between some subgoals might reveal more dependency. Moreover, when relationships between different component and the roadmap formal description are lacking altogether, planning the roadmap to be used in the generic service team, is difficult: e.g. concerning when is the right time to adopt software-defined network. A roadmap for SDN is appropriate only, if it facilitates when is the right time to adopt SDN because this is what the management of Capgemini wants.

Roadmap definition

There are many different roadmaps focusing on different fields with different purposes. In the paper Technology Roadmapping: The Integration of Strategic and Technology Planning for Competitiveness, it says, any technology roadmap aims to create a technology planning when technology investment decisions are not clear. A technology roadmap helps in generating a plan comprising general activities (Olin H. Bray, 1997). As a goal of using the roadmap is to answer the question when is the appropriate time to adopt the new technology SDN, a technology roadmap is a desirable roadmap.

In the paper Technology roadmapping-A planning framework for evolution and revolution, it proposed a T-Plan fast start approach, which has been developed as part of a three-year applied research program. As this multilayer roadmap is the most common form, and the most flexible in application (Robert Phaal, 2004), this thesis will use this technology roadmap for software-defined network roadmap and explore whether this roadmap is a suitable one for routinely reviewed and updated (Olin H. Bray, 1997). This T-plan fast start approach consists of the following dimensions: time, layers, annotation, and process. Based on the paper, the general explanation of these dimensions will be shown .

5.3 Creation of the roadmap

This chapter introduces the detailed elements and the content of different layers in the T-plan fast start approach. The elements focus on the components of the roadmap and the content of different layers including: layer why, what, and how.

To ensure the context of the roadmap is specified for SDN, this part will explain the chosen elements in the T-plan fast start approach and the creation of the roadmap. Figure 20 shows the chosen elements, including time, layers, and annotation, based on generalized technology roadmap architecture in the paper (Robert Phaal, 2004). Each element has to be specifically adapted for software-defined network (SDN). The detailed underpinning of each element is as follows.

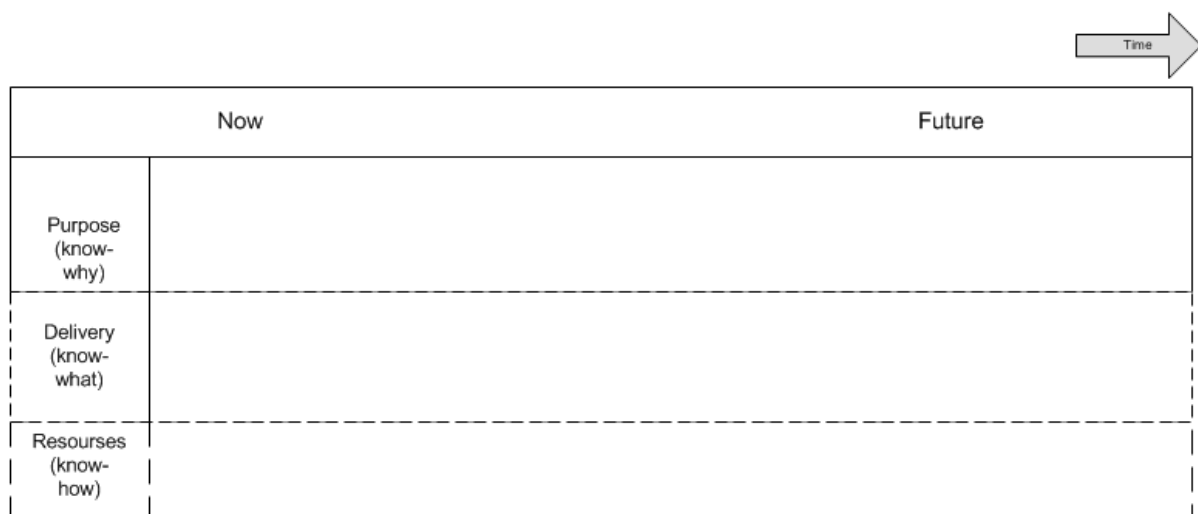


Figure 20 T-plan fast approach template

Time

Based on the paper (Robert Phaal, 2004), time dimensions could be time horizon, scale, and intervals. Usually, for infrastructure sectors, the time horizon is much longer comparing with some other sectors, including software and e-commerce. In the generalized SDN technology roadmap, the time dimension use intervals, including “Now” and “Future”. “Now” refers to the current situation and problems. “Future” refers to the results or goals one wants to achieve. The distance between “Now” and “Future” in the roadmap stands for the gap between the current situation and the future goals.

According to Gartner’s report, the new technology software-defined data center (SDDC) is not mature and will change a lot during the next five years (Gartner, 10 Sep 2014). By 2020, 75% of Global 2000 companies will consider SDDC (Gartner, 10 April 2015). As SDN is one of the components

in SDDC, we assume the time of the SDN roadmap is five year, from 2015 to 2020 based on the predictions from Gartner.

Additional information as contained within the layers serves as annotation.

Layers

Layers form the vertical axis, which need fit for particular problem in an organization generally. There are three different layers: top layer, middle layer, and bottom layer. The top layer relates to the purpose, which is about knowing “why”. “The bottom layer relates to resources (particularly technology knowledge) that will be deployed to address the demand from the top layer of the roadmap (know-how); the middle layer provides a bridge or delivery mechanism between the purpose and resources (know-what)” (Robert Phaal, 2004). This thesis will first focus on the top layer (why) and explain in general what should be included in the middle layer(what) and bottom layer (how).

5.3.1 Top layer

The top layer relates to the purpose driving the roadmap (Robert Phaal, 2004), which is about knowing “why”. Technology road mapping is driven by a need, not a solution (Olin H. Bray, 1997). For example, there may exist even appropriate solutions, but it’s essential to start with the needs, and not from a pre-defined solution. The key issue is to fulfill the needs.

Therefore, software defined network could be the desirable solution, but it’s also possible that there are other options. It’s crucial to clarify the initiatives of why to adopt the new technology software defined network. For example, the needs of Google to adopt SDN is that they want to solve the current private WAN connection problems across the planet and this project took 3 years for production deployment (Google , 2013).

The paper (Robert Phaal, 2004) mentions possible factors in the purpose part, such as market, customer, competitors, environment, industry, business, trends, drivers, threats, objectives, milestones, and strategy. These factors could be divided into several categories that are related to the specific context of Capgemini Netherlands branch. As this thesis introduces the SDN roadmap at a high level, the thesis only analyzes the needs lying at the basis of this thesis. There is a SDN market analysis conducted in 2014 (Capgemini, June 2014), which prepares a market study of Network Functions Virtualization (NFV) and software-defined network (SDN) technology evolution. This market study gives insight in what benefits SDN can bring to Capgemini’ s data centers.

Comparing with market analysis, this thesis provides a different view point of the reasons for Capgemini to adopt software-defined network. These reasons are driven by needs, which are **customers and drivers** mentioned in Phaal’ s paper (Robert Phaal, 2004). Firstly, clients require both stable network service and quick adapting of network service (on requests) in Capgemini data center. There is a gap between clients requirements and current solutions in Capgemini. Besides, there are also potential customer requirements due to computing trends, which are driving network changes (Open Networking Foundation, 2015). Open Networking Foundation, which is an organization aims to develop SDN via open standards, mentions limitations of current networks, including complexity that leads to stasis, inability to the scale, vendor dependence. These limitations of the static architecture of current traditional network technologies can’t meet the needs of the dynamic computing and storage of data centers.

Customers and Drivers	<ul style="list-style-type: none"> • Capgemini data center clients require both stable network service and quick adapting of network service (on request) • Potential customer requirements of dynamic computing and storage of data centers .
-----------------------	--

Table 6 Purpose of adopt SDN

5.3.2 Middle layer and Bottom layer

“The middle layers (know what) provides a bridge or delivery mechanism between the purpose (know why) and resources (know how)” (Robert Phaal, 2004). This paper also mentions some situations which could be included in the middle layer. In this SDN roadmap, this middle layer clarifies the capabilities of SDN, that is to say, what benefits SDN solution can deliver for the initiatives in top layer. The reasoning why middle layer should clarify SDN capabilities is as follows. The goal of this roadmap is to answer the question: when Capgemini could solve the problem in the data center by using SDN. But as mentioned in former chapters, SDN is at the early stage of maturity. It’s crucial to clarify what concrete benefits current SDN solutions can deliver and also what desirable solutions Capgemini wants. It will also help if clarify what SDN cannot deliver yet.

The bottom layer relates to resources. The paper mentions possible elements which could be included in the bottom layer: technology, competences, knowledge, skills, partnership, suppliers and so on.

5.4 Conclusion for roadmap

Due to the limitation of time, we are not able to do more and we can’t investigate details. Thus, there is no concrete conclusion for the question when is the appropriate time for Capgemini to adopt this new technology. This chapter is only a preparation for further software-defined network

research. But it gives a good impression of what should be investigated in more detail to find a more concrete answer. In general, when investigation result in solving the SDN issues and not using more than available resources, the time is appropriate. In addition it is worthwhile to keep in mind, SDN solutions have the potential to solve concerns like security issues.

6 Chapter 6. Discussion and Conclusion

As mentioned in the introduction, this thesis research has two main practical goals. One goal is to introduce network automation using traditional product NetMRI, the other goal is to figure out to what extent network automation using NetMRI can fit into the roadmap of SDN for Capgemini.

Evaluation This thesis is structured by combining different components to answer the two research goals by analysis, design, implementation, and future implication, i.e. by creating a roadmap for software-defined network. The first goal is clarified by analyzing “as is” process using UML, designing “to be” process using flow charts, and implementation of the test lab. The second goal is addressed by creating a high level software-defined network roadmap. The analysis in Chapter 2 is based on UML and on expertise knowledge of Dynamic Multiple Virtual Network (DMVPN). The design of Chapter 3 is conducted using flow chart and knowledge of NetMRI. The implementation of the test lab in Chapter 4 is realized using Perl language and knowledge of NetMRI. The software-defined roadmap is created by critical literature review. But due to the limitation of time and complexity of the second research question, which acts as a future implication of using network automation software NetMRI, the roadmap is only at a high level. The analysis and design are valid because they consist of implementation and original procedures. Besides, necessary improvements have been applied to the analysis and design chapters during the implementation.

NetMRI Results The purpose of this thesis is to solve the provisioning problems in current procedures at Capgemini in order to meet the clients’ requirements. The result showed that the network automation software NetMRI can send provisioning command automatically, but there are lots of issues need further investigation. NetMRI as such is working, because the test lab implementation in Chapter 4 shows it is possible to send commands to a router via a script in NetMRI.

NetMRI and SDN The second goal, addressing the SDN roadmap, is conducted only at a high level due to the limitation of time. To be specific, NetMRI could be regarded as a preparation for adopting new technologies, like SDN, but there is no concrete answer.

Chapter 5 tries to answer the question: when is the appropriate time for Capgemini to adopt the software-defined network. Although there is no concrete answer for this question, the high-level roadmap gives a good estimate of what should be investigated in the future. The initiatives for using NetMRI and software-defined network are the same: clients of the Capgemini data center require both stable network service and adapting network service requests quickly, especially to shorten the provisioning time. That's the main reason we regard software-defined network as a future step for NetMRI. But there is no direct link between NetMRI and software defined network. There are two possible ways to fill the gap between NetMRI and SDN in the future. Firstly, Infoblox (the mother company of NetMRI) might have SDN solutions in the futures, then there should be some recommended solutions if Capgemini chooses to use SDN. But it maybe not an open source solution.

The second possible way is as follows. NetMRI could be regarded as a preparation of the “how” stage in the roadmap mentioned in Chapter 5. Firstly, using NetMRI for network automation can make employees familiar with the transformation procedures. Besides, it is also a good chance to give employees time and direction to gain knowledge that will be useful in the future, for example, programming capabilities for engineers in infrastructure. If software-defined network solutions are mature and Capgemini Netherlands decides to adopt SDN, the current employees can get used to the transformation concepts and procedures, which can save time and money in view of the large scale of recruitment changes.

Implications Based on the results of the test lab, NetMRI as a tool working, but further investigation to solve the issues is needed. So it can be concluded that the management of Capgemini needs to decide whether to continue further investigating in NetMRI. As for the second research question, more research should be conducted regarding the initiatives of adopting software-defined network and what concrete functions software-defined network can provide, as well as what are the current available resources in Capgemini and what resources are needed if one chooses to adopt the software-defined network in order to meet the customers' growing requirements.

Future improvements This thesis is limited by narrowing the scope and limitation of time. There are two fields for future improvements. One is in NetMRI implementation, the other is the improvements in software-defined network roadmap.

Improve NetMRI implementation The future improvements are mainly focusing on the assumptions made during analysis. The list of future improvements has grown with narrowing the scope, which can be seen from Chapter 2 to Chapter 4.

The list in Chapter 4 for future improvements, includes the issues to be solved, it addresses completing the subgoals of the test lab, and it improves the script to allow changing different provisioning parameters. The future improvements should be focusing on the issues list first because some crucial decisions should be made regarding these issues. Besides, the management of Capgemini needs to decide whether to continue to investigate in NetMRI based on the solutions of these issues. After solving these issues, the next step is to complete the subgoals of the test lab and improving the script for different parameters. These parameters are based on the assumptions made in the test lab implementation, which is Chapter 4.3 Set-up of the test lab. For example, changing to another type of router(s).

As Chapter 2 mentions, this thesis only focuses on the standardized services and the first 29 steps in one of procedures documents. In this standardized service, there are also some options: Ethernet, 3G, and 4G. This thesis furthermore focuses on Ethernet only. A solution should be made to address the different options in the standardized services. Future improvements could also investigate the customized services and the remaining steps, which includes provisioning documents both inside and outside Capgemini.

Improve SDN roadmap As for the second research question, more research should be conducted regarding the three layers mentioned in the roadmap model. Firstly, the initiatives of adopting software-defined network and what concrete functions software-defined network can provide need more investigation. Although there is a SDN market analysis (Capgemini, June 2014) available in Capgemini, there is no analysis specifying of what Capgemini wants to achieve and what concrete functions software-defined network can provide. Secondly, more research is needed on the current available resources in Capgemini and what resources are needed if one chooses to adopt the software-defined network, which is the “how” layer in the roadmap model.

To sum up, this paper contributes to a process of analyzing current “as is” provisioning procedures, of designing “to be” procedures, and of implementing the design in a test lab environment. Besides, it provides an overview for further research on adopting software-defined network.

Bibliography

Accenture. (2015). *Boosting business agility through software-defined networking*. Retrieved from Accenture: <https://www.accenture.com/bw-en/insight-business-agility-software-defined-networking>

Accenture. (2013, Feb). *Technology Vision 2013: Software-Defined Networking—Video*. Retrieved Jul 2015, from Accenture: <http://www.cas-us.com/us-en/Pages/insight-software-defined-networking-video.aspx>

Arendsen, E. (2014). *Capgemini DMVPN Spoke Installation Procedure v2.3*.

Bell, D. (2004). UML basis: The component diagram. *developerWorks* .

Borao, M. (1999). Use case Modeling Guidelines.

Capgemini. (2014). *High Level Design Maxnet*.

Capgemini. (2015). *LAB Installation procedure DMVPN-location*.

Capgemini. (June 2014). *Market Study of NFV and SDN Evolution - V1.0*.

Capgemini. (2014). *Wide Area Network - DMVPN spoke connection Product Description Document*.

CCNA Intro Exam Certification Guide. Cisco Press.

Cisco. Chapter 9 Configuring PKI. In *Cisco VPN Services Port Adapter Configuration Guide*.

Cisco Routers. (n.d.). *CONFIGURING POINT-TO-POINT GRE VPN TUNNELS - UNPROTECTED GRE & PROTECTED GRE OVER IPSEC TUNNELS*. Retrieved April 9, 2015, from Firewall.cx: <http://www.firewall.cx/cisco-technical-knowledgebase/cisco-routers/868-cisco-router-gre-ipsec.html>

Cisco Services and Technologies. (n.d.). *UNDERSTANDING CISCO DYNAMIC MULTIPOINT VPN - DMVPN, MGRE, NHRP*. Retrieved April 9, 2015, from Firewall.cx: <http://www.firewall.cx/cisco-technical-knowledgebase/cisco-services-tech/896-cisco-dmvpn-intro.html>

Gartner. (2014). Basic API Management Will Grow Into Application Services Governance.

Gartner. (10 April 2015). *Should Your Enterprise Deploy a Software-Defined Data Center?* Gartner.

Gartner. (2015). *The Top 10 Strategic Technology Trends for 2015*.

Gartner. (10 Sep 2014). *What is the Value of a Software-Defined Data Center?* Gartner.

Google . (2013). *B4: Experience with a Globally-Deployed Software Defined WAN*.

IBM. (2014). *Implementing IBM Software Defined Network for Virtual Environments*. Retrieved 2015, from IBM: <http://www.redbooks.ibm.com/abstracts/sg248203.html?Open>

Infoblox. Defining Triggered Jobs. In *Infoblox Network Automation Administrator Guide Release 6.9* (p. 228).

Infoblox. (2014). *Infoblox Network Automation Administrator Guide Release 6.9*. Infoblox.

- Jones, D. (2006). *The Shortcut Guide To Automating Network Management and Compliance*. In D. Jones. Realtime publishers.
- Kruiswijk, B. (2013). ICT – Architecture course, Service Oriented Architecture, MoSCoW prioritising.
- L.P.J.Groenewegen, A.W.Stam, P.J.Toussaint, & Vink, E. (2005). *Paradigm as Organization-Oriented Coordination Language*. 14.
- (2009). MoSCow Analysis . In *A Guide to the Business Analysis Body of Knowledge*. International Institute of Business Analysis.
- Muhammad H.Raza, S. C. (2014). *A Comparison of Software Defined Network (SDN) Implementation Strategies*. Elsevier B.V.
- Object Management Group. (2005, July). *Introduction To OMG's Unified Modeling Language (UML)*. Retrieved July 12, 2015, from Object Management Group: http://www.omg.org/gettingstarted/what_is_uml.htm
- Object Management Group. (2014). *Use Case Diagrams*.
- Olin H. Bray, M. L. (1997). *Technology Roadmap: The Integration of Strategic and Technology Planning for Competitiveness*. Albuquerque, New Mexico.
- Open Networking Foundation. (2015). *Computing Trends are Driving Network Change*. Retrieved from Open Networking Foundation (ONF): <https://www.opennetworking.org/sdn-resources/sdn-definition>
- Oracle. (2005). *Getting Started With Use Case Modeling*.
- Puppet Labs. (2015). *What is Puppet*. Retrieved from Puppet Labs: <https://puppetlabs.com/puppet/what-is-puppet>
- Robert Phaal, C. J. (2004). Technology roadmappig-A planning framework for evolution and revolution. *Technological Forecasting & Social Change* , 5-26.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2004). Chapter 10 Deployment view. In *Unified Modeling Language Reference Manual* (p. 752). Addison-Wesley Professional.
- Saunders, M. (2009). *Research Methods for Business Students* 5th Edition.
- Visual Paradigm. (2011, December 08). *Writing Effective Use Case*. Retrieved April 07, 2015, from Visial Paradigm Website: <https://www.visual-paradigm.com/tutorials/writingeffectiveusecase.jsp>
- Witteveen, L. (2013). *High Level Design Public Key Infrastructure Services*.
- Zhensheng Zhang, Y.-Q. Z. (2004). An Overview of Virtual Private Networking (VPN): IP VPN and Optical VPN. *Photonic Network Communications* , 213-225.

Appendix

1. OSI Layer 2 and Layer 3 Functional Summary

OSI stands for Open System Interconnection, and consists of seven discrete layers, which are responsible for different functions regarding the communication between various network's entities. This section provides a brief overview of the basic functions in Layer 2 and Layer 3, and the interaction between these two layers.

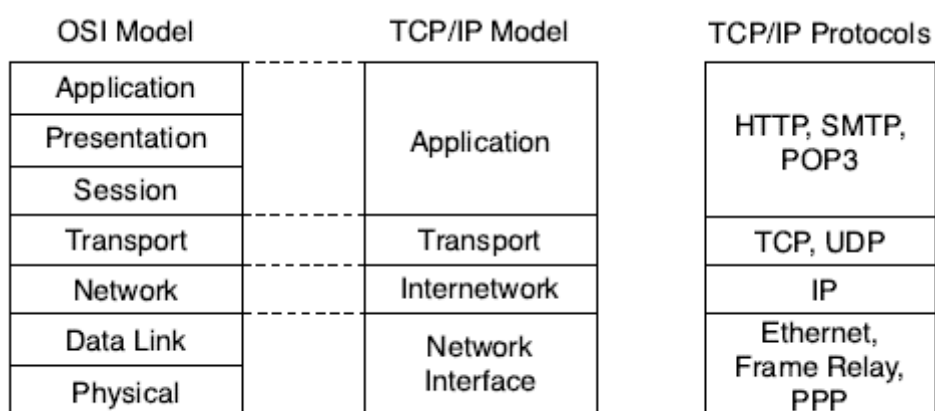


Figure 21 Using OSI Layers for Referencing Other Protocols (CCNA Intro Exam Certification Guide)

OSI Layer	Functional Description	Example protocols
Network (Layer 3)	Routers use logical addressing (IP address) for path determination	IP
Data link (Layer 2)	-Combination of bits into bytes, and bytes into frames. -Switches access to the media using MAC address. -Error detection and error recovery.	HDLC, Ethernet

Table 7 OSI Layer2 and Layer3 Functional Summary

MAC address: Media Access Control address

A standardized data link layer address that is required for every device that connects to a LAN. Ethernet MAC addresses are 6 bytes long and are controlled by the IEEE. Also known as a hardware address, a MAC layer address, and a physical address (CCNA Intro Exam Certification Guide).

MAC addresses are the physical address in the computer. Each computer could have two MAC addresses, one for physical through the cable, the other is for wireless.

Local Area Network (LAN) standards and protocols define how to network between devices that are relatively close together (CCNA Intro Exam Certification Guide).

Ethernet protocol is used in Layer 2 by using cables, which are used to near the location. The computers are connected by Ethernet protocol in LAN.

2. LAN vs. WAN

Local area network (LAN) standards and protocols define how to network between devices that are relatively close together (CCNA Intro Exam Certification Guide).

Wide area network

- WAN standards and protocols define how to network between devices that are relatively far apart.
- WAN protocols used on point-to-point serial links provide the basic function of data delivery across that one link. The two most popular data-link protocols used on point-to-point links are High-Level Data Link Control (HDLC) and Point-to-Point Protocol (PPP) (CCNA Intro Exam Certification Guide)
- For the PPP only one IP added.

HDLC: High Level Data Link Control, data-link protocols used on point-to-point links

- HDLC defines framing that includes an address field, a frame check sequence(FCS) field, and a protocol type field (CCNA Intro Exam Certification Guide).

PPP: Point-to-Point Protocol, data-link protocols used on point-to-point links

- PPP behaves exactly like HDLC. The framing looks identical. There is an address field, but the addressing does not matter. PPP does discard error frames that do not pass the FCS check (CCNA Intro Exam Certification Guide).

The Figure 22 below shows a Local Area Network (LAN) with two buildings and two switches in each building.

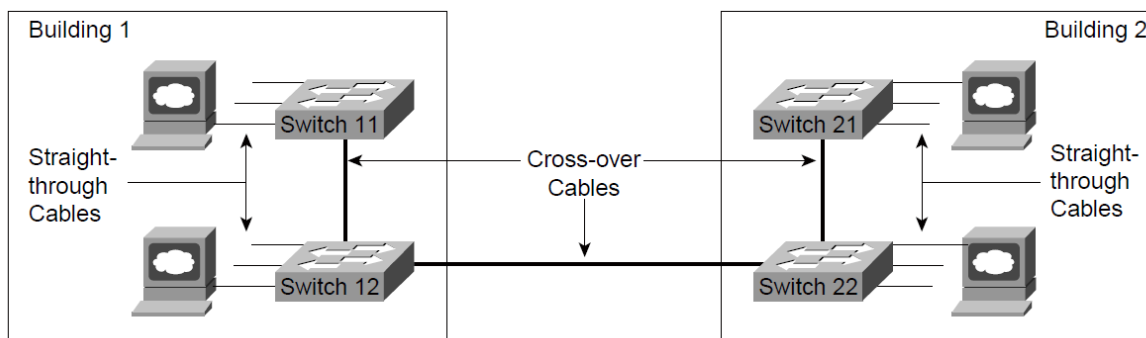


Figure 22 Example LAN, Two Buildings (CCNA Intro Exam Certification Guide)

The big distinction between LANs and Wide Area Network (WANs) relates to how far apart the devices can be and still be capable of sending and receiving data. LANs tend to reside in a single building or possibly among buildings in a campus using optical cabling approved for Ethernet. WAN connections typically run longer distances than Ethernet.

The connection between different LANs and Wide Area Network (WAN) is used on routers with serial network.

As the Figure 23 below shows, Point-to-point WAN links provide basic connectivity between two points.

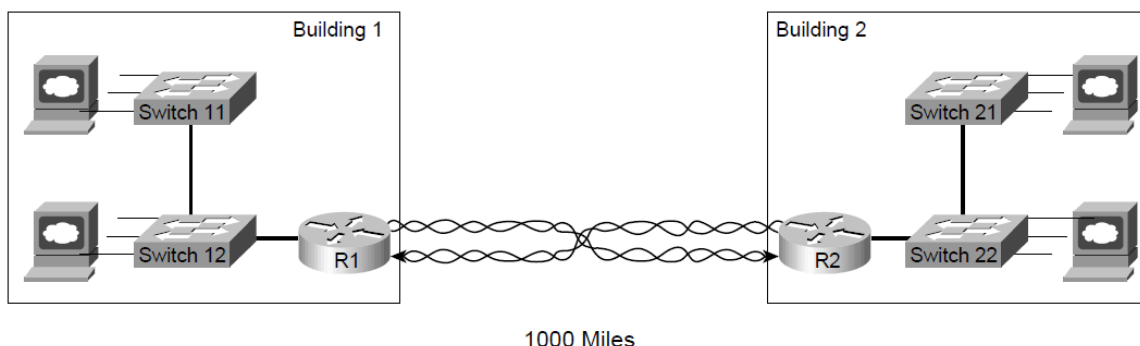


Figure 23 Point-to-point WAN (CCNA Intro Exam Certification Guide)

3. VLAN and Trunking

VLANs allow a switch to separate different physical ports into different groups so that traffic from devices in one group never gets forwarded to the other group. Also, multiple switches can be connected together, with traffic from multiple VLANs crossing the same Ethernet links, using a feature called trunking.

For example, in one piece device with 24 ports, we could set 12 ports for the finance department and 12 ports for the mail system. To connect finance department with the mail system, by connecting the

ports in that device, might combine with firewall, to ensure the direction of the connection is only one way, that is to say, from the finance department to the mail system.

Trunk:

- When using VLANs in networks that have multiple interconnected switches, you need to use VLAN trunking between the switches. When sending a frame to another switch, the switches need a way to identify the VLAN from which the frame was sent. With VLAN trunking, the switches tag each frame sent between switches so that the receiving switch knows which VLAN the frame belongs to.
- With trunking, you can support multiple VLANs that have members on more than one switch. For instance, when Switch1 receives a broadcast from a device in VLAN1, it needs to forward the broadcast to Switch2. Before sending the frame, Switch1 adds another header to the original Ethernet frame; that new header has the VLAN number in it. When Switch2 receives the frame, it sees that the frame was from a device in VLAN1, so Switch2 knows that it should forward the broadcast only out its own interfaces in VLAN1.

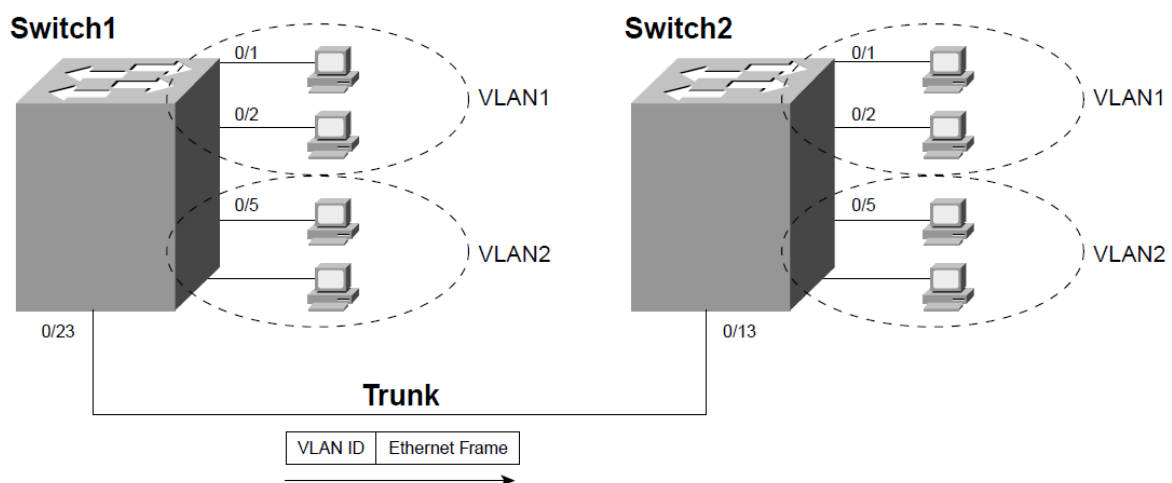


Figure 24 VLAN Trunking Between Two Switches (CCNA Intro Exam Certification Guide)

4. CLI

Cisco uses the acronym for CLI to refer to the terminal user command-line interface to the IOS. The term CLI implies that the user is typing commands at a terminal emulator, or a Telnet connection. The user can access the router through the console, through a dialup device through a modem attached to the auxiliary port, or by using Telnet. The router has RJ-45 receptacles for both the console and the auxiliary port (CCNA Intro Exam Certification Guide).

The figure below shows the cable pinouts.

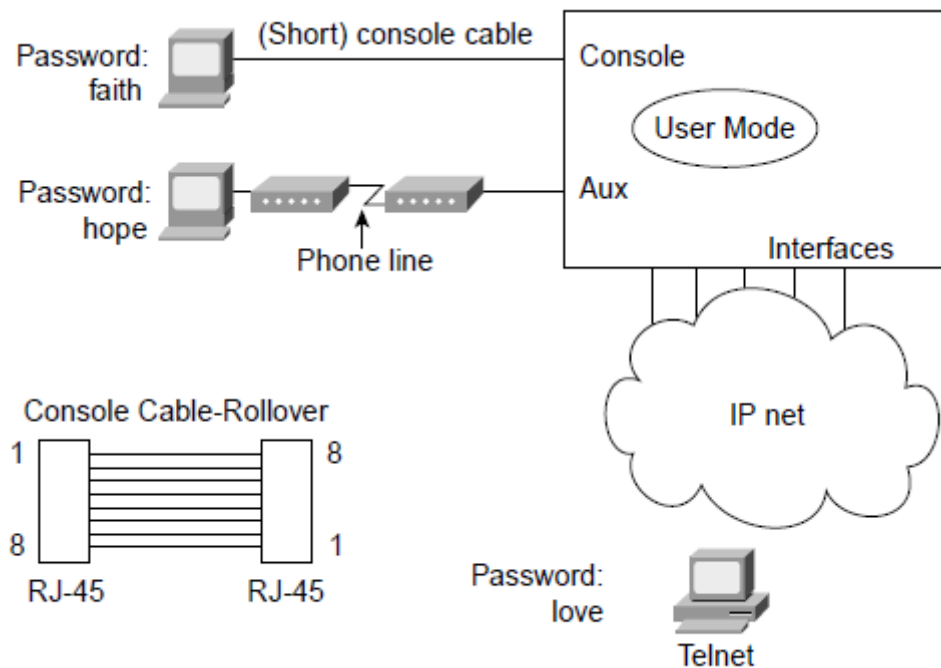


Figure 25 CLI Access (CCNA Intro Exam Certification Guide)

5. PKI

Public Key Infrastructure (PKI) is a two key (asymmetric) encryption system for communication. The key pair includes a public key and a private key. The public key is available and distributed to the public. The private key is secret and it being used only by the key owner. Instead of a specific technology, PKI is a framework, providing authentication and confidentiality.

A PKI is composed of the following entities:

- Certificate authority (CA)

Certificate authority (CA) is an organization responsible for issuing, revoking, and distributing digital certificates. Certificate authority could be a third party organization. Companies or organizations can also have an in-house CA. The Microsoft Public Key Infrastructure supports a hierarchical CA trust model, called the certification hierarchy (Witteveen, 2013). The CA at the top of the hierarchy is called a root CA, which is self-certified by using a self-signed CA certificate.

- Digital certificates (DA)

Digital certificates (DA) are issued by a certificate authority (CA). Digital certificates (DA) contain information, such as the certificate validity period, peer identity information, encryption keys that are used for secure communications, and the signature of the issuing

CA. Authentication is realized by Digital Certificates (DA), which is a certificate that verifies whom the public key belongs to.

- Registration authorities(RA)

Registration authorities (RA) offload the CA by processing enrollment requests. Registration authorities (RA) verify the prospective key owner's identify and send it to the CA to issue a certificate.

To participate in the secured communications, every entity (a person or a device) needs to enroll in the PKI, a process where the entity generates a key pair and has their identity validated by a trusted entity (also known as a CA or trustpoint) (Cisco, p. 2 Configuring PKI). After each entity enrolls in a PKI, every peer (also known as an end host) in a PKI is granted a digital certificate that has been issued by a CA. (Cisco, p. 2 Configuring PKI) Peers need to exchange digital certificates when they must negotiate a secured communication session (Cisco, p. 2 Configuring PKI).

Configuring PKI involves the following tasks: (Cisco, p. 2 Configuring PKI)

- The end host (e.g. router) must generate a pair of RSA keys and exchange the public key with the certificate authority to obtain a certificate and enroll in a PKI.
- Configuring authorization and revocation of certificates within a PKI. A properly signed certificate is authorized to use methods such as certificate maps, PKI-AAA, or a certificate-based access control list (ACL). The revocation status is checked by the issuing CA.
- Configuring certificate enrollment, which is the process of obtaining a certificate from a certificate authority.
- Storing public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adelman (RSA) keys and certificates. These credentials can be stored in the default location on the router.

Understanding Manual Certificate Enrollment (Cisco)

The manual certificate enrollment feature allows users to generate a certificate request and accept certification authority (CA) certificates as well as the switch's certificates.

Command	Purpose
1. Router (config) # crypto pki trustpoint <i>name</i>	Declares the CA that your switch should use and enters ca-trustpoint configuration mode.
2. Router (ca-trustpoint) # enrollment terminal	Specifies manual cut-and-paste certificate enrollment.
3. Router(ca-trustpoint) #crypto pki	Authenticates the CA (by obtaining the

authenticate <i>name</i>	certificate of the CA)
4. Router (ca-trustpoint) # exit	Exits ca-trustpoint configuration mode and returns to global configuration.

Table 8 Configuring certificate enrollment (Cisco)

6. Ping

Ping refers to Packet INternet Groper, which uses the Internet Control Message Protocol (ICMP), sending and receiving message called an ICMP echo request to verify connectivity with another IP address.

The ICMP protocol provides a wide variety of information about the health and operational status of a network.

7. Background knowledge for DMVPN

The Client wants a secure internet connection. As the normal internet connection is not secured, Virtual Private Network (VPN) is a promising solution comparing with alternative approaches, especially for cost saving and scalability (Zhensheng Zhang, 2004). Virtual Private Network (VPN) refers to the secure communication between a set of sites and a closed user group (Zhensheng Zhang, 2004). For example, it could be the connection between remote locations and the central sites (data centers) across untrusted networks (internet).

VPNs support at least three different modes of use, including remote access client connections, LAN-to-LAN (local area network) internetworking and controlled access within an Intranet. In this thesis, we only focus on the LAN to LAN VPN.

Dynamic multipoint VPN (DMVPN) is a dynamic tunneling architecture of VPNs based on standard protocols GRE, IPsec and Next Hop Resolution Protocol (NHRP) to meet the increasing demands of enterprise companies to be able to connect branch offices with head offices. DMVPN provides the capability of creating a dynamic-mesh VPN network without having to pre-configure all possible static tunnel endpoints. DMVPN could ensure the connection between branch offices with head offices while keeping costs low, minimizing configuration complexity and increasing flexibility.

The DMVPN connection service connects the local network in a branch office to the company network at the data center. Additionally, the DMVPN connection service offers connectivity from the branch office to external services such as internet web and cloud services, EFT and PIN providers, and other external services that the customer needs for his business.

With DMVPN, one central router, which usually placed at the head office, acts the role of the hub while all other branch routers are spokes can access the company's resources by connecting to the hub router (Cisco Services and Technologies). The hub router is assigned a static public IP address while the branch spoke routers can be assigned static or dynamic public addresses. There are two mainly deployment designs: one is the hub and spoke, which used to perform headquarters to branch interconnections, the other is spoke to spoke, used to perform branch to branch interconnections.

Combining multiple Generic Routing Encapsulation (mGRE) Tunnels, IPsec encryption and Next Hop Resolution Protocol (NHRP) to perform its job and save the administrator's need to define multiple static crypto maps and dynamic discovery of tunnel endpoints (Cisco Services and Technologies).

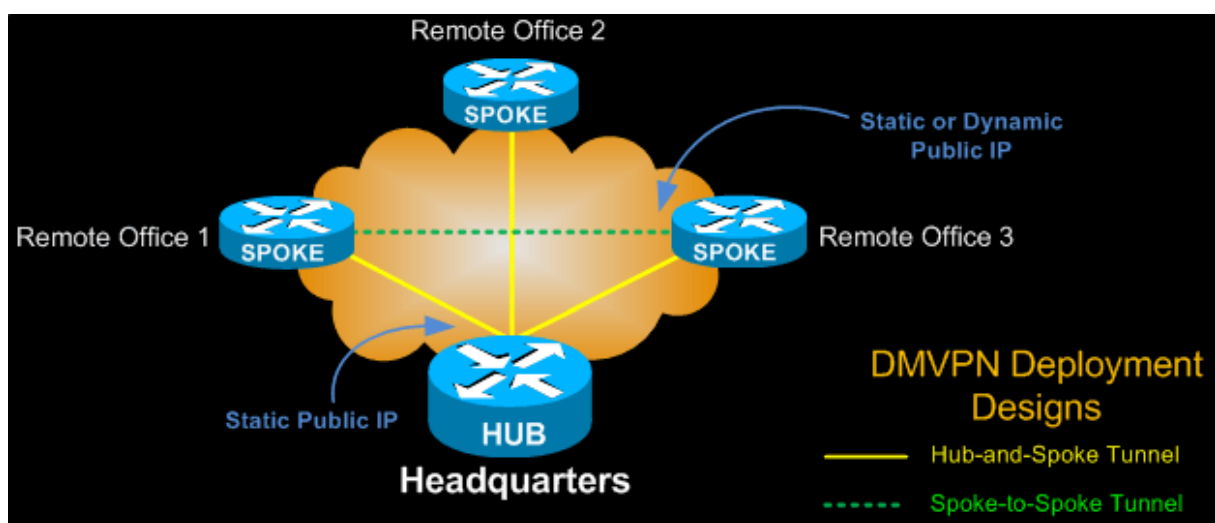


Figure 26 DMVPN Two Deployment Design(Cisco Services and Technologies)

NHRP is layer 2 resolution protocol and cache, much like Address Resolution Protocol (ARP) or Reverse ARP (Frame Relay) (Cisco Services and Technologies). It is pre-shared key authentication of spoke routers to central hub routers. ARP is an internet protocol used to map an IP address to a MAC address.

mGRE Tunnel Interface is used to allow a single GRE interface to support multiple IPSecs tunnels and helps dramatically to simplify the complexity and size of the configuration. Generic Routing Encapsulation (GRE) is a tunneling protocol developed by Cisco that allows the encapsulation of a wide variety of network layer protocols inside point-to-point links (Cisco Routers). A GRE tunnel is used when packets need to be sent from one network to another over the Internet or an insecure network. A GRE interface definition includes an IP address, a tunnel source, a tunnel destination, and an optional tunnel key. The mGRE interfaces do not have a tunnel destination, because it cannot be used alone. NHRP fills this gap by telling mGRE where to send the packets.

How it works:

The hub router, which maintains a special NHRP database with the public IP addresses of all configured spokes, undertakes the role of the server while the spoke routers act as the client, registering its public IP address with the hub and queries the NHRP database for the public IP addresses of the destination spoke, which needs to build a VPN tunnel (Cisco Services and Technologies).

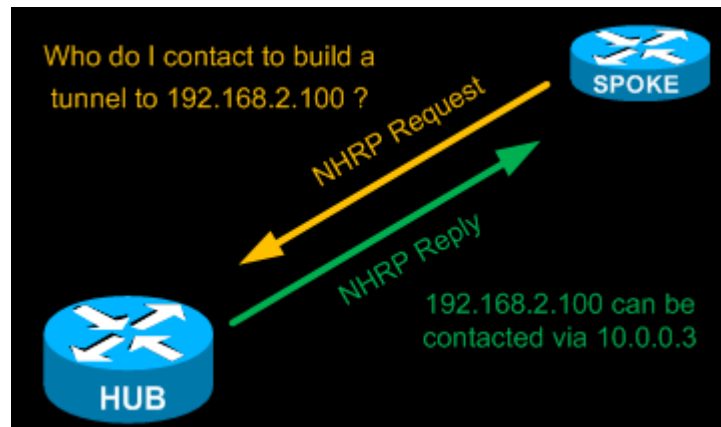


Figure 27 Spoke registers public IP address

There are two devices involved in total: the spoke router and the internet router, which is R1 and R2 respectively. These two routers are installed in parallel.

Spoke router at remote locations establish a VPN to the hub router at central sites and create an initial hub-to-spoke network; VPNs between spokes can be dynamically built on demand via NHRP.

The internet router, **R2** is an interface that provided by the provider for the new internet connection. This device has the interface to connect the WAN depending on the situation of local network provider, for example, cable internet, fiber cable.

Hub router only needs a single tunnel interface for a DMVPN; spoke routers are added to the DMVPN without having to change the initial DMVPN configuration of the hub router.

Dynamic routing protocols such as EIGRP, OSPF and BGP are generally run between the hubs and spokes to allow for growth and scalability.

But the provisioning of the routing protocols is still a lot of manual work.

DMVPN benefits:(Cisco Services and Technologies)

The benefits of DMVPN are listed below, following the detailed description.

- Simplified Hub Router Configuration

- Full Support for Spoke Routers with Dynamic IP Addressing
- Dynamic Creation of Spoke-to-spoke VPN Tunnels
- Lower Administration Costs
- Optional Strong Security with IPsec

Below are the detailed description.

- Simplified Hub Router Configuration

Comparing with the traditional configuration, which needs to configure both sides of the routers, DMVPN could reduce half of the work by using the hub router on the company side, meaning the user only needs to configure the spoke router.

No more multiple tunnel interfaces for each branch (spoke) VPN. A single mGRE, IPsec profile without any crypto access lists, is all that is required to handle all Spoke routers. No matter how many Spoke routers connect to the Hub, the Hub configuration remains constant.

- Full Support for Spoke Routers with Dynamic IP Addressing

Spoke routers can use dynamic public IP Addresses. Thanks to NHRP, Spoke routers rely on the Hub router to find the public IP Address of other Spoke routers and construct a VPN Tunnel with them.

- Dynamic Creation of Spoke-to-Spoke VPN Tunnels

Spoke routers are able to dynamically create VPN Tunnels between them as network data needs to travel from one branch to another.

- Lower Administration Costs

DMVPN simplifies greatly the WAN network topology, allowing the Administrator to deal with other more time-consuming problems. Once setup, DMVPN continues working around the clock, creating dynamic VPNs as needed and keeping every router updated on the VPN topology.

- Optional Strong Security with IPsec

Internet Protocol security (IPsec) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session (Witteveen, 2013).

IPSecurity can be configured to provide data encryption and confidentiality. IPsec is used to secure the mGRE tunnels by encrypting the tunnel traffic using a variety of available encryption algorithms.

By using router protocols for routers in DMVPN, the routers could change the routing automatically to the other method if there is something wrong with the original routing path.

8. SNMP

Consisting of a group of protocols, Simple Network Management Protocol (SNMP) is an application layer protocol used specifically for network device management (CCNA Intro Exam Certification Guide). For instance, the Cisco Works network management software product can be used to query, compile, store, and display information about the operation of a network. In order to query the network devices, Cisco Works uses SNMP protocols.

9. DHCP

A protocol to assign IP address to the device automatically.

10. VRF

VPN Routing and Forwarding (VRF) is a technology used in computer networks that allows multiple instances of a routing table to co-exist within the same router at the same time. Because the routing instances are independent, the same or overlapping IP addresses can be used without conflicting with each other.

11. CIDR

IP subnetting creates a vastly larger number of smaller groups of IP addresses by introducing the subnet part of the address between the network and host parts of the addresses. Computers use a mask to define the size of the network and host parts of an address. Routers can route based on the combined network and subnet parts. With 32-bit binary number, a subnet mask helps define the structure of an IP address. With the calculation of the IP address and subnet mask, subnet mask could be 24/30/32 bits (Chapter 12 CCNA). For example, 10.0.0.0/24, the network 10.0.0.0 followed by the notation of /24, called prefix notation. It is simply a shorter way to write the mask.

12. Telnet

Telnet is the standard terminal emulation protocol in the TCP/IP protocol stacks used for remote terminal connection, enabling users to log in to remote systems and use resources as if they were connected to a local system.

13. Putty

PuTTY is an SSH and Telnet client, which is a free and open-source terminal emulator supporting several network protocols. In this test lab, we use the Telnet protocol to test manually the procedures of building the tunnel.

14. Preparation for the test lab

- Introduction of NetMRI
- Required changes from Infoblox
 - Initial settings

The NetMRI is connected to the network, which the test lab is also accessible. That's to say, NetMRI and test lab should physically connected. Besides, the chapters "set up" and "Discovery with a New Network Automation Deployment" are finished, which are in the document "Infoblox Network Automation Administrator Guide, Release 6.9.

- Updates the version supporting VRF

As mentioned in the beginning of the thesis, the NetMRI was bought a year ago, without using or testing for how to use it. The current version does not support Virtual Routing Forwarding (VRF). The software upgrade is needed.

- Required changes inside Capgemini

The test lab is built separately instead of using the production environment. This is to protect the production environment and ensure the data center is operating without interfering.

- Management VPN
- User account
- Terminology
- Preparation for the spoke router

The spoke router have to be configured as it would have been if it was installed by an on-site engineer from the third party. The document "LAB Installation procedure DMVPN-location" session 1: Preparation addresses the detail procedure for the preparation for the spoke router.

15. SDDC

The software-defined data center is the underpinning for achieving business agility. DevOps and cloud services require an SDDC in order to enable greater end-to-end automation through the use of APIs, policy and orchestration. Service provider infrastructure. Public cloud service providers most often take on new workloads from their customers, and, thus, do not have to contend with legacy constraints. Moreover, providers seek to streamline processes and to improve agility and speed of provisioning, which fits well with software-defined goals. They also have a mindset to automate everything that is expected to be done more than once. Gartner believes that service providers are in

a very good position to exploit the SDDC, and will be early adopters of it as they justify new business (Gartner, 10 April 2015).

An SDDC enables an automated, agile and programmable infrastructure (Gartner, 10 April 2015).

16. Control plane

Control plane makes decisions about where traffic is sent. The control plane functions include the system configuration, management, and exchange of routing table information. The router controller exchanges the topology information with other routers and constructs a routing table based on a routing protocol.

17. Data Plane (Forwarding Plane)

Forwarding traffic to the next hop along the path to the selected destination network according to control plane logic.