UiO : Department of Informatics
University of Oslo

Dynamic distribution of SDN controllers

For Tactical Networks

Ida Marie Frøseth

Master's Thesis Spring 2016

Dynamic distribution of SDN controllers

Ida Marie Frøseth

14th March 2016

Abstract

The text of the abstract; typically, 2–5 sentences.
What it is all about?? Title page, abstract, ...

Preface

This thesis was written as a part of my Masters degree in Informatics at the University of Oslo. Most of the work was done in the period from August 2016 to May 2017. The thesis was done in collaboration with Norwegian Defence Research Establishment, at Kjeller.

The thesis is the original and independent work by the author, Ida Marie Frøseth, supervised by Frank Eliassen(UiO) and Ole–Ingar Bentstuen(FFI).

INCLUDE: How literature have been examined and where the publications have been found

A large part of the work was to study related work, and state of the art technologies, and this knowledge has been carefully cited.

Acknowlegement

The former TACOMS community in general and a special thanks to the Swedish team with Janne SjÃ¶lander, Joachim Orrblad and Per CarlÃ©n which have not only - answered to all my questions about TACOMS during the work of this thesis, - but also helped me when I have been in trouble with the Norwegian node at the TACOMS CIT exercies. Their expertise. They have always answered my questions and helped me when Google did not have the answer. - Design

I also want to raise a special thanks to Roger Langfelt which have been very supportive about my topic and, by following the TACOMS project since its birth, have provided me with insight information about its history ... -> History,

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

1.1 Motivation

There is no doubt that the Internet has been a great success, and the distribution of network protocols have made networks autonomous and fault tolerant[**SDN_Comprehencieve**]. However, during the last decades, the computing trends have changed drastically [4] from being static homogeneous client-to-server communication to highly mobile and heterogeneous networks [**future_internet_ANA**] [4]. The content have also changed from static text or web pages to include real-time multimedia and big data. In addition, the security demands are driven to a whole new level by the dependency of IT in both government, banking and transport. The change is due to the evolution of virtualized servers, the rise of cloud services, the explosion of different (mobile) computing devices and wireless communication. Since IP networks never has been developed to meet the needs of todays network, it have made them almost unmanagable and very rigid.

//////////THIS DEFINITLY HAS TO BE REWRITTEN!//////////////////////// The tight coupling between the control plane (the part of the network that make the forwarding decision) and the data plane (the part of the network/a switch that actually forwards a packet from one port to another) have lead to a complexity that are very expensive, static and almost unmanageable.

In the military domain the same problem arise, and the largest problem is often the lack of flexibility and agility.

To remove the disadvantages we have in todays network with highly coupled data / control plane and rigid configuration, the best solution is often to centralize the controller, and this is true for big data centers, but in a military context this is not always desirable -> because we need each node to be autonomous in case they are seperated from the core network (as they usually are). Military networks need some kind of automatic and dynamic way of assigning SDN controllers to switches. As you will see in the related work section some have tried to this before but.... maybe aiming for a different domain?

This thesis will look at mechanisms to best solve the distribution of

SDN controllers in a military context but at the same time achieve the most advantages from having the controllers centralized.

–What is wrong with todays networks and how would SDN help us? – What problem do we have in military networks? => New era within networking -SDN

NOTES - Use cases for SDN in a military operation?

- More flexible networks - save time and money
- QoS
- Added security (?)

1.2 Derived problem description

How to dynamically place SDN controllers in a military networks to.....?

1.3 Method

Software engineering -> lab experiment

1.4 Related work

[2] implemeted what they called DIstributed SDN COntrol plane (DISCO) for mission-critical networks. They implemented the AMQP a message-oriented (pub/sub) middelware protocol on top of the Floodlight controller (A Messenger which subscribe to recive $Packet_IN$ from the core module / OpenFlow events, and several Agent which handles pub/sub operations with other nodes). The drawback with their solutions is that they donÂ´t support dynamically domains. When a controller is assign a domain, they are in charge for that domain. Does not support a hierarchi of controllers. In tactical networks which are highly dynamic and XXX there is a demand for nodes being autonomous. With the solution of [2] it is not possible to take advantages of a centralized controller. To take advantages of centralized controll, but at the same time being autonomous it can be some sort of hierarchy similar to NTP, where stratum 0 is the top tier, and if there is a stratum 0 controller in the network, he is in charge. Else if there is only stratum 1 nodes in the network the controllers communicate and cooparate in a distributed manner. More like a ANA approach with a bootstrap method, Membership database ++(?)[**ana**]

[6] propose a method for distributing the load among controllers by developing a Dynamic Resource Discovery Protocol for Software Defined Networks as an alternative to the usual Link-Layer Discovery Protocol.

"a protocol that allows to the controllers discover the network topology and distribute its management by themselves."

"The process of creating the control layer is executed in two phases, the forwarding phase where the controllers announce their presence and the nodes that start the creation of the control channels (leaf nodes) are discovered, and the backward phase where the nodes decide which node to join in direction of a controller, thus creating the control layer."

"time-efficiently managed since the switches are managed by their nearest controller"

" Although this work only takes into account the delay other parameters may be considered when building the control layer."

"Recently, Google has presented their experience with B4 [1], a global SDN deployment interconnecting their datacenters. In B4, each site hosts a set of master/slave controllers that are managed by a gateway."

[11] proposed an approached called InitSDN to bootstrap the distributed SDN architecture and deploy distributed controllers. Their solution relied on dividing the network in two slices, one control slice for controlling controllers and the other for data.

Their architecture consists of six modules (modular software). Two basic services named Discovery and Topology which is responsible for discovering of the switches and hosts by sending LLDP packets and Network Hypervisor which slice the network in control and data slice (here they used the Flowvisor - NFV controller). In addition they have four control plane configuration modules namely "control plane logic partitioning configuration", "Control Plane Logic Synchronization configuration" (uses Apache Zookeeper), "Control Plane Topology Configuration", "Host Remote Access Configuration"

"SDN architecture envisions a centralized control plane, which may result in adverse consequences to the reliability and performance [3]. Recent efforts have proposed a logically centralized but physically distributed control plane [3]."

"designed to make SDN more flexible, reliable, fault-tolerant without adding complexity to the controllers."

"InitSDN can increase or decrease the size of slices dynamically, change the topology of the controlslice, change the coordination mechanism among the controllers (e.g. use Zookeeper or Chubby, etc) to adapt to network topology changes or to dynamic network loads or simply as part of an upgrade."

[3] Read more "ElastiCon introduces a mechanism to change the active control connections from one controller to another, dynamically on run time. The mechanism defines the message exchange required to achieve the control path migration."[1]

"To address this problem, we propose ElastiCon, an elastic dis- tributed controller architecture in which the controller pool is dy- namically grown or shrunk according to traffic conditions. To ad- dress the load imbalance caused due to spatial and temporal varia- tions in the traffic conditions, ElastiCon automatically balances the load across controllers thus ensuring good performance at all times irrespective of the traffic dynamics. We propose a novel switch mi- gration protocol for enabling such load shifting, which conforms with the Openflow standard. We further design the

algorithms for controller load balancing and elasticity. We also build a prototype of ElastiCon and evaluate it extensively to demonstrate the efficacy of our design." Maybe we want a controller to controll another controller?

[1] READ MORE Network Virtualization and Virtual SDN controllers for loadbalancing.

1.5 Thesis structure

Chapter 2

State of the Art

2.1 Software defined networks

In this section I will first try to define what a SDN are. Then we will look at different components of an SDN like the controller, the networking element and the different APIs.

2.1.1 SDN terminology and definition

If you read books or papers about SDN you would most likely find many different flavor of a definition for SDN. Some will define it as a framework [9], other defines it as an architecture with a centalization of the control plane [5] [14]. Nevertheless, what they usually have in common is that they define SDN to be a way to make networks programmable by separating the control and data plane and connecting them through well defined APIs[9]. Where the dataplane is the part of the network that forward data from one port to another based on the Forwarding Information Base (FIB) or forwarding table and the control plane is the part of the network that make forwarding decision and populate the FIB.

I will use this definition through this thesis and I would also emphasize that the location of the control plane does not have to be a single centralized controller. There are many advantages by centralizing the control plane like faster response to link failure because the information does not have to propagate through multiple nodes, loop avoidance since the controller has the complete picture, added agility when there is only one controller to configure, increased reliability due to less direct human interaction and centralized policy enforcement [9]. The obvious drawbacks are a single point of failure and attack focus, scale and latency when updating the switch which can lead to temporarily microloops (also an issue in todays networks) [9] [**p 48**, 5]. In addition military networks are usually connected with radio links or even sattelite with high latency and low bandwidth which amplifies the drawbacks with a centralized controller. As we will see later in this thesis there has to be some sort of dynamic controll assignment so that the network can take advantage of centralized controller when it is in a stable state but also has some kind of mechanism to function

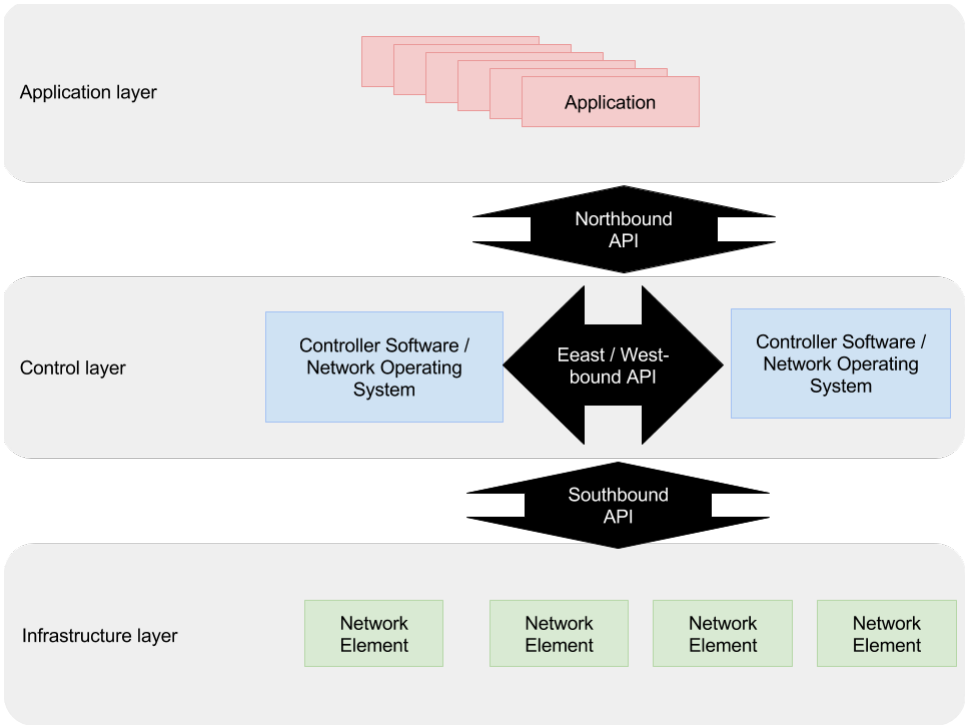

Figure 2.1: SDN components

autonomously when disconnected from the controller.

Figure 2.1 shows the components of a SDN, with the data plane which resieds on a Network element, the control plane on the SDN controller, the management plane which are annotated as applications or services and standardised APIs to communicate between these elements. The next few section would go more in details about these component.

2.1.2 Networking Element

In traditional networks it is usual to distinguish between routers which operates on layer three and switches which operates on layer two,there are even a third class of devices known as L3 switches. The biggest difference between these devices lies in the control plane and how they get knowledge about the network and populate the FIB. When looking at the data plane all devices have the same mission - to forward data entring on one port to another port. In SDN the control plane is logically separated from the physical data plane, hence all these devices are really just forwarding elements.

2.1.3 SDN Controller

======THIS SECTION HAS TO BE REWRITTEN======== The device that holds the controller logic a software defined network is often called a SDN controller or a Network Operating System (NOS). Many examples: OpenDaylight (open source controller) Floodlight, Cisco XNC, POX, NOX,

ONOS, Beacon, Trema, Maestro, NodeFlow. There are huge differences between these controllers. Both regarding their APIs and what features they support.

2.1.4 XX-bound API

As mentioned in 2.1.1 SDN is about making networks programmable through well defined APIs. The purpose of an API is to enable interaction between components, and . As figure 2.1 shows it is usual to categories the APIs in three different categories. The Southbound API is the interface towards the networking element, and OpenFlow and Netconf are examples of open source Southbound APIs while OnePK is a Cisco propiritary protocol. The Northbound API is the interface of the controller towards applications or services. These API make some abstractions so the application should not have to care about the detailed configuration of the networking element. A often used Northbound API are Representational state transfer (REST) API. A REST API often communicates over HTTP and uses a standard representation format like JavaScript Object Notation (JSON), HyperText Markup Language (HTML) or Extensible Markup Language (XML). Other Northbound APIs are programming languages like JAVA or propritary APIs. The East / Westbound API are APIs to communicate between NOS or Controllers SDNi, BGP, AMQP.

OpenFlow

OF protocol, instruction set, architectue => interface[**sdn_Anatomy_of_OpenFlow**]
OF components:

2.1.5 ONOS controller

2.2 Distribution

2.2.1 Techniques

2.2.2 Leader Election

2.2.3 Synchronization

2.3 Federated Networks

2.3.1 Federated Mission Network

[**http://www.act.nato.int/fmn**] Federated Mission Network (FMN) and Joining Memebership and Exiting Instructions (JMEI). Define number plans and protocols for how to connect to each nation. BGP, NTP, SNMP, ... Five service: Each nation has to manually configure their BGP peers before each .

Four leel of capability, Mission Network Element (MNE), Mission Network eXtension (MNX) - self served but may not include sufficient mission essential services, Hosted User, other entities

FMN Affiliate : " NATO and Non-NATO nations are encouraged to become FMN Affiliates, which implies they will maintain and further develop capabilities for federated mission networks and ensure CIS security and interoperability compliance with FMN standards and principles. FMN Affiliates will be able to contribute FMN-ready forces to a mission on short notice and with minimal preparation."

2.3.2 Protected Core Networks

2.3.3 TACOMS

2.4 TACOMS

Tactical Communications (TACOMS) was a multinational initiative with the goal to specify how North Atlantic Treaty Organization (NATO) nations should connect in order to achieve interoperability in military operations on a tactical level. The next section will briefly take you through TACOMS more than 30 years of history, including the different versions of the framework called TACOMS phase 1, TACOMS phase 2 and TACOMS+. In the following sections the overall architecture of TACOMS+ will be explain together with a detailed description of the autoconfiguration specification for TACOMS+. The technical details for the other TACOMS specifications are not relevant in this thesis. I will conclude this section with my own thoughts regarding the design of TACOMS autoconfiguration specification including the advantages and weaknesses.

2.4.1 History

TACOMS was established in 1985 as a multinational initiative by 12 NATO nations [1]. The nations saw the requirement for a new NATO standard for telecommunication due to the rising need for interoperability in the battlefield. [2] [3] [**https://www.tacoms.org/pages/history.aspx**]. In June 2010 TACOMS phase 1 Standardization Agreement (STANAG)s (STANAG 4637 - 4647) were successfully promulgated as official NATO standards [**https://www.tacoms.org/pages/about.aspx**].The STANAG 4637-family define a protocol suite and addressing scheme for the physical layer up to the transport layer in the model in addition it describe how Voice over IP (VoIP) should be handled up to the application layer. However only the Netherlands implemented the STANAG 4637-family TACOMS in their tactical communication system. I think one reason for this was that some of the protocols were at the end-of-life when the STANAGs were finally finalized, like the H.323 signalling protocol which lost terrain against Session Initiation Protocol (SIP). Another reason can be that NATO started their own standardization project for information sharing and interoperability namely the FMN, which was based on the success and lessons learned from Afghan Mission Network (AMN) [4] **??**. .

After completion of the STANAGs in 2010, TACOMS continued as TACOMS phase 2 with a MOU / agreement ending in 2012. With phase 2, the first flavor of SDN or programmable networking were introduced with the Statement of work (SOW) 6 - Flexibility, Agility and Scalability **??**. Some nations withdraw from the project when entering phase 2, due to FMN and other reasons (lost thrust because of the time it took for STANAG 4637 to

[1] Belgium, Canada, France, Germany, Italy, The Netherlands, Norway, Portugal, Spain, Turkey, United Kingdom and United States

[2] TACOMS were not restricted to NATO nations, but included Partnership for Peace nations. Sweden and Finland joined in 2007 when signing the 2nd memorandum of understanding (MOU).

[3] Poland was part of TACOMS for a short period of time from 2003 to 2007 when the 2nd amendment of the MOU was signed.

[4] the operational NATO network in Afghanistan

come to life). By end of the second MOU, TACOMS decided to change the strategy from writing STANAGs to rather contribute to the FMN work. The same year TACOMS phase 1, with some quick wins (like replacing H.323 with SIP), was accepted as the transport layer for FMN. The third MOU was signed and later extended until March 2016 in order to complete the work [5].

(Proof of concept at the Coalition Warrior Interoperability eXploration, eXperimentation, eXamination, eXercise (CWIX) exercise in 2015 in Bydgoszcz, Poland. SDN track within FMN?

)

TACOMS work group finalized their work during March 2016.

Lost many nations when they needed an extended MOU (Italy, Germany while Canada, USA exit earlier) continued 2014-2016

2.4.2 TACOMS+ Technical

Architecture

âWhile implementation of the UNI was national concern, implementation of NIP is coalition concern. Therefore, functions and protocols exchanged across the NIP are standardized in a detail.â âWhat does TACOMS+ provide? TACOMS+ provides the standards that enable seamless interoperability at network domain between differing national networks to effectively create a single, coherent Mission Network. TACOMS+ creates the interface between national networks but does not impact on the national systems or capabilities. TACOMS+ utilises open standards and industry best practice to define the interoperability interfaces and the TACOMS+ standards provide the recommendations and approaches to be taken when implementing those protocols. TACOMS+ is therefore not a bespoke hardware solution, neither is it a system or a stand-alone item of equipment.â [**tacoms_architecture_concept**]

The different areas in TACOMS are covered by a number of profiles. Each profile is documented in one or more documents. The documents have been produced by sub teams of TACOMS, called Statement of Work (SoW), as an answer to the statement of need (SoN) decided by TACOMS PSG for each team.

Area Profile Documents Technical specifications, implementation examples etc Network Infrastructure/ Network services Dual Stack CL Forwarding Routing Autoconnectivity CL Forwarding Auto Connectivity Bearers Transmission Mediation (moved from CL Forwarding) CO communication CO Forwarding Future bearers Bearers —– Voice

CO Forwarding FMN Spiral 1 for Media Addressing Addressing CL Forwarding - Address and Numbering Plan Network Support Services Name Resolution Network support services- Name Resolution Time Synchronization Network support services- Time Synchronization Security Border Protection System CL Forwarding Border Protection System Public

[5] Norway, Spain, Turkey, UK, Sweden, Finland, The Netherlands signed the extension. Italy withdraw, but participated in some of the work

Key Infrastructure Network support services- Public Key Infrastructure Authentication CL Forwarding Authentication

TACOMS includes two different automatic -> âautoconnectivityâ and âservice announcementâ. Autoconnectivity includes discovery peers, mutual authentication and automatic peering of GRE tunnels, BGP routing and multicast routing.

Autoconnectivity

TACOMS autoconnecitivity technical specification [**tacoms_t6_autoconf_techspec**] BGP, GRE, MSDP, The entities are successfully authenticated with IKE over IPSEC. TACOMS autoconnectivity technical specification [**tacoms_t6_autoconf_techspec**] divides the process for autoconfiguration in three phases: discovery, authentication and automated configuration of peers. Phase one and three rely on routing-protocols to share configuration information and a custom software to use this information to configure peers, while phase three rely on IKE over IPsec.

Discovery PRECONFIGURATION: The nodes are preconfigured with a 111.[Entity$_n umber$].$x.y/8 IPv4. Used as source for RIPv2 packets and later as source/destination for the GRE tunn$ $shared key and MD5 authentication Announced 110.[Entity_number].x.y/32 address This network is not used for an$ $tunnel(111./8 address)$.

=> The Discovery phase is used to discover other TACOMS nation by announcing a /32 subnet with RIPv2. With the second octet identifying the entity it is possible to decide who would serve as a master and slave in the preceding steps. MD5 authentication with pre-shared key are used to reduce DoS attacks.

With the extracted source of the RIPv2 announcement it is possible to start configuring a GRE tunnel with IPSEC adding the physical source and destination. The nation with the highest entity number serves as the master and start to configure the logical address of the tunnel. IPv6 is enabled at the tunnel endpoint which will enable RIPng packets to start flowing, hence phase two and autoconfiguration.

The interface meant for TACOMS connectivity are configured with a address where the second and this octet identify the nation.

The spec claims that - âThis document uses the address plan designed by the SoW6-group. The discovery mechanism is not bound to that exact address plan, and can be adjusted to cohere with other address plans. When IPv6 becomes the dominant network layer technology, the entire address plan and the auto-discovery mechanisms should be updated to utilize new functionality in IPv6. â however to identify a configuration announcement / service announcement feature it rely on the on the script / nation software

[**tacoms_t6_autoconf_techspec**]

TACOMS Design choices [**T6-Concept_options**] â...rationale for the design choices of TACOMS SON 6 groupâ = > Autconfiguration. Authentication: Options X509v3 certificates Phase 1 Discovery Protocol: Options Custom protocol and mDNS = not part of COTS feature set. OSPF RIPv2 IPv6 Routing Protocols Phase 2 Service Announcement Protocol Options

Custom protocol IPv4 Routing protocols OSPF for IPv6 RIPng Service Announcement Options DNS UDDI BGP

[**http://blog.ipspace.net/2015/10/survey-vendor-netconf-and-rest-api.html**] Routers Vendor NETCONF Pure XML? REST API Alcatel Lucent Yes

? Brocade MLX Yes

? Cisco IOS / IOS XE Yes Some No Cisco CSR 1000V Yes Some Yes (JSON) Cisco IOS XR Yes

? Juniper Yes Yes Yes Updates: IOS XR has NETCONF; REST API is only available on CSR 1000V, not on IOS or IOS XE in general; Juniper MX routers have REST API in Junos release 15.1; Brocade MLX routers and ALU routers have NETCONF support;

—DRAW PROTOCOL in VSD—-

RIPv2 - discovery (with pre-shared key) RIPng - autconf (Over GRE tunnel GRE w/Ipsec - Authentication

2.4.3 TACOMS+ Thoughts

Advantages

Rely on existing standards Except the software / script that are triggered by an autoconf event, all the code is configuration of existing standards Introduces agility and flexibilty to FMN in a secure (?) manner Tested Though at every exercise something failed

Critique

TACOMS autoconfiguration has a very âdirtyâ design, this is due to the inflexibility to future changes, mixing of concerns, highly coupled and dependent, duplication of functionality and lack of abstractions.

TACOMS encapsulate information in IPv6 addresses to share configuration information,. An IPv6 address is 128 bits long, where the first 32 bits are used to identify the type of information being encapsulated. This leaves 96 bits for configuration information, and in most cases they use additional 16 or 32 bits to identify the entity sending the information. Even though TACOMS has proved that this information serves the purpose for configuring both , and Border Gateway Protocol (BGP), it makes the protocol inflexible to future changes. What if FMN decides to change the routing protocol? What if the length of the telephone number i changed, exceeding the length of an IPv6 address? With the method of encapsulation information in IPv6 addresses TACOMS makes the protocol inflexible to future changes in other part of the framework. (makes it also dependent upon IPv4 - how do you encapsulate a IPv6 within IPv6???). To make things happen faster they have to tweak the timers of the , and BGP. = > Now 60 sec for peering. Dependent upon the addressing scheme

BGP is both used for announcing regular IPv6 routing information and TACOMS configuration information. The Service announcement routes are fake and not routable. TACOMS specify these routing processes to be separated, but this will actually in many cases mean an additional bgp

router or software router (Cisco, as example, only support one bgp routing instance in each , and one tunnel can only have one ...)

TACOMS rely on for discovery, for sharing peering information and BGP for service announcement. Even though these protocol have been around for a while and would most likely not change in the future, it is not a good design principles to rely on these protocols. Tacoms software would Much static / pre-configuration that needs tweaking / workarounds of a regular router to work Since there are so many protocols involved in the autoconfiguration process there is a need for a lot of preconfiguration/static of the IOP. Some of the pre-configuration needs workarounds (eks when having more than one interface means having the same ip subnet for multiple interfaces on a router) because it is not standards procedure. Injecting non-existing network addresses by adding loopback addresses.

Lack of abstraction => Not possible for a collaborative software development, and reuse of software. The implementation is highly dependent on the underlying HW / SW implementation.

Many protocols used for other purposes than its original design TACOMS uses RIPv2 for discovery, but it is not the announced address which is of interest but the source of the routing information. RIPng to share GRE tunnel address, Multicast information link-by-link (not redistributed BGP IPv6 for service annoucement (encoded by FD00:<serviceId>:<NationId>:<NationDomain>:) TACOMS also specify regular IP v4/v6 forwarding with BGP and demands that the nation uses two different routing instances, one for forwarding and one for SA. Not consistent - in some situation you only uses the announced ip address and other also the announced next-hop address. Some are redistributed depending on the service ID. => Has to âinjectâ ipv4/ipv6 addresses that are not addressable Latency of convergence because of timeout and trigger interval of the different protocols Rely on pre-shared certificates Maybe not possible to avoid if you will keep it secure Error prone conversion of IPv6 addresses Both hex-bin, hex-decimal in the same operation Depending of the prefix length and the announced service, what part of the address which are hex-bin and hex-dec are changes. Depends on the prefix mask and how the ipv6 address is represented (if aggregated with :: for 0000 for example) Inconsistency of address allocation Master node is the node with the largest entity number, and is the one deciding the interface addresses.

BGP to share information. SOW6-8. Delivered the first STANAG TACOMS phase 1 in 2010. TACOMS+ next phase. Terminated March 2016. IPO opened in 1998 [**tacoms.org**]. 8 Statement of Needs. Architecture, Phase 1 QuickWins, Reference implementation, Support to NATO FMN, Security, Agility and Flexibility, CO Services, Future IOP Bearers and Interfaces. Based on standard routing protocol to share information. RIP, RIPng and BGPv6.

Part II

Implementation

Chapter 3

Implementation

3.1 Design

3.2 Architecture

3.3 Functionality

Chapter 4

Evaluation

4.1 Security

4.2 SDN controllers

4.3 Future work

Part III

Conclusion

Chapter 5

conclusion

[10], [7] [15], [8], [**marschke_software_2015**], [**sdn_2013**], [14], [**_software_2014**], ,[12], [13]