FFI-rapport 2013/00529

Multi-Topology Routing

QoS functionality and results from CoNSIS field experiment

Margrete Allern Brose, Mariann Hauge, Jan Erik Voldhaug and Jostein Sander

Norwegian Defence Research Establishment (FFI)

11. februar 2013

FFI-rapport 2013/00529

1175

P: ISBN 978-82-464-2234-3
E: ISBN 978-82-464-2235-0

Keywords

Nettverksbasert forsvar

Taktisk kommunikasjonsnett

Mobilt ad hoc nett

Tjenestekvalitet

Internett protokoll (IP)

Ruting

Heterogent nettverk

Approved by

Torunn Øvreås	Project Manager
Anders Eggen	Director

English summary

This report describes the major contributions of FFI project 1175 to the multinational research project CoNSIS (Coalition Networks for Secure Information Sharing). Norway, France, Germany and the United States participated in the project which focused on efficient and secure information sharing between coalition partners using mobile and deployed networks.

We report on mechanisms that can support interconnection of radio networks from different units and different nations. We also describe mechanisms that can improve network resource management and support for differentiated quality of service for traffic in disadvantaged networks. All mechanisms described in this report support IPv6.

As part of the Norwegian contributions to CoNSIS a demonstrator was delivered. This demonstrator is based on the Intelligent Tactical IP router. In this report we show the improved functions and their use in heterogeneous mobile military networks. We also report on the performance of these mechanisms in the multinational field test staged by CoNSIS.

Samandrag

Denne rapporten skildrar arbeid som FFI-prosjekt 1175 har gjort for det multinasjonale forskingsprosjektet CoNSIS (Coalition Networks for Secure Information Sharing). Noreg arbeida saman med Frankrike, Tyskland og USA i CoNSIS. Prosjektet hadde som mål å utvikle løysingar for effektiv og sikker informasjonsutveksling mellom koalisjonspartnarar på taktisk og stridsteknisk nivå.

Me visar i denne rapporten til løysingar som kan hjelpe til å knytte saman ulike mobile nett basert på ulikt radioutstyr frå ulike nasjonar, og dermed bidra til informasjonsutveksling mellom forskjellige militære einingar og nasjonar. Me skildrar og mekanismar som kan gje ei betre utnytting av resursane i mobile nettverk, og som kan støtte ein viss grad av differensiert tenestekvalitet for trafikk i nettverket. Alle løysningane til CoNSIS støttar IPv6.

Som ein del av arbeidet med CoNSIS vart det utvikla ein demonstrator. Demonstratoren er ei vidareutvikling av Intelligent taktisk IP-rutar, som er utvikla i samarbeid med Thales Noreg og Forsvaret. I denne rapporten forklarar me korleis ulike typar funksjonalitet i den taktiske rutaren best kan brukast i eit militært mobilt nettverk. Me rapporterer også frå den multinasjonale felttesten som vart arrangert av CoNSIS, og oppførselen til ruterdemonstratoren vår i denne testen. Mykje av denne informasjonen er også presentert i ein CoNSIS task1 rapport.

Contents

1 Introduction

New ways of operating require information exchange between units that traditionally do not have much interaction. Multinational operations also require efficient information exchange between coalition partners. In order to establish networks that support efficient end to end communication in such settings, we need mechanisms that can handle highly heterogeneous networks (networks deployed with incompatible equipment from different units or nations). In current and previous Norwegian Defence Research Establishment (FFI) projects we have focused on this topic. In [1] we describe two router demonstrators with mechanisms that focus on heterogeneous networking. One of these router demonstrators (Thales Intelligent Tactical IP Router) has been extended with new functionality. This improved demonstrator was a deliverable from FFI project 1175 to the multinational research project CoNSIS (Coalition Networks for Secure Information Sharing). This report describes the current functionality of the router demonstrator and its performance in the multinational CoNSIS field test. The functionality and the test results have also been reported in a CoNSIS task1 report.

Norway participated in CoNSIS along with France, Germany and the USA. The terms of the project are given in a memorandum of understanding (MoU). The CoNSIS MoU states: "The objective of the project is to design, implement, test and demonstrate technologies, methods and architectures for the secure sharing of information and services between nations in ad-hoc coalitions, and between military systems and civil systems for Civilian Military Cooperation e.g., with Non-Governmental Organizations (NGOs), within the communications constraints of mobile tactical forces."

The work in CoNSIS was organized in five tasks:

- Task 1, Communication Services
- Task 2, Information and Integration Services (SOA)
- Task 3, Security
- Task 4, Management
- Task 5, Architecture, Test and Demonstration Coordination

More information on the CoNSIS project can be found in [2].

The technology and tests described in this report represent some of the work that has been performed in Task 1, "Communication Services". In this task our concern was to provide a transparent network and information infrastructure (NII), based on and harmonized with IP technology. The focus has been to demonstrate solutions that will work within the communications constraints and dynamic topology imposed by highly mobile tactical networks. It was required that the proposed mechanisms could support IPv6. [3] gives an overview of all the work done in task 1.

This document reports on a mechanism that has been used in the CoNSIS land mobile network to solve two network challenges:

1. Integrate different radio networks and satellite communication links present in a coalition operation, into one common transport network.
2. Provide some support for differentiated services and network resource management in the heterogeneous coalition network described above.

The rest of the report is organized as follows: In chapter 2 we give the background and motivation for our work. We point the reader to related work in chapter 3. In chapter 4 we describe the Multi-Topology routing solution and the mechanisms proposed to connect a Multi-Topology routing domain with a Single-Topology routing domain. The QoS architecture is explained in chapter 5. In chapter 6 we describe the CoNSIS convoy test network and its unicast and multicast support. The field tests and results are presented in chapter 7. Finally we give a short conclusion and some lessons learned in chapter 8.

2 Background

In a coalition operation the participating nations will typically bring their national radio equipment into the theater. Usually the equipment will consist of a wide range of technologies and products from different vendors, reflecting the normally long lifetime of military radio systems. These various radios will most likely not be compatible on the air, and if they are, they will probably not use compatible network protocols, security solutions, management or services for the end user. The main goal of the CoNSIS project is to propose and demonstrate mechanisms that enable secure information sharing despite of these interoperability issues. CoNSIS proposes solutions to improve interoperability in all the above mentioned areas. This report describes the Multi-Topology routing concept as used by CoNSIS, to glue available networks together and to provide differentiated Quality of Service (QoS) in land mobile networks that utilize many different transmission technologies for internal communication, as well as reach-back to the deployed headquarters.

To provide a reliable network for different operation types and in varying terrains, a tactical mobile network infrastructure must consist of a variety of wireless network types, e.g., long-range communication for reach-back connections, and a higher bandwidth network for local communication. A single transmission technology, e.g. a VHF network, will not be able to support all communication types and bandwidth requirements. This combined with the fact that the different nations usually bring national radios manufactured by different vendors to the battlefield result in a situation with a large number of different, non-compatible radio systems present in the mission network. The aim of Task 1 is to be able to combine all available radio systems in an operation to provide an efficient, common network for coalition use. This gives the operator a single entry point to the complete heterogeneous coalition network. A common network will be better utilized, and multiple transmission technologies and routing paths will also improve the network reliability by providing alternative routing paths during e.g. jamming attempts. The resulting coalition network will consist of radios which have large variations in properties such as transmission capacity and range. It is however challenging to administer, admit, and route traffic flows in these networks.

In a mobile tactical network there will in most cases be limited capacity. It is therefore crucial to support prioritization of mission critical traffic. It is also desirable to use the tactical network in the most optimal manner, and thus make sure that only traffic that has a high chance of reaching the destination is admitted into the network. One way to increase the network throughput is to take advantage of parallel paths in the heterogeneous network and efficiently exploit all bandwidth resources.

Since the transmission means used in tactical networks have large variations in capabilities, CoNSIS finds it advantageous to define multiple routing topologies in the network in order to support different QoS classes. These topologies are then used to ensure that data packets are only forwarded on topologies with sufficient capacity to support the requirements of the dataflow. In this report we describe an architecture where we combine Multi-Topology routing (MT-routing) [4][5] and traditional DiffServ-like [6][7] mechanisms to utilize all available transmission means in the tactical network and increase the robustness of the network. We name this design “MT-supported QoS architecture”. In this report we describe how this architecture is used in the land mobile CoNSIS network and how we have solved the interaction between a network running MT-routing and adjacent networks running non-MT capable domains. An extract of this report is presented in [8].

2.1 CoNSIS scenario

As part of the work in CoNSIS, a scenario that takes place in a country torn by civil war has been defined. An international coalition is involved in this conflict to protect civilians and initiate the peace process. The coalition has a land based component, a naval component and an air based component.

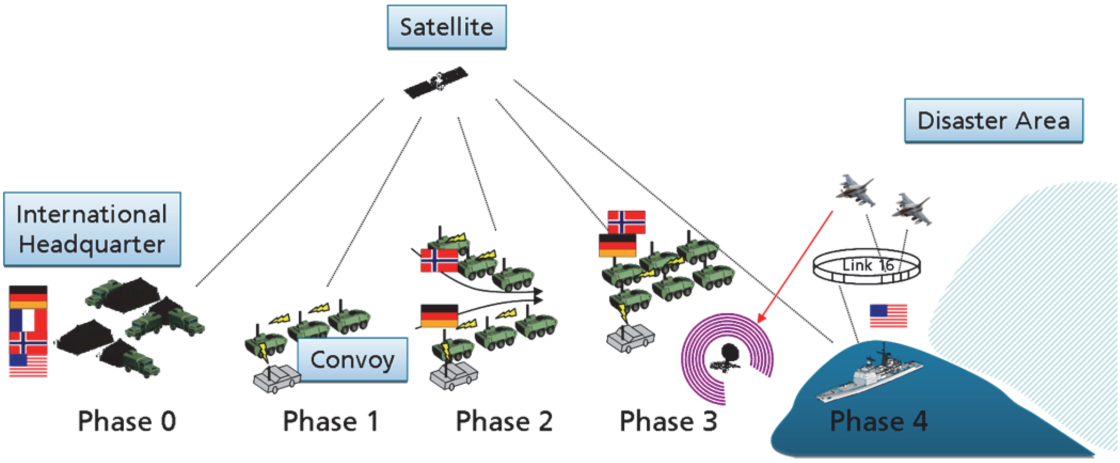

Figure 2.1 The CoNSIS scenario in a nutshell

In the scenario a natural disaster strikes in a rural area outside the control of the coalition. The coalition decides to establish a coalition convoy to escort a number of NGO vehicles to the disaster area. The mechanisms described in this report are used to improve network communication in the coalition convoy. Figure 2.1 describes some of the events in the scenario. More information about the scenario can be found in [9]. In this report, the network deployment used for the convoy in the CoNSIS field test exemplifies the use of MT-routing in CoNSIS.

2.2 The network reference model

The CoNSIS network reference model is similar to the Protected Core Network (PCN) [10] architecture. In CoNSIS we wanted to build our network mechanisms around the PCN model; however we did not want to be bound by the current PCN description and interface. We therefore gave our own names on the PCN network components to avoid conflicts. The CoNSIS model (Figure 2.2) describes the transport network as a set of Transport Network Segments (TNS). National TNSs (N-TNS) are managed by a nation while Coalition TNSs (C-TNS) are managed by the coalition. In order to provide the necessary protection of the user data, traffic is encrypted by an IPSec crypto device prior to leaving the Colored Enclave (CE). It is possible to have a CE inside another CE; this encapsulated CE is then called the Inner CE (ICE).

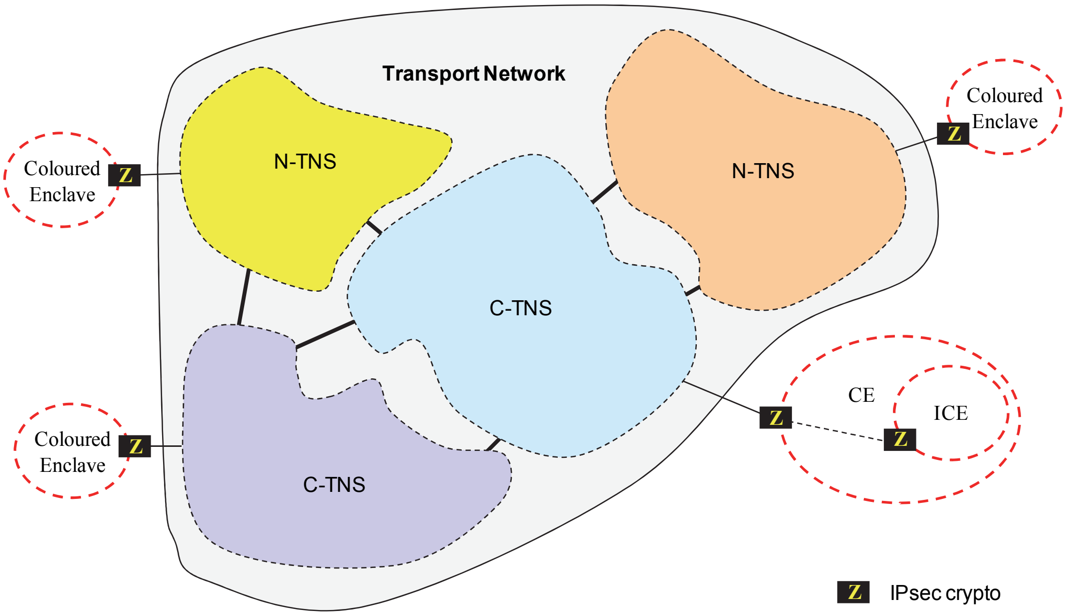

Figure 2.2 The CoNSIS network reference model.

The Land Mobile Network which is the target for the mechanisms described in this report will be a C-TNS in the reference model. Each platform (vehicle) in the network will have one or more CEs attached to the TNS.

3 Related Work

During the last 10 years a lot of research has been done to achieve predictable QoS in mobile ad hoc networks (MANET). This is a difficult task due to the agile changes in the network topology, and the fluctuating channel quality in such networks. Much focus has been put in the area of QoS-routing. QoS-routing aims to find a route which provides the required service quality for a specific traffic type. This can be done using routing metrics based on parameters like delay, data rate, signal to noise ratio, route stability, etc. Such protocols must be combined with a resource manager and a traffic classifier (e.g., DiffServ-like classification) to support end-to-end QoS in the network. Two survey papers [11][12] give a comprehensive overview of many of the available QoS-routing proposals.

Most of the QoS protocols covered in the two survey papers discover a single path that supports a certain QoS requirement. This QoS requirement can be described by one parameter (e.g., maximum bottle-neck data rate), or by several parameters (e.g., maximum bottle-neck data rate and lowest end-to-end delay). Some protocols also maintain multiple paths to the destination for the purpose of e.g., load balancing, fault tolerance, higher aggregated bandwidth and reduced route discovery latency after link breaks. In [13] important multipath protocols are covered. In [14][15][16][17] multipath is established explicitly for QoS support. Some of these also make a point of combining DiffServ and multipath routing.

However, most of the QoS-routing schemes, and all the mentioned multipath protocols are reactive routing protocols. We believe proactive protocols will be necessary in tactical MANETs to reduce the routing response time and increase the predictability of the network availability. We also think it is beneficial to store several routes with different characteristics to support separate QoS requirements. This is important for a heterogeneous wireless network that is established with radios that utilize different transmission technologies.

The MT-supported QoS architecture that we suggest for the land mobile network in CoNSIS is based on the proposal presented in [18] and further studied in [23]. It is a simple but powerful scheme with a proactive routing protocol that maintains multiple topologies in the routing domain, and consequently provides multiple paths from source to destination. Each topology/path is associated with a single, or multiple QoS class(es). Similar ideas (based on a very different routing scheme) are presented in [19]. In this reference, network information is maintained proactively, and different paths for the required QoS classes can be calculated with different metrics based on a single routing database.

In [20] MT-routing is combined with a dynamic topology and traffic pattern analysis tool to provide a flexible load balancing solution. In [21] MT-routing is utilized in a satellite network, both for fault tolerance and for traffic separation of traffic having different QoS requirements. Both of these papers exploit a similar technique as the one presented in this report. The main difference is that our focus is to support admission control and efficient resource utilization in a very heterogeneous military mobile ad hoc network. In [22] we report on a national field experiment with the first version of the MT-router. In [23] we presented our findings when using this technique on an isolated test bed network in our lab. The MT-supported QoS architecture was also utilized by the Web Services admission control broker in [24]. The software (SW) for the MT-router has been extensively modified for the CoNSIS project to provide better support for IPv6 and to allow interaction between Single Topology-routing domains and the MT-routing domain.

4 Multi-Topology Routing Architecture

4.1 Multi-Topology routing

A traditional link state routing protocol maintains one routing table with one entry for “the best route” to all destinations in a network domain (or several of the best routes for load balancing purposes). The best route is calculated based on the chosen metric, e.g., shortest path first (SPF) or lowest cost, where the cost parameter can be established based on any set of link parameters.

A Multi-Topology routing protocol maintains several topologies within the network domain at the cost of a few extra bytes in the routing packets. Each topology spans a subset of the physical topology. A shortest path first calculation (other metrics can be used if available) is performed for each topology to discover the best routes within the topology. The cost of one link can be set different for the different topologies. Only the links belonging to the actual topology are included in the calculation. The results of each SPF calculation are stored in one forwarding table for each topology. In Figure 4.1 we show a network where three topologies are defined on top of the physical topology. A number of topologies can be defined on a single physical link. All the physical links in the domain must be part of the default topology. The default topology is used for routing traffic and ensures that routing information is flooded to the whole network.

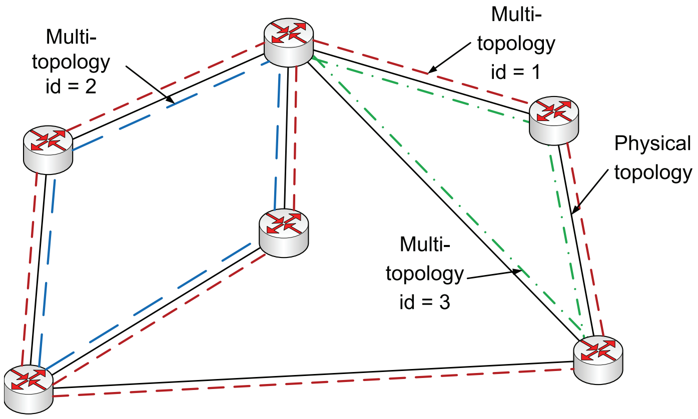

Figure 4.1 Network with three different topologies.

During network configuration, topologies can be tailored to represent many different purposes. MT is used for the following cases in CoNSIS:

- Topologies can be created that have sufficient (maximum) resources to support a certain QoS class, or multiple QoS classes.
- A specific topology can be created to be used for external traffic into the network and transit traffic through the network.

MT-routing is a powerful tool that can be used to solve many situations where a certain end-to-end behavior is needed in tactical networks. This comes at the cost of a small signaling overhead and more complex network configuration.

The protocol operation of OSPFv3-MT is similar to OSPFv3. After the routers have formed adjacencies with their selected neighbors, and the Hello-protocol has been initiated, link state information is flooded in the OSPFv3 area. Most link-state advertisements (LSAs) include information about the link cost, IP address and subnet mask. To avoid problems with backward compatibility, a set of new LSAs has been defined in [4] for MT-OSPFv3. In the new LSAs, the entry of each interface is defined in a type-length-value field (TLV). One example is the new

Link Description TLV (LD-TLV). We describe the structure of the LD-TLV here. Other LSA types are coded in a similar manner. The LD-TLV holds a set of sub-TLVs called Router Multi-Topology sub-TLV (RMT-sTLV) (ref Figure 4.2). The RMT-sTLV carries the Multi-Topology identifier(s) (MT-id) for each neighbor. One link may belong to multiple topologies; this requires multiple advertisements with an MT-id and an MT metric per topology. Still, only a single adjacency is formed with each of the selected neighboring nodes even if the interface participates in multiple topologies. The same link may have a different MT metric for each of the topologies it participates in. All link advertisements are stored in the link-state database. The calculation of the forwarding table for each topology is based on the information in this database.

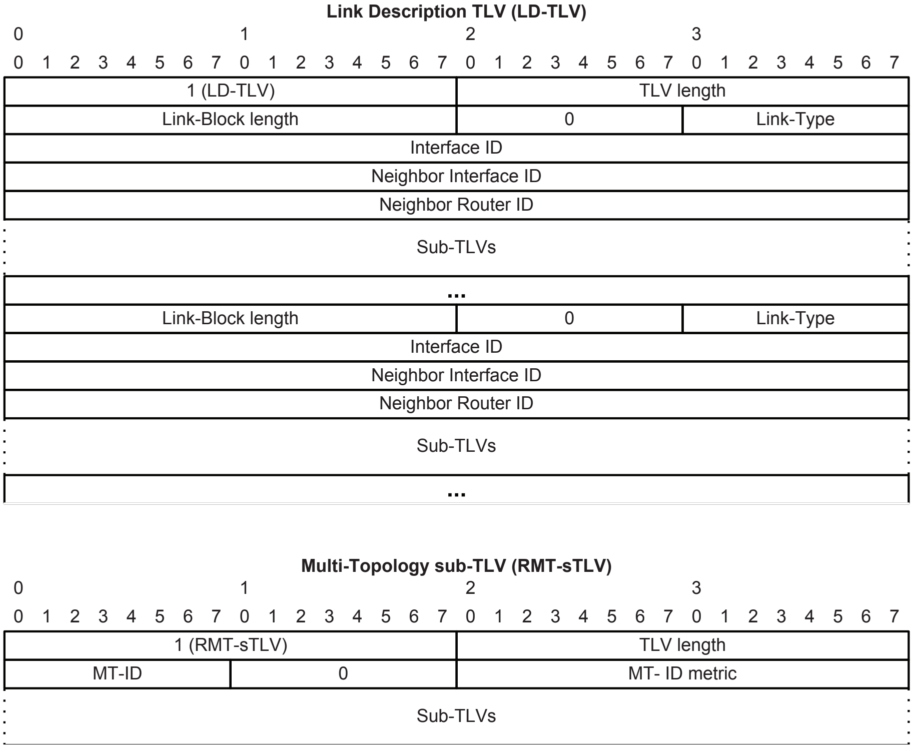

Figure 4.2 The new TLV/sTLVs for MT transport in OSPFv3.

In order to prepare the router for standalone multi topology routing, the following LSAs were created with multi-topology TLV entries:

- E-Router-LSA
- E-Network-LSA
- E-Inter-Area-Prefix-LSA
- E-Inter-Area-Router-LSA
- E-Link-LSA

The MT-routing RFC [4] specifies MT for OSPFv2 and the draft [5] specifies MT for OSPFv3. In CoNSIS we use MT-routing in a MANET environment with high frequency of link breaks. Standard OSPFv3 is not the best suited protocol for this network type. In order to make the MT-routing protocol better suited for MANETs we therefore implemented MT-routing also for one of the suggested mobile extensions to OSPFv3 (RFC 5614 [25]).

4.2 Interaction between a Multi-Topology routing domain and a Single-Topology routing domain

The MT-routing draft and RFC [4][5] both describe interaction with Single-Topology routing (ST-routing) through the default topology *main* (designated table 0 in MT). We do not view this approach as suitable for a mobile military network. The main reasons for this are:

- The default topology covers the entire network and does not take into account transmission characteristics for the respective links.
- For IPv6 the routing protocol load would be close to doubled, since the layout structure of the MT LSAs are incompatible with standard IPv6 OSPF. In order to obtain compatibility with ST-routers, the MT capable routers have to transmit both encodings.

Furthermore, there exists no description of how to import routing information from an adjacent ST-routing protocol into the MT-routing protocol, without using the default topology. This can be regarded as a weakness in the specification, since it will only be the high capacity topologies of the MT-routing domain that are usable for connection with external ST-routing networks. The default topology normally does not have the ability to differentiate between traffic. In the CoNSIS project we wanted to have the interaction both between the MT-routing protocol and an exterior gateway protocol (EGP) as well as an interior gateway protocol (IGP).

First we consider the task of importing and redistributing routing information from an adjacent ST-routing protocol into the MT-routing protocol. Most ST-routing protocols maintain routing information in the main forwarding table (known as table 0 or default topology in MT). To avoid conflicts the default topology should not be used by the MT-routing protocol when MT-routing is used for QoS purposes. According to RFC [5] tables 32 to 127 are reserved for development, experimental and proprietary features and can be used for our purposes.

The adjacent network information that we want to redistribute in the MT-routing network may have very different characteristics; it can be a homogeneous radio network with certain characteristics or it can be a deployed network with different typical characteristics. The radio network we might want to import into one or more specific topologies, whereas the deployed network should be imported into all topologies. For this reason we wanted to make a very flexible solution that allowed us to specify network import into (none or) any number of topologies. This involves both redistribution of the adjacent ST-routing protocol information into the different topologies, and a copy of the ST-routing information made available to the MT-routing forwarding tables. Since redistribution only provides the routing information to neighboring nodes and not to the unit itself, this has to be a copy.

If several networks are connected to one gateway MT-router and we want to redistribute the information from these protocols to different topologies, then these networks should use different routing protocols. If different routing protocols are not used, then the MT-router will have difficulty separating the routes made available from network 1 from the routes made available from network 2. It might be possible to use route-maps for each topology to separate routes maintained by different instances of one routing protocol. We have not tested this.

Next we consider the task of making routing information from the MT-routing protocol available for adjacent networks. Here we would also like to have the same flexible solution of providing information from (none or) any number of topologies to the ST-routing protocol. In practice this means to provide the union of the routes available in the relevant MT-routing tables to the ST-routing protocol. In theory this flexibility is available in the current SW, however the configurations we have tested and validated involves making routing information from 0 or 1 of the available topologies available to the routing protocol in connected networks.

It should be noted that some planning is necessary to use the flexible mechanisms for import and export of routes in the best manner. One should be careful not to (by accident) import routing information from a network (e.g., network 1) into other topologies than the one that is made available for redistribution into network 1. If this mismatch happens there will be asymmetry in the network routing information and some traffic will only be able to flow one way. However, in some cases this mismatch in routing information can be the correct configuration. E.g., in a QoS architecture there could be a policy saying for example that ST-routing networks should be given the same or more routing information than the MT-routing network. Traffic with QoS tags that cannot be supported by the current MT-routing network will then be dropped at the entry point to the disadvantaged mobile MT-routing network.

As a special case we gave the interaction between the MT-routing protocol and BGP [26] some extra thought. Providing the routing information in one topology for redistribution in BGP limits the visibility of the MT-routing network for BGP connected networks. This method can be used to provide a topology for transit traffic through the MT-routing network and make the complete MT-routing network available only for local traffic.

The following LSAs had to be added to the base MT-routing protocol to support redistribution of routes from external protocols and support for multiple areas in OSPFv3:

- E-AS-External-LSA
- E-Intra-Area-Prefix-LSA

5 QoS Architecture

The CoNSIS QoS architecture for the network layer in the land mobile network divides the QoS operations in two functional entities:

- One entity that supervises the network resource management. This mechanism is needed at the ingress of the network.
- One entity that handles network congestion, packet forwarding and packet prioritizing required by the different data flows. This mechanism is needed in all forwarding elements in the network.

The resource management entity decides if a new traffic flow can be supported by the network. This mechanism must identify the network resources required by the flow associated with a specific QoS class. If the routing path is classified to be able to sustain the traffic type, the session will be admitted. Thus, there is a need for a resource management mechanism that attempts to estimate the available capacity of the network. If mechanisms are available to support resource reservation, this will be done by the resource manager.

The prerequisites for admittance of a flow may change after the flow is admitted. A session of very high importance may try to access a fully loaded network. Pre-emption of a low importance session may then be required. Similarly, due to node mobility, jamming, etc., the network capacity may change over time. This must be acted upon by the resource manager.

Short term network congestion due to fluctuations in the radio channel capacities and temporary overload of the network must be handled by the forwarding component of the network routers. This component must also tailor packet queues and packet scheduling to effectuate the delay requirements of the packet's QoS class, and the military priority of the packet. In overload situations this mechanism makes sure that the important traffic is prioritized by the network at the expense of less important traffic which might then experience a very high packet loss due to queue overflow.

For this architecture a set of QoS classes must be defined that describe the network requirements (in terms of data rate, jitter, delay, reliability, etc.) needed by the dataflow labeled with the specific QoS class. The traffic flows must be tagged with this information.

In CoNSIS we propose to use MT-routing to support the entity that supervises the resource management of the network. In the MT-supported QoS architecture, we configure and maintain several network topologies that each spans a subset of the physical topology. Each topology has its own forwarding table that is used to forward packets classified as belonging to that specific topology. If a destination address is not available in the forwarding table associated with the QoS class, then no path exists in the network where the specific QoS class is allowed to be transported. Thus the flow should not be admitted to the network. Traffic is stopped at the network edge and not (in a worst-case scenario) forwarded through the entire network, just to find that the last hop to the destination is a link not able to support the flow's QoS requirements.

When there is a route to the destination in the correct topology and the traffic flow is admitted to the network, the DiffServ mechanisms come into play. A queue hierarchy and packet scheduling mechanism prioritizes the sequence of transmitted packets on each interface. For each network interface we also define a traffic shaper, whose purpose is to keep the traffic transmitted on each link below a certain threshold, to avoid network congestion. We use queue and scheduling tools to tailor the queue to the requirements of the associated QoS class, and to implement packet scheduling for traffic priorities. Queue length, head/tail drop and drop-precedence are important queue parameters, while the packet scheduler could be designed for a strict priority scheme or a situation with more fairness in the scheduling process.

6 CoNSIS Convoy Test Network

The CoNSIS test network consisted of several components, (see Figure 6.1) The MT-routing mechanisms were deployed in the land mobile network component, and used for some QoS support and simple admission control in a very heterogeneous mobile network.

The CoNSIS network was configured according to the “Addressing and Naming Plan” [27].

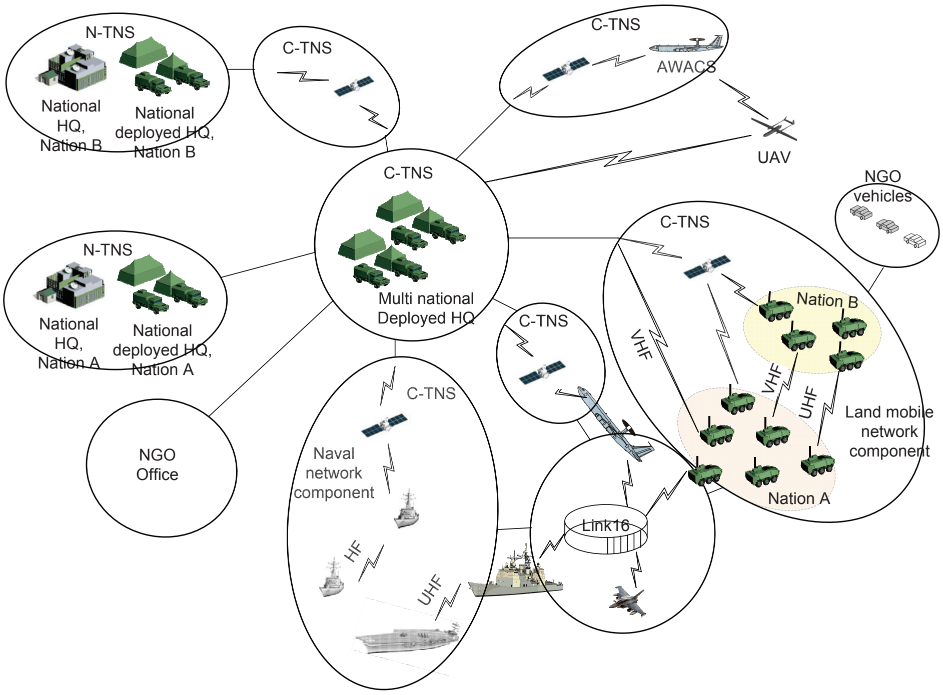

Figure 6.1 The network elements that participate in the CoNSIS scenario [1].

6.1 MT- routing SW

We have implemented the Multi-Topology support for OSPFv3 and OSPFv2, as well as “MANET OSPFv3 MANET Designated Routers (MDR)” into the Vyatta 6.3 (Napa version) [28] Linux distribution. This is based on the Quagga [29] open source routing application running on a Debian system with Linux kernel 2.6.37 (ATOW). The MANET OSPFv3 base protocol was fetched from [30]. The router implementation allows easy configuration of OSPFv2-MT and OSPFv3-MT information. Metrics can be set up for each topology on each interface. The Linux platform is set up to utilize multiple forwarding tables, and Quagga's interface towards forwarding tables in Linux has been adjusted to allow the use of multiple routing tables. In addition to OSPFv2-MT and OSPFv3-MT routing, the implementation also supports configuration of static MT-routes. A flexible import and redistribution of routes from other routing protocols via the *main* routing table is supported, as well as customized export of MT-routes to the *main* routing table to make the routes available for redistribution in other routing protocols.

Due to experienced instabilities in the MANET OSPFv3-MT protocol (RFC 5614), OSPFv3-MT was used in the CoNSIS field experiment.

It should also be noted that the expanded encoding of the OSPF Options described in the draft [4], is in conflict with bits allocated by OSPF Link-Local Signaling [31]. Link-Local Signaling is also part of the MANET OSPFv3 implementation.

6.2 MT-router configuration and issues

The configuration of the MT-router in Vyatta is well described in the Vyatta documentation [28], and in the addendum written by Thales Norway AS for the MT-routing configuration [32]. In this chapter we give some insight into critical configuration parameters that are needed in order to create a multi topology routing environment that can be used to support the QoS architecture.

In the QoS architecture supported by MT-routing we associated a set of QoS classes with a specific routing table. Part of the concept is that traffic should be blocked if there is no route to the destination present in the chosen routing table. For this design it is therefore important to ensure that only the specified routing table is chosen as the forwarding table and not allow routes in *main* routing table or other QoS routing tables to be used. In Linux the routing tables are ordered according to priority. If there is no rule defined that associate an incoming packet to a certain forwarding table then the table with the lowest numerical priority is chosen. If there is no route to the destination in the chosen forwarding table, then the next table in prioritized order will be checked, and so on until a route is found or there are no more routing tables to try. For the MT-supported QoS architecture to work, it is important that only the chosen table is used for route lookup.

We solved this with three techniques:

1. IPtables [33] was used to make rules to associate a label (mark) with a packet. One label also exists for each topology forwarding table. To bind these two operations, the label associated with a specific traffic class must be the label of the topology table that should be used for that traffic class. Linux only allows packets with a matching label to do lookup in a routing table that has an assigned label. All packets can do lookup in tables without labels assigned (Catch-all).
2. The main routing table was given very low priority (a high number). The other routing tables were given higher priority. The mutual order of the higher priority tables was not important.
3. A default black hole entry was configured in each routing table to block traffic to destinations that did not have a routing entry in the routing table. A black hole route is a routing table entry that goes nowhere. This entry captured traffic that could not be routed via the other entries in the routing table and dropped this traffic. This enforced that only one forwarding table (the one with the correct label) was used to look for routes to the destination for a packet.

Example to show how traffic with the Type of Service (TOS) tag 0x28 is marked with the label 0x21:

```
Ip6tables -A FORWARD -t mangle -m tos --tos 0x28 -j MARK --set-mark 0x21
Ip6tables -A PREROUTING -t mangle -m tos --tos 0x28 -j MARK --set-mark 0x21
Ip6tables -A POSTROUTING -t mangle -m tos --tos 0x28 -j MARK --set-mark 0x21
Ip6tables -A INPUT -t mangle -m tos --tos 0x28 -j MARK --set-mark 0x21
Ip6tables -A OUTPUT -t mangle -m tos --tos 0x28 -j MARK --set-mark 0x21
```

Example to show how traffic with a specific label is associated with a specific forwarding table:

```
ip -6 rule add fwmark 0x21 table 33 prio 10
```

Example of forwarding tables in an MT-router in the field experiment:

```
root@NOR3:~# ip -6 rule
0:         from all lookup local
5:         from all fwmark 0x20 lookup 32
10:        from all fwmark 0x21 lookup 33
15:        from all fwmark 0x22 lookup 34
32000:     from all lookup 99
32766:     from all lookup main
```

In the example above we have explicitly created four forwarding tables in the table range reserved for experimental use. Our tables are numbered 32, 33, 34 and 99. The first three tables are associated with a mark meaning that only packets labeled with the same mark can use the forwarding table. Table 99 is a table that catches all traffic and is used for routing, management and traffic that has not been labeled with a QoS label. Table *main* is given the lowest priority in our configuration, and hence no traffic will ever use this table for routing decisions. The following example shows how the configuration looks like in the Vyatta configuration file, to create the above forwarding tables:

```
 topology 32 {
    name low-bit-rate
    priority 5
    target ipv6-only
    traffic-class 0x48 {
    }
    traffic-class 0x50 {
    }
}
topology 33 {
    name high-bit-rate
    priority 10
    target ipv6-only
    traffic-class 0x28 {
    }
}
topology 34 {
    name low-delay
    priority 15
    target ipv6-only
    traffic-class 0xb8 {
    }
}
topology 99 {
    catch-all
    name base-topology
    priority 32000
    target ipv6-only
}
```

The following example shows the Vyatta configuration of the black hole entry in one of the forwarding tables:

```
protocols {
    static {
        table 32 {
            route6 0::0/0 {
                blackhole {
                }
            }
        }
    }
}
```

In addition to the QoS topology tables, we need a forwarding table that represents the complete network topology, our base topology. We could have used the *main* table for this, but in our architecture the *main* table had to be reserved for a special role. We needed the *main* table to function as a repository for routes to redistribute between connected routing domains. More information about the redistribute mechanisms is given at the end of this chapter. Due to the special use of the *main* table we therefore created an explicit topology to hold the base topology. This topology was given a lower priority (higher number) than the QoS topologies to make sure that the QoS topology tables were used to forward traffic with the associated traffic class label.

Topology 99 in the example Vyatta configuration above holds the base topology. This topology is used for routing traffic, management traffic, etc. In the CoNSIS experiment we also chose to use this forwarding table for all traffic in the network that did not have a QoS tag in the Traffic Class field (the `catch-all` command specifies this). Such a decision should be taken with care since with this configuration all best effort traffic and traffic that is not controlled according to the QoS classification of the network will use this topology. The base topology holds all links and thus also very low capacity links that do not have enough capacity to handle much best effort traffic. For operational use a different configuration must be used, or a strict admission control scheme should be in place.

The multiple forwarding tables were populated by the OSPFv3-MT routing protocol, this made sure that all available routes to remote destinations were present in the forwarding tables. However in the CoNSIS architecture we assumed that all servers and clients were connected to a Local Area Network (LAN) attached to the MT-router. The LAN is a *directly connected network* for the router. Information about directly connected networks is written to the *main* routing table. No mechanism was available in the Vyatta router to specify that a *directly connected network* should be visible in other forwarding tables than the default *main*. In initial tests of the MT-router we observed that packets were routed correctly to the final router, but dropped at this router since no route to the *directly connected* LAN that hold the destination address was available in the specified MT-routing forwarding table. To fix this, Quagga's Zebra module was modified and a new configuration statement for configuration of interfaces was added to Vyatta. This made it possible to specify which topologies to make the *directly connected* LAN available for.

The following example shows an interface definition in Vyatta and how the interfaces are associated with topologies:

```
interfaces {
    ethernet eth0 {
        address fc10:f115:200:0004::1/64
        description LAN
        duplex auto
        execute-script LAN-TAG-IPv6-GRE-C
        hw-id 00:10:f3:21:79:c0
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 1
                    description "low bit-rate"
                }
                topology 33 {
                    cost 1
                    description "high bit-rate"
                }
                topology 34 {
                    cost 1
                    description "low delay"
                }
                topology 99 {
                    cost 1
                    description "base topology"
                }
                transmit-delay 1
            }
        }
        smp_affinity auto
        speed auto
        topology-address fc10:f115:200:0004::1/64 {
            topology 99
            topology 32
            topology 33
            topology 34
        }
    }
}
```

Under the *ospfv3* section, OSPFv3 protocol parameters for the interface is specified, as well as information of which topology this interface should be announced by OSPFv3 to participate in. The cost to be used for the topologies in path calculation for this interface is also given. The topology-address command makes sure that the *directly connected* interface address segment is also written to the specified topology forwarding tables.

In order to use the MT-routing protocol with a wide variety of military radios and SatCom systems, it is in many cases necessary to create a routing overlay with tunnels to bind the different systems together. The example below shows configuration of an ip6ip6 type tunnel with MT-routing configuration:

```
tunnel tun804 {
    address fc10:f11f:8000:2354::54/64
    description WM-TUN-DEU3-NOR4
    encapsulation ip6ip6
    ipv6 {
        dup-addr-detect-transmits 1
        ospfv3 {
            cost 1
            dead-interval 8
            hello-interval 2
            instance-id 0
            topology 32 {
                cost 75
            }
            topology 33 {
                cost 75
            }
            topology 34 {
                cost 75
            }
            topology 99 {
                cost 75
            }
        }
    }
    local-ip fc10:f115:200:14::1
    parameters {
            hoplimit 3
            tclass inherit
        }
    }
    remote-ip fc10:f115:200:1f::1
    topology-address fc10:f11f:8000:2354::54/64 {
        topology 32
        topology 33
        topology 34
        topology 99
    }
}
```

The configuration of a tunnel interface is very similar to the configuration of a physical interface. However, for the tunnel we also had to support configuration to request a copy (`tclass inherit`) of the *traffic class* field in the original packet header to the *traffic class* field or the *type of service* field in the tunnel header. This was necessary to enable the QoS mechanisms to treat a packet wrapped in a tunnel header according to the original packet's QoS classification. Note that for the Vyatta Napa release, there is a problem in the kernel that leads to kernel crash (kernel panic) when we try to inherit the *traffic class* field in an IPv6 packet to the *type of service* field in a Generic Routing Encapsulation (GRE) tunnel. This problem is most likely solved in newer versions of Vyatta, but this has not been verified by us. The consequence of this problem was that we were not able to do the desired QoS queuing for packets tunneled over IPv4 radio networks in the CoNSIS field test.

We also specified a tunnel hop limit of 3 (`hoplimit 3`). This limited the length of a tunnel to be an internal hop in the source and destination router plus one radio hop. With this configuration the overlay routing protocol was never allowed to use tunnels over multiple wireless hops and thus had a realistic hop count in its shortest path calculation.

As described in chapter 4.2 we spent quite some time making a flexible interaction between an MT-routing protocol and an ST-routing protocol. Most existing routing protocols operate on the default forwarding table *main*. In border routers that support both an MT-routing protocol and one or more ST-routing protocols (both interior and exterior protocols), the ST-routing protocols will maintain the *main* forwarding table and the MT-routing protocol will maintain multiple other forwarding protocols. Routing entries in the *main* forwarding table is tagged with the identifier of the routing protocol that provided the entry, thus this routing table is the repository for redistribution of routes between protocols. In CoNSIS we wanted to be able to specify which topologies that should be allowed to redistribute routes from an external ST-routing protocol. We also wanted to choose which topology to make available for redistribution by external ST-routing protocols. We needed to use the *main* forwarding table for this interaction. It was therefore important that the *main* forwarding table was not used as one of the MT-routing forwarding tables in this architecture.

An entry in the *topology* configuration (`clone ospfv3`) specified which topologies to make available for external protocols to redistribute. This command instructed Quagga to maintain a copy of the routing entries in the chosen forwarding tables also in the *main* table.

The following example shows a situation where topology 33 is made available for other routing protocols to redistribute. The `clone ospfv3` command in the topology configuration makes sure that a copy of the topology 33 routing entries is also maintained in the *main* forwarding table.

```
topology 33 {
    clone ospfv3
    name high-bit-rate
    priority 10
    target ipv6-only
    traffic-class 0x28 {
    }
```

The mechanisms for choosing which topology(ies) that should redistribute routing information from an external protocol were twofold:

1. One command made sure that the routing entries tagged with a specified routing protocol identifier in the *main* table were also maintained in one or more topology tables.
2. One command told the MT-routing protocol to redistribute the routing entries, with the specified protocol identifier, in its forwarding table to the rest of the MT-routing domain.

The copy of routing entries from *main* to the topology table was configured similarly to the reverse situation. In this example the chosen external protocol is BGP:

```
topology 33 {
    clone ospfv3
    clone bgp
    name high-bit-rate
    priority 10
    target ipv6-only
    traffic-class 0x28 {
    }
```

Configuration of redistribution of external routes in the MT-domain is configured in the routing protocol configuration. An example is given below where routes from the external BGP protocol are redistributed into topology 32:

```
protocols {
    ospfv3 {
        area 0.0.0.0 {
            interface eth0
            interface tun701
                .
                .
                .
            interface tun915
        }
        parameters {
            disable-rfc2740-compatibility
            router-id 2.0.0.5
        }
        topology 32 {
            redistribute {
                bgp {
                }
            }
        }
```

An example configuration of the router in one of the vehicles in the CoNSIS field experiment (NOR3), and of the stationary gateway between the mobile network and the deployed head quarter (HQ) (DEU5) is given in 0.

6.3 The CoNSIS convoy platforms

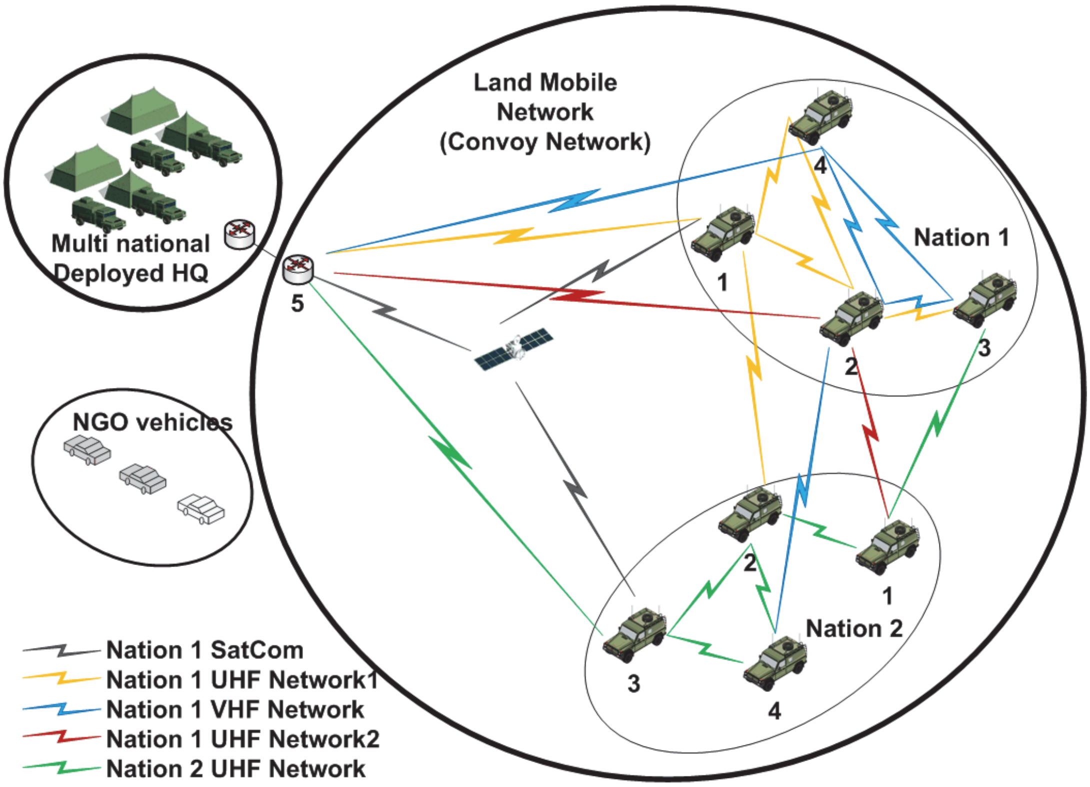

Figure 6.2 The land mobile network in CoNSIS (coalition convoy).

The land mobile network component in the CoNSIS network architecture is represented by a multinational convoy in the scenario and in the field test (Figure 6.2). The network consists of a German (Nation 1) and a Norwegian (Nation 2) convoy segment. Each segment consists of four mobile nodes. A stationary router co-located with the multinational HQ is also part of this network segment. All radio connections between the deployed network and the mobile network is terminated at this stationary gateway. Several gateways to the mobile network can be used, but in such cases the QoS architecture requires that a tunnel is established over deployed infrastructure between the gateways in the land mobile network segment.

The convoy network is connected to a multi-national deployed headquarter and the rest of the network (Figure 6.1). The NGO vehicles also have a network connection to the military convoy. This connection is not visible in Figure 6.2 since this network is not allowed to be part of the coalition transport network. Traffic is sent to/from the NGO segment via application gateways handled by other CoNSIS task groups.

6.3.1 Radio networks in the CoNSIS convoy

The convoy network deployed in the CoNSIS field test consisted of five different radio networks. It was therefore a highly heterogeneous MANET. Table 6.1 gives some details of the radios that were present in the CoNSIS experiment. The heterogeneous network was used for internal convoy communication and reach-back to the deployed headquarters.

	Radio Type	**Number of radios in the network**	**Shared channel data rate***
Nation 1 SatCom	Thrane &Thrane BGAN Ex. 727	unknown	384kb/s
Nation 1 UHF Network1	IABG, HiMoNN	6	1Mb/s - 11Mb/s
Nation 1 VHF Network	Harris , RF-7800S	5	64kb/s
Nation 1 UHF Network2	Rockwell Collins, FlexNet-Four	3	1Mb/s
Nation 2 UHF Network	Kongsberg, WM600	6	920kb/s - 2400kb/s

* The data rates are approximate values

Table 6.1 Radios used in the CoNSIS Convoy test network

SatCom modems and antennas were installed on the specified vehicles and on the gateway node collocated with the HQ (node 5/DEU5), but due to problems in the network of the service provider we were not able to use this connection during the two weeks of field test. Despite of this we chose to keep the satellite connections in the network drawings in this chapter since the MT-routing and QoS architecture is designed to handle also SatCom transport.

6.3.2 Routing overlay – some comments

For the CoNSIS field experiments we chose to create a routing overlay over all five deployed radio networks and radio links. Another option would have been to create a partial overlay and use route redistribution of routing information from the MANET protocol in the different radio networks into one or several topologies in the MT-routing overlay. We chose a full overlay for several reasons. For one, the SatCom BGAN connections required an overlay to participate in a routed network since we would not be allowed to interface the internal routing protocol of the SatCom provider. The SatCom connection also only supported IPv4, thus an overlay was needed to provide IPv6 routes. The FlexNet radios were also configured for IPv4. Additionally, the cost of creating an overlay over FlexNet was not large since only three of these radios were deployed in the field test. The HiMoNN radios were also configured for IPv4 and thus an overlay was needed to provide IPv6 routing. The WM600 radios were configured for IPv6. However the protocol made available on the wired interface to the external router was either OSPFv3 or OSPFv3 MDR [25]. OSPFv3-MT is based on OSPFv3 but not compatible with OSPFv3, and it is not possible to do route redistribution between instances of the same protocol. Quagga's zebra routes created by OSPFv3, OSPFv3 MDR or OSPFv3-MT are all tagged to be OSPFv3 routes. Kongsberg Defence, the manufacturer of the WM600 radio, also participated in CoNSIS, and through cooperation with them we were given a configuration option to run the Optimized Link State Routing (OLSR) protocol [34] on the wireless and wired interface of the WM600 radio. OLSR routes can be redistributed into the OSPFv3-MT protocol. However, due to time constraints we were not able to test this prior to the field test and thus decided to build an overlay also over the WM600 radio.

Building a full mesh overlay is not a scalable solution. The CoNSIS land mobile network segment consisted of 9 nodes and 5 different radio systems. A full mesh over this fairly small but complex network (Figure 6.2) required 46 different two-way tunnels and generated much configuration work and lots of cases of erroneous configuration. One mobile node could have up to 14 tunnel interfaces and the collocated gateway node had up to 20 tunnel interfaces. This experiment was a proof of concept for the MT-routing for QoS support, therefore using such overlay was acceptable. For future product development, however, solutions must be found for better integration between the MT-routing and the routing protocol of the radio networks.

Using route redistribution between the radio networks and the MT-protocol in the routers that interconnect the different networks is not a good solution either. No military radio provides the MANET routing protocol that is used on the wireless interface on the wired interface to the external router. Thus one route redistribution must be done between the wireless protocol and the protocol on the wired side and another route redistribution must be done between the radio's protocol on the wired interface and the MT-routing protocol (assuming that the MT-protocol is not available on the radio). All these route redistributions delay the propagation of route changes, and all route metric information that might have been present in the original routes are lost on the way. For a scenario like the CoNSIS Convoy, it would be beneficial to run the same routing protocol on the wireless network and on the wired network; the combined heterogeneous network is also a MANET and requires the same frequent route updates etc. Unfortunately, the radio manufacturers typically run a proprietary MANET protocol on the wireless interface, thus even if they had made this protocol available on the wired side of the radio, an external router would not have been able to interface the protocol. Yet another problem with route redistribution is how to handle redistribution in a wireless multi hop network with several gateways? When the radio network is partitioned it must be known in the external network which gateway to use in order to reach the destination in the MANET. This means that the ip address of single platforms must be made available to the external protocol. This is not a scalable solution.

A third option to network overlay and route redistribution, is to provide an option to turn off the wireless routing protocol in the radio for scenarios where efficient interaction between different small radio networks is more important than optimal internal routing in one larger radio network. In this situation an external router can use the radio as a layer 2 device and run the MT-routing directly on the wireless links instead of as an overlay. In this situation the routing protocol in the external router can make improved routing decisions with the aid of protocols that provide some cross layer information from the radio to the router (e.g., the protocols described in [35][36][37]).

The discussion on how to find the most efficient solution for how to integrate the MT-routing protocol and its QoS mechanisms with the radios in a heterogeneous mobile military network is out of scope for this report and will be left for future work.

Due to the large difference in transmission characteristics (data rate, delay, etc.) for the different radio networks in the CoNSIS convoy, the OSPFv3-MT protocol parameters had to be tuned to suite the different carriers. For the low data rate networks it was important to not overload the links with routing traffic, and for the high data rate networks it was important to identify route changes quickly. To support these requirements the Hello interval had to be set differently for these two types of network. Note that different timer values for the same routing parameter in one network increase the risk of routing loops. This problem is mentioned in [38]. Some other OSPFv3 parameters were also set different from the default value. Table 6.2 show the chosen values for the OSPFv3 timer parameters.

Radio Type	Hello interval	Dead interval	Retransmit interval	Transmit delay
Nation 1 SatCom	20	80	15	5
Nation 1 UHF Network1	2	8	5	1
Nation 1 VHF Network	40	160	15	5
Nation 1 UHF Network2	2	8	5	1
Nation 2 UHF Network	2	8	5	1

Table 6.2 OSPFv3 timer parameters used for the different convoy network interfaces

6.3.3 Topology configuration

The different transmission technologies present in the planned experimental network have substantially different characteristics when it comes to e.g., transmission delay, transmission range and data rate. Given the heterogeneous network as shown in Figure 6.2, the end-to-end network capacity could change from a relatively high data rate of several Mb/s to a few tens of Kb/s when a node moves from UHF coverage to a path that includes one or more VHF and/or SatCom on-the-move links. This large variation in available data rate is difficult to handle for the resource management entity. In such a scenario it is also important that the network is able to prioritize the mission critical data traffic in overload situations.

In the CoNSIS network architecture for the land mobile network we interconnect the different links and networks present in the network with the OSPFv3-MT routing protocol in one flat routing domain. This allows full dynamics in the network.

To demonstrate the use of multiple topologies for QoS purposes we defined three topologies in the CoNSIS convoy network in addition to the base topology:

- A high data rate topology
- A low data rate topology
- A low delay topology

Table 6.3 shows how the different radio networks in the CoNSIS convoy network are associated with the three defined topologies. All radio networks also participate in the base topology.

Radio Type	**Low data rate topology**	**High data rate topology**	**Low delay topology**
Nation 1 SatCom	X	-	-
Nation 1 UHF Network1	X	X	X
Nation 1 VHF Network	X	-	X
Nation 1 UHF Network2	X	-*	X
Nation 2 UHF Network	X	X	X

* Originally these wideband radios were also intended to participate in the high data rate topology, but for testpurposes, as the SatCom network was not available for the field test, we chose to use this network as one of the networks that does not participate in all topologies.

Table 6.3 The use of the radio networks in the topologies

The low data rate topology includes all links. The high data rate links are also included in this topology to increase connectivity and network robustness. However, the topology cannot guarantee more than a low data rate capacity. The low data rate topology is intended to be the topology used for most of the typical applications for tactical operations, e.g., command and control applications. These application types must not require high data rate.

The best path within each topology is calculated based on the MT-routing cost parameter for each link between source and destination. The UHF networks are given low cost whereas the SatCom and VHF networks are given a high cost (ref Table 6.1). We set the same costs for all topologies, but acknowledge that it could be beneficial in some cases to use different costs for different topologies, and thereby prioritize the utilization of the network types differently for different traffic types.

Radio Type	**OSPFv3 cost**
Nation 1 SatCom	500
Nation 1 UHF Network1	50
Nation 1 VHF Network	1300
Nation 1 UHF Network2	150
Nation 2 UHF Network	75

Table 6.4 OSPFv3 cost for the radio networks in the field test.

Figure 6.3 shows a radio topology where Nation2's portion of the convoy is driving into a terrain with difficult channel propagation conditions for Nation2's UHF radio. Table 6.5 shows the routing table for the three topologies for all the vehicles in Nation2 for the radio connectivity represented in the figure. This example assumes that the SatCom connection is operational.

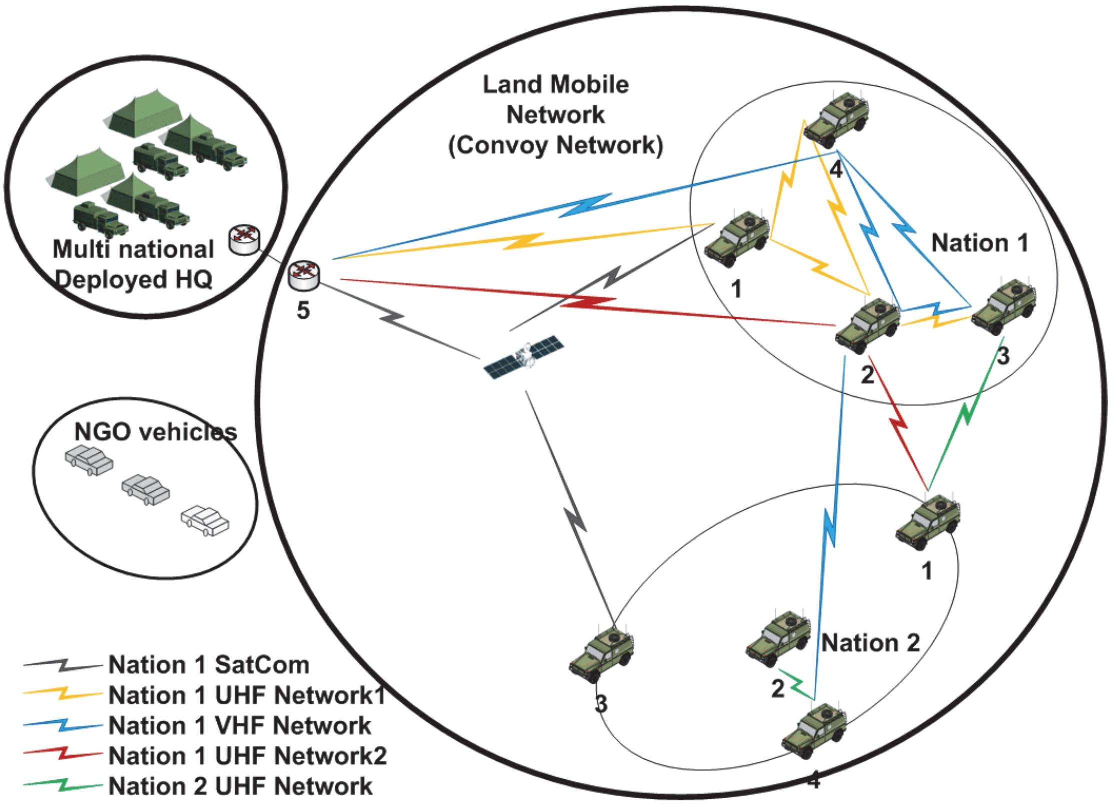

Figure 6.3 Network connectivity in terrain with difficult radio propagation for Nation 2's UHF network. This example assumes that the SatCom connection is operational.

Nation 2 vehicle no.	Low data rate topology	High data rate topology	Low delay topology
1	All vehicles	All Nation 1 vehicles	All except Nation2:3
2	All vehicles	Nation2:4	All except Nation2:3
3	All vehicles	-	-
4	All vehicles	Nation2:2	All except Nation2:3
* The destinations are represented as follows: Vehicle no. 3 in Nation2 is written as Nation2:3			

Table 6.5 Routes Available in the three different routing tables in the vehicles of Nation 2 in Figure 6.3*

6.3.4 QoS classes and configuration

In the MT-supported QoS architecture we require that all traffic in the network is tagged with the appropriate QoS tag. We have chosen to use the traffic class field in the IPv6 header, to mark the packets. We use this field to encode the QoS class (named Service-based Class (SBC) in [39]), and traffic priority (IP Military Precedence Level (IP MPL)) as suggested in [39]. Figure 6.4 shows the chosen format.

For the CoNSIS QoS architecture we decided that there should not be a fixed association between a traffic type and a SBC and IP MPL. We believe that it is wise to allow network planners of an operation to define the SBC for a service. E.g., in some operations it might be important to provide frequent high resolution images, while other operations would rather spend the data rate on other services. In such a setting, an application (service) can be tagged with one SBC in one operation and another SBC in the next. Nevertheless we created an example list of services and signaling traffic for the CoNSIS experiment and associated these with the SBC and IP MPL as shown in Table 6.6. Table 6.7 then shows how some selected services from Table 6.6 are associated with the topologies created for the experiment.

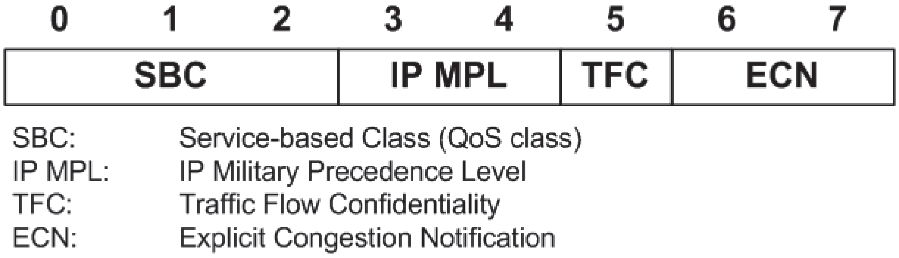
0
1
2
3
4
5
6
7
SBC
IP MPL
TFC
ECN
SBC: Service-based Class (QoS class)
IP MPL: IP Military Precedence Level
TFC: Traffic Flow Confidentiality
ECN: Explicit Congestion Notification

Figure 6.4 Suggested use of the IPv6 traffic class field.

SBC	**Service**	**One example of mapping between CoNSIS services and the SBC**		**DSCP**	
NETR	Network Infrastructure	- Routing (e.g. OSPFv3-MT, BGP, OLSR) - Management, ICMP Error Messages - TIBER Auto detection of classified enclaves		CS6	110000
OAM	Network Management	- Security management		CS2	010000
SIG-T	Call Signaling	- VoIP signaling - Notification Management Service - Service Discovery Service		CS5	101000
VOICE	Voice		F		101010
			P		101100
		- MELPe	R	EF	101110
VIDEO	VTC		F	AF41	100010
			P	AF42	100100
			R	AF43	100110
STREAMING	Streaming media		F	AF31	011010
			P	AF32	011100
			R	AF33	011110
LDELAY	Low latency data	- Operational Alarm Messages - NFFI Blue Force Tracking Service	F	AF21	010010
		- Chat Application	P	AF22	010100
		- Network Services (e.g. DNS, DHCP)	R	AF23	010110
BULK	Bulk	- Image messaging service	F	AF11	001010
			P	AF12	001100
			R	AF13	001110
NORM	Best effort	Other applications		BE	000000

Table 6.6 CoNSIS services mapped to SBCs

CoNSIS service	**Low data rate topology**	**High data rate topology**	**Low delay topology**
NFFI Service (AF21)	X	-	-
Chat application (AF22)	X	-	-
VoIP (MELPe 2400) (EF)	-	-	X
Image msg. service (AF11)	-	X	-

Table 6.7 Mapping between selected services and the defined network topologies

6.3.5 Packet scheduler and QoS queue configuration

For each interface we also needed a QoS queue structure, a packet scheduling policy and traffic shaping parameters.

For the low data rate interfaces we have chosen to configure a strict priority queue (PRIO) [40] with no fairness in the packet scheduling. This ensures that the highest priority traffic types are given all resources until the queue for the high priority traffic is empty, and then the next priority queue is served. If the traffic load is higher than the configured network capacity, then the lowest priority traffic is never served. Since there is no flow control on the interfaces between the MT-router and the connected radio systems, this interface looks like a gigabit interface to the router. In order to control the data rate that is allowed to flow over this interface we set shaping parameters on each interface. The shaping data rate is a static value and in CoNSIS we set this value based on the theoretical maximum capacity of the radio network, and on the network topology that was expected to be the typical topology for the experiment. See Table 6.8 for the queue type defined on each interface type, and the shaping data rate configured on each interface type.

	Radio Type	Scheduler	Shaping b/s
Nation 1 SatCom	Thrane &Thrane BGAN Ex. 727	Strict priority (PRIO)	64kb/s
Nation 1 UHF Network1	IABG, HiMoNN	Hierarchical Token Bucket (HTB)	5Mb/s
Nation 1 VHF Network	Harris , RF-7800S	Strict priority (PRIO)	30kb/s
Nation 1 UHF Network2	Rockwell Collins, FlexNet-Four	Hierarchical Token Bucket (HTB)	350kb/s
Nation 2 UHF Network	Kongsberg, WM600	Hierarchical Token Bucket (HTB)	1Mb/s

Table 6.8 Queue type and shaping rate for the radios used in the CoNSIS field test.

For the high data rate interfaces we use the hierarchical token bucket (HTB) [41] queuing structure for Linux, and associate a share of the shaped bandwidth to each of the QoS classes. This supports traffic priority but also provides some fairness in the packet scheduling, i.e., best effort traffic is also given its share of the shaping rate during high load situations. The classes were allowed to borrow capacity from each other up to a *ceiling* rate that was set to about 80% of the shaping rate.

The packet drop probability is defined by the priority of the QoS classes, and the queue length. When a queue is full, packets are dropped. QoS classes that need low delay are set up with short queues, as are QoS classes with periodic traffic where it is important to always transmit the most recent message. The best effort class is set up with fairly long queues, to support delayed responses rather than lost responses due to queue overflow. Table 6.9 and Table 6.10 show how the HTB and PRIO queues were configured for the field test. Refer to Table 6.6 for the association between CoNSIS traffic types and the traffic class tag (SBC).

Note that the queue configuration that we used during the experiment does not cover all traffic classes specified in Table 6.6. We defined explicit queues only for the traffic types that we expected to see in the experiment. The queue configuration in a real operation will most likely involve more queues and should be configured as part of the preplanning process of an operation. It should take traffic load, traffic types and traffic priority into account in the configuration phase

HTB queues	Traffic Class 0xC0 (CS6)	Traffic Class 0xb8 (EF)	Traffic Class 0x48 (AF2.1) 0x50 (AF2.2)	Traffic Class 0x28 (AF1.1)	Other classes and BE
Portion of shaping rate allocated	15%	10%	40%	10%	25%
Queue lengths in packets	10	5	20	50	100

Table 6.9 HTB queue configuration

PRIO queues	Traffic Class 0xC0 (CS6)	Traffic Class 0xb8 (EF)	Traffic Class 0x48 (AF2.1) 0x50 (AF2.2)	Traffic Class 0x28 (AF1.1)	Other classes and BE
Priority	1	2	3		4
Queue length (in packets)	10	5	10		60

Table 6.10 PRIO queue configuration

The *ip6tables* functionality in Linux is used to mark MT-routing traffic with the correct QoS class. All user traffic in the CoNSIS network is encrypted by IPSec solutions, the user traffic must therefore be marked with the correct QoS class by the source. This marking is also used to associate the QoS classes with the forwarding table for the correct topology. The Linux traffic control (*tc*) [42] tool is used to setup the queuing and scheduling mechanisms. In Appendix B the queue and scheduler configuration for each interface type is given.

It was briefly mentioned in chapter 6.2 that we experienced problems when we tried to copy the *traffic class* field from the IPv6 data packet into the *type of service* field in the IPv4 tunnel header of a GRE tunnel. When we tried to do this, the kernel crashed. Since the MT-routing protocol ran in a network overlay, all packets that arrived at the queue and scheduling mechanisms of each interface was wrapped in a tunnel header. The QoS queues and prioritized scheduling were therefore not used as planned for all interfaces that connected to an IPv4 radio network. All traffic was treated as BE traffic by the queues and the scheduler for these interfaces.

6.4 Multicast support

End-to-end multicast support was also a requirement in the CoNSIS network. End-to-end multicast via several different radio networks has received very little attention by the radio manufacturers. Many radio manufacturers provide a multicast solution for the wireless interface, but very few provide mechanisms to forward multicast traffic to/from connected networks to/from the wireless multicast distribution protocol. In CoNSIS we needed a connected multicast service both internally in the heterogeneous land mobile network, and between the land mobile network and the deployed coalition backbone. For the land mobile network we therefore chose to run a multicast protocol on the overlay network described in chapter 6.3.

In [43] different group communication protocols for mobile military networks are discussed. The conclusion is that as a minimum, a flooding based protocol should be available for such networks. For CoNSIS we also chose to use a flooding based protocol for this network type. An implementation of Simplified Multicast Flooding (SMF) [44] by Naval Research Lab (NRL) [45] was integrated in the Vyatta environment described in chapter 6.1.

When testing the NRL SMF implementation on the overlay network, we discovered that the ip6ip6 tunnels did not accept the SMF frames. NRLSMF transmits Ethernet frames when forwarding multicast packets. On a GRE tunnel this is handled by built-in EoIP (Ethernet over IP) support in the Linux kernel. This support was not available for IPv6 tunnels. For CoNSIS we therefore implemented a special function to facilitate the transport of Ethernet frames over the ip6ip6 tunnel. The small utility program *ip6ip6tap* is placed between SMF and the tunnel, and acts as an encapsulator for Ethernet frames.

In order to configure SMF for the CoNSIS field experiment we had to set up the utility program for each ip6ip6 tunnel. The example below shows a partial tunnel configuration with the command under *parameters, ipv6* that set up the utility. The program needs to know the IP address given to the remote end of the tunnel, and the port number dedicated to the encapsulated traffic:

```
tunnel tun815 {
        • • •
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:2554::25
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
    }
```

Next we configured the SMF program with the correct interfaces:

```
service {
    nrlsmf {
        parameters "hash MD5"
        parameters ipv6
        parameters "idp off"
        parameters "cf eth0,tap-tun804,tap-tun808,tap-tun811,tap-tun813,tap-
tun815"
    }
}
```

The solution chosen to support multicast in the land mobile network provided us with connected multicast support in a very heterogenous environment. It should be noted, however, that it was a very inefficient solution.

The next challenge that had to be solved was the interconnection of the SMF protocol in the land mobile network segment and PIM-SM (Protocol Independent Multicast - Sparse Mode) [45] multicast protocol that was chosen for the deployed coalition backbone. Some work has been done to solve the interconnection of PIM-SM and SMF, e.g., [47] and [48]. For CoNSIS we did a static configuration, where the land mobile network joined all interesting multicast groups in the deployed network and always sent multicast traffic generated in the land mobile network to the PIM-SM entity in the gateway(s). Due to a reverse path check in the Cisco PIM-SM router we had to fool the router into believing that the complete land mobile network was directly connected to the Cisco router. While the solution worked, it was not elegant.

Multicast in a heterogeneous network environment is an area where much more work is needed.

7 MT-routing Tests and Results

During the CoNSIS field experiment we performed several tests to demonstrate the functionality of the MT-routing (MTR) overlay. Three MTR tests were specified for the field experiment:

- MTR-2: Demonstrate seamless mobility in a heterogeneous wireless network
- MTR-3: Test the use of multiple topologies for QoS purposes
- MTR-4: Limiting convoy network visibility for adjacent networks

The two weeks of experimentation were split into two parts. Most of the period was reserved for individual, tightly directed tests, while the last few days were reserved for common testing. In the latter phase parts of the scenario were played out, and selected functionality from the different tasks was demonstrated at the same time. All three MTR tests were conducted during the first test period and MTR-2 was also run during the scenario test.

Each morning during the test period we synchronized the clocks on the routers and on the traffic generators and receivers to the GPS time with 1s accuracy.

7.1 Tightly directed tests at the WTD-81 premises

The purpose of the tightly directed MTR tests on the WTD-81 premises, was to perform the tests in a controlled manner, with tightly directed traffic load and mobility to trigger topology changes. This allowed us enough control of the network connectivity to validate the performance of the MT-router against the expected results based on theoretical predictions. In most of these tests we disconnected the convoy network from the rest of the CoNSIS test network at the gateway node (DEU5 in Figure 7.3).

Most of the tests were performed on WTD81's test field (Figure 7.). An old airfield tower on this field (Figure 7.2) was used as an elevated platform for reach-back from the mobile convoy to the deployed HQ. All radio types deployed in the convoy were installed in this tower. The gateway node between the land mobile networks (convoy) and the deployed HQ was also placed here.

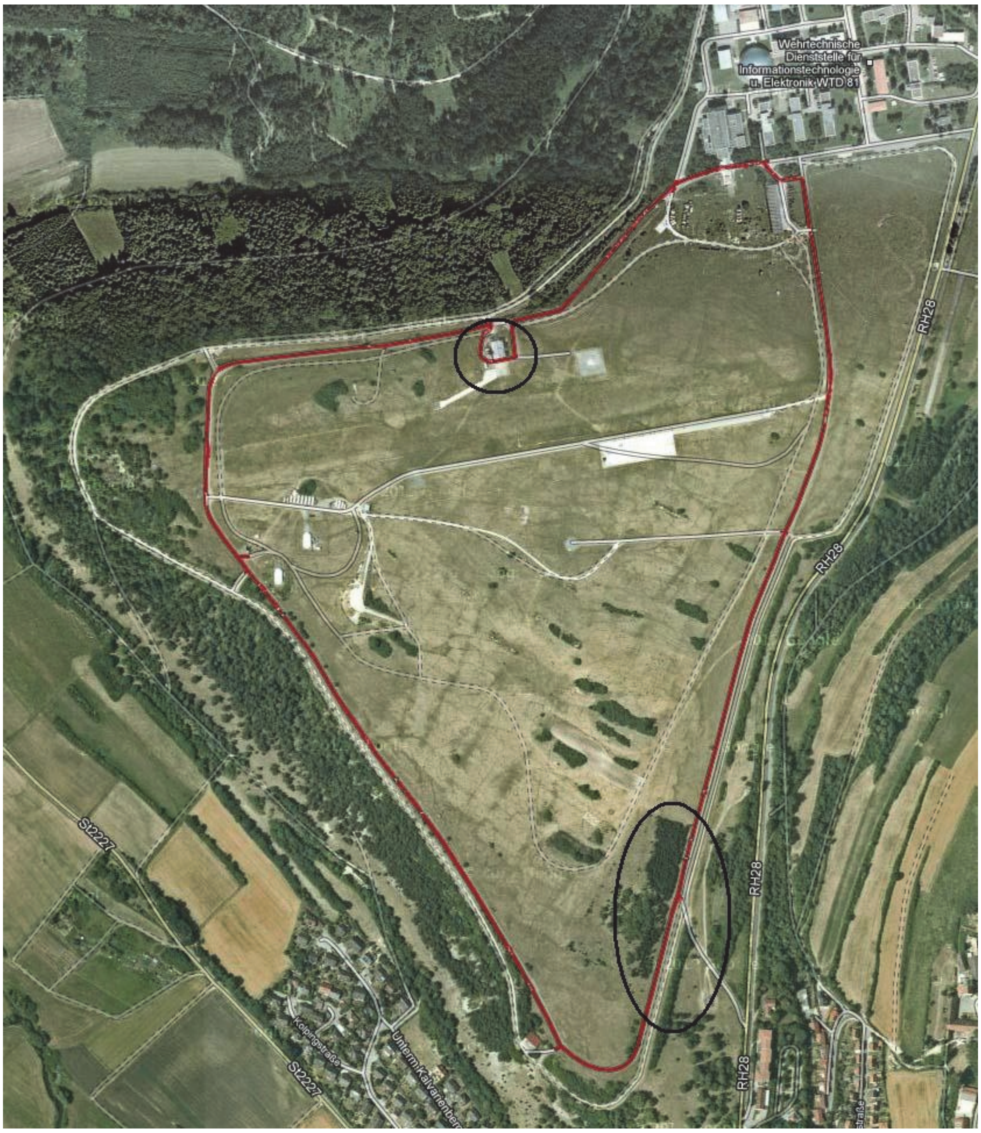

Figure 7.1 A picture from Google maps (http://maps.google.com) showing the field where we performed most MTR tests. The black circle at the top shows the position of the tower where the gateway node was placed (DEU5). The circle at the bottom shows an area where there were bad channel conditions on the UHF frequency band.

We had originally planned to run all three MTR tests with a more complex network connectivity situation, where more nodes and more radio networks had to be involved to provide a route from the source to the destination. However, due to time constraints and problems with the availability of the SatCom network, and also some configuration problems with the VHF radios and one of the UHF networks, we completed the scaled down tests presented below.

Figure 7.2 *Tower where the gateway node between the convoy and the rest of the CoNSIS network was placed. All radio types used by the convoy were also installed in this tower.*

7.1.1 MTR-2: Demonstrate seamless mobility in a heterogeneous wireless network.

As described in chapter 6 we chose to create an overlay network over all the deployed radio systems in the land mobile network. The purpose of this overlay was to interconnect the different systems to provide one flat heterogeneous transport network to the end-user. With this design all available networks in the coalition operation are handled as a common resource. If a link breaks somewhere on the way between a source and destination, an automatic reroute happens if a route exists from the source to the destination through a combination of the available radio networks. If several routes exist, the OSPFv3 cost associated with the links of each network type is used to choose the best route. The purpose of this test was to show how well the MT-router overlay was able to reroute traffic when a link break happened in one of the radio networks.

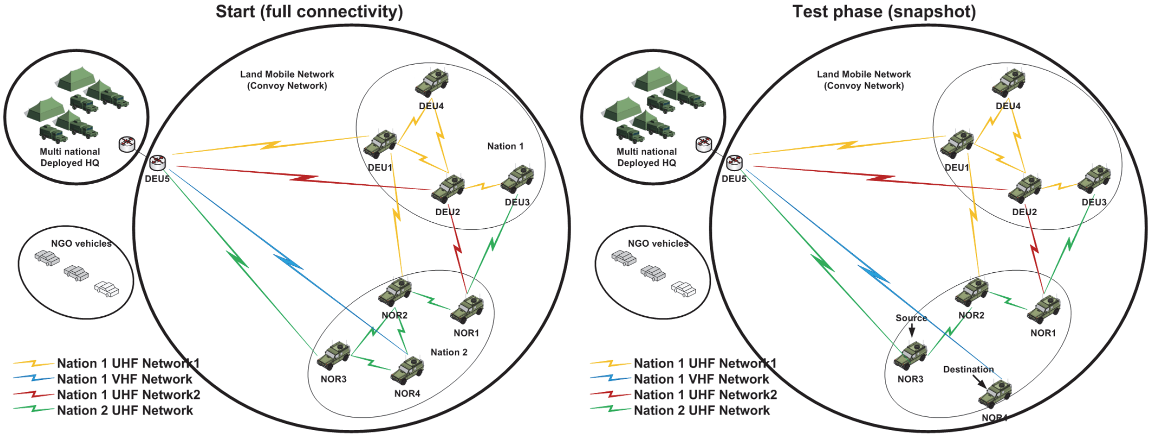

Figure 7.3 *Initial convoy network setup. The colored links in the figure identify which vehicles were equipped with which radio type. Only a subset of the actual connections is shown. At the initial stage all nodes had a one-hop connection to all other nodes in each radio network. The test phase (right hand side) shows the actual connections at the destination node at one stage of the test.*

Figure 7.3 shows the radio interfaces that were used during this test. All of Nations1's UHF Network 1 radios were participating in the network, as were all of Nation2's UHF Network radios and all of Nation1's UHF Network 2 radios. Only two of Nations1's VHF radios were used (to form one link between the convoy and the tower). During this test we sent test traffic from NOR3 to NOR4. The traffic was generated with the Multi-Generator (MGEN) [49] tool.

At the start of the test, all convoy vehicles were moved to an area next to the airfield tower. The routing tables in all the NOR nodes showed the theoretical minimum cost to all the other nodes in the convoy, due to the situation with full connectivity in all radio networks. At time t0 NOR4 goes for a slow drive around the test range (Figure 7.). At the start of the drive there is a one-hop connection between NOR3 and NOR4. When NOR4 approaches the area with reduced connectivity, this link breaks and the route must go via the VHF link from the tower. A short time later NOR4 can again be reached on Nations2's UHF Network via the tower. The cost graph in Figure 7.4 shows the cost of the path chosen by the routing protocol during a section of the test (Table 6.4 shows the OSPFv3 cost assigned to the different radio types). The test phase drawing in Figure 7.3 shows a snapshot of the network connectivity at NOR4 at some point during the test.

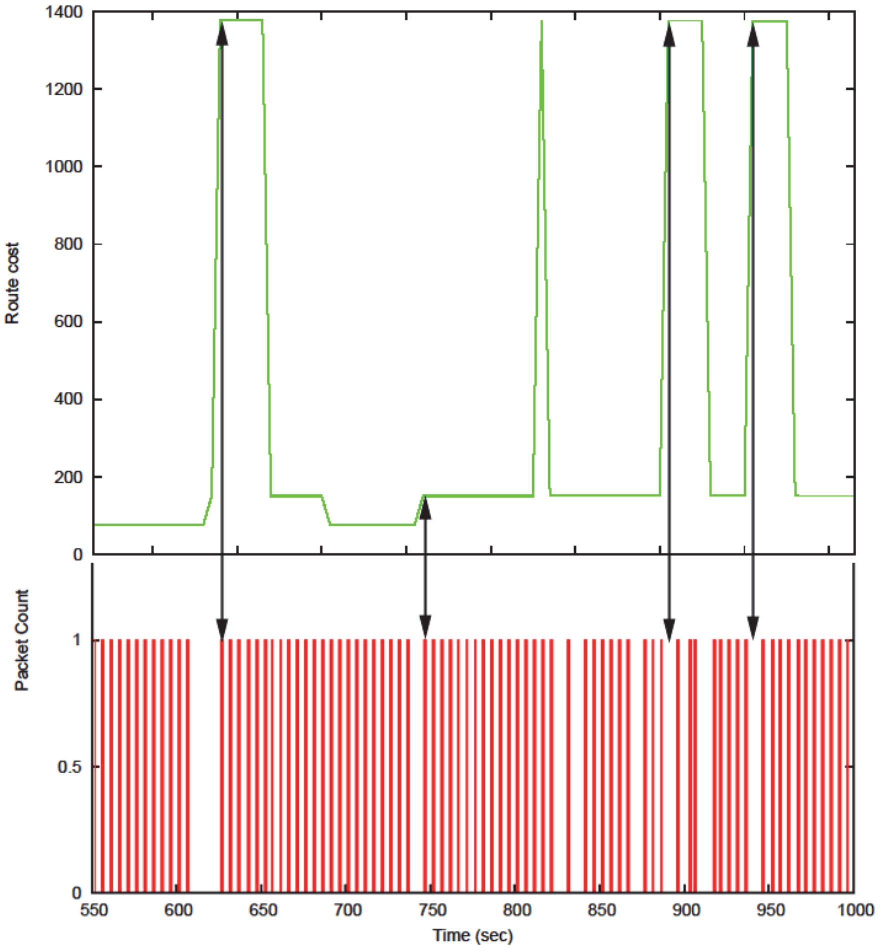

Figure 7.4 Route cost on the route from NOR3 to NOR4 (Figure 7.3) and the associated gaps in reception of packets while the routing protocol establishes a new route.

Figure 7.4 also shows the received packets at NOR4 for the time period when many connectivity changes happen. The figure also shows the route cost for the route from NOR3 to NOR4 for the same time frame. We logged the routing tables of the MT-routers every five seconds. We also sent test traffic with low intensity (one packet every five seconds), thus it is not possible to collect accurate data for the time needed by the routing protocol to detect a link break and provide a new route. However, in the MTR-3 test (described in chapter 7.1.2) we also sent traffic with frequency of two packets per second. Based on the logs of this traffic, we see that it typically takes less than ten seconds to detect a link break and establish a new route. This time frame was expected since we configured OSPFv3 to send Hellos every two seconds.

This test clearly demonstrates that the MT-routing overlay is able to efficiently utilize all available radio networks for its route calculations in the convoy network. From the route cost graph we can see that at least three different routes were used during the test.

7.1.2 MTR-3: Test the use of multiple topologies for QoS purposes.

In this test we wanted to show how topologies could be used to provide different paths for different traffic classes in a heterogeneous network. We also wanted to demonstrate how the topology concept could help block traffic at the source for flows that could not be supported by the current network connectivity. We defined three different topologies for the CoNSIS field experiment in addition to the base topology (see Table 6.3). For this test we also chose to configure the low data rate topology with the same interfaces and cost as the base topology. Hence, the low data rate topology included all links.

We initially planned to run this test for all topology types in the network. However, due to time constraints, we chose to run the test with traffic in the *low data rate* and *high data rate* topologies only. The network connectivity at the start of the test is the same as for the MTR-2 test (chapter 7.1.1). Figure 7.5 shows the connectivity for the two topologies. See Table 6.3 for the association between the different radio networks in the convoy and the topologies.

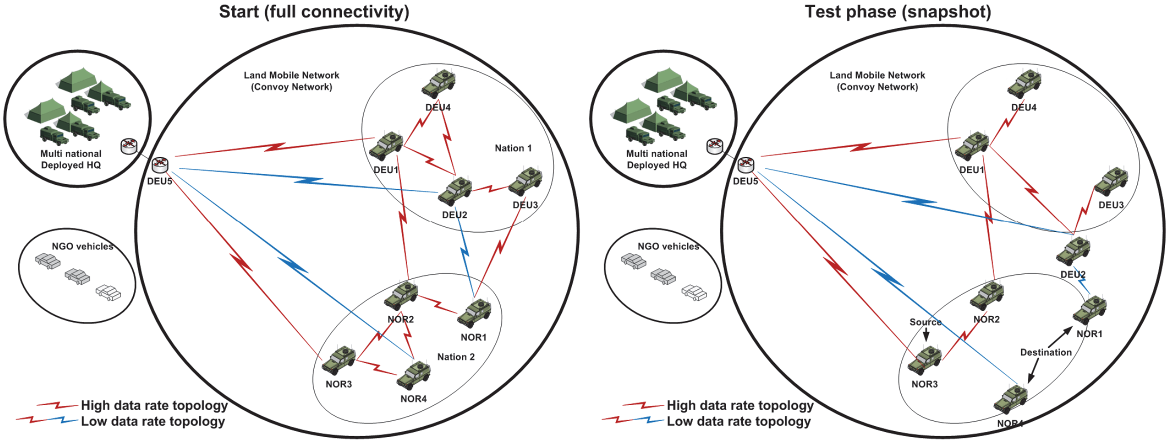

Figure 7.5   Network connectivity for two different topologies at the start of the test and at the test phase with bad connectivity for Nation1's UHF radio. All links (both red and blue) participate in the low data rate topology

The test starts with minimum route cost and full connectivity in both topologies. Traffic on both topologies is sent from NOR3 to all other NOR nodes. The traffic is marked in the IPv6 *traffic* *class* field with traffic classes that are associated with each of the two topologies (see Table 6.7 for the traffic classes associated with each topology). This test is also carried out on the road shown in Figure 7.. NOR4 is the first car that starts moving, and it keeps going until it eventually reaches a spot where there is no connectivity to other vehicles on the *high data rate topology*. Figure 7.6 shows how traffic on the *high data rate topology* is blocked at this point, while traffic on the *low data rate topology* keeps flowing. The figure also shows the route changes (cost) for the route from NOR3 to NOR4 for the two topologies during the same timeframe. A route cost of 0 means that there is no route available in the topology table to the specified destination.

Next, NOR1 starts moving followed (a distance behind) by DEU2 in the same area. These vehicles also reach a spot where there is bad connectivity for the UHF radio of Nation 1. Figure 7.6 shows how traffic is again blocked on the *high data rate topology*, and the route cost on the route from NOR3 to NOR1.

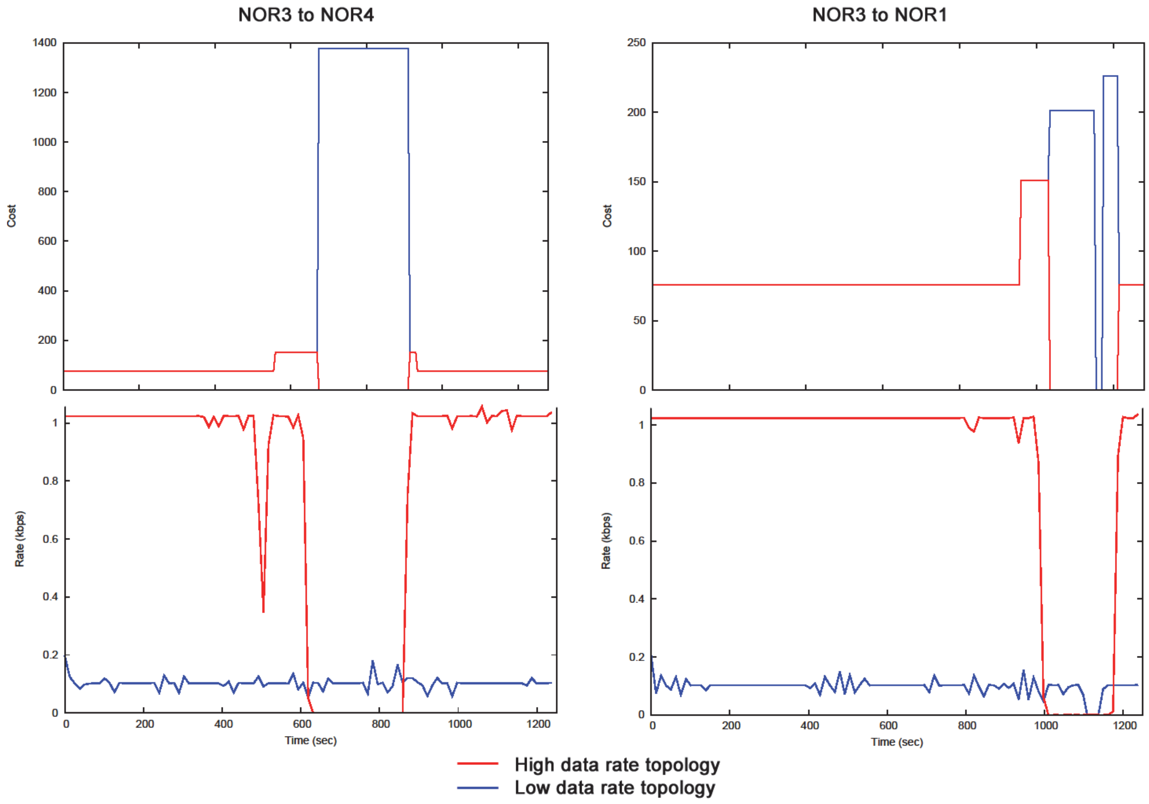

Figure 7.6 Cost of the path to the destination from the source, and the received traffic at the destination for traffic from NOR3 to NOR4 and NOR1.

The test clearly shows how traffic from traffic classes that cannot be supported end-to-end with the current network connectivity, is blocked at the source. With a classic DiffServ architecture and no other admission control mechanisms, the packets would have been sent into the network and dropped at the bottleneck link. The CoNSIS approach removes this garbage traffic from the mobile networks, and thus allows the scarce capacity of these networks to be better utilized.

The test also shows that traffic tagged with different QoS classes can be routed on separate paths through the heterogeneous CoNSIS convoy network. This allows optimal choice of routing path for a QoS type while at the same time preserving the robustness and resource efficiency present in a common heterogeneous transport network.

Normal DiffServ mechanisms for prioritizing, drop precedence and traffic shaping are configured on each network interface. This allows optimal utilization of the network resources in the different radio systems that are present in the convoy network.

7.1.3 MTR-4: Limiting convoy network visibility for adjacent networks.

As described in chapter 4.2, we implemented a flexible mechanism for route redistribution between a MT-routing domain and connected single-topology domains in CoNSIS. This functionality was included in the MT-routing protocol to be able to dynamically connect the MT-routing domain to a backbone network. We also wanted this functionality to reduce the size of the overlay network. If routes from e.g., OLSR could be imported by the MT-routing protocol, we would not need to establish a routing overlay over a radio network running OLSR.

A nice side effect of this flexible mechanism, is that we could to some extent control which routes in the land mobile network to make visible for connected networks. We could, e.g., create a topology for external traffic and transit traffic from external networks and announce only the routes in this topology to, e.g., a BGP protocol.

In the MTR-4 test we demonstrate this behavior. For simplicity we did not create a new topology for this purpose, but instead used the high data rate topology. We configured the MT-router at the gateway (DEU5) to the rest of the CoNSIS network to only make the routes from the high data rate topology available for the BGP protocol instance running at this node. The BGP protocol did a normal redistribution of these routes.

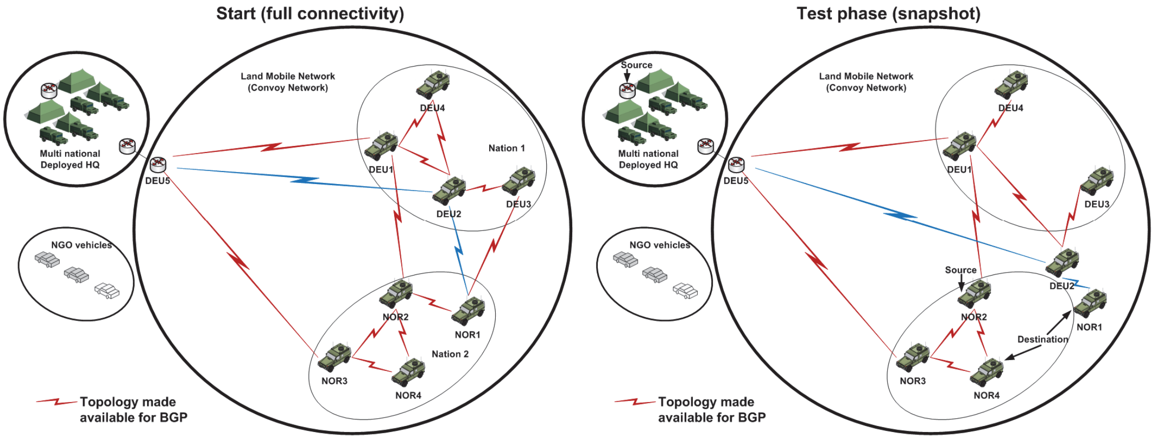
Start (full connectivity)
Multi national Deployed HQ
NGO vehicles
Land Mobile Network (Convoy Network)
DEU5
DEU4
DEU1
DEU2
DEU3
Nation 1
NOR2
NOR1
NOR3
NOR4
Nation 2
Topology made available for BGP
Test phase (snapshot)
Source
Multi national Deployed HQ
NGO vehicles
Land Mobile Network (Convoy Network)
DEU5
DEU4
DEU1
DEU3
DEU2
Source
NOR2
NOR1
Destination
NOR3
NOR4
Topology made available for BGP

Figure 7.7 Network connectivity and sources and destinations for the MTR-4 test.

The test started as all the other tests with minimum cost and full connectivity in all topologies. Best effort traffic was sent from a NOR client in the multinational deployed HQ to all NOR nodes in the convoy. Best effort traffic was also sent from NOR2 to all other nodes in the convoy (Figure 7.7). During one test phase NOR1 loses connectivity on the topology that is being announced to BGP. The route to NOR1 is therefore removed from the routing table in the HQ network. The traffic from NOR2 to NOR1 keeps flowing. Traffic from NOR HQ is also still flowing to other destinations in the convoy network (Figure 7.8).

This test clearly shows that topologies can be useful also for other purposes than to provide differentiated services in a heterogeneous network. A topology can be created to e.g., define which network resources to make available for external traffic and for transit traffic.

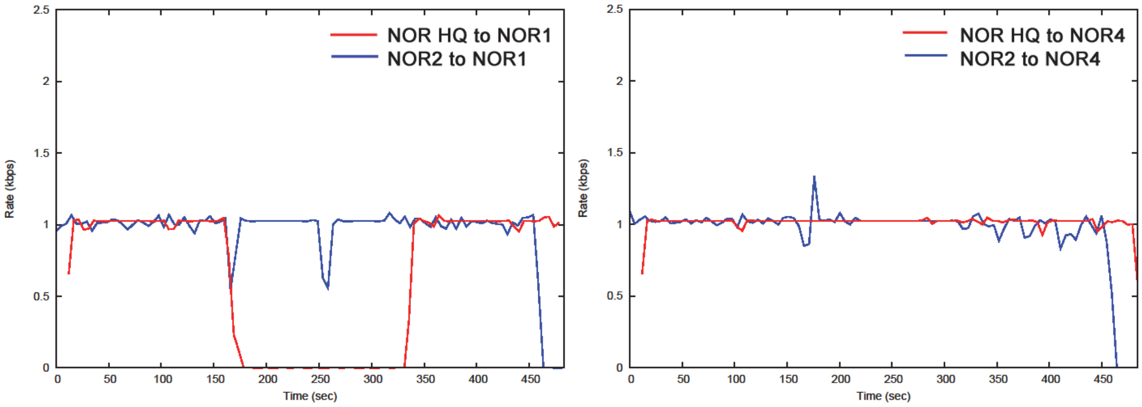

Figure 7.8 Traffic received at NOR1 and NOR4

7.2 Tests during scenario execution in the Greding area

The last days of the field experiment were reserved for a test common to all task groups, where parts of the CoNSIS scenario [9] were played. During this phase the MT-routing overlay provided the network service for traffic from the other tasks in the land mobile network. The MT-routing network represented the unprotected C-TNS (ref Figure 2.2) of the convoy. The traffic on the convoy network from Task 2, “Information and Integration Services (SOA)”, and Task 4, “Management,” was encrypted by the security solutions provided by Task 3, “Security”, prior to entering the convoy transport network. Due to the packet encryption, the network layer in the transport network could not do any packet inspection to deduct the traffic type of the packet to identify the required traffic class. In order for the network layer to provide a differentiated service for the network load, the data packet had to have a traffic class tag made available for inspection.

Unfortunately, Task 2 identified a shortcoming in the Java programming language for IPv6 network handling. The consequence of this problem was that Task2 was not able to tag their services with a traffic class. During the scenario runs we maintained the three QoS topologies as described in the previous test, but only the base topology that handled best effort traffic was used for the traffic load on the network. The best effort class was configured with long queues, which was not the ideal choice for the chat and NATO Friendly Force Information (NFFI) services that were provided by Task 2.

During the scenario runs, we also observed a very high load on the network. As mentioned in chapter 6.3.5, we were not able to copy the traffic class in the IPv6 header to the TOS field in the IPv4 header for GRE tunnels, due to a problem in the Linux kernel of the Vyatta Napa release. This meant that the queue and scheduling mechanisms intended to prioritize packets in overload situations did not work for the interfaces that used an IPv4 carrier. All radio systems in the

CoNSIS field experiment, except Nation2's *UHF Network* (Kongsberg WM600 radio) ran IPv4. For the IPv4 interfaces we were therefore not able to prioritize routing traffic. During the test we observed a very high network load. This unfortunately also led to reduced network stability due to packet loss of routing packets.

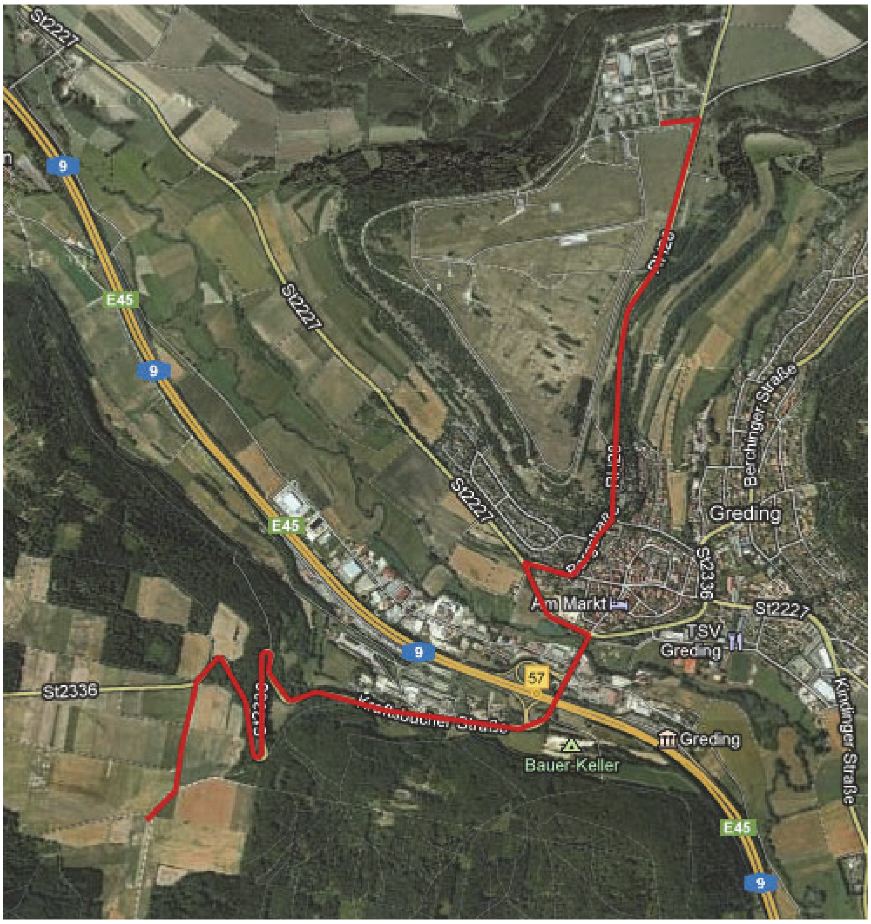

Figure 7.9 The scenario route for the convoy

7.2.1 MTR-2: Demonstrate seamless mobility in a heterogeneous wireless network.

During the scenario plays we logged the routing tables periodically at all Norwegian nodes. We also sent some test traffic between the Norwegian nodes with low intrusiveness during the test.

The route that was used by the vehicles in the scenario plays is depicted in Figure 7.9. In the scenario the DEU (Nation 1) convoy part set off on the scenario route first. After a short time, it lost network connection to the NOR (Nation 2) convoy part. This represents phase 1a in Figure 7.10. Sometime later, the NOR convoy part set out on the same drive and eventually reached a spot where the networks of the two convoy parts merged (phase 1b in Figure 7.10).

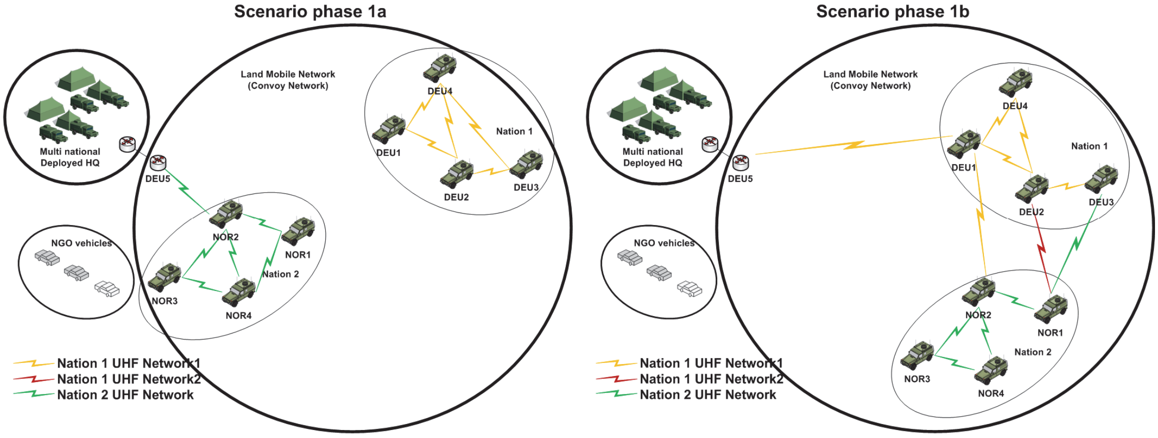

Figure 7.10 Convoy network connectivity during phase 1a and 1b of the scenario

Figure 7.11 shows the cost of three routes between three of the NOR vehicles and three of the DEU vehicles and the cost between the same NOR vehicles and the deployed HQ (DEU5). NOR2 was equipped with the (DEU) *Nation 1 UHF network 1* and DEU3 was equipped with the (NOR) *Nation2 UHF network*. Thus when the convoy parts came within communication range of each other these routes should have a one-hop connection with the cost of the OSPFv3 cost given to the specific radio type. Both of these radio types were also present at DEU5 in the deployed HQ. From the figure we see that most of the time when there is a route available, there is direct connection between the convoy parts. For short periods of time the cost doubles, which means that the route most likely goes via DEU5 in the airfield tower at WTD81. The antennas on this tower were elevated compared to the vehicle mounted antennas, the connectivity to the tower was therefore better than the connectivity between vehicles in the hilly environment. NOR4 and DEU1 were not equipped with compatible radios, hence the cost of this route was always minimum one (DEU) *Nation 1 UHF network 1* hop and one (NOR) *Nation 2 UHF network* hop.

A rough description of the connectivity in the convoy during the scenario play based on the routing table cost and approximate start times given in Figure 7.11 goes as follows: The DEU convoy part starts driving at approximately 250s after test start. At time t=330s the path between the convoy segments is routed via DEU5 in the airfield tower. At time t=360s the DEU convoy segment loses all reach-back connections. Approximately 500s after test start, the NOR convoy part starts driving from the base. A short time after this there is a brief (50s) connection from the DEU convoy via the tower to the NOR convoy. At time t=600s the NOR convoy segment is also isolated without reach-back connection. At time t=685s a direct connection between the two convoy segments is established over a bad radio channel. The NOR convoy part has a reach-back connection to the deployed HQ via the DEU segment. The convoy parts are again separated at time t=750. At time t=840s both convoy segments are reconnected via DEU5 at the tower, and at t=955s a direct radio connection between the convoys is established for the rest of the test period.

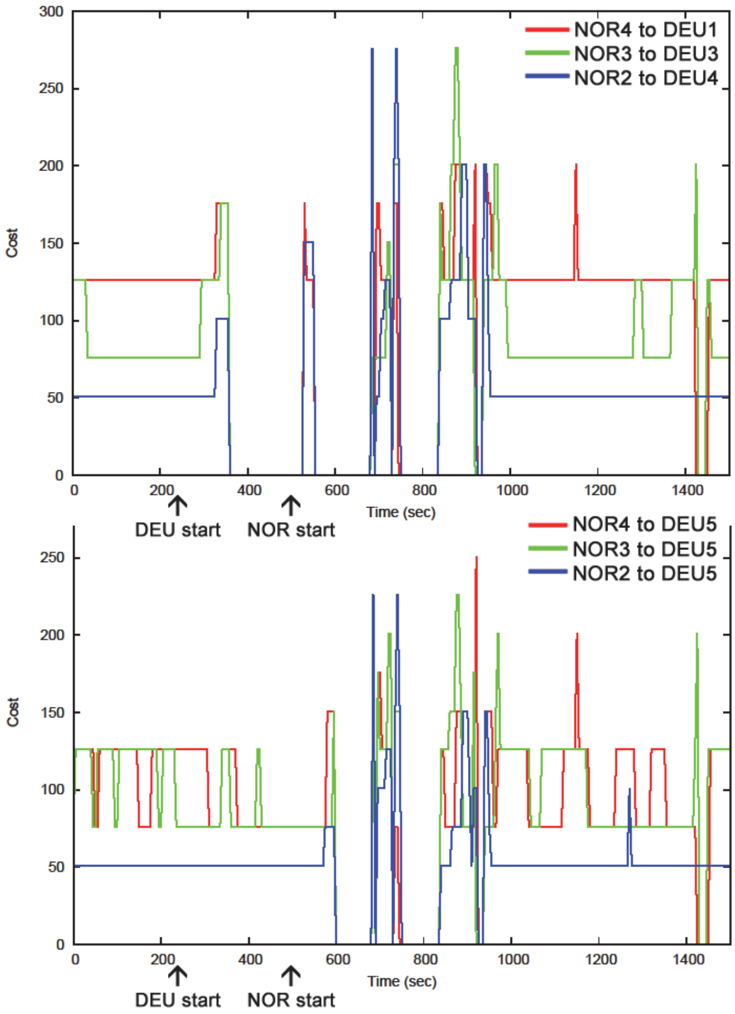

Figure 7.11 Route costs between selected NOR vehicles and selected DEU vehicles, and route cost between NOR vehicles and DEU5 (Deployed HQ). A cost of zero means that there is no route.

This test clearly shows that the MT-routing overlay is able to efficiently utilize all available radio networks in the convoy to connect all nodes in the convoy. The route between NOR4 and DEU1 always has to traverse two different underlying radio networks. We see that this route is updated with the same accuracy as the two other routes that often traverse only one underlying radio network. We see from other logs that it took approximately 10s to establish a new route when a link was lost.

However, we also see that there is a high frequency of route changes. This means that the network is unstable and will show a high percentage of packet loss. We believe that two important reasons for the unstable network connections is the following; first, the three radio types used in this experiment are operating in the UHF frequency band, thus the channels are sensitive to obstacles (threes, buildings, etc.). Second, the high load on the network, and the fact that we were not able to prioritize routing messages on all interfaces, also lead to packet loss of routing messages, and thus erroneous selection of new routing paths.

8 Conclusions

In this report we describe how Multi-Topology routing (MT-routing) can aid the design of end-to-end QoS support in a land mobile network. The MT-routing protocol builds topologies based on static link characteristics that are valid at all times.

The use of multiple topologies, paired with a DiffServ-like architecture, is a simple, but powerful, tool to dynamically block traffic at the source for flows that cannot be supported by the current network topology, and thereby improving the QoS and available capacity for admitted traffic.

The architecture also allows traffic tagged with different QoS classes to be routed on separate paths through the heterogeneous network. This allows optimal choice of the routing path for a QoS class, while at the same time preserving the robustness and resource efficiency present with a common heterogeneous transport network. This mechanism can also enforce some load balancing in the network.

We have also implemented a very flexible interaction between MT-routing network domains and Single-Topology routing (ST-routing) domains for the CoNSIS network architecture.

Since the CoNSIS QoS architecture operates based on the code in the IPv6 *traffic class* field, the only requirement to the IP encryption device placed between the information domain and the wireless transport network is that the encrypted tunnel must inherit the QoS tag of the data packets.

The tests performed in the CoNSIS field test, show that the technology works as expected.

8.1 Lessons learned and future work

The tests performed during the field experiment demonstrated that the MT-supported QoS architecture can be used to improve resource utilization in a heterogeneous mobile network. The multiple topologies give us a tool to control how network resources are used. The signaling cost to support multiple topologies is not big but MT-routing does, however, complicate network configuration. We observed during the tests that it was very easy to make configuration mistakes in the routers. Both the topology configuration and the tunnel configuration for the overlay resulted in a complex configuration file for this fairly small network (only 9 routers and 4 different radio networks). Semi-automated configuration procedures must be in place to reduce the risk of configuration errors on larger networks. Other solutions than a full mesh routing overlay, as discussed in chapter 6.3.2., should also be considered.

In order to run one common routing protocol over radio links with very different transmission characteristics, the timer values in the routing protocol for periodic signaling messages must be tuned. Dissimilar timer values result in an increased risk for inconsistent routing tables. Different transmission delays on the links also lead to the same problem. Inconsistent routing tables mean that there is a high chance for routing loops. We did not measure packet loss due to routing loops explicitly in the field test. We did, however, observe very unstable routes, with lots of route updates during tests with many network topology changes. We also observed that the routing updates kept flowing some time after the network topology had changed. This means that some routing tables were most likely inconsistent some time after the topology changes happen. The conclusion is that it is necessary to do more work to study efficient routing in heterogonous networks.

One other reason for the unstable network during parts of the scenario runs was most likely loss of routing packets. Due to a problem with the Vyatta Napa kernel we were not able to prioritize routing packets in the network. We were able to maintain multiple routes in the network for QoS purposes, but could not distinguish between packets of different QoS classes in the queuing, and scheduling mechanisms on each interface. This underlines the importance of a common cross-layer QoS architecture for optimal network performance.

We also observed that the Kongsberg WM600 radio modified the QoS class of the OSPFv3-MT-routing packets that were transmitted by this radio. It turned out that the radio used the same QoS class (0xC0) for internal routing messages as we used for the overlay routing messages. The policy in the radio was to retag all messages with this tag from external clients to a lower priority traffic class to ensure enough capacity for its own routing messages. Seen from the radio's point of view this was a good idea, but this meant that the MT-routing messages were bundled with traffic from other traffic classes, which was not the intention with the CoNSIS QoS table. This is another example that shows how important it is to have a common QoS architecture operating on all layers and all nodes in the network.

In the current version of the MT-routing protocol, we build topologies based on static predefined link characteristics. The benefit of this is that this value is always a correct "typical value". If there is no route to the destination in the chosen forwarding table, then it is certain that the traffic flow cannot be sustained. If there on the other hand is a route available, it is not certain that there is capacity on this route to sustain the traffic. In future work, we want to investigate if dynamic parameters representing the real time resource situation for the links can be incorporated efficiently with the MT-routing protocol, to better support the resource management mechanism. Alternatively, additional resource management mechanisms based on e.g., polling techniques [50] can be combined with the MT-supported QoS architecture to incorporate dynamic changes in e.g., channel quality and traffic load to further improve the scheme for admission control purposes. The resource mechanism must be executed for all defined topologies.

End-to-end multicast support in a heterogeneous network environment is also a topic that has many unsolved challenges. First and foremost, it is necessary to find solutions to be able to connect different multicast protocols to support end-to-end multicast. Next, differentiated quality of service for multicast traffic should be provided. Multicast is an area where much future work is needed.

Finally, in retrospect, more time should have been allocated to setup of equipment on site before the field experiment, as well as pre-experiment testing and configuration. It is impossible to predict every possible problem that may occur, especially in a setting outside the lab with participants and equipment from multiple nations and different task areas. However, if everything is in place and adequately tested, more of the time allocated for the experiment can more likely be used for the actual experiment. This will also increase the chances of completing the planned tests. Also, while the plan for the two weeks of testing was good and structured, the first two days of preparations should have been better coordinated between the tasks. In addition, more distributed tests between the various labs should have been conducted to resolve more issues in advance. All of these points are the mutual responsibility of the participating nations, and should be brought along into future projects as important lessons learned.

Bibliography

[1] M. Hauge and S. Haavik, *Intelligent Tactical IP Router*, FFI-rapport 2009/01708, Norwegian Defence Research Establishment (FFI), Dec. 2009.

[2] CoNSIS/Task5, *"Task5 - Final Report"*

[3] CoNSIS/Task1/DU/001, *"Final Report – Task 1".*

[4] S. Mirtorabi and A. Roy, "Multi-topology routing in OSPFv3 (MT-OSPFv3)." *draft-ietf-ospf-mt-ospfv3-03.txt (work in progress)*, July 2007

[5] P. Psenak, S. Mirtorabi, A. Roy, L. Nguyen, and P. Pillay-Esnault, "Multi-topology (MT) routing in OSPF." *RFC 4915*, June 2007.

[6] S. Blake et al., "An architecture for differentiated serv." *RFC2475*, 1998.

[7] D. Grossman, "New terminology and clarifications for diffserv." *RFC 3260*, 2002.

[8] M. Hauge, J. Andersson, M. A. Brose, and J. Sander, "Multi-Topology Routing for QoS Support in the CoNSIS Convoy MANET", *MCC* , vol. 1, pp. 179-197, Gdansk, Polen, Oct. 2012.

[9] CoNSIS/Task5/DL/002 "System and Experimentation Architectures", Version 1.0, Sept. 2011.

[10] G. Hallingstad and S. Oudkerk, "Protected core networking: an architectural approach to secure and flexible communications", Communications Magazine, IEEE, vol. 46, No. 11, pp. 35-41, Nov. 2008.

[11] L. Hanzo-II and R. Tafazolli, "A survey of QoS routing solutions for mobile as hoc networks." *COMST*, vol. 9, no. 2, pp.50-70, 2007.

[12] R. Asokan, "A review of Quality of Service (QoS) routing protocols for mobile Ad hoc networks." ICWCSC, Chennai, India, 2010.

[13] N. S. Kulkarni, I. Gupta, and B. Raman, "On demand routing protocols for mobile ad hoc networks: A review." *IACC*, Patiala, India, 2009.

[14] P. Jeon and G. Kesidis, "Pheromone-aided robust multipath and multipriority routing in wireless MANETs." *PE-WASUN*, pp. 106-113, Montreal, Quebec, Canada, 2005.

[15] L. Xuefei and L. Cuthbert, "Multipath QoS routing of supporting DiffServ in mobile ad hoc networks." *SNPD/SAWN*, pp. 308-313, Baltimore, MD, USA, 2005.

[16] S. Venkatasubramanian and N. P. Gopalan, "A QoS-based robust multipath routing protocol for mobile ad hoc networks." *AH-ICI*, pp. 1-7, Kathmandu, Nepal, 2009.

[17] L. Chengyong, L. Kezhong, and L. Layuan, "Research of QoS-aware routing protocol with load balancing for mobile ad hoc networks." *WiCOM*, pp. 1-5, Dalian, China, 2008.

[18] A.F.Hansen, T.Cicic, and P.E.Engelstad, "Profiles and Multi-Topology Routing in Highly Heterogeneous Ad Hoc Networks," *INFOCOM, Poster and Demo session*, Barcelona, Spain, April 2006.

[19] J. A. Stine and G. de Veciana, "A paradigm for quality-of-service in wireless ad hoc networks using synchronous signaling and node states." *J-SAC*, vol. 22, no. 7, pp.1301-1321, Sept. 2004.

[20] S. Bae and T. R. Henderson, "Traffic Engineering with OSPF Multi-Topology Routing," *MILCOM* , Orlando, FL, USA, Oct. 2007.

[21] X. Gou, H. Yan, F. Yi, G. Long, and Q. Wu, "Modeling and simulation of small satellite constellation networking using multi-topology routing," *ICCASM* , vol. 12, pp. 143-147, Taiyuan Shanxi, China, Oct. 2010.

[22] B. Rossow, I. Sorteberg, and M. Hauge, "Multi-topology routing in resilient tactical networks." (NATO Unclassified), *RTO-MP-SCI-18*, Amsterdam, The Netherlands, 2008

[23] M. Hauge, J. Andersson, M. A. Brose, and J. Sander, “Multi-topologyrouting for improved network resource utilization in mobile tactical networks,” *MILCOM*, San Jose, CA, USA, 2010.

[24] F. T. Johnsen, T. Hafsoe, M. Hauge, O. Kolbu, "Cross-layer Quality of Service based admission control for Web services," *HeterWMN*, pp.315-320, Houston, TX, USA, Dec. 2011.

[25] R. Ogier and P. Spagnolo, "Mobile ad hoc network (MANET) extension of OSPF using CDS flooding." *RFC 5614*, Aug. 2009.

[26] Y. Rekhter, T. Li and S. Hares (Ed.’s) " A Border Gateway Protocol 4 (BGP-4)” *RFC 4271*, Jan. 2006

[27] P. Guivarch, “CoNSIS, Addressing and Naming Plan”, CoNSIS/Task 1/D/003, Ver. 1.0, Feb. 2012.

[28] Vyatta, http://www.vyatta.com.

[29] Quagga Routing Suite, http://www.quagga.net.

[30] OSPFv3 MANET MDR, Boeing, http://cs.itd.nrl.navy.mil/work/ospf-manet/.

[31] A. Zinin, A. Roy, L. Nguyen, B. Friedman and D. Yeung “Ospf Link-Local Signaling” *RFC 5613*, Aug. 2009.

[32] Thales Norway AS, Thales Tactical Router - Vyatta Addentum Examples, Report no. 3AQ 26240 AAAA.

[33] ip6tables - IPv6 packet filter administration manual (man) page, http://linux.die.net/man/8/ip6tables.

[34] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol." *RFC 3626*, 2003.

[35] B. Berry, S. Ratliff, E. Paradise, T. Kaiser, and M. Adams, “PPP Over Ethernet (PPPoE) Extensions for Credit Flow and Link Metrics,” *RFC 5578*, Feb. 2010.

[36] S. Ratliff, B. Berry, G. Harrison, S. Jury, and D. Satterwhite, “Dynamic Link Exchange Protocol (DLEP),” *draft-ietf-manet-dlep-03.txt (work in progress)*, Aug. 2012.

[37] D. Dubois, A. Kovummal, B. Petry, and B. Berry, “Radio-Router Control Protocol (R2CP),” draft-dubois-r2cp-00 (work in progress), March 2011.

[38] L. Landmark, K. Øvsthus, and O. Kure, "Routing trade-offs in sparse and mobile heterogeneous multi-radio ad hoc networks,", in proceedings MILCOM , pp. 2229-2236, San Jose, CA, USA, 31 Oct. 2010.

[39] R. M. van Selm, G. Szabo, R. van Engelshoven, and R. Goode, *Ip QoS standardisation fo the NII*, RD-2933, NC3A,(Nato Unclassified), Apr. 2010.

[40] PRIO - Priority qdisc manual (man) page, http://linux.die.net/man/8/tc-prio.

[41] HTB - Hierarchy Token Bucket manual (man) page, http://linux.die.net/man/8/tc-htb.

[42] tc - show / manipulate traffic control settings manual (man) page, http://linux.die.net/man/8/tc.

[43] M. A. Brose and M. Hauge, *Group Communications in mobile military networks*, FFI/Report 2012/00294, Norwegian Defence Research Establishment (FFI), Feb. 2012.

[44] J. Macker(ed.), "Simplified Multicast Forwarding." *RFC6621*, May 2012, http://www.ietf.org.

[45] NRL Simplified Multicast Forwarding (SMF) Engine, http://cs.itd.nrl.navy.mil/work/smf/index.php.

[46] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)." *RFC4601*, 2006, http://www.ietf.org.

[47] C. Danilov, T. R. Henderson, P. A. Spagnolo, T. Goff, and J. H. Kim, "MANET Multicast with Multiple Gateways," *MILCOM* , San Diego, CA, USA, 2008.

[48] Y. Lacharite, W. Maoyu, L. Lamont, and L. Landmark, "A Simplified Approach to Multicast Forwarding Gateways in MANET," *ISWCS* , pp. 426-430, Trondheim, Norway, 17 Oct. 2007.

[49] Multi-Generator (MGEN), http://cs.itd.nrl.navy.mil/work/mgen/index.php.

[50] A. Mohammad, O. Brewer, and A. Ayyagari, "Bandwidth estimation for network quality of service management." *MILCOM*, Orlando, FL, USA, 2007.

Abbreviations

AF	Assured Forwarding
BE	Best Effort
CoS	Class of Service
C-TNS	Coalition Transport Network Segment
CE	Colored Enclave
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
EGP	Exterior Gateway Protocol
EoIP	Ethernet over IP
FEC	Forward Error Correction
GRE	Generic Routing Encapsulation
HF	High Frequency
HTB	Hierarchical Token Bucket
HQ	Headquarters
ICE	Inner Colored Enclave
IGP	Interior Gateway Protocol
IP	Internet Protocol
LAN	Local Area Network
LD-TLV	Link Description Type-Length-Value
LSA	Link State Advertisement
MANET	Mobile ad-hoc Network
MDR	MANET Designated Routers
MLPP	Multi Level Precedence and Priority
MoU	Memorandum of Understanding
MPL	Military Precedence Level
MT	Multi-Topology
NFFI	NATO Friendly Force Information
NGO	Non-Governmental Organization
NII	Network and Information Infrastructure
N-TNS	National Transport Network Segment
OLSR	Optimized Link State Routing
OSPFv3	Open Shortest Path First v3
OSPFv3-MT	Open Shortest Path First v3 - Multi-Topology
PCN	Protected Core Networking
PHY	Physical layer
PIM-SM	Protocol Independent Multicast - Sparse Mode
PRIO	Priority queue
QoS	Quality of Service
RMT-sTLV	Router Multi-Topology sub-Type-Length-Value
SA	Situational Awareness
SBC	Service-Based Classes
SMF	Simplified Multicast Flooding

ST	Single-Topology
SW	Software
TFC	Traffic Flow Confidentiality
TNS	Transport Network Segment
TLV	Type-Length-Value
TOS	Type of Service
UHF	Ultra High Frequency
VHF	Very High Frequency

Appendix A MT-router configuration

A.1 Vyatta configuration file in vehicle NOR4

```
interfaces {
    ethernet eth0 {
        address fc10:f115:200:0004::1/64
        description LAN
        duplex auto
        execute-script LAN-TAG-IPv6-GRE-C
        hw-id 00:10:f3:21:79:c0
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 1
                    description "low bit-rate"
                }
                topology 33 {
                    cost 1
                    description "high bit-rate"
                }
                topology 34 {
                    cost 1
                    description "low delay"
                }
                topology 99 {
                    cost 1
                    description "base topology"
                }
                transmit-delay 1
            }
        }
        smp_affinity auto
        speed auto
        topology-address fc10:f115:200:0004::1/64 {
            topology 99
            topology 32
            topology 33
            topology 34
        }
    }
    ethernet eth1 {
        address fc10:f115:200:14::1/64
        description WM600
        duplex auto
        execute-script Wm600-QoS-IPv6
        hw-id 00:10:f3:21:79:c1
        smp_affinity auto
        speed auto
        topology-address fc10:f115:200:14::1/64 {
            topology 99
        }
    }
    ethernet eth2 {
        address 172.32.204.2/24
        description 7800
        duplex auto
        hw-id 00:10:f3:21:79:c2
        smp_affinity auto
        speed auto
    }
    ethernet eth3 {
        address dhcp
        duplex auto
        hw-id 00:10:f3:21:79:c3
        smp_affinity auto
        speed auto
    }
    loopback lo {
    }
    tunnel tun504 {
        address fc10:f11f:5000:2154::54/64
        description Harris-TUN-DEU1-NOR4
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 80
                hello-interval 20
                instance-id 0
                priority 1
                retransmit-interval 10
                topology 32 {
                    cost 1300
                }
                topology 34 {
                    cost 1300
                }
                topology 99 {
                    cost 1300
                }
                transmit-delay 5
            }
        }
        local-ip 172.32.204.2
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 172.32.201.2
        topology-address fc10:f11f:5000:2154::54/64 {
            topology 99
        }
    }
    tunnel tun507 {
        address fc10:f11f:5000:2254::54/64
        description Harris-TUN-DEU2-NOR4
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 80
                hello-interval 20
                instance-id 0
                priority 1
                retransmit-interval 10
                topology 32 {
                    cost 1300
                }
                topology 34 {
                    cost 1300
                }
                topology 99 {
                    cost 1300
                }
                transmit-delay 5
            }
        }
        local-ip 172.32.204.2
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 172.32.202.2
        topology-address fc10:f11f:5000:2254::54/64 {
            topology 99
        }
    }
    tunnel tun510 {
        address fc10:f11f:5000:2454::54/64
        description Harris-TUN-DEU4-NOR4
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 120
                hello-interval 30
                instance-id 0
                priority 1
                retransmit-interval 20
                topology 32 {
                    cost 1300
                }
                topology 34 {
                    cost 1300
                }
                topology 99 {
                    cost 1300
                }
                transmit-delay 10
            }
        }
        local-ip 172.32.204.2
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 172.32.203.2
        topology-address fc10:f11f:5000:2454::54/64 {
            topology 99
        }
    }
    tunnel tun511 {
        address fc10:f11f:5000:2554::54/64
        description Harris-TUN-DEU5-NOR4
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 160
                hello-interval 40
                instance-id 0
                priority 1
                retransmit-interval 15
                topology 32 {
                    cost 1300
                }
                topology 34 {
                    cost 1300
                }
                topology 99 {
                    cost 1300
                }
                transmit-delay 5
            }
        }
        local-ip 172.32.204.2
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 172.32.205.2
        topology-address fc10:f11f:5000:2554::54/64 {
            topology 99
        }
    }
    tunnel tun804 {
        address fc10:f11f:8000:2354::54/64
        description WM-TUN-DEU3-NOR4
        encapsulation ip6ip6
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 75
                }
                topology 33 {
                    cost 75
                }
                topology 34 {
                    cost 75
                }
                topology 99 {
                    cost 75
                }
                transmit-delay 1
            }
        }
        local-ip fc10:f115:200:14::1
        multicast enable
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:2354::23
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
        remote-ip fc10:f115:200:1f::1
        topology-address fc10:f11f:8000:2354::54/64 {
            topology 32
            topology 33
            topology 34
            topology 99
        }
    }
    tunnel tun808 {
        address fc10:f11f:8000:5154::54/64
        description "WM-TUN NOR4-NOR1"
        encapsulation ip6ip6
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 75
                }
                topology 33 {
                    cost 75
                }
                topology 34 {
                    cost 75
                }
                topology 99 {
                    cost 75
                }
                transmit-delay 1
            }
        }
        local-ip fc10:f115:200:14::1
        multicast disable
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:5154::51
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
        remote-ip fc10:f115:200:11::1
        topology-address fc10:f11f:8000:5154::54/64 {
            topology 99
            topology 32
            topology 33
            topology 34
        }
    }
    tunnel tun811 {
        address fc10:f11f:8000:5254::54/64
        description "WM-TUN NOR2-NOR4"
        encapsulation ip6ip6
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 75
                    description "low bit-rate"
                }
                topology 33 {
                    cost 75
                    description "high bit-rate"
                }
                topology 34 {
                    cost 75
                    description "low delay"
                }
                topology 99 {
                    cost 75
                    description "base topology"
                }
                transmit-delay 1
            }
        }
        local-ip fc10:f115:200:14::1
        multicast disable
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:5254::52
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
        remote-ip fc10:f115:200:12::1
        topology-address fc10:f11f:8000:5254::54/64 {
            topology 99
        }
    }
    tunnel tun813 {
        address fc10:f11f:8000:5354::54/64
        description "WM-TUN NOR3-NOR4"
        encapsulation ip6ip6
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 75
                }
                topology 33 {
                    cost 75
                }
                topology 34 {
                    cost 75
                }
                topology 99 {
                    cost 75
                }
                transmit-delay 1
            }
        }
        local-ip fc10:f115:200:14::1
        multicast disable
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:5354::53
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
        remote-ip fc10:f115:200:13::1
        topology-address fc10:f11f:8000:5354::54/64 {
            topology 99
            topology 32
            topology 33
            topology 34
        }
    }
    tunnel tun815 {
        address fc10:f11f:8000:2554::54/64
        description "WM-TUN NOR4-DEU5"
        encapsulation ip6ip6
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 75
                }
                topology 33 {
                    cost 75
                }
                topology 34 {
                    cost 75
                }
                topology 99 {
                    cost 75
                }
                transmit-delay 1
            }
        }
        local-ip fc10:f115:200:14::1
        multicast disable
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:2554::25
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
        remote-ip fc10:f115:200:15::1
        topology-address fc10:f11f:8000:2554::54/64 {
            topology 99
            topology 32
            topology 33
            topology 34
        }
    }
}
protocols {
    ospfv3 {
        area 0.0.0.0 {
            interface eth0
            interface tun804
            interface tun808
            interface tun811
            interface tun813
            interface tun815
            interface tun511
        }
        parameters {
            disable-rfc2740-compatibility
            router-id 0.0.5.4
        }
    }
    static {
        route 172.32.200.0/21 {
            next-hop 172.32.204.1 {
            }
        }
        table 32 {
            route6 0::0/0 {
                blackhole {
                }
            }
            route6 fc10:f115:200:10::/60 {
                next-hop fc10:f115:200:14::ffff {
                    distance 1
                    interface eth1
                }
            }
        }
        table 33 {
            route6 0::0/0 {
                blackhole {
                }
            }
            route6 fc10:f115:200:10::/60 {
                next-hop fc10:f115:200:14::ffff {
                    distance 1
                    interface eth1
                }
            }
        }
        table 34 {
            route6 0::0/0 {
                blackhole {
                }
            }
            route6 fc10:f115:200:10::/60 {
                next-hop fc10:f115:200:14::ffff {
                    distance 1
                    interface eth1
                }
            }
        }
        table 99 {
            route6 0::0/0 {
                blackhole {
                }
            }
            route6 fc10:f115:200:10::/60 {
                next-hop fc10:f115:200:14::ffff {
                    distance 1
                    interface eth1
                }
            }
        }
    }
}
service {
    nrlsmf {
        parameters "hash MD5"
        parameters ipv6
        parameters "idp off"
        parameters "cf eth0,tap-tun804,tap-tun808,tap-tun811,tap-tun813,tap-
tun815"
    }
    snmp {
        community consis {
            authorization ro
        }
    }
    ssh {
        allow-root
        port 22
        protocol-version v2
    }
}
system {
    config-management {
        commit-revisions 20
    }
    console {
    }
    host-name NOR4
    login {
        user root {
            authentication {
                encrypted-password $1$jv4dUZZk$z1pXwsXqXmh26nAslJyIK1
                plaintext-password ""
            }
            level admin
        }
        user vyatta {
            authentication {
                encrypted-password $1$A8b3h1hS$/amNrYgisDYbZGwGjLgkC/
            }
            level admin
        }
    }
    ntp {
        server 0.vyatta.pool.ntp.org {
        }
        server 1.vyatta.pool.ntp.org {
        }
        server 2.vyatta.pool.ntp.org {
        }
    }
    package {
        auto-sync 1
        repository community {
            components main
            distribution stable
            password ""
            url http://packages.vyatta.com/vyatta
            username ""
        }
    }
    syslog {
        global {
            facility all {
                level notice
            }
            facility protocols {
                level debug
            }
        }
    }
    time-zone Europe/Oslo
}
topology 32 {
    name low-bit-rate
    priority 5
    target ipv6-only
    traffic-class 0x48 {
    }
    traffic-class 0x50 {
    }
}
topology 33 {
    name high-bit-rate
    priority 10
    target ipv6-only
    traffic-class 0x28 {
    }
}
topology 34 {
    name low-delay
    priority 15
    target ipv6-only
    traffic-class 0xb8 {
    }
}
topology 99 {
    catch-all
    name base-topology
    priority 32000
    target ipv6-only
}


/* Warning: Do not remove the following line. */
/* === vyatta-config-version: "dhcp-relay@1:content-
inspection@3:wanloadbalance@3:firewall@4:vrrp@1:conntrack-
sync@1:webproxy@1:zone-policy@1:system@5:quagga@2:qos@1:cluster@1:nat@3:dhcp-
server@4:ipsec@3:webgui@1:config-management@1" === */
/* Release version: 999.mtnapa.02152143 */
```

A.2 Vyatta configuration file in the gateway DEU5

```
interfaces {
    ethernet eth0 {
        address 10.2.205.254/24
        address fc10:f112:5::ffff/48
        description DEU-MLAN-205
        duplex auto
        execute-script LAN-TAG-IPv6-GRE-C
        hw-id 00:01:c0:0c:08:b9
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 1
                }
                topology 33 {
                    cost 1
                }
                topology 34 {
                    cost 1
                }
                topology 99 {
                    cost 1
                }
                transmit-delay 1
            }
        }
        smp_affinity auto
        speed auto
        topology-address fc10:f112:5::ffff/48 {
            topology 32
            topology 33
            topology 34
            topology 99
        }
    }
    ethernet eth1 {
        address 192.168.0.205/24
        description Management_Net_and_Radio_VLANs
        duplex auto
        hw-id 00:01:c0:0c:08:ba
        smp_affinity auto
        speed auto
        vif 11 {
            address 10.2.95.1/24
            description HIMONN
        }
        vif 12 {
            address fc10:f115:200:15::1/64
            description KONGSBERG
            topology-address fc10:f115:200:15::1/64 {
                topology 99
            }
        }
        vif 13 {
            address 172.16.101.1/24
            description FLEXNET
        }
        vif 14 {
            address 172.32.205.2/24
            description HARRIS
        }
        vif 15 {
            address 10.165.165.72/24
            description SAT-BEGAN
        }
        vif 90 {
            address fc10:f230:1::205/48
            address 10.23.1.205/24
            description UPLINK_TO_ROUTER30
        }
    }
    loopback lo {
    }
    tunnel tun402 {
        address fc10:f11f:4000:2125::25/64
        description SatCom-Tunnel_DEU-1_DEU-5
        disable
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 40
                hello-interval 10
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 500
                }
                topology 99 {
                    cost 500
                }
                transmit-delay 1
            }
        }
        local-ip 10.165.165.72
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 10.165.167.2
        topology-address fc10:f11f:4000:2125::25/64 {
            topology 99
        }
    }
    tunnel tun701 {
        address fc10:f11f:7000:2225::25/64
        description FlexNet-Tunnel_DEU-2_DEU-5
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 20
                hello-interval 5
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 150
                }
                topology 34 {
                    cost 150
                }
                topology 99 {
                    cost 150
                }
                transmit-delay 1
            }
        }
        local-ip 172.16.101.1
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 172.16.102.1
        topology-address fc10:f11f:7000:2225::25/64 {
            topology 99
        }
    }
    tunnel tun703 {
        address fc10:f11f:7000:2551::25/64
        description FlexNet-Tunnel_DEU-5_NOR-1
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 20
                hello-interval 5
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 150
                }
                topology 34 {
                    cost 150
                }
                topology 99 {
                    cost 150
                }
                transmit-delay 1
            }
        }
        local-ip 172.16.101.1
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 172.16.103.1
        topology-address fc10:f11f:7000:2551::25/64 {
            topology 99
        }
    }
    tunnel tun805 {
        address fc10:f11f:8000:2325::25/64
        description Kongsberg-Tunnel_DEU-3_DEU-5
        encapsulation ip6ip6
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 75
                }
                topology 33 {
                    cost 75
                }
                topology 34 {
                    cost 75
                }
                topology 99 {
                    cost 75
                }
                transmit-delay 1
            }
        }
        local-ip fc10:f115:200:15::1
        multicast enable
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:2325::23
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
        remote-ip fc10:f115:200:1f::1
        topology-address fc10:f11f:8000:2325::25/64 {
            topology 32
            topology 33
            topology 34
            topology 99
        }
    }
    tunnel tun809 {
        address fc10:f11f:8000:2551::25/64
        description Kongsberg-Tunnel_DEU-5_NOR-1
        encapsulation ip6ip6
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 75
                }
                topology 33 {
                    cost 75
                }
                topology 34 {
                    cost 75
                }
                topology 99 {
                    cost 75
                }
                transmit-delay 1
            }
        }
        local-ip fc10:f115:200:15::1
        multicast enable
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:2551::51
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
        remote-ip fc10:f115:200:11::1
        topology-address fc10:f11f:8000:2551::25/64 {
            topology 32
            topology 33
            topology 34
            topology 99
        }
    }
    tunnel tun812 {
        address fc10:f11f:8000:2552::25/64
        description Kongsberg-Tunnel_DEU-5_NOR-2
        encapsulation ip6ip6
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 75
                }
                topology 33 {
                    cost 75
                }
                topology 34 {
                    cost 75
                }
                topology 99 {
                    cost 75
                }
                transmit-delay 1
            }
        }
        local-ip fc10:f115:200:15::1
        multicast enable
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:2552::52
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
        remote-ip fc10:f115:200:12::1
        topology-address fc10:f11f:8000:2552::25/64 {
            topology 32
            topology 33
            topology 34
            topology 99
        }
    }
    tunnel tun814 {
        address fc10:f11f:8000:2553::25/64
        description Kongsberg-Tunnel_DEU-5_NOR-3
        encapsulation ip6ip6
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 75
                }
                topology 33 {
                    cost 75
                }
                topology 34 {
                    cost 75
                }
                topology 99 {
                    cost 75
                }
                transmit-delay 1
            }
        }
        local-ip fc10:f115:200:15::1
        multicast enable
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:2553::53
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
        remote-ip fc10:f115:200:13::1
        topology-address fc10:f11f:8000:2553::25/64 {
            topology 32
            topology 33
            topology 34
            topology 99
        }
    }
    tunnel tun815 {
        address fc10:f11f:8000:2554::25/64
        description Kongsberg-Tunnel_DEU-5_NOR-4
        encapsulation ip6ip6
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 75
                }
                topology 33 {
                    cost 75
                }
                topology 34 {
                    cost 75
                }
                topology 99 {
                    cost 75
                }
                transmit-delay 1
            }
        }
        local-ip fc10:f115:200:15::1
        multicast enable
        parameters {
            ipv6 {
                enable-ip6ip6tap {
                    remote-ip fc10:f11f:8000:2554::54
                    udp-port 5016
                }
                encaplimit 4
                flowlabel 0x00000
                hoplimit 3
                tclass inherit
            }
        }
        remote-ip fc10:f115:200:14::1
        topology-address fc10:f11f:8000:2554::25/64 {
            topology 32
            topology 33
            topology 34
            topology 99
        }
    }
    tunnel tun904 {
        address 10.254.15.96/24
        address fc10:f11f:9000:2125::25/64
        description HiMoNN-Tunnel_DEU-1_DEU-5
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 50
                }
                topology 33 {
                    cost 50
                }
                topology 34 {
                    cost 50
                }
                topology 99 {
                    cost 50
                }
                transmit-delay 1
            }
        }
        local-ip 10.2.95.1
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 10.2.99.1
        topology-address fc10:f11f:9000:2125::25/64 {
            topology 99
        }
    }
    tunnel tun908 {
        address 10.254.25.96/24
        address fc10:f11f:9000:2225::25/64
        description HiMoNN-Tunnel_DEU-2_DEU-5
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 50
                }
                topology 33 {
                    cost 50
                }
                topology 34 {
                    cost 50
                }
                topology 99 {
                    cost 50
                }
                transmit-delay 1
            }
        }
        local-ip 10.2.95.1
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 10.2.98.1
        topology-address fc10:f11f:9000:2225::25/64 {
            topology 99
        }
    }
    tunnel tun911 {
        address 10.254.35.96/24
        address fc10:f11f:9000:2325::25/64
        description HiMoNN-Tunnel_DEU-3_DEU-5
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 50
                }
                topology 33 {
                    cost 50
                }
                topology 34 {
                    cost 50
                }
                topology 99 {
                    cost 50
                }
                transmit-delay 1
            }
        }
        local-ip 10.2.95.1
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 10.2.97.1
        topology-address fc10:f11f:9000:2325::25/64 {
            topology 99
        }
    }
    tunnel tun913 {
        address 10.254.45.96/24
        address fc10:f11f:9000:2425::25/64
        description HiMoNN-Tunnel_DEU-4_DEU-5
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 50
                }
                topology 33 {
                    cost 50
                }
                topology 34 {
                    cost 50
                }
                topology 99 {
                    cost 50
                }
                transmit-delay 1
            }
        }
        local-ip 10.2.95.1
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 10.2.96.1
        topology-address fc10:f11f:9000:2425::25/64 {
            topology 99
        }
    }
    tunnel tun915 {
        address fc10:f11f:9000:2552::25/64
        description HiMoNN-Tunnel_DEU-5_NOR-2
        encapsulation gre
        ipv6 {
            dup-addr-detect-transmits 1
            ospfv3 {
                cost 1
                dead-interval 8
                hello-interval 2
                instance-id 0
                priority 1
                retransmit-interval 5
                topology 32 {
                    cost 50
                }
                topology 33 {
                    cost 50
                }
                topology 34 {
                    cost 50
                }
                topology 99 {
                    cost 50
                }
                transmit-delay 1
            }
        }
        local-ip 10.2.95.1
        multicast enable
        parameters {
            ip {
                ttl 3
            }
        }
        remote-ip 10.2.94.1
        topology-address fc10:f11f:9000:2552::25/64 {
            topology 99
        }
    }
    wireless wlan0 {
        hw-id 00:0d:f0:95:87:d1
        mode g
        physical-device phy0
        type monitor
    }
}
policy {
    prefix-list6 bgp {
        rule 1 {
            action permit
            prefix 2001:700::1/64
        }
        rule 2 {
            action permit
            prefix fc10:ffff::/32
        }
        rule 3 {
            action permit
            prefix fc10:3::/32
        }
        rule 4 {
            action permit
            prefix fc10:f300::/24
        }
        rule 5 {
            action permit
            prefix fc10:f240::/32
        }
        rule 6 {
            action permit
            prefix fc10:f210:0000::/40
        }
        rule 7 {
            action permit
            prefix fc10:f220:0000::/40
        }
        rule 8 {
            action permit
            prefix fc10:f600::/24
        }
    }
    prefix-list6 tunnel {
        rule 1 {
            action permit
            le 128
            prefix fc10:f11f::/32
        }
    }
    route-map CISCO-import {
        rule 10 {
            action deny
            match {
                ipv6 {
                    address {
                        prefix-list bgp
                    }
                }
            }
        }
        rule 20 {
            action permit
        }
    }
    route-map VYATTA-export {
        rule 10 {
            action deny
            match {
                ipv6 {
                    address {
                        prefix-list tunnel
                    }
                }
            }
        }
        rule 20 {
            action permit
        }
    }
}
protocols {
    bgp 64805 {
        address-family {
            ipv6-unicast {
                network fc10:f112:5::/48 {
                }
                redistribute {
                    ospfv3 {
                        route-map VYATTA-export
                    }
                }
            }
        }
        neighbor fc10:f230:1::ffff {
            address-family {
                ipv6-unicast {
                    route-map {
                        import CISCO-import
                    }
                }
            }
            remote-as 64853
        }
        parameters {
            disable-network-import-check
            log-neighbor-changes
            router-id 2.0.0.5
        }
    }
    ospf {
        area 0 {
            network 10.2.205.0/24
            network 10.254.15.0/24
            network 10.254.25.0/24
            network 10.254.35.0/24
            network 10.254.45.0/24
        }
        parameters {
            abr-type cisco
            router-id 2.0.0.5
        }
    }
    ospfv3 {
        area 0.0.0.0 {
            interface eth0
            interface tun701
            interface tun703
            interface tun805
            interface tun809
            interface tun812
            interface tun814
            interface tun815
            interface tun904
            interface tun908
            interface tun911
            interface tun913
            interface tun915
            range fc10:f112:5::/48 {
            }
        }
        parameters {
            disable-rfc2740-compatibility
            router-id 2.0.0.5
        }
        topology 32 {
            redistribute {
                bgp {
                }
            }
        }
        topology 33 {
            redistribute {
                bgp {
                }
            }
        }
        topology 34 {
            redistribute {
                bgp {
                }
            }
        }
        topology 99 {
            redistribute {
                bgp {
                }
            }
        }
    }
    static {
```

```
route 0.0.0.0/0 {
    next-hop 10.23.1.254 {
    }
}
route 10.2.94.0/24 {
    next-hop 10.2.95.254 {
    }
}
route 10.2.96.0/24 {
    next-hop 10.2.95.254 {
    }
}
route 10.2.97.0/24 {
    next-hop 10.2.95.254 {
    }
}
route 10.2.98.0/24 {
    next-hop 10.2.95.254 {
    }
}
route 10.2.99.0/24 {
    next-hop 10.2.95.254 {
    }
}
route 10.2.200.0/24 {
    next-hop 10.23.1.254 {
    }
}
route 10.165.167.0/24 {
    next-hop 10.165.165.71 {
    }
}
route 172.16.0.0/16 {
    next-hop 172.16.101.201 {
    }
}
route 172.32.0.0/16 {
    next-hop 172.32.205.1 {
    }
}
route6 0::0/0 {
    next-hop fc10:f230:1::ffff {
    }
}
table 32 {
    route6 ::/0 {
        blackhole {
        }
    }
    route6 fc10:f115:200:10::/60 {
        next-hop fc10:f115:200:15::ffff {
            interface eth1.12
        }
    }
    route6 fc10:f200::/24 {
        next-hop fc10:f230:1::ffff {
            distance 1
            interface eth1.90
        }
    }
}
table 33 {
    route6 ::/0 {
        blackhole {
                }
            }
            route6 fc10:f115:200:10::/60 {
                next-hop fc10:f115:200:15::ffff {
                    interface eth1.12
                }
            }
            route6 fc10:f200::/24 {
                next-hop fc10:f230:1::ffff {
                    distance 1
                    interface eth1.90
                }
            }
        }
        table 34 {
            route6 ::/0 {
                blackhole {
                }
            }
            route6 fc10:f115:200:10::/60 {
                next-hop fc10:f115:200:15::ffff {
                    distance 1
                    interface eth1.12
                }
            }
            route6 fc10:f200::/24 {
                next-hop fc10:f230:1::ffff {
                    distance 1
                    interface eth1.90
                }
            }
        }
        table 99 {
            route6 ::/0 {
                blackhole {
                }
            }
            route6 fc10:f115:200:10::/60 {
                next-hop fc10:f115:200:15::ffff {
                    interface eth1.12
                }
            }
            route6 fc10:f200::/24 {
                next-hop fc10:f230:1::ffff {
                    distance 1
                    interface eth1.90
                }
            }
        }
    }
}
service {
    nrlsmf {
        parameters "idp off"
        parameters "hash MD5"
        parameters ipv6
        parameters "cf eth0,eth1.90,tun701,tun703,tap-tun809,tap-tun812,tap-
tun814,tap-tun815,tun904,tun908,tun911,tun913,tun915"
    }
    snmp {
        community CoNSISnmp {
            authorization rw
        }
        community consis {
            authorization ro
        }
    }
    ssh {
        port 22
        protocol-version v2
    }
}
system {
    config-management {
        commit-revisions 20
    }
    console {
        device ttyS0 {
            speed 9600
        }
    }
    host-name DEU-MTR-5
    login {
        user vyatta {
            authentication {
                encrypted-password ****************
                plaintext-password ****************
            }
            level admin
        }
    }
    package {
        auto-sync 1
        repository community {
            components main
            distribution stable
            password ****************
            url http://packages.vyatta.com/vyatta
            username ""
        }
    }
    syslog {
        global {
            facility all {
                level notice
            }
            facility protocols {
                level debug
            }
        }
    }
    time-zone GMT
}
topology 32 {
    clone bgp
    name low-bit-rate
    priority 5
    target ipv6-only
    traffic-class 0x48 {
    }
    traffic-class 0x50 {
    }
}
topology 33 {
    clone bgp
    name high-bit-rate
    priority 10
    target ipv6-only
    traffic-class 0x28 {
    }
}
topology 34 {
    clone bgp
    name low-delay
    priority 15
    target ipv6-only
    traffic-class 0xb8 {
    }
}
topology 99 {
    catch-all
    clone ospfv3
    clone bgp
    name base-topology
    priority 32000
    target ipv6-only
}
```

Appendix B Scripts to setup the interface queues

B.1 IABG HiMoNN (Nation 1 UHF Network 1)

```
#!/bin/sh
TC=/sbin/tc
IPTABLES=/sbin/iptables

. /lib/lsb/init-functions

: ${vyatta_env:=/etc/default/vyatta}
source $vyatta_env

declare progname=${0##*/}
declare action=$1; shift

# The following parameters apply to HiMoNN

declare IF=$1; shift
let RATE0=5000
declare RATE=${RATE0}kbit
declare CEILING=4000kbit
let RATE1=$RATE0*15/100
let RATE2=$RATE0/10
let RATE3=$RATE0*4/10
let RATE4=$RATE0*10/100
let RATE5=$RATE0*25/100
declare RATE11=${RATE1}kbit
declare RATE12=${RATE2}kbit
declare RATE13=${RATE3}kbit
declare RATE14=${RATE4}kbit
declare RATE15=${RATE5}kbit

###[ ! -d "/sys/class/net/$IF" ] && logger -p error -t TTR-QOS "interface $IF
not available" && exit 0


stop()
{
    # Remove (reset) qdisc on this interface
    $TC qdisc del dev $IF root 2> /dev/null > /dev/null
}

start()
{
    # Define htb as the qdisc on the interface.
    # Classid 1:1x associate a minimum rate to classid 1:1x, the classid can be
given up to $CEILING rate.
    # Classid 1:1x will share unused capacity according the the minimum rate
share, not according to priority (since prio is not included in the command)
    echo "Installing HTB root on HiMoNN"
    $TC qdisc add dev $IF root handle 1: htb default 16
    $TC class add dev $IF parent 1: classid 1:1 htb rate $RATE ceil $RATE
    $TC class add dev $IF parent 1:1 classid 1:11 htb rate $RATE11 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:12 htb rate $RATE12 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:13 htb rate $RATE13 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:14 htb rate $RATE14 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:15 htb rate $RATE15 ceil $CEILING

    # The filters pick the traffic that is associated with each classid 1:1x.
    # Prio specifies the order which the filters are executed.
    echo "Setting up IPv4 flow rules for traffic"
    $TC filter add dev $IF protocol ip parent 1:0 prio 1 u32 match ip tos 0xc0
0xff flowid 1:11
    $TC filter add dev $IF protocol ip parent 1:0 prio 3 u32 match ip tos 0xb8
0xff flowid 1:12
    $TC filter add dev $IF protocol ip parent 1:0 prio 3 u32 match ip tos 0x48
0xff flowid 1:13
    $TC filter add dev $IF protocol ip parent 1:0 prio 3 u32 match ip tos 0x50
0xff flowid 1:13
    $TC filter add dev $IF protocol ip parent 1:0 prio 3 u32 match ip tos 0x28
0xff flowid 1:14
    $TC filter add dev $IF protocol ip parent 1:0 prio 4 u32 match ip dst
0.0.0.0/0 flowid 1:15

    # Switch the default qdisc for classid 1:1x with pfifo (packet fifo) where
the queuelenght in packets are given with limit p.
    $TC qdisc add dev $IF parent 1:11 handle 11: pfifo limit 10
    $TC qdisc add dev $IF parent 1:12 handle 12: pfifo limit 5
    $TC qdisc add dev $IF parent 1:13 handle 13: pfifo limit 20
    $TC qdisc add dev $IF parent 1:14 handle 14: pfifo limit 50
    $TC qdisc add dev $IF parent 1:15 handle 15: pfifo limit 100

#    echo "Listing the queues:"
#    $TC -s qdisc ls dev $IF
#    $TC -s filter ls dev $IF

}

case "$action" in
    start) start ;;
    stop)  stop ;;
    restart|force-reload) stop && start ;;
    *)    log_failure_msg "usage: $progname [ start|stop|restart ] interface" ;
          false ;;
esac

exit $?
```

B.2 Rockwell Collins, FlexNet-Four (Nation 1 UHF Network 2)

```
#!/#!/bin/sh
TC=/sbin/tc
IPTABLES=/sbin/iptables

. /lib/lsb/init-functions

: ${vyatta_env:=/etc/default/vyatta}
source $vyatta_env

declare progname=${0##*/}
declare action=$1; shift

# The following parameters apply to FlexNet

declare IF=$1; shift
let RATE0=350
declare RATE=${RATE0}kbit
declare CEILING=300kbit
let RATE1=$RATE0*15/100
let RATE2=$RATE0/10
let RATE3=$RATE0*4/10
let RATE4=$RATE0*10/100
let RATE5=$RATE0*25/100
declare RATE11=${RATE1}kbit
declare RATE12=${RATE2}kbit
declare RATE13=${RATE3}kbit
declare RATE14=${RATE4}kbit
declare RATE15=${RATE5}kbit

###[ ! -d "/sys/class/net/$IF" ] && logger -p error -t TTR-QOS "interface $IF
not available" && exit 0


stop()
{
    # Remove (reset) qdisc on this interface
    $TC qdisc del dev $IF root 2> /dev/null > /dev/null
}

start()
{
    # Define htb as the qdisc on the interface.
    # Classid 1:1x associate a minimum rate to classid 1:1x, the classid can be
given up to $CEILING rate.
    # Classid 1:1x will share unused capacity according the the minimum rate
share, not according to priority (since prio is not included in the command)
    echo "Installing HTB root on FlexNet"
    $TC qdisc add dev $IF root handle 1: htb default 16
    $TC class add dev $IF parent 1: classid 1:1 htb rate $RATE ceil $RATE
    $TC class add dev $IF parent 1:1 classid 1:11 htb rate $RATE11 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:12 htb rate $RATE12 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:13 htb rate $RATE13 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:14 htb rate $RATE14 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:15 htb rate $RATE15 ceil $CEILING

    # The filters pick the traffic that is associated with each classid 1:1x.
    # Prio specifies the order which the filters are executed.
    echo "Setting up IPv4 flow rules for traffic"
    $TC filter add dev $IF protocol ip parent 1:0 prio 1 u32 match ip tos 0xc0
0xff flowid 1:11
    $TC filter add dev $IF protocol ip parent 1:0 prio 3 u32 match ip tos 0xb8
0xff flowid 1:12
    $TC filter add dev $IF protocol ip parent 1:0 prio 3 u32 match ip tos 0x48
0xff flowid 1:13
    $TC filter add dev $IF protocol ip parent 1:0 prio 3 u32 match ip tos 0x50
0xff flowid 1:13
    $TC filter add dev $IF protocol ip parent 1:0 prio 3 u32 match ip tos 0x28
0xff flowid 1:14
    $TC filter add dev $IF protocol ip parent 1:0 prio 4 u32 match ip dst
0.0.0.0/0 flowid 1:15

    # Switch the default qdisc for classid 1:1x with pfifo (packet fifo) where
the queuelenght in packets are given with limit p.
    $TC qdisc add dev $IF parent 1:11 handle 11: pfifo limit 10
    $TC qdisc add dev $IF parent 1:12 handle 12: pfifo limit 5
    $TC qdisc add dev $IF parent 1:13 handle 13: pfifo limit 20
    $TC qdisc add dev $IF parent 1:14 handle 14: pfifo limit 50
    $TC qdisc add dev $IF parent 1:15 handle 15: pfifo limit 100

#    echo "Listing the queues:"
#    $TC -s qdisc ls dev $IF
#    $TC -s filter ls dev $IF

}

case "$action" in
    start) start ;;
    stop)  stop ;;
    restart|force-reload) stop && start ;;
    *)      log_failure_msg "usage: $progname [ start|stop|restart ] interface" ;
            false ;;
esac

exit $?
```

B.3 Harris, RF-7800S (Nation 1 VHF Network)

```
#!/bin/sh

IPTABLES=/sbin/iptables
TC=/sbin/tc
LOGGER=/usr/bin/logger

#Write to syslog
$LOGGER -s Read Harris-QoS

## HarrisHF-Qos; version 1.1 htb

. /lib/lsb/init-functions

: ${vyatta_env:=/etc/default/vyatta}
source $vyatta_env

declare progname=${0##*/}
declare action=$1; shift

# The following parameters apply to Harris VHF
declare IF=$1; shift
declare RATE=30kbit


stop()
{
    # Remove (reset) qdisc on this interface
    $TC qdisc del dev $IF root 2> /dev/null > /dev/null
}

start()
{
    ## Define htb as root qdisc with rate equal RATE
    echo "Installing HTB root on Harris"
    $TC qdisc add dev $IF root handle 1: htb default 1
    $TC class add dev $IF parent 1: classid 1:1 htb rate $RATE quantum 1500

    ## Define qdisc for class 1:1 to be prio with 4 priorities
    echo "Installing PRIO parent 1:1 with 4 priorities"
    $TC qdisc add dev $IF parent 1:1 handle 2: prio bands 4 priomap 3 3 3 3 3 3
3 3 3 3 3 3 3 3 3 3

    ## define "leaf" qdisc to be pfifo (packet fifo) where queue lenght is
specified in number of packets with "limit p".
    echo "Adding pFIFO queues"
    $TC qdisc add dev $IF parent 2:1 handle 10: pfifo limit 10
    $TC qdisc add dev $IF parent 2:2 handle 20: pfifo limit 5
    $TC qdisc add dev $IF parent 2:3 handle 30: pfifo limit 10
    $TC qdisc add dev $IF parent 2:4 handle 40: pfifo limit 60

    ## Associates traffic to the queues.
    echo "Setting up ip flow rules for traffic"
    $TC filter add dev $IF protocol ip parent 2:0 prio 1 u32 match ip tos 0xC0
0xff flowid 2:1
    $TC filter add dev $IF protocol ip parent 2:0 prio 2 u32 match ip tos 0xb8
0xff flowid 2:2
```

```
    $TC filter add dev $IF protocol ip parent 2:0 prio 2 u32 match ip tos 0x28
0xff flowid 2:3
    $TC filter add dev $IF protocol ip parent 2:0 prio 2 u32 match ip tos 0x48
0xff flowid 2:3
    $TC filter add dev $IF protocol ip parent 2:0 prio 2 u32 match ip tos 0x50
0xff flowid 2:3
    $TC filter add dev $IF protocol ip parent 2:0 prio 3 u32 match ip dst
0.0.0.0/0 flowid 2:4

#    echo "Listing the queues:"
#    $TC -s qdisc ls dev $IF
#    $TC -s filter ls dev $IF

}

case "$action" in
    start) start ;;
    stop)  stop ;;
    restart|force-reload) stop && start ;;
    *)     log_failure_msg "usage: $progname [ start|stop|restart ] interface" ;
           false ;;
esac

exit $?
```

B.4 Thrane &Thrane BGAN Explorer 727 (Nation 1 SatCom)

```
#!/bin/sh

IPTABLES=/sbin/iptables
TC=/sbin/tc
LOGGER=/usr/bin/logger

#Write to syslog
$LOGGER -s Read BGAN-QoS

## BGAN-Qos; version 1.1 htb

. /lib/lsb/init-functions

: ${vyatta_env:=/etc/default/vyatta}
source $vyatta_env

declare progname=${0##*/}
declare action=$1; shift

# The following parameters apply to BGAN
declare IF=$1; shift
declare RATE=64kbit


stop()
{
    # Remove (reset) qdisc on this interface
    $TC qdisc del dev $IF root 2> /dev/null > /dev/null
}

start()
{
    ## Define htb as root qdisc with rate equal RATE
    echo "Installing HTB root on BGAN"
    $TC qdisc add dev $IF root handle 1: htb default 1
    $TC class add dev $IF parent 1: classid 1:1 htb rate $RATE quantum 1500
    ## Define qdisc for class 1:1 to be prio with 4 priorities
    echo "Installing PRIO parent 1:1 with 4 priorities"
    $TC qdisc add dev $IF parent 1:1 handle 2: prio bands 4 priomap 3 3 3 3 3 3
3 3 3 3 3 3 3 3 3 3

    ## define "leaf" qdisc to be pfifo (packet fifo) where queue lenght is
specified in number of packets with "limit p".
    echo "Adding pFIFO queues"
    $TC qdisc add dev $IF parent 2:1 handle 10: pfifo limit 10
    $TC qdisc add dev $IF parent 2:2 handle 20: pfifo limit 5
    $TC qdisc add dev $IF parent 2:3 handle 30: pfifo limit 10
    $TC qdisc add dev $IF parent 2:4 handle 40: pfifo limit 60

    ## Associates traffic to the queues.
    echo "Setting up ip flow rules for traffic"
    $TC filter add dev $IF protocol ip parent 2:0 prio 1 u32 match ip tos 0xC0
0xff flowid 2:1
    $TC filter add dev $IF protocol ip parent 2:0 prio 2 u32 match ip tos 0xb8
0xff flowid 2:2
    $TC filter add dev $IF protocol ip parent 2:0 prio 2 u32 match ip tos 0x28
0xff flowid 2:3
    $TC filter add dev $IF protocol ip parent 2:0 prio 2 u32 match ip tos 0x48
0xff flowid 2:3
    $TC filter add dev $IF protocol ip parent 2:0 prio 2 u32 match ip tos 0x50
0xff flowid 2:3
    $TC filter add dev $IF protocol ip parent 2:0 prio 3 u32 match ip dst
0.0.0.0/0 flowid 2:4

#    echo "Listing the queues:"
#    $TC -s qdisc ls dev $IF
#    $TC -s filter ls dev $IF

}

case "$action" in
    start) start ;;
    stop)  stop ;;
    restart|force-reload) stop && start ;;
    *)     log_failure_msg "usage: $progname [ start|stop|restart ] interface" ;
           false ;;
esac

exit $?
```

B.5 Kongsberg, WM600 (Nation 2 UHF Network)

```
#!/bin/sh
TC=/sbin/tc
IPTABLES=/sbin/ip6tables

. /lib/lsb/init-functions

: ${vyatta_env:=/etc/default/vyatta}
source $vyatta_env

declare progname=${0##*/}
declare action=$1; shift

# The following parameters apply to WM600

declare IF=$1; shift
let RATE0=1000
declare RATE=${RATE0}kbit
declare CEILING=800kbit
```

```
let RATE1=$RATE0*15/100
let RATE2=$RATE0/10
let RATE3=$RATE0*4/10
let RATE4=$RATE0*10/100
let RATE5=$RATE0*25/100
declare RATE11=${RATE1}kbit
declare RATE12=${RATE2}kbit
declare RATE13=${RATE3}kbit
declare RATE14=${RATE4}kbit
declare RATE15=${RATE5}kbit

###[ ! -d "/sys/class/net/$IF" ] && logger -p error -t TTR-QOS "interface $IF
not available" && exit 0


stop()
{
    # Remove (reset) qdisc on this interface
    $TC qdisc del dev $IF root 2> /dev/null > /dev/null
}

start()
{
    # Define htb as the qdisc on the interface.
    # Classid 1:1x associate a minimum rate to classid 1:1x, the classid can be
given up to $CEILING rate.
    # Classid 1:1x will share unused capacity according the the minimum rate
share, not according to priority (since prio is not included in the command)
    echo "Installing HTB root on WM600"
    $TC qdisc add dev $IF root handle 1: htb default 16
    $TC class add dev $IF parent 1: classid 1:1 htb rate $RATE ceil $RATE
    $TC class add dev $IF parent 1:1 classid 1:11 htb rate $RATE11 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:12 htb rate $RATE12 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:13 htb rate $RATE13 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:14 htb rate $RATE14 ceil $CEILING
    $TC class add dev $IF parent 1:1 classid 1:15 htb rate $RATE15 ceil $CEILING

    # The filters pick the traffic that is associated with each classid 1:1x.
    # Prio specifies the order which the filters are executed.
    echo "Setting up IPv6 flow rules for traffic"
    $TC filter add dev $IF protocol ipv6 parent 1:0 prio 1 u32 match u16 0x0c00
0x0ff0 at 0 flowid 1:11
    $TC filter add dev $IF protocol ipv6 parent 1:0 prio 2 u32 match u16 0x0b80
0x0ff0 at 0 flowid 1:12
    $TC filter add dev $IF protocol ipv6 parent 1:0 prio 3 u32 match u16 0x0480
0x0ff0 at 0 flowid 1:13
    $TC filter add dev $IF protocol ipv6 parent 1:0 prio 3 u32 match u16 0x0500
0x0ff0 at 0 flowid 1:13
    $TC filter add dev $IF protocol ipv6 parent 1:0 prio 3 u32 match u16 0x0280
0x0ff0 at 0 flowid 1:14
    $TC filter add dev $IF protocol ipv6 parent 1:0 prio 6 u32 match ip6 dst
::/0 flowid 1:15

    # Switch the default qdisc for classid 1:1x with pfifo (packet fifo) where
the queuelenght in packets are given with limit p.
    $TC qdisc add dev $IF parent 1:11 handle 11: pfifo limit 10
    $TC qdisc add dev $IF parent 1:12 handle 12: pfifo limit 5
    $TC qdisc add dev $IF parent 1:13 handle 13: pfifo limit 20
    $TC qdisc add dev $IF parent 1:14 handle 14: pfifo limit 50
    $TC qdisc add dev $IF parent 1:15 handle 15: pfifo limit 100

#    echo "Listing the queues:"
#    $TC -s qdisc ls dev $IF
#    $TC -s filter ls dev $IF
}

case "$action" in
    start) start ;;
    stop)  stop ;;
    restart|force-reload) stop && start ;;
    *)     log_failure_msg "usage: $progname [ start|stop|restart ] interface" ;
           false ;;
esac

exit $?
```