Coalition Networks for Secure Information Sharing

CoNSIS Task 4
Final Report

Version 1.0

28th November 2012

Record of Amendments

Amendment Number	Amendment Pages	Date Entered	Signature
1	All	26.07.2012	Barz
2	Experiment Evaluation	31.08.2012	Barz
3	Future Work	18.10.2012	Barz
4	Minor Changes	28.11.2012	Barz
5			

Table of Contents

0. Abstract

Secure information exchange is a key factor for the success of military operations. International coalition missions are especially challenging because of heterogeneous communication and C2IS equipment. The international project CoNSIS is targeted to fill in technical gaps regarding interoperability which occur in a reference scenario, consisting of a multinational convoy of military and non-governmental vehicles. The convoy forms an ad-hoc radio network and shares a common operational picture with an international headquarters, mainly via a satellite link. This paper addresses network management challenges and technical solutions for this federated scenario. Both the core network interconnecting different national headquarters with an international headquarters and the ad-hoc radio network of the convoy are addressed in a single, seamless concept. In June 2012, field tests including the convoy were carried out in order to evaluate the different technical solutions.

1. Introduction

CoNSIS – Coalition Networks for Secure Information Sharing – is an international project with France, Norway, Germany and the US currently participating. Based on the work done in INSC – Interoperable Networks for Secure Communications – it aims to work towards Network Enabled Capability (NEC). Heterogeneous networks from different nations are to be connected and form a federated environment in which to securely share information. CoNSIS concentrates on wireless networks in the tactical domain, but also considers deployed high speed networks as well as communication in-between. On the higher network layers, it places emphasis on a service-oriented architecture as stipulated in the NNEC Feasibility Study [1].
Work in CoNSIS is performed in five distinct groups. Task 1 is concerned with communication services. Task 2 is responsible for the integration of the SOA frameworks of the different nations. Task 3 is concerned with security, and task 4 with network management. Task 5 is responsible for the overall architecture and a field test scenario (see below) which serves as golden thread for all technical developments. The project concludes its first phase with the field tests in June 2012 and the evaluation and documentation of the results. This document concentrates on the work done in the network management task.
The CoNSIS scenario as depicted in Figure 1 is set in a country torn by civil war. International coalition troops are deployed in the country to stabilize the situation, protect the population and initiate the peace process. Larger cities are controlled by coalition forces, but the situation outside the cities is still unstable. Convoys and outposts are constantly at risk of attack. The coalition troops have established an international headquarters (HQ) which has fixed network connections to several national headquarters. There are also naval forces from different nations patrolling the waters around the conflict area. The naval vessels form a wireless ad-hoc network and are connected to the other forces via satellite. There is also a backup HF radio connection.
In this situation, a natural disaster occurs in a part of the country not controlled by the coalition forces. The coalition decides to aid in disaster relief efforts by escorting the vehicles of a humanitarian Non-Governmental Organization (NGO) to the disaster area and secure it. The military vehicles are connected by different broadband military radio technologies operating mainly in the UHF frequency spectrum, forming another ad-hoc network. As with the naval vessels, communication with the headquarters is ensured via satellite technology installed on a few specifically equipped vehicles. The NGO vehicles are also connected to the military convoy by terrestrial radio. Shortly after setting out, the convoy is joined by a second group of military vehicles from another nation. This group uses radios not compatible with the convoy's, but a few vehicles in both groups have radios with compatible waveforms to bridge the communication between the two groups. Following a reorganization of the network in the wireless domain, they now form a comprehensive ad-hoc network.
Making its way to the disaster area, the radio communication within the newly combined convoy is suddenly disrupted by a radio jammer. Satellite communication remains unaffected. The jamming is recognized, reported to headquarters, and finally eliminated by an air strike.

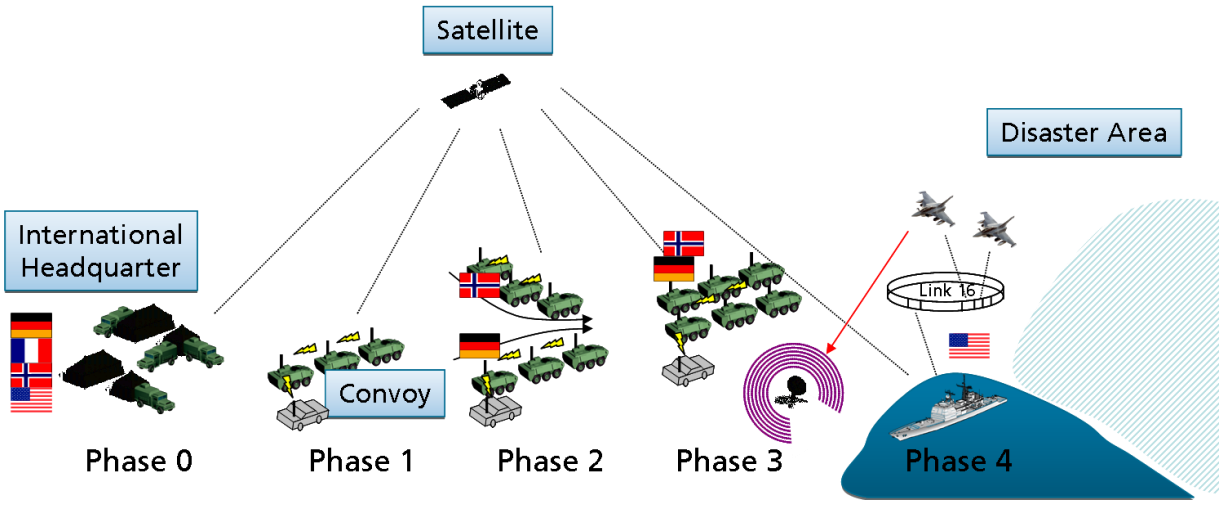

Figure 1: The CoNSIS network

2. Management Concept

2.1. Network Reference Model

This section gives a brief overview of the Network Reference Model described in "System and Experimentation Architectures - Version 1.0" [17]. Readers that are familiar with the model can safely skip this section.

The CoNSIS reference model consists of a core network to which user domains are connected via IPsec crypto devices. The core network itself is composed of a number of interworking networks operated by different administrative authorities. Figure 2 shows the main elements of the CoNSIS architecture.

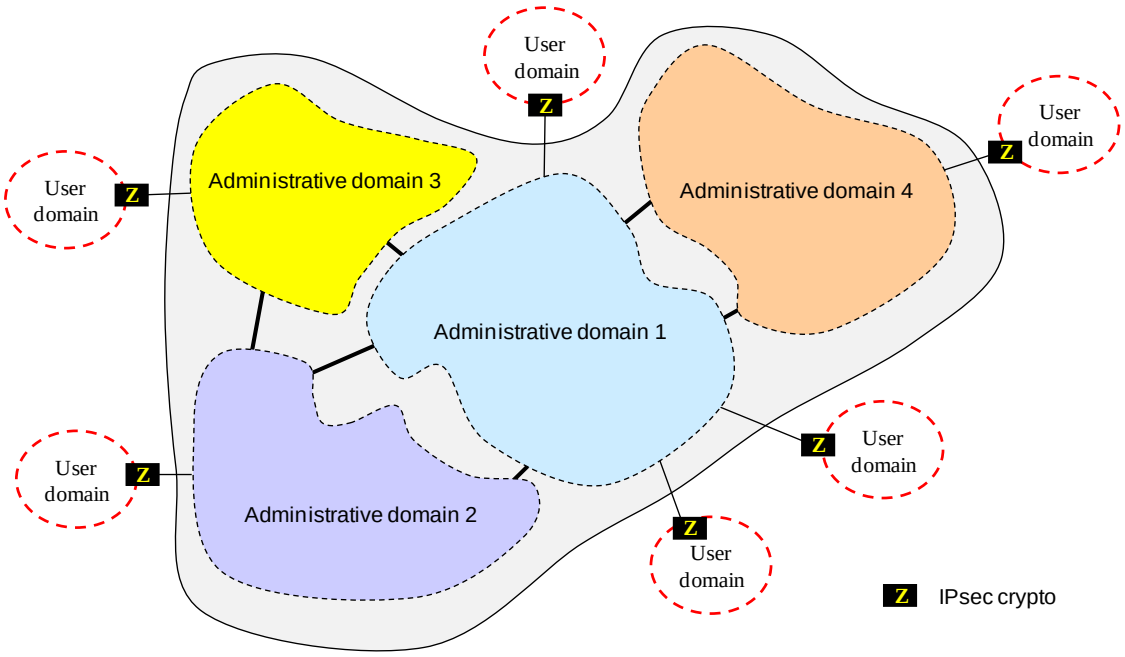

Figure 2: Administrative Domains

This architecture is close to the Protected Core Network (PCN) [2] approach.

In the PCN concept, secure red networks are represented by the Coloured Clouds (CCs), while the unprotected black network represents the Protected Core. PCN now requires the existence of certain distinguished nodes, the E-nodes, in the black network, which ensure availability and offer reliable transport to the CCs. These routers may be clustered to Protected Core Segments (PCSs) which together form the PCN. There are certain functionalities like traffic concealment that are associated with the E-nodes. In addition, the PCN concept defines interfaces between different PCS and between PCS and the Coloured Clouds.
The CoNSIS network architecture is based on this concept, but the two reference models are not identical. In particular, CoNSIS administrative domains are not assumed to have exactly the same functions as PCSs regarding e.g. security protection and the management of SLAs. The administrative domains interwork via interfaces which are not supposed to have the same features as the PCS-1 interface. Likewise, the generic interface between CoNSIS user domains and the core network is not necessarily compliant with the PCS-2 interface.
In order to reflect the above-mentioned divergence, objects of the CoNSIS reference model are given names intentionally different from their PCN counterparts (see Figure 3):

- The core network (counterpart of the PCN protected core) is referred to as the **Transport Network** (TN).
- The TN is a collection of interworking **Transport Network Segments** (TNS) (counterpart of PCSs), each TNS being defined as a set of network elements under a single administrative authority. A segment administered by a national authority is referred to as an N-TNS while a segment administered by the coalition is a C-TNS.
- User domains are referred to as **Coloured Enclaves** (CE) (counterparts of coloured clouds), separated from the TNS by IPSec. A CE can be embedded within another CE; in that case it is called an **Inner Coloured Enclave** (ICE).

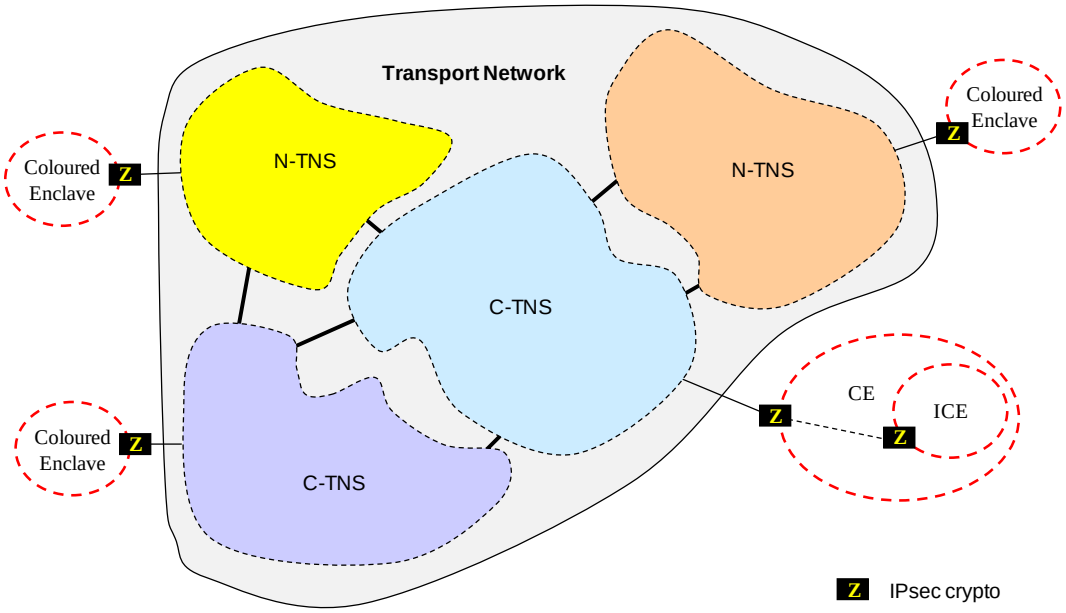

Figure 3: Network Segments and Colored Enclaves

2.2. Management Challenges related to the Network Reference Model

As mentioned in section 3 in the "System and Experimentation Architectures - Version 1.0" [17] the concept of a Transport Network consisting of Transport Network Segments which are managed under the administrative authority of different countries can be conceived as a multi-provider network. In general, the challenges of delivering end-to-end inter-provider QoS that were addressed in the network research community (e.g. [8]) also apply to the context of coalition networks. In addition to the standard information hiding requirements of network providers, special security considerations regarding the Coloured Enclaves have to be addressed when sharing monitoring data and managing the Transport Network Segments for military use. The general challenges that were identified in [8] are:

- *Common service definitions* for all administrative domains
- *Common performance metrics* to support end-to-end SLAs

Common service definitions are already addressed by CoNSIS task 1 in [11] (DSCP/Application Requirements). Without this standardisation a meaningful end-to-end service is hard to obtain.
Common performance metrics must be used if performance information needs to be concatenated across the different providers. This does not only include the definition of the metrics themselves, but also the definition of common aggregation periods for samples and the use of

reference times. Concatenation of measurements of different network segments enables a scalable approach to the control of end-to-end SLAs. This can be achieved by sectioning the network into multiple measurement segments, allowing the reuse of these measurements for different end-to-end paths. Note that the segmentation of the Transport Network already induces measurement segments. A framework for the concatenation of performance metrics [13][14] was development by the IETF.
Multi-provider/multi-segment QoS paths result in the need for mechanisms to allocate budgets for different network impairments (e.g. delay, jitter, …) that are defined on an end-to-end basis along the path to the different network segments which are separate administrative domains. Here, approaches include a static, a dynamic and a hybrid allocation of the acceptable end-to-end impairments. In the static approach, the maximum number of Transport Network Segments in the path could be assumed. The impairments are then equally distributed between these segments. However, this approach is less efficient and may rule out possible inter-TNS paths. The dynamic negotiation approach is most efficient but requires signalling between the TNSs. In the hybrid approach, all impairments are shared equally only with segments on the path. Thus, it does not support situations in which only an unequal distribution of impairments would result in an acceptable SLA.
This leads to the discussion of provider/segment interconnection models for dynamic QoS negotiation. Here, a hierarchical third party model (e.g. realized by NATO in the form of NATO service classes) can be envisioned, as well as a cooperative model. To respect the autonomy of the different countries managing the Transport Network Segments as well as for resilience reasons, the distributed cooperative negotiation model in combination with a centralized definition of common service classes and performance metrics seems to be the most appropriate solution. In addition, a distributed approach may be more resilient to outages. Here, knowledge of the E-Node topology might be beneficial for assessing the end-to-end connectivity and for finding an impairment allocation.
A similar challenge may arise within Transport TNSs if they are also organized as overlay networks. Links between E-Nodes may be realized by several lower layer links by one or more independent providers. If these providers do not offer common NATO service classes the next better national service classes would have to be chosen.

2.3. Management Interfaces

As described in section 2 in the document “Management Organization in a C-TNS” [20] the management and monitoring architecture is defined for coalition networks on the basis of TNSs and Technical Management Areas (TMAs) within the TNSs. As depicted in the following sections, the concept comprises three different interfaces related to monitoring (see Figure 4). Other management interfaces regarding configuration management are still to be defined in detail. However, Figure 5 and the description MI 4 and MI 5 provide first suggestions regarding configuration architecture.
MI 1 [18]: The network monitoring interface MI 1 specifies the communication between measurement points and measurement archives and is national concern. It might be either based on standard network management protocols like SNMP, a proprietary solution or based on a standardized Web service interface. The latter case should be preferred. For existing tools a wrapper to encapsulate implementation specific communication has to be implemented.
MI 2 [18] is used to transport monitoring information from measurement archives in the TMAs to the corresponding measurement archives in the national CEs. A transformation service can be used to transform raw measurement data into a format that can be shared between the different CEs. Task 3 provides a discussion about a transfer channel for the national moni-

toring data and its security implications [25]. Ways to accomplish this without compromising the confidentiality of red data are discussed in [6].

MI 3 [18] specifies the communication for distributing measurement and monitoring information between the CEs. A lookup service is responsible for advertising available measurements and to make the results available to search queries. The service will be based on SOA. Task 2 is responsible to provide the monitoring UI.

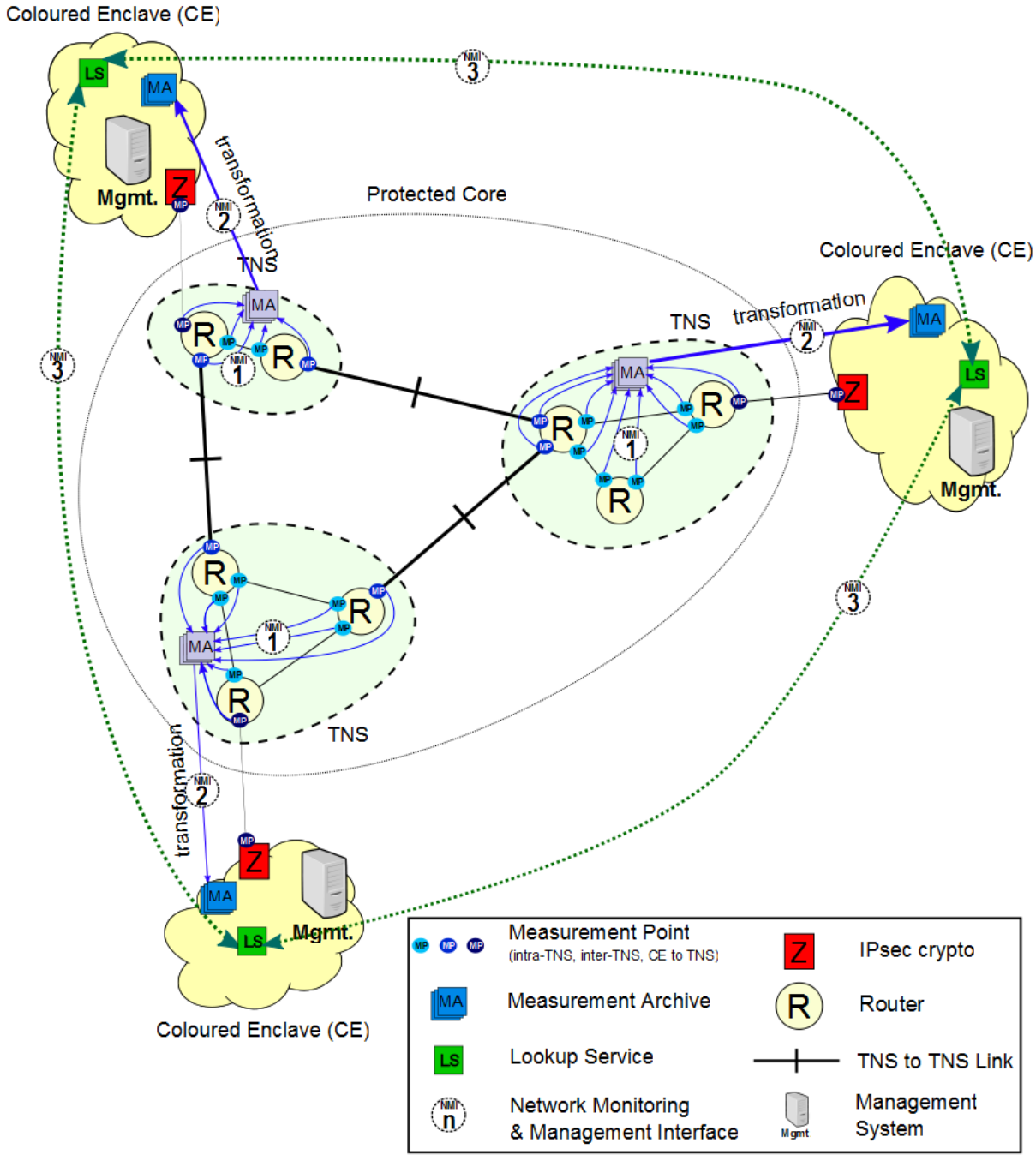

Figure 4: Refined Performance Monitoring Architecture

MI 4 [19] specifies an SLA negotiation/agreement interface between an Overall Coalition Manager and the different TNS Managers. This multi-domain QoS negotiation mechanism will work via bilateral communication between the Overall Coalition Manager and the Local

TNS Managers. The communication resources are under the authority of the local TNS Managers which act as a "management decision point". Reference [9] presents a similar approach.

MI 5 [20] specifies a configuration interface between the local TNS Managers and the Technical Management Areas under their administration. It is assumed that each TMA has special configuration management tools that might be proprietary. The TNS Manager will act as "management decision point". MI 5 should comprise high level technology agnostic configuration commands that need to be translated into a technology specific configuration by the appropriate configuration management tools.

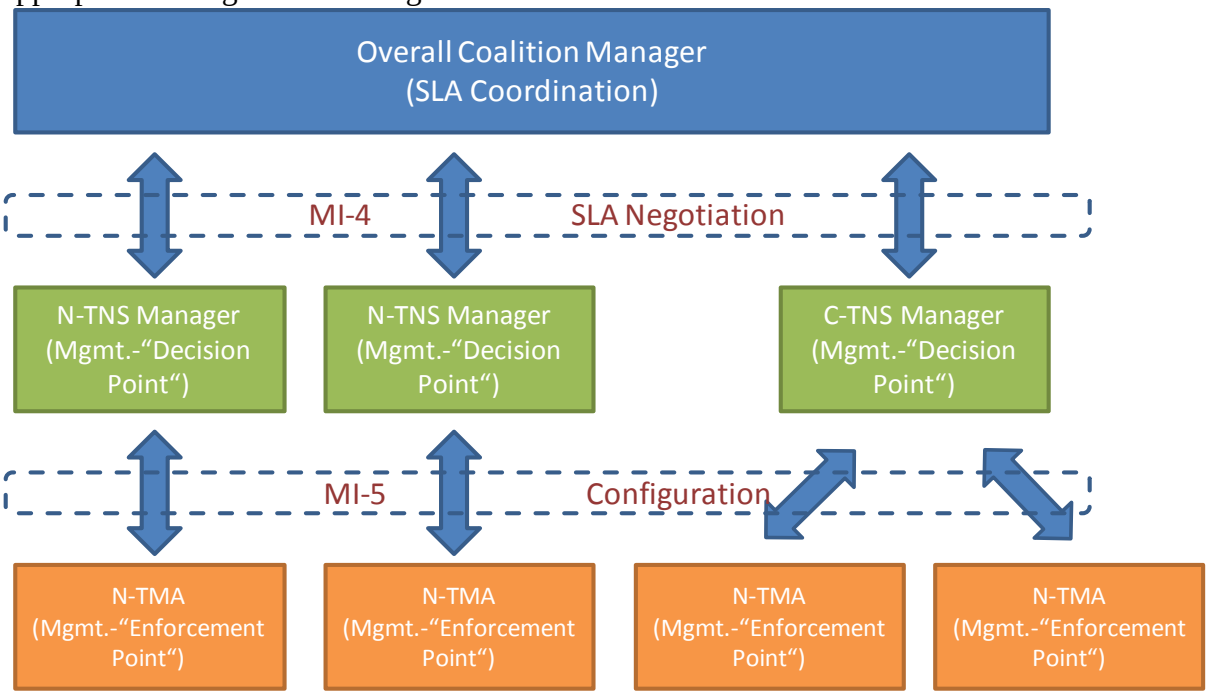

Figure 5: SLA Negotiation and Configuration Architecture

2.4. CoNSIS Task 4 Documents

The CoNSIS Management Concept is detailed in different documents:

- "System and Experimentation Architectures - Version 1.0" [17] describes the high level system architecture used in CoNSIS. From the network management perspective, it specifies a common network reference model which is also used for network management in task 4. In addition, it specifies the high level network management interfaces.
- "Management Concept Adaption" [18] analyses the network reference model. On this basis, an architecture for network performance monitoring and information sharing on a federated basis is elaborated and functional components are identified. In addition, measurement challenges related to the network reference model are identified and existing software solutions and standards are discussed for the realization of the functional components.
- "Coalition Management Philosophy" [19] and "Management Organisation in a C-TNS" [20] give an outline of a configuration management architecture.
- "IP Network Metrology – Architectures and Tools Applicable in a Coalition Network" [18] discusses the goals of measurements in a CoNSIS-type architecture, what is specific of measurements in a tactical environment, and it proposes generic solutions and operating principles.

- “Fair Queuing and active Measurement Methods” [21] discusses ways to perform measurements with active test flows despite the bias which may be introduced by routers implementing fair queuing techniques..
- “Multilevel Security and network management in CoNSIS” [25] provides a general overview of Multilevel Security and discusses approaches to exchanging data between CEs and TNSs and their security implications.
- “Jamming Indicators in Wireless Networks” [22] describes a cooperative approach using a cross-layer framework to detect jamming incidents without specialized hardware. As proof of concept, off-the-shelf Wi-Fi components were used.

3. CoNSIS Experimentation

Experimentations were conducted on an international testbed which consisted of both fixed and mobile networks. The fixed part of the testbed was made of routers in laboratories in France, Germany and the USA which constituted 6 different IP autonomous systems interconnected together via the Internet of national links

3.1. Field Test Setup

Experimentation in CoNSIS has a strong focus on the mobile part of the network, i.e. the convoy. It consists of three parts: NGO vehicles, Norwegian military vehicles (the original convoy), and German military vehicles (which join the convoy in phase 2). The German vehicles use three different types of radio, HiMoNN (IABG), FlexNet-4 (Rockwell Collins) and Harris radios. The Norwegian part uses Kongsberg WM600 radios and the NGOs commercial WLAN. None of these radio types are interoperable, which is why one Kongsberg radio is passed to the German convoy and one FlexNet-4 to the Norwegian one. In addition, at least one German and one Norwegian vehicle have a satellite connection. All UHF military radios in our scenario perform ad-hoc routing within their technology domain, which normally cannot be deactivated and provides no information about the internal topology. In addition, multi topology routing is not supported so far. Thus, these incompatible technologies need to be tied together in an overlay network with multi topology routing [10] support to cope with the heterogeneity of the different technologies. To overcome these limitations, a liaison with COALWND, the interoperable coalition wideband networking waveform for military radios under development, is planned to eliminate the need for a second layer of routing.
Jammer detection is usually done by dedicated, strategically placed units. In CoNSIS, there is an experimental option of the jammed systems doing the detection themselves. To detect a jamming incident locally, information from different network layers must be correlated, which requires a cross-layer information architecture. Besides reporting the incident to the international headquarters, local measures may be taken to circumvent the jamming, such as changing frequency or modulation or reconfiguring the routing.

3.2. Experiment Analysis

3.2.1. Fixed Network Experiments

MA-Basic: Maintain a common network picture

Purpose: Show how monitoring information can be shared between the HQs of the different nations regarding the network state within the different non-mobile ASs and on the inter-ASs links/tunnels. The experiment helps to identify problems within the TNSs and the inter-TNS links.

Test setup: PerfSONAR measurement archives collecting SNMP information from core TNS routers within each AS are installed as depicted in Figure 6.

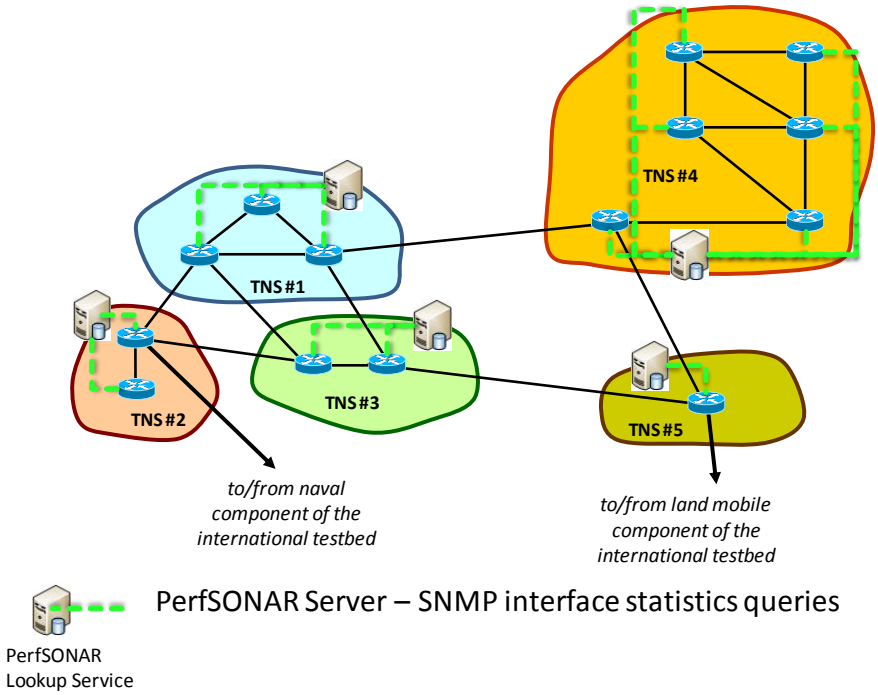

Figure 6: PerfSONAR SNMP Interface Statistic Queries

In addition, measurement archives collecting Iperf and OWAMP measurements from the inter-TNS links are installed. The information will be archived so the experiment also supports an offline analysis also regarding other experiments.

Walkthrough: Deployment and activation of the service is performed prior to the experiment. All nations can access the information via a Web based client during the whole experiment. All information is stored in local measurement archives. If necessary, measurement archives are cleared before the experimentation so there is enough storage capacity. In regular intervals the archives were backed up.

Prerequisites and special requirements: For autonomy reasons, PerfSONAR Measurement Archives (and a Lookup Service) were installed in every AS participating in the measurements (see Figure 7). Participating nations set up one or more virtual machines. In addition, an NTP server was needed to synchronize the measurements.

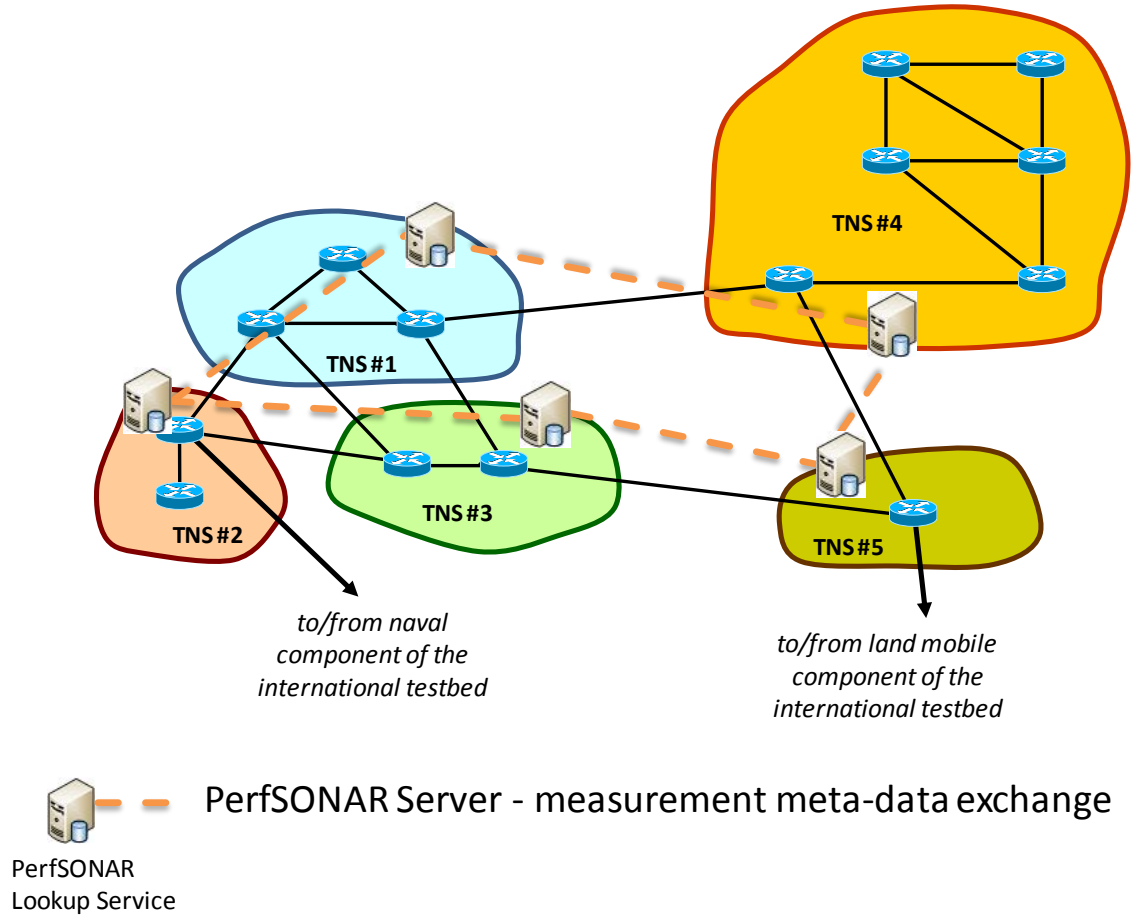

Figure 7: PerfSONAR Measurement Metadata Exchange

Experiment Analysis:

Before the experiments in Greding began PerfSONAR Network Component had been successfully installed in all locations except in San Diego. Because of the high security requirements, the installation of the components made this difficult. In each Location a working Measurement Archive and a Global Lookupservice exists. See Table 1: PerfSONAR Network Structure.

Location	IP-Adress	AS	TNS	VM-Name	VM-Typ	available
Fraunhofer-FKIE	10.24.1.150	AS64854	-	FKIE-GLS	GLS	+
Fraunhofer-FKIE	10.24.1.160	AS64854	-	FKIE-HLS	MA	+
IABG-Ottobrunn	10.21.100.21	AS64851	TNS4	DEU10-Mgmt-MA	MA	+
IABG-Ottobrunn	10.21.100.22	AS64851	TNS4	DEU10-Mgmt-GLS	GLS	+
WTD81-Greding	10.2.200.110	AS64851	TNS5	N-HQ-GLS	GLS	+
WTD81-Greding	10.2.200.120	AS64851	TNS5	N-HQ-MA	MA	+
Spawar-San Diego	10.96.176.5	AS65051	TNS7	US-HQ-GLS	GLS	-
Spawar-San Diego	10.96.176.6	AS65052	TNS7	US-HQ-HLS	MA	-

Table 1: PerfSONAR Network Structure

At the beginning of the experiments in Greding each location sees all MAs. The list of the MAs is split by service type. Each MA has a running SNMP service and an OWAMP service. (See Figure 8: List of all MAs created by any gLS). The screenshot was taken at the beginning of the experiment. Therefore we are missing the MAs from TNS7.

SNMP Services

Service Name	*Service Type*	*Address*	*Description*	*View*
SNMP-MA by N-HQ	MA	http://10.2.200.120:9990 /perfSONAR_PS/services/SNMPMA	Network-Traffic to N-HQ	Query
SNMP-MA at DEU10-MNGT-MA-IABG (IABG/Ottobrunn)	MA	http://10.21.100.21:9990 /perfSONAR_PS/services/SNMPMA	Network-Traffic at TNS4 (IABG/Ottobrunn)	Query
SNMP-MA at FKIE-AS-64854	MA	http://10.24.1.160:9990 /perfSONAR_PS/services/SNMPMA	Network-Traffic at FKIE-AS-64854	Query

PSB_OWAMP Services

Service Name	*Service Type*	*Address*	*Description*	*View*
perfSONARBUOY OWAMP N-HQ-MA	MA	http://10.2.200.120:8085 /perfSONAR_PS/services/pSB	One Way Measurments from N-HQ-MA	Query
perfSONARBUOY OWAMP at DEU10-Mgmt-MA IABG	MA	http://10.21.100.21:8085 /perfSONAR_PS/services/pSB	One Way Measurments from TNS4(IABG)	Query
perfSONARBUOY OWAMP at the FKIE MA	MA	http://10.24.1.160:8085 /perfSONAR_PS/services/pSB	One Way Measurments from MA at FKIE-AS-64854	Query

Figure 8: List of all MAs created by any gLS

At the end of the experiment the perfSONAR machines in San Diego were running but the connection was not optimal. This means packet loss and a long wait for the generation of the perfSONAR sites. Opening the web site containing the list of measurements from the US-perfSONAR machine needed an average time of about 30 to 50 seconds. (See Figure 9: Summary of packet dump for accessing the US web site)

Traffic	Captured	Displayed	Marked
Packets	19435	15311	0
Between first and last packet	49,328 sec	38,911 sec	
Avg. packets/sec	393,992	393,486	
Avg. packet size	565,385 bytes	691,175 bytes	
Bytes	10988256	10582576	
Avg. bytes/sec	222757,267	271967,309	
Avg. MBit/sec	1,782	2,176	

Figure 9: Summary of packet dump for accessing the US web site

To see how much time perfSONAR needs to build the OWAMP graphs for the different timelines the packets are captured with Tcpdump. Figure 10 shows the results for different timelines.

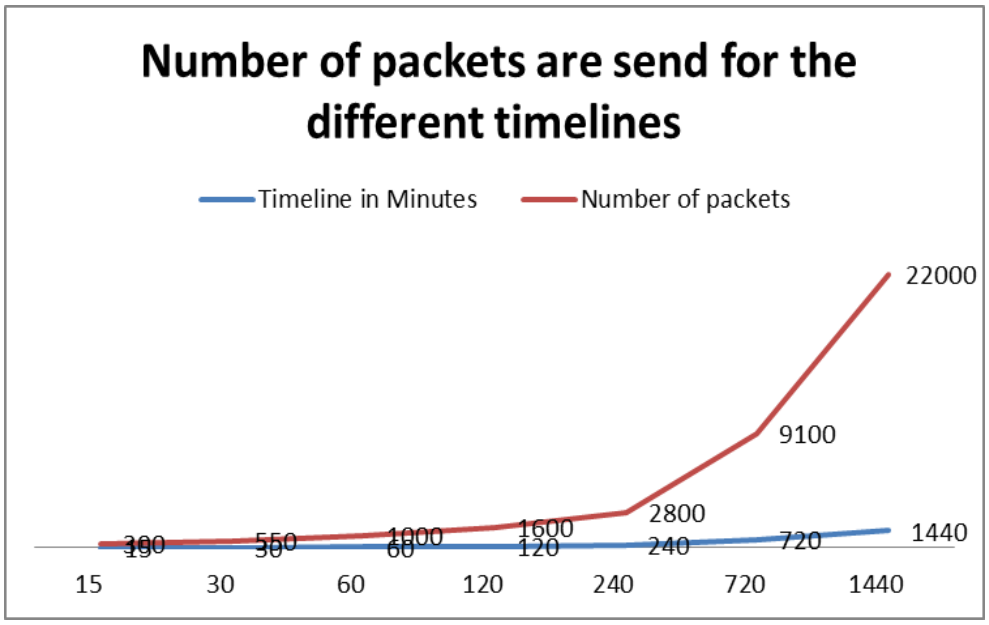

Figure 10: Number of IP packets send for different measurement timelines

When the timeline is less than 4 hours (240 minutes) the time for querying and displaying the information is well acceptable. For timelines greater than 4 hours, the time to query the information and building the graph takes too long. See Table 2: Required time and packages to create the OWAMP graph for different timelines. Another incident can occur when a certain amount of packet loss is exceeded. In that case the default browser gets a timeout after 320 seconds).

Timeline in Minutes	Number of packets	kbytes	Time in sec until statistic was displayed
15	300	170	4
30	550	335	8
60	1000	660	13
120	1600	1100	19
240	2800	1950	29
720	9100	6600	104
1440	22000	16200	Canceled by browser at 320

Table 2: Required time and packages to create the OWAMP graph for different timelines

During the test with a 24 h period the browser experiences the aforementioned timeout after 320 sec problem.

The average delay of a one way ping to the U.S. HQ is between 70 ms and 90 ms per packet. From US-HQ back to FKIE a delay of 74 ms occurs.

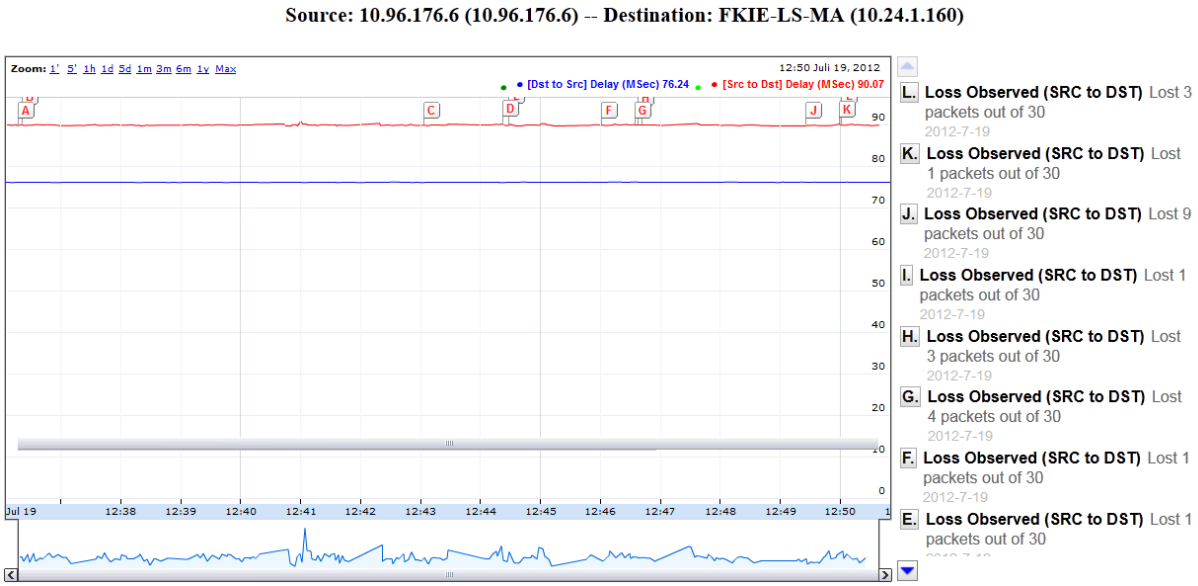

Figure 11: OWAMP measurements between the US HQ and a German location

For comparison OWAMP was compared with ping.

With a ping, we would have received the sum of both directions. This is actually the case. See the test case below:

1. The time from OWAMP of the direction from US to Germany.

   --- owping statistics from [US-HQ-hLS]:52322 to [10.24.1.160]:42346 ---

   100 sent, 0 lost (0.000%), 0 duplicates

   *one-way delay min/median/max = 85/**85.8**/735 ms, (err=0.185 ms)*

   one-way jitter = 5.3 ms (P95-P50)

   Hops = 2 (consistently)

2. The time of the direction from Germany back to US.

   --- owping statistics from [10.24.1.160]:60656 to [US-HQ-hLS]:39366 ---

   100 sent, 0 lost (0.000%), 0 duplicates

   *one-way delay min/median/max = 82.8/**83.1**/93.1 ms, (err=0.185 ms)*

   one-way jitter = 0.9 ms (P95-P50)

   Hops = 2 (consistently)

3. The round trip time (RTT) from a normal ping. The RTT is comparable to the sum of delay times of both directions from the OWAMP measurements.

   ---[root@US-HQ-hLS ~]# ping FKIE-LS-MA

   PING FKIE-LS-MA (10.24.1.160) 56(84) bytes of data.

   64 bytes from FKIE-LS-MA (10.24.1.160): icmp_seq=1 ttl=62 time=**169** ms

   64 bytes from FKIE-LS-MA (10.24.1.160): icmp_seq=2 ttl=62 time=**168** ms

Another important question is: How much space do the measurements of OWAMP consume. The average amount of data is 100MB per day for OWAMP measurements with 300 packets sent per minute.

All virtual machines with perfSONAR are monitored by Nagios (See Figure 12). Each machine has a plugin for ssh, ping, number of processes and the perfSONAR lookup service. If a process does not respond, it will be instantly highlighted in red.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
FKIE-LS-MA	Current Load	OK	2012-08-27 05:43:16	7d 0h 40m 28s	1/4	OK - load average: 0.00, 0.03, 0.01
	Current Users	OK	2012-08-27 05:44:26	96d 0h 37m 31s	1/4	USERS OK - 2 users currently logged in
	PING	OK	2012-08-27 05:44:00	0d 0h 37m 27s	1/4	PING OK - Packet loss = 0%, RTA = 1.95 ms
	SSH	OK	2012-08-27 05:40:54	0d 2h 54m 40s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	2012-08-27 05:44:26	32d 0h 55m 35s	1/4	PROCS OK: 140 processes
	hLS-FKIE	OK	2012-08-27 05:44:53	0d 2h 55m 44s	1/4	perfSONAR service replied "The echo request has passed."
FKIE-gLS	Current Load	OK	2012-08-27 05:41:43	9d 4h 26m 0s	1/4	OK - load average: 0.00, 0.05, 0.02
	Current Users	OK	2012-08-27 05:42:38	95d 23h 58m 52s	1/4	USERS OK - 2 users currently logged in
	PING	OK	2012-08-27 05:42:49	0d 0h 32m 42s	1/4	PING OK - Packet loss = 0%, RTA = 0.43 ms
	SSH	OK	2012-08-27 05:44:49	3d 1h 26m 11s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	2012-08-27 05:44:49	28d 9h 1m 35s	1/4	PROCS OK: 141 processes
	gLS	OK	2012-08-27 05:44:53	3d 1h 26m 11s	1/4	perfSONAR service replied "The echo request has passed."
IABG-MA	Current Load	OK	2012-08-27 05:44:43	9d 4h 25m 46s	1/4	OK - load average: 0.02, 0.03, 0.01
	Current Users	OK	2012-08-27 05:44:26	87d 22h 46m 6s	1/4	USERS OK - 2 users currently logged in
	IABG-hLS	CRITICAL	2012-08-27 05:44:53	23d 21h 40m 4s	1/4	CHECK_NRPE: Socket timeout after 10 seconds.
	SSH	CRITICAL	2012-08-27 05:44:53	23d 21h 37m 45s	1/4	CRITICAL - Socket timeout after 10 seconds
	Total Processes	OK	2012-08-27 05:44:49	32d 0h 55m 35s	1/4	PROCS OK: 141 processes
IABG-gLS	Current Load	OK	2012-08-27 05:41:03	9d 4h 25m 50s	1/4	OK - load average: 0.01, 0.05, 0.03
	Current Users	OK	2012-08-27 05:44:26	87d 22h 44m 38s	1/4	USERS OK - 2 users currently logged in
	IABG-gLS	CRITICAL	2012-08-27 05:44:53	23d 21h 37m 45s	1/4	CHECK_NRPE: Socket timeout after 10 seconds.
	SSH	CRITICAL	2012-08-27 05:44:53	23d 21h 37m 45s	1/4	CRITICAL - Socket timeout after 10 seconds
	Total Processes	OK	2012-08-27 05:44:48	45d 3h 57m 28s	1/4	PROCS OK: 137 processes
US-HQ-hLS	Current Load	OK	2012-08-27 05:44:53	0d 0h 20m 38s	1/4	OK - load average: 0.02, 0.03, 0.01
	Current Users	OK	2012-08-27 05:40:46	0d 0h 19m 45s	1/4	USERS OK - 2 users currently logged in
	SSH	OK	2012-08-27 05:41:39	0d 0h 18m 52s	1/4	SSH OK - OpenSSH_4.3 (protocol 2.0)
	Total Processes	OK	2012-08-27 05:42:32	0d 0h 17m 59s	1/4	PROCS OK: 137 processes
	hLS	OK	2012-08-27 05:43:25	0d 0h 17m 6s	1/4	perfSONAR service replied "The echo request has passed."
localhost	Current Load	OK	2012-08-27 05:42:47	9d 4h 25m 59s	1/4	OK - load average: 0.00, 0.03, 0.02
	Current Users	OK	2012-08-27 05:40:35	96d 2h 38m 11s	1/4	USERS OK - 2 users currently logged in
	SSH	OK	2012-08-27 05:45:21	24d 7h 30m 25s	1/4	SSH OK - OpenSSH_5.3p1 Debian-3ubuntu7 (protocol 2.0)

Figure 12: Nagios monitoring of the measurement machines and services

Findings: Sharing monitoring data between the different TNSs worked well in general. The performance of intra- and inter-TNS links in the non-mobile part of the network was accessible.

MA-SOA: Provide access to PerfSONAR data via the SOA architecture

Purpose: Access to OWAMP data stored by PerfSONAR in an MA via a standard Web service as utilized in the CoNSIS SOA architecture. One application is the provision of data for generating technical profiles (see experiment "Technical Profiles" later in this section).

Test setup: A SOA service acts as consumer to the PerfSONAR OWAMP measurement archives of the different nations. In turn, it offers access to the latest packet loss rate in a queried TNS for a queried class of service.

One particular client application that uses the data is the technical profile process in one or several TNSs. The communication is based on SOAP messages (standard Web service).

Walkthrough: The client process queries the information needed to generate a technical profile from the SOA service. This will be packet loss rates on various links and tunnels. The SOA service determines the archives holding the relevant information and retrieves the data. The retrieved data is processed and the packet loss rate is forwarded to the client.

Prerequisites and special requirements: The SOA service needs to be available in the TNS part of the network (not in the CEs). It does not use WS-Discovery but is statically configured.

Experiment Analysis:

The experiment was split into two parts. The wrapper service itself can be divided into three parts: The consumer part, where the wrapper calls the PerfSONAR MA using the PerfSONAR

interface, the provider part which offers its own interface to third-party consumers, and the internal part which does the actual processing and bridges those two interfaces. Part 1 of the experiment thus involved the technical profile client calling the provider interface, with the provided data being generated on the fly. This experiment was performed in Greding. Part 2 involved the consumer part of the future wrapper querying the PerfSONAR OWAMP MA. This experiment was later performed at FKIE.

Part 1 of the experiment was set up as follows: The technical profile process was located in the French part of the CoNSIS wired testbed (see also experiment ”Technical Profiles: Use measurement results to automatically update the description of network capabilities”). It incorporated a Java client for the SOA wrapper. The test application for this experiment was a website which, depending on the current measured loss rates, displayed a picture in high or low resolution (see Figure 13). Accordingly, the SOA provider alternated the faked loss rates between 0% and around 90%, changing every ten minutes.

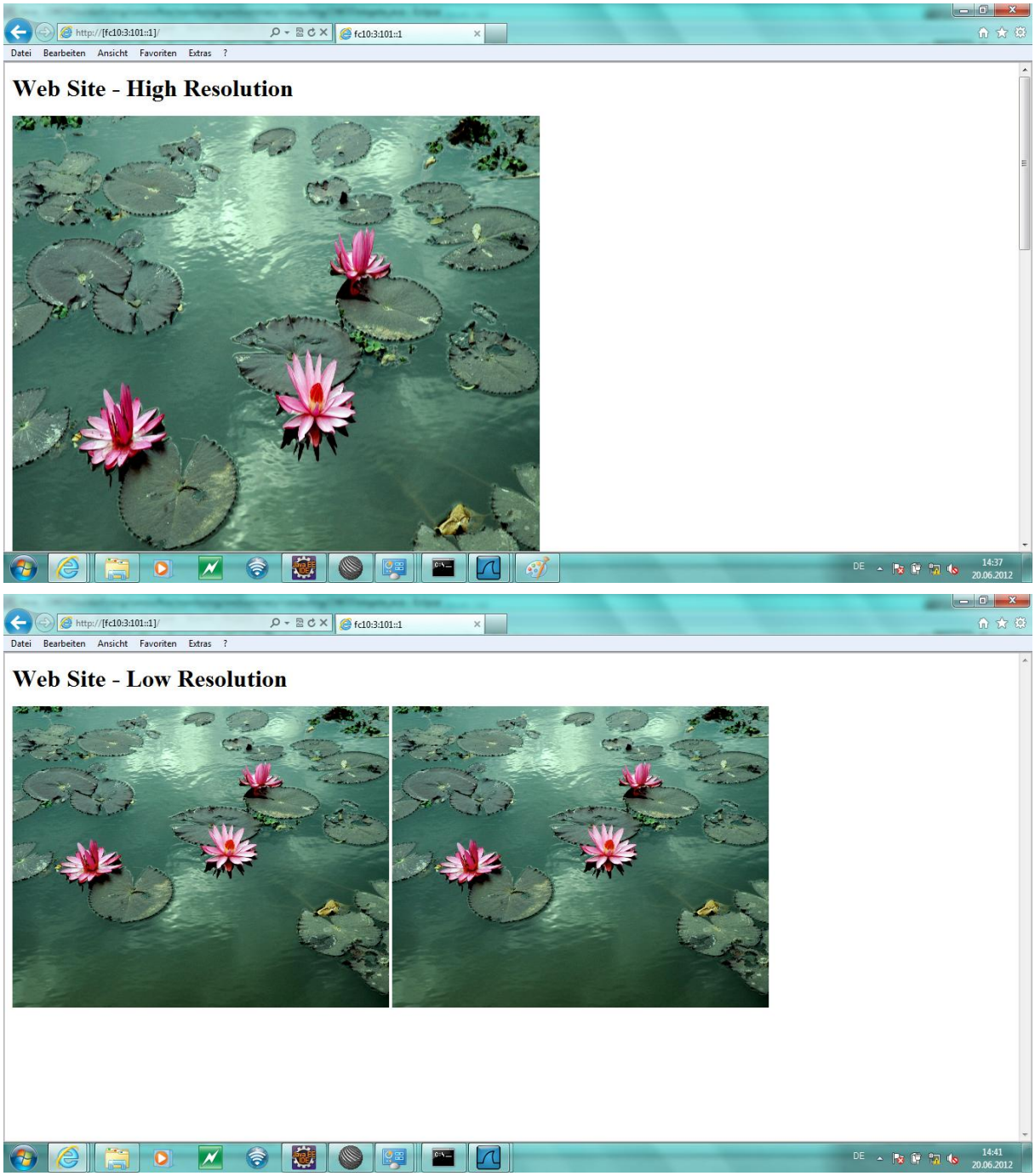

Figure 13: High (top) and low (bottom) resolution version of the website automatically selected.

Request and response looked like this:

```
   POST /soap/owpl/ HTTP/1.1
Content-Type: text/xml; charset=UTF-8
Accept: */*
SOAPAction: "requestOWPacketLoss"
User-Agent: Apache CXF 2.5.3
Cache-Control: no-cache
Pragma: no-cache
Host: [fc10:f220:1::cdcd:103]:8888
Connection: keep-alive
Content-Length: 274

   <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
       <soap:Body>
        <OWPacketLossRequest xmlns="http://task2.consis.org/management/packetLoss/">
            <ASIdentifier>AS1</ASIdentifier>
            <classOfService>EF</classOfService>
        </OWPacketLossRequest>
       </soap:Body>
   </soap:Envelope>


   HTTP/1.1 200 OK
Content-Type: text/xml;charset=UTF-8
Content-Length: 317
Server: Jetty(7.5.4.v20111024)

   <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
       <soap:Body>
        <OWPacketLossResponse xmlns="http://task2.consis.org/management/packetLoss/">
            <ASIdentifier>AS1</ASIdentifier>
            <classOfService>EF</classOfService>
            <packetLoss>90.1479438312709</packetLoss>
        </OWPacketLossResponse>
       </soap:Body>
   </soap:Envelope>
```

Upon change from 0% to 90% loss rate, the technical profile process allowed the web server to immediately choose the low resolution page.

Part 2 was more comprehensive. The PerfSONAR API is based on SOAP webservices, just like the CoNSIS SOA environment, but it does not use the WS-I stack of standards. The message format, standardised by the NMWG, is described using Relax-NG, which is not wholly compatible with XML Schema. As a result, the schema converted from Relax-NG to XML Schema allows more variations than the original one and still cannot be consumed by JAXB, the Java XML binding at the root of all service implementations in CoNSIS. Only after extensive manual editing, severely restricting valid messages compared to the Relax-NG schema, would it be possible to generate the Java code skeleton from the schema. The resulting interface is such that a PerfSONAR service should accept any message from the Java service, but the Java service will accept only about a third of the possible messages from a PerfSONAR service, necessitating the transformation of a message before it can be consumed. Additionally,

the NMWG schema is very general in the sense that up to a point, all compliant messages contain the same elements – but at that point, they branch out by substituting different definitions of one element, something which cannot be modelled by use of XML schema. The PerfSONAR documentation is moreover very lacking in describing which service uses which element definition. Only by request to the PerfSONAR mailing list was it possible to determine that one should always use the most general terms when querying a PerfSONAR MA, something which was apparent neither from the documentation nor the schema files present in the MA directory, thus violating the SOA principle that it should not be necessary to know more than a service's interface to successfully invoke it.

The test setup was as follows. The PerfSONAR MA was located in the FKIE part of the CoNSIS testbed. The SOA client was located in the FKIE company network. The connection between them was tunneled over an access router at the edge of both these networks. The client sent a request for the data of the last fifteen minutes to the OWAMP MA, which looked like this:

```
POST /perfSONAR_PS/services/pSB HTTP/1.1
Content-Type: text/xml; charset=UTF-8
Accept: */*
SOAPAction: "requestMeasurement"
User-Agent: Apache CXF 2.5.2
Cache-Control: no-cache
Pragma: no-cache
Host: 128.7.5.54:8085
Connection: keep-alive
Content-Length: 1748

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns4:message
      xmlns:ns4="http://ggf.org/ns/nmwg/base/2.0/"
      xmlns:ns8="http://ggf.org/ns/nmwg/ops/select/2.0/"
      xmlns:ns13="http://ggf.org/ns/nmwg/topology/2.0/"
      id="message.1347541715"
      type="SetupDataRequest"
    >
      <ns4:metadata id="metadata.1347541715">
          <ns4:subject id="subject">
            <ns13:endPointPair>
              <ns13:src value="10.24.1.160" type="ipv4"/>
              <ns13:dst value="10.96.176.6" type="ipv4"/>
            </ns13:endPointPair>
          </ns4:subject>
      </ns4:metadata>
      <ns4:metadata id="metadata.1347541715.chain">
        <ns8:subject metadataIdRef="metadata.1347541715" id="subject.1347541715"/>
        <ns8:parameters id="parameters.1347541715">
          <ns4:parameter name="startTime" value="1347455315"/>
          <ns4:parameter name="endTime" value="1347541715"/>
        </ns8:parameters>
        <ns4:eventType>http://ggf.org/ns/nmwg/ops/select/2.0</ns4:eventType>
      </ns4:metadata>
      <ns4:data metadataIdRef="metadata.1347541715.chain" id="data.1347541715"/>
    </ns4:message>
  </soap:Body>
</soap:Envelope>
```

Notice that there is no indication of which type of data is requested or what format it will be given in; this is determined solely by the MA. The response with the requested data accordingly gave the data in two formats, only one of which would have been desired. As indicated before, the SOA service was unable to process the response. An XSL transformation of the message was set up, but was prepared only for the desired data format and thus aborted with an error.

Findings:

A CoNSIS-compliant SOA wrapper for the PerfSONAR framework is perfectly possible, but requires a lot more work than would have been expected from a framework which uses the same underlying technologies. It is possible that some of the problems arose from the Perl im-

plementation of PerfSONAR, which is why it is recommended to test the alternative Java implementation.

Measurement Probes: Assess the usability and trustworthiness of various measurement probes

Purpose: Determine how and to what extent software probes can be used in a tactical network to provide measurement information.

Because they require in-depth investigation and comprehensive procedures, tests of this series were actually performed before the field experimentations described in this document, but on the same testbed and with the same technical means. They paved the way for the use of appropriate measurement tools during the field tests themselves.

Test setup: A broad array of measurement tools were tested, including Iperf, Internet2 OWAMP and Cisco IP SLAs which turned out to be the best ones.

Software probes have a high appeal in a tactical environment for the obvious reason that they do not imply additional hardware and thus have no detrimental effect on the compactness of deployed assets. Conversely, they have the downside of providing results with a lower precision, and of requiring active test flows (i.e. specific measurement packets) which may be a source of overhead.

The precision and trustworthiness of software probes was assessed whenever needed by comparing the results they supply with those provided by hardware measurement tools such as Smartbit or Ipanema whose performance in terms of accuracy is acknowledged.

Findings: The major lessons learnt through the tests concern the precautions a network operator should take when using software probes, the precision that can be expected in measurements, and the consistency of results supplied by probes of different types.

Overall, all three above-mentioned software tools proved to be usable and to provide valuable information as long as they are operated in an appropriate environment and within their normal range. It was indeed an important discovery that *each measurement probe has a range* (e.g. of data rates, of number of packets per second) within which it will work properly, but beyond which it may supply erroneous, incomplete or inconsistent data. The recommendation is thus that a network operator should only use measurement devices whose range of valid operation has been duly tested prior to deployment.

It was also shown that, with appropriate procedures, the overhead due to active measurement flows could be kept under control and remains marginal as compared to user traffic, even in a narrow-bandwidth network.

Finally, special care must be taken when conducting active measurements in IP systems which implement such mechanisms as weighted fair queuing (WFQ). As WFQ creates a dissymmetry in the treatment of flows even if they belong to the same class of service, it may result in active test flows experiencing a different quality of service than the user flows they are intended to represent, and thus lead to erroneous conclusions in network performance monitoring. This is but one more illustration of the well known principle that measurements cannot be performed in total ignorance of the system they apply to.

Another important finding is that a given probe will provide results which are consistent with themselves, but not always as consistent with those of other probes. For example, Internet2 OWAMP used in two different segments of a network will indicate jitter values which can be directly compared to each other, and so will Iperf, but the measurements supplied by the two tools may not be consistent with one another. This bias which may exist between two different probes is no serious hindrance per se since a theoretical study leads to the conclusion that high precision in the measurements conducted in an IP network is not needed and should not be sought. However, when comparisons are made between measured values within a net-

work, or when measurements are composed in space or in time, the same type of tool should be used throughout the system. PerfSONAR and its message format provide the means to distinguish measurements of different tools.

Technical Profiles: Use measurement results to automatically update the description of network capabilities

Purpose: A technical profile is a data set which describes the current capabilities of a TNS (e.g. the quality of service it is able to support, whether it is subject to sudden major alterations due to e.g. high mobility, jamming). This data set is intended to be communicated to users or adjacent networks so they will optimize the way they use the services of the relevant TNS.

Technical profiles were studied under CoNSIS task 1 (communication services), but an important aspect of their definition is that they should be kept up to date automatically so as to actually reflect the current transport conditions prevailing in a TNS. One essential way to update a technical profile is of course to use the results of measurements conducted according to the methods and procedures recommended by task 4.

Test setup: Host A is a web server, host B is a web client. They are located in two different colored enclaves connected to TNS 1 and TNS 4 respectively.

The technical profiles of these two TNSs are held by their respective Network Management Systems (NMSs). They are kept up to date thanks to measurements periodically performed by probes deployed throughout the two networks.

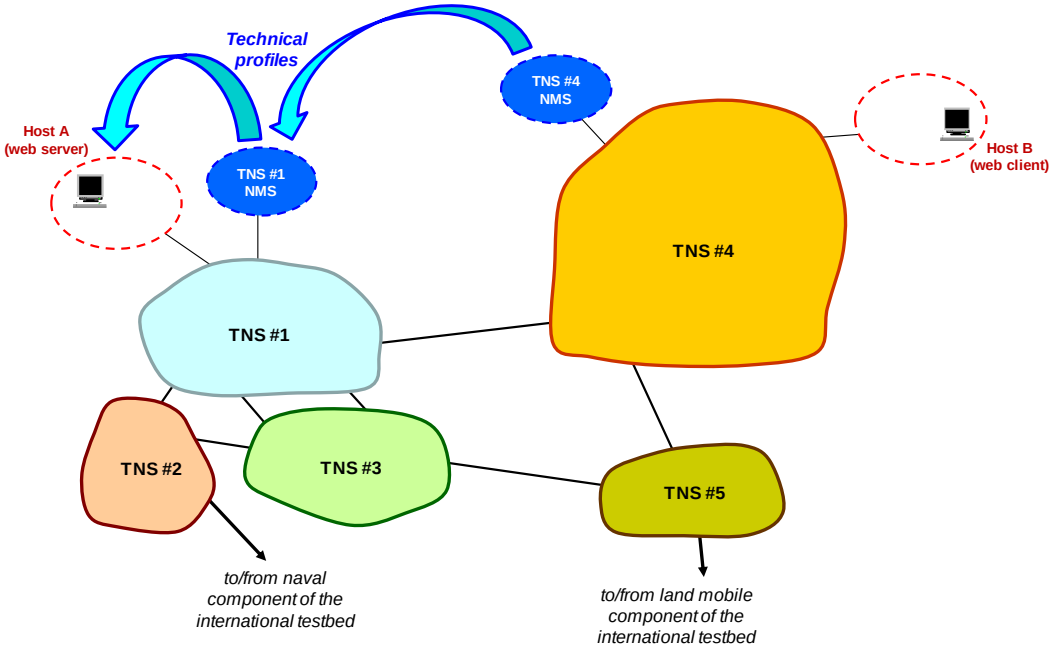

Figure 14: Technical profile repositories and user systems which will use these technical profiles

Walkthrough: When technical profile mechanisms are not enabled and when traffic conditions in TNS 4 are adverse, it takes an unacceptable time for host B to download a HTML page from host A.

When technical profile mechanisms are enabled, measurements permanently conducted in all TNSs allow the detection of a degradation of transport conditions (in this case a high packet loss rate in TNS 4), and this situation is reflected in the relevant technical profiles.

Whenever it receives a request from host B, host A first determines which TNSs will be traversed by the data flow it is about to send to the Web client. Then it fetches the technical profiles of TNSs 1 and 4 and composes them to find out that the path will be affected by a high packet loss rate.

Knowing this, it decides to send a HTML page with reduced contents (i.e. with lower-resolution pictures) to host B. The time it takes for host B to download the page returns to an acceptable value.

At the end of this test, the best compromise has been automatically discovered to ensure end-to-end quality of service in the presence of degraded transport conditions detected through measurements.

Prerequisites and special requirements: Interface MI 2, as described in section 2.3 of this article, is required to convey to the colored enclave information pertinent to the black networks.

3.2.2. Experiments regarding the convoy

SNMP-Mob: Extend the common operational picture to the convoy

Purpose: Collect SNMP-based information from the mobile domain.

Test setup: A perfSONAR SNMP measurement archive is installed on the border router of the mobile domain. Data about interface/tunnel statistics is requested from the mobile MTR routers that have a direct SAT connection to the border router of the mobile domain.

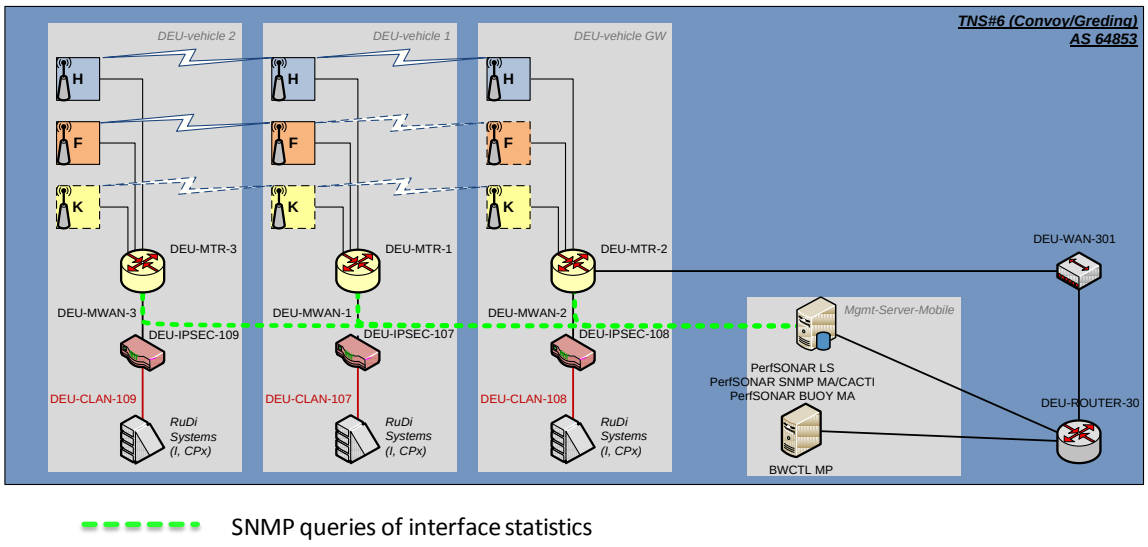

Figure 15: SNMP interface statistics queries in the wireless domain

Walkthrough: Periodic queries via SNMP from the measurement archive co-located with the border router of the mobile domain are performed. This data can be accessed via the perfSONAR framework.

Prerequisites and special requirements: SNMP data is fetched remotely via the satellite links. This has to be taken into account by experiments related to the convoy part of the network. The impact on the satellite links is supposed to be not relevant.

Experiment Analysis:

All MTRs are found and the list of interfaces of the MTRs can be shown via perfSONAR (See Figure 16)

http://ggf.org/ns/nmwg/characteristic/utilization/2.0 @ http://10.2.200.120:9990/perfSONAR_PS/services/SNMPMA									
Address	**Host Name**	**If. Index**	**If. Name**	**Description**	**If. Address**	**Capacity**	**Graph**	**Flash Graph**	**Open Flash Graph**
10.2.205.254		8	eth1.11	eth1.11	10.2.95.1	1 Gbps	-- Select --	-- Select --	-- Select --
10.2.205.254		7	eth1.12	eth1.12		1 Gbps	-- Select --	-- Select --	-- Select --
10.2.205.254		6	eth1.13	eth1.13	172.16.101.1	1 Gbps	-- Select --	-- Select --	-- Select --
10.2.205.254		5	eth1.14	eth1.14	172.32.205.2	1 Gbps	-- Select --	-- Select --	-- Select --
10.2.205.254		10	eth1.15	eth1.15	10.165.165.72	1 Gbps	-- Select --	-- Select --	-- Select --
10.2.205.254		9	eth1.90	eth1.90	10.23.1.205	1 Gbps	-- Select --	-- Select --	-- Select --
10.2.205.254		29	tun402	tun402			-- Select --	-- Select --	-- Select --
10.2.205.254		22	tun406	tun406			-- Select --	-- Select --	-- Select --
10.2.205.254		17	tun503	tun503			-- Select --	-- Select --	-- Select --
10.2.205.254		35	tun506	tun506			-- Select --	-- Select --	-- Select --
10.2.205.254		30	tun509	tun509			-- Select --	-- Select --	-- Select --
10.2.205.254		25	tun511	tun511			-- Select --	-- Select --	-- Select --
10.2.205.254		23	tun701	tun701			-- Select --	-- Select --	-- Select --
10.2.205.254		19	tun703	tun703			-- Select --	-- Select --	-- Select --
10.2.205.254		14	tun805	tun805			-- Select --	-- Select --	-- Select --
10.2.205.254		12	tun809	tun809			-- Select --	-- Select --	-- Select --
10.2.205.254		33	tun812	tun812			-- Select --	-- Select --	-- Select --
10.2.205.254		27	tun815	tun815			-- Select --	-- Select --	-- Select --
10.2.205.254		26	tun904	tun904	10.254.15.96		-- Select --	-- Select --	-- Select --
10.2.205.254		24	tun908	tun908	10.254.25.96		-- Select --	-- Select --	-- Select --
10.2.205.254		21	tun911	tun911	10.254.35.96		-- Select --	-- Select --	-- Select --
10.2.205.254		20	tun913	tun913	10.254.45.96		-- Select --	-- Select --	-- Select --
10.2.205.254		18	tun915	tun915			-- Select --	-- Select --	-- Select --

Figure 16: List of the interfaces of the captured MTRs by perfSONAR

The traffic increase by generating ICMP echo requests (ping6) packets with a size of 1200 bytes from MTR5 to MTR3 perceived successfully by the measurement point via cacti (see Figure 17).

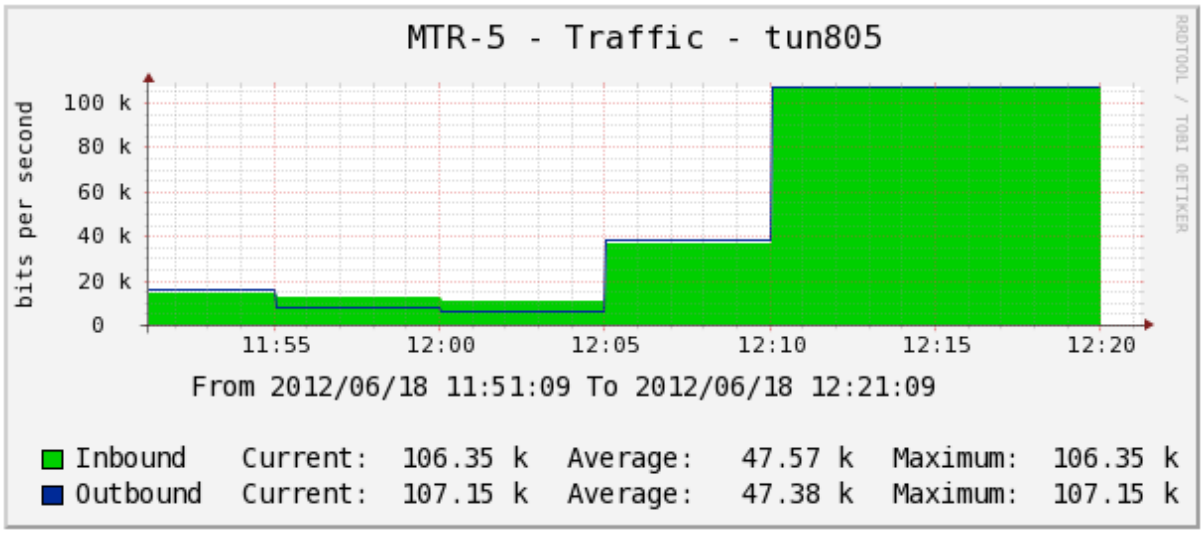

Figure 17: Inbound and outbound traffic from the Kongsberg tunnel from MTR5 to MTR3

The interval of measurement from cacti is set to 5 minutes. The rise of the incoming and outgoing packets increases from 15 kbytes to 100 kbytes by the created ICMP packets

The results from the traffic flow are successfully shared by perfSONAR. Figure 18 shows the recorded values from cacti above represented by the perfSONAR-UI.

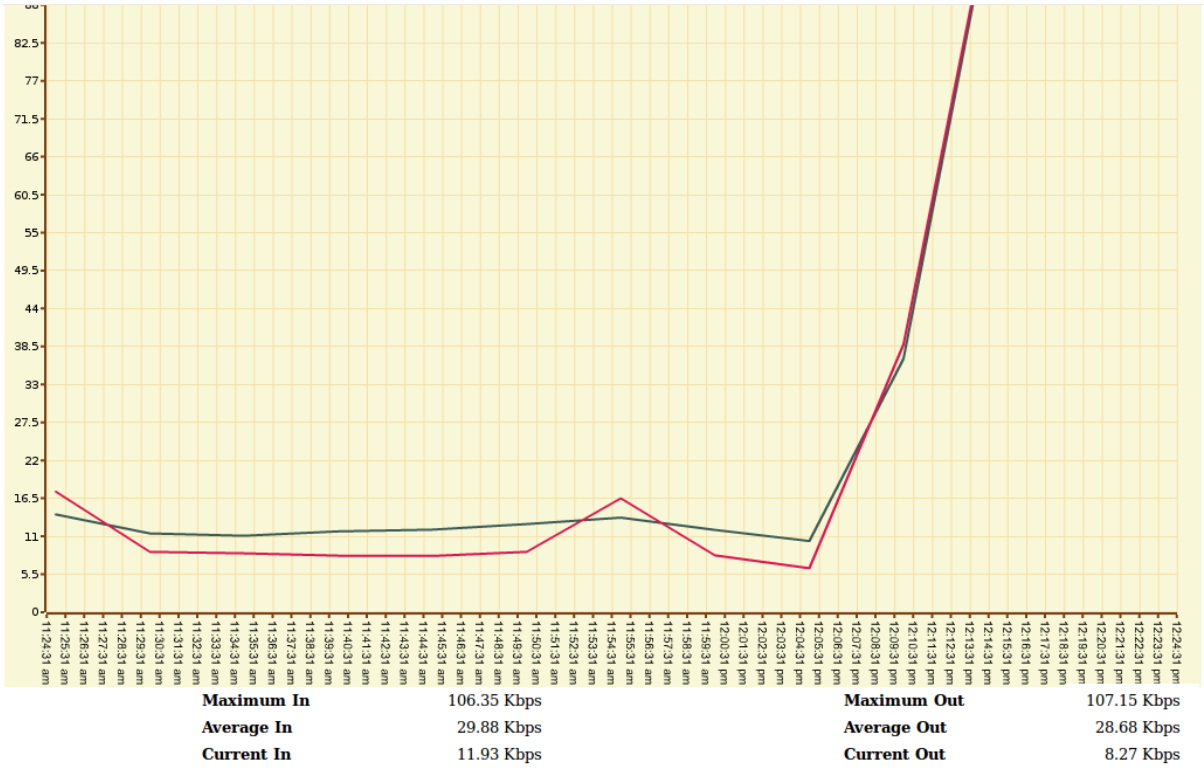

Figure 18: Inbound and outbound traffic from MTR5 to MTR3 represented by perfSONAR-UI

To see how much traffic is generated by measuring the interfaces by SNMP, the measurement queries for all MTRs are activated and the traffic flow from the interface at the MTR5 to the measurement archive is captured (see Figure 19).

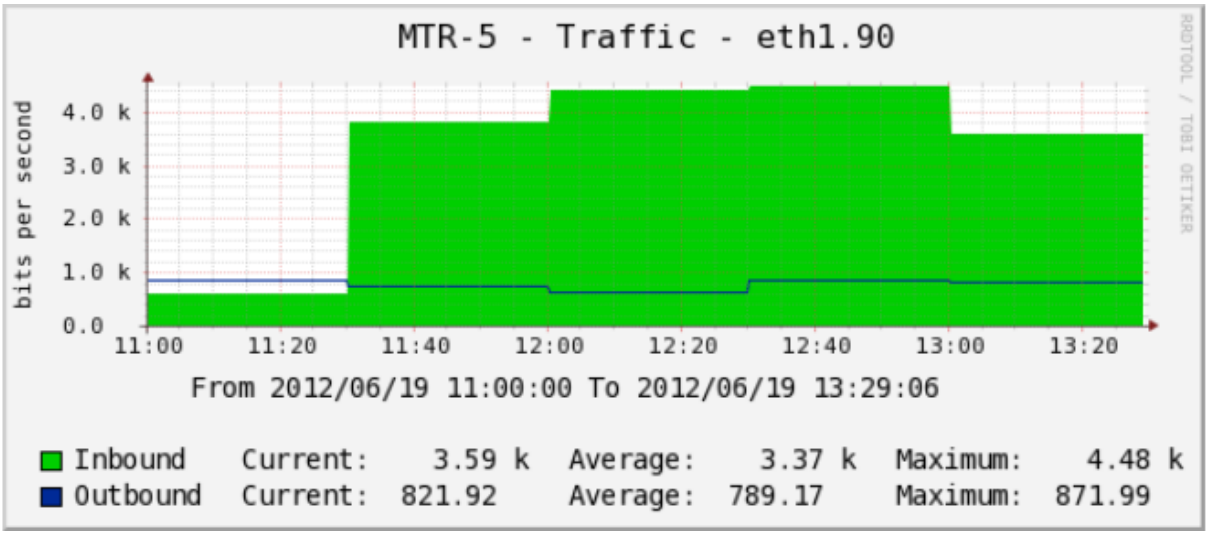

Figure 19:Inbound and outbound traffic from the MTR5 interface to the MTRs at the convoy

The network traffic from the SNMP measurements is not high. When all SNMP measurements are running the maximum traffic is 4,48kbits per second and the average is 3,37kbs.

To present all interfaces of the convoy is beyond the scope of this report. So here are the most interesting events.

The Diagrams at Figure 20 shows the interfaces from MTR1. They use the HiMoNN radios at the first vehicle of the convoy.

Figure 20:Inbound and outbound traffic on the HiMoNN tunnels from MTR1

- Tun901 is the HiMoNN connection between the DEU1-vehicle and DEU2-vehicle
- Tun902 is the HiMoNN connection between the DEU1-vehicle and DEU3-vehicle
- Tun903 is the HiMoNN connection between the DEU1-vehicle and DEU4-vehicle
- Tun904 is the HiMoNN connection between the DEU1-vehicle and the N-HQ

- Tun905 is the HiMoNN connection between the DEU1-vehicle and NOR2-vehicle

Disconnections between the vehicle and the other vehicles occurred between 15:35 to 15:45, 16:10 to 16:20, and 16:30 to 16:40. These disconnects were caused by the attempt to jam the HiMoNN connection of vehicle DEU1 or disconnections due to the geographic environment. To determine the exact reason you need additional indicators (like signal strength and noise).

Another event is when DEU1 is jammed crawler sends a note that vehicle DEU-1 was jammed. Due to an implementation error, the number of notification messages was so high that the total bandwidth of the radio is utilized (see the diagrams at Figure 21).

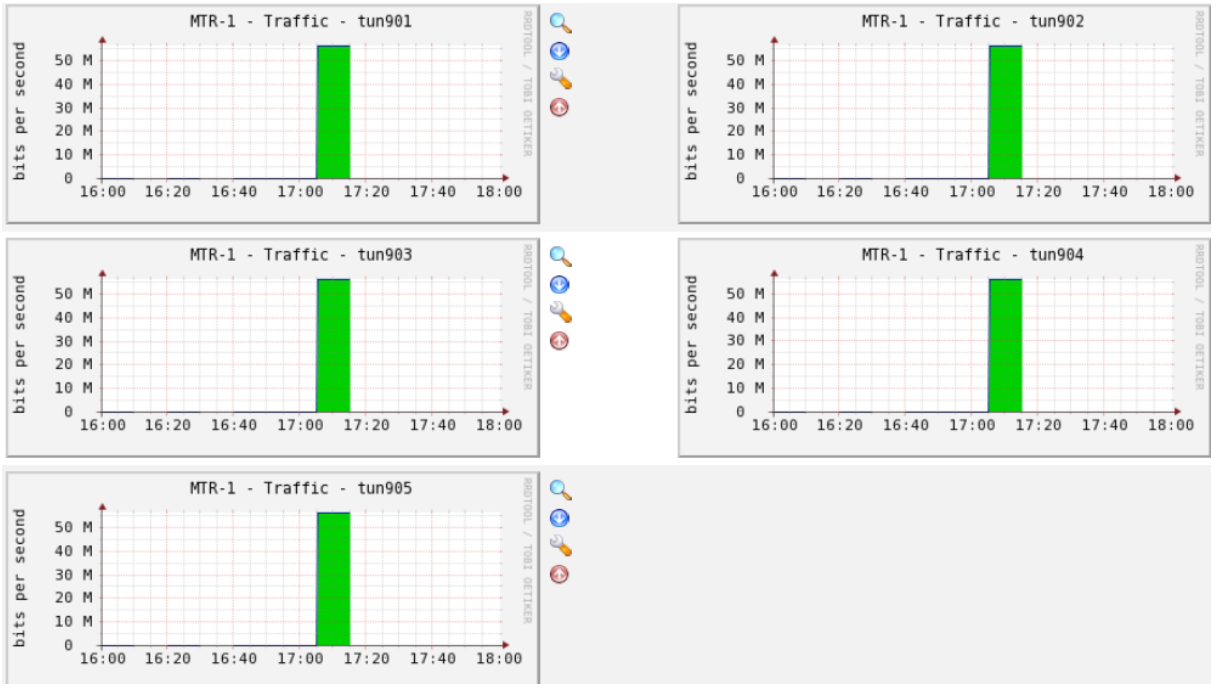

Figure 21:Traffic by crawler on the HiMoNN tunnels at the MTR1

When looking at the diagrams of Figure 21, it seems that before 17:05 and after 17:15 no data was sent. However, this is only due to the scale of the chart. This difference in scale shows that the normal data traffic is small compared to the jammer notification messages.

Findings: Throughput from the mobile domain was extracted without overloading the mobile links. Because of the full mesh of overlay tunnels for every radio technology (which was only a fallback because of the military off-the-shelf radio equipment), it was even possible to identify traffic to/from different nodes just using interface statistics. The results also clearly showed when nodes were isolated. Follow-up measurement samples need to be interpreted accordingly. However, identifying the cause (limited transmission range vs. jamming) will only be possible by correlating this data with cross layer information like the noise level.

OSPF-Topo: Monitoring of the OSPF topology of the mobile domain

Purpose: Providing real time information about the communication status and connectivity within the convoy with minimal/no communication overhead to the mobile component's OSPF domain. This includes the terrestrial radio links as well as the satellite links to the HQ.

Test setup: The OSPFv2 MIB of the non-mobile MTR located at the Multi National Deployed HQ is queried via SNMP by a software tool running also in the HQ. The OSPFv3 MIB is not yet supported by the MTR implementation based on Vyatta Linux. The information provided consists of a snapshot of the link state database of the MTR and thus provides a local view of the OSPFv2 topology of the mobile domain.

Walkthrough: The OSPFv2 link state information was shown on a computer located at the Multi National Deployed HQ and continuously provided information regarding the communication status of the convoy to the other experiments.

In addition, a packet capturing process was started at the MTR in the HQ to allow for an offline analysis of the routing protocol behavior (OSPFv2 and OSPFv3) in the post processing of the experiments.

Prerequisites and special requirements: Remote access to the MTR is needed.

Experiment Analysis:

OSPF-VIZ created for the analysis by each measurement four web pages with the following names (Graph, Device, Analyze, Area). When all vehicles stayed OSPF-VIZ generated these pages:

The tab "graph" shows the routers and the tunnel links in-between (see Figure 22).

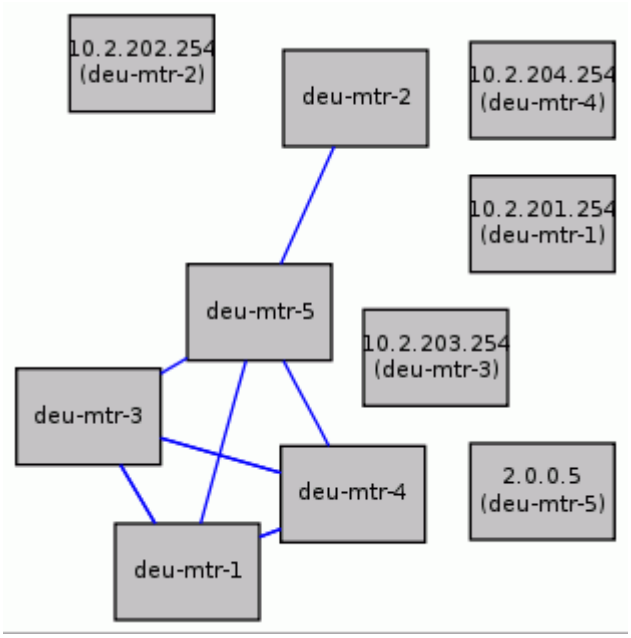

Figure 22:Picture of the topology by OSPF-VIZ

Observations:

- All MTRs are found.
- The tunnels from MTR2 are not captured.
- All routers have been shown twice.
- GraphVIZ need lots of fine tuning in the ospfviz.conf

The tab “Devices” shows a list of the network devices and tunnel interfaces per router. As an example here is a clipping from the list of devices on the MTR3 (See Figure 23)

Available OSPF routers

deu-mtr-1 deu-mtr-2 deu-mtr-3 deu-mtr-4 deu-mtr-5

Router deu-mtr-3

OspfRouterId 10.2.203.254, Router types are Backbone [top]
Router hardware is Vyatta 999.mtnapa.05031221

Interface	Description	neighbor IP/mask	Bandwidth	Area	Metric	Neighbor
eth0	DEU-MLAN-203	10.2.203.254/24	1 Gbps	0	noSuchInstance	
tun910	HiMoNN-Tunnel_DEU-3_DEU-4	10.254.34.96/24	n/a	0	noSuchInstance	deu-mtr-4
tun902	HiMoNN-Tunnel_DEU-1_DEU-3	10.254.13.95/24	n/a	0	noSuchInstance	deu-mtr-1
tun911	HiMoNN-Tunnel_DEU-3_DEU-5	10.254.35.96/24	n/a	0	noSuchInstance	deu-mtr-5

Figure 23: List of the network devices and the tunnels from the convoy by OSPF-VIZ

- All tunnels running OSPF2 are listed from MTR3
- Bandwidth will not be displayed with tunnels
- The list is consistent with the graph.

The site “Analyses” provide four pieces of information.

- Ping to the first router,
- The different areas with the routers on it,
- The area from which the test is started,
- The areas without authentication.

The tab “area” (See Figure 24) gives a summary of all OSPFv2 enabled interfaces sorted by areas. Since there is only one area in the test, there is only one list.

Available OSPF areas

0

Area 0

There are 18 links in area 0 [top]

link IP/mask	Interface	Bandwidth	Metric	Router Id
10.2.201.254 / 24	eth0	1 Gbps	0	10.2.201.254
10.2.202.254 / 24	eth0	1 Gbps	0	10.2.202.254
10.2.203.254 / 24	eth0	1 Gbps	0	10.2.203.254
10.2.204.254 / 24	eth0	1 Gbps	0	10.2.204.254
10.254.13.95 / 24	tun902	n/a	0	10.2.203.254
10.254.13.96 / 24	tun902	n/a	0	10.2.201.254
10.254.14.95 / 24	tun903	n/a	0	10.2.204.254
10.254.14.96 / 24	tun903	n/a	0	10.2.201.254
10.254.15.95 / 24	tun904	n/a	0	2.0.0.5
10.254.15.96 / 24	tun904	n/a	0	10.2.201.254
10.254.25.95 / 24	tun908	n/a	0	2.0.0.5
10.254.25.96 / 24	tun908	n/a	0	10.2.202.254
10.254.34.95 / 24	tun910	n/a	0	10.2.204.254
10.254.34.96 / 24	tun910	n/a	0	10.2.203.254
10.254.35.95 / 24	tun911	n/a	0	2.0.0.5
10.254.35.96 / 24	tun911	n/a	0	10.2.203.254
10.254.45.95 / 24	tun913	n/a	0	2.0.0.5
10.254.45.96 / 24	tun913	n/a	0	10.2.204.254

Figure 24: Summary of all network devices found by the different routers by OSPF-VIZ

The interface from which the measurement is started is not in the list.

Findings:

The program does not only query a single OSPFv2 router via SNMP for its Link State Database. It queries all other routers discovered in the first step to get some additional information about the system. This introduces additional communication overhead not wanted in a mobile scenario just to get non mandatory information.

Unfortunately, due to the lack of support of the OSPFv3 MIB, only a single router running OSPFv2 was shown. The other routers only enabled OSPFv3 and thus had IPv6 support only.

However, with small changes to the procedure of OSPF-VIZ (only query a single router) and OSPFv3 (MT) support, valuable information about the connectivity can be provided with minimal communication overhead by standardized management protocols when just querying the border router of the mobile domain, which is located in the HQ.

Jammer-Basic: Cooperative detection of the jammer

Purpose: Show that the detection of a jammer is possible within the convoy with cross-layer information aggregation of data from the radios.

Test setup: A cross-layer framework called CRAWLER [15][16] was installed on two dedicated routers equipped with Wi-Fi cards. One of the nodes was located on a German military vehicle. The other node was located on an NGO vehicle.

Walkthrough:

The CRAWLER framework needed to be installed and configured on both ends of the communication. The connection between the German military vehicle and the German NGO vehicle was established. Special jamming equipment was placed between both nodes. In addition, plausibility checks of cross-layer information were performed via the CRAWLER

framework. Thus, the presence of a possible jamming incident was detected based on this local information.

Prerequisites and special requirements: Jamming equipment for Wi-Fi as well as a military and an NGO vehicle equipped with Wi-Fi devices connected to the black part of the network and as well as the CRAWLER service. No dynamic routing was performed on this link.

Experiment Analysis:

The objective of these experiments is to demonstrate that the attack of a constant jammer can be detected in a mobile environment by using cross-layer design. For this purpose, CRAWLER and the proposed application for jamming detection and notification are utilized.

The test scenario was conducted on a military testing ground in Greding, Bavaria, Germany. This testing ground is surrounded by a spacious rural area permitting wireless communication without major disturbances.

Within the test scenario, a military vehicle was instructed to escort an NGO in order to protect the NGO from adversaries including jamming entities. Therefore, both vehicles were equipped with embedded computers running CRAWLER and the application for jamming detection and notification.

Starting from outside of the jammer's area of influence, both vehicles were ordered to drive along a road with constant speed, as outlined and illustrated in Figure 25 and Figure 26. Prior to this, a constant jammer, Figure 27 and Figure 28, had been hidden next to the road. This jammer was equipped with a directional antenna and an additional amplifier of 1W enabling the jammer to disrupt wireless communication entirely, up to a distance of at least 460m. Because of that, both vehicles that were approaching the jammer during the test scenario, were affected by the noise signal of the jammer at some point in time.

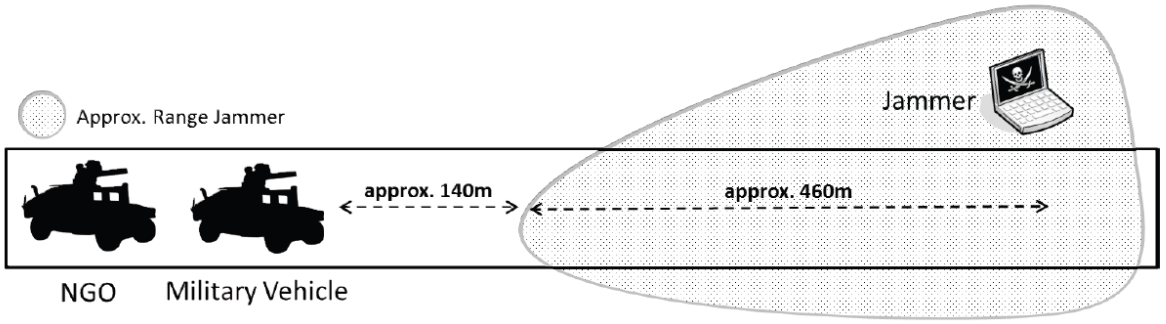

Figure 25: Outline of the test scenario. The military vehicle and the NGO were driving along a road, starting around 600m away from the constant jammer. After approximately 140m, the vehicles reached the affected area of the jammer and further approached the jammer until they reached the jammer after approximately further 460m.

The test scenario was repeated several times with varying parameters including vehicle speed, distance between convoy units, the direction the jammer is approached from, and finally the strength of the noise caused by the jammer. These test setups are summarized in Table 3. Note, that all experiment results and logs were recorded at the military vehicle.

In addition to this, the experiments have different durations because the vehicles did not move directly after an experiment had started or the vehicles stayed in the affected area of the jammer for a while.

Figure 26: Use case scenario

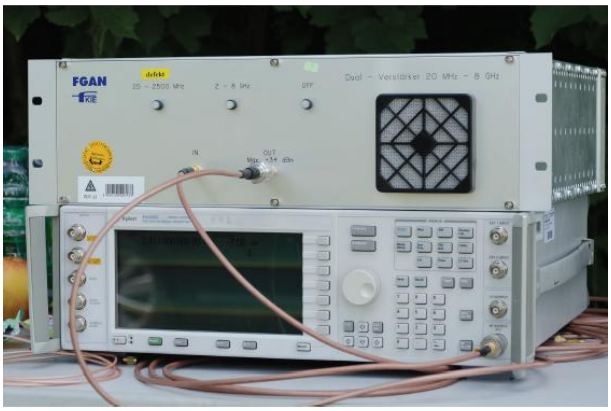

Figure 27: Constant jammer with 1W amplifier

	Detection Parameters for CRAWLER			Experiment parameters			
Experiment ID	**PDR**	**Noise**	**Packet Exchange Interval**	**Distance**	**Speed**	**Direction**	**Jammer (noise strength)**
1	65%	-114 dBm	1s	20m	20km/h	To jammer	-13dBm
2	65%	-114 dBm	1s	30m	10km/h	To jammer	-13dBm
3	65%	-114 dBm	1s	30m	30 km/h	To jammer	-13 dBm
4	65%	-114 dBm	1s	30m	30 km/h	To jammer	-19 dBm
5	65%	-114 dBm	1s	30m	30 km/h	To jammer	-25 dBm

Table 3: Detection parameters and other important parameters that describe the different experiments that were conducted in the use case scenario

Experiment (1)

In the first experiment, NGO and military vehicle were 20m apart. They drove with a constant velocity of 20km/h in the direction of the constant jammer whose antenna was directed at the approaching vehicles, according to Table 3. Based on this scenario, Figure 29 shows that the conditions of the wireless channel were stable at the beginning of the experiment, i.e. outside the range of the jammer. In addition, communication between both vehicles was possible reliably outside the range of the jammer. For instance, the data that has been sent by the NGO could be decoded successfully by the military vehicle as indicated in Figure 29 by the high maximum PDR value at the beginning.

Figure 28: Picture of the constant jammer's hiding place. It shows the jammer with the directed antenna hidden next to the road.

With decreasing distance to the constant jammer, both vehicles entered the area that was affected by the jammer. This resulted in an increase of the noise level, -62dBm at maximum, and in a reduced RSS of -82dBm at minimum (Note that the RSS depends on the current noise characteristics. Therefore, it is reduced in presence of the jammer). The collisions between packets, sent from the embedded computers inside the vehicle, and the jamming signal caused the maximum PDR to drop until a PDR of 0% has been reached. As a result, the jammer was detected and sending of a notification message could be initiated.

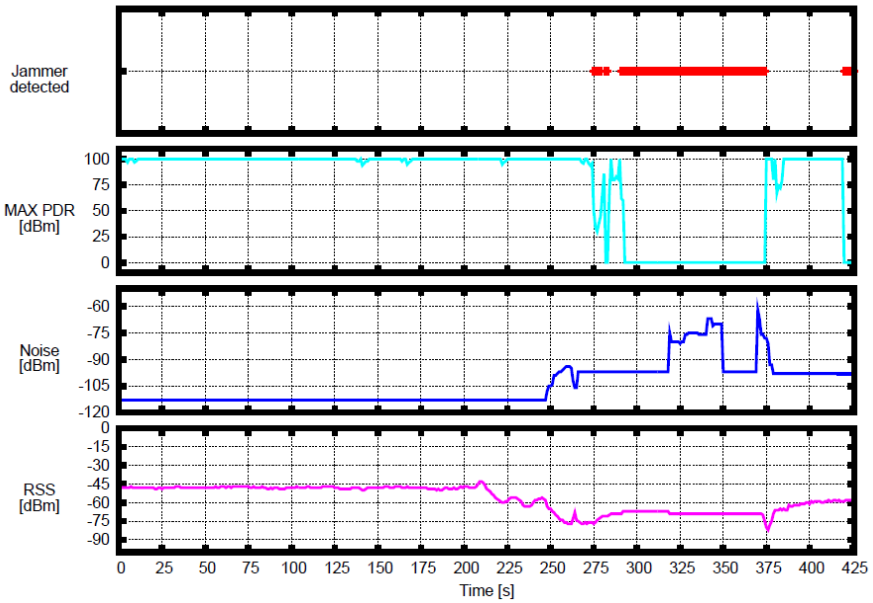

Figure 29: Results of the first experiment of the use case scenario. Both vehicles were 20m apart and drove with nearly constant speed of 20km/h. The constant jammer was emitting a jamming signal of -13dBm amplified by 1W.

Experiment (2)

In the second experiment (see Figure 30), the characteristics of maximum PDR, noise, and RSS were close to those of the first scenario in spite of the increased distance between both vehicles. Noise rose again when approaching the constant jammer. In addition, the PDR collapsed in range of the jammer, thus leading to a reliable detection. Actually, the constant jammer was so effective for a short period of time (around second 350) that no RSS measurements have been reported to cfg80211 as no packets were received correctly.

Considering the initial noise level in Figure 30, this is the only difference to the first experiment. The actual reason for this remains unclear as the vehicles started outside the range of the jammer and no disturbances should be present. However, a possible assumption is that the ANI (Adaptive Noise Immunity) performed by the Atheros wireless network cards of the embedded computers had increased its immunity level in a previous experiment due to jamming. Hence, the immunity level was decreased in this experiment, i.e. the sensitivity of the receiver was increased, because of the absence of a jammer leading to a reduced number of false packet detections.

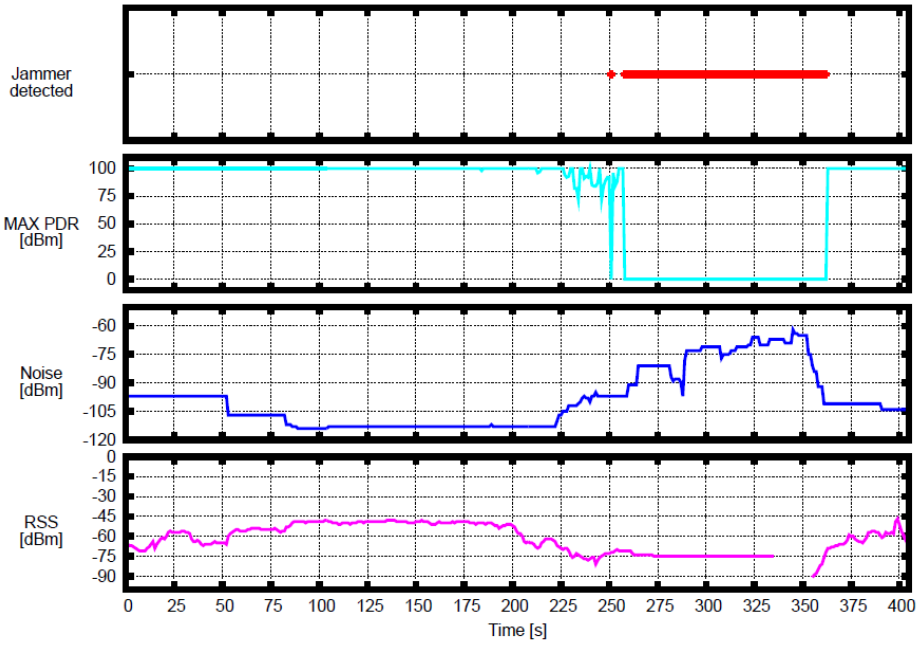

Figure 30: Results of the second experiment of the use case scenario. Both vehicles were 30m apart and drove with nearly constant speed of 10km/h. The constant jammer was emitting a jamming signal of -13dBm amplified by 1W.

Experiment (3)

The observations of the third experiment, as illustrated in Figure 31, were again very similar to the results of the first experiment although distance (30m) and speed (30km/h) were increased. As a result, the jammer could be detected reliably when entering the area affected by the jammer. Nevertheless, it is to remember that the shorter duration of the experiment is caused by the fact that both vehicles were driving directly at the beginning of the experiment.

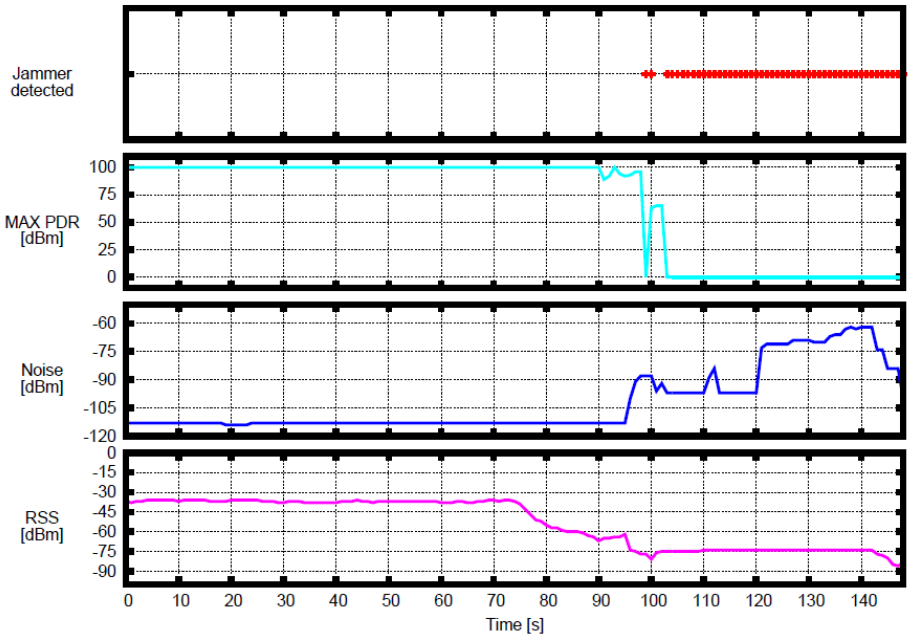

Figure 31: Results of the third experiment of the use case scenario. Both vehicles were 30m apart and drove with nearly constant speed of 30km/h. The constant jammer was emitting a jamming signal of -13dBm amplified by 1W.

Experiment (4)

In general, the results of this experiment are similar to the previous ones, i.e. the jammer could be detected reliably. Nevertheless, some interesting new insights can be gained in Figure 32 as the vehicles were located in range of the jammer for a longer period of time. There, it is shown that the noise level was fluctuating although the vehicles stayed in the proximity of the jammer. Thus, it is assumed that this is caused by ANI which probes different noise immunity configurations (in discrete steps) due to false signal detections. Despite of ANI, the embedded computers were not able to decode received packets correctly or rather did not detect packets anymore indicated by the maximum PDR at 0% and the missing RSS reports. As a result, communication was not possible.

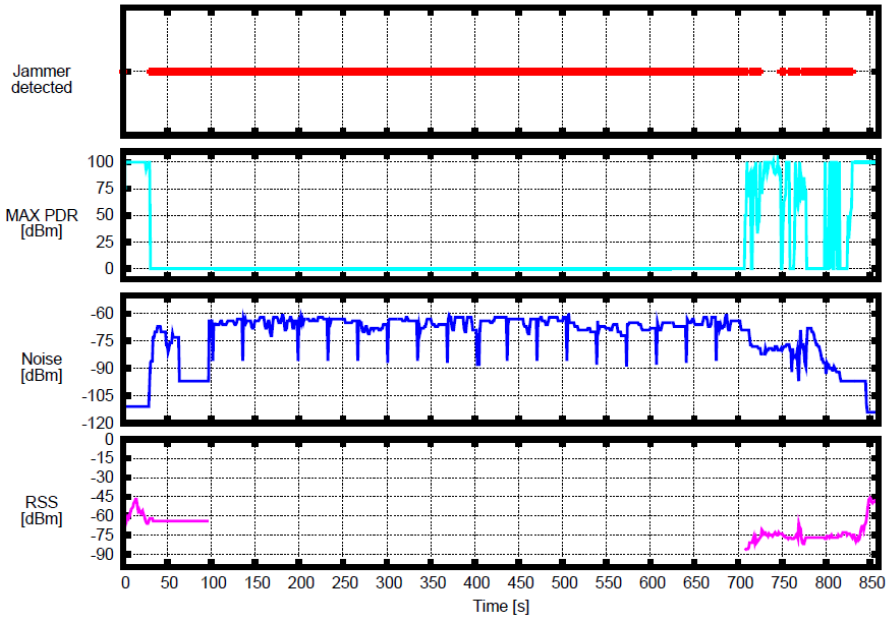

Figure 32: Results of the fourth experiment of the use case scenario. Both vehicles were 30m apart and drove with nearly constant speed of 30km/h. The constant jammer was emitting a jamming signal of -19dBm amplified by 1W.

Experiment (5)

Finally, the results of the last experiment where the jammer emits a weaker noise signal, -25dBm, compared to previous experiments are illustrated in Figure 33. There, it can be observed that the PDR was not affected much, except a sporadic drop, when NGO and military vehicle had been in range of the jammer. Thus, an increase in noise did not cause a drop in the maximum PDR directly. As a result, the communication between both vehicles was not affected until they were close to the jammer. Hence, the jammer was detected not till then.

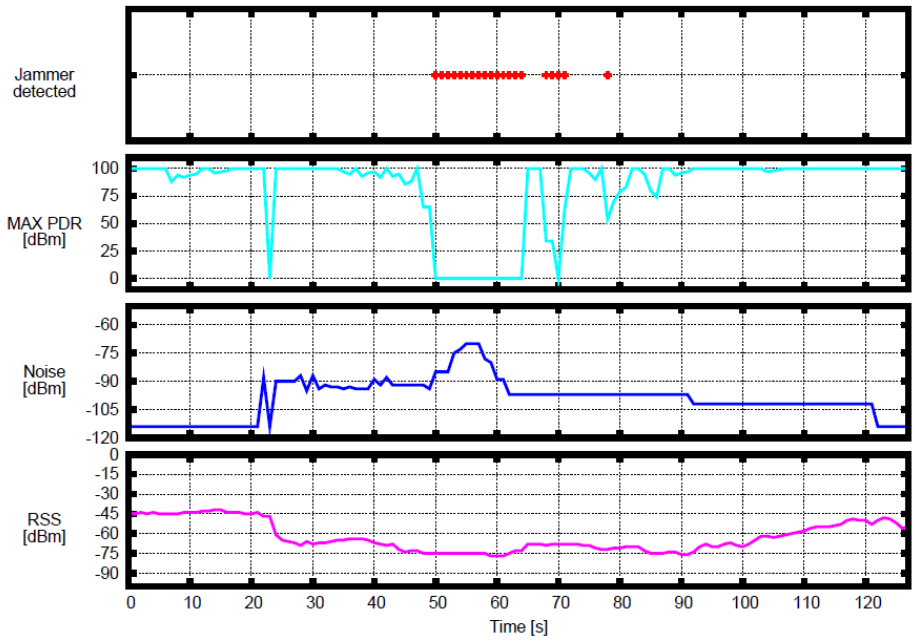

Figure 33: Results of the fifth experiment of the use case scenario. Both vehicles were 30m apart and drove with nearly constant speed of 30km/h. The constant jammer was emitting a jamming signal of -25dBm amplified by 1W.

Jammer-Notification: Automated notification of the HQ by the CRAWLER application via the Operational Message Service (OMS)

Purpose: Provided by task 2, the Operational Message Service (OMS) is a notification-based service intended to distribute commands, information, warnings and alerts. By using the OMS to raise the alarm about a suspected jamming incident, two purposes are answered: One, the OMS is notification-enabled and allows any interested party to subscribe to the alerts, without the necessity of setting up any new information structures or configurations; and two, it is a good example of a task-comprehensive test. Note: Since the OMS provider and the notification broker are located in the red network and CRAWLER in the black, a cross domain guard was needed to allow for a controlled forwarding of the message.

Test setup: The CRAWLER application includes a WS-Notification client with a publish method set up for the Operational Message Service to send an alert about a detected potential jamming attack to a WS-Notification broker (see Figure 34). The German and Norwegian portable Command and Control Information System (C2IS) subscribed to alerts of this kind and were capable of displaying the jamming incident at the geographic location in the GUI.

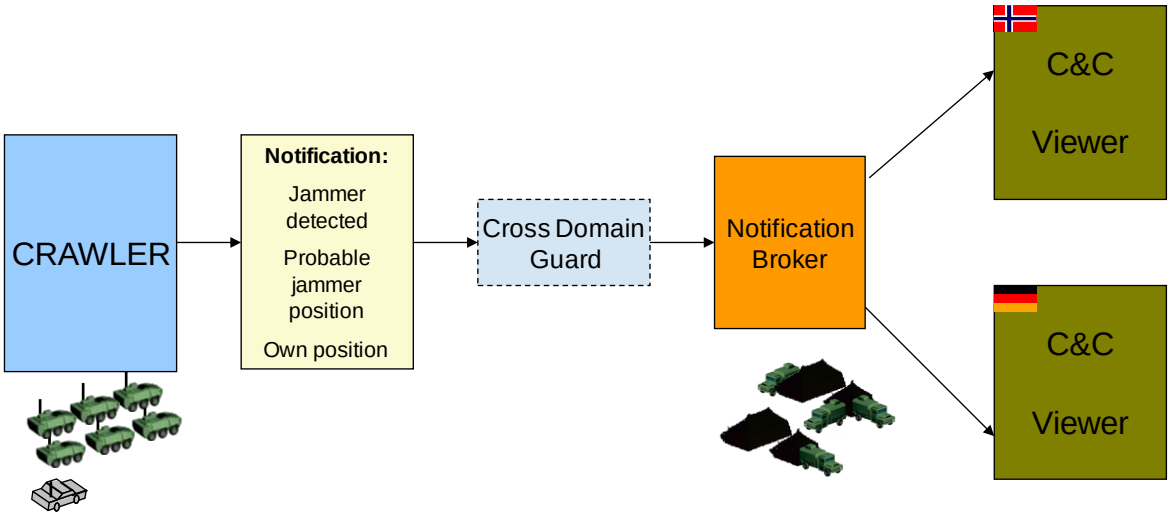

Figure 34: Test setup for the Jammer Notification massage

Walkthrough: Upon detection of a potential jamming incident an alert was sent via Operational Message Service. Subscribers, namely the German portable C2IS system, received the alert and processed it. The content was displayed in the C2IS GUI.

Experiment Analysis:

The notification messages that are transmitted to the HQ traverse several components until the message is displayed. These are the Filter (or the cross domain guard tool), the Notification Broker, and the Notification Consumer.

The Filter is a simple tool that enables messages from the insecure black network to enter the secure red network. It is a simple TCP proxy. It serves as a placeholder for a Cross Domain Guard required for an operational system. The Notification Broker and the Notification Consumer are required for the transmission of the notification message. They are working as defined by the WS-Brokered Notification specification, which is part of the WS-Notification specification enabling a notification pattern by a publish/subscribe functionality. The underlying principle is shown in Figure 35. There, the broker (Notification Broker) represents a service that allows interested services like the consumer (Notification Consumer) to subscribe to the broker. In parallel, the broker waits for new messages, e.g. the notification of a detected jamming attack, to be published by the producer (the application for jamming detection and notification). Finally, the broker notifies the subscribed consumers in case a message has been published by the producer.

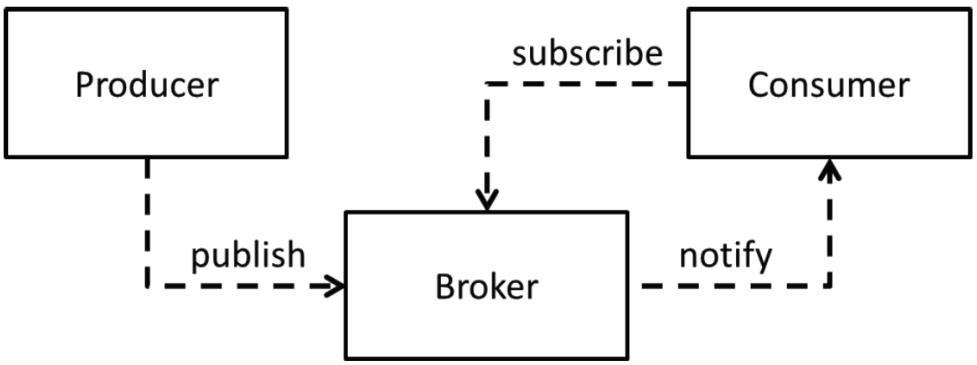

Figure 35: Principal concept of the WS-Brokered Notification

Findings: Upon detecting the constant jammer, the jammer detection and notification application was able to reliably send the notification message(s) over a not-jammed connection (over the HiMoNN connection) to the HQ.

Remark: The interval of sending periodic notification messages is an important parameter due to link/application overloading issues.

Summary

To summarize the experiments, the application for jamming detection and notification using CRAWLER was able to detect the constant jammer reliably in spite of mobility and other effects introduced by the environment. In particular, those effects did not influence the detection strategy, based on PDR and noise, significantly. However, the experiments were conducted in a spacious rural area without major disturbances. It remains to show how the metrics such as PDR and noise are affected in other scenarios, e.g. in an urban environment, or under a different mobility pattern.

4. Conclusions and Future Work

We introduced both challenges and technical solutions for federated network management with special requirements regarding security and information hiding, as well as addressing the wireless and core domains. Building on the Protected Core Networking concept, we showed how a well established framework PerfSONAR for sharing network performance measurements between different research networks can be extended and applied to the CoNSIS scenario. In addition, we presented the first prototype of an architecture that can use the performance measurements to adjust SLAs between the different nations of the coalition.
We conclude with details about the experiments performed within the CoNSIS field tests. These tests and some earlier experiments lay the foundations and serve as proof-of-concept.

Some of the important findings regarding the experiments are:

- The overhead introduced by a combination of passive segment monitoring and active end-to-end probing, is small compared to the overall network traffic in the CoNSIS convoy scenario.
- Every measurement probe has a parameter area where it provides valid results. Outside this area the results are not reliable.
- Cross-layer solutions can optimize network and operational performance.

The identified functional gaps, as well as the results of the experiments will set the agenda for the second phase of the CoNSIS project.

In the area of *network performance management*, CoNSIS II will focus on passive measurement techniques to assess network performance, a tighter integration of measurement services into the SOA architecture, and network monitoring services with a higher level of abstraction.
Regarding *network configuration management*, CoNSIS II will focus on dynamic Service Level Agreements, mapping of Service Level Agreements to network/device configurations, and standardized interfaces for network configuration.
In addition, an integration of cross-layer mechanisms into network management will be examined.
Please note, that these research and experimentation topics are well aligned to the NATO activities regarding STANAG 4711 [23][24], addressing many of the requirements for the later deployment phases described in these documents.

5. References

[1] NATO Network Enabled Capability Feasibility Study Executive Summary v. 2.0, October 2005.

[2] G. Hallingstad and S. Oudkerk, “Protected core networking: an architectural approach to secure and flexible communications”, Communications Magazine, IEEE, 2008, 46, pp. 35 -41

[3] PerfSONAR Homepage <http://www.perfsonar.net/> (last accessed May 24, 2012).

[4] perfSONAR MDM release 3.0 – Product Brief.

[5] Instantiating a Global Network Measurement Framework http://acs.lbl.gov/~tierney/papers/perfsonar-LBNL-report.pdf.

[6] P. Steinmetz, “Use of Cross Domain Guards for CoNSIS network management“, MCC 2012, Gdansk, Poland

[7] PerfSONAR: A Service Oriented Architecture for Multi-domain Network Monitoring.

[8] P. Jacobs and B. Davie , “Technical challenges in the delivery of interprovider QoS”, Communication Magazine, IEEE, Vol. 43, No. 6, 2005, pp. 112-118.

[9] D. Duda et al., “The QoS Policy Agreement System for Federation of Communications and Information Systems”, MCC 2011, Amsterdam, The Netherlands, Oct. 2011.

[10] M. Hauge, M.A. Brose, J. Sander, and J. Andersson, "Multi-topology routing for improved network resource utilization in mobile tactical networks," *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010* , vol., no., pp. 2223-2228, Oct. 31 2010-Nov. 3 2010.

[11] M. Hauge, CoNSIS Task 1, “QoS-classes for the CoNSIS test and demonstration architecture”.

[12] “Cacti - The Complete RRDTool-based Graphing Solution”, http://www.cacti.net/ (last accessed June 13, 2012).

[13] A. Morton and S. Van den Berghe, “Framework for Metric Composition”, IETF RFC 5835, December 2010.

[14] A. Morton and E.Stephan, “Spacial Composition of Metrics”, IETF-RFC6049, January 2011.

[15] I. Aktas, J. Otten, F. Schmidt, and K. Wehrle, “Towards a Flexible and Versatile Cross-Layer-Coordination Architecture,” Proceedings of the 29th International Conference on Computer Communications (INFOCOM 2010), pp. 1-5, March 2010.

[16] I. Aktas, F. Schmidt, M. H. Alizai, T. Drüner, and K. Wehrle, “CRAWLER: An Experimentation Architecture for System Monitoring and Cross-Layer-Coordination,” Proceedings of the 13th International Symposium on a World of Wireless, Mobile and MultimediaNetworks (WoWMoM'12), pp. 1-9, June 2012, in press.

[17] “System and Experimentation Architectures - Version 1.0”, CoNSIS/Task 5/DL/002, September 2011

[18] “IP Network Metrology –Architectures and Tools applicable in a coalition network”, Document CoNSIS CG/UM-ESIO/IDRE/10.107/V1.1, July 2010

[19] “Coalition Management Philosophy”, CG/UM-ESIO/IDRE/11.029/V1.0, January 2011

[20] “Management Organisation in a C-TNS”, CG/UM-ESIO/IDRE/11.062/V1.0, March 2011

[21] “Fair Queuing and active Measurement Methods”, CG/UM-ESIO/IDRE/11.260/V1.0, November 2011

[22] Fatih Abut, “Jamming Indicators in Wireless Networks” – CoNSIS Task 4, Nov. 2012

[23] STANAG 4711, Interne Protocol Quality of Service (IP QoS) (draft 3), NATO Standardization Agency, Dezember 2011 (NATO/EAPC Unclassified)

[24] R.M. van Selm, G. Szabo, R. van Engelshoven, R. Goode, NATO Consultation, Command and Control Agency Technical Note TN-1417 draft release candidate 10 IP QoS Standardization for the NII NC3A, Den Haag, Januar 2011 (NATO Unclassified

[25] P. Steinmetz, “Multilevel Security and network management in CoNSIS”, Fraunhofer FKIE Technical Report, 2012