Cross-layer design and optimizations

Frank T. Johnsen, Joakim Flathagen, Mariann Hauge, Eli Gjørven, Terje M. Mjelde and Frode Lillevold
Cross-layer design and optimizations

Frank T. Johnsen, Joakim Flathagen, Mariann Hauge, Eli Gjørven, Terje M. Mjelde and Frode Lillevold

Norwegian Defence Research Establishment (FFI)

3 July 2014
FFI-rapport 2014/00985

1312


Keywords

Krysslag
Ledelsessystem
Nettverksbasert Forsvar

Approved by

Ole-Erik Hedenstad       Project Manager
Anders Eggen             Director
English summary

Cross-layer optimization (CLO) is an approach that coordinates protocol behaviors at different layers to improve overall system performance. There are four approaches to CLO identified: 1) Creation of new interfaces, 2) Merging of adjacent layers, 3) Design coupling without new interfaces, and 4) Vertical calibration across layers. Further, there are two orthogonal approaches to implementing CLO, that is, the so-called non-manager method and the manager method. The fundamental difference between these two is that the former modifies and extends an existing protocol stack with CLO functionality, whereas the latter creates a separate, vertical control plane which handles all CLO functionality. We think the latter is the best approach to implementation, since it allows the protocol stack to retain backward compatibility with non-CLO systems.

In the report we survey CLO approaches in selected areas: Wireless networks, middleware, information assurance, service management and control, wireless sensor networks, and video streaming, and discuss how this knowledge can be used to enhance the Quality of Service of communication. CLO is important to consider in the tactical domain, because it can potentially make better use of the available resources than a pure, traditional layered design.

The backdrop of the report is FFI-project 1312, which aims to investigate conceptual approaches towards realizing the next generation C2 system for the Norwegian armed forces in the land domain. The overall goal is reaching NML4. The report gives a few suggestions and guidelines regarding CLO, and how further research can be focused within the presented areas in order to support the project towards defining an NML4 capable C2 capacity.
Sammendrag


I rapporten diskuterer vi krysslagsløsninger innen utvalgte områder: Trådløse nettverk, mellomvare, informasjonssikkerhet, tjenestehåndtering og kontroll, trådløse sensornett, og direkteavspilling av video. Videre argumenterer vi for hvordan denne kunnskapen kan brukes til å forbedre tjenestekvaliteten i et kommunikasjonssystem. Krysslagsløsninger muliggjør en bedre utnyttelse av tilgjengelige ressurser enn et tradisjonelt lagedelt design. Derfor er det spesielt viktige å vurdere bruken av slike løsninger i taktiske nett.

1 Introduction

FFI project 1312 “Taktisk ledelsessystem for landdomenet” (TLL) supports P8043. The aim is to describe a conceptual approach towards realizing the next generation C2 system for the Norwegian armed forces in the land domain. The idea is to explore and develop this concept in key with the overall project vision, which is further described in [94]. In short, the goal is to develop the conceptual solution for a C2 system. In project 1312 we adopt the term “system” in the broadest sense of the word, i.e., encompassing all aspects of DOTMLPFI\(^1\) related to providing and maintaining a C2 capacity.

The scope of this report is limited to part of the C2 system’s communication aspect, more precisely the communication stack of C2IS. This means that we only address a subset of DOTMLPFI here. We focus on the technical aspects, placing us mostly in the dimensions related to Material (encompasses e.g., the information infrastructure) and Interoperability (we are only concerned with technical interoperability, the lowest form for interoperability, in this report). We do not address networks with fixed infrastructure. We do not address the other aspects of DOTMLPFI here, but if one chooses to pursue certain of the approaches we suggest in this report, then the impact may be beyond that of the Material and Interoperability lines (for example, certain new functionality will require additions to or modifications of the “T” (Training and Education) aspect). Finally, as TLL does not focus on strategic networks, we limit our focus to C2 in the tactical domain, meaning that we are only concerned with communication solutions that are viable in the tactical domain in this report.

The goal in TLL is for the system to achieve NATO Network Enabled Capability (NNEC) Maturity Level 4 (NML4 — “Collaborate”) [97], which requires certain aspects to be fulfilled by the system:

- This level of maturity is characterized by continued transformational improvements especially in situational awareness, interoperability and adaptive planning and execution.
- A common unified infrastructure based on a single network will allow the seamless sharing of data and facilitate large scale advanced horizontal and vertical interactive collaboration for planning and execution.
- Technically, a force at NML 4 is characterized by flexible services providing coherent functionality using information that resides in the common infrastructure.
- Important aspects that need to be solved efficiently include such functionality as being able to discover services and information and publishing/subscribing to information.

In the above, we see that there is an ambitious requirement for users at all operational levels to seamlessly exchange information. In order to achieve efficient information exchange between these users, one needs to work with different types of information and communication systems.

Systems and equipment used at the various levels are different, and the information exchange must be adapted to fit the capacity of the systems used.

The NATO Network Enabled Capability (NNEC) Feasibility Study [6] presents a discussion of technology, focusing on the needs of future interoperable military communications. An information infrastructure will have to allow for communication across system and national boundaries while at the same time taking legacy systems into account. This leads to a requirement for a flexible, adaptable and agile information infrastructure which can support all the information needs of national forces, and at the same time support interoperability. For the network level, the study highlights NATO’s ”Everything over IP” strategy. As a consequence we consider only IP based networks in this report. Further, the study identifies the Service-Oriented Architecture (SOA) concept and Web services technology as the key enablers for NNEC. Thus, we limit our discussion of middleware to Web services in this report (for an overview of other middleware, see [68]).

We present the traditional layered approach to building communication systems, before we move on to discuss other approaches that can be considered for TLL. More precisely, how we perceive that optimizations can be leveraged to utilize the scarce network resources in the tactical domain better (by “better” we mean optimizations along multiple axis, such as improved Quality of Service, improved management, and improved security). It is important to note that this report is not intended to be a complete survey of all areas that can benefit from cross-layer optimization. Rather, it provides an overview of how cross-layer optimization relates to the authors’ respective fields of expertise.

2 Approaches to cross-layer optimization (CLO)

Traditionally, protocol architectures follow strict layering principles, which ensure interoperability, fast deployment, and efficient implementations. However, cross-layer optimization can be a beneficial approach in many settings.

2.1 Layered design

The Open Systems Interconnection (OSI) Model introduces seven distinct layers that form a communication stack where each layer solves and is responsible for a specific task related to communication (see Figure 2.1). By separating the network communication into logical smaller pieces, the OSI Model simplifies how network protocols are designed.

Successful and widely adopted standards all utilize layering to a certain extent, as the overview in Figure 2.2 shows. Seldom are all seven layers of the OSI model implemented in practice. For example, it has become mainstream to consider a so-called hybrid model [98] when discussing TCP/IP networks. Here, one collapses OSI layers as is done in the four layers of the DoD model, yet one retains the physical layer as well, which yields the following five layer TCP/IP model (see Figure 2.3).
Figure 2.1 The Seven Layers of the OSI Model (from [78])

Application (7)
Provides services directly to user applications. Because of the potentially wide variety of applications, this layer must provide a wealth of services. Among these services are establishing privacy mechanisms, authenticating the intended communication partners, and determining if adequate resources are present.

Presentation (6)
Performs data transformations to provide a common interface for user applications, including services such as reformating, data compression, and encryption.

Session (5)
Establishes, manages, and ends user connections and manages the interaction between end systems. Services include such things as establishing communications as full or half duplex and grouping data.

Transport (4)
Insulates the three upper layers, 5 through 7, from having to deal with the complexities of layers 1 through 3 by providing the functions necessary to guarantee a reliable network link. Among other functions, this layer provides error recovery and flow control between the two end points of the network connection.

Network (3)
Establishes, maintains, and terminates network connections. Among other functions, standards define how data routing and relaying are handled.

Data-Link (2)
Ensures the reliability of the physical link established at Layer 1. Standards define how data frames are recognized and provide necessary flow control and error handling at the frame level.

Physical (1)
Controls transmission of the raw bitstream over the transmission medium. Standards for this layer define such parameters as the amount of signal voltage swing, the duration of voltages (bits), and so on.
As the TCP/IP model is frequently encountered in literature related to the Internet as well as both layered and cross-layer protocol design and optimizations in general, we will be using it for the remainder of the report.

The layered approach has provided several advantages for network designers. First, by working in layers, the implementation and design effort can be parallelized. Thus, designers can independently focus on particular layers with the assurance that the final system will interoperate. This makes it possible to upgrade individual modules (e.g., the routing protocol) without necessitating a redesign of the complete system. Second, by defining one layer in the model as the “narrow waist”, interconnection between different networks is possible. In the TCP/IP model for example, the network layer (IP) constitutes this narrow waist. Third, layered models provide natural abstractions to deal with. This increases the synergy between research efforts and facilitates the progress towards working systems.

In wired networks layered network protocols have been tried and tested, and stand out as efficient and functional approaches to providing networking functionality. However, in wireless environments there has been an increase of novel approaches and solutions that involve violating the
traditional layered design (see, e.g., [35, 36]). As tactical environments involve wireless communication, it makes sense to have a glance at recent developments in civil systems and within NATO relating to optimizations that go beyond the traditional layered design – so-called cross-layer optimizations:

Cross-layer optimization (CLO) is an approach that coordinates protocol behaviors at different layers to improve overall system performance [108].

Although strict boundaries between the layers have several advantages, there is always a temptation to take architectural shortcuts to increase the performance.

2.2 Cautionary remarks

Cross-layer design refers to protocol design done by actively exploiting the dependence between protocol layers to obtain performance gains [95]. By doing this, cross-layer solutions may violate the modular layered approach. A violation of a layered architecture involves giving up the luxury of designing protocols at different layers independently. Such optimizations should therefore be used with caution as cross-layer interactions can have undesirable consequences on system performance [55]. However, in some situations, cross-layer interactions are inevitable to eliminate the redundancies associated with repeating similar tasks found on adjacent layers [96].

2.3 Areas that can benefit from CLO

CLO has been identified as an approach to improve many areas of wireless communication [35, 36]. Overall, there are three distinct system aspects that can benefit from CLO in particular, as shown in Figure 2.4: Security, Quality of Service (QoS), and Mobility.

2.3.1 Security

Security encompasses everything related to confidentiality, integrity, and availability. In short, all aspects pertaining to information assurance. Information assurance is covered in further detail in Section 4.1.

2.3.2 Quality of Service (QoS)

In [103], Quality of Service (QoS) is defined as

\[
\ldots\text{the set of those quantitative and qualitative characteristics of a distributed multimedia system necessary to achieve the required functionality of an application.}
\]

The Internet provides no QoS beyond the best effort treatment of traffic. High quality communication is ensured by over-provisioning a network so that capacity is based on peak traffic load estimates.
In contrast to the over-provisioned Internet, TLL is based on radio links with very limited bandwidth. Hence, it is essential that TLL has a QoS architecture. The QoS architecture must perform resource management and admission control (at the ingress of the network) as well as handling network congestion and traffic prioritizing (in the forwarding elements in the network).

In TLL it might be necessary to prioritize data packets by tailoring packet queues and packet scheduling to effectuate the delay requirements according to a QoS class. In addition, it might be necessary to prioritize according to the military priority of the packet. This means that the forwarding entity, and the lower protocol layers need application layer information. Differentiated Services (DiffServ)\(^2\) has become the de-facto standard for providing such functions in IP networks and is basically a method to transfer packet priority levels between communication layers. DiffServ defines Per Hop Behavior (PHB) of aggregate flows and acts upon packet classification values in the DSCP field in the IP header.

It is evident that QoS is a very broad category, including aspects like timely delivery, reliability, priorities and preemption. In this report QoS is a topic in almost all the CLO we discuss, ranging from networking aspects to middleware needs and video streaming requirements.

---

\(^2\)DiffServ relies on tagging the IP header’s type-of-service field with a bit pattern corresponding to a certain traffic class. This enables DiffServ-enabled routers to prioritize the IP packets accordingly with a deterministic per-hop-behavior (PHB). DiffServ is covered by several RFCs: RFC 2474 [77] (Definition of the Differentiated Services Field in the IPv4 and IPv6 Headers), RFC 2475 [11] (An Architecture for Differentiated Services), RFC 2597 [48] (Assured Forwarding PHB Group), RFC 3140 [10] (Per Hop Behavior Identification Codes), RFC 3246 [23] (An Expedited Forwarding PHB), RFC 3260 [41] (New Terminology and Clarifications for DiffServ), and RFC 4594 [5] (Configuration Guidelines for DiffServ Service Classes).
2.3.3 Mobility

Improving mobility includes all aspects related to making the network more robust and responsive when nodes move. This includes, but is not limited to, making routing and neighbor discovery more efficient and allow faster convergence. Also, functionality like service management and control is affected by mobile nodes, and requires special attention in e.g., MANETs. Service management is discussed in Section 4.2.

2.4 Categorizing approaches

One of the first attempts to combine functions from several layers was Integrated Layer Processing (ILP) [20]. The purpose of ILP was to combine the data manipulation functions from several protocol layers in a single processing loop. Now, the use of architectural shortcuts, actively exploiting the dependence between protocol layers for performance gains, is commonly referred to as cross-layer interactions. That being said, a CLO solution may be elaborate and involve all layers of the communication stack, or it may merely involve small, subtle changes like merging two existing layers, or exposing information that is available in one layer to others, despite the lack of standardized interfaces to do so. In general, different cross-layer proposals can be categorized in the following four categories defined by Srivastava and Motani [95]:

1. Creation of new interfaces.
2. Merging of adjacent layers.
3. Design coupling without new interfaces.

In the first category, new interfaces can be created both downward from a higher layer to a lower layer and upward from a lower layer to a higher layer. One example of downward communication is a routing protocol that dictates the radio transmit power, as in the work [19]. An example of upward communication could be an application layer protocol taking advantage of the Received Signal Strength Indicator (RSSI) from the physical layer. This approach of creating new interfaces is common in many CLO schemes, for example as seen in Wireless Sensor Networks (WSNs) (see Section 5.1) and middleware (see Section 3.3).

The second type of cross-layer designs is to merge adjacent layers. As we have seen in Figure 2.2, many standards have been designed this way, i.e., collapsing one or more OSI layers into one layer. In experimental solutions this is also a quite common approach. For example, for WSNs there are several proposed schemes that involve a complete integration of the MAC protocol and the routing protocol. One example of such integration is [90].

The third category involves coupling of two or more layers without any extra interfaces for information sharing. Here, mechanisms in one layer implies that another layer is capable of performing certain operations. Hence, it may not be possible to replace one of these layers without changing the other layer.
The final category involves setting parameters across several layers. For example the application layer can dictate certain operations at the network layer, which in turn dictates the preferred modulation at the physical layer. The use of cross-layer interactions for wireless networks, mobile ad hoc networks (MANETs), WSNs, and other areas can have a wide range of motivating factors such as improving reliability, throughput, lowering energy consumption, and so on.

2.5 Implementing cross-layer interfaces

A variety of architectural frameworks have been proposed to address some of the common challenges in introducing cross-layer optimizations (see, e.g., [37]). These frameworks enable the use of cross-layer optimizations without violating the architecture or creating dependencies that hinder future system extensions. However, these frameworks are not free of costs. First, the desire for generic architectures and frameworks has the drawback of added complexity. Second, the memory footprint and the extra processing required for these frameworks can sacrifice performance for energy and memory constrained platforms. Further, schemes may require synchronizing CLO information with other nodes across the network. In a bandwidth-constrained environment like a tactical network, this is clearly undesirable unless the gains are significantly greater than the drawbacks.

There are two orthogonal approaches to information sharing between the layers [36]: CLO can be classified according to this, in which case the first group of cross-layer designs employs the non-manager method (see Figure 2.5) and the second group leverages the manager method (see Figure 2.6). In the non-manager approach the data exchange takes place directly between any two layers, whereas in the manager method there is a vertical plane to manage data exchange between the layers. For interoperability reasons the manager approach is preferable, as the communication stack remains vertically unmodified. This means that it retains its interoperability in communication with unmodified stacks, which is a desirable property in a federated environment where everyone may not utilize the same optimizations. This remains a valid point until NATO eventually develops a complete set of standards and guidelines for used in federated mission networks, in which case the non-managed mode could also be an option.

Regardless of the implementation approach taken, the efficiency of a CLO scheme is proportional to the accuracy and validity of the cross-layer information used to make decisions.

2.6 Quality and validity of cross-layer datasets

We envision a future network-node architecture with a vertical layer as shown in Figure 2.6 where cross-layer information can be made available in a common database for any layer to utilize. In some cases there is also a need for horizontal exchange of network data for e.g., QoS routing, resource management and admission control. This data might very well be measured/produced on different layers than the requesting layer. In all situations where a network mechanism or protocol is using data outside of context of where the data was measured it is very important to associate the measurements with information of the quality and validity of the data.
Figure 2.5 Non-manager method (from [36]).

Figure 2.6 Manager method (from [36]).
For example, in research on routing protocols for MANETs it is well known that these protocols must have a high signaling frequency in order to keep up with state changes in the network. It is not so well understood how the accuracy and freshness of cross-layer information influences the routing decisions. Certainly, the information must be “new” in order to describe the current situation and not the past, but how long is it valid and how correct is the measurement? In [60] it is shown that the parameter used for QoS routing decisions in that paper often leads to an inaccurate picture of the traffic load that it was meant to describe. It is also shown that if the data is not accurate the result can be worse than if the QoS information is not used to make routing decisions at all. In [61] it is shown that the link state accuracy also have severe impact on the delivery success for the SMF multicast protocol.

A motivation for utilizing cross-layer information in a decision making process on another layer is to improve the accuracy of the mechanism’s knowledge of, e.g., a link state or channel condition. The exchange of efficient and precise information is the key for correct decision making. But, as shown above, such data must not be used uncritically. There is a need for a data model for cross-layer information that associate metadata with the measurement in a compact manner to describe at a minimum the expected accuracy of the measurement and the age/lifetime of the information. We are not aware of any work that has described such a model. Ideas for information representation and data validity used to synchronize data for a common operational picture in [85] can be one approach for a data model.

2.7 An example of ways to reduce congestion on a path using CLO

In the TCP/IP model (Figure 2.3) the layers operate independently. It is possible to call for slightly different functionality in the different layers by passing different parameters or accessing different Service Access Points (SAP) between the layers as data is making its way from the uppermost layer down the stack. However there is no negotiation between the layers and the different layers do what they are optimized to do when different situations occur.

Table 2.1 shows an example of how different layers individually can change their behavior in order to reduce congestion on a path between a source and destination.

This example motivates for the availability of a vertical layer in the node design like the model shown in Figure 2.6. In such design important parameters of the network state from the different layers can be made available in a database in the vertical plane for other layers to use. This will help informing the different layers that a problem has occurred. The vertical layer should also have some intelligence and be familiar with the mechanisms that are available at the different layers to solve the identified problem. This layer can then

1. make sure that the solutions chosen by the different layers do not oppose each other,
2. trigger the mechanisms that can cooperate to create the best solution, or
3. trigger the mechanisms with the consequences for the users that are acceptable for the operation.
<table>
<thead>
<tr>
<th>Layer</th>
<th>Solution</th>
<th>Consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Compress data</td>
<td>Increased delay, Reduced quality</td>
</tr>
<tr>
<td>Transport</td>
<td>Reduce TCP’s contention window</td>
<td>Increase transmission time</td>
</tr>
<tr>
<td>Routing (IP)</td>
<td>Find another path with available capacity</td>
<td>Delay, Influence other users,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Possibly longer, unstable path</td>
</tr>
<tr>
<td>Link (MAC)</td>
<td>Use more time-slots, Switch channel</td>
<td>Influence other users</td>
</tr>
<tr>
<td>Physical layer</td>
<td>Change modulation/coding</td>
<td>Power consumption, Transmission range, Noise level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for other users</td>
</tr>
<tr>
<td>Admission control&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Pre-empt another flow</td>
<td>Influences other users</td>
</tr>
</tbody>
</table>

<sup>a</sup> Admission control can be implemented on almost any layer

Table 2.1 How different layers can solve the problem of an overloaded path

In some cases a mechanism can be used at some layer that assumes a specific protocol at another layer (i.e., category 3 in Section 2.4). For example the Random Early Detected (RED) [33] queue type that can be used at the Data Link layer is designed to trigger a certain behavior in the TCP protocol in the Transport layer. The RED queue starts to drop some packets according to a statistical function when traffic are increasing on a path. This behavior makes TCP believe that the path is already congested and TCP reduces its transmission rate. If TCP is not used at the transport layer, the RED functionality only increases the packet loss in the network but does nothing to avoid congestion. In a situation like this a vertical layer with some knowledge of mechanisms on different layers can make sure that the assumptions are fulfilled.

This is just one of many possible examples to show that some interaction between layers can be beneficial, and in some cases necessary for good network performance.

2.8 Summary

We have presented benefits and drawbacks related to CLO in this section, and outlined some areas that can benefit from this kind of optimization. In the tactical domain wireless communication will be utilized, as will sensor networks to monitor the environment for incidents like enemy movement or the release of toxic substances. Further, to ensure interoperability between all entities acting in the land domain; C2 system, weapons systems, other participating nations in a NATO coalition force, etc., standards and interoperable middleware as identified by NATO must be used. Standards are important and should be used when possible to ease interoperability. The middleware supports the necessary machine-to-machine message-oriented exchange to build a decentralized and interoperable system. However, streaming data such as a video feed (e.g., a surveillance camera or a video conference) requires different technologies and approaches. Naturally, all information exchange must be adequately managed and secure. Thus, in the following chapters we pursue CLO for the following topics in further detail:
3 CLO from a layer perspective

In the previous chapter we established that despite the potential pitfalls of CLO, there are cases where the benefits can outweigh the drawbacks. In this chapter we explore some of these cases in further detail.

3.1 Network layer

In many cases it is possible to make measurements at a given layer (e.g., application) to deduce certain characteristics of the underlying network. CLO may reduce the overhead and increase the accuracy of such measurements by supplying information from the network layer directly, for example.

3.1.1 CLO and routing

While routing IP over fixed infrastructure is well understood, the tactical edge poses unique challenges due to its disconnected, intermittent, and low-bandwidth nature. To address this challenge, researchers and developers have often attempted to leverage link layer information (link quality, data rate, latency, etc.) to make better multi-hop routing decisions. The purpose can be manifold. Examples are; to improve path stability, to identify parallel paths for load balance or to provide a path to fulfill a certain QoS requirement. Utilizing cross-layer information can also reduce signaling load on the network.

Routing protocols in the early multi-hop narrow-band military Combat Net Radios (CNR), all leveraged cross-layer information in their routing decisions to some extent. [63] describe important design criteria for these radios. The multi-role radio (MRR) of the Norwegian Armed forces used measured noise as well as signal strength for the path calculation. MRR also spied on all traffic in the network and made this available for the routing protocol to deduce available routes in the network. Details of the protocol can be found in a RESTRICTED report [1]. In the ongoing work on the Narrow Band Waveform (NBWF) STANAG [76], similar cross-layer information will be utilized by their routing protocol. Also in this design two-hop neighbor information maintained
at the network (routing) layer will be made available for the medium access (MAC) protocol to assist in the time-slot assignment on this layer.

In military radios, much effort has been put into optimizing the performance for the protocol stack in each specific radio-type. Different manufacturers have made proprietary solutions for the best for their radio design. The opposite approach was taken in the protocol work for MANETs by the IETF. In their work it was assumed that the network layer protocols must work over any lower layer protocol, thus the routing protocol in the network layer cannot depend on receiving any information from lower layers. In much of this work the wide-band IEEE 802.11 protocol was used as an example of the lower layers in these networks. This radio type provides a much higher bit-rate than a military narrow-band radio, thus the designers can accept the cost of a higher signaling load. But as the time has evolved more and more of the research on MANET protocols involve cross-layer mechanisms (e.g., [96]).

Link layer notification (LLN) from the link layer of a broken link is the most commonly used mechanism. The Optimized Link State Routing Protocol (OLSR) [104] is one example of a protocol that can utilize this information if it is available. Link layer notification makes the routing protocol react quicker to topology changes in the network. In most cases LLN improves the network operation, however there is also a higher risk of route flapping when LLN is used. This can have unwanted consequence for higher layers e.g., TCP [88].

Almost all of the multitude of proposals for QoS routing assumes that some cross-layer information is made available to the routing protocols. [45] and [4] are two surveys that present some of the many proposals that exist. The work that we have done at FFI on Multi Topology (MT) routing [14] also assumes that some information about the lower layers is made available to the protocol for the protocol to calculate different paths in a network of networks environment for different QoS requirements. We are also doing a study on how sampling theory and compressive sensing can be utilized in order to use measured noise and signal strength to learn which direction to route traffic to avoid network congestion [59].

All the ongoing work on utilization of cross-layer information to identify stable routes or routes that fulfill certain QoS requirements are necessary in order to make the correct routing decision. More work is needed here to understand what are the important parameters to make decisions on. In the paper “The next 10 years of DOD wireless networking research” [16] one of the mentioned focus areas is to better learn how to characterize the network to be able to use collected statistics on different layers to better predict the future of the network state. As the research on utilization of cross-layer information in routing matures, this will improve network stability and utilization in homogeneous radio environments. However the lack of standard interfaces between the radio and router have led to interoperability issues in environments with a heterogeneous mix of radio systems. There is a need to reintroduce the notion that the routing protocols should be able to run on any lower layer technologies in the cases where many different radio networks are present in an operation. As a result, the desire to standardize a radio-
to-router interface for cross-layer information exchange has increased.

3.1.2 Radio to router interface

A standardized radio-to-router interfaces provides the means to separate radio and router functionality and to allow greater interoperability between systems. The MANET working group in the The Internet Engineering Task Force (IETF) is currently hosting this standardization effort. Three main protocol proposals provide the background for the effort: RFC 5578 “PPP over Ethernet (PPPoE) Extensions for Credit Flow and Link Metrics” [9], “R2CP Radio-Router Control Protocol” [24], and DLEP “Dynamic Link Exchange Protocol” [89]. In [18] the authors compare these protocol proposals. A radio to router interface consist of three different parts as shown in Figure 3.1: The radio-specific layer 2 information made available by the radio, the radio-to-router transport mechanism and the protocols in the router making use of the layer 2 information from the radio. In the implementation of the draft protocol by Fraunhofer [80], they have used an additional element in the information path that conform with the vertical plane in Figure 2.6, where a database for cross-layer information can be thought to be present.

Most modern military radios make use of layer 2 information in their routing decisions, thus both step 1 and 3 in Figure 3.1 are performed using an internal interface for step 2. However, how the internal routing protocol and admission protocol make use of the layer 2 information, and which information they use are proprietary design choices by different vendors. If a standard interface between radio and router were to be created, then much has been achieved: By standardizing the required and optional datasets that should be made available between layer 2 and layer 3 (and made available for other layers), radios from different vendors will provide similar QoS and traffic engineering behavior. This will ease network planning and management. When the layer 2 information is available on an external interface to a connected router, this can potentially improve the utilization of the network resources in an operation where equipment from different vendors are used. It can improve the ability for the Signal Corps to apply a common policy for network utilization and traffic engineering in an operation, to the network of radio networks. The radio-to-router interface will not only improve routing decisions, it can also provide valuable input to admission control mechanisms that must be in place in order for the network to operate in a predictable manner with finite transmission delays.

It is likely that a modern radio with a radio-to-router interface will keep providing an optimized proprietary internal protocol stack as is the case today. The external router can then either see the
radio network as a black box with the network properties made available by the radio-to-router interface, or the internal routing protocol can be switched off and the external routing protocol can take over full control. The result is a very flexible radio network that can be used in the manner that best suits the situation and the network of networks available in an operation.

In the current draft of the protocol only a few parameters are mandatory, however the intention is that the protocol can in due course specify more mandatory fields and optional fields. It is out of scope for the standard to specify how the routing protocol make use of the available parameters. The routing protocol’s metric (or set of metrics for an MT-routing protocol) decides how different link information can influence the path choice. Metrics can be created in many different ways. In [62] an overview of different metric types, are given.

The effort to standardize a radio-to-router interface is good news for small nations like Norway that want to acquire radio equipment from different vendors. The availability of such a protocol will enable external routers (e.g., Cisco) to make better routing decisions to improve network stability and utilization in an environment with several connected radio networks. Network connectivity can be improved and this is one of several important prerequisites to improve interoperability between cooperating units in an operation.

3.1.3 Resource management and admission control

Resource management and admission control are tightly coupled mechanisms. Furthermore it is not possible to provide differentiated QoS in a network with little capacity if there is no admission control. QoS mechanism such as priority queuing and prioritized access to the shared radio channel can be used to prioritize important traffic over less important traffic. However, these mechanism’s abilities to differentiate traffic of different priority levels are close to nonexistent when the network is heavily congested. Therefore, for such QoS mechanisms to work, admission control is mandatory. It is not possible to do admission control that adapt to the available capacity in the network if there is no network resource management. Cross-layer information from the network layer and Link layer e.g., provided by QoS routing like MT-routing and the radio-to-router protocol mentioned above are data that can deliver information to support resource management and admission control mechanisms. It can also be advantageous if tools on these layers can accept instructions from a resource management entity.

Having the example in Table 2.1 in mind. This example describe different approaches, on separate layers, to reduce congestion on a network path. Several of these solutions need cross-layer information in order to be aware of the congestion or in order to reduce own signaling to learn of the congestion. It is also interesting to support a cross-layer service for resource management that can choose which mechanisms to perform in order to minimize the impact of the solution on the ongoing operation.

Performing efficient resource management and admission control in the highly agile environment of multi-hop radio networks are complicated tasks. Some of the specific challenges related to radio
networks are:

1. The available maximum capacity in the network can change in an instant depending on the topology of the terrain and the inferred radio channel quality and depending on the availability (coverage) of different radio technologies in the area of operation. In a fixed network this capacity is constant.

2. In a mobile ad hoc network all nodes in the network are both entry nodes, forwarding nodes and exit nodes thus all nodes need to make local admission control decisions. In a fixed network one normally assumes that new traffic arrives and exits at a set of entry and exit nodes. Many of the nodes in the network are only forwarding elements.

3. In mobile networks there are frequent topology changes in the network, after such events re-routing of a batch of admitted flows might need to be placed on a new path that is already filled up with other admitted flows. Link failures that lead to similar situations in fixed networks are rare.

4. Since the capacity of a path can be very low, admission of a new flow can add significant load on the path. It is difficult to predict the full consequences of the admission in such situation. In fixed networks with much capacity admission of one flow is barley noticeable.

5. Admission control signaling uses the same channel as the data traffic. In situations when network congestion occurs, the risk of losing the important signaling messages needed to repair the network state (reduce the traffic) is high.

[46] presents a survey of admission control schemes mainly for networks based on IEEE 802.11 radios. Most of these schemes are tightly coupled with the routing protocol and they compete on how well the schemes are able to measure and estimate the local load on the radio channel. Cross-layer information form the radio channel are used, in some cases also with additional measurements performed on higher layers. Few of these schemes are meant to work on other lower layer technologies or in a heterogeneous radio network environment.

The Pre-Congestion Notification (PCN) Architecture [28] for admission control and flow termination, is an effort by IETF. [64] presents a study of this architecture’s applicability for MANETs. That work states the importance of being independent of lower layer protocols but still assumes that a measure of local channel load is available from lower layers. One of the required parameters in the current draft of the radio-to-router protocol would suite this need. In addition to using local cross-layer information this paper also proposes to do some end-to-end probing on the current layer to improve the picture of the network. The combination of cross-layer information and some additional measurements performed on local layer is probably the best solution for most mechanisms that can benefit from cross-layer information. The local in-layer measurements can also be made available for other layers to use if a common database for cross-layer information is available in a vertical layer as in Figure 2.6.

In the ongoing FFI project 1249 we are working on a routing protocol that is intended to glue different military radio networks together to form one common network [47]. The protocol
is planned to provide both QoS routing and admission control. In this protocol we are also planning to use cross-layer information as well as own end-to-end measurements. In this work we experienced that the use of cross-layer information in mobile networks is still not a mature topic. It is not well understood what cross-layer information is useful and can add value to the routing and admission control decision process. The value and accuracy of the cross-layer information will vary for different scenarios (e.g., terrains, mobility levels, radio technology). One of the challenges that has not received much attention is the accuracy and validity of cross-layer information, as discussed in Section 2.6.

Assuming a common database where different layers can read and write useful network state information is available, then it is important to make sure that the data provided by the different layers describes the same network state (snapshot). If, e.g., a MT-routing protocol has stored information of available QoS paths to a destination and the transport layer protocol has measured and stored the packet loss ratio on the path to the same destination, it is important that the time stamp for these data sets are inspected to verify a likelihood that the packet loss information is measured on the route that the routing protocols currently uses.

### 3.2 Transport layer

The User Datagram Protocol (UDP) [82] and the Transmission Control Protocol (TCP) [84] are the most widely used transport protocols in an IP network. The UDP protocol is a simple datagram protocol with very little functionality, whereas the TCP protocol is a complex end-to-end protocol that attempts to:

- Provide ordered data transfer,
- retransmission of lost packets,
- error-free data transfer,
- flow control, and
- congestion control.

TCP is the focus of the discussion in this section.

TCP has been, and continues to be, an essential protocol for Internet communication. Without its rate control, traffic congestion would have rendered the Internet useless. However, TCP makes several assumptions about the network. It assumes that network congestion, and not transmission errors, causes packet loss. It also assumes that the Round Trip Time (RTT) is relatively constant (little jitter) and that rerouting happens very quickly. None of these assumptions are easily satisfied in military radio networks, which results in TCP having substantial problems when employed in such environment. An FFI report by E. Larsen [27] describes challenges and possible solutions for TCP in MANETs. This section summarizes the most important findings from that report.

One does not normally consider TCP a cross-layer solution. However, the TCP congestion control algorithm breaks layer transparency as it makes an assumption about the behavior of the lower
layers. The congestion control algorithm monitors packet loss and assumes that packet losses indicate congestion [3]. However, in wireless networks, packet losses happen both in bursts and regularly as the channel quality varies, as well as when the link is congested. Consequently, when a lower layer does not behave according to TCP’s assumptions, the performance drops. Thus, one may argue that this satisfies the requirements of a category 3 cross-layer solution.

3.2.1 Cross-layer enhancements to TCP

Some of the well known techniques to improve TCP’s performance in radio networks are the ICMP/TCP interface, the RED queue and ECN. These are briefly described below.

Internet Control Message Protocol (ICMP) [83] is used by network devices to send diagnostic, control or error messages. ICMP has a number of subtype messages, and “Source Quench” is one of these. A router or host might send an ICMP Source Quench message when it receives datagrams at a rate that is too fast to be processed. TCP responds to a source quench message by triggering a slow start, as if a retransmission timeout had occurred. The implementation of ICMP source quench requires a special interface between the IP/ICMP layer and the TCP layer. Such an interface can be classified as a category 1 approach.

Source quench has rarely been used in the Internet and it was deprecated in 2012 by RFC6633 [40] due to the development of Explicit Congestion Notification (ECN). Source Quench is, however, still used by some tactical radios to inform the sender (i.e., a computer connected to the radio) that it should slow down its data transfer speed or wait for a certain amount of time before attempting to send more data. This type of functionality is currently also part of the required functionality of the radio-to-router cross-layer protocol described in section 3.1.2.

Random Early Detection (RED) [33] is an active queue management algorithm, as well as a congestion avoidance algorithm. In the traditional tail drop algorithm, a router buffers as many packets as it can, and simply drops the ones it cannot buffer. Tail drop distributes buffer space unfairly among traffic flows. Tail drop can also lead to TCP global synchronization as all TCP connections “hold back” simultaneously, and then step forward simultaneously. RED monitors the average queue size and drops packets based on statistical probabilities. If the buffer is almost empty, all incoming packets are accepted. As the queue grows, the probability for dropping an incoming packet grows too. RED is more fair than tail drop, in the sense that it is not biased against bursty traffic that uses only a small portion of the bandwidth. Early detection helps avoid TCP global synchronization.

ECN [86] is an extension to TCP that allows end-to-end notification of network congestion without dropping packets. It can be seen as an improvement of RED where packet drops are avoided, but it requires support by the TCP sender implementation. The extension is an optional feature that is only used when both endpoints support it and are willing to use it, and it depends on underlying network support to be effective. ECN-aware router may set a mark in the IP header instead of dropping a packet in order to signal impending congestion. The receiver of the packet
echoes the congestion indication in the TCP ECN field to the sender, which must react as though a packet was dropped. ECN separates its congestion control algorithm from error detection and retransmission, which is an advantage for applications with strict latency requirements. However, ECN is not supported by all routers and end systems, and often it is disabled by default\(^3\). As ECN is not always available on the network layer, its behavior is not known in advance. Some routers drop packets with the ECN field set, and some reset the field.

As ECN is based on upward information flow through the creation of a new interface between the network and transport layer, we classify it as a category 1 cross-layer solution.

An enhancement to ECN is possible by letting the application layer access the ECN bits. This can for example allow a video streaming application to alter some parameters in order to ensure QoS during congestion. For further considerations on cross-layer and video streaming, see the case study in Section 5.2.

Many proposals for TCP enhancements in MANETs exists. Some attempt to cope with TCP’s interpretation of packet loss as network congestion like ECN, others try to cope with long round-trip delays over, e.g., satellite links and long rerouting times in MANETs. Many of the proposals use assumed or explicit cross-layer information in their solutions. In [27], Larsen presents a survey of these protocols and classifies them according to how much modification the proposals impose on the TCP protocol:

1. No changes needed.
2. Changes needed only to one or both of the end nodes.
3. Changes to the relay nodes as well as the endpoints.

Notably, solutions that require ECN signaling from relay nodes are classified as requiring changes to relay nodes, since the ECN mechanism is not universally available.

### 3.2.2 Performance-enhancing proxy

Performance-enhancing proxies (PEPs) (e.g., [13]) are often mentioned for TCP enhancements. PEPs are installed on the edges or on all nodes in a network segment that need improved TCP performance. Many different PEPs exist, some split the TCP connection, where the proxy terminates the TCP flow from the source and establish a new TCP flow with the destination. This has a consequence for end-to-end flow control, but allows TCP to be tuned differently for different network types. This solution is often used to handle the long RTT on satellite connections. Similarly a TCP solution that requires special support from relay nodes in a network can be used only in the network segment where this behavior is needed and does not require the whole network to support the new functionality.

Other PEPs can listen to the TCP flow and modify, discard or introduce control packets e.g., ACK

\(^3\)Standard installations of current Linux and Windows end user systems were checked (a Ubuntu 12.04 system and a Windows 7 system), and both had the ECN field disabled by default.
in the connection in order to avoid rate reduction, termination etc. that could otherwise be the
erroneous consequence of typical network behavior.

3.3 Application layer (middleware)

Logically, middleware is just below the application layer, but in practice it is realized at the
application level. When leveraging middleware, the idea is to build the application on it, rather
than bypassing it to access the underlying network directly. This means that when discussing
middleware below, the discussion is to be seen as relating to the application layer. This section
presents middleware with a special focus on Web services, and discusses some experimental
approaches to CLO.

Even so, the technology is not meant to be exposed to users directly. It is not for implementing
User-Facing Capabilities, but rather be building blocks for such capabilities. That is, the technol-
ogy can be used for realizing aspects of the Technical Services, in particular the Information &
Integration Services. The NATO C3 Classification Taxonomy defines these terms. See Appendix B
for further details.

3.3.1 Introduction to middleware

Middleware is a term that in different contexts can mean quite different things. The general
understanding is that middleware is a software layer between the application and some underlying
set of heterogeneous resources. By leveraging middleware, the application can focus on using
the interfaces provided by the middleware and utilize underlying resources regardless of their
actual realization. Along these lines solutions like hibernate\(^4\) can be seen as a middleware for
accessing data stores (SQL databases, no-SQL databases, and so on). A solution like phonegap\(^5\)
can be seen as a middleware for developing platform independent software for mobile devices,
enabling the application to function across different hardware devices and different operating
systems. For building decentralized systems, middleware often aims to overcome differences in
networks, operating systems, and/or programming languages employed. Web services, currently
one of the most widely employed middleware approaches for building loosely coupled service-
oriented systems, address all these aspects for machine-to-machine communication. This means
that services built using Web services technology can potentially be leveraged by a larger audience
and a wider array of systems than competing technologies that focus less on interoperability and
universal deployment (see e.g., [68] for a discussion of other middleware solutions).

3.3.2 Web services

Web services as defined by the World Wide Web consortium [105]:

A Web service is a software system designed to support interoperable machine-

\(^4\)http://hibernate.org/
\(^5\)http://phonegap.com/
to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Web services technology is becoming increasingly popular for implementing loosely coupled civil systems. Interoperability is the main concern, and thus Web services are based on standards. NATO has identified Web services standards as a key enabler for interoperability between the different military systems of the various NATO nations.

3.3.3 QoS and Web services

Web services technology originated for use in civil enterprise networks and across the Internet. This means that the technology works well in fixed infrastructure networks with high throughput. In military networks we want the implementation flexibility and interoperability that Web services can provide. However, there are several challenges that need to be addressed before the technology can be used in such networks. Previous research (see e.g., [15, 21, 42, 50, 52, 53, 67, 71]) has shown that it is feasible to use Web services in tactical networks provided certain techniques that can be employed to optimize Web services are in place: Content filtering to reduce application overhead, data compression to address the Web services overhead, and using specialized transport protocols that can overcome the limitations and disruptive nature of military networks. Given these adaptations are employed, the technology should be deployable in military networks. However, it still leaves something to be desired regarding Quality-of-Service (QoS), as current standards only address certain aspects like reliability and transaction management. There are no standards related to prioritizing and preempting requests related to Web services. See [51] for an in-depth discussion of QoS related to Web services and standard vs proprietary solutions. That report advocates the use of CLO for bringing improved QoS to systems leveraging Web services, however such optimizations are not founded on Web services standards. Thus, it is important that, for the sake of interoperability, any cross-layer approach added to Web services technology should retain compatibility with COTS clients and services.
The Web services lifecycle is illustrated in Figure 3.2. It addresses the different stages encountered when developing, deploying, and employing Web services. Let us explore the lifecycle in further detail, while paying special attention to potential extensions necessary to support QoS:

1. The **Advertisement** is the process in which a service provider makes a service available (publishes it using a service discovery mechanism). Necessary extensions to support QoS:
   - Provide information about the different quality aspects and parameters supported by the service.
   - Make this information available in addition to the standard service description.
   - Quality handling requirements include a common understanding/interpretation of quality, need for a standardized way of describing supported quality – and how to use the quality information, and the service discovery mechanism must be able to handle additional metadata.
   - CLO approaches can potentially make the advertisement process more efficient from a communication viewpoint, as well as providing more refined (that is, higher quality, more accurate) measurements for the quality information.

2. **Discovery** is the process in which a service consumer queries the discovery mechanism, and finds suitable services. Necessary extensions to support QoS from a Web services viewpoint:
   - Return quality information along with the service description (simple approach), or
   - Allow the consumer to use quality parameters in their search. In this latter case the discovery mechanism must understand the quality parameters, an approach which enables more advanced matchmaking than the former.
   - Quality handling requirements (in addition to the above) involves the need for a way of describing the quality needs of the client and/or the quality offered by a service.
   - Network knowledge is also desirable at this point, as the network will affect the QoS experienced by the service consumer.
   - As service discovery is included as a part of Service Management & Control (SMC) in the C3 Taxonomy (see Appendix B) we cover the CLO approaches for service discovery in Section 4.2.

3. **Selection** is the process of selecting which services to invoke, based on the Discovery results. Necessary extensions to support QoS:
   - Enough information needed to make an informed choice.
   - Need to choose which quality to ask for in addition to which service to invoke.
   - Quality handling requirements (in addition to the above) include quality aware matchmaking.
   - This step is a pure computational step based on information from the previous steps. Thus, as no communication is involved this step cannot be directly optimized by a CLO approach. It can, however, become more refined and make better decisions if the
input (e.g., quality information which may stem from CLO approaches) is of a higher quality than before.

4. **Composition** is fulfilling a requirement by combining independent services to provide new, value-added functionality. Necessary extensions to support QoS:
   - Handling combinations of quality parameters.
   - Some parameters require run-time calculations.
   - Quality handling requirements (in addition to the above) includes the need for a means to express the quality of the combined service.
   - Again, no direct benefit from CLO, but an indirect benefit may occur through the increased accuracy of measurements provided to the composition process.

5. **Invocation** is the process in which a consumer calls a service, by sending a request message, and receiving a response. Necessary extensions to support QoS:
   - Messages must be sent using required network QoS.
   - Standards can provide coarse QoS mechanisms like reliability and transaction management.
   - Quality handling requirements (in addition to the above) involves network knowledge.
   - This step can benefit greatly from CLO approaches including, but not limited to:
     - Middleware quality needs and suggestions should be signaled to lower levels.
     - Application priority to network priority mapping.
     - Adaption to current resource availability could be a potential gain.

As standards covering most of these aspects are lacking, we glance at experimental solutions in the following section.

### 3.3.4 CLO approaches for Web services

There are currently no finished standards for describing and handling the quality aspects of Web services, but work is ongoing within OASIS. Their Web Service Quality Definition Language (WS-QDL) [79] is an XML based language for describing QoS attributes of services, but this effort is geared towards Internet and business use of Web services, and thus has a focus on more high level QoS parameters such as cost and billing. To support QoS for Web services in MANETs a more flexible and light-weight solution for QoS handling is presumably required. Existing frameworks that support QoS for Web services are largely limited to experimental solutions.

Tian et al. [99] have built an infrastructure for handling prioritization of Web service requests, with a focus on mapping application level QoS descriptions to priorities on the network layer. This includes traversal of two different network technologies, namely a wired network using DiffServ (see Section 2.3.2), and UTMS, with translation of network priority between these technologies. This work focuses on the signaling of priorities during transmission.

In [93], Sliwa and Duda discuss a framework for adaptive Web services supported by QoS in the network. They use SIP to signal an application’s QoS requirements, and map to DiffServ to
handle network level QoS. Further, they discuss how a mediation service can be used to adapt Web services traffic to better utilize the available resources and meet the required QoS demands, i.e. the QoS profile given by the client.

Wang et al. [107] create a QoS-aware selection model for semantic Web services. They create a simple QoS ontology for expressing qualitative metrics and values. These metrics and values are arranged in a matrix, which is used for matching provided QoS with the client’s QoS profile. The QoS expression and matching is limited to numerically quantifiable QoS aspects, making this solution best suited for exposing network parameters such as delay, packet loss, etc. to an application.

Yu et al. [111] have investigated issues regarding service selection with multiple QoS constraints and proposed several algorithms. Their approach can be leveraged when you have multiple available services to choose from. One could anticipate combining CLO with this approach to obtain fine grained metric information to feed into the algorithms.

Achieving QoS support for Web services is complex and features many different approaches. In project 1176 we investigated some of the necessary building blocks that are needed for cross-layer end-to-end QoS for Web services.

In order for a system to be agile, finding services and determining how to use them must be done in a manner that requires as little manual configuration as possible. One issue with Web services standards is the inability to fully describe what the service is capable of doing. Semantic Web services (SWS) [70] extend the service descriptions of current Web services technology with rich, explicit semantics to improve service discovery, selection and invocation significantly. We have performed preliminary experiments with SWS in conjunction with discovery of Web services and orchestration. For further information, see [43, 44].

From a security viewpoint, role-based access control (RBAC) is an approach to restricting system access to authorized users. In a QoS setting the concept of a role can be extended to include not only traditional security aspects regarding authentication and authorization, but also to signal the priority of messages traversing the network according to the role issuing said messages. We have investigated how role based admission control can be used in conjunction with QoS to ensure higher success rates for the most important Web services traffic. Cross-layer signaling of QoS data (i.e., networking information exposed from a router) was used to ensure efficient admission control and corresponding resource use. A prototype was implemented and tested as part of a student task at FFI, see [57]. Continuing this work, a student group under supervision by FFI implemented RBAC in an open source enterprise service bus (ESB)\(^6\). In this work the security standard SAML was employed to represent the roles, and cross-layer signaling (i.e., setting the appropriate DiffServ class in the network) was used to prioritize each message involved in the

\(^6\)An ESB is a middleware product which implements an infrastructure upon which a SOA can be built. It is not SOA in itself, but provides important functionality such as message routing, and translation between message formats and different transport protocols. For interoperability reasons ESBs often leverage a subset of the different available Web services standards. For more on ESBs, see [66].
communication. For further details, see Appendix A in [66]. The drawbacks of both these student works is that they by now are somewhat outdated: There is only IPv4 support, and they are not fully aligned with the current direction NATO is taking regarding access control, which is geared towards claims-based access control, which offers more control than RBAC. The concepts are proven, however, and could be pursued further as a part of a holistic CLO approach for further research. In that case the interface discussed in Appendix A could be leveraged to obtain network information for use by the middleware.

The above works all aim for a high degree of automation, where the goal is to involve the user as little as possible in making decisions. From an agility point of view such an approach is beneficial, because decisions can be made by software immediately and according to policy. It is, however, hard to achieve in practice. The policies need to be elaborate and cover all cases, they must be machine-readable, and more importantly, they must be trusted to perform exactly as expected. Given these challenges, a simpler approach is to obtain necessary information through CLO, but leave the ultimate decisions to the user. This latter approach has been taken by Fraunhofer FKIE in their SOA interim solution [7], where the cross-layer aspect is limited to expose network information (basically link load and availability) to the user. This allows the user to become aware of changes, limitations, and outages in the network, and adjust expectations and use of the C2 system correspondingly.

As we have seen above, current CLO approaches for optimizing Web services mostly follow the first of the four approaches discussed in Section 2, and can roughly be divided in three categories: Those that attempt to impose their priority and quality needs on lower layers by using a downward communication, those that glean information from lower layers using upward communication in an attempt to make better choices on the application layer, and finally those that are bidirectional in that they attempt to do both.

4 Cross-layer services identified by NATO

In the NATO C3 Classification Taxonomy (see Appendix B), both Information Assurance (IA) and Service Management & Control (SMC) are vertical (i.e. cross-layer) groups of services. It should be noted that in the taxonomy, a service is merely a set of functionality. Also, the taxonomy describes the functionality on a conceptual level, leaving the implementation details to ongoing and future work. This is in contrast to the previously discussed middleware, where a service is synonymous with a Web service.

---

7You can use claims to implement RBAC. Roles are claims, but claims can contain more information than just role membership. So, claims-based security is a more fine-grained mechanism than RBAC. For an introduction to claims-based identity and access control, see [72].
4.1 Information assurance (IA)

The Information Assurance (IA) group of services shall help ensure confidentiality, integrity, availability, non-repudiation and authentication of information. The NATO C3 Classification Taxonomy broadly divides IA into two areas: Cyber Defence and Cyber Security. Cyber Defence is classic and static information security often based on cryptographic techniques, while Cyber Security aids the dynamical handling of concrete incidents and attacks.

Many aspects of Cyber Security can be handled in a single layer, but activities are in principle cross-layer. Whether an operation is defensive or offensive, the ability to analyze or control all aspects of the data is a valuable asset. A typical example is an intrusion detection system where traffic is analyzed from many different layers and perspectives.

In Cyber Defence, security mechanisms are usually applied at one layer at a time. This means that a mechanism like encryption is often applied independently at different layers. This gives added security in depth, since an attacker will have to circumvent several mechanisms to attack the system. On the negative side there is some waste of resources. The main concern in our setting should be the low transmission rate in tactical radio systems. Mechanisms like cryptographic checksums and digital signatures increase message lengths. If they are applied at several layers, message expansion can be problematic, especially for short messages. This problem can sometimes be reduced by a cross-layer design.

Another aspect of IA in connection with cross-layer design is that some security mechanisms, mainly encryption, can prevent the use of otherwise reasonable cross-layer optimizations by making necessary information inaccessible between layers.

4.2 Service management & control (SMC)

Service management & control (SMC) encompasses such functionality as the planning, coordination and execution of the work and management of the services involved. The overall goal is to ensure its effective support of and contribution to the continued uptime and functionality availability of the overall system. An important part of SMC is service discovery (see [12]).

Inherently, service discovery is a middleware service, that is, it originates at the application layer. It has been defined by the World Wide Web Consortium as [106]:

> Discovery is the act of locating a machine-processable description of a Web service related resource that may have been previously unknown and that meets certain functional criteria.

NATO has pointed to the OASIS standard Universal Description, Discovery, and Integration (UDDI) for service discovery. This centralized solution offers discovery, but has potential issues in tactical networks as discussed in [50]. At present other approaches are being considered, such as in the currently active NATO STO/IST-118 working group “SOA recommendations for
disadvantaged grids in the tactical domain” [52].

Cross-layering in a service discovery context means all optimizations done by taking advantage of information found on lower layers – such as examining the routing table or measuring signal quality. So, the term cross-layer service discovery may for instance refer to a service discovery solution that utilizes the routing process to disseminate service discovery messages. Routing-layer support was first introduced by Koodli and Perkins [58]. Now, several different proposals exist both for reactively routed and proactively routed MANETs. It is evident (see e.g., [30]) that in environments with limited resources there are gains in the service discovery process when involving CLO approaches. See [30, 54] for more on approaches to cross-layer service discovery.

5 Case studies

The two case studies in this section span multiple of the layers discussed above.

5.1 Wireless Sensor Networks (WSNs)

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to monitor physical or environmental conditions [75].

5.1.1 Introduction to WSNs

As mentioned previously in this report, there are many arguments against the use of cross-layering. One of the reasons for the skepticism against cross-layering is that such optimizations, although they may appear simple on paper, can be very difficult to implement. On a PC for example, passing information between protocol layers can be very a difficult task considering that the different communication layers are constituted by many software components implemented in different programming languages. Efforts to make these software components interact in new ways can quickly lead to spaghetti-like code that is hard to maintain. This and other issues with cross-layering are described in detail in [55].

It is important to realize that cross-layering does not imply an integration and joint optimization of all layers with a complete elimination of the layered approach. Still, there exist some standalone sensor systems and simple networked sensors that employ such single-layered architectures. In the early stage of WSNs, specialized, one-of-a-kind and non-interworking systems were the common rule. From a mere performance viewpoint, tying together all networking components into one such super-layer can have great benefits. A one-layer solution also reduces the code size, which is a major limiting factor in embedded systems. Modern WSNs do, however, seldom rely on one single super-layer, but instead use a layered approach just as in a networked PC. At the same time, there are still some important differences between WSNs and traditional PCs when it comes to protocol layering.

Firstly, whereas PCs and other network devices are multi-purpose by design and can serve several
applications simultaneously, WSN nodes are often built for one single application purpose (e.g., military surveillance or environmental monitoring). This means that it is difficult, and often unnecessary, to generalize lower protocol layers.

Secondly, WSN nodes also differ from other computer platforms in being relatively simple devices, basically built around a low-power microcontroller and a radio circuit. Since it is a single-purpose design, the entire operating system and all protocol layers are often implemented using the same programming language and they can even belong to the same code base. This is the case for both TinyOS [100] and Contiki [25], the two most popular operating systems for WSNs. In other words, the protocol layers are loosely bounded and passing information between them is therefore reduced to simple function calls. This makes it relatively easy to implement cross-layer interactions without the danger of introduce bugs or destroying the code structure.

Lastly, since a WSN node is often battery powered, it has very limited resources in terms of computational power and memory. Consequently, even relatively simple cross-layer optimizations can give potentially huge performance gains in terms of system lifetime and performance. Numerous such optimizations are therefore proposed in the literature. To sum up, while there are good reasons to maintain a general skepticism against cross-layer optimizations, there are many reasons to investigate such techniques for WSNs.

5.1.2 Examples of CLO for WSNs

In a WSN node, there are several application layer functions that can benefit from exploiting lower-layer functionality. As an example, there are in some circumstances a need to coordinate data amongst several WSN nodes. It is difficult to perform such coordination in an efficient way using only the mechanisms available on the application layer. A more efficient approach would be to create an interface between the application layer and the network layer (i.e., a category 1 solution). Via such an interface one can exploit that the routing protocol has the capability to perform efficient distribution. An example of this kind of cross-layer interaction between the application layer and the network layer is the work by Flathagen et al. [31]. The scheme presented in this work includes distribution of node-to-clusterhead memberships (information needed by the application layer for data aggregation) by using the distribution mechanism in the routing protocol.

An example of a more extensive use of cross-layering interacting with the application layer is the combined localization and routing approach in [32]. In this work, the localization method at the application layer uses signal strength information from the physical layer and link information from the network layer. This information is fed to an application layer protocol, which combines the information and feeds it back to the routing protocol. The routing protocol then forwards the information to the sink for further processing. In other words, this scheme uses back-and forth cross-layer interaction involving three protocol layers.

Most prevalent WSN routing protocols employ cross-layer mechanisms to a greater or lesser degree to increase the efficiency and to improve the energy and bandwidth utilization. Routing
protocols typically use downward or upward interfaces to adjacent layers for information sharing. For example, a routing protocol can perform topology control by transmitting notifications to the physical layer about a preferred radio transmit power as in the work by Chipara et al [19].

Another way to do cross-layer design is to merge adjacent layers. For WSNs, there are several proposed schemes that involves a complete integration of the MAC protocol and the routing protocol. One example of such integration is the work in [90].

A routing protocol can also rely on hints from lower layers, such as information about the residual energy on the node or information about signal strength or link quality from the physical layer. An example of a routing protocol doing the latter is MultihopLQI [101]. MultihopLQI relies on the Link Quality Indication (LQI) from the physical layer. LQI gives information about the quality of the decoding of an incoming packet, which can be used as a part of the routing metric. A limitation with MultihopLQI is that the use of LQI is limited to one particular radio circuit (i.e., Chipcon CC2420). Such a hard dependency between the protocol layers is a typical disadvantage for many cross-layer optimizations.

LOAD [56] is another protocol that takes advantage of link information from the physical layer in addition to the hop distance in the routing decision. However, while MultihopLQI is tailored for one particular radio chip (hard dependency), LOAD can be used on top of all radio chips that can provide some kind of simple link quality measurements (soft dependency). The cross-layer interaction here is therefore simpler and more flexible. The basic mechanism is as follows: If the quality value measured in LOAD is below a certain threshold value, the link is considered weak. The route cost then becomes a combination of the number of hops and the number of weak links. If the radio chip does not support link measurements or a cross-layer interface is unavailable, the protocol operation is identical to minimum hop count routing as in AODV.

The idea behind LOAD is to exploit cross-layer interactions between the routing layer and the physical layer without being bound to one particular radio circuit. The four-bit wireless link estimation module [34], which is used by CTP [39], follows the idea of such a general link estimation method a step further. It combines information from the network, link, and physical layers when estimating the link quality. The scheme provides simple interfaces between these layers. Despite the fact that the method definitively is cross-layer, which could indicate that the method is highly proprietary, the interfaces provided by the method enable a generalized link estimation method that can be applied for a wide range of routing protocols and physical layers.

5.1.3 The future of CLO in WSNs

As briefly discussed in the introduction of this chapter, WSNs have generally progressed from being single-layered and highly optimized proprietary systems towards having a more modular and layered protocol design. Standards have now been designed for several of the protocol
layers. Many of these protocol standards are now seen as de-facto WSN protocols and are used regardless of the WSN application task. One of the first protocol standards that emerged was the IEEE 802.15.4 specification for the physical, media access, and data link layers. More recently, the IETF has standardized IPv6 by the means of 6LoWPAN [74] and the RPL protocol [109] for the routing layer. The IETF is also working on standards for the application layer through the Constrained Application Protocol (CoAP) initiative [92].

With a growing range of standardized protocols, there will also be an increasing need to establish standardized ways of communicating between the protocol layers. Culler et al. claim in the work [22] that “...the primary factor currently limiting progress in sensornets is not a specific technical challenge but instead is the lack of an overall sensor network architecture...” The authors propose a layered architecture with a Sensornet Protocol (SP) placed between the network layer and the link layer. One aspect of SP is to provide cross-layer services. SP can for example buffer packets while a power-aware MAC turns the radio off to save energy. Protocol layers above SP can schedule such radio activity using cross-layer interactions through the architecture.

There are also other proposed designs that enable the use of cross-layering without violating the architecture and without creating dependencies that hinders future system extensions. TinyCubus [69] and Chameleon [26] are two such examples. Although such generic architectures seem appealing, they are certainly not free of costs, due to their added complexity. The memory footprint and the extra processing required for these architectures can sacrifice performance on energy and memory constrained WSN nodes. The antithesis of such generic architectures are found in [87] advocating that in the WSN research community there is “...an unjustified desire for generic solutions...”. The authors further argue that a WSN system design should be highly bottom-up and application specific. Ultimately this means that an extremely optimized and cross-layered WSN, or even a single-layer WSN, can be justified if this is the most appropriate solution for the application task.

In short, there are two “schools” in WSN architecture design, whereas the “old school” argues for an application-driven and bottom-up design with only a vague definition of protocol layers, and the “new school” argues for strict layering and standardized protocols as seen in PC-networking. Considering the prevalence of IP-enabled sensor networks, including the development of IPv6, the “Internet of Things” and Smart Grid, the “new school” will unquestionably be dominant in the future. Taken into account that a generic and strict layered design is seldom the best suited for a given WSN application, it will always be a great need for application-driven cross-layer optimizations to extend the layered design according to whatever the application requires. That is to say, both the desire for a general standardized architecture and the desire for effective cross-layer optimizations can be achieved by creating a cross-layer architecture based on a vertical plane functioning as a manager across standardized layers, as illustrated in Figure 2.6.
5.2 Video streaming

Video streaming applications are applications where a video stream, consisting of a sequence of picture frames, are transported over a network to a receiver which presents the video continuously as it arrives. Video streaming applications, which include both unidirectional playback of video as well as bidirectional video conferences, are particularly challenging in the tactical environment, due to their bandwidth and timeliness requirements [17]. A thorough discussion of the challenges application of video streaming applications in the tactical domain, existing solutions, and areas which need more research, can be found in [38].

Considering bandwidth, as picture frames are very bandwidth-demanding compared to for example text messages and documents, video streaming applications pose significant challenges for the tactical domain. Even when advanced video compression techniques have been applied, video streams require in the order of hundreds to thousands of kilobits per second (Kbps). While this may be a modest requirement in today’s civil broadband networks, it can be a very hard requirement, even impossible, to satisfy on a tactical radio.

Considering timeliness, users of video streaming applications expect the video presentation to be continuous and without jitter, and near “real time”. How near real time depends on the application, but for interactive applications, such as a video conference, each frame must be delivered less than 150 ms after it has been recorded for the user to be satisfied with the QoS experienced. Strict timeliness requirements pose a challenge both in tactical and non-tactical environments, as it requires a different behavior from data transport protocols than other types of applications, such as message delivery and file transfer. Indeed, it is particularly challenging in any environment that requires the use of wireless communication.

As network conditions are poor and unreliable in the tactical environment, optimization of video streaming applications is particularly important in order to handle the requirements mentioned above. Below, we look at different types of adaptation mechanisms that can optimize video streaming applications, at which layers they are typically applied, and how coordinating mechanisms on different layers improve the effect of adaptation. These adaptation mechanisms, and several proposals for how to use them for CLO, are described in more detail in [91].

One must accept that in some situations in the tactical environment, it is not possible to communicate video streams. In these cases, one must ensure that no video streaming application is allowed to send data into the network, as video data may flood the network and render it useless for all its users. However, we focus on situations where video streaming is possible, and how running streams can be optimized using cross-layer adaptation mechanisms.

---

9In this report we do not consider applications where an entire video, or a chunk of a video, is downloaded in its entirety, before it is presented, as this type of application is rather similar to a file transfer, which is a different kind of service and requires different approaches to solving [38].
5.2.1 Video source adaptation

Video source adaptation is an application layer mechanism where the bit-rate and quality of a video stream is adapted by adjusting the parameters of the video encoder producing the stream. The main reasons for performing video source adaptation are

1. adjusting the stream bit-rate to the available bandwidth to avoid bursty losses due to stream data being dropped in the network, and
2. optimizing the quality of the video within the available bandwidth to give the user the best possible perceived QoS.

The bit-rate and the quality of a video can be adapted using different techniques, as for example by varying the number of frames per second, or by varying the quality of each frame. As an example, [8] discusses how video streams can be adapted to tactical devices by transcoding between different bit-rates and quality levels, and presents a prototype of such a tactical device. The work presented in [8] is focused on broadcast applications, and not interactive applications.

Application layer video source adaptation should always be applied in the tactical environment, in order to scale the video up or down to a bit-rate that is suitable to communicate through the network. Due to dependencies between encoded frames, the stream will most likely be irreparably damaged if random packets are dropped by lower layers. Thus, it is not possible to scale down the bit-rate of a video stream from thousands of Kbps to hundreds of Kbps without involving the video encoder. However, the effect of video source adaptation can be optimized if combined with mechanisms on the lower layers. For example, [102] propose to perform video source adaptation based on rate adjustment requests passed from the link layer to the application, communicating the bit-rate which can be successfully transmitted at the moment. We classify this solution as a category 1 cross-layer solution.

In the sections below, we discuss other mechanisms on the lower layers that are combined with video source adaptation.

5.2.2 Congestion control

When streaming video into a network with limited capacity, there is a definitive danger of causing network congestion, which is discussed in general in Section 3.2. To avoid congestion, congestion control mechanisms should be applied which adjust the rate of which video data is transmitted into the network. As discussed in Section 3.2, the prevalent transport protocol in the Internet, TCP, includes congestion control mechanisms which do not perform well on wireless links and in MANETs. Thus, several cross-layer enhancements to TCP have been proposed.

The disadvantage of implementing congestion control at the transport layer is that the congestion control algorithm cannot take into consideration the delivery requirements of the application data. For applications with strict timeliness requirements, such as interactive video streaming applications, congestion control should attempt to keep the stream as smooth as possible. As the
bit-rate of a video stream may change dynamically depending on the encoder used and the content of the video, smoothing may require coordination from the application layer. In [113], Zhu et al. describe a cross-layer solution to congestion control for wireless video streaming where the video source bit-rate (on the application layer) and the sending rate (on the transport layer) are adjusted by a coordinator component, which interfaces both layers. Furthermore, the solution is adapted to the wireless domain by incorporating an algorithm that uses link layer information about the loss ratio on the wireless network. As the solution described by [113] contains both a coordinator component which interfaces both the application and transport layers, and a new interface between transport and link layer, we classify it as a category 4 solution implemented using the manager method, as described in Section 2.5.

5.2.3 QoS-aware routing

QoS-aware routing, as discussed in Section 3.1, can be particularly useful for streaming video in a heterogeneous tactical environment where resources generally are poor, but where combinations of different network technologies may provides routes with very different characteristics. For example, video streams can be routed through low delay paths, while applications that do not have strict timeliness requirements can be passed through high delay paths. [2] describes such a solution, where a QoS-aware multipath routing protocol for video streaming in MANETS is combined with application level requirements to route video data through the network. We classify this solution as a category 1 cross-layer approach.

5.2.4 Packet loss: Retransmission or Recovery

As discussed in Section 3.2, packet loss happens frequently on a wireless network, and it is not necessarily caused by congestion. First, a radio link may have a steady loss, which can be caused by the static distance between radios, or by obstacles between them, such as buildings, mountains, etc. The steady loss is independent of the traffic on the network. Second, it can suffer from bursty losses caused by unfortunate events, such as mobility (a device or an antenna is moving), sudden interference, or congestion. There are two strategies for handling packet loss:

- Retransmission
- Forward Error Correction (FEC)

For applications that need reliable packet delivery, retransmission is most suitable. TCP, detects and mitigates packet loss by the use of acknowledgments and retransmission. The disadvantage of retransmission is that it takes time. Retransmission can be performed faster if executed by the link layer, a mechanism called Automatic Repeat Request (ARQ) [29]. However, as video streaming data that does not meet its deadline is useless, packets carrying such data should not be retransmitted at all, as they would merely waste bandwidth.

[49] proposes a solution where application layer information is used to optimize the link layer transmission algorithm. Video frames are split up into packets which are stamped with a timeout
which reflects the maximum tolerable delay for that frame. If one packet reaches its timeout value before it is delivered, all packets belonging to that frame is discarded. If applied to retransmission protocols, this approach could possibly make retransmission more efficient, by avoiding retransmitting packets that will not reach their deadline. Let us consider the alternative to retransmission for video streaming over wireless networks: FEC – a technique where the sender encodes data into packets with enough redundant information to allow reconstruction if up to a certain number of the transmitted packets are lost. The advantage of FEC over retransmission in the case of video streaming, is that FEC only causes a very small added delay, in order for FEC to collect enough packets to reconstruct a missing one. On the other hand, FEC has a bandwidth cost as it continuously transmits redundant information: The higher the loss FEC can protect against (called the FEC protection level), the higher is the overhead. In a dynamic environment such as the tactical environment, the FEC protection level should ideally be dynamically adapted to the current network characteristics.

FEC is normally applied at the link layer. When applying FEC in the lower layers, the FEC protection level cannot be coordinated with the application needs unless a CLO scheme applied. For video streaming in a wireless environment, coordination between the FEC protection level and the application opens up for several optimizations. For example, the solution presented in [49] applies a solution which combines video source encoding with adaptive FEC on the physical layer, called unequal error protection. Information about the significance of packets is passed down to the channel coder on the physical layer. The information is used to set a high FEC protection level on important data, and a low level on less important data. This approach is a category 1 cross-layer solution.

As FEC reconstruction depends on that a certain part of a sequence of packets arrive, FEC is sensitive to bursty loss. As wireless links can suffer from bursty loss, the combination of FEC and retransmission will give a more reliable wireless link, though at the cost of delay when retransmission is performed.

5.2.5 Priority-based scheduling

Packet scheduling happens in all nodes that transmit packets out on a link, to decide which packets are transmitted first. In the Internet, which provides best-effort transmission, routers often use First-In-First-Out (FIFO) scheduling. In priority scheduling, packets are tagged with a priority indicator, which are taken into consideration by the scheduler. Priority-based scheduling can be performed on the network layer or on the link layer.

IP has its own mechanism for priority-based scheduling, namely DiffServ (see Section 2.3.2), which uses a designated IP header field to to classify packets into different traffic classes. Routers in the network can be configured to schedule, and when necessary drop, packets based on their traffic class. Several cross-layer solutions based on DiffServ have been proposed in order to improve its efficiency in wireless networks:
For example, in [112], DiffServ is combined with a link-layer routing protocol taking into account the load of different routes, in order to both support the QoS-requirements of soft real-time applications, and evenly distribute load in the network. This is yet another category 1 solution.

In [65] proposes a link-layer scheduling algorithm for multiple connections in wireless networks, where each connection belongs to a service class which is associated with QoS requirements. The algorithm assigns a priority to each connection based on its QoS requirements and information about channel status, provided by the physical layer, and the connection which has the highest priority is scheduled first. By combining information from the physical layer with QoS requirements, this solution provides both QoS guarantees, and efficient bandwidth utilization. This is also a category 1 cross-layer solution.

6 Conclusion and recommended future work

In this report we have discussed various approaches to cross-layer optimization (CLO) and seen that solutions can be categorized in different manners. The four approaches to CLO are: 1) Creation of new interfaces, 2) Merging of adjacent layers, 3) Design coupling without new interfaces, and 4) Vertical calibration across layers.

On the other hand, when the main focus is on how the information is shared, then the two main approaches are the manager and the non-manager method. The main distinction here is whether the solution modifies the existing communication stack directly, or introduces a separate, parallel management plane for optimization. The different solutions have different strengths and drawbacks, as we have discussed previously in this report. Regardless of the way we choose to approach CLO solutions, the main challenges of CLO remain:

- How can CLO solutions co-exist with existing deployments? This is important when joining a federated mission network, for example.
- How can cross-layer signaling be realized in an effective yet efficient manner? As TLL focuses on the tactical domain where resources are scarce, it is important that attempts at optimization do not introduce significant signaling overhead.
- How can we introduce cross-layered solutions for maximum benefit while still minimizing the impact of violating layered design? There is no overall agreed-upon approach to CLO, meaning that at the present time there is a risk of trading quick CLO gains for long-time drawbacks like complex and costly maintenance due to proprietary elements in the system.

There are no quick, easy answers to these questions regarding CLO for the areas we have covered in this report. Therefore, further research is needed. Below we highlight some potential future work that could be pursued in the context of TLL for the areas covered in this report.
6.1 Wireless networks

To provide a reliable network for different applications in different operation types and in varying terrains, a tactical mobile network must consist of a variety of wireless network types, e.g., long-range communication for reach-back connections and a higher data-rate network for local communication. A single transmission technology will not be able to support all communication types and data-rate requirements. The consequence is that several different radio systems might be present on a platform (e.g., SatCom on the move, Radios from P8151, MRR MLU, NBWF etc.). In a situation like this it will be beneficial if the available network resources can be presented to the warfighter as one common network. In order for this to work, the different radio networks must be interconnected with a routing protocol that can provide different routes for different QoS requirements and that can implement the network policies that the signal corps want to enforce in a specific operation. As shown in this report, cross-layer information can greatly improve QoS routing. Many proposals exist for QoS routing over homogeneous networks, but not so much work is done on QoS routing for network of networks (heterogeneous networks). This is an important area to pursue in order to reach the NML4 goal of “a common unified infrastructure”. A consequence for heterogeneous QoS routing is that the protocol need to access cross-layer information from different radio systems, thus work on standardizing an interface between network layer and link layer as shown in the effort on the DLEP protocol, is important in this context. Fraunhofer has a working implementation of the DLEP draft and are interested in cooperation with FFI on this topic. Finally, when a routing protocol wants to use cross-layer information provided from different protocols in different radio systems it is important to have information about the age, quality, accuracy etc of the cross-layer information to decide how the information can be utilized. We believe it is important to place more research effort into these three areas: Better QoS routing for heterogeneous networks, standardization of the types of cross-layer parameters that are relevant to provide, and definition of a data model to describe lifetime, quality etc. of cross-layer information with.

6.2 Sensor networks

Wireless Sensor Networks have gradually evolved from being single-layered and highly optimized proprietary systems toward having a more modular and layered architectural design. This enables new possibilities to interconnect WSNs with the tactical network infrastructure and, more specifically, to provide end-to-end communication from a WSN node to an end user (typically a soldier C4IS or a battlefield management system). It is crucial to continue further standardization efforts of WSN protocols in order to reach the NML4 goal of “a common unified infrastructure”.

Although WSNs now share the basic layered architecture with the rest of the tactical network infrastructure, the protocols must still be adapted and optimized to be suitable for energy and bandwidth limited sensor nodes. One such adaptation is 6LoWPAN, which provides IPv6 capabilities for low-power WSNs by compressing the header fields in IPv6. Such adaptations do, however, mean that full compatibility between WSNs and external networks cannot be guaranteed.
Furthermore, there are some protocol functions that are very difficult to transfer from the external network side to the WSN side. This may for example apply to service discovery and multicast. In order to implement such protocols, cross-layer optimizations are inevitable due to the limited energy, memory and bandwidth in WSNs. In particular, cooperation between the physical layer, the mac-layer and the routing protocol needs further exploration.

Most current cross-layer optimizations for WSNs are single purpose and they are often based on proprietary protocols. One important aspect that should be pursued for a future tactical network is therefore to design a cross-layer architecture for WSN based on a vertical plane functioning as a manager across standardized layers.

6.3 Middleware

Web services are lacking in standardization when it comes to QoS. This means that we must both keep an eye on developments within standardization and experimental solutions. Standards are preferable when they exist due to interoperability concerns, but if no standard is available then proprietary solutions can be employed. If proprietary solutions are chosen, then it is important to limit their impact by striving for backward compatibility with COTS services and clients (which shall receive best-effort when lacking proprietary extensions. Bringing SOA and Web services fully into the tactical domain has been a topic for quite some time, and as Section 3.3 summarized there are some experimental solutions that leverage CLO. As a minimum, we recommend that TLL considers the CLO approaches discussed in that section:

- Signaling of QoS (e.g., priority) down to lower layers, and
- Signaling of available resources up from lower layers.

Here, one should be able to establish a testbed in not too long a time, by using knowledge from past experiments and the interface in Appendix A as building blocks in a demonstrator.

Further synergies can be achieved by attempting to combine QoS-aspects with claims-based access control (along the same lines that our previous solution attempted to leverage RBAC). Here it is important to harmonize such an attempt with NATO’s standards for claims-based access control, as NATO’s solutions should be used to secure the middleware anyway. Further, one could look deeper into using semantics for QoS descriptions (some form of rich metadata is needed to manage and exchange QoS information – not only by the middleware but generally for the entire CLO approach).

Pursuing these aspects can be a first step towards CLO for middleware in a TLL context. However, a more holistic approach is preferable, considering the NML4 goal of “a common unified infrastructure”, one should look at CLO for the entire communication stack in an attempt to go beyond simply communicating upward/downward between two layers (i.e., the approach of creating a few new interfaces), but instead makes an attempt at a complete vertical calibration across layers. This latter approach should preferably be investigated along the lines of the manager method shown in Figure 2.6.
6.4 Video streaming

Current works claim that their cross-layer solutions are able to optimize video streaming in wireless networks and MANETs. As video streaming in the tactical environment can be extremely challenging, and sometimes impossible, CLO may contribute to exploiting narrow bandwidth connections better, to provide video streaming services of a higher quality.

As discussed in [38], state-of-the-art video streaming technologies are capable of dynamically adapting to the resource availability, in particular the available bandwidth. In particular, they do perform application layer video source adaptation, to ensure that the video bitrate is scaled to fit the currently available bandwidth. An interesting experiment to execute in TLL would be combining these existing products with mechanisms in the lower layers, in order to compensate for potential shortcomings of the products, or to enhance their effect.

One candidate mechanism for such an experiment is multipath routing. As the paths available in a tactical network can have very different characteristics, the combination of video source adaptation and multipath routing may be particularly effective with regard to optimize network resources utilization. In particular, multipath routing may provide failover in the case of network partition, which can be particularly challenging in the tactical network. Multipath routing can be further combined with other mechanisms in the lower layers (physical, link, network, and link layers), to ensure that packets are either prioritized and delivered quickly, or discarded, avoiding that the network is clogged by data which will not reach its deadline anyway.

Most of the solutions mentioned in Section 5.2 are category 1 solutions, where new interfaces are created downward from a higher layer to a lower layer, or upward from a lower layer to a higher layer. A potential problem of such cross-layer solutions is increased complexity in the layers involved. Furthermore, the layers are bound inherently together, and the benefits of layer independence in future maintenance and development can be lost, with the consequence that COTS solutions cannot be used. Thus, as pointed out in Section 2.2, cross-layer mechanisms should be used with caution.

In TLL video streaming is an interesting case, as it poses different challenges for the communication solution than e.g., message-oriented middleware does. Both kinds of application can, as we have seen in this report, in their own way benefit from CLO. However, it is important to approach CLO in a unified manner, in order to reach the intended NATO maturity level (i.e., NML4). As a final remark, it is worth noting that video is just one of the Unified Communication and Collaboration (UCC) services. For a discussion of UCC services, see [38].

6.5 IA

Simplicity is in general very important for analysis and implementation of information security mechanisms. The potential gain has to be substantial before the added complexity of cross-layer solutions should be considered.
One area of concern in TLL is to avoid unnecessary traffic in tactical networks due to information security. The main tool to achieve this should be to choose security mechanisms and protocols suitable for a tactical environment. However, it could also be beneficial to see the information security architecture of the system from a cross-layer perspective to avoid duplicating expensive functionality. Here it would be important to harmonize the choice of mechanisms with the current trends within IA in NATO, to ensure interoperability with the coalition and reach the overall NML4 – “Collaborate” – goal. Thus, we have not identified any specific IA aspects that should be pursued in TLL at present, but merely point out the need to consider the impact of security requirements when pursuing CLO in the other areas mentioned in this report.

6.6 SMC

Service management and control functions could also be looked into. One subtopic that stands out here is that of service discovery, which, as previous research has shown, can benefit from CLO. Currently NATO has pointed to UDDI for service discovery in the SOA baseline, and again in the description of FMN. As UDDI is a centralized registry solution it may not be directly applicable in the tactical domain since it constitutes a single point of failure. Thus, we should follow what happens with service discovery in NATO in the future. In any case it is not interesting to investigate UDDI for CLO specifically, since it just employs a standard Web service interface. That means that if we pursue the optimizations for middleware suggested above, we will automatically gain the same benefits for UDDI. Conversely, when a mechanism that is more suitable for the tactical domain than UDDI is identified, it would be interesting to consider it for CLO specifically. Currently, both the TIDE community (involved with making so-called transformational baselines), and STO/IST-118 “SOA recommendations for disadvantaged grids in the tactical domain” are considering different approaches to service discovery. FFI project 1277 is already involved in this work, so we recommend that TLL makes use of any relevant findings there.
References


FFI-rapport 2014/00985 49


Appendix AVyatta MTR QoS information retrieval system

The Vyatta Multi Topology Router (MTR for short) is a software based IP-router built on Linux, which can be run on an ordinary PC. In addition to ordinary packet routing, the MTR’s job is to keep track of available data carriers and associated routes for any given traffic flow, and to prioritize traffic according to a set of rules or to discard traffic that shall not or cannot be sent over certain types of carriers (e.g. video stream over HF-radio). All routing decisions are made solely by the router, and all the information that the router bases those decisions on are available only inside the router. However, sometimes it could be beneficial for an application to know what quality of service (QoS) it can expect in a given situation before it decides whether or not to transmit data. To accomplish this, a specialized service module named SOAQoS\textsuperscript{10} has been created which can be added to the Vyatta MTR. Details for how to build Vyatta modules for MTR can be found in [73].

A.1 Querying for information

When the SOAQoS service is running on an MTR, a remote application can get a list of possible routes (both IPv4 and IPv6 addresses are recognized) with associated topology- and route information by sending a text-based UDP-message to the router and wait for a response. The service listens to port 19151 by default, but this can be configured as needed. There can be only one query per message, but the replies can consist of multiple lines separated by a \textbackslash n-character (ASCII 0x0a). All messages are terminated by a zero-byte, thus an empty response message will still be one byte.

In the current implementation SOAQoS recognizes two types of queries:

\textbf{address} Lists all available routes to a particular address or subnet

\textbf{net} Lists all available routes that can be considered a subnet under the specified net

Below is an example of an “address”-query. Note that indented lines means that the text is continued from the previous line with no line break in between.

\underline{Query:}

\begin{verbatim}
address fc10:f510:400::1
\end{verbatim}

\underline{Response:}

\begin{verbatim}
 1 address=fc10:f510:400::1 route=fc10:f510:400::/64 tc=0x28 min-bw=0 max-bw=10000 min-delay=2 stability=3 distance=110 cost=60
 2 address=fc10:f510:400::1 route=fc10:f510:400::/64 tc=0x18 stability=1 distance=110 cost=120
\end{verbatim}

\textsuperscript{10}The name stems from the assumption that it will most likely be used by middleware SOA-applications.
Here, the router reports to active routes to the destination address fc10:f510:400::1. We can select which one we wish to use by tagging the traffic with the associated tc (Traffic Class = DSCP) value. The other values indicates the quality and type of the route. Two of them – “distance” (administrative distance) and “cost” – are gathered from the routing table while the others are user-defined and must be specified per-scenario. The values for “min-bw” and “max-bw” (bandwidth) are meant to be specified in bits per second, “min-delay” should be given in milliseconds, while “stability” is an arbitrary value which must have a shared definition for all participating routers. Undefined values are never included in the response.

A route belonging to the main-table (table number zero) or which belong to a topology marked as “catch-all” will by default not be returned since we normally should not be using those. They are “fallback”-routes which will only be used when the router has nowhere else to send a packet. If we want them included anyway we can specify “show main” and/or “show catch-all” in the query, like this:

```
address show main fc10:f510:400::1
address show catch-all fc10:f510:400::1
address show all fc10:f510:400::1
```

In the third query above, “show all” is the same as requesting both “main” and “catch-all”.

Here is an example of a “net”-query:

**Query:**

```
net fc10:f510::/32
```

**Response:**

```
1 net=fc10:f510::/32 route=fc10:f510:100::/64 tc=0x28 distance=110 cost=60
2 net=fc10:f510::/32 route=fc10:f510:400::/64 tc=0x28 min-bw=0 max-bw=10000 min-delay=2 stability=3 distance=110 cost=60
3 net=fc10:f510::/32 route=fc10:f510:400::/64 tc=0x18 stability=1 distance=110 cost=120
```

The router reports its active routes which have an equal or narrower subnet than the one given in the query. Of the three routes here, two of them (fc10:f510:400::/64) are the same route but with different tc-values and thus presumably they have different quality.

**A.2 Configuration in the MTR**

Configuration of the SOAQoS-daemon is done using the Vyatta command-line interface (CLI). The service will start as soon as the following configuration path exists:

```
service soaqos
```

The service settings can be tweaked from their default value via a set of sub-parameters.
To specify the port SOAQoS listens on for queries (default port 19151):

```
service soaqos listenport port#
```

To enable logging of all queries and responses to default file /var/log/soaqos:

```
service soaqos log
```

To specify a different log file name:

```
service soaqos log filename filename
```

To activate verbose logging which gives much more information:

```
service soaqos log verbose
```

To include debugging information in the log file:

```
service soaqos debug
```

It is also possible to temporarily disable the daemon by setting the following configuration parameter:

```
service soaqos disable
```

The scenario-specific topology path weighting values are defined under each topology’s traffic-class:

```
topology nn traffic-class xx soaqos max-bw value
topology nn traffic-class xx soaqos min-bw value
topology nn traffic-class xx soaqos min-delay value
topology nn traffic-class xx soaqos stability value
```
Appendix B  The NATO C3 Classification Taxonomy

The NATO C3 Classification Taxonomy is an categorization of the functionality that is expected to be found in NATO’s information infrastructure (NII). More precisely, it is a conceptual framework for sorting the capability concepts that are relevant to producing enterprise computer system in support of Consultation, Command and Control (C3) in NATO. As interoperability with NATO and key partners is a major concern and an outspoken goal by the Norwegian Chief of Defense (see [97], section on interoperability) it makes sense to consider NATO’s C3 Taxonomy. The key points of the taxonomy (summarized from [110]) are highlighted below. For a more complete discussion of the taxonomy and how it relates to the architecture for the Norwegian defense information infrastructure (INI), see [12].

B.1 Introducing the taxonomy

The NATO Network Enabled Capabilities (NNEC) Feasibility Study of 2005 proved that the communications and information systems (CIS) environment for NATO must adapt a Service Oriented Architecture (SOA): organizing software in the form of independent, interoperable services that can be composed and recomposed to fulfill multiple business requirements [6].

The study presented a Technical Services Framework, with a hierarchical arrangement of technical services in four horizontal classifications, plus two vertical groups: for Information Assurance (IA) and Service Management & Control (SMC). Soon it was recognized that this was not sufficient for a full representation of the C3 structure, and that a taxonomy should have a broader and deeper scope.

An important observation was that the requirements for future C3 capabilities are not purely technical in nature. A framework for CIS services would only address the back-end technology solutions, and would not give any resolution about quality and quantity of services required for a particular mission. The new C3 Classification Taxonomy therefore presents both the definition of CIS capabilities and their operational context. The top-level view of the taxonomy is illustrated in Figure B.1.

B.2 Operational Context

The Operational Context describes the environment in which CIS capabilities are defined and used. This information provides the organizational perspective in which the CIS technology solutions will be deployed in order to achieve success in NATO’s future missions.

B.3 CIS Capabilities

With the operational context set, it is time to link it to a technical framework of applications, services, and equipment. The CIS Capabilities span two significant categories: the User-Facing Capabilities and the Technical Services.
The first category provides an end user with *User Applications* that are designed to help the user to perform singular or multiple related tasks, analogue to the apps on modern phones and tablets.

The second category provides the foundation for the better use of technology: A set of related software and hardware functionality that can be used in support of each other, and in support of the applications mentioned above.

![Figure B.1 NATO C3 Classification Taxonomy (from [110])](image-url)