FFI-RAPPORT

19/00602

Information-Centric Networking for mobile military networks

Mariann Hauge
Lars Landmark
Øivind Kure
Frank T. Johnsen

Information-Centric Networking for mobile military networks

Mariann Hauge
Lars Landmark
Øivind Kure
Frank T. Johnsen

Norwegian Defence Research Establishment (FFI) 5 June 2019

Keywords
Mobile nettverk
Kommunikasjonsnettverk
Kommunikasjonsprotokoller
Informasjonsinfrastruktur
Tjenesteorientert arkitektur

FFI-rapport
19/00602

Prosjektnummer
1367

ISBN
P: 978-82-464-3192-5
E: 978-82-464-3193-2

Approvers
Jan Erik Voldhaug, *Director of Research*

The document is electronically approved and therefore has no handwritten signature.

Summary

Modern warfare requires an information infrastructure that facilitates extensive information sharing. It is challenging to build networks that can support this in an efficient manner, particularly for mobile forces.

Information-Centric Networking (ICN) is the name of a group of new network architectures that might form the basis for the future Internet. ICN introduces a completely new way of accessing information in a network by addressing the *name* of the information instead of the IP address of the server that produces or stores the information. This allows for a better integration of the information infrastructure with the communication infrastructure and can potentially leverage faster and more efficient information sharing. ICN also has some characteristics that can potentially be beneficial for the performance of future mobile military networks. Some of ICN's interesting characteristics are its ability to efficiently support communication both to a group of receivers and between the source and a single receiver within the same architecture as well as ICN's ability to handle moving nodes and highly unstable network connections.

Given these interesting characteristics, this report explores this emerging and disruptive technology to assess if the technology should be considered for the Norwegian Armed Forces' future mobile military networks. Our method has been a combination of literature review and experimentation with demonstrators that implement a popular ICN architecture called Named Data Networking (NDN). We have reviewed the literature that studies NDN for mobile military networks and a selection of other relevant studies. As part of the process of building a demonstrator we have implemented an extension to the existing open source code. Our extension supports the search for information in mobile military networks. The extension can be used as a stepping stone for further optimizations of such networks.

We have studied how NDN can potentially improve the efficiency and stability of mobile military networks, and we list a range of potential advantages as well as known challenges with the use of NDN in mobile military networks. We have studied in more detail methods used to find information in the network since this is important for the tradeoff between efficiency and robustness in mobile military network. It is also important for NDN's ability to support the information infrastructure. Finally we have studied how well NDN can support the information infrastructure and have compared NDN with two relevant Service Oriented Architecture (SOA) approaches.

We conclude that NDN is worth studying closer, but that the maturity is low. The studies have strengthened our view that this architecture has interesting characteristics for use in mobile military networks. We also acknowledge that there are unresolved challenges associated with the NDN architecture that must be solved before this architecture can be considered for deployment in the military networks. Examples are; design of a scalable namespace, confidentiality protection of the search for information as well as efficient and robust search for information in mobile military networks. We recommend further studies of this architecture as it matures, to see if the challenges can be solved in a sound manner.

Samandrag

I dagens militære operasjonar er det nødvendig med ein informasjonsinfrastruktur som gjer det enkelt å dela store mengder informasjon. Det er vanskeleg å byggja slike nettverk, spesielt for mobile styrkar.

Information-Centric Networking (ICN) er fellesnamnet på fleire forslag til nye nettverksarkitekturar som kanskje kjem til å verta ein del av teknologien for internett i framtida. ICN introduserer ein ny måte å finna informasjon i eit nettverk på ved å bruka *namn* som identifikator på informasjonen i staden for IP-adressa til sørvaren som produserer eller lagrar informasjonen. Dette kan gje betre samhandling mellom informasjonsinfrastrukturen og kommunikasjonsinfrastrukturen og setja nettverket i stand til å tilby kjappare og meir effektiv deling av informasjon. ICN har også nokre eigenskapar som kan vera fordelaktige for framtidige militære mobile nettverk. Støtte for kommunikasjon både til ei gruppe med mottakarar og til ein enkelt mottakar innanfor same arkitektur, og dessutan støtte for mobile plattformar og ustabile radiolinkar, er eksempel på fordelaktige eigenskapar.

I denne rapporten studerer me denne nybrotteknologien for å kunne vurdera om teknologien bør koma i betraktning for ny arkitektur for mobile militære nettverk til det norske Forsvaret i framtida. Metoden me har brukt i arbeidet er ein kombinasjon av litteraturstudiar og eksperimentering med demonstrator. Me valde å byggja demonstratorane ved hjelp av Named Data Networking (NDN), som er ein populær ICN-arkitektur. Som ein del av arbeidet med å laga ein demonstrator har me utvikla eit tillegg til den opne NDN-kjeldekoden. Tillegget vårt gjev betre støtte for søk etter informasjon i mobile militære nettverk. Dette tillegget kan brukast som eit utgangspunkt for ytterlegare optimalisering av NDN-arkitekturen for slike nettverk.

Me har studert korleis NDN kan betra effektiviteten og stabiliteten til mobile miltære nettverk, og me viser både potensielle fordelar ved bruk av slike arkitekturar i tillegg til kjente utfordringar. Me studerer spesielt metodar for å søkja etter informasjon i mobile miltære nettverk sidan denne funksjonen er viktig for å kunne styra effektiviteten til nettverket (effektiv ressursbruk) opp mot nødvendig motstand mot pakketap. Søkjemetoden er også viktig for at NDN skal kunne gje god støtte til informasjonsinfrastrukturen. Me har også gjort ein kvantitativ studie av NDN mot to relevante løysingar for tenesteorientert arkitektur for å vurdere kor effektiv NDN si støtte til informasjonsinfrastrukturen er.

Me konkluderer med at NDN er interessant og bør studerast vidare, men at arkitekturen ikkje er moden. Studiane me har gjort, styrkjer trua vår på at denne arkitekturen har eigenskapar som kan vera nyttig i mobile miltære nettverk. Samtidig er det viktig å understreka at arkitekturen har utfordringar som må løysast før den kan brukast i operative nettverk. For eksempel må ein skalerbar namnestruktur på plass. Det er også nødvendig å støtte konfidensielt søk etter informasjon og dessutan effektive løysingar for søk etter informasjon i mobile militære nettverk. Me anbefaler å fortsetja å forska på denne arkitekturen for å finna løysingar på dei viktige utfordringane.

Contents

Preface

The work is a deliverable for the NORDEFCO project “Group-communication for military Mobile Ad Hoc Networks (MANET)”. This project is a bilateral effort between FOI and FFI where we have studied different solutions for efficient support for group communication in mobile military networks. This report is the final report of work package 2 (WP2) of the NORDEFCO project.

Kjeller, 29th of March 2019

Mariann Hauge, Lars Landmark, Øivind Kure og Frank T. Johnsen

1 Introduction

Modern warfare requires an information infrastructure that facilitates extensive information sharing. It is technologically challenging to build networks that can support this in an efficient manner, particularly for mobile forces, and mobile military networks typically have difficulties providing the necessary connectivity and data-rates.

In the 2005 NATO Network Enabled Capability (NNEC) Feasibility Study [1], Service-Oriented Architecture (SOA) and a unified communications networking infrastructure were identified as two key components in supporting NNEC. SOA is a way of making military resources available as services so they can be discovered and used by different entities in an efficient manner. A unified communications networking infrastructure makes sure that those services can be accessed and utilized in different locations and by different consumers. In NATO's Consultation, Command and Control (C3) taxonomy (Figure 1.1), these two components are elements of the Core Services[1] and Communication Services respectively. Since the NNEC Feasibility Study, the military community at large has committed to these ideas for how to build infrastructures and it is currently the approach leveraged for NATO's Allied Command Transformation's (ACT) activity to specify how to do Federated Mission Networking (FMN) [2]. FMN defines information networks and processes for efficient information sharing and collaboration between different nations in multinational military operations.

The challenges of mobile military networks (e.g. intermittent connectivity and low data-rates) make it difficult to sustain a stable unified communications networking infrastructure as well as providing a Service-Oriented Architecture (SOA) with good performance. In most of the remaining text we refer to these two components as the communication infrastructure and the information infrastructure respectively. In the remainder of this chapter we define the future network scenario assumed for the studies described in the report, detail the problem description for the studies, briefly explain the technology we have studied and present the method we used for the work.

[1] Core Services provide generic and domain independent technical functionality realizing SOA principles. Core services include communication and collaboration, Web and information services, processing, composition and mediation, to name a few. Generally speaking, core services enable and facilitate the operation and use of information technology resources independent aspects related to communication, information assurance and service management and control.

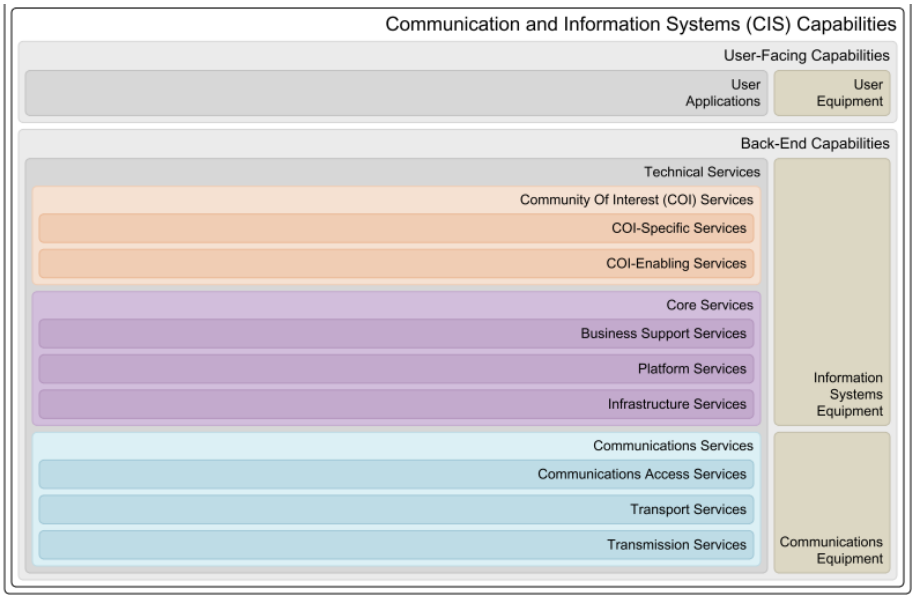


Figure 1.1 The figure shows the Communication and Information Systems (CIS) Capabilities part of the C3-Taxonomy [3].

1.1 Assumed future network scenario

For this study we foresee a challenging future scenario (Figure 1.2) with one or several of the characteristics described in the following.

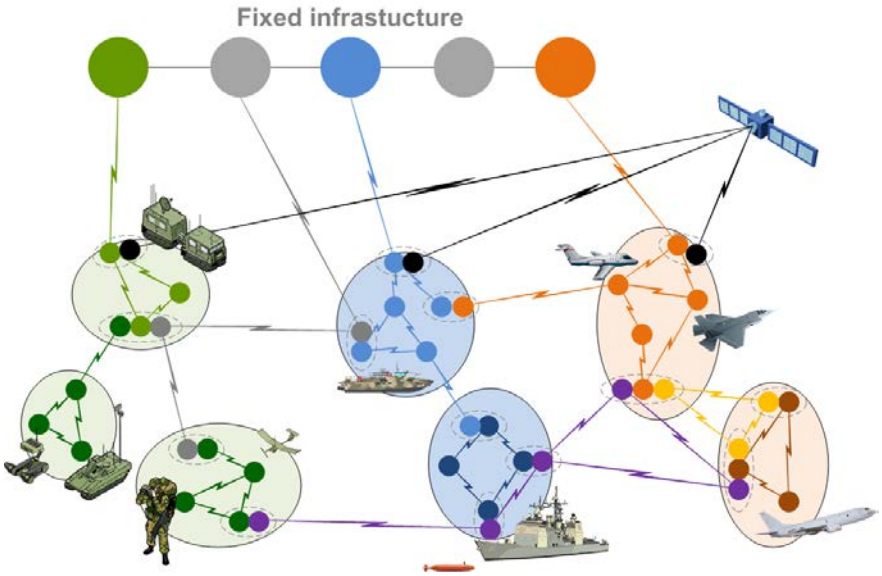


Figure 1.2 The illustration shows a mobile military network that connects a range of sensors and effectors. The different colored radio signals represent different transmission technologies.

1) We expect to see a need for more network capacity (e.g. higher data-rates) also in the mobile parts of military networks. This can be mitigated with deployment of more mobile networks, by utilizing existing networks better or a combination of both. Some of the drivers that push this need are:

- Emergence of Internet of Battlefield Things (IoBT) [4] that attempts to connect all deployed military sensors as well as useful civilian sensors (e.g., traffic cameras and weather stations) in one network for improved situational awareness and intelligence. We believe this will lead to a higher density of deployed sensors (also at the tactical edge and in enemy territory) and thus will require more network capacity.
- The future will likely see an increasing number of deployed autonomous nodes and swarms as these technologies matures (see e.g., [5]). The swarms must coordinate their behavior over a network and the autonomous nodes require network capacity to receive orders and report sensor information or other findings.
- New communication patterns are also likely to emerge as a consequence of the deployment of an increasing number of information sources (e.g. sensors) that can be exploited in both planned and ad hoc manner as military forces move through areas that are being supported by different sensors. As the information becomes attractive for a growing number of receivers, more network capacity is needed to carry the information.
- An increasing focus on joint and combined operations (see e.g., [6]) requires better information flow between different military branches and/or coalition partners. This also means that there will be more traffic on the network.
- Joint and combined operations will benefit from Joint Intelligence, Surveillance and Reconnaissance (JISR) as well as joint firing and targeting. These activities will also push for a higher network capacity.

2) We expect to see a complex heterogeneous mobile network[2] environment that might have to sustain frequent topology changes[3] in the mobile part of the military network. Some of the drivers for this change are:

- It is expected that the Norwegian Armed Forces often will have to sustain electronic warfare (EW) attacks (e.g. [5]). Networks under attack by enemy EW-capacities will likely become very unstable, and have intermittent connectivity. One way to partially mitigate this threat is to deploy several networks based on different transmission technologies[4] in the mobile military network, since it is more difficult to attack several

[2] A heterogeneous mobile network is a network that consists of multiple transmission technologies, e.g., terrestrial military radios with different waveforms, terrestrial civilian radios, and different types of satellite communication.
[3] Frequent topology changes means that as the nodes in the network moves, radio-connections between the nodes come and go and the route that a data packet must take in order to get from the producer to the consumer via one or several relays, changes frequently. It is very difficult to maintain connected routes in such networks.
[4] We use the term transmission technology to refer to a network segment that uses one unique transmission technology. The transmission technology encompasses physical (PHY) layer design, medium access (MAC) layer design and logical link control (LLC) layer design.

different transmission technologies simultaneously. The consequence is a heterogeneous mobile network with frequent topology changes.

- In [5] it is also foreseen that there will be a need for data communication to deployed and mobile nodes over long distances in areas with little existing network infrastructure. This is needed to provide network connectivity to highly mobile military units as well as autonomous nodes and an increasing number of deployed sensors. A heterogeneous network that consists of a mixture of different transmission technologies might be necessary to provide the required connectivity.
- In order to provide the network capacity that was discussed in 1) above, a heterogeneous mobile network that consists of wideband high data-rate wireless transmission technologies in combination with low data-rate transmission technologies, might be used. High data-rate technologies typically have much shorter range than low data-rate technologies and thus require more radio-hops (relays) to get from the producer to the consumer. The consequence is often frequent topology changes and unstable network connections (e.g., [7]).

3) We also foresee a future with agile services, i.e., services that come and go as the mission changes character and that can utilize information that becomes available at different sites in an ad hoc manner. Utilization of civilian sensor information from smart cities is one example which is further explored in [8]. In order to find the available information and facilitate extensive and quick information sharing, improved interaction between the communication network infrastructure and the information infrastructure is advantageous.

1.2 Problem description

The current Internet architecture based on IP technology uses a host-centric communication model, where the IP address of the host is the key to access the information produced or stored by the host. This model was appropriate for coping with the needs of the early Internet users. Internet usage has evolved however, with most users today mainly interested in accessing (vast amounts of) information, irrespective of its physical location. In the early Internet times, all users were stationary, whereas now more and more content is requested by mobile users. Over the years since IP was introduced many protocols have therefore been added to the IP architecture to support new usage patterns as well as to make IP more flexible. Some examples are:

- Functions in the communications networking infrastructure
    - Protocols for mobile ad hoc networks (MANETs) [9] to build infrastructure-less mobile networks.
    - Protocols for disruption/delay tolerant networking (DTN) [10] for very unstable networks.

- Protocols for traffic to a single consumer (unicast) or to a group of consumers (multicast) [11].
- Mobile IP [12] and Network Mobility (NEMO) [13] protocols to handle node and network mobility.

- Functions in higher network layers
  - Domain name servers (DNS) that associate a domain name with an IP address prefix.
  - A range of information dissemination protocols (e.g., Data Distribution Service (DDS) [14]) to facilitate efficient information sharing in the information infrastructure.
  - Content delivery networks (CDN) [15] that build an infrastructure with distributed caches and protocols to facilitate efficient information sharing.

There has been much research on MANETs and other wireless network technologies during the last three decades and many proposals and products exist that solve different aspects of the range of challenges and requirements that these networks meet such as much packet loss, relatively low and varying data-rates and unstable network connections. A fundamental problem with the host centric IP model is that it relies on a connected route from the producer to the consumer for the duration of a data-flow, to deliver the data. This sets an upper bound on the number of link outages (due to e.g., mobility or EW attacks) that these networks can sustain. DTN solutions introduce extra functionality in the IP networks to allow some traffic to flow also in very disruptive networks where a connected route is often not available. DTN is not a general solution and can only be utilized in specialized networks. Protocols for efficient group communication are another add-on to the IP architecture. A range of protocols exist that are tailored for different optimizations (level of mobility, robustness, etc,). Consequently, one major problem with the pool of IP based solutions is that they are not general and that there is a need for a variety of different protocols for different needs. This results in a need for management solutions that can decide when and where to use which of the many specific protocols and mechanisms. This is difficult to do automatically and likely requires costly and error prone manual configuration. Several of these protocols also have a high overhead and put much load on a mobile military network. Furthermore different transmission technologies often implement different protocols from the pool of solutions. Often the protocols (e.g. for group communication) are not interoperable. This makes it very hard to come up with IP based solutions that utilize heterogeneous mobile networks efficiently. Several of the points above are also stressed in [16].

We have performed research on methods for efficient group communication[5] in mobile military networks for some time and have studied classical IP multicast protocols for MANETs [11, 17]

[5] We use this term to describe information exchange between a group of receivers. The network can support this type of communication with different means with different characteristics e.g., reliability, efficiency, delay. As an example, multicast protocols [11] is one way to support this type of traffic.

and hybrid IP multicast [18, 19]. In those works we proposed solutions that could improve the IP network performance, but observed that the space of solutions becomes more and more complex. A new architecture that can reduce the complexity of the network architecture is welcomed.

Similarly we have looked for good solutions to build stable and efficient heterogeneous mobile networks with the IP technology in our past work[7, 20-23] with the same conclusions as above.

Since it is difficult to provide the necessary support for efficient information sharing over a heterogeneous mobile environment in the communication network infrastructure, a range of middleware solutions as well as dedicated architectures has been developed to improve the performance of the IP architecture (e.g., CDNs and data dissemination protocols for SOA). However, it has also been difficult to take full advantage of SOA for the information infrastructure in mobile military networks. Building an information infrastructure in the military domain differs from building one in the civilian domain due to the intermittent connectivity and low data-rates of the mobile military networks, and civilian solutions can rarely be used out of the box [24]. Both NNEC and the early spirals of FMN focus on interoperability for fixed network infrastructures and deployed semi static installations, where network resources are abundant. Therefore, the standards recommended for implementing the various core services were chosen merely based on their suitability as a federation mechanism. Hence, said standards are not necessarily well suited for use in mobile military networks where network capacity typically is low. A new architecture that can provide an efficient realization of the SOA paradigm in the information infrastructure in such networks is welcomed.

1.3 Target technology

Information-Centric Networking (ICN) [25] is the name of a group of clean slate network architectures that might form the basis for the future Internet. ICN introduces a completely new way of accessing information in a network by addressing the *name* of the information instead of the IP address of the server that produces/stores the information. This allows for a better integration of the communication infrastructure and the information infrastructure and can potentially leverage faster and more efficient information sharing.

ICN has some characteristics that can potentially be beneficial for the performance of both the information infrastructure and the communication infrastructure for the assumed future network scenario described in section 1.1. ICN does not depend on long lived stable network connections to retrieve information. ICN also handles the change in traffic pattern for information meant for a single consumer and for a group of consumers, seamlessly. This is also the case for the change in traffic forwarding for fairly stable networks to networks that experience intermittent connectivity.

Given these interesting characteristics, this report explores this emerging and disruptive technology from the view of both the communication infrastructure and the information infrastructure to better understand how the technology works and to reach a preliminary

decision if it is worth studying the technology further as a candidate architecture for the Norwegian Armed Forces' future mobile military network. We've focused on the use of ICN for the assumed future network scenario described in section 1.1 since it has proven to be challenging and complex to build stable IP networks for these network types.

1.4 Method

Some of the early ICN work (e.g., [26]) was published more than a decade ago. But since ICN is a clean slate architecture that can be used instead of the current IP architecture it takes a long time for the architecture to mature. Up until recently most of the research has been done on typical Internet infrastructures. Our purpose on the contrary, was to study how well ICN can handle mobile military networks. One of the best methods to get well acquainted with a technology is to build a demonstrator. Since open source code was available for at least one of the interesting ICN architecture proposals this allowed us to use this method for our work.

We completed two parallel activities that explored the use of ICN in mobile military networks. One activity studied how ICN could potentially improve the network performance of mobile military networks. The other activity studied how ICN could potentially improve the information infrastructure for mobile military networks. For both activities we 1) performed extensive literature studies to understand the theory and open research topics and 2) experimented with an ICN demonstrator. We chose one candidate ICN architecture called Named Data Networking (NDN) [27] for our demonstrators since this is a popular architecture with an active open source codebase. For the activity that focused on how ICN could potentially improve the network performance of mobile military networks we wanted to build an NDN demonstrator that was fitted for use with shared channel radios. To achieve this we chose to implement an extension to the NDN codebase. By going through the process of modifying the code and building a demonstrator we aimed to get better understanding of how NDN works in more detail than we would have done through literature studies alone. Furthermore, by modifying the code we would get a better understanding of how simple or hard it would be to introduce other functionality to the code.

For the activity that studied how ICN can potentially improve the information infrastructure for mobile military networks we wanted to do a quantitative study where we compared the performance of NDN when used as the underlying communications architecture for SOA, with standards that are relevant to SOA both from a civilian but also from a NATO core services perspective. For this we used the unmodified NDN codebase in the demonstrator. The services that provided data for the experiment had to be tailored specifically for this experiment, both with regard to the industry standards and the NDN implementation. Our evaluation compared the selected methods for some important performance parameters.

The remainder of this report is organized as follows: Chapter 2 explains how NDN works. Chapter 3 gives a high-level list of expected advantages and disadvantages with the technology. Chapter 4 goes into more detail of some advantages and disadvantages associated with the use of NDN in mobile military networks, and describes some NDN code structure as well has how

we chose to implement a wireless extension to the NDN codebase. Chapter 5 describes an experiment we did to compare NDN's information retrieval with selected publish/subscribe implementations. Chapter 6 provides a list of literature for further reading and discussed relevant related work. Finally chapter 7 presents concluding remarks and suggestions for further work.

2 Information-Centric Networking (ICN)

There exist several ICN architectures. See [25] for a good survey that compares the most important proposals. For a more skeptical view of ICN see the survey in [28]. We have based our examination of ICN on the proposal by Van Jacobsen et. al. called Content Centric Networking (CCN) [26]. This is one of the most popular ICN architecture proposals with much available literature. There is also an active open source implementation of this architecture supervised by the NDN project [27] that we use as the basis for the implementation of our demonstrator. For the current chapter and chapter 3, 4 and 5 the discussions are done in context of the NDN architecture, thus we will refer to NDN and not ICN in the discussions. Several of the made points are general and would fit all ICN architectures but to avoid confusion with a mixture of NDN and ICN notation, we choose to specify NDN all over in these chapters.

2.1 The Named Data Networking (NDN) design

NDN is a clean slate architecture that does away with the host centric architecture of the IP protocol. In NDN the focus is on finding the information that a consumer wants to retrieve irrespective of where it is stored. This is done by addressing the information by *name* rather than IP address. The glue in the NDN architecture is chunks of information (called content chunks) as opposed to the IP protocol in the classical Internet architecture. Figure 2.1 compares the IP and NDN protocol stacks. NDN may eventually replace the IP architecture. However, in a transition phase, it can run on top of IP, encapsulating the packets in UDP/IP.

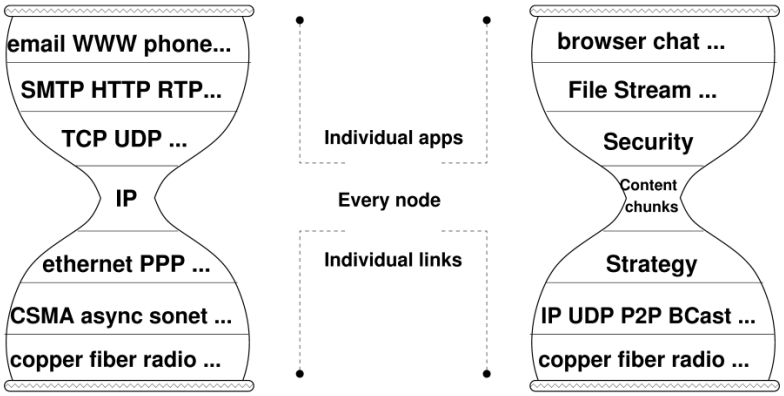


Figure 2.1 NDN exchanges the universal component of the network stack from IP to chunks of named content [26].

NDN is built on two simple basic primitives; request for information and the responding information element. In NDN the two packets that perform these primitives are called *Interest* and *Data* (Figure 2.2). This has some similarities with the pull-pattern (request and response) that is a much used communication pattern in SOA (see subsection 5.1.1 for more information).

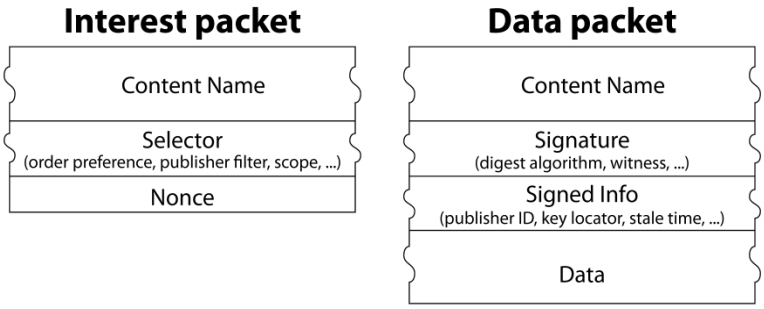


Figure 2.2 NDN packet types [26].

In NDN any information that a consumer wants is called content. In order for any content to flow in the network, the consumer must issue an *Interest* that specifies the *name* of the content that the consumer in looking for (see Figure 2.3). How the content is named must be commonly agreed upon for all consumers and producers in the information domain where the consumer is looking for content (e.g. mission network, national network, etc.). In the IP architecture we assign IP addresses and build catalog systems like DNS and CDNs to solve a similar function today. For SOA solutions the bindings between the producers, the service (*name*) and the publishing of the service is an integrated part of the architecture and works as described in Box 2.1.

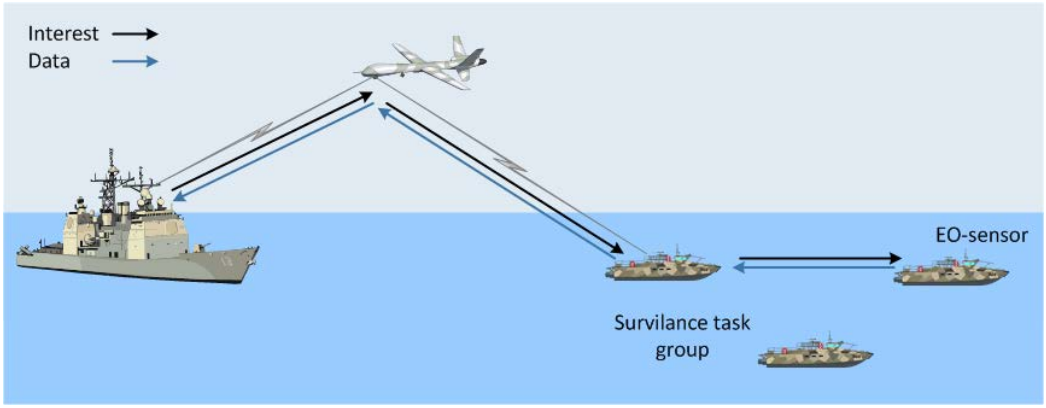


Figure 2.3 In order for the Frigate in the illustration to access the Electro Optic (EO) sensor at the surveillance vessel, an Interest must be sent from the Frigate that will result in a Data response with the EO-data from the sensor.

Box 2.1 The bindings between producers, the service and the publishing of the service in the SOA principles

SOA solutions uses the two principles loose coupling and late binding, which leads to three roles: *Provider*, *consumer* and *registry* for the purpose of discovering available services and finding the producers of the service in the network. Three operations define the interactions between these three roles; *publish*, *find* and *bind*. Interoperability between *provider*, *consumer*, and *registry* is ensured by a standardized service contract. Following these principles, we get a loose coupling between the clients and the services, with respect to both time (enabling asynchronous communication) and place (the location of both client and service can be changed without need for reconfiguration). For more information on the SOA principles, see [29, 30].

In NDN, the content name is built in a hierarchical manner and is humanly readable and looks something like this:

`Mission_network_xx/Intelligence_reports/Geographic_area_x,y/role_xx/today`

`Mission_network_xx/weather_sensor/Geographic_area_x,y/windspeed/current`

A namespace in NDN is the equivalent of an IP address space for the IP architecture and can represent the whole hierarchy of *names* or only a subset of the utilized *names*. One namespace could for example be all the *names* associated with a prefix of one of the *name* examples above:

`Mission_network_xx/Intelligence_reports/`

When the consumer has issued the *Interest* specifying the *name* of the required content, the *Interest* is being forwarded in the network in search for the content. When the content is found,

the content is wrapped in a *Data* packet and follows the reverse path of the *Interest* packet back to the consumer. The *Interest – Data* primitives are implemented with the functions described below in all network nodes (routers, consumers and producers) in the NDN architecture. A model of the functions is shown in Figure 2.4. The following description is a conceptual explanation; the order of the different functions might differ in actual implementations.

When an *Interest* is generated by the application, it is handed over to the forwarding engine of the node over an internal interface (*Face* 2 in Figure 2.4). In NDN all interfaces over which *Data* is sent or received are called *Face.* Thus a *Face* is a generic point over which to exchange packets in NDN. A *Face* can be e.g., an internal interface towards higher layers (the application), a network interface or other types of channels like a UDP or TCP connection (when NDN is run over IP).

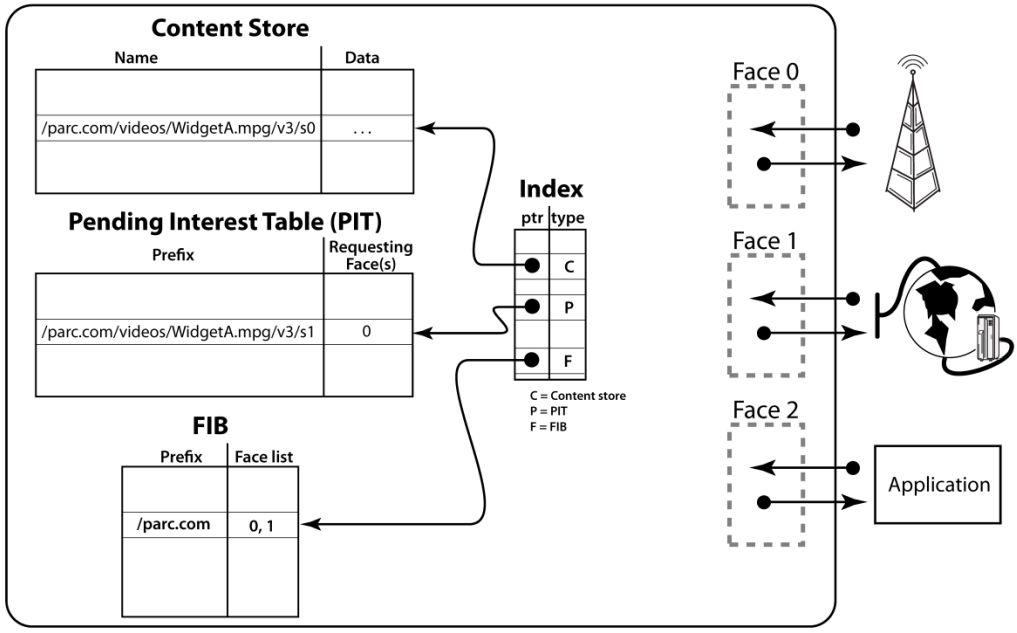


Figure 2.4 Overview of the forwarding engine in all network nodes in NDN [26].

When an *Interest* arrives from any *Face* to the forwarding engine, the engine checks if the content that is being requested is available in its own *Content Store*. The *Content Store* is a large cache that the nodes use to store copies of recent *Data* packets that have been processed by the forwarding engine (i.e. produced, consumed or forwarded by the node). If the content is not in the local *Content Store*, the engine checks if the *Interest* is already registered in its Pending Interest Table (PIT). If it is not, the interest is stored in PIT with the information over which *Face* the *Interest* arrived. If the *Interest* is already in the PIT, that means that someone else has already asked for the same content and is waiting for the *Data* response, the existing PIT entry must be updated with the incoming *Face* for the last *Interest* to make sure that the *Data* is copied to all requesting parties (*Face*(s)) upon arrival. For the case when the *Interest* is not already in the PIT, the forwarding engine checks its Forwarding Information Base (FIB) to see

which *Face(s)* to forward the *Interest* on in order to start looking for the content in the network. The FIB is similar to the routing table in IP architectures. The *Interest* is forwarded on one or several of the *Face*(s) that the FIB points at. For the case when the *Interest* was already registered in the PIT, the forwarding engine can choose to forward a new (repeated) *Interest* to improve the robustness in the search for the content or to wait for the response to the first *Interest*. This decision is made by the *strategy* for the namespace that the *Interest* is asking for content for. As seen in Figure 2.1, NDN has a *strategy* layer. The roles of the *strategies* in the *strategy* layer are described in more detail below.

The procedure above is repeated in all forwarding nodes until the *Interest* arrives at a node where the FIB points at the *Face* to the application that produces the content or there is a match in the *Content Store.* The application resolves the interest by responding with the *Data* packet. If there is a match in the *Content Store* the *Data* is retrieved from the *Content Store*. In both cases the *Data* is sent out on the *Face*(s) that is listed as incoming *Face*(s) for the relevant *Interest* in the PIT. Finally the PIT entry is deleted since the *Interest* is now resolved. The *Data* packet starts its journey back to the consumers. For all the nodes on the path; when the *Data* packet arrives on a *Face*, the forwarding engine copies the content to its local *Content Store* in order to be able to answer future *Interests* for this content from other consumers. Next the engine checks the entry of the corresponding *Interest* in the PIT to see which *Face*(s) to forward the *Data* on and deletes the PIT entry. There is no need to use the FIP for the forwarding of *Data* since this packet follows the bread crumb trail from the path taken by the *Interest*, back to the consumer(s).

The *Interest – Data* procedure of NDN is a stateful forwarding technique since the *Interest* leaves a state in the forwarding nodes that is deleted upon the arrival of the *Data* packet (or after a timeout). The stateful forwarding ensures loop-free routes since the *Interests* have a unique identifier (Nonce) and the forwarding of an *Interest* will be stopped if it arrives at a node that has seen the *Interest* before (stored in the PIT). The simple *Interest – Data* procedure is repeated for all content the consumer wants. This is a pull type architecture where the consumer pulls the content from the producer. The producer will never send any content unless it has received an *Interest* first.

If the content that is being asked for does not exist in the network or the *Interest* or *Data* packet is lost somewhere between the consumer and the producer, the PIT entries are removed upon a soft state timeout. In the NDN architecture it is the responsibility of the application to ensure that it receives the content it asked for. The network does not provide reliable delivery. If the application does not receive *Data* within a timeout period, the application will issue a new *Interest* repeatedly until it decides that the content is not there or cannot be found.

The reason why NDN can use such flexible caching approach (the local *Content Stores*) is due to its security architecture (implemented by the security layer in Figure 2.1). The primary focus of the proposed security architecture is on the integrity (and optionally confidentiality) of the content chunks. This is contrary to the IP architecture that typically secures the path that the information uses (e.g., IPSec [31] tunnel). Securing the content chunks allows NDN to avoid many of the host-based vulnerabilities that plague IP networking. In NDN the content chunk

carried by the *Data* packet is authenticated and integrity protected with a digital signature across the content chunk and the *name*. Optionally private content can also be protected with encryption. This allows the consumer to validate that the responding *Data* can be trusted to be the correct response to the *Interest*. If you are to retrieve the content from any storage (e.g., the closest available *Content Store*) this validation is crucial. The architecture relies on access to a trust chain (e.g., Public Key Infrastructure (PKI)).

In the above description of how NDN works we said that the FIB provides a list of *Faces* that the forwarding engine uses to decide how the *Interest* should be forwarded through the network in search for content. As seen in Figure 2.1, NDN has a network layer called *strategy*. The *strategy* layer allows for customized forwarding of *Interests* for different namespaces. This layer holds a series of policies and rules about forwarding *Interest* in the network. The forwarding engine asks the FIB for the list of all *Faces* that it is aware of that can lead to the requested content. The FIB is populated by one or several routing functions. Using the list of *Faces*, the *strategy* decides how to forward a specific *Interest*. Example of *strategies* can be to forward the *Interest* over one *Face* (IP-like *strategy*), to forward the *Interest* on multiple *Faces* for robustness or even to flood the *Interest* on all *Faces*. The last mentioned *strategy* does not require a routing protocol. It would also be the *strategy* that decides if an *Interest* that arrives at a node that has a pending identical *Interest* from a different consumer, should be stopped or resent for improved robustness. We will revisit the *strategy* topic and discuss efficient search for content in mobile military networks in more detail in chapter 4. Different *strategies* can be used for different content if desired.

For more details of the basic NDN functionality see [26, 27].

3 NDN for mobile military networks

With the basic principles of NDN in mind we will here underline some aspects of NDN that, in our opinion, make the architecture appealing for mobile military networks. We will also point at some of the major challenges that we think must be resolved before the architecture can be considered for use in mobile military networks.

3.1 Advantages of NDN

The NDN architecture seamlessly handles both traffic to a single consumer and traffic to a group of consumers. If the consumers are in the vicinity of each other, NDN resembles the behavior of multicast protocols that tries to minimize the number of packets that must be forwarded in a network to reach all consumers. NDN may aggregate *Interest* such that only one *Interest* per content is sent over a connection and also only one responding *Data* packet is sent

over a connection (Figure 3.1). The *Data* packet is replicated in branching points like the behavior of multicast protocols. NDN is also able to decouple the *Interest* requests and the *Data* responses in both space and time. IP multicast protocols handle simultaneous request for information to several consumers (space), but do not handle request for the same information at different times (time). NDN handles both since *Interests* that arrive at a node in the timeframe from the first *Interest* is received until the *Data* response arrives is aggregated and a single copy of the *Data* packet can be used to respond to multiple identical *Interests*. In addition, since NDN leaves a copy of the *Data* in the *Content Store* of all nodes on the path of the *Interest – Data* query, popular content is automatically made available close to the consumer even if the latter's *Interest* comes later in time than the first *Interest*.

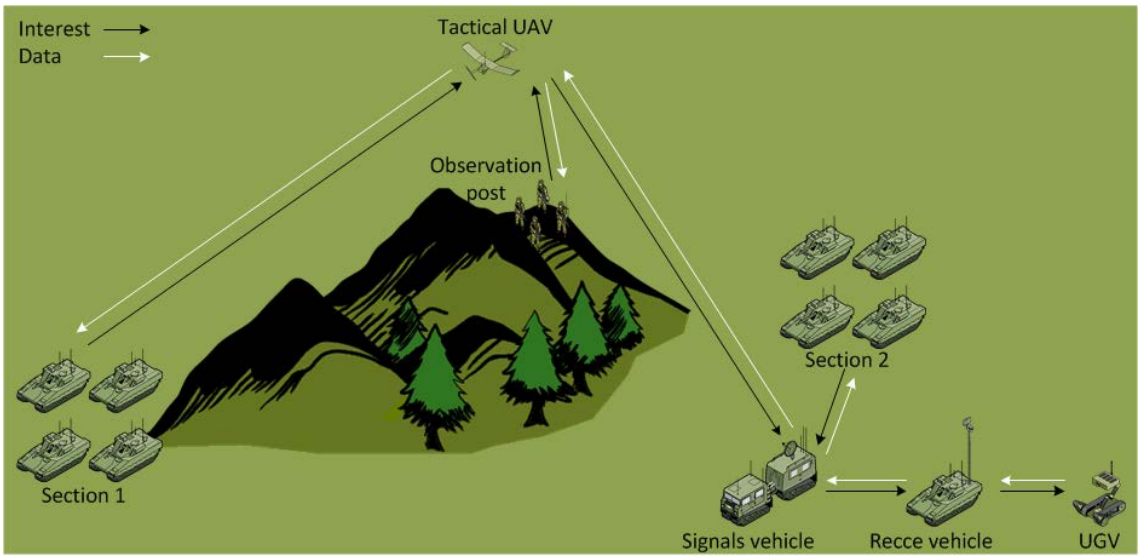


Figure 3.1 The figure shows how the Interests from three consumers (Section 1, Section 2 and the Observation post) flow towards the UGV data producer. The Interest – Data primitives of NDN behave similar to IP multicast protocols when adjacent nodes are interested in the same data. In this figure only one copy of the Interest packet and the Data packet is sent over each link.

Reliability is the responsibility of the application in the NDN architecture. The network does not guarantee reliable delivery. If the application does not receive *Data* within a defined timeout period, the application may issue a new *Interest*. Due to NDN's flexible search method and use of *Content Stores*, the second *Interest* might find the content chunk at a node in a different direction than the first search tried. An option might also be to ask the network to use another more robust *strategy* for the repeated *Interest* (e.g., forwarding the *Interest* on multiple paths).

Unlike native IP, NDN can also work in very disruptive network environments. DTN [10] is a set of protocols that has been added to IP to improve IP's behavior for very disruptive networks. NDN automatically has a DTN like interaction model that decouples the producer and consumer and is not dependent on a stable connection between the producer and the consumer. In NDN, if at some point in time the *Interest* message can get to the producer or a *Content Store* that holds

the content, the responding *Data* packet can start its way back to the consumer and will be stored at each hop. If it cannot make it all the way due to broken links, the reliability function in the application may resend the *Interest.* The second *Interest* has a higher likelihood of succeeding since the content is now likely stored in a *Content Store* closer to the consumer or a wider *Interest* search might find the content at another *Content Store*. In this way a store and forward functionality is implemented that can improve network performance, and may provide robustness in the face of disruption. In contrast, in the IP architecture the content must be fetched at the producer and thus the architecture depends on a stable connection between the producer and the consumer to access the content.

Node mobility is also a problem area that NDN is addressing. Node mobility can be split in consumer mobility and producer mobility. Consumer mobility is handled automatically by the soft state functionality of NDN. A consumer that has moved asks for new content in the standard way by issuing an *Interest*. The *Interest* builds a path from the new position of the consumer to the first visited node that holds a copy of the *Data* packet. The *Data* follows the reverse path back to the new node position (Figure 3.2). If a node moves after it has issued an *Interest* and before it has received the *Data*, then the reliability function in the application resends the *Interest* after a timeout and a new path from the new position is established. The efficiency of the search for the content from the new position of the consumer relies on the existence of routes to the producer in the FIB or the availability of good search methods in the NDN architecture (see chapter 4 for more information).

Producer mobility is more problematic. When a producer moves, the new position of the producer must either be pushed (published) to the network with a proactive routing protocol or the consumers can choose robust flooding *strategies* for the *Interest* upon multiple timeouts in order to find the new location of the producer. If the producer moves frequently this problem is similar to mobility in IP based MANET.

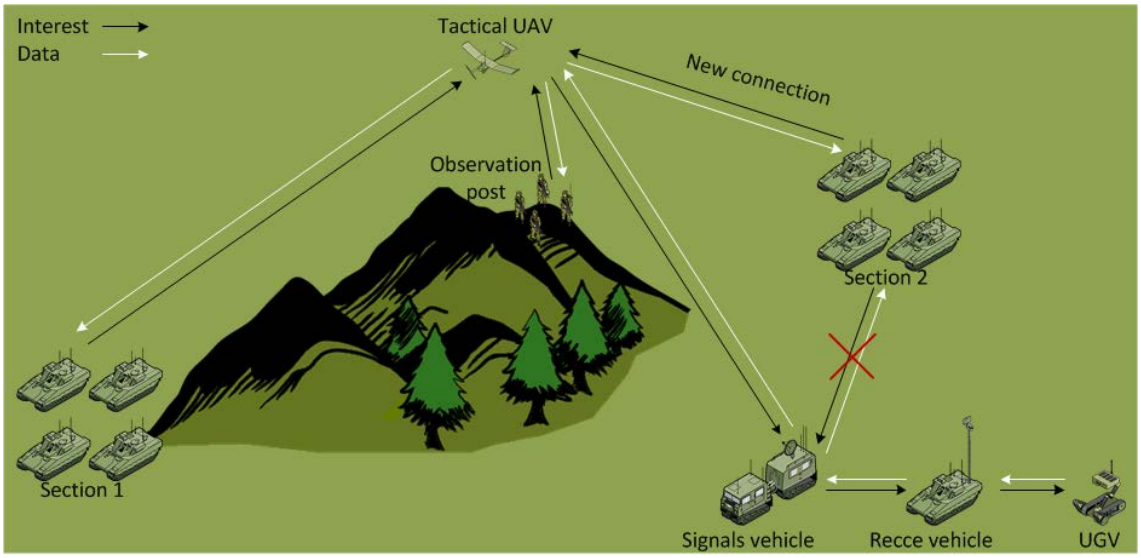


Figure 3.2 Section 2 moves and loses connection to the signals vehicle. Upon the next transmission of an Interest from Section 2, the Interest message finds the Tactical UAV and can get the Data from the UAV's Content Store.

The *Interest – Data* primitives of NDN where the consumer asks for small content chunks, make the architecture very responsive to topology changes and other disruptions. Each *Interest* can take a new route and therefore be able to avoid link breaks due to mobility, avoid network congestions, or avoid traversing links with any other specified characteristics, with short response time. In NDN a link error is detected per *Interest*/*Data* exchange, whereas in IP one typically has to wait for a relatively longer timeout in the routing protocol. By properly tuning the *Interest* transmission rate in NDN, a node can control the packet rate according to the available network resources. This gives NDN a flexible packet-rate control function that works in a distributed manner and can make local decisions. A router can for example set a limit on the number of pending *Interest* that it allows on an outgoing *Face*. Forwarding in NDN is stateful, this ensures loop-free routes. The stateful forwarding also enables NDN to perform much more intelligent forwarding decisions than IP (which is stateless). This functionality can be used to enable access control (only allow *Interest*s with certain *name*s to be forwarded), intelligent caching (cache and forward *Data* based on its priority, which can be *name*-driven), as well as robustness to disruption (cache content based on known delay/disruption on the incoming link). This functionality in combination with the *strategy* layer can support powerful traffic management in NDN.

The functions described above allow the use of a generic interface between an application and the underlying communication network. The network seamlessly supports communication to a single consumer, to multiple consumers and store and forward techniques for highly disruptive networks and node mobility. The application can use one interface for all of these functions as opposed to the IP architecture where the application must be programed to use one or the other of the range of protocols (e.g., unicast, multicast and DTN). NDN can also utilize a heterogeneous mobile network environment with multiple transmission technologies. This is

challenging for an IP environment due to lack of interoperable routing protocols and the need for stable paths from producer to consumer. NDN supports both low-overhead (and hence less robust) content forwarding, as well as robust content forwarding to ensure delivery of important content (but with higher communication overhead). Finally the purpose of NDN is to efficiently find and deliver content as opposed to providing a connection between a client and a server in the IP architecture. The NDN architecture aims to make it easier to find interesting ad hoc content in mobile military networks. This is likely to become a sought-after feature in military operations in the future where information is expected to be available all over and the challenge lies in finding the right information.

3.2 Challenges of NDN

Clearly there are a number of issues that must be studied closer in order for NDN to be useful in mobile military networks. The advantages listed above raise many functional questions that cannot be easily answered. In this section we list some of the important areas that need more attention for NDN to become mature.

Most importantly a scalable namespace that is standardized for each information domain must be in place. While it will be very hard to devise a standardized naming scheme that applies to a wide range of application domains for traditional Internet applications, we anticipate that this problem can be simpler for military networks. Military applications typically consider a significantly smaller amount of content types than Internet applications might do. Furthermore, military information is already presented in standardized formats that provide common information structures, which can be used to regroup contents and obtain content taxonomies. These properties can simplify the design of naming schemes, and also potentially increase its effectiveness since it is directly tailored to the information. NDN uses human readable *names* and a hierarchical namespace but there are proposals for binary *names*, flat namespaces, etc. A solution for an efficient namespace must be in place before NDN can be deployed in a military network.

The stateful forwarding of NDN allows for smart forwarding as well as loop-free routes at the cost of potentially very large PIT. The scalability of the PIT is a challenge and is an ongoing research topic [32]. We believe that this problem is less severe for military networks since the nodes in these networks will handle much fewer *Interests* during a given time period than nodes in the Internet. Particularly mobile military networks will not have to sustain a high rate of arriving *Interests* since the low data-rates of the transmission technologies limit the number of *Interests* that can be sent during a certain time. On the other hand the mobile military networks will suffer from many PIT entries that will not be resolved due to loss of the *Data* packet response. There is a need to study this closer to verify the scalability of the PIT on devices with little computing power.

The NDN security architecture [33] is very flexible but raises some challenges particularly for mobile military networks. In NDN information security (INFOSEC) is placed on the content. Each content chunk is signed and optionally encrypted. This is flexible but introduces overhead.

The public key of the signer, a certificate for that public key or a pointer to them must be sent with the packet and a trust chain must be in place. For low data-rate mobile connections this can be problematic. Another issue is the fact that all producers must have the ability to sign content and the hardware and software must therefore be certified for the necessary classification level. This will lead to a high increase in nodes that must be security certified. Traffic analysis is also a challenge, in NDN the *Interest* is sent in clear text. This may not be adequate for military networks [34]. The security model does not handle network security (NETSEC). There must be functions in place that perform network security. This is some of the open issues related to security. The security discussion in [28] and [35] sheds some light on some of the security challenges. Some traditional solutions for mobile military networks such as preloaded keys and link level encryption can solve some of these problems but more research is needed.

The *strategy* layer is a powerful component in the NDN architecture but must be studied and tuned to achieve its full potential. It is the *strategy* that enables the possibility to tailor dissimilar forwarding of *Interest* for different traffic types. NDN allows implementing a wide range of *strategies*, from naive ones that simply mimic standard IP communication mechanism (e.g. unicast, multicast) to more sophisticated ones that continuously analyze the *Interest – Data* exchange and self-learn how to improve the network performance. Different choices can be made that influence both the robustness of the content delivery and the cost of the content delivery; the tradeoff between efficiency and robustness. The routing functions in the network must work in close cooperation with the *strategy* layer. It is the routing functions that provide the routes the *strategy* layer can choose from. Experience from the vast research on routing in IP MANETs can to some extent be reused here for mobile military networks. The survey in [35] gives a comprehensive overview of different routing approaches for mobile NDNs. The policies of the *strategy* layer as well as the supporting routing functions are open research areas.

Since NDN is a relatively new technology, the performance tuning of the NDN architecture is an aspect that still needs to be thoroughly investigated. For instance cache replacement (*Content Store*) strategies and the timeout values (related retransmission timeout (RTO)) for *Interests* in the PID and in the application are critical to quickly recover packet losses while limiting useless retransmissions. This also influences the response time (delay and jitter) of the *Data* as well as availability (robustness). Cache replacement strategies of classic IP architectures are a mature research field and this knowledge can be reused here. More work is needed to learn how to best combine different cache replacement strategies with *Interest* timeout values and search *strategies*. This combination can be set differently for different namespaces and network types. More experience that can result in guidelines for how to set the parameters is needed. In [36] cache replacement strategies for NDN are discussed.

The NDN network must provide the necessary quality of service (QoS) for *Data* that is being fetched. This means to deliver *Data* with the required minimum delay and jitter. The mechanism to support this will be different from the solutions in IP networks. For NDN several of the functions we have discussed here must in cooperation provide QoS. The routing functions must find suitable network connections. Whether the connection is suitable (e.g., have enough

capacity) must be learned from the underlying network layers. The *strategy* layer must choose how to distribute the *Interest* message e.g., on one or several suitable network connections. If the *Interest* ought to be sent in parallel via multiple *Faces* or over a single *Face* is a tradeoff between delay in fetching the content and network resource utilization. A routing solution that also announces cached content allows the network to find both *off-path* and *on-path* caches. This can reduce delay and improve robustness at the cost of more control traffic. *On-path* cache means caches that lie on the path between the consumer and the producer and can be hit as the *Interest* is routed towards the producer, whereas an *off-path* cache lies in other paths in the network. There are several challenges associated with the potential benefit of announcing *off-path* cache (e.g., the lifetime of the content and the probability of similar content to also become available at the same *Content Store*). This must be studied further to assess the benefit. The *Interest* time-out is also a value that influences delay. QoS for NDN has not received much attention in the research community yet.

Chunk size is another research topic for NDN. How large or how small content chunks should the consumer be allowed to ask for in one *Interest*? The NDN architecture promotes tiny chunks, as small as single voice samples or video frames. The advantage of this is a very responsive network. The *Interest* can be routed a different way for each voice sample and thus be able to handle mobility and avoid network congestion very quickly. Small chunks also promote reuse since it is more likely that small content chunks are interesting to multiple consumers than large content chunks that might provide more specialized information. The small chunks come at the cost of a large overhead; an *Interest* packet and headers in the *Data* packet (including a security certificate) for each tiny chunk of content. Large chunks such as whole documents or videos reduce the overhead but also reduce some of the NDN architecture's qualities. The problem of chunk size is touched upon by the survey in [28]. For mobile military networks the problem has more aspects than the ones discussed in [28]. For important military services such as push to talk and friendly force tracking the information elements are small thus the content chunks will also be small. But for other traffic such as different types of sensor data, plans and orders etc., this is a relevant problem.

NDN uses a pull (or on-demand) service model, where the consumer starts any communication by sending a requested for content (*Interest*) and there is a 1-to-1 relationship between *Interest* and the responding *Data*. For military services it is also interesting to provide a push service model where either the consumer subscribes once to a service type and keeps receiving information until the subscription is stopped (similar to a publish/subscribe mechanism of many information infrastructures). It is also interesting for the producer to be able to push content to consumers that has not already shown interest in the produced content. Dissemination of friendly force tracking, sensor feeds and alarms are examples that can benefit from both of these traffic models. The survey in [35] points to some work that has proposed modifications to NDN to support such models. Extensions to send unsolicited content and the use of long-lived *Interests* are some approaches. This problem area is related to the previous discussion on chunk sizes. An alternative to large chunks is to subscribe to a sequence of chunks. More insight regarding the benefits or disadvantages of supporting such traffic models in the NDN framework is needed in order to evaluate the performance for mobile military networks.

In this chapter we have given an overview based on the NDN literature, of the most important characteristics of NDN that we believe make this technology interesting for mobile military networks. We've also presented a list of open research topics that must be solved for NDN to be useful in such networks. The problem that needs most attention is the definition of the namespace. For use in mobile military networks, methods that utilize the broadcast environment of the wireless transmission technologies in an efficient and robust manner to search for content and deliver *Data* is also a key challenge, as well as the security architecture. In this activity we have chosen to study closer the former of those two challenges and leave the security architecture for future work. In chapter 4 we will discuss in more details a few of the challenges based on the lessons learned while building an NDN demonstrator for mobile military networks. In chapter 5 we do a quantitative study of the efficiency of NDN's search for information compared with relevant SOA standards.

4 The search for content in mobile military networks

One of the important challenges with the use of NDN in mobile military networks is to balance the robustness and responsiveness of the network against resource utilization. How to forward the *Interests* and the responding *Data* packets is an important part of this tradeoff. This is where we chose to make an extension to NDN as part of our effort to build an NDN demonstrator for mobile military networks.

In this chapter we describe in more detail how NDN searches for content. We also explore different solutions from the literature for search in mobile networks. Thereafter we explain important functions in the NDN codebase and how these are used in the NDN architecture. This is used to explain the modifications we made to the codebase for the realization of an NDN demonstrator for mobile military networks. Finally we summarize the results of the activity.

4.1 The search for content in NDN

In this section we describe the different mechanisms that cooperate in order to find content in the network. We also review related work regarding search for content in mobile wireless networks.

NDN is designed to take advantages of distributed caching (*Content Stores*). A consumer can therefore fetch temporarily stored content from the nodes' *Content Store*. In order to find the producer or a *Content Store* with the relevant *Data*, a search process is required. The search is associated with a network cost, for instance in terms of consumed network-capacity and/or delay of fetching the content, or other cost factors. NDN has two mechanisms that support the search process; 1) the routing protocols that populates the FIB and 2) the *strategy* for how to

forward the *Interests*. In terms of routing, NDN is not limited to proactive, reactive or opportunistic routing, but can use a combination of different approaches.

For different routing schemes, the vast research on routing in IP MANETs can to some extent be reused. For the routing function it is a discussion/tradeoff if temporary content in the *Content Stores* should also be announced by the routing protocol. This tradeoff involves many parameters such as the lifetime of the cached content, the size of the cached namespace and the stability of the network. For example, if the *Content Store* contains content for very small namespaces that have a short lifetime, the signaling overhead to maintain the routing tables will be large and the routing tables will have many entries (scalability). On the other hand if the FIB can point to nearby *Content Stores* with the content that is sought after, this can improve the performance of unstable networks. Many different approaches for routing have been studied for NDN [35]. For reactive routing approaches, the routing will often be merged with the forwarding of the *Interest* which requests the content.

The other mechanism that supports the search is the *strategy*. The *strategy* decides how to forward the *Interest* based on potentially many input parameters; information from the FIB, history of earlier searches for similar content, the stability of the network, the required response time for the *Data*, etc. For wireless networks the decision to broadcast[6] or partially broadcast the *Interest* and/or the *Data* in the whole network is one important decision point. The authors of [37] group different forwarding *strategies* for the *Interest – Data* primitives for wireless common channel networks in 4 groups and discus advantages and disadvantages superficially:

1. *Interest* broadcast and *Data* broadcast. This approach is robust for packet loss and broken links due to redundant forwarding of both *Interest* and *Data*. The drawback is an excessive overhead due to many redundant packets.

2. *Interest* broadcast and *Data* unicast[7]. This approach is also robust. It has less overhead than 1) since the *Data* is not broadcast, but the overhead can still be large if the content is available at many different *Content Stores*.

3. *Interest* unicast and *Data* unicast. In this approach there is no redundant transmissions, thus this is the least robust approach but the advantage is a low overhead. This solution requires a reliable routing protocol to maintain the FIB.

4. *Interest* unicast, *Data* broadcast. This is not a useful combination. *Data* that is broadcast will be dropped by all other nodes than the ones with an entry in the PIT, thus there will not be much added redundancy due to the *Data* broadcast.

[6] The broadcast term used in NDN context means that an NDN packet is sent over multiple *Faces* and/or that the *Face* utilizes the broadcast service of the underlying transmission technology to forward the packet to the neighboring NDN nodes.
[7] The unicast term used in context of NDN means that an NDN packet is sent (cast) over a single *Face* that utilizes a unicast service of the underlying transmission technology to forward the packet to the neighbor NDN node.

Radios in mobile military networks often have omnidirectional antennas and broadcast technology is typically used to reach all nodes within transmission range of the radio. The receiving nodes rebroadcast the packet to their neighbors and in this manner the packet will eventually reach the whole network. This is often also referred to as flooding. In a mobile military network this type of broadcast (or flooding) is costly since one packet consumes much of the available data-rate as the packet is spread in the network. It is therefore of high interest to come up with different approaches for how *Interest* and *Data* are forwarded in such networks. To illustrate the complexity we use approach 1 above as an example. Approach 1 is described as very robust, but with excessive overhead. What happens is that a consumer will broadcast the *Interest* for the content. All neighbors within the transmission range of the consumer's radio range will receive the *Interest* and create a PIT entry for the relevant namespace. If the requested content is not available at any of the consumer's neighbors, the *Interest* will be re-broadcast by all neighbors. If no limitations are set, the *Interest* will in effect eventually flood the entire network. An *Interest* stops when it finds the content in a local *Content Store* or reaches the producer. Potentially many *Content Stores* as well at the producer might respond by broadcasting the *Data* along the reverse path of the *Interest*. In effect both the *Interest* and the responding *Data* (potentially many copies) are being flooded throughout the network and hence consume much network capacity. It is therefore of high importance to find good search solutions that balance the cost of searching for content with the required robustness for successful content retrieval. A set of different search *strategies* will likely be needed to support different network scenarios (level of mobility, different traffic patterns, EW threat, etc.).

The research community has done some studies on different search *strategies* that fall into the first three approaches listed above. The authors of [37] promote the use of the third approach in an Internet of Things (IoT) environment due to the robustness that an Automatic Repeat Request (ARQ) mechanism can give for a unicast type *Face*. We study NDN for a network where we expect frequent topology changes, thus for us approach 1) and 2) are better options. The literature suggests several possible alternatives for how to tailor these approaches to reduce that overhead while preserving robustness.

[38-40] follow approach 1). In [38] the Listen First Broadcast Later (LFBL) routing protocol is proposed. This protocol tries to reduce the number of transmission of both *Interest* and *Data* by building some state information that gives the node an indication if it is on a good path between the consumer and producer or not. Nodes on a good path broadcast the *Interest* and *Data* messages early. Other nodes listen for a while for others to broadcast, if no one does or the path through the node in question is likely to be better, the node broadcasts the message. In this manner there will be some redundant transmissions but a reduced broadcast storm[8].

[39] proposes to extend NDN with two extra signaling messages in order to choose a single path for the *Data* response to an *Interest*. The purpose is to reduce the number of *Data* responses that might be the result of the broadcast of *Interests*. The producer or nodes with the *Data* in the

[8] A broadcast storm represents the situation when a network is overwhelmed by continuous broadcast traffic. When different nodes broadcast data and the receiving nodes rebroadcast the data in response, this results in an overwhelming amount of traffic in the network.

Content Store send a *Reply* message to the consumer. The consumer then selects one path with a *Request* message. The path towards the consumer can be stored in the FIB for later use.

In [40] approach 1) is used in a vehicle to vehicle (V2V) context. Here the authors play with timers and distance in a similar manner as in [38], however here the solution is tailored for the V2V scenario and does not work as well in a general mobile network.

[41, 42] follow approach 2). In [41] a basic flooding of *Interest* is used to establish a sequence of unicast *Face* representing the reverse path of the flooded *Interest* from the consumer to the producer. This route is also written to the FIB. The paper does not discuss the problem of *Data* response storm in the case when the content is stored at multiple off-path caches. However the authors suggest optimizing the forwarding of *Data* by utilizing a multicast *Face* to transmit one *Data* message to downstream consumers in situations when multiple neighbors have issued *Interests*.

[42] proposes Neighborhood-Aware Interest Forwarding (NAIF). This is an opportunistic protocol that aims to reduce the number of redundant *Interests* as a function of how many *Data* responses it overhears. The idea is that the network will be self tuned to flood the *Interest* just redundantly enough to get enough *Data* responses for the necessary robustness.

In [43] several models are compared. One flooding solution (approach 2) is compared with a unicast solution (approach 3) and a solution doing routing based on Geographic Hash Tables (GHT) [44]. Modifications are proposed to the CCN design that was the predecessor to NDN. The authors recommend different techniques for different network characteristics: For example, they recommend flooding for small networks with unpopular content and where it is unpredictable where the content is stored and recommend GHT when it is more predictable where the content is stored and with uniform content popularity.

Approach 2) is also the approach that we decided to follow for our implementation of a wireless search for the NDN demonstrator for a mobile military network. Approach 2 provides a basic robust method that can be optimized in many different ways.

4.2 The wireless demonstrator

We used the codebase available at [27] as the basis for the NDN demonstrator that we built in this activity. In the following we describe in more detail some of the functions of NDN that are important in order to understand the challenges that appear when NDN is applied to wireless networks. We also do a more detailed discussion of some of the challenges with NDN in such networks than what the brief overview in chapter 2 allowed. Finally we describe the extension that we made to the NDN codebase in order to support wireless technologies for our demonstrator. Our extension is named a wireless *Face*.

4.2.1 NDN Routing

The NDN implementation contains the Named Data Link State Routing Protocol (NLSR) [45]. In short, it is based on the well-known IP routing method Open Shortest Path First (OSPF) [46]. OSPF is a link state protocol announcing end points by IP address-ranges for the IP networks. NLSR uses a similar method for announcing namespaces. Since it is based on OPSF and OSPF is optimized for wired environments, it is inefficient in mobile environments, but more than sufficient for smaller scale demos and prototypes.

For future work an interesting extension to the current codebase would be to adapt the Optimized Link State Routing (OLSR) protocol [47] for used in NDN. This would give NDN both a unicast routing functionality that would populate the FIB as well as the possibility to do optimized flooding of *Interests*.

Interest and *Data* forwarding in NDN is per packet, while namespace announcing and routing is refreshed with a certain update-rate that can be configured according to how much network resources the routing function is allowed to consume. For mobile military networks, the entries in the FIB will therefore in many cases not be correct. To mitigate this problem the *strategy* can opt to apply a search method with redundant *Interest* packets that simultaneously search in different directions for the requested content. Thus an *Interest* can be forwarded based on unicast, broadcast or something in between. One benefit of *Interest* broadcast is that no routing protocol is needed; only a default route entry in the FIB that points at a broadcast *Face*. Unicast on the other hand requires a routing protocol to populate the FIB. As explained in section 4.1, this broadcast search can consume much network resources

One simple and efficient method that combines a robust search with a more efficient routing solution is to initially broadcast new *Interests* and feed routes that are being learned as the *Data* makes its way along the breadcrumb trail from the producer to the consumer, into FIB. This method shares many similarities with the Ad hoc On-Demand Distance Vector (AODV) routing protocol [48] for IP networks. The benefit of this approach is that only the first *Interest* for a namespace must be broadcast, while consecutive *Interest* for the same namespace can be routed based on entries in the FIB. If an *Interest* that follows the entries in the FIB fails to find the content, a second search can broadcast the *Interest* for a wide search for the content and simultaneously update the FIB with more recent routing information. This method is simple and works well for fairly stable networks and where the namespace is compact. By compact we mean that consecutive *Interests* request content within the same namespace prefix.

Another central problem regarding the NDN routing and *strategy* for mobile military networks is how to treat temporary content in the *Content Stores*. The question is whether the temporary content should be announced by the routing protocols and the path included in the FIB, or not. If we choose not to keep track of routes to content in the *Content Stores*, the risk is that an *Interest* will be routed towards the producer even though cached versions are available in *Content Stores* closer by. However, keeping routes to cached content updated will be very resource consuming. Cached content can be long-lived or short-lived and can come and go at many different *Content Stores*. This will produce a high routing overhead. Interesting approaches to explore in order to

be able to find content close by is to announce cached data locally. One approach is to broadcast periodically the *names* of long lived content to one-hop neighbors. This gives an indication for what content the neighbor might have in its *Content Store*. The overhead for this method is fairly low. Another option is to broadcast *Interests* with a local scope (i.e. with a low limit on how many hops the *Interest* is allowed to propagate). This could be performed similar to AODV's expanding ring search, and hence allow *Interest* to initially be broadcasted one hop (or a few hops). If no *Data* match is found at nearby neighbors, the *Interest* should be forwarded according to the entries in the FIB.

4.2.2 The NDN *Face*

As shown in Figure 2.4, NDN uses *Faces* and the concept of channels for communication. A *Face* is used for forwarding of *Interest* and *Data*. An analog in the Internet architecture would be a tunnel. To accommodate changes in topology, *Faces* must be established dynamically. A channel is used for this, and it establishes a *Face* in the same manner as the listen() [49] call in Linux. There is a channel for each protocol family of *Faces*. The channel listens for incoming packets belonging to a specific protocol family. When a packet arrives at the channel it establishes a *Face* towards the sender of the packet. The incoming packet is tagged with the newly established *Face* as the incoming *Face* in the state-full forwarding used by NDN. The *Interest* that arrived on the cannel is next sent to the forwarding engine of the NDN node for further processing.

Multiple *Faces* can be mapped to a physical interface. A *Face* is similar to a socket in terms of functionality. Standard *Faces* in the NDN codebase are:

- Multicast *Face* (the multicast *Face* assumes that the multicast distribution tree/mesh is established out of band).
- Raw broadcast *Face* (broadcast interface to a directly attached network)
- Raw Ethernet *Face* (Ethernet interface to a directly attached network)
- Interprosess communication (e.g. between the forwarding daemon and local applications that produce content)

And for the case when NDN is installed over IP, additional IP transport *Faces*:

- UDP *Faces* bound to particular end point tuples (IP address and port)
- TCP *Faces* bound to particular end point tuples (IP address and port)

The properties of a *Face* depend on the context NDN is used in. As already mentioned, the multicast *Face* depends on how the distribution tree is established. UDP and TCP *Faces* assume that NDN is installed on top of IP and depend on whether IP-routing is enabled or not. If routing is enabled all IP based *Faces* may look like they are one hop apart from the viewpoint of the

NDN framework. The IP routing will make sure that the packets are forwarded correctly via multiple hops to the receiving *Face*. This route is invisible to the NDN node. If IP routing is disabled NDN must handle the routing and a routing mechanism is needed to populate NDN's FIB to support the routing of the *Interests* in the network.

Internally, NDN uses three abstractions in the design of a *Face*. Each *Face* has an NDN-transport service and an NDN-link service. In addition there is a service representing technology dependent functions. The link service provides higher layer functionality like packet fragmentation and reassembly, and error detection. The transport service is a wrapper to the particular network service used to implement the *Face* (see Figure 4.1).

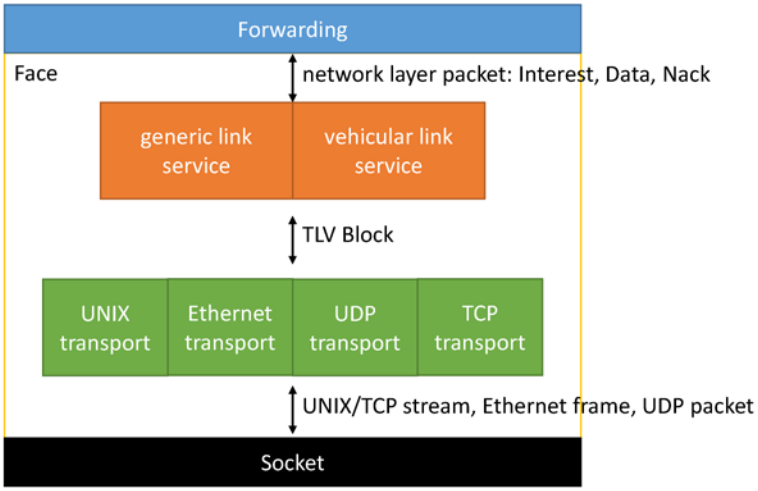


Figure 4.1 The figure shows the NDN transport and link layers. If the forwarding element is an IP network there will be additional IP transport and link layers [50].

For our demonstrator we wanted to include a wireless *Face*, a *Face* that assumes a common channel radio network as the transmission technology. The purpose of this wireless *Face* was to improve the NDN performance for such network types. The wireless *Face* should give efficient support in mobile military networks for the search procedure utilized to discover and fetch content.

4.2.3 Forwarding of *Interests*

The forwarding in NDN is stateful as described in section 2.1. The *Interests* are routed to perform the search for the content, while the responding *Data* follows the breadcrumb trail of the *Interest* that requested the *Data*. The *strategy* decides how to search for content utilizing different inputs. One *strategy* can be bound to one or more namespaces with a default *strategy* bound to the top level of the naming hierarchy (similarly as a default route in the IP architecture). A namespace is similar to a network prefix in the IP architecture. The namespace represents a subset of the *name* structure. NDN enables to route *Interest* based on different namespaces, and hence different network service can be given to different applications.

The code that implements the forwarding of *Data* and *Interest* in NDN is a pipeline with a small number of entry points with calls to functions in the *strategy* layer. Only *strategy* functions valid for the namespace in question are relevant. The entry points are 1) after an *Interest* is received, 2) before an *Interest* is fulfilled and the node is ready to respond with *Data*, 3) after a *NACK*[9] is received and 4) before an *Interest* expires. Any new *strategy* must contain routines for these entry points, but the default implementation is *null* (see Figure 4.2 for an overview of the modules).

Entry point 1) that happens after an *Interest* is received, is where routing of an *Interest* is performed. This function will have access to the FIB data structure. The FIB data structure is protected and can during runtime only be updated by authenticated protocol messages or authorized node managers. The FIB data structure is well documented in the Named Data Networking Forwarding Daemon (NFD) developers guide [51]. The *strategies* already implemented in NFD provide good examples on the structure and expected functionality at the different entry points. Conceptually the function performed at entry point 1) is similar to how routing is done in Linux. There is a forwarding table containing the *Face(s)* for the next hop towards the content for different namespaces. Each routing protocol has its own data structure and builds its own routing table and the entries from that routing protocol is merged with entries from other routing protocols into the FIB. However, to our knowledge, NFD lacks the concept of administrative distance[10] to reconcile different forwarding *Faces* for the same namespace.

If we in our future work opt to implement a solution to push unsolicited content in the NDN network, it will likely be necessary to modify the existing pipeline in the *strategy* layer. The current NDN implementation rejects unsolicited content unless it is received on a local *Face*.

[9] The *NACK* is a local message that can come from the NDN-link layer if that layer for some reason is not able to send the *Interest* or *Data* packet.

[10] Administrative distance is a number used by routers to rank routes from most preferred (low administrative distance value) to least preferred (high administrative distance value). When multiple paths to the same destination are available in its routing table, the router uses the route with the lowest administrative distance.

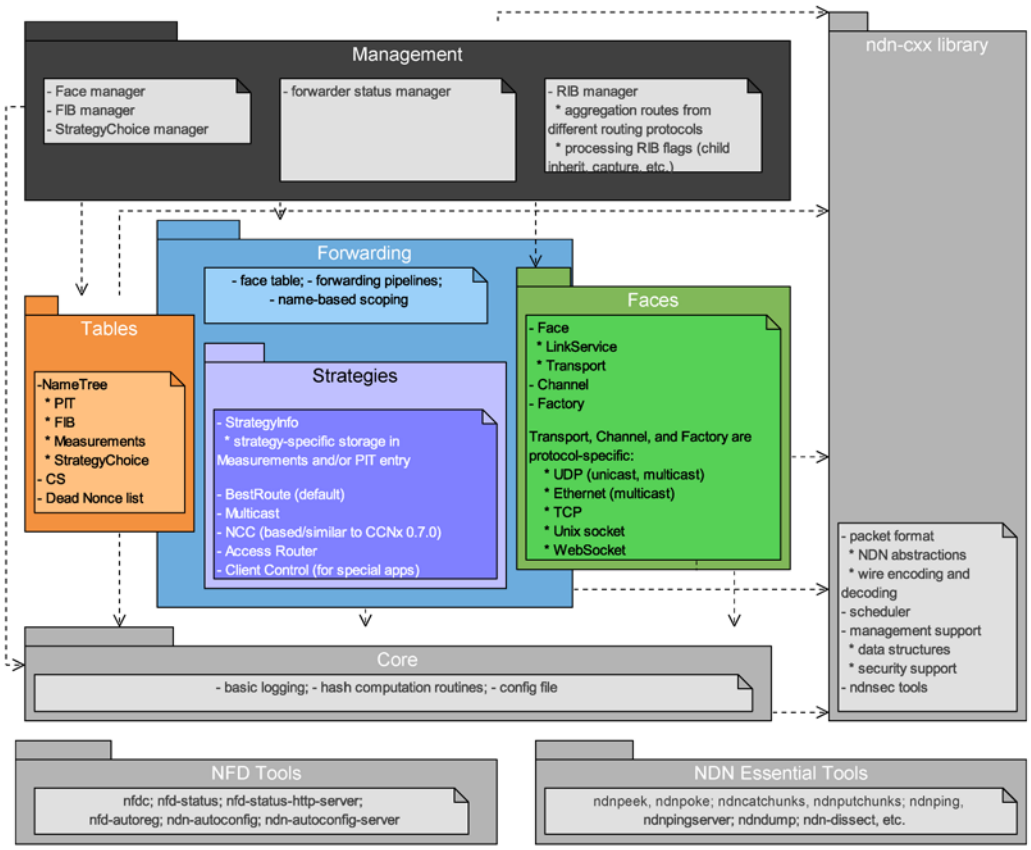


Figure 4.2 Overview of Networking Forwarding Daemon (NFD) modules[50].

The *strategy* pipeline is also the place to extend the code with functionality for smart forwarding of *Interests* based on the history of previous *Interests* and corresponding *Data*. This is also a very relevant activity for future work. Since both *Interest* and *Data* are available through the different entry points, the necessary information is accessible here and necessary metadata can be maintained in a cache of hints.

4.2.4 Wireless *Face*

In the NDN codebase there are three alternative ways of establishing *Faces* over a wireless technology. If the wireless interface is treated as an Ethernet interface all standard *Faces* (see section 4.2.1) can be used. However, the UDP and TCP *Faces* do not utilize the broadcast environment that is available with wireless technologies. It is therefore interesting to look for other alternatives. The two other existing alternatives are multicast *Face* and raw broadcast *Face*. The latter two represent non-optimized versions of approach 1) described in section 4.1. As shown, this method will consume much network resources with flooding of both *Interest* and the responding *Data*.

In order to extend the existing codebase with improved functionality for mobile military networks, we chose to implement a new *Face* called a wireless *Face*. Our solution was a first

step towards an environment where the robustness of the *Interest – Data* forwarding can be better controlled. We chose to implement a solution that follows approach 2) in section 4.1 i.e. to broadcast the *Interest* and unicast the responding *Data* packet. What happens is that an *Interest* is broadcast on the transmission technology and arrives at the wireless channel of all one hop neighbors. Upon the reception of the *Interest* each of the neighbors establishes a unicast UDP *Face* pointing pack to the one-hop neighbor that broadcasts the *Interest*. The *Interest* is next stored in the PIT with the UDP *Face* as the *Face* to be used to forward the responding *Data* on. This is done hop by hop until the *Interest* finds the producer or a content match in a *Content Store*.

If the content item is popular, the broadcast of the *Interest* will find multiple copies resulting in a storm of *Data* back. There are several ways that this problem can be mitigated. One alternative is to utilize the expanding ring search method in AODV. Another alternative is to borrow methods for avoiding response storms used in reliable multicast protocols such as introducing a random delay when a node respond with *Data* to an *Interest*. Only if the node does not overhear a response before it is scheduled to respond, will it send the *Data*. The work in [38, 40, 42] uses similar approaches.

Both of the mentioned optimization methods can be used in conjunction with each other and both can use stochastic forwarding of *Interest* and *Data*. The flooding of the *Interest* can also be optimized by use of mechanisms that utilize Connected Dominating Set (CDT) such as the Multi Point Relay (MPR) function of OLSR to minimize the number of broadcasts needed to flood the *Interest* in a network. Another optimization would be to store the route taken by the *Data* in FIB and use this for unicast routing of succeeding *Interests*. We chose to leave all of these optimizations for future work to keep the code as simple as possible for this first activity.

4.2.5 Implementation

In order to implement the wireless *Face* functionality in the NDN codebase several approaches were tried.

The initial and simplest approach was to add functionality to the processing of an *Interest*. The data structure of the *Interest* could easily be modified to include the identification of the node sending the broadcast *Interest* packet. The serialization and de-serialization functions could also easily be modified to accommodate this. However, the updating of other necessary data structures (e.g. the PIT) was harder since these were part of the *Face* structure. After a short time, this approach was abandoned.

The alternative was to develop a new wireless channel and wireless *Face* within the existing structure of link and transport services. Conceptually, the model was the UDP channel and its associated *Face*. The UDP channel is set-up as part of the initialization of an NDN-node. The receiver and sender have different implementations. The sender of an *Interest* will send to a channel. The sending *Face* will wrap the *Interest* in a UDP packet. The receiver has a standard UDP listen socket. When it receives a UDP packet, it establishes a UDP *Face* with the destination address of the sender's UDP address.

Our implementation of the wireless channel and wireless *Face* reuses building blocks from the UDP channel and the multicast *Face*. The wireless channels use a multicast socket with a known preconfigured address. When an *Interest* is received on a listen socket at the receiving node's channel (node B), a unicast UDP *Face* is established back to the sender of the *Interest* (Node A). The sender (Node A) will not have a UDP *Face* towards the receiver. However, when the receiver (Node B) sends the responding *Data* back over the established UDP *Face*, the UDP channel at the sender of the *Interest* (Node A) will establish the UDP *Face* towards Node B. In addition, there are some bookkeeping of data structures that was implemented, since the newly established *Face* in Node B must be used as the stored incoming *Face*, for the *Interest* in the PIT.

Since the wireless channels must be established at the initiation of the NDN node, the configuration section must be extended with the new abstraction.

The advantage of this implementation approach was that to a large extent existing code could be copied, modified and combined. The disadvantage is that multicast and unicast are different and have different abstractions in Linux. Consequently, unnecessary sockets were established and abandoned.

4.2.6 Lessons learned

NDN represents a clean slate functionality. It is difficult to grasp all implications of the new architecture (e.g., namespace routing and caching in *Content Stores*) through merely literature studies. The authors are all schooled in the classical IP architecture and thus much of our thoughts and ideas were affected by concepts from the IP architecture. The effort to build an NDN demonstrator and further extend the codebase with new functionality allowed us to get a much better understanding of the concept and implications when trying the concept on mobile military networks.

The NDN code uses templates and libraries extensively. It is therefore necessary to understand the base classes and be familiar with the library functions first. This was perhaps the largest problem we had in understanding the code.

The developers guide [50] is dense. A substantial part of the information was therefore overlooked during the first passes through the guide. Understanding the guide down to sentence level is well worth the effort.

During the development and testing the codebase changed. These changes were not always correctly handled by the build system. Once we detected the problem, raw reinstallation of all modules solved the problems.

The NDN code is well structured. In particular, the forwarding pipeline (see section 4.2.3) with the entry points makes it fairly easy to implement new *strategies* and modify the forwarding behavior. The implementation of managers, *Faces*, and core function requires more effort to understand.

4.3 Summary— the search for content in mobile military networks

We have successfully built an NDN demonstrator for mobile military networks. The NDN codebase has been extended with our first effort to make a wireless *Face* for NDN that targets the tradeoff between resource utilization and the robustness and responsiveness of the network. The wireless *Face* provides a starting point that can be used as a stepping stone for further studies and improvements. Modifying the *Interest* flooding with expanding ring search and optimizing *Interest* flooding by using extended dominated sets are examples of two very interesting approaches for future work.

The effort of building a demonstrator has given us insight into the workings of the NDN architecture that we would not have gained from solely doing literature studies. The new insight strengthens the understanding we had prior to starting the project, that NDN have potential to solve many problems associated with mobile military networks. However there are unresolved challenges associated with the NDN architecture (see section 3.2) that must be solved before this architecture can be fielded.

5 NDN's support for the information infrastructure

In this chapter we report from the activity that studied how well NDN can support the information infrastructure for mobile military networks. First we introduce central SOA terminology, and the standards and approaches that our work is based on. Then we discuss the experiment setup and results and finally we summarize the results of the activity

Both NNEC and the early spirals of FMN focus on interoperability for fixed network infrastructures and deployed semi static installations, where network resources are abundant. Therefore, the standards recommended for implementing the various core services (see Figure 1.1) in the information infrastructure were chosen merely based on their suitability as a federation mechanism and not on their ability to scale to network infrastructures with low data-rates. NATO has chosen the OASIS standard WS-Notification [52] for publish/subscribe in its SOA baseline [53]. WS-Notification is a part of the family of SOAP Web services standards. SOAP services promote interoperability, but being based on XML the cost is increased overhead compared to other protocols. Typically, it performs well if you have a network with 1 Mbps throughput or more, which has been shown in previous experiments [54, 55]. For resource constrained networks, on the other hand, other solutions must be applied.

In this activity we perform a small-scale comparative evaluation of overhead of WS-Notification with another publish/subscribe standard: Message Queuing Telemetry Transport (MQTT) [56]. We measure how these standards compare to the novel approach of NDN under the same networking conditions. Though fundamentally different, these approaches can all be

used to realize the SOA paradigm. The purpose of the activity is to get a better understanding of how these approaches can perform for mobile military networks which are typically resource constrained networks.

Recent work on publish/subscribe in federated networks has shown that it is feasible to mediate between several different publish/subscribe standards using a gateway approach. In [41] the authors describe and evaluate a multi-protocol broker implementation that is able to translate between WS-Notification and several other protocols. Hence, it supports the work we did here, since it shows that even if we here recommend to use something else than NATO's protocol of choice in mobile military networks, it is doable to provide the same data to others using WS-Notification through mediation gateways.

In [57] Carzaniga et al. compare NDN with publish/subscribe methods and observe that the traditional IP publish/subscribe paradigm and ICN are optimized for two different traffic types. ICN targets long lived data, while for short lived data the authors argue that publish/subscribe is a better approach. Both short lived and long lived data should be supported by the network. In their work, Carzaniga et al. propose a common content-based network layer that supports both request/response content delivery and publish/subscribe event notification. By using one common content-based network layer, both publish/subscribe and on-demand content delivery can share forwarding tables and hence, require only a single routing infrastructure. In our activity we have chosen to use NDN also for short lived data and leave possible optimizations for different traffic types for future work.

5.1 Service-Oriented Architecture (SOA)

SOA is a paradigm giving an approach to building loosely coupled distributed systems. As such, it does not prescribe any specific technology for its implementation. However, to ensure interoperability, one important SOA principle that is condoned by NATO is that of using open standards when possible. The OASIS SOA reference model provides the SOA definition that we use in this report [58]:

"SOA is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations."

The central concept in a SOA is the service, which [58] defines as:

"A service is a mechanism to enable access to a set of one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. A service is provided by one entity – the service provider – for use by others, but the eventual consumers of the service may not be known to the service provider and may demonstrate uses of the service beyond the scope originally conceived by the provider."

In this chapter we focus on SOA realized using different approaches. This supports SOA principles, which state that SOA can be implemented using different technologies. Naturally, we investigate WS-Notification, which is NATO's choice [53].

We also include MQTT, which was found to have promising properties in an earlier experiment [54]. MQTT is a light-weight approach to publish/subscribe compared to WS-Notification. Finally, as an alternative to publish/subscribe, we include the novel concept of NDN to see how this compares to the two industry standards.

5.1.1 Publish/subscribe

Publish/subscribe [59] is a term used to describe a communication pattern where clients that require information set up a subscription indicating the type of information they need. Setting up a subscription may be done using topics (or keywords), content filters or both. Once a subscription has been set up and new information becomes available, then the data is pushed to the interested client(s) based on the active subscriptions. The data is sent either directly by the information producer or via a broker, an approach which offloads producers from the task of doing both subscription management and notification dissemination.

Because publish/subscribe can efficiently support the requirement for on-demand information distribution, it has been identified as part of the Message-Oriented Middleware Services in NATO's C3 Taxonomy [60]. Interoperable core services is an important enabler in the FMN concept, which aims to serve as a common network platform for nations working together on common missions. A drawback of broker-based publish/subscribe is that the broker typically constitutes a single point of failure. To successfully leverage publish/subscribe in a tactical environment, one would need to mitigate this single point of failure either by making a multi-broker setup or by using an approach that does not rely on brokers.

5.1.2 WS-Notification

WS-Notification is a set of three standards from OASIS: WS-BaseNotification [52], which gives an approach to publish/subscribe and defines the basic message exchange and associated roles and formats. WS-BrokeredNotification [61] extends WS-BaseNotification with support for brokered publish/subscribe, whereas WS-Topics [62] defines different approaches to structuring the topics used for subscriptions and how to parse and interpret them (WS-BaseNotification has only a simple string-based topic expression called Simple Topic).

All three standards are included in the NATO messaging core service. Therefore, we investigate WS-Notification as extended by WS-BrokeredNotification rather than the more basic WS-BaseNotification.

5.1.3 MQTT

MQTT is also an OASIS standard [56]. It has recently become popular as a light-weight approach to publish/subscribe in the commercial sector. It is often the protocol of choice for

reliable publish/subscribe in smart devices (e.g., Android phones), and is much used for Internet of Things (IoT) applications.

Just like WS-Notification it is broker-based. However, MQTT is built directly on TCP, thus doing away with some of the inherent overhead of WS-Notification since it doesn't use HTTP and SOAP. Since this means that the protocol inherently has less overhead than WS-Notification, MQTT was recommended by the NATO IST-118 "SOA Recommendations for Disadvantaged Grids in the Tactical Domain" research task group [24] for a closer study of its performance in mobile military networks.

5.1.4 Architectural discussion

5.1.4.1 Push versus pull

In traditional publish/subscribe solutions, data is pushed to the subscribers soon after its creation. This is in contrast to NDN, where an *Interest* for the *Data* must be received before the *Data* is transmitted.

Different military applications have dissimilar network requirements. Time critical traffic will, in most situations, benefit from a push design similar to a publish/subscribe solution. A pull design, here represented with NDN, is more suitable for data that does not have strict timeliness constraints. However, important military applications for the tactical edge such as friendly force tracking often have periodic characteristics and can successfully be served by both architectures.

NDN has three aspects that affect the responsiveness of the architecture. 1) How often a service consumer issues its *Interest* messages. 2) The *Interest's* timeout and data freshness requirements can be configured. 3) The last configuration is the *Data* validity time, which is set by the provider.

Similarly, publish/subscribe protocols can handle different lifetime requirements. The exact mechanisms supported will vary from protocol to protocol, but common features include notification frequency, notification timeout and subscription lifetime configuration. Hence, both publish/subscribe and NDN can be configured to handle a range of different lifetime configurations.

5.1.4.2 Central broker versus distributed solution

NDN differs from the publish/subscribe protocols we consider, as these offer brokered communication. A broker-less architecture is beneficial in terms of avoiding a single point of failure or the overhead of building and maintaining an overlay network between brokers to improve the robustness of broker failure. That said, there exist experimental publish/subscribe approaches that are broker-less. But, here we consider only standards for publish/subscribe.

5.1.4.3 Multicast versus unicast

In the standard publish/subscribe protocols the provider unicasts data to the broker, which then unicasts the data to each subscriber. Hence, many of the unicast connections will transfer the same data multiple times over the same link. NDN is more scalable than publish/subscribe solutions with such an architecture. NDN inherently provides many of the traditional multicast features by caching the data. For the validity of an information object, the data is sent only once over a specific link. This is not possible using, e.g., MQTT or WS-Notification without breaking standard compliance. Consequently, the publish/subscribe architecture does not scale by the number of subscribers unless it takes advantage of multiple brokers or includes multicast support.

5.1.5 NATO Friendly Force Information

In our experiment we use the NATO Friendly Force Information (NFFI) data format in our services. The dissemination mechanisms discussed above provide the functionality necessary to distribute information from a provider to the interested consumers. The reason for choosing the NFFI data format (described in draft STANAG 5527) is that it has been used with great success in many contexts, after it originally emerged to support interoperable friendly force tracking in the Afghan Mission Network (AMN). It is therefore a good example of a representative standard payload for the data dissemination comparison experiment.

5.2 Experiment setup

Our goal is to compare inherent protocol approach overhead in order to establish an information infrastructure with efficient information dissemination and a robust solution. NATO's specification for publish/subscribe in the SOA baseline [53] prescribes using WS-Notification. However, another solution could be used at the tactical level if it proves to be more efficient. The solution at the tactical level could be bridged with WS-Notification at higher echelons. This is easy to do through gateways, as has been shown for publish/subscribe [63]. For the sake of these experiments we have one design (that of SOA using the publish/subscribe pattern) but we have three different implementations.

We expect other alternatives than WS-Notification to be more resource efficient (and possibly also more robust) so we test our implementation in a comparative small-scale evaluation and discuss how our findings may later apply to a larger scale deployment. In pursuit of these objectives we use stock implementations of software where applicable, leaving optimizations and further adjustments for future work.

The experiment includes four nodes, two service consumers, one service provider and one broker/router (see Figure 5.1). Each node is a Raspberry Pi 3 (RPi3), which was chosen to reflect an example of small form factor, off-the-shelf equipment. Such nodes are likely to be used on vehicles or carried by dismounted personnel to realize capable yet light-weight SOA service providers and consumers, for example using MQTT. In fact, such deployment is also in

key with architectural suggestions that is pursued on integrating civilian IoT aspects with military systems to increase situational awareness [8, 64].

All four RPi3 nodes were set up with Raspbian OS, and Oracle Java 1.8 was installed. The WS-Notification broker and MQTT broker were both installed on the same node. NDN was installed on all nodes.

We used a closed-source implementation of WS-Notification. However, this implementation has been tested for interoperability at the NATO Coalition Warrior Interoperability eXercise (CWIX) in 2014, where it was shown that the functions used (subscribing to a topic, publishing to a topic, and notifying the subscribers of new data) in our experiments are indeed compliant with the standard [65].

For MQTT we used the open source mosquitto broker which is freely available [66]. For NDN we used the ndncxx implementation which is also open source and freely available [67].

5.2.1 Providers and consumers

All providers offered NFFI v1.3 data. All consumers received NFFI v1.3 data. Since our main goal was to compare inherent protocol approach overhead, we used a “perfect” (i.e., wired, high-capacity with no packet loss) network in our tests. By doing this, we can identify which approach(es) that seem viable and should be further investigated with respect to deployment in mobile military networks. We captured all network traffic in the central broker node, analyzing it with respect to the number of packets sent by the different solutions.

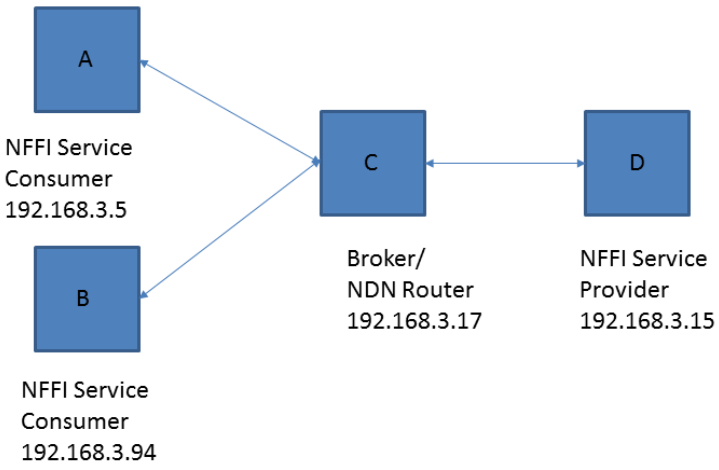


Figure 5.1 Testbed.

5.2.1.1 WS-Notification

The WS-Notification provider and consumers used the closed source counterpart of the above mentioned WS-Notification broker. The provider was set up to publish an NFFI track every 10

seconds. Upon start, each consumer would set up a subscription for the topic string "/nffi" to the broker. The consumer is single threaded, thus blocking and waiting to receive a message until it arrives.

5.2.1.2 MQTT

The MQTT provider and consumers were implemented using the mqtt-client-1.7-uber library [68]. Here, the provider was set up to publish an NFFI track every 10 seconds. Just like for WS-Notification above, the consumers set up a subscription to the topic "/nffi" and wait to receive data, only in this case the MQTT broker is the central part involved.

5.2.1.3 NDN

The NDN provider and consumers were implemented using the jndn library [69]. The NDN implementation realizes what can be perceived as a hybrid approach between publish/subscribe and request/response. Even though data that is produced by the provider has a limited lifetime (in our tests set to 5 seconds), it is not sent unsolicited to the network. Instead, the provider registers with the network that it may fulfill the *Interest* "/nffi", and awaits the arrival of an *Interest*. The consumers issue an *Interest* for "/nffi" every 10 seconds, this *Interest* is either propagated through the network to the provider, which then responds to the request, or it is served by an intermediate route that has a valid (not expired) cached copy in its *Content Store* of the *Data*. Prior to the tests, the nodes were configured with UDP *Faces* (the NDN equivalent of ports) and static routes to ensure that all traffic was forced through the central node (the broker for the publish/subscribe solutions). This was done to ensure that all solutions used the same route. It enabled us to capture the network traffic in a uniform way across the three experiments.

Table 5.1 Number of packets and bytes generated in the network.

Technique	# packets	Bytes transmitted	Bytes per packet
NDN	476	319872	672
MQTT	645	242968	377
WS-Notification	2721	457621	168

5.3 Experiment results

In this section our results are analyzed and discussed. It is important to note that the analysis is based on using the specific implementations mentioned above.

Table 5.1 shows the results for our small testbed shown in Figure 5.1. Three different methods for sharing NFFI messages were evaluated. We saw that there was a large difference in the number of packets and bytes transmitted by the different methods. The NATO standard for publish/subscribe, WS-Notification, required almost six times as many packets as NDN. The

main reason for this is that WS-Notification is built on top of HTTP and TCP. Both HTTP and TCP add additional packets to the WS-Notification flow. Furthermore, WS-Notification initiates and stops the TCP connection for each publish/subscribe message being sent. That is, the connection between any subscriber and the broker and between the broker and the publisher is not kept up, but starts and stops for each message. The result is an abundance of small packets on the network. Figure 5.2 shows a packet sequence diagram for WS-Notification. Note that this is not all a result of the WS-Notification standard, much of this behavior is dependent on the underlying HTTP library used to realize the standard.

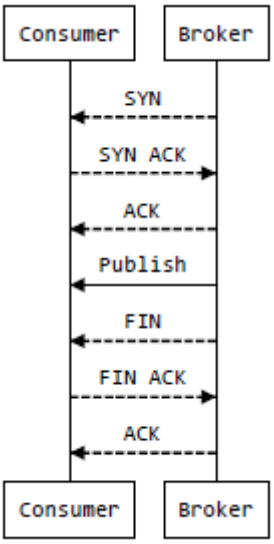


Figure 5.2 Packet flow between the broker and consumer nodes using WS-Notification. Each request establishes a new TCP session, and hence many TCP packets are required to set up and tear down connections. Data passing from the provider to the broker (which occurs prior to the sequence shown above) involves the same procedure.

A similar message sequence diagram is shown in Figure 5.3. Contrary to WS-Notification, MQTT holds the TCP connection open and consequently does not need to reestablish a new TCP connection for each message exchange. Furthermore, MQTT avoids the inherent overhead of WS-Notification since it does not need HTTP and SOAP.

The result is that fewer packets and fewer bytes are required to exchange messages. MQTT sends periodic keepalive packets to keep the TCP connections up. These messages are needed in case the provider does not generate information often enough to keep the TCP-connection up. The default period of these messages is 60 s. These infrequent keep-alive packets are not shown in the figure, but they are counted as part of the protocol traffic and included in our analysis.

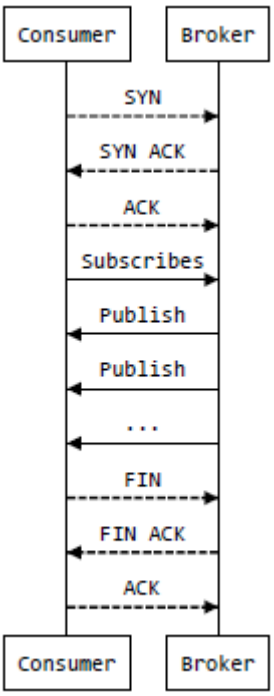


Figure 5.3 Packet flow between broker and consumer using MQTT. MQTT, in contrary to WS-Notification, only initializes one TCP connection and keeps it alive until the consumer terminates the relationship to the broker.

NDN does not maintain any connections for data exchange. Instead, information is requested periodically. One request is needed for each produced information element. Figure 5.4 shows the sequence diagram for NDN. The expectation is that MQTT and NDN should have almost identical performance in our setup. This is because each TCP data packet is associated with an ACK. Similarly each NDN data packet is associated with an interest. The difference is that TCP should give some added overhead since it is connection oriented and needs handshaking to setup and tear-down. The number of MQTT packets should therefore be higher. The results in Table 5.1 support this expectation, where it is shown that NDN exchanges the smallest number of packets.

Comparing the total number of bytes transmitted, NDN transmitted more bytes than MQTT. The larger overhead is mainly due to the NDN header size. Each NDN packet consists of a UDP and an NDN header. The size of the NDN header depends on the packet type (*Interest* or *Data*). The header size of an NDN packet is not fixed since the *name* field, selector field (*Interest*) and signed info (*Data* packet) are all variable in size. In our testbed all NDN data was signed with a default signature. The signature adds overhead compared to both WS-Notification and MQTT, which don't offer any security features by default. UDP also adds overhead since NDN is run as an overlay network over IP, thus NDN packets are encapsulated in UDP packets.

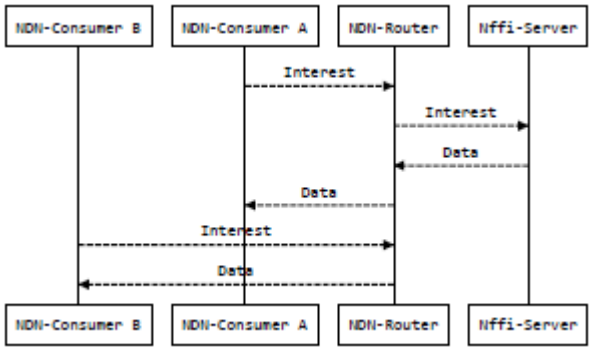


Figure 5.4 Packet Flow in an NDN network with two consumers and one producer.

The comparison of byte and packet numbers are only indicative, since they depend on the actual physical layer (PHY) and medium access (MAC) solutions of different transmission technologies. Technologies with larger layer 2 overhead will result in a relatively larger byte overhead for the protocol alternative with the largest fraction of small packets. Our small testbed does not capture well the gain from the caching effect in NDN. With a larger number of nodes where many are likely to be interested in the same data, the NDN design will behave like a multicast protocol that builds source specific trees.

In a larger network with many consumers of the same data chunks, the caching feature of NDN pulls the data closer to the consumers and reduces the number of transmitted packets for each new consumer. This effect is not present in the publish/subscribe techniques.

Table 5.1 shows a difference in average packet size for the three evaluated methods. The optimal packet size would typically depend on the environments as the effect of packet size would typically be different within wireless environments compared to wired environments. The optimal packet size in wireless environments will typically depend on, but is not limited to, interfering traffic, channel quality and mobility. Further information on the effect on packet size can be found in [30].

5.4 Summary – NDN's support for the information infrastructure

We have performed a small-scale comparative evaluation of the two publish/subscribe standards WS-Notification and MQTT, as well as the novel hybrid push/pull approach provided by NDN. While WS-Notification is NATO's standard of choice for publish/subscribe and should be used where it is applicable, it is not well suited for use over low capacity mobile military networks due to its overhead. In resource-constrained networks other standards like MQTT offer similar functionality but with less overhead. Apart from overhead, another drawback when considering publish/subscribe in mobile military networks is that the standards rely on a central broker to offload such tasks as subscription handling and message dissemination. The broker may constitute a single point of failure. Using multiple brokers mitigates the single point of failure,

but might increase the overall network traffic since the brokers need to synchronize information about subscriptions between themselves.

A more efficient approach seems to be leveraging the NDN concept. Here, the network layer handles caching of data and forwarding of interests, an approach which in our particular experiment proved to be the most efficient. Furthermore, the NDN implementation is broker-less by design, and could thus be better suited in a mobile military network. Hence, we recommend to pursue NDN further, both due to the low overhead and the added robustness that arises from it being without a central point of failure. NDN can be implemented as an overlay or as the IP network replacement. We only evaluated the overlay, which is a good migration strategy because it can be used on existing IP radios.

Due to the built-in caching and efficiency of the network level data dissemination used by NDN, we expect it to be scalable in larger networks than we used. This scalability, and especially the effects caching have on performance, should be investigated further.

This experiment has also been reported in a conference paper [70].

6 Related work and further reading

In this section we list some general ICN literature for further reading as well as reporting on related works for the use of ICN in mobile military networks. There are several surveys that compare different ICN solutions e.g., [25, 28, 71]. Of these [25] is a very well written comprehensive optimistic survey and [28] represents a more critical view. Most of the ICN surveys focus on use in high-capacity fixed Internet infrastructure. The use of ICN in wireless networks has some other advantages and disadvantages. [35] provides a survey of ICN work for wireless and mobile networks. This paper is well worth reading and goes into detail on several of the topics that we have briefly discussed in this report.

Research initiatives have studied ICN for use in several types of mobile wireless networks; general Mobile Ad Hoc Networks (MANET), Vehicular MANET (VANET), Wireless Sensor Networks (WSN) as well as mobile military networks. In this report we are interested in ICN for the latter use. While the characteristics of general MANETs are similar to mobile military networks, the characteristics of VANETs and WSNs differ. For a starting point for literature about ICN in VANETs and WSN see [35]. The few studies that we are aware of that report on ICN for military mobile military networks will be described below.

The authors of [72] discuss how the basic principles of NDN can be beneficial for military networks. The aim of the paper is to show opportunities, not challenges, thus the paper does not go into technical details and is not critical. The authors argue that since NDN does not use a

host model, it is well suited to deal with mobility. The multipoint nature of content retrieval in NDN combined with the broadcast capabilities of today's network devices, provides flexibility to maintain communication in highly dynamic environments. Also, using a cache, a mobile node may serve as the network medium between disconnected areas, or provide delayed connectivity over intermittent links. NDN promote the use of an intelligent forwarding plane that can make smart forwarding decision based on traffic history etc. Specific to NDNs, the *strategy* module can control multi-path forwarding, decide on performance-based optimal paths and avoid dead paths even before the operational routing protocols detect failure. The flexibility of the security architecture and the fact that NDN is loop-free is also underlined. The authors use a ship to shore network (The US Navy's Automated Digital Networking System (ADNS)) and connections via WIN-T (The US Army's Warfighter Information Network-Tactical) as example networks where NDN can be beneficial. In [73] they have emulated the two network types using Emulab [74] and report on the results. The reported results are encouraging; however the experiments are designed to show the benefits and do not highlight potential problems. Chunk-sizes, timers, alternative behavior of wireless *Face* and *strategy* are not discussed.

In [39] the authors claim that some modifications of NDN are needed to better exploit the potential of NDN in this domain. NDN is a pull-based architecture where the consumer always asks for content. Here the authors extend the architecture to also support push-based traffic. They propose to divide the content in topic based content (e.g., data files, video and audio files, etc.) and spatial/temporal content (e.g., situation awareness data and sensors information) where the latter traffic is prioritized. It is suggested to push the temporal situational content with geographic routing and use more classical NDN approaches for the topic based data. To avoid broadcast collision on *Data* when the *strategy* layer has chosen to send *Interests* on multiple interfaces and broadcast channels, a node sends *Reply* to *Interest* with a corresponding *Request* on the chosen path before broadcasting the content chunk. The authors also promote the use of pre-stored keys to simplify the security architecture. A simple experiment is conducted where NDN is compared with a solution for IP routing and a file sharing overlay.

[75] also postulates that built-in functionalities of NDN can enhance network efficiency and robustness in disruptive, intermittent connectivity, and low bandwidth (DIL) environments. By doing in-network caching, data naming, stateful forwarding and securing content, NDN can potentially advance the performance for these networks. The paper describes an experiment using CORE (Common Open Research Emulator) [76], on a very disruptive ship – shore network that uses unreliable satellite communications and Unmanned Aerial Vehicles (UAVs) with a postman functionality to connect the ships to the shore. The paper compares unicast (Optimized Link State Routing Protocol (OLSR) [77] and multicast (Simplified Multicast Forwarding (SMF) [78]) distribution for IP with NDN. It is shown how NDN outperforms the IP architecture for this scenario and how NDN can mix the unicast and multicast data dissemination models to achieve localized robustness to disruption. The paper does not discuss challenges with the NDN architecture.

In [79] the authors suggest to use NDN as the tactical network in Gray Zone[11] conflicts. Two interesting scenarios are evaluated using the mininet [80] environment. In the first scenario a heterogeneous WIN-T network consisting of the Brigade command network, the Battalion command network, and the Company command network is used. In this scenario images are retrieved from the upper echelon Imagery Server to the Battalion client using UDP, TCP, and NDN protocols. The second scenario is an IoT sensor fusion network that utilizes Long Term Evolution (LTE) connections and the Internet. In this scenario ten sensors transmit 136 bytes of telemetry once a second, for one hour to a sensor fusion host using UDP/IP-multicast and NDN. These experiments also show that NDN outperforms the other solutions as the loss-rates of the connections increase. However, as expected, the push-based IP-multicast architecture exhibits lower delays than the pull-based NDN architecture in the sensor experiment. The WIN-T testbed is reused in [34] for a similar experiment. In this paper the authors discuss some of the advantages and disadvantages with the NDN security model. They highlight the need to encrypt the *Interest* messages for privacy and bring up the tradeoff that this gives regarding efficient use of caches. The experiment does not seem to support the discussed security improvements.

The potentials of NDN are in [81] discussed in lieu of a scenario that describes a wide-area surveillance system that delivers imagery of sites of interest around a fixed location. The sensors and communication assets are owned by different coalition partners. In addition to the benefits listed in the papers described above, this paper also suggests that the namespace can be used to steer how and which content to produce by stating: "Data *names* can identify not only the existing data, but also data to be produced, effectively integrating network connectivity, storage, and processing engines under the same architectural umbrella." The paper also lists some topics that need more research e.g., naming conventions, *name* confidentiality and policy management for NDN *strategies*, flow control and access control.

NDN's potential impact on tactical application development is discussed in [82]. The authors claim that the classic IP architecture struggles to support: 1) challenging and dynamic communication environments (e.g., battlefield scenarios) and 2) resource-limited devices (e.g., IoT). They further claim that tactical application designs can be simplified while hardening security and robustness under challenging network conditions. The authors' views are that the following characteristics of the NDN architecture can help ensure this: Host-independent behavior, multicast communication, pervasive network-accessible storage, opportunistic communication, namespace synchronization as transport, and data-centric security.

We also observe that a few research programs in the USA which focus on ICN-solutions for military networks have been started. The ones we are aware of are Defense Advanced Research Projects Agency (DARPA) Sharing Battlefield Information at Multiple Classification Levels (SHARE) [83] and National Science Foundation (NSF)/Intel ICN at Wireless Edge Networks ICN-WEN [84].

[11] "Gray Zone conflicts happens somewhere in the "Gray Zone" of the continuum between strict diplomacy at the lowest intensity of the spectrum and open warfare at the highest intensity. Often there is ambiguity on the exact nature, specific parties, and ultimate goals of the conflict, but a critical aspect of Gray Zone operations is sharing of information in order to modify the perceptions and beliefs of the involved parties." [79]

NATO Science and Technology Organization (STO) has also started a research task group on this topic: IST-161 «Efficient group and information centric communications in mobile military heterogeneous networks». FFI participates in this one.

7 Concluding remarks and future work

ICN is a clean slate network architecture that collapses much of the communication infrastructure and information infrastructure into one common architecture. In this report we have studied the NDN variant of ICN from two different viewpoints. We have studied the applicability of NDN in the communication infrastructure of mobile military networks and how NDN can also perform some of the functionality which is usually associated with the information infrastructure. The target for the study was future mobile military networks and services that are expected to run over these networks in the same timeframe (about 10 – 20 years into the future). We expect that these future networks will be highly heterogeneous i.e. consisting of different radio technologies and that highly agile services are needed that can find both planned and ad hoc produced content from a large range of producers in an efficient manner.

Regarding the communication infrastructure we observe that a range of protocols and mechanisms has been added to the classical IP network architecture in order to tailor the IP network to handle different scenarios and traffic models. There are protocols to handle MANETs, node and network mobility, highly disruptive networks, traffic to single consumers or groups of consumers, etc. This results in a patchwork of protocols that often cannot be used simultaneously, complicating both configuration and management of the networks

NDN is designed to handle many of the mentioned functionality natively. It is able to automatically handle different levels of network disruption caused both by difficult terrain and electronic warfare threats. It can handle consumer mobility well and automatically supports both traffic to a single consumer and a group of consumers. It is built around the search for content and thus aims for efficient content distribution.

In this activity we built an NDN demonstrator to get good understanding of how the NDN architecture works. The available open source NDN codebase was not optimized for wireless networks. In order to evaluate if the expected advantages of NDN hold also in a mobile military network environment, the codebase had to be extended with wireless functionality. In this activity we implemented a wireless *Face* that can be used as a stepping stone for further work to tailor the architecture for mobile military networks. The purpose of the activity was to get a better understanding of how NDN works and reach a preliminary decision if it is worth studying the technology further for potential use in the Norwegian Armed Forces' future mobile military network. The experience we have gained with the use of NDN has strengthened our view that

this architecture has interesting characteristics for military use. This activity did not reveal any new major showstoppers for use in such networks. However, neither did the activity close any of the expected unresolved challenges associated with the NDN architecture (e.g., design of a scalable namespace). Thus we recommend further studies of this architecture as it matures, to see if the challenges can be solved in a sound manner.

Regarding NDNs support for the information infrastructure we compared NDN with NATO's SOA-baseline for publish/subscribe (WS-Notification) as well as MQTT which is better adapted for use over mobile military networks. Both of the publish/subscribe protocols rely on a broker. A drawback of broker-based publish/subscribe is that the broker typically constitutes a single point of failure. To successfully leverage publish/subscribe in mobile military networks, one would need to mitigate this single point of failure either by making a multi-broker setup or by using an approach that does not rely on brokers. NDN's search for content is fully distributed and does not rely on a broker. NDN's information retrieval is also integrated with the architecture and does not require a middleware protocol layer above the communication infrastructure as the publish/subscribe solutions does. The purpose of this activity was to get a first answer to how well NDN is able to support the information infrastructure. We did a quantitative study to measure the network capacity required by the three different protocols to provide an NFFI service in a four node network. NDN did well in the experiment. However it was noted that a push service would be beneficial

Both activities resulted in a positive evaluation of NDN for mobile military network with the acknowledgment that there are many unsolved problems that must be resolved before this can become a fielded technology. For future work it is important to do more work to tailor NDN for mobile networks as well as quantitative evaluation in larger network scenarios. For future work we hope to perform a large scale experiment in context of the NATO research task group IST-150 "NATO Core Services Profiling for Hybrid Tactical Networks" or IST-161 "Efficient group and information centric communications in mobile military heterogeneous networks" using a network emulator and the Anglova scenario [85]. This experiment can test both information dissemination as well as network performance for mobile military networks. The final steps will involve using NDN in an actual mobile military network, supporting C2 services.

References

[1] NATO NC3A, *NATO Network Enabled Capability Feasibility Study (Version 2.0)*, NATO, 2005.

[2] ACT, *Federated Mission Networking (FMN) Portal*. Available: https://tide.act.nato.int/tidepedia/index.php/Federated_Mission_Networking_(FMN)_Portal [Accessed: 05.01.2018]

[3] NATO, *C3 Taxonomy*. Available: https://tide.act.nato.int/em/index.php/C3_Taxonomy. [Accessed: 06.03.2018]

[4] S. Russell and T. Abdelzaher, "The Internet of Battlefield Things: The Next Generation of Command, Control, Communications and Intelligence (C3I) Decision-Making," in proceedings *IEEE MILCOM*, Los Angeles, CA, USA, pp. 737-742, 2018.

[5] E. Skjelland, *et al.*, *Hvordan styrke forsvaret av Norge? Et innspill til ny langtidsplan (2021–2024)*, FFI Rapport 19/00328, 2019, Available: https://www.ffi.no/en/Publications.

[6] Forsvaret, *Forsvarets fellesoperative doktrine*, Forsvarsstaben, 2014, Available: http://hdl.handle.net/11250/224031.

[7] L. Landmark, M. Hauge, and K. Ø, "Routing Loops in Mobile Heterogeneous Ad Hoc Networks," in proceedings *IEEE MILCOM*, San Diego, CA, USA, pp. 112-118, Nov. 2013.

[8] M. Pradhan, C. Fuchs, and F. T. Johnsen, "A survey of applicability of military data model architectures for smart city data consumption and integration," in proceedings *WF-IoT*, Singapore, pp. 129-134, 2018.

[9] C. E. Perkins, *Ad Hoc Networking*, Addison-Wesley Professional, 2000.

[10] Y. Cao and Z. Sun, "Routing in Delay/Disruption Tolerant Networks: A Taxonomy, Survey and Challenges," *Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 654-677, 2013.

[11] M. A. Brose and M. Hauge, *Group communication in mobile military networks*, FFI Rapport 2012/00294, 2012, Available: https://www.ffi.no/en/Publications.

[12] C. Perkins (Ed.). "IP Mobility Support for IPv4, Revised", IETF, *RFC5944*. Nov. 2010. Available: http://www.ietf.org.

[13] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. "Network Mobility (NEMO) Basic Support Protocol", IETF, *RFC53963*. Jan. 2005. Available: http://www.ietf.org.

[14] Object Management Group (OMG), *Data Distribution Service (DDS)*. Available: https://www.omg.org/omg-dds-portal/. [Accessed: 10. Jan. 2018]

[15] F. Guidec and Y. Maheo, "Opportunistic Content-Based Dissemination in Disconnected Mobile Ad Hoc Networks," in proceedings *UBICOMM*, Papeete, France, pp. 49-54, 2007.

[16] K. Scott, *et al.*, "Robust communications for disconnected, intermittent, low-bandwidth (DIL) environments," in proceedings *IEEE MILCOM*, Baltimore, MD, USA, pp. 1009-1014, 2011.

[17] M. Hauge, M. A. Brose, and O. I. Bentstuen, "Group communication in tactical networks: A discussion," in proceedings *MCC*, St. Malo, France, pp. 1-8, 2013.

[18] J. Grönkvist, *et al.*, *Robust group communications for mobile ad hoc networks*, Totalförsvarets forskningsinstitut-rapport FOI-R--4610--SE, 2018, Available: https://www.foi.se/en/foi/reports.html.

[19] M. Hauge and L. Landmark, "Limiting the flooding of simplified multicast forwarding to a defined scope," in proceedings *ICMCIS*, Warsaw, Poland, pp. 1-8, 2018.

[20] L. Landmark, E. Larsen, M. Hauge, and Kure, "Resilient internetwork routing over heterogeneous mobile military networks," in proceedings *IEEE MILCOM*, Tampa, Fl, USA, pp. 388-394, Oct. 2015.
[21] L. Landmark, K. Øvsthus, and K. Ø, "Routing trade-offs in sparse and mobile heterogeneous multi-radio ad hoc networks," in proceedings *IEEE MILCOM*, San Jose, CA, USA, pp. 2229-2236, Nov. 2010.
[22] M. A. Brose, M. Hauge, J. E. Voldhaug, and J. Sander, *Multi-Topology Routing - QoS functionality and results from CoNSIS field experiment*, FFI Rapport 2013/00529, 2013, Available: www.ffi.no/en/Publications.
[23] M. Hauge (ed.), *Heterogeneous tactical networks – improving connectivity and network efficiency*, NATO STO-TR-IST-124-PART-I (to be published), 2019, Available: https://www.sto.nato.int/publications/.
[24] F. T. Johnsen, T. H. Bloebaum, and P.-P. Meiler, "Improving Integration between Tactical and HQ Levels by making SOA applicable on the Battlefield," presented at the ICCRTS, Los Angeles, CA, USA, 2017.
[25] G. Xylomenos, *et al.*, "A Survey of Information-Centric Networking Research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024-1049, 2014.
[26] V. Jacobson, *et al.*, "Networking named content," in proceedings *ACM CoNEXT*, Rome, Italy, pp. 1-12, 2009.
[27] Named Data Networking project, *Named Data Networking (NDN) - A Future Internet Architecture*. Available: https://named-data.net/. [Accessed: 3. Jan. 2018]
[28] A. Ghodsi, *et al.*, "Information-centric networking: seeing the forest for the trees," in proceedings *ACM HotNets*, Cambridge, Massachusetts, pp. 1-6, 2011.
[29] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR, 2005.
[30] F. T. Johnsen, *Pervasive web services discovery and invocation in military networks*, FFI Rapport 2011/00257, 2011, Available: https://www.ffi.no/en/Publications.
[31] S. Kent and K. Seo. "Security Architecture for the Internet Protocol", IETF, *RFC2401*. Dec. 2005. Available: http://www.ietf.org.
[32] R. Alubady, S. Hassan, and A. Habbal, "A TAXONOMY OF PENDING INTEREST TABLE IMPLEMENTATION APPROACHES IN NAMED DATA NETWORKING," *JATIT*, vol. 91, pp. 411-423, 2016.
[33] D. Smetters and V. Jacobson, *Securing Network Content*, PARC Tech Report, Oct. 2009, (Smetters-2009).
[34] J. B. Evans, S. G. Pennington, and B. J. Ewy, "Named data networking protocols for tactical command and control," in proceedings *SPIE Defense + Security*, Orlando, FL, United States, p. 7, 2018.
[35] M. Amadeo, C. Campolo, A. Molinaro, and G. Ruggeri, "Content-centric wireless networking: A survey," *Computer Networks*, vol. 72, pp. 1-13, Elsevier, 2014.
[36] G. Carofiglio, V. Gehlen, and D. Perino, "Experimental Evaluation of Memory Management in Content-Centric Networking," in proceedings *IEEE ICC*, Kyoto, Japan, pp. 1-6, 2011.
[37] P. Kietzmann, *et al.*, "The need for a name to MAC address mapping in NDN: towards quantifying the resource gain," in proceedings *ACM ICN*, Berlin, Germany, pp. 36-42, 2017.
[38] M. Meisel, V. Pappas, and L. Zhang, "Ad hoc networking via named data," in proceedings *ACM MobiArch*, Chicago, Illinois, USA, pp. 3-8, 2010.
[39] S. Y. Oh, D. Lau, and M. Gerla, "Content Centric Networking in tactical and emergency MANETs," in proceedings *IFIP Wireless Days*, Venice, Italy, pp. 1-5, 2010.

[40] L. Wang, *et al.*, "Rapid traffic information dissemination using named data," in proceedings *ACM NoM*, Hilton Head, South Carolina, USA, pp. 7-12, 2012.
[41] E. Baccelli, *et al.*, "Information centric networking in the IoT: experiments with NDN in the wild," in proceedings *ACM-ICN*, Paris, France, pp. 77-86, 2014.
[42] Y. Yu, *et al.*, "Interest propagation in named data manets," in proceedings *ICNC*, San Diego, CA, USA pp. 1118-1122, 2013.
[43] M. Varvello, *et al.*, "On the design of content-centric MANETs," in proceedings *WONS*, Bardonecchia, Italy, pp. 1-8, 2011.
[44] S. Ratnasamy, *et al.*, "GHT: a geographic hash table for data-centric storage," presented at the ACM WSNA, Atlanta, Georgia, USA, 2002.
[45] Named Data Networking project, *NLSR - Named Data Link State Routing Protocol*. Available: http://named-data.net/doc/NLSR/current/. [Accessed: 4. Feb. 2018]
[46] J. Moy (ed.). "OSPF Version 2", IETF, *RFC2328*. Apr. 1998. Available: http://www.ietf.org.
[47] T. Clausen, C. Dearlove, P. Jacquet, and U. Herberg. "The Optimized Link State Routing Protocol Version 2", IETF, *RFC7181*. April 2014. Available: http://www.ietf.org.
[48] C. Perkins, E. Belding-Royer, and S. Das. "Ad hoc On-Demand Distance Vector (AODV) Routing", IETF, *RFC3561*. July 2003. Available: http://www.ietf.org.
[49] Linux Programmer's Manual, *Listen()*. Available: http://man7.org/linux/man-pages/man2/listen.2.html. [Accessed: 06.03.2018]
[50] Named Data Networking project, *NFD Developer's Guide*. Available: https://named-data.net/publications/techreports/ndn-0021-10-nfd-developer-guide/. [Accessed: 4. Feb. 2018]
[51] Named Data Networking project, *NFD - Named Data Networking Forwarding Daemon*. Available: https://named-data.net/doc/NFD/current/. [Accessed: 4. Feb. 2018]
[52] OASIS, *Web Services Base Notification 1.3 (WS-BaseNotification)*, OASIS Standard, 1 Oct. 2006, Available: http://docs.oasis-open.org/wsn/wsn-wsbase notification-1.3-spec-os.pdf.
[53] Consultation Command and Control Board (C3B), *CORE ENTERPRISE SERVICES STANDARDS RECOMMENDATIONS: THE SOA BASELINE PROFILE VERSION 1.7. Enclosure 1 to AC/322-N(2011)0205*, NATO (NATO Unclassified releasable to EAPC/PFP), 11. Nov. 2011.
[54] T. H. Bloebaum and F. T. Johnsen, "Evaluating publish/subscribe approaches for use in tactical broadband networks," in proceedings *IEEE MILCOM*, Tampa, FL, USA pp. 605-610, 2015.
[55] M. Manso, *et al.*, "SOA and Wireless Mobile Networks in the tactical domain: Results from experiments," in proceedings *IEEE MILCOM*, Tampa, FL, USA, pp. 593-598, 2015.
[56] OASIS, *MQTT Version 3.1.1.* , OASIS Standard, 29 Oct. 2014, Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf.
[57] A. Carzaniga, M. Papalini, and A. L. Wolf, "Content-based publish/subscribe networking and information-centric networking," in proceedings *ACM ICN*, Toronto, Ontario, Canada, pp. 56-61, 2011.
[58] C. M. MacKenzie, *et al.*, *Reference model for service oriented architecture 1.0 OASIS standard*, 12 Oct 2006, Available: http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf.
[59] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114-131, 2003.

[60] Consultation Command and Control Board (C3B), *C3 Taxonomy Baseline 2.0*, NATO AC/322-D(2016)0017, 14 March 2016, Available: https://www.nato.int/ebookshop/.
[61] OASIS, *Web Services Brokered Notification 1.3 (WSBrokeredNotification)*, OASIS Standard, 1 Oct. 2006, Available: http://docs.oasis-open.org/wsn/wsn-ws brokered notification-1.3-spec-os.pdf.
[62] OASIS, *Web Services Topics 1.3 (WS-Topics)*, OASIS Standard, 1 Oct. 2006, Available: http://docs.oasis-open.org/wsn/wsn-ws topics-1.3-spec-os.pdf.
[63] E. Bertelsen, *et al.*, "Federated Publish/Subscribe Services," in proceedings *IFIP International Conference on New Technologies, Mobility and Security*, Paris, France pp. 1-5, 2018.
[64] M. Manso, M. R. Brannsten, and F. T. Johnsen, "A Smart Devices Concept for Future Soldier Systems," presented at the ICCRTS, Los Angeles, CA, USA, 2017.
[65] T. H. Bloebaum and F. T. Johnsen, *CWIX 2014 core enterprise services experimentation*, FFI Rapport 2014/01510, 2014, Available: https://www.ffi.no/en/Publications.
[66] Eclipse, *Eclipse MosquittoTM An open source MQTT broker*. Available: https://mosquitto.org/.
[67] Named-data, *ndn-cxx: NDN C++ library with eXperimental eXtensions*. Available: http://named-data.net/doc/ndn-cxx/current/.
[68] Fusesource, *An MQTT client*. Available: https://github.com/fusesource/mqtt-client.
[69] Named-data, *NDN Client Library for Java*. Available: https://github.com/named-data/jndn.
[70] F. T. Johnsen, *et al.*, "Publish/Subscribe Versus a Content-Based Approach for Information Dissemination," in proceedings *IEEE MILCOM*, Los Angeles, CA, USA, pp. 1-9, 2018.
[71] B. Ahlgren, *et al.*, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26-36, 2012.
[72] B. Etefia and L. Zhang, "Named Data Networking for military communication systems," in proceedings *IEEE Aerospace Conference*, Big Sky, MT, USA, pp. 1-7, 2012.
[73] B. Etefia, M. Gerla, and L. Zhang, "Supporting military communications with Named Data Networking: An emulation analysis," in proceedings *IEEE MILCOM*, Orlando, FL, USA, pp. 1-6, 2012.
[74] The University of Utah, *Emulab*. Available: https://www.emulab.net/. [Accessed: 21. Jan. 2018]
[75] M. T. Refaei, S. Ha, Z. Cavallero, and C. Hager, "Named Data Networking for tactical communication environments," in proceedings *IEEE NCA*, Cambridge, MA, USA, pp. 118-121, 2016.
[76] *Common Open Research Emulator (CORE)*. Available: https://www.nrl.navy.mil/itd/ncs/products/core. [Accessed: 23. Jan. 2018]
[77] T. Clausen, C. Dearlove, P. Jacquet, and U. Herberg. "The Optimized Link State Routing Protocol Version 2", IETF, *RFC7181*. Apr. 2014. Available: http://www.ietf.org.
[78] J. Macker(ed.). "Simplified Multicast Forwarding", IETF, *RFC6621 (Experimental)*. May, 2012. Available: http://www.ietf.org.
[79] J. B. Evans, S. G. Pennington, and B. J. Ewy, "Communication networks for the tactical edge," in proceedings *SPIE Defense + Security*, Anaheim, California, United States, p. 9, 2017.

[80] *Mininet - An Instant Virtual Network on your Laptop (or other PC)*. Available: http://mininet.org/. [Accessed: 23. Jan. 2018]
[81] C. Gibson, *et al.*, "Opportunities and challenges for named data networking to increase the agility of military coalitions," in proceedings *IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI*, San Francisco, CA, USA, pp. 1-6, 2017.
[82] J. Burke, A. Afanasyev, T. Refaei, and L. Zhang, "NDN Impact on Tactical Application Development," in proceedings *IEEE MILCOM*, Los Angeles, CA, USA 2018.
[83] DARPA, *Sharing Battlefield Information at Multiple Classification Levels via Mobile Handheld Devices (SHARE)*. Available: https://www.darpa.mil/news-events/2017-01-10. [Accessed: 29. March 2019]
[84] NSF, *NSF/Intel Partnership on Information-Centric Networking in Wireless Edge Networks (ICN-WEN)* Available: https://www.nsf.gov/pubs/2016/nsf16586/nsf16586.htm. [Accessed: 29. March 2019]
[85] N. Suri, *et al.*, "The Angloval Tactical Military Scenario and Experimentation Environment," in proceedings *ICMCIS*, Warsaw, Poland 2018.

Abbreviations

ACK	Acknowledgement
ACT	Allied Command Transformation
ADNS	The US Navy's Automated Digital Networking System
AODV	Ad hoc On-Demand Distance Vector
AMN	Afghan Mission Network
API	Application Programming Interface
ARQ	Automatic Repeat Request
C3	Consultation, Command and Control
CCN	Content Centric Networking
CDN	Content Delivery Networks
CDT	Connected Dominating Set
CIS	Communication and Information Systems
CORE	Common Open Research Emulator
CWIX	NATO Coalition Warrior Interoperability eXercise
DARPA	Defense Advanced Research Projects Agency
DDS	Data Distribution Service
DIL	Disruptive, Intermittent connectivity, and Low bandwidth
DNS	Domain Name Servers
DTN	Disruption/delay tolerant networking
EW	Electronic Warfare
FIB	Forwarding Information Base
FMN	Federated Mission Networking
GHT	Geographic Hash Tables
HTTP	Hypertext Transfer Protocol
ICN	Information-Centric Networking
ICN-WEN	ICN at Wireless Edge Networks
IoBT	Internet of Battlefield Things
INFOSEC	Information Security
IoT	Internet of Things
IP	Internet Protocol
JISR	Joint Intelligence, Surveillance and Reconnaissance
QoS	Quality of Service
LFBL	Listen First Broadcast Later
LTE	Long Term Evolution
MAC	Medium Access Layer
MANET	Mobile Ad Hoc Networks
MQTT	Message Queuing Telemetry Transport
MPR	Multi Point Relay
NAIF	Neighborhood-Aware Interest Forwarding
NDN	Named Data Networking
NETSEC	Network Security
NEMO	Network Mobility

NFD	Networking Forwarding Daemon
NFFI	NATO Friendly Force Information
NLSR	Named Data Link State Routing Protocol
NNEC	NATO Network Enabled Capability
NSF	National Science Foundation
OASIS	Organization for the Advancement of Structured Information Standards
OLSR	Optimized Link State Routing
OS	Operating System
OSPF	Open Shortest Path First
PHY	Physical layer
PIT	Pending Interest Table
PKI	Public Key Infrastructure
RPi3	Raspberry Pi 3
RTO	Related Retransmission Timeout
SHARE	Sharing Battlefield Information at Multiple Classification Levels
SMF	Simplified Multicast Forwarding
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
STO	Science and Technology Organization
TCP	Transmission Control Protocol
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UGV	Unmanned Ground Vehicle
VANET	Vehicular MANET
V2V	Vehicle to Vehicle
WIN-T	The US Army's Warfighter Information Network-Tactical
WSN	Wireless Sensor Networks
WS-Notification	Web Services-Notification
XML	Extensible Markup Language

About FFI

The Norwegian Defence Research Establishment (FFI) was founded 11th of April 1946. It is organised as an administrative agency subordinate to the Ministry of Defence.

FFI's MISSION

FFI is the prime institution responsible for defence related research in Norway. Its principal mission is to carry out research and development to meet the requirements of the Armed Forces. FFI has the role of chief adviser to the political and military leadership. In particular, the institute shall focus on aspects of the development in science and technology that can influence our security policy or defence planning.

FFI's VISION

FFI turns knowledge and ideas into an efficient defence.

FFI's CHARACTERISTICS

Creative, daring, broad-minded and responsible.

Om FFI

Forsvarets forskningsinstitutt ble etablert 11. april 1946. Instituttet er organisert som et forvaltningsorgan med særskilte fullmakter underlagt Forsvarsdepartementet.

FFIs FORMÅL

Forsvarets forskningsinstitutt er Forsvarets sentrale forskningsinstitusjon og har som formål å drive forskning og utvikling for Forsvarets behov. Videre er FFI rådgiver overfor Forsvarets strategiske ledelse. Spesielt skal instituttet følge opp trekk ved vitenskapelig og militærteknisk utvikling som kan påvirke forutsetningene for sikkerhetspolitikken eller forsvarsplanleggingen.

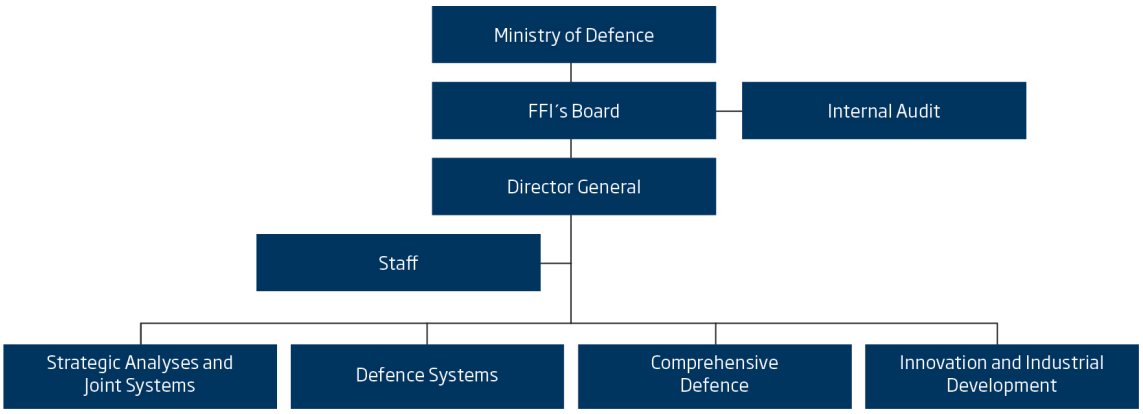
FFIs VISJON

FFI gjør kunnskap og ideer til et effektivt forsvar.

FFIs VERDIER

Skapende, drivende, vidsynt og ansvarlig.

FFI's organisation



Forsvarets forskningsinstitutt
Postboks 25
2027 Kjeller

Besøksadresse:
Instituttveien 20
2007 Kjeller

Telefon: 63 80 70 00
Telefaks: 63 80 71 15
Epost: ffi@ffi.no

Norwegian Defence Research Establishment (FFI)
P.O. Box 25
NO-2027 Kjeller

Office address:
Instituttveien 20
N-2007 Kjeller

Telephone: +47 63 80 70 00
Telefax: +47 63 80 71 15
Email: ffi@ffi.no

FFI Forsvarets forskningsinstitutt
Norwegian Defence Research Establishment