Publish/Subscribe Versus a Content-Based Approach for Information Dissemination

Frank T. Johnsen, Lars Landmark, Mariann Hauge, Erlend Larsen, Øivind Kure
Norwegian Defence Research Establishment (FFI)
Kjeller, Norway

***Abstract*—NATO has identified the WS-Notification standard from OASIS to support event-driven communication in the NATO enterprise and when building coalition networks. Using this standard promotes interoperability. However, there is significant overhead associated with WS-Notification since it is built on SOAP Web services (WS). Overhead can be problematic in networks with scarce resources. In this paper we perform a small-scale comparative evaluation of overhead of WS-Notification with another publish/subscribe standard: Message Queuing Telemetry Transport (MQTT). We also measure how these standards compare to the novel approach of content-based networking under the same networking conditions. We use the Named Data Networking (NDN) flavor of content-based networking for our experiment. Though fundamentally different, these approaches can be used to realize the Service-Oriented Architecture (SOA) paradigm.**

The drawback of standard publish/subscribe approaches is that they usually rely on a broker, which constitutes a single point of failure. NDN, on the other hand, has no broker which makes it interesting to consider for tactical networks. We use NATO Friendly Force Information (NFFI), which is much used for friendly force tracking, as the data format for the payload in all our tests.

In the paper we focus on the respective approaches' network resource consumption. Based on the results we argue that the content-based approach seems promising and should be investigated further.

I. Introduction

Modern warfare requires an information infrastructure that facilitates extensive information sharing. 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.

Since then, the military community at large has committed to these ideas for how to build infrastructures and it is currently the approach leveraged for Federated Mission Networking (FMN) [28]. Building an information infrastructure in the military domain differs from building one in the civilian domain, especially at the tactical level, and civilian solutions can rarely be used out of the box [6]. Thus, it has not yet been possible to take full advantage of SOA in military infrastructures. NATO has identified a set of *Core Services*, which provide common machine-to-machine (M2M) communication functionality that other services (functional area services and community of interest services, e.g., C2 services) depend upon. An example of a Core Service is messaging. M2M messaging includes both request/response and publish/subscribe services. In this paper, we focus specifically on publish/subscribe services.

Both NNEC and the early spirals of FMN focus on interoperability at the strategic and operational levels, 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. NATO has chosen the OASIS standard WS-Notification for publish/subscribe in its SOA baseline [2]. 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. Hence, it is not necessarily well suited for use in tactical networks where network capacity typically is low.

Named Data Networking (NDN) starts the path moving from current host-centric approaches towards a data-centric network architecture where data is identified by name rather than through an IP address [23]. The idea is that this paradigm shift puts focus on the data, named content, rather than addressing the physical location where the data should be obtained from. This is motivated by current Internet traffic patterns, where one often sees data being sent from a source to several users. NDN has important features by design, such as security measures in the protocol. NDN does also have some interesting features when it comes to the support of delay tolerant traffic. Another benefit is the support of unicast/multicast within the same framework. In NDN there is no need to run two services, one for multicast and one for unicast as they are both supported by the design. These features make NDN worth considering for use also in military networks, and is the reason we include it in our study even though it is not a standard at this point in time.

The contribution of this paper is a comparative evaluation of the WS-Notification standard, the Message Queuing Telemetry Transport (MQTT) standard, and NDN with respect to overhead.

The remainder of the paper is organized as follows: Section II introduces central SOA terminology, and the standards and approaches that our work is based on. Section III describes the methodology used. Sections IV and V discuss the test setup and results, respectively. Section VI presents relevant related work. Finally, Section VII concludes the paper and outlines future work.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/MILCOM.2018.8599786

II. Service-Oriented Architecture

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 paper [3]:

> 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 [3] 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 paper 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 [2]. We also include MQTT, which was found to have promising properties in an earlier experiment [4]. 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.

A. Publish/subscribe

Publish/subscribe [16] 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 one of the Core Services in NATO's C3 Taxonomy [26]. 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.

B. WS-Notification

WS-Notification is a set of three standards from OASIS: WS-BaseNotification [17], which gives an approach to publish/subscribe and defines the basic message exchange and associated roles and formats. WS-BrokeredNotification [18] extends WS-BaseNotification with support for brokered publish/subscribe, whereas WS-Topics [19] 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. Therfore, in this paper we investigate WS-Notification as extended by WS-BrokeredNotification rather than the more basic WS-BaseNotification.

C. MQTT

MQTT is also an OASIS standard [5]. 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 [6] for a closer study of its performance in tactical networks.

D. Named Data Networking

Information Centric Networking (ICN) is an architecture that has received increasing interest over the last decade. Several approaches to ICN have been proposed. Our work is based on NDN [22], [23], an open source refinement/development of the original concept proposal CCNx [27]. In NDN, consumers request content by the aid of an *interest* packet. The content is identified by a hierarchical unique naming structure (name space). In response to the interest message is the corresponding *data* chunk. Upon reception of an interest, each router stores a pointer back to the previous hop for the interest. When the interest reaches a node that has the requested content, the data is forwarded along the reverse path of "breadcrumbs" (i.e., trail of the interest). The routing function is therefore only concerned with forwarding of interests. NDN can have multiple routing strategies. These can differ between different name spaces. For an infrequently used name space it may

DOI til publisert versjon/DOI to published version: 10.1109/MILCOM.2018.8599786

be beneficial to broadcast (flood) the interest on all outgoing interfaces, while popular name spaces can benefit from a maintained shortest path routing. Examples of name spaces can be ”/video/sensors/” and ”/nffi/”. When a data packet is forwarded in response to an interest, the data may be cached in all the intermediate routers. Scalability is ensured by choosing an appropriate local caching strategy. When a router that has a specific data chunk in its cache receives an interest for this data, the router can respond to the interest with the cached data, thereby reducing overhead on upstream links

In IP networking, reliability and network congestion are usually handled by TCP at OSI-layer 4. NDN, on the other hand, places the responsibility for reliability and flow control in the applications themselves. In the NDN architecture, the application issues interests and waits for the data response. If no data is received after a timeout, the interest is assumed lost and the application will resend the interest. NDN can impact the robustness of the interest forwarding by sending the interest along one or several paths.

The exchange of interest and data provides an inherent congestion control on the used path. NDN adjusts the rate of transmitted interest within the application itself and/or by intermediate routers. In case an intermediate router is congested, it can choose to drop the interest or send a negative ACK of the interest back to the previous hop notifying the previous hop that it is congested.

NDN may eventually replace the IP forwarding, routing and location mapping. However, in a transition phase, it can run on top of IP, encapsulating the packets in UDP/IP. For the experiments in this paper we run NDN over IP, which is in key with NATO's ”Everything over IP” principle.

One interesting feature of NDN is that security is built into the protocol: All data chunks must be cryptographically signed. In NDN, making signing data a part of the architecture ensures that it is applied to all data that is sent. By signing the data, the consumer can trust data received since the integrity is guaranteed. However, this functionality of signing data adds overhead to the NDN header.

E. Architectural discussion

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 it is transmitted.

Military applications have different 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.

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, in this paper we consider only standards for publish/subscribe.

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.

F. 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. It is therefore a good example of a representative standard payload for the data dissemination comparison test.

III. Methodology

For the work in this paper we apply Denning's *design approach* [24] which is well suited to applied research. The design approach consists of four stages: 1) Perform requirements analysis, 2) derive a specification based on the requirements, 3) design and implement the system, 4) test the system. The hypothesis is that the system fulfills the specification and thereby meets the requirements.

As previously stated, our goal is to compare inherent protocol approach overhead. Using Denning's approach, we

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/MILCOM.2018.8599786

have identified that the requirements are efficient information dissemination and a robust solution. NATO's specification for publish/subscribe in the SOA baseline [2] 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 [25]. For the sake of this paper 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.

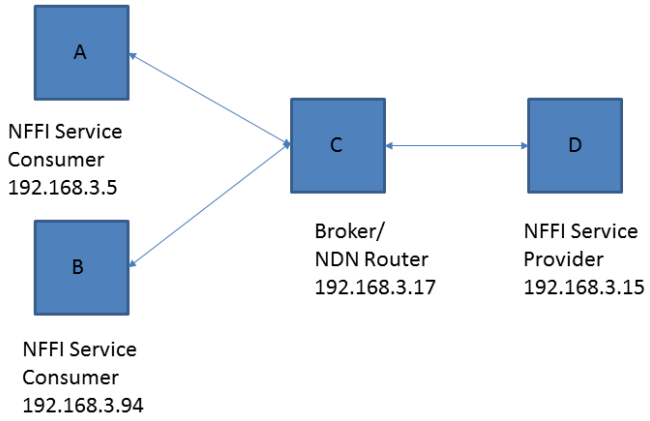

Fig. 1. Testbed

IV. Test setup

The test setup includes four nodes, two service consumers, one service provider and one broker/router (see Figure 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 [7], [8].

A. Nodes

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.

B. Brokers

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 [29].

For MQTT we used the open source mosquitto broker which is freely available [9]. For NDN we used the ndn-cxx implementation which is also open source and freely available [10].

C. 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 tactical network deployment. We captured all network traffic in the central broker node, analyzing it with respect to the number of packets sent by the different solutions.

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.

2) MQTT: The MQTT provider and consumers were implemented using the mqtt-client-1.7-uber library [11]. 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.

3) NDN: The NDN provider and consumers were implemented using the jndn library [12]. 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 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.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/MILCOM.2018.8599786

TABLE I
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

V. TEST 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.

A. Results analysis

Table I shows the results for our small testbed shown in Figure 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. This results in an abundance of small packets on the network. Figure 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.

A similar message sequence diagram is shown in Figure 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, as described in Section II-C, 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 keep alive 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.

NDN does not maintain any connections for data exchange. Instead, information is requested periodically. One request is needed for each produced data element. Figure 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

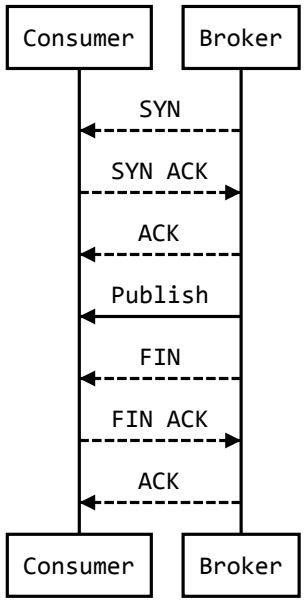

Fig. 2. Packet flow between the broker and consumer nodes using WS-Notification. Each request establishes a new TCP session, and hence many TCP packets are required to set up and tear down connections. Data passing from the provider to the broker (which occurs prior to the sequence shown above) involves the same procedure.

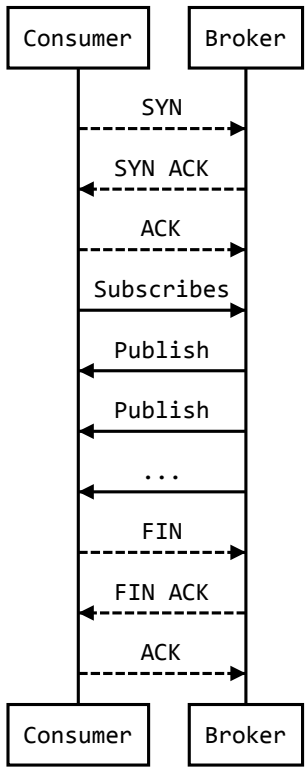

Fig. 3. Packet flow between broker and consumer using MQTT. MQTT, in contrary to WS-Notification, only initializes one TCP connection and keeps it alive until the consumer terminates the relationship to the broker.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/MILCOM.2018.8599786

needs handshaking to setup and tear-down. The number of MQTT packets should therefore be higher. The results in Table I 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 consist 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.

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 I shows a difference in average packet size for the three evaluated methods. The optimal packets 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].

VI. Related Work

WS-Notification has been shown to work in tactical broadband networks. Typically, it performs well if you have a network with 1 Mbps throughput or more, which has been shown in previous experiments. Examples include WS-Notification over Rinicom's PodNode radios [13], [14] and using the WM600 radios from Kongsberg [4]. For resource constrained networks, on the other hand, other solutions must be applied.

An example of an efficient but proprietary approach to publish/subscribe in tactical networks is the Mist publish/subscribe protocol, which relies on opportunistic communication and the high mobility of nodes to convey information [15]. It works best in a dense network, or possibly with one or more nodes functioning as message ferries between clusters of nodes. Also, unlike the standard publish/subscribe approaches, like WS-Notification and MQTT, there are no security measures that can be easily enabled for the protocol.

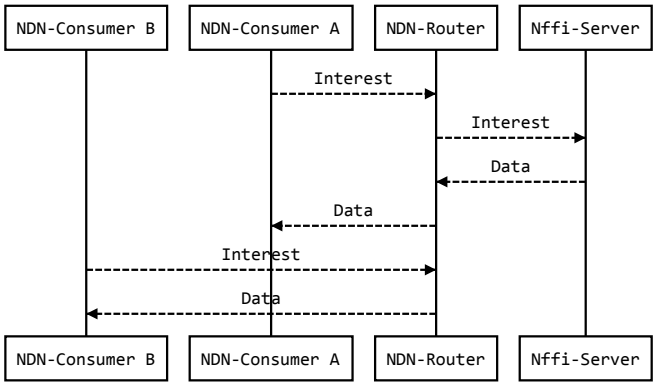

Fig. 4. Packet Flow in an NDN network with two consumers and one producer.

In [20] Carzaniga et al. 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.

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 [25] 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 do in this paper, since it shows that even if we here recommend to use something else than NATO's protocol of choice in tactical networks, it is doable to provide the same data to others using WS-Notification through mediation gateways.

VII. Conclusion and future work

In this paper 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 to low-capacity tactical 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 tactical networks is that the standards rely on a central broker to offload such tasks as subscription handling and message

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/MILCOM.2018.8599786

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 test proved to be the most efficient. Furthermore, the NDN implementation is broker-less by design, and could thus be better suited in a tactical 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 in this paper. This scalability, and especially the effects caching have on performance, should be investigated further. For future work we plan to perform a large scale experiment in context of the NATO research task group IST-150 "NATO Core Services Profiling for Hybrid Tactical Networks" using a network emulator and the *Anglova scenario* [21]. The final steps will involve using NDN in an actual tactical network, supporting C2 services.

ACKNOWLEDGMENT

The authors would like to thank Ketil Lund and Trude H. Bloebaum for proofreading and providing useful comments during the writing process.

REFERENCES

[1] P. Bartolomasi, T. Buckman, A. Campbell, J. Grainger, J. Mahaffey,R. Marchand, O. Kruidhof, C. Shawcross and K. Veum. NATO network enabled capability feasibility study. Version 2.0, October 2005.

[2] 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 Unclassified releasable to EAPC/PFP, 11 November 2011.

[3] OASIS. Reference model for service oriented architecture 1.0 OASIS standard, 12 october 2006. C. Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter F. Brown, and Rebekah Metz (eds.), http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

[4] Trude H. Bloebaum and Frank T. Johnsen, Evaluating publish/subscribe approaches for use in tactical broadband networks. IEEE Military Communications Conference (MILCOM), Tampa, FL, USA, October 2015.

[5] OASIS. MQTT Version 3.1.1. OASIS Standard 29 October 2014. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf

[6] Frank T. Johnsen, Trude H. Bloebaum and Peter-Paul Meiler. Improving Integration between Tactical and HQ Levels by making SOA applicable on the Battlefield. ICCRTS 2017.

[7] Manas Pradhan, Christoph Fuchs, and Frank T. Johnsen. A Survey of Applicability of Military Data Model Architectures for Smart City Data Consumption and Integration. IEEE 4rd World Forum on Internet of Things (WF-IoT) 2018, February 2018, Singapore

[9] Eclipse. Eclipse Mosquitto^{TM} An open source MQTT broker. https://mosquitto.org/

[8] Marco Manso, Marianne R. Brannsten, and Frank T. Johnsen. A Smart Devices Concept for Future Soldier Systems. ICCRTS 2017.

[10] Named Data Networking. ndn-cxx: NDN C++ library with eXperimental eXtensions. http://named-data.net/doc/ndn-cxx/current/

[11] Fusesource. An MQTT client. https://github.com/fusesource/mqtt-client

[12] Named-data. NDN Client Library for Java. https://github.com/named-data/jndn

[13] Marco Manso et al. SOA Experiments on Wireless Broadband Mobile Networks in the Tactical Domain 20th International Command and Control Research and Technology Symposium (ICCRTS), 2015.

[14] Marco Manso et.al. SOA and Wireless Mobile Networks in the Tactical Domain: Results from Experiments IEEE Military Communications Conference (MILCOM), Tampa, FL, USA, October 2015

[15] Magnus Skjegstad et al. Mist: A Reliable and Delay-Tolerant Publish/Subscribe Solution for Dynamic Networks 5th IFIP International Conference on New Technologies, Mobility and Security (NTMS 2012), Istanbul, Turkey, 7-10. May 2012

[16] P.T. Eugster et al. (2003) The Many Faces of Publish/Subscribe ACM Computing Surveys, 35(2), June 2003.

[17] OASIS. Web Services Base Notification 1.3 (WS-BaseNotification). OASIS Standard, 1 October 2006. http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf

[18] OASIS. Web Services Brokered Notification 1.3 (WS-BrokeredNotification). OASIS Standard, 1 October 2006. http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf

[19] OASIS. Web Services Topics 1.3 (WS-Topics). OASIS Standard, 1 October 2006. http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf

[20] Carzaniga, A., Papalini, M., and A. Wolf. Content-based Publish/Subscribe Networking and Information-centric Networking. Proceedings of the ACM SIGCOMM workshop on Information-centric networking, ACM, 2011

[21] Suri, N., Hansson, A., Nilsson, J., Lubkowski, P., Marcus, K., Hauge, M., Lee, K., Buchin, B., Misirlioglu, L., and Peuhkuri, M. A Realistic Military Scenario and Emulation Environment for Experimenting with Tactical Communications and Heterogeneous Networks. In Proceedings of the IEEE International Conference on Military Communications and Information Systems (ICMCIS 2016), Brussels, Belgium, May 23rd-24th, 2016.

[22] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. Named Data Networking. SIGCOMM Comput. Commun. Rev. 44, 3 (July 2014), 6673

[23] L. Zhang et al. Named data networking (NDN) project. NDN, Technical Report NDN-0001, 2010. http://named-data.net/techreport/TR001ndn-proj.pdf

[24] P. J. Denning, D. Comer, D. Gries, M. C. Mulder, A. B. Tucker, A. J. Turner, and P. R. Young. Computing as a discipline. Commun. ACM, 32(1):923, 1989.

[25] Eirik Bertelsen et al. Federated Publish/subscribe Services. 9th IFIP International Conference on New Technologies, Mobility & Security 26 to 28 February 2018. Paris, France.

[26] C4ISR Technology & Human Factors (THF) Branch, Allied Command Transformation (ACT). The C3 Taxonomy. Technical report, 2016. Document generated from the ACT Enterprise Mapping Wiki in November 2016.

[27] Parc. Content Centric Networking (CCNx). https://github.com/ProjectCCNx

[28] Allied Command Transformation (ACT). Federated Mission Networking. http://www.act.nato.int/fmn

[29] Trude H. Bloebaum and Frank T. Johnsen. CWIX 2014 core enterprise services experimentation. FFI-report 2014/01510. https://www.ffi.no/no/Rapporter/14-01510.pdf

[30] Jun Yin, Xiaodong Wang and D. P. Agrawal, "Optimal packet size in error-prone channel for IEEE 802.11 distributed coordination function," 2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733), 2004, pp. 1654-1659 Vol.3. doi: 10.1109/WCNC.2004.1311801