Toward an Adaptive Data Distribution Service for Dynamic Large-Scale Network-Centric Operation and Warfare (NCOW) SystemsTo be sumitted to the 2008 Military Communications Conference

Nanbor Wang
Tech-X Corporation
nanbor@txcorp.com

Douglas C. Schmidt
Department of Electrical Engineering
and Computer Science
Vanderbilt University
schmidt@vanderbilt.edu

Hans van't Hag
PrismTech Corporation
hans.vanthag@prismtech.com

Angelo Corsaro
PrismTech Corporation
angelo.corsaro@prismtech.com

Abstract

To achieve the goal of information dominance, the DoD has adopted the doctrine of net-centric operations and warfare (NCOW). The Global Information Grid (GIG), Future Combat System (FCS), C2 Constellation, and FORCEnet are examples of net-centric operations where multiple systems-of-systems integrate thousands of platforms, sensors, decision nodes, weapons, and warfighters through heterogeneous wire-line and wireless networks. NCOW provides superior collection, fusion, analysis, and use of information to help the DoD exploit information superiority and achieve strategic and tactical goals.

Standard-based publish/subscribe (pub/sub) middleware, such as the Object Management Group (OMG)'s Data Distribution Service (DDS), is a key enabling technology to build and evolve large-scale and long-lived distributed, real-time, and embedded NCOW systems. DDS is particularly relevant since it is the only standards-based pub/sub middleware that can satisfy the stringent quality-of service (QoS) requirements for a wide variety of tactical net-centric applications. The current OMG DDS specification, however, does not define services that (1) enable large-scale applications to execute in highly dynamic wide-area network environments, where information producers and consumers join and leave the information exchange and (2) rigorously maintain the necessary QoS required by NCOW systems.

This paper provides three contributions to enabling dynamic NCOW applications to utilize standard-based pub/sub middleware effectively. First, it investigates and surveys key challenges in building dynamic NCOW applications, using DDS as an example. Second, it examines existing DDS standards and DDS implementations and identifies their pros and cons in the context of NCOW. Finally, we propose an adaptive discovery service framework that enables the application of DDS in large-scale and highly dynamic data-critical NCOW systems over diverse operating environments.

D.2.13Software EngineeringReusable SoftwareReuse Models

Design, Performance, Reliability

Data Distribution Service, Scalability, Real-time, Secure Communication, Large-Scale Distributed Systems, Adaptive Middleware, Gap Analysis

1. Introduction

To achieve the goal of information dominance, the DoD has adopted the doctrine of Net-Centric Operations and Warfare (NCOW) [3, 10, 7]. The Global Information Grid (GIG), Future Combat System (FCS), C2 Constellation, and FORCEnet are examples of NCOW environments. These environments integrate thousands of platforms, sensors, decision nodes, weapons, and warfighters through heterogeneous wireline and wireless networks.

Recent advances in standards-based COTS QoS-enabled publish/subscribe (pub/sub) middleware are well-suited to the information distribution needs of delivering the right information at the right place and the right time in net-centric systems. In particular, the OMG's Data Distribution Service (DDS) [9] is emerging as a promising technology for

net-centric operations. The DDS standard offers advanced features, such as data durability, priority, update frequency, and data delivery latency, to satisfy the stringent QoS requirements for a wide variety of tactical net-centric applications [12].

System functional and QoS requirements can put different constraints and demands on how a specific DDS system can/-should be built. The diversity of execution environments and QoS requirements of net-centric systems-of-systems can impose different requirements on the underlying DDS platform. Moreover, increased security threats in WAN environments make information assurance (IA), such as secure communication, user authentication, access control, and data integrity, a key capability a DDS infrastructure must support. It is essential for DDS implementations to provide the required QoS and performance trade-offs scalably and adaptively based on these constraints and requirements.

Most DDS implementations, including PrismTech's OpenSplice DDS, RTI's RTI DDS, and OCI's OpenDDS, use multicast mechanisms to enable efficient and reliable data distribution to multiple recipients. Unfortunately, for existing DDS implementations, setting up underlying multicast mechanisms require static preconfiguration of nodes, which is incompatible with the needs of a highly dynamic WAN environment, such as the GIG. One key missing feature in supporting dynamic DDS usage over a WAN-based net-centric environment is a discovery framework that can bootstrap the DDS infrastructure by dynamically (re)configuring the underlying multicasting mechanisms to deliver the right information to the right place(s) at the right time.

To support the mission-critical systems in a diverse WAN-environment, a discovery service for DDS must operate with real-time, fault-tolerance, and IA QoS assurances [4]. Alternative discovery mechanisms are thus needed to assist the DDS infrastructure to support these QoS requirements based on the combination of system requirements and environmental constraints. This paper proposes a framework-based approach that allows the deployment of alternative discovery strategies based on specific system's QoS requirements, such as time constraints and reliability, and the type of network(s) upon which it operates.

The remainder of this paper is organized as follows: Section 2 gives an overview of OMG's DDS; Section 3 illustrates the challenges in applying DDS in a dynamic NCOW environment; Section 4 presents how an adaptive discovery framework can help DDS address these challenges; Section 5 reviews how a DDS discovery service can help provision the information assurance supports of NCOW; and Section 6 presents concluding remarks.

2. Overview of DDS

The OMG Data Distribution Service (DDS) specification defines a data-centric communication standard for wide variety of computing environments, ranging from small networked embedded systems up to large-scale information backbones. DDS provides a scalable, platform-independent, and location-independent middleware infrastructure to connect information producers to consumers. DDS also supports many quality-of-service (QoS) properties, such as asynchronous, loosely-coupled, time-sensitive and reliable data distribution at multiple layers (*e.g.*, middleware, operating system, and network).

At the core of DDS is the Data-Centric Publish-Subscribe (DCPS) model, which defines standard interfaces that enable applications running on heterogeneous platforms to write/read data to/from a global data space in a net-centric system. Applications can use this global data space to share information with other applications by declaring their intent to publish data that is categorized into one or more topics of interest to participants. Similarly, applications can use this data space to access topics of interest by declaring their intent to become subscribers. The underlying DCPS middleware propagates data samples written by publishers into the global data space, where it is disseminated to interested subscribers. The DCPS model decouples the declaration of information access intent from the information access, thereby enabling the DDS middleware to support and optimize QoS-enabled communication.

The following DDS entities (also shown in Figure 1) are involved in creating and using a DCPS-based application:

- **Domain** – DDS applications send and receive data within a domain, which provides a virtual communication environment for participants having the same domain id. This environment also isolates participants associated with different domains, *i.e.*, only participants within the same domain can communicate, which is useful for isolating and optimizing communication within a community that shares common interests.

- **Domain Participant** – A domain participant is an entity that represents a DDS application's participation in a domain. It serves as factory, container, and manager for the DDS entities described below.

- **Data Writer and Publisher** – Applications use data writers to publish data values to the global data space of a domain. A publisher is created by a domain participant and used as a factory to create and manage a group of data writers that publish their data in the same logical partition within the global data space.

Data writers and publishers have related QoS policies that drive their behavior as DDS entities.

- **Subscriber and Data Reader** – Applications use data readers to receive data. A subscriber is created by a domain participant and used as a factory to create and manage data readers. A data reader can obtain its subscribed data via two approaches, as shown in Figure 2: (1) listener-based, which provides an asynchronous mechanism to obtain data via callbacks in a separate thread that does not block the main application and (2) waitset-based, which provides a synchronous mechanism that blocks the application until a designated condition is met.
- **Topic** – A topic connects a data writer with a data reader, *i.e.*, communication does not occur unless the topic published by a data writer matches a topic subscribed to by a data reader. Communication via topics is anonymous and transparent, *i.e.*, publishers and subscribers need not be concerned with how topics are created nor who is writing/reading them since the DDS DCPS middleware manages these issues.

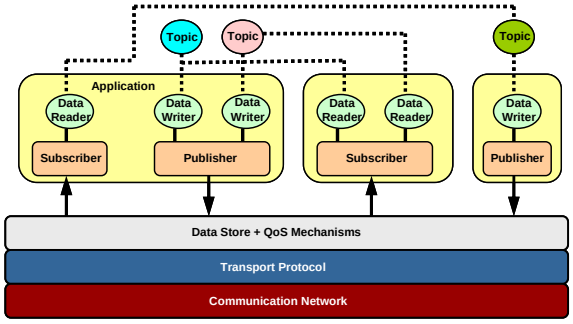


Figure 1. Architecture of DDS

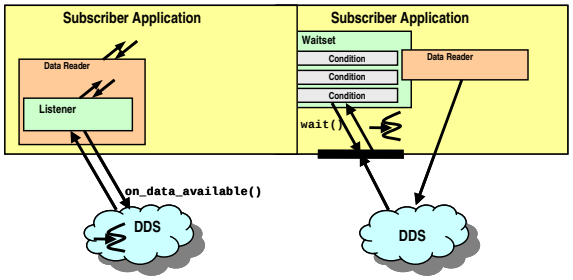
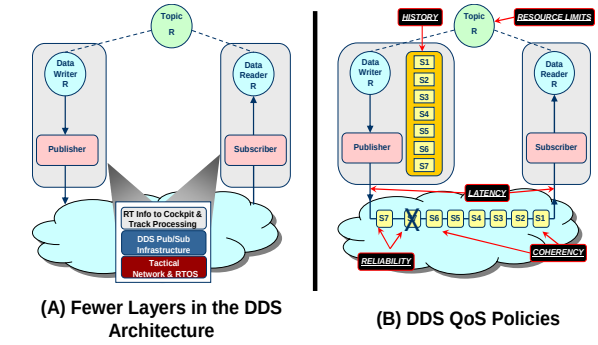


Figure 2. Listener- and Waitset-based Notification

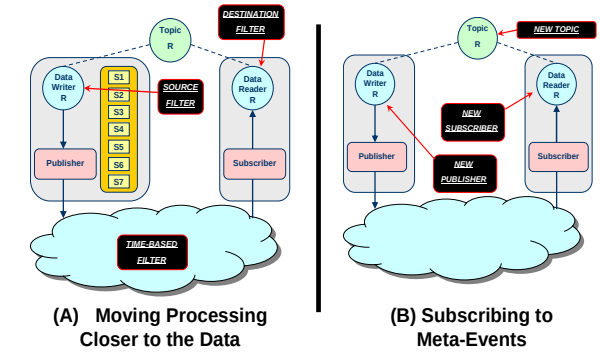
Below, we describe the benefits of DDS relative to client/server-based SOA middleware (such as CORBA, .NET, J2EE, and SOAP-based Web Services) and conventional pub/sub middleware (such as the CORBA Event Service and the Java Message Service).



(A) Fewer Layers in the DDS Architecture

(B) DDS QoS Policies

Figure 3. Optimizations and QoS Capabilities of DDS



(A) Moving Processing Closer to the Data

(B) Subscribing to Meta-Events

Figure 4. Listener- and Waitset-based Notification

Figures 3 and 4 show DDS capabilities that make it better suited than conventional middleware platforms as the basis of net-centric systems. Figure 3(A) shows that DDS has fewer layers in its architecture than conventional Service-Oriented Architecture (SOA) standards, which minimizes its latency and jitter significantly, as shown in Section 4. Figure 3(B) shows that DDS supports many QoS policies, such as:

1. The durability of information, *i.e.*, if and how long published information is to be maintained within the global information space for late joining subscribers.

2. The degree and scope of coherency for information updates, *i.e.*, whether a group of updates can be received as a unit and in the order in which they were sent.
3. The frequency of information updates, *i.e.*, the rate at which updated values are sent and/or received.
4. The maximum latency of data delivery, *i.e.*, a bound on the acceptable interval between the time data is sent and received
5. The priority of data delivery, *i.e.*, the priority used by the underlying transport to deliver the data.
6. The reliability of data delivery, *i.e.*, whether missed deliveries will be retried.
7. How to arbitrate simultaneous modifications to shared data by multiple writers, *i.e.*, to determine which modification to apply.
8. Mechanisms to assert and determine liveliness, *i.e.*, whether or not a publish-related entity is active.
9. Content awareness, *i.e.*, fine-grained expressions of interest by data receivers which allows filtering and/or querying information based upon its actual content
10. The duration of data validity, *i.e.*, the specification of an expiration time for data to avoid delivering "stale" data.
11. The depth of the "history" included in updates, *i.e.*, how many prior updates will be available at any time, *e.g.*, "only the most recent update," "the last n updates," or "all prior updates".

These QoS policies can be configured at various levels of granularity (*i.e.*, topics, publishers, data writers, subscribers, and data readers), thereby allowing application developers to construct customized contracts based on the specific QoS requirements of individual entities. Since the identity of publishers and subscribers are unknown to each other, the DCPS middleware is responsible for determining whether the QoS policies offered by a publisher are compatible with those required by a subscriber, allowing data distribution only when compatibility is achieved.

Figure 4(A) shows how DDS can migrate processing closer to the data source, which reduces bandwidth in resource-constrained network links. Figure 4(B) shows how DDS enables clients to subscribe to meta-events that they can use to detect dynamic changes in network topology, membership, and QoS levels. This mechanism helps net-centric systems adapt to operating environments that change continuously.

3. Challenges in Applying DDS for Large-scale Distributed Applications

Most DDS implementations, including PrismTech's OpenSplice DDS, RTI's RTI DDS, and OCI's OpenDDS, use multicast mechanisms to enable efficient and reliable data distribution to multiple recipients. Unfortunately, configuring underlying multicast mechanisms in existing DDS implementations requires static preconfiguration of nodes, which is incompatible with the needs of a highly dynamic wide-area network (WAN) environments needed to support NCOW. One key missing feature in supporting dynamic DDS usage over a WAN-based net-centric environment is a discovery framework that can bootstrap the DDS infrastructure by dynamically (re)configuring the underlying multicasting mechanisms to deliver the right information to the right place(s) at the right time.

Specifically, operating a pub/sub such as DDS over the WAN must address the following challenges:

1. **Increased dynamism** – A key strength in DDS is its scalability. Not every node participating in an information exchange needs to know about every other node in the exchange, or the topology of the overall connection and the exact flow of information. In a net-centric environment, however, new and arbitrary nodes may need to participate in information exchange of a certain topic. Without *a priori* knowledge of how to join an existing multicasting network of an existing DDS system, a new node may have trouble joining the existing information exchange and start a new, separate information distribution system of the same topic. Conversely, the dynamic nature of net-centric systems makes it hard to preconfigure all systems as it will make the overall system susceptible to minor changes, such as node joining and leaving multicast groups.

2. **Predictability and dependability** – Mission-critical applications often have stringent timing constraints. Certain stimuli must produce proper responses within a certain time bound. Information consumers with such requirements need to start reacting to incoming information as soon as they subscribe to the information. Likewise, information producers expect the information they publish to be handled promptly once they start publishing it. Likewise, mission-critical applications must be able to tolerate faults. Nodes participating in an information exchange may leave the network without notice due to a cyber attack or hardware failure. Likewise, new nodes must be able to join an information exchange regardless of faults.

3. **More diversified environment** – Many different types of networking technologies and hardware platforms can coexist in WAN-based net-centric systems. Since net-centric operations strive to integrate the overall warfighting entities over the WAN, a pub/sub-based net-centric system can span multiple types of networks, including high-bandwidth, low-latency fiber optic connections; conventional enterprise LANs; and high-latency, narrow-bandwidth, and unreliable wireless links. Similarly, net-centric

systems may need to connect diverse hardware devices with different resources and computing power such as PDAs for warfighters, different sensors, and high-performance target tracking clusters. DDS implementations must therefore satisfy the functional and systemic requirements of the system, even in such demanding environments.

4. **Increased security threats** – Unlike LAN-based DDS deployments (such as on a single airplane), where IA can typically be guaranteed via strict physical protection to network infrastructure and preconfigured encryption tokens, providing IA in QoS-enabled pub/sub middleware in a WAN environment presents several new challenges. First, the confidentiality of the information is threatened when it is transferred over unprotected communication links that connect to unauthorized parties. Second, the integrity of the information could be vulnerable due to the large numbers of potential erroneous or malicious information emitters. Finally, the availability of information may be hindered by various cyber attacks, such as distributed denial-of-service.

4. Requirements for an Adaptive Discovery Framework

To satisfy the consraints mentioned in Section 3, there is a need to separate the discovery mechanism from the DDS implementation, thereby allowing DDS users to employ diverse discovery strategies based on the operational and environmental constraints. At the highest level, the key functionality of the discovery service in the context of DDS infrastructure is to inform internal DDS entities that share a common interest and wish to exchange data, the existence of each other and provide hints on how they can communicate with each other. Key requirements that must be addressed by the design and implementation of an adaptive discovery service include:

• **Efficiently associating DDS entities with shared mutual interests.** There are many levels and granularities where DDS entities can share mutual interests and exchange different data. Different DDS implementations employ different strategies and thus different programming modules (such as domain participants in a Real-Time Publish-Subscribe (RTPS)-based [5] implementation or daemon processes in a hierarchical message routing implementation) to establish the multicasting connections. Regardless of the implementation strategies, the modules responsible for establishing the multicasting connections within the same domain share common interests regarding what is available in this domain, *e.g.*, what topics are available. These modules must therefore exchange information with each other to identify common interests, *i.e.*, various DDS built-in and user-defined topics.

Determining and matching common interests, however, not only involves a common topic, but also various QoS properties associated with the topic. For example, a data writer can *offer* certain QoS properties, whereas a data reader can *request* certain QoS properties. Moreover, even when a data writer and reader share an identical topic, these two entities will communicate with each other only when the offered and requested QoS properties are compatible.

Other QoS properties not directly associated with topics can also affect their mutual interests. For example, domain participants hosting topics can have a list of partition QoS properties that segments the global data space further. Only topics whose domain participants share at least one common partition are considered mutually interested in each other. Likewise, other DDS built-in topics, such as USER_DATA that is often used to attach user credentials for authentication, can affect the interests among data readers and data writers and their ability to communicate.

Providing a discovery service to manage communication between domain participants is the most important and fundamental need for a DDS infrastructure over a dynamic WAN environment. Only when all relevant participants can communicate with each other can they establish the next level of discovery service, *i.e.*, associating topics and their data readers and data writers. Building such discovery service data can be performed within the previously established domain participant network. All the associations will be a subnet or the domain participant network.

• **Supporting robust communication hints.** Other than providing information on other entities sharing a common interest, a WAN-DDS discovery service should also provide robust communication hints on how these entities can be reached efficiently. The actual information of such communication hints provided by the discovery service depends on the actual multicasting mechanisms a DDS implementation employs. For example, a combination of Internet address, port number describe the most basic information for reaching an entity using low level transports such as IP Multicast or overlay multicast networks using unicast over virtual multicast trees. More advanced multicast mechanisms used by a DDS implementation may provide better QoS support, such as priority, and can benefit from more comprehensive communication hints, *e.g.*, providing information on multiple communication endpoints for different communication priorities.

• **Seamless integration with standards.** To facilitate message exchange between publishers and subscribers that use different implementations of DDS, a separate OMG DDS Interoperability Wire Protocol Specification [8] defines the network protocol based on the RTPS wire protocol specification [5] for Fieldbus, which is a networking

mechanism for industrial control and measurement based on LAN. All DDS implementations must support this specification when they interoperate with other implementations. RTPS-based protocols can provide good performance and predictability for applications deployed in a LAN environment or relatively controlled network environment.

Discovery services are also available in the RTPS protocol. In particular, the RTPS specification defines a discovery service for domain participants called the *Participant Discovery Protocol* (PDP) and another for matching data readers and data writers called the *Endpoint Discovery Protocol* (EDP). Two specific interoperable protocols (SPDP and SEDP) are layered atop RTPS using special built-in topics and data readers/writers. These discovery protocols present potential scalability concerns due to the amount of data exchanges periodically and the number of multicast groups required. They also require the need to preconfigure nodes to bootstrap the discovery service. The scalability issue of such decentralized peer-to-peer "heavyweight groups" approach is well-documented [13, 11] in the group communication research literature.

Taking into account of other characteristics of discovery services—together with the two key requirements of DDS discovery service describes above—we have identified the following key features needed for a WAN-DDS discovery service once information about other aspects of systems are available:

- The discovery service must allow the use of alternative discoverying strategies to adapt to the specific operating environment. For example, a hierarchical discovery service using a fixed set of servers may scale well for DDS applications running over an enterprise network. Alternatively, a discovery service using peer-to-peer-based protocols, such as the Simple Service Discovery Protocol (SSDP), may better serve DDS applications running over mobile ad-hoc network.

- The discovery service must be integrated with the underlying DDS configuration system seamlessly to allow dynamic reconfiguration of multicasting network. This integration allows a DDS applications to avoid the need of a multicasting network that incorporates all possible participants by establishing multiple smaller overlaying multicasting networks.

- An adaptive discovery service framework needs to address the scalability issue in RTPS protocol by limiting the messages for new participant to join a new DDS domain.

5. Informantion Assurance Support in the Discovery Service

Information assurance (IA) has become a key issue for NCOW applications. Key IA capabilities include, identify authentication, access control, information integrity and information encryption. Unlike a LAN where physical access can easily be enforced, the openness of a WAN requires careful integration of IA with the pub/sub middleware. Since discovery services provide the bootstrapping mechanisms for pub/sub middleware in a WAN environment, it is essential that IA support be an integral part of the discovery service. Such integration not only protects basic information of what DDS entities (such as topics, publishers, and subscribers) are on the network, but also helps provision IA support in the pub/sub middleware.

The integration described above is consistent with our approach of having the discovery service interacts with the underlying multicast mechanisms and configure an autonomous overlay network. The discovery framework should provide the overall infrastructure mechanism to ensure that the security enforcement mechanisms cannot be bypassed, are always invoked, and are tamper-proof. This requirement applies to all the operations of pub/sub middleware, including discovery, writing (publishing), and reading (subscribing) of data. The distribution mechanism of the pub/sub middleware must also support multi-layered security to enforce information compartment based on different security levels of application and user identity.

We believe DDS middleware should support IA through the following approaches:

- Industrial standards such as public key certificates and X.509 public key infrastructure (PKI), are promising solutions. Conversations with our Phase II subcontractor, Lockheed Martin MS2, also show that they are the foundation of current DoD IA infrastructure.

- Discovery service should be the entry point of the overall group communication IA infrastructure by seamless integration with existing IA systems. A new entity wishing to join the data exchange using DDS should first identify itself with proper key certificate. The DDS service can then utilize existing IA service to examine the identity and decide whether it will be allowed to use the discovery service. Industrial standard such as XACML [1] and SAML [2] are good potential protocol for administrating and enforcing access control.

- After being admitted to the discovery service, an entity can acquire a session identity proxy for the identification purpose during the session. Likewise, a topic reader or writer will need to be able to identify itself

and get admitted to a node daemon or a DomainParticipant before it can exchange data over the DDS infrastructure.

- Group keys are used for secure communication. Group key management is the key in secure group communication systems. The Tree-based Group Diffie-Hellman (TGDH) [6] protocol is a good candidate, among others. Group key management, however, can incur high overhead triggered by the membership changes. A layered group communication system may therefore provide better scalability than a peer-to-peer group communication system, such as RTPS.
- The certificate proxy is important even after an entity has joined the data exchange. Unlike typical static pub/sub systems where a subscriber doesn't care where a piece of information is published, many DoD applications do care about the authenticity and the origin of the information. Warfighters care less about how they acquire the information than they do about the origin of the information and who processed the information. It is therefore important for the DDS infrastructure to support some digital signing capability.

6. Concluding Remarks

This paper motivated the need to enhance DDS with an adaptive discovery services framework for large-scale NCOW applications and also surveyed the key requirements and features of such framework. A range of discovery mechanisms are needed for different environmental constraints and systemic requirements to provide the proper trade-offs in performance, resource usage, dependability, and predictability.

Based on our investigation, the existing RTPS-based DDS interoperability wire protocol specification has significant scalability limitations in terms of the number of house-keeping messages with regards to the number of participants (readers/writers), topics, the number joining and leaving the data exchange, and the ability to support prioritized message delivery. The same scalability issue also applies to group key management needed for secure communication in IA. In comparison, a layered messaging protocol is both much more scalable and predictable although it imposes slight overhead on each message delivery compared to RTPS-based messaging protocol.

We described the validity of a framework-based approach where multiple discovery mechanisms can be plugged into a DDS implementation to support various different operating environments and group communication messaging protocols. We also showed how the discovery framework can be implemented to collaborate with the existing DoD IA infrastructure to set up the IA support in a DDS environment. In future work, we will implement the various features needed to realize such an adaptive discovery framework. We also plan to standardize the resulting discovery framework interfaces and the RTPS protocol enhancement via the OMG standardization process.

Acknowledgments

The research presented in this paper is partially based upon work supported by the US. Navy Sea Systems CommandX under Contract No. M65538-07-M-0113. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the Navy Sea Systems Command.

References

[1] Oasis security services, extensible access control markup language (xacml) tc. `http://www.oasis-open.org/committees/xacml`.

[2] Oasis security services, security assertion markup language (saml) tc. `http://www.oasis-open.org/committees/security`.

[3] D. S. Alberts, J. J. Garstka, and F. P. Stein. *Network Centric Warfare – Developing and Leveraging Information Superiority*. The Command and Control Research Program Publication, `www.dodccrp.org`, 2000.

[4] J. Hoffert, D. Schmidt, and A. Gokhale. A QoS Policy Configuration Modeling Language for Publish/Subscribe Middleware Platforms. In *Proceedings of International Conference on Distributed Event-Based Systems (DEBS)*, pages 140–145, Toronto, Canada, June 2007.

[5] IEC. *IEC/PAS 62030 (2004-11): Digital data communications for measurement and control – Fieldbus for use in industrial control systems*, chapter Section 2: Real-Time Publish-Subscribe (RTPS) Wire Protocol Specification Version 1.0, pages 97–159. International Electrotechnical Commission, Geneva, Switzerland, 1.0 edition, 2004. It is intended that the content of this PAS will be incorporated in the future new editions of the various parts of IEC 61158 series according to the structure of this series.

[6] Y. Kim, A. Perrig, and G. Tsudik. Tree-based Group Key Management. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):60–96, Feb. 2004.

[7] Network centric operations and warfare (ncow) wiki. `http://ncow.nps.edu/wiki/index.php/Main_Page`.

[8] Object Management Group. *The Real-time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification*. Object Management Group, OMG Document ptc/06-08-02 edition, Aug. 2006.

[9] Object Management Group. *Data Distribution Service for Real-time Systems Specification*, 1.2 edition, Jan. 2007.

[10] *The Implementation of Network Centric Warfare*. Office of Force Transformation, The U.S. Department of Defense, 2003.

[11] L. Rodrigues, R. Guerraoui, and A. Schiper. Scalable atomic multicast. In *Proceedings of the 7th IEEE International Conference on Computer Communications and Networks (IC3N'98)*, pages 840–847, Lafayette, Louisiana, USA, 1998.

[12] D. C. Schmidt, A. Corsaro, and H. V. Hag. Addressing the Challenges of Tactical Information Management in Net-Centric Systems with DDS. *CrossTalk - The Journal of Defense Software Engineering*, Mar. 2008.

[13] Yair Amir and Claudiu Danilov and Jonathan Robert Stanton. A Low Latency, Loss Tolerant Architecture and Protocol for Wide Area Group Communication. In *Proceedings of International Conference on Dependable Systems and Networks*, pages 327–336, June 2000.