

# A Study on the Usages of DDS Middleware for Efficient Data Transmission and Reception

Yeongwook Jeong\*

## Abstract

Data Distribution Service(DDS) provides the communications service programmers need to distribute time-critical data between embedded and/or enterprise devices or nodes. In this paper, I propose efficient methods for transmitting and receiving messages of various characteristics in real-time using DDS middleware. For high-frequency characteristic data, I describe several DDS packet types and various default and extended DDS QoS policies. In particular, the batching method is probably the best solution when considering several performance aspects. For large-capacity characteristic data, I will show a method using extended DDS QoS policies, a segmentation and reassembly method, and transmitting and receiving a large-capacity data with low priority method considering network conditions. Finally, I simulate and compare the result of performance for each methods. This results will help determine efficient methods for transmitting and receiving messages of various characteristics using DDS middleware.

▶ Keyword: Data Distribution Service, High-Frequency, Large-Capacity, Performance, Quality of Service

## I. Introduction

다양한 산업 분야에서 사용되는 기기종의 많은 장치들은 분산 네트워크를 이용하여 실시간으로 끊임없이 데이터를 분배하고 공유할 뿐만 아니라 시간의 경과에 따라 데이터를 주고받는 참여자들의 동적인 참여와 탈퇴 과정을 쉽게 처리할 수 있도록 데이터 중심의 통신 기술(Data-Centric Communication)을 요구한다. 데이터 중심 통신을 위한 많은 기술들이 존재하는데 그 중 발간(publish)/구독(subscribe) 기술이 다양한 응용 분야에서 많이 사용되고 있으며, 이를 이용하는 대표적인 프로토콜이 DDS(Data Distribution Service)이다[1-4].

DDS는 OMG(Object Management Group) 기구[3]에서 제정한 미들웨어로써 분산 환경에서 네트워크 데이터 도메인에 자유로운 참여나 탈퇴가 가능하여 노드 구성의 유연성을 극대화 해주고, 발간/구독 기술을 기반으로 하여 실시간으로 데이터를 공유 및 배포하는 기능을 제공한다. 국방, 항공, 우주 산업, 교통, 로봇, 증권, 의학, 금융거래 및 다양한 분야의 자동화 시스템 등의 많은 분야에서 사용되고 있으며 미국 RTI 社[4]의 RTI Connex가 전 세계 상용 DDS 시장의 대부분을 점유하고 있다. 국내에서도 많은 연구소와 기업에서 DDS를 자체적으로 개발하여 사용하는 추세이다. 특히, 한화시스템에서 개발한 SMART DDS는 방공C2A체계, FFX-II 및 PKX-B 전투체계, 다중데이터링크 통합처리기 등의 무기 또는 체계관련 국방 사업 분야에 활발하게 적용되어 외산 DDS 제품을 대체하여 사용되고 있다[5].

DDS를 이용하여 사용자 데이터를 송수신할 때, topic 크기와 DDS QoS(Quality of Service) Policy 튜닝은 성능에 영향을 끼치는 대표적인 요소들이다. Topic은 DDS domain에서 유

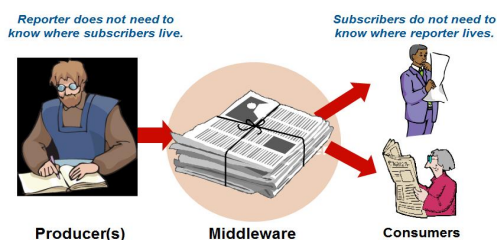


Fig. 1. Publish/Subscribe Model

• First Author: Yeongwook Jeong, Corresponding Author: Yeongwook Jeong  
\*Yeongwook Jeong (yeongwook.jeong@hanwha.com), Naval R&D Center, Hanwha Systems  
• Received: 2018. 09. 06, Revised: 2018. 10. 15, Accepted: 2018. 10. 17.

일한 이름을 가지고, 수신 또는 송신하고자 하는 사용자 데이터의 구조이다. 시스템 요구사항 또는 네트워크 상태 등을 고려하여 사용자 데이터 크기와 동일한 크기로 topic을 정의하여 한번에 하나의 사용자 데이터를 송수신할 수 있고, 작은 크기의 topic을 정의하여 여러 번 나누어 하나의 사용자 데이터를 송수신할 수도 있다. DDS QoS Policy는 네트워크상의 Data-Flow에 대한 설정 값으로 실시간 동작제어를 위한 파라미터 집합이다. 표준 DDS QoS Policy는 약 20여 개의 종류가 있고, 하나의 DDS QoS Policy는 수 개에서 수십 개의 속성 값을 가진다. 예를 들어 DURABILITY\_SERVICE QoS Policy의 경우에는 history\_kind, history\_depth, max\_samples 등 6개의 속성 값이 정의되어 있다. 그리고, 표준 DDS QoS Policy 이외에 DDS 미들웨어 제품 별로 vendor specific 하게 확장된 DDS QoS Policy를 제공하는데 본 연구에 사용한 RTI 社의 RTI Connnext의 경우에는 확장된 DDS QoS Policy가 30개 이상 더 있다[10]. DDS QoS Policy 튜닝이란 시스템 개발자들이 동작 제어, 통신 품질, 송신 타이밍 등의 시스템 요구사항에 맞게 DDS QoS Policy의 속성 값들을 설정하는 것이다[1,6-11]. DDS QoS Policy 튜닝 과정에서 시스템 요구사항을 만족시키기 위하여 필요한 DDS QoS Policy의 속성 값들을 설정할 때에는 기존에 설정된 같은 또는 다른 DDS QoS Policy의 속성 값의 특성과 서로 상충되지 않도록 설정해야하기 때문에 많은 시간과 노하우가 필요하다. QoS incompatible 에러가 발생한다면 시스템의 성능적인 측면의 문제 수준이 아니라 데이터 송수신 자체가 불가능할 수도 있고, QoS incompatible 에러의 원인을 찾는 것이 무척 힘들기 때문이다. QoS incompatible 에러가 발생하면 튜닝한 DDS QoS Policy 속성들을 모두 재검토하거나 별도의 디버깅 도구를 구입하여 QoS incompatible 에러의 원인이 되는 범위를 축소한 후 디버깅해야 한다. 이렇게 DDS QoS Policy 튜닝에 대한 어려움이 있기 때문에 DDS 미들웨어 제작사들은 DDS 미들웨어 비용과는 별도로 시스템 요구사항에 맞게 DDS QoS Policy 튜닝을 위한 추가의 컨설팅 비용을 청구한다. 컨설팅 비용은 각 DDS 미들웨어 제작사 및 시스템 규모 별로 상이하지만 대부분 DDS 미들웨어 구입 비용만큼 비싸기 때문에 많은 사업에서는 컨설팅을 요청하지 않고, 기본적으로 제공되는 샘플의 DDS QoS Policy를 조금씩 변경하여 사용하는 것이 보통이다.

본 논문에서는 기본적으로 제공되는 샘플의 DDS QoS Policy로는 시스템 요구사항을 만족하기 어려운 사용자의 위치 정보, 표적 정보와 같이 크기가 작고, 전송 빈도가 높은 사용자 데이터와 오디오, 비디오, 데이터베이스와 같이 크기가 큰 사용자 데이터를 송수신할 때 적용할 수 있는 다양한 DDS QoS Policy 및 이를 이용한 솔루션을 제시한다. 그리고 각각의 방법에 대하여 자체 개발한 시험 프로그램으로 성능을 측정하고, 결과 분석을 통하여 특정 조건에서 효율적인 송수신 방법을 결정하는 데 도움을 줄 수 있도록 DDS 미들웨어 사용 가이드를 제공할 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 DDS 미들웨어를 이용하여 사용자 데이터를 송수신하기 위한 몇 가지 기본적인 DDS QoS Policy 및 이를 이용한 방법에 대해서 설명하고, 3장에서는 사용자 데이터 특성에 맞게 적용할 수 있는 향상된 솔루션을 제시한다. 4장에서는 앞에서 제시된 여러 방법들의 성능을 자체 개발한 시험 프로그램을 이용하여 시험하고, 그 결과에 대해서 분석한다. 마지막으로 5장에서 결론을 맺는다.

## II. Related Works

DDS에서 사용자 데이터는 topic을 이용하여 발간(publish)/구독(subscribe)하기 때문에 사용자 데이터의 포맷이 텍스트, 오디오 또는 비디오 등의 타입 보다는, 사용자 데이터 및 적용한 topic의 크기, 송수신 빈도 등의 요소를 고려하여 DDS QoS Policy 튜닝을 수행하여야 한다.

앞에서 언급한 바와 같이 기본적으로 제공되는 샘플의 DDS QoS Policy로는 크기가 작으면서 송수신 빈도가 높은 특성, 데이터 크기가 큰 특성, 그리고 데이터 크기가 크지만 우선순위가 낮아서 시스템의 성능에 영향을 최소화하여 송수신을 해야 하는 특성을 가지는 사용자 데이터를 처리하기 힘들거나 불가능하다. 이번 장에서는 세 가지 특성의 사용자 데이터를 송수신하기 위하여 사용할 수 있는 표준 DDS QoS Policy와 기본적인 송수신 방법을 설명한다.

### 1. Methods for high-frequency data

데이터 크기가 작으면서 송수신 빈도가 높은 고빈도 데이터는 짧은 시간에 많은 수의 데이터를 송수신하기 때문에 다음의 DDS QoS Policy 속성들이 기본적으로 고려되어야 한다.

- RESOURCE\_LIMIT - 고빈도 데이터를 송수신하므로 버퍼 크기는 한 번에 전송하는 데이터 개수 이상 설정
- HEARTBEAT - 송신측의 버퍼를 빨리 비워야 하므로 주기를 빠르게 설정
- MAX\_HEARTBEAT\_RETRIES - 그룹에서 탈퇴한 수신단을 빠르게 inactive로 처리하기 위하여 데이터 전송 주기와 HEARTBEAT 주기 및 RESOURCE\_LIMIT 크기를 고려하여 설정
- RELIABILITY - 위치 정보와 같이 주기적으로 송신되고, 최신 데이터 값이 유효한 경우, BEST\_EFFORT로 설정 (또는 RELIABILITY가 RELIABLE인 경우, HISTORY 속성을 KEEP\_LAST로 설정)

DDS를 이용하여 송수신할 때, 패킷 구조는 아래 Fig. 2와 같이 세 가지 타입을 기본적으로 사용할 수 있다[10].

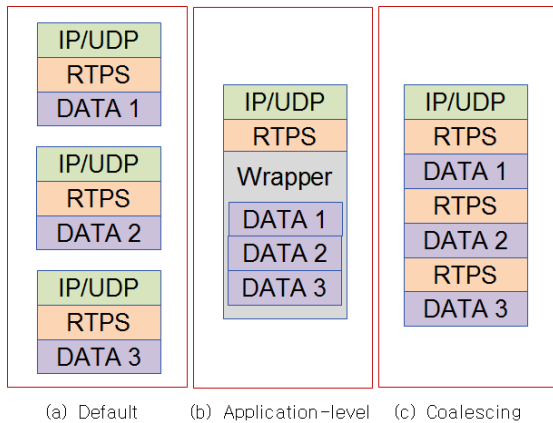


Fig. 2. DDS packet types

Fig. 2의 (a) 구조는 데이터 전송을 위한 DDS의 기본적인 패킷 구조이다. 하나의 IP 패킷에 DDS RTPS 헤더와 사용자 데이터가 각각 하나씩 포함된 구조이다. 이 경우, 1,000개의 사용자 데이터를 전송하게 되면 1,000개의 IP 헤더, DDS RTPS 헤더와 사용자 데이터인 RTPS의 submessage가 송수신된다. 문제 발생 시, 재전송을 위한 단위도 IP 헤더, DDS RTPS 헤더와 RTPS의 submessage가 포함된 전체 패킷이 된다. Fig. 2의 (b) 구조는 사용자가 application-level에서 여러 개의 사용자 데이터를 묶어서 하나의 새로운 사용자 데이터를 정의하여 사용하는 방법이다. 이 경우에는 기본적인 패킷 구조를 이용하는 Fig. 2의 (a) 구조보다 송수신 및 분석해야 하는 DDS RTPS와 IP 헤더 정보가 적어서 해당 정보를 처리하는 부하량이 줄어들기 때문에 기본적인 패킷 구조를 이용하는 경우보다 효율적인 솔루션이라 할 수 있다. 하지만 처음 정의한 사용자 데이터가 아니라 처음 정의한 사용자 데이터들이 wrapping된 새로운 사용자 데이터이기 때문에 DDS 개발 회사 등에서 제공하는 디버깅 도구 사용이 힘들고, DDS에서 제공하는 key[10]를 사용할 수 없는 단점들이 있다. 그리고, 새로운 사용자 데이터로 취급하고, 취급된 것을 다시 나누는 추가 로직이 필요하므로 응용 구현이 보다 복잡해지는 문제점이 추가된다. 마지막으로 Fig. 2의 (c) 구조는 하나의 IP 패킷에 여러 개의 DDS RTPS 헤더와 사용자 데이터 정보의 쌍이 모이는 coalescing 구조이다. 전송 측의 관련 DDS QoS Policy 값만 설정하면 coalescing 구조로 송수신이 가능하다. 이 방법은 IP 헤더의 수도 줄여주고, 사용자가 별도로 wrapping 관련 처리부를 추가할 필요 없고, key 및 디버깅 도구 사용이 가능하여 앞의 두 방법에 비하여 여러 가지 측면에서 나은 방법이다. Coalescing 구조는 주로 NACK 응답에 대하여 한꺼번에 많은 수의 데이터 재전송이 필요한 경우에 사용된다.

## 2. A method for large-capacity data

기본적인 DDS QoS Policy를 적용한다면 한 번에 전송할 수 있는 데이터의 크기는 8KB 미만이기 때문에 8KB 이상의 사용자 데이터를 송수신 할 때에는 전송 관련 에러가 발생한다.

8KB 이상의 사용자 데이터 전송을 위해서는 관련되는 DDS QoS Policy들의 튜닝이 필요하다[10]. 특히, 대용량 데이터는 오디오, 비디오 및 데이터베이스 등과 같이 신뢰적 전송이 필요한 데이터가 대부분이므로 기본적으로 아래와 같은 DDS QoS Policy 속성들이 고려되어야 한다.

- RESOURCE\_LIMIT - Topic 크기가 큰 경우, 버퍼 크기를 크게 설정하면 응용 프로그램이 사용하는 메모리 또한 비례하여 증가. 노드의 가용 메모리를 고려하여 설정 (메모리가 작은 임베디드 시스템은 특히 주의해서 설정)
- RELIABILITY - 사용자 데이터를 손실 없이 송수신해야 하므로 KEEP\_ALL로 설정 (RESOURCE\_LIMIT로 설정한 버퍼에 가득 찬 경우, 전송 측에서는 버퍼가 비워질 때까지 설정된 시간만큼 대기하게 되므로 일시적 또는 영구적으로 통신이 단절된 것처럼 동작)
- MAX\_MESSAGE\_SIZE, RECV\_SOCKET\_BUFFER, MAX\_FRAGMENTS\_PER\_SAMPLE - Topic 및 사용자 데이터의 크기, 전송 프로토콜의 종류를 고려하여 설정

## 3. A method using communication group with specific node information

같은 도메인을 사용하는 응용프로그램들 간에도 통신의 필요성에 따라서 특정 노드들을 선별하여 디스커버리를 수행한 후 데이터를 송수신 하는 방법이 있다[11]. 통신할 노드를 찾는 디스커버리 과정에서 DomainParticipant 관련 DDS QoS Policy 속성 중 예약되어 사용하지 않는 USER\_DATA 필드에 통신그룹 값을 설정함으로써 통신그룹 별로 구분하여 디스커버리를 처리하여 데이터 통신이 필요 없는 노드와는 데이터 송수신을 하지 않는다. 이 방법은 고빈도 및 대용량 데이터 특성 등 모든 종류의 사용자 데이터 송수신에 적용할 수 있으며 디스커버리와 데이터 송수신에 소비되는 시간, 컴퓨터 자원 등의 사용률을 감소시킬 뿐만 아니라 네트워크 사용률까지 줄일 수 있다.

## III. Proposed Methods for Data Transmission and Reception of Various Characteristics

앞에서 분류한 세 종류의 데이터 특성을 조금 더 자세히 정의하면 다음과 같다. 첫 번째는 고빈도 데이터로 1KB 이하의 크기를 가진 사용자 데이터를 초당 수천 번의 빈도로 송수신하는 경우이다. 두 번째는 대용량 데이터로 초당 수 MB의 크기를 가진 사용자 데이터를 한 번씩 송수신하는 경우이다. 세 번째는 우선순위가 낮은 대용량 데이터(백그라운드 대용량 데이터)로 수십 MB의 크기를 가진 사용자 데이터를 다른 우선순위가 높거나 보통인 데이터 교환에 영향을 끼치지 않고 송수신하는 경우인데, 특별히 요구되는 시간 제약은 없다. 본 장에서는 세 가지 데이터 특성에 적용할

수 있는 DDS QoS Policy 튜닝 또는 송수신 방법을 제시한다.

### 1. A Method for high-frequency data using extended DDS QoS policy

RTI 社の DDS인 RTI Connex의 경우에도 RTI Connex에 서만 지원하는 확장된 DDS QoS Policy를 이용하여 추가 기능을 다양하게 제공한다. 그 중 batch QoS Policy는 고빈도 데이터 송수신 방법에서 설명한 coalescing 구조를 조금 더 개선한 방법으로 다양한 설정으로 batching이 가능한 DDS QoS Policy이다[1, 10]. Batching 구조는 아래 Fig. 3과 같다.

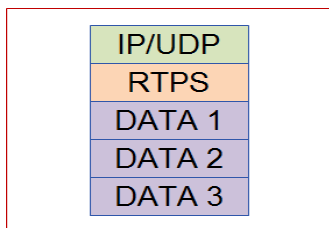


Fig. 3. Batching DDS Packet Type

Batching을 이용하면 하나의 IP 패킷에 하나의 DDS RTPS 헤더와 여러 개의 사용자 데이터가 모이는 구조로 전송된다. 기본적으로 DDS는 사용자가 write() API를 호출함과 동시에 즉시 사용자 데이터를 전송한다. 하지만 batch QoS Policy를 설정하면 DDS 스택 내부에 batch라는 저장소를 생성하고, 사용자 데이터를 batch에 모은 다음 전송 조건이 만족될 때 모았던 사용자 데이터를 한꺼번에 submessage로 전송한다. 전송 조건은 batch에 누적된 총 데이터 크기, 사용자 데이터 개수, flush 주기의 세 가지가 있다. 사용자 데이터 크기, 네트워크 성능과 CPU, 메모리 등의 컴퓨터 자원의 상태에 맞게 batch의 크기와 하나의 batch에 모으는 데이터 개수, flush 주기 값을 조절할 수 있기 때문에 앞에서 설명한 다른 방법들보다 유연하게 사용할 수 있다.

### 2. A method for large-capacity data using segmentation and reassembly

대용량 데이터 송수신을 위해서는 기본적으로 대용량 데이터 전송 관련 DDS QoS Policy를 튜닝하는 방법도 있지만, 네트워크 성능을 고려하여 사용자 데이터를 application-level에서 작게 나눈 후, 고빈도 데이터 송수신 방법을 이용할 수도 있다. 즉, 5MB의 사용자 데이터를 전송하기 위하여 5MB 크기의 데이터를 사용자 레벨에서 한 번에 보낼 수도 있지만 사용자 레벨에서 미리 작은 크기의 데이터로 분할하여 여러 번에 나누어 송수신하고, 수신 단에서는 다시 취합한 후 5MB 크기로 만들어서 전달하는 방법을 사용하면 된다[8, 10]. 이 경우, 송수신하는 topic 크기에 따라서 다양하게 성능을 조절할 수 있기 때문에 노드와 네트워크 성능 및 상태에 따라서 적절한 topic 크기를 결정하는 것이 아주 중요하다. 다음의 Fig. 4는 대용량 데이터를 작

은 크기의 topic으로 나누어 송수신하는 경우와 하나의 큰 topic을 이용하여 송수신하는 경우의 차이점을 보여준다.

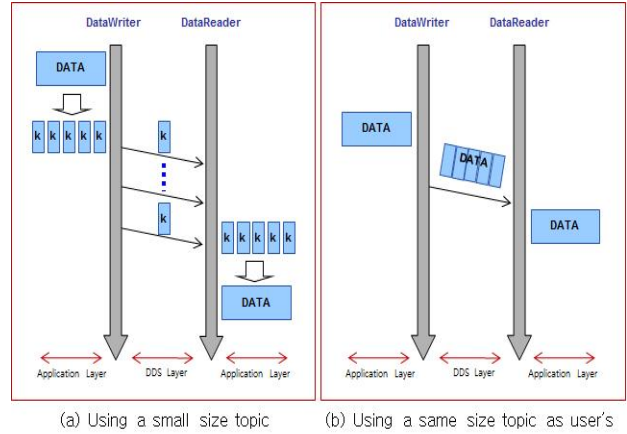


Fig. 4. Comparison for message sending and receiving methods

Fig. 4의 (a)와 같이 작은 크기의 topic을 이용하여 송수신하는 경우에는 application-level에서 하나의 대용량 데이터를 여러 개의 작은 사용자 데이터로 나누어 송신하고, 수신된 작은 사용자 데이터를 다시 하나의 대용량 데이터로 취합하는 추가 로직이 필요하다. 그리고 Fig. 4의 (b)의 방법에 비하여 작은 사용자 데이터로 나누고, 다시 취합하는데 시간 지연, CPU 및 메모리 자원 사용 등의 추가 비용이 발생하게 된다. 하지만 송수신되는 데이터 크기가 작아서 손실된 DDS 패킷에 대하여 빠르게 복구가 가능하고, 전송 시 delay를 추가하여 중간에 조건을 주어 전송 타이밍을 조절할 수 있다는 장점들이 있다. Fig. 4의 (b)의 경우는 application-level에서의 추가 작업에 따른 비용이 발생하지 않지만 크기가 큰 사용자 데이터를 한 번에 송수신하기 때문에 손실된 DDS 패킷 복구에 발생하는 비용이 Fig. 4의 (a) 보다는 크고, 대용량 데이터를 송신하고, 수신하는 긴 시간 동안 다른 DDS 패킷 처리가 불가능한 단점이 있다.

### 3. A method for large-capacity data with low priority in consideration of network conditions

기본적으로 DDS는 송수신되는 순서대로 데이터를 처리하기 때문에 실시간성의 요구사항이 필요 없는 사용자 데이터를 송수신하는 경우에도 우선순위가 높거나 실시간성이 보장되어야 하는 사용자 데이터의 교환에 영향을 끼치게 된다. 우선순위가 높거나 실시간성이 보장되어야 하는 사용자 데이터 처리를 위하여 사용자가 적절한 타이밍을 계산하여 백그라운드로 우선순위가 낮은 대용량 데이터를 컨트롤하여 천천히 전송하기는 힘들다. 이 경우 사용할 수 있는 것이 Asynchronous DataWriter와 Flow Controller QoS Policy이다. 이 두 종류 또한 RTI Connex에서만 제공하는 확장된 DDS QoS Policy이다.

아래의 Fig. 5는 Synchronous DataWriter, Asynchronous DataWriter 및 Flow Controller QoS Policy의 동작 과정을 보여준다.



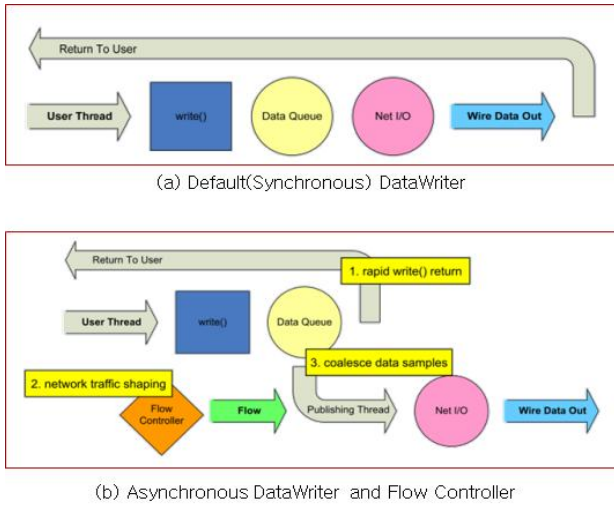


Fig. 5. Comparison with synchronous and asynchronous method

DDS의 DataWriter는 기본적으로 Fig. 5의 (a) 방법과 같이 synchronous로 동작한다. 사용자 레벨에서 사용자 데이터 전송을 위하여 API(write())를 호출하면 DDS 내부에서 네트워크로 전송이 완료된 후 그 결과를 사용자 레벨로 반환하고, 계속해서 다음 작업을 진행하는 구조이다. Fig. 5의 (b) 방법은 DataWriter가 asynchronous로 동작하는 경우를 설명하는 그림으로, 사용자 레벨에서 사용자 데이터 전송을 위하여 API(write())를 호출하면 사용자 데이터를 DDS 내부 큐에 넣고 즉시 사용자 레벨로 반환되는 것을 보여준다. 저장되는 사용자 데이터는 DDS 내부에서 별도의 스레드가 처리한다. Flow Controller는 asynchronous 동작에서 별도의 스레드를 이용하여 네트워크 트래픽을 조절할 수 있는 DDS QoS Policy로 다른 속성 값보다 tokens\_added\_per\_period 값[10]이 성능에 많은 영향 미친다. 이 값이 크면 한 번에 전송하는 사용자 데이터 수가 많아지고, 작으면 한 번에 전송하는 데이터 수가 적어지게 되기 때문이다. tokens\_added\_per\_period 값을 포함하여 네트워크 상황에 맞게 관련 속성들의 설정 값들을 조절하여, 실시간 및 우선순위가 높은 데이터의 전송에 영향을 받지 않도록 데이터를 송수신할 수 있다.

## IV. Performance Test and Evaluation

### 1. Test environment

앞에서 설명한 사용자 데이터의 특성에 따라 고빈도 데이터, 대용량 데이터, 우선순위가 낮은 백그라운드 대용량 데이터 송수신에 대한 성능을 시험하였고, 시험 환경은 아래 Table 1과 같다.

Table 1. Test Environment

Item	Quantity	Specification (Remark)
PC	2	<ul style="list-style-type: none"> <li>■ CPU: Intel i7 CPU@3.40 GHz</li> <li>■ Memory: 8GB</li> </ul>
Switch	1	<ul style="list-style-type: none"> <li>■ CISCO WS-C3560X-24T-S</li> </ul>
Test Program	2	<ul style="list-style-type: none"> <li>■ Self-Development</li> <li>1) Test Program for High-Frequency Data</li> <li>2) Test Program for Large-Capacity Data</li> <li>3) Test Program for Large-Capacity Data with low priority</li> </ul>
DDS	2	<ul style="list-style-type: none"> <li>■ RTI Connex 5.0</li> </ul>
OS	2	<ul style="list-style-type: none"> <li>■ Microsoft Windows 7 Enterprise K(64bit)</li> </ul>

DDS는 가장 범용성이 있는 RTI社의 RTI Connex이며, 사용한 버전은 5.0이다. 1Gbps의 네트워크 환경에서 Windows 7 64bit 운영체제가 설치된 PC 2대를 이용하여 시험하였다. 성능 측정을 위한 시험 프로그램은 사용자 데이터 특성에 따라 각각 사용할 수 있도록 자체 개발한 고빈도, 대용량, 백그라운드 대용량 데이터 송수신 프로그램이다. 이 프로그램은 고빈도, 대용량, 백그라운드 대용량 데이터 각각의 특성을 식별하고, 각각의 경우에 대하여 시험환경에서 최적의 DDS QoS Policy 튜닝을 적용하였으며, 원하는 크기의 사용자 데이터 및 topic을 원하는 횟수만큼 송수신할 수 있다. 입력된 횟수만큼 반복하여 각각의 지연 시간을 측정하고, 측정된 결과 값의 최소, 최대, 평균, 표준편차 등을 자동으로 계산하여 출력하도록 구현하였다.

## 2. Tests of high-frequency data

### 2.1 Test methods

고빈도 데이터 송수신 시험에서 시험 프로그램은 아래 Fig. 6과 같이 동작한다.

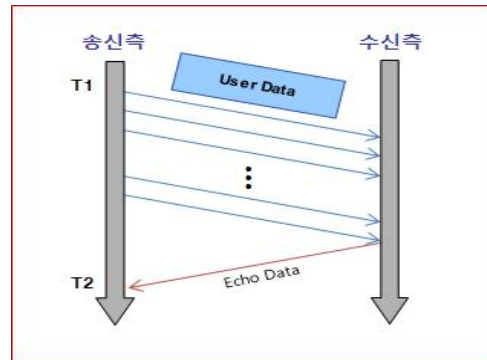


Fig. 6. A program for high frequency data test

송신측에서 원하는 크기의 사용자 데이터를 n회 전송하고, 시험 시작 시각(T1)을 저장한다. 수신측에서 송신측에서 전송하는 데이터를 n회 받은 후, 송신측에 데이터를 모두 받았다는 의미로 “Echo Data”를 전송한다. 송신측에서는 “Echo Data”를 받는 시간(T2)을 저장한 후, 시험 시작 시각(T1) 값과 시험 종료 시각(T2) 값을 이용하여 coalescing 방법과 batching 방법 각각에 대하여 DDS QoS Policy 값을 변경하며 최적의 성능이 나올 때까지 시험한다. 최적의 성능이 나올 때의 DDS QoS Policy를 이용하여 토픽 크기 및 송수신 횟수별로 각각

20회씩 반복하여 지연 시간을 측정한다.

### 2.2 Test results

아래의 Table 2, Fig. 7 및 Fig. 8은 coalescing과 batching 방법 각각에 대하여 topic의 크기를 256Bytes, 512Bytes, 1KB로 변경하고, 송수신 횟수를 3,000회, 10,000회로 변경하며 시험한 결과이다.

Table 2. A result of high-frequency data transmission and reception test

Number of samples	Applying method	Topic size	Average latency time (ms)	Minimum latency time (ms)	Maximum latency time (ms)	Standard deviation
3,000	Coalescing	256B	106.3	94	120	6.99
		512B	159.75	143	188	13.65
		1KB	274.3	254	340	17.71
	Batching	256B	61.85	58	66	2.54
		512B	98.8	94	105	3.54
		1KB	174.8	164	185	7.02
10,000	Coalescing	256B	257.25	224	288	19.52
		512B	437.05	396	481	27.45
		1KB	787.38	721	872	38.91
	Batching	256B	146.8	140	156	4.97
		512B	226.6	211	245	11.83
		1KB	468.3	440	490	17.47

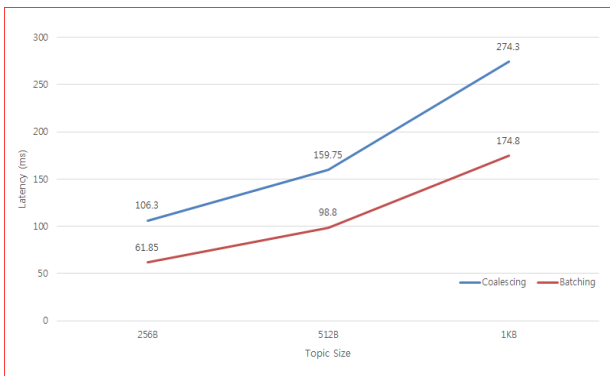


Fig. 7. A result of 3,000 times transmission and reception test

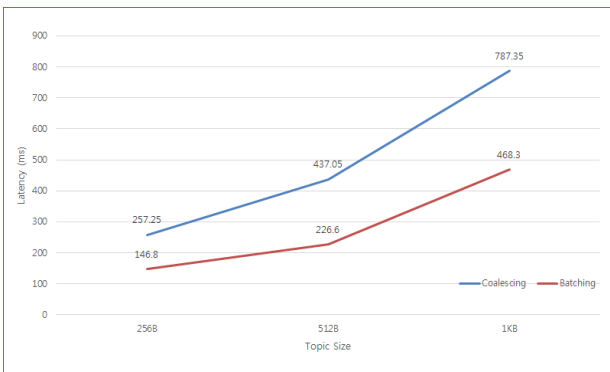


Fig. 8. A result of 10,000 times transmission and reception test

### 2.3 Analyze test results

Topic의 크기와 송수신 횟수에 상관없이 시험한 모든 경우에 대해서는 coalescing 방법보다 batching 방법이 표준 편차가 낮아서 가장 안정적이고, 지연 시간이 짧다.

Batching 방법은 하나의 DDS RTPS 헤더를 이용하여 여러 개의 사용자 데이터를 한꺼번에 처리 가능하다. 그렇기 때문에 수신측에서는 meta-information 처리에 대한 overhead가 줄어들고, CPU 등의 자원 사용량도 감소한다. 또한, 신뢰성 있는 데이터 전송 시에, 여러 개의 사용자 데이터가 포함된 batch 한 개에 대하여 ACK 또는 NACK 처리를 하므로 사용자 데이터 한 개에 대해서 ACK 또는 NACK 처리를 하는 다른 방법보다 throughput 측면의 성능이 훨씬 향상됨을 고빈도 성능 시험 결과로 확인할 수 있다.

## 3. Tests of large-capacity data

### 3.1 Test methods

본 시험에서의 대용량 데이터는 한 번에 전송하려는 사용자 데이터의 크기를 5MB인 경우로 정한다. 대용량 데이터 송수신 시험 프로그램은 고빈도 데이터 송수신 시험 프로그램과 동일하게 동작하는데 “III-2. A method for large-capacity data using segmentation and reassembly”에서 설명한 분할/재조립 방법을 이용하였다. 분할/재조립 방법을 이용하면 5MB 사용자 데이터 송수신 시험 시 5MB 크기의 topic을 1회 송수신할 때 측정된 지연 시간과 1KB 크기의 topic을 5,120회 송수신할 때 측정된 지연 시간을 비교 분석할 수 있다. 대용량 데이터 송수신 시험은 512Bytes, 1KB, 256KB, 512KB, 1MB, 5MB 크기의 topic을 이용하여 각각 20회씩 반복하여 5MB 크기의 사용자 데이터를 송수신할 때 소요되는 지연 시간을 측정한다.

### 3.2 Test results

아래의 Table 3, Fig. 9는 512Bytes, 1KB, 256KB, 512KB, 1MB, 5MB 크기의 topic을 이용하여 5MB의 사용자 데이터를 송수신할 때, 각각의 topic 크기에 따른 지연 시간을 측정한 시험 결과이다.

Table 3. A result of large-capacity data transmission and reception test

Topic size	Number of samples	Average latency time (ms)	Minimum latency time (ms)	Maximum latency time (ms)	Standard deviation
5MB	1	57.93	55	62	2.11
1MB	5	51.93	51	56	1.29
512KB	10	53.87	52	58	1.5
256KB	20	58.8	57	66	2.23
1KB	5,120	155.73	150	173	5.26
512B	10,240	217.2	199	245	12.63

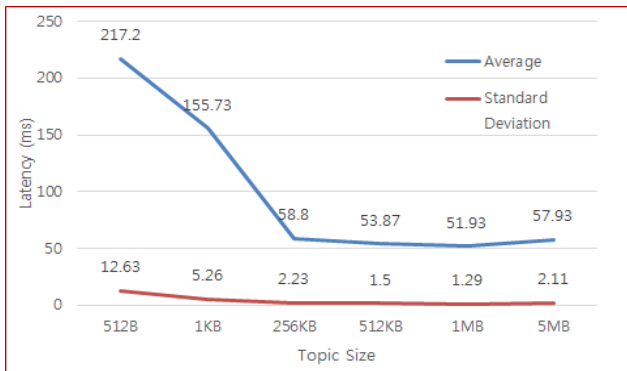


Fig. 9. A result of large-capacity data transmission and reception test

### 3.3 Analyze test results

Topic의 크기가 5MB인 경우의 평균 지연시간은 57.93ms, topic의 크기가 1MB인 경우는 51.93ms, topic의 크기가 512KB인 경우는 53.87ms, topic의 크기가 256KB인 경우는 58.8ms, topic의 크기가 1KB인 경우는 155.73ms, topic의 크기가 512B인 경우는 217.2ms이다. 5MB의 사용자 데이터를 전송하는 경우, 1MB 크기의 topic을 이용하여 5번 송수신하는 방법이 가장 안정적이고, 송수신이 완료되는 시간도 가장 빠름을 알 수 있다. 상대적으로 topic 크기가 작고 송수신 횟수가 많은 경우가 topic 크기가 크고, 송수신 횟수가 적은 경우보다 속도 측면에서의 성능이 낫다. 송수신하는 패킷 개수가 늘어나기 때문에 처리해야 하는 DDS 헤더 정보와 분할 및 취합하는 프로세싱 시간이 더 많아지기 때문이다. 같은 크기의 대용량 데이터를 송수신할 때, 전혀 분할하지 않거나 분할하는 topic의 샘플 개수가 많아지면 시간적인 측면에서는 좋지 않음을 대용량 데이터 시험 결과로 확인할 수 있다.

## 4. Tests of large-capacity data with low priority

### 4.1 Test methods

백그라운드 대용량 데이터 송수신 시험 프로그램은 100MB 크기의 사용자 데이터를 작은 크기의 topic으로 나누어 송수신이 완료될 때의 지연 시간을 측정한다. 최적의 성능이 나올 때의 DDS QoS Policy를 이용하여 지연 시간과 네트워크 사용률을 20회씩 반복하여 측정하여 백그라운드 대용량 데이터 송수신 시험을 진행한다.

### 4.2 Test results

아래의 Table 4는 100MB의 사용자 데이터를 32KB 크기의 topic을 이용하여 송수신 할 때, token\_added\_per\_period 속성 값을 UNLIMITED, 30, 20, 10, 5, 1개로 각각 변경하며 지연 시간과 네트워크 사용률을 측정한 시험 결과이다.

Table 4. A result of background data transmission and reception test

Values of token_added_per_period property	Average latency time (ms)	Network usage (%)
UNLIMITED	5918.4	12
30	5916.2	12
20	6004.5	10
10	12107.1	6
5	23564.0	3.5
1	116765.3	0.4

### 4.3 Analyze test results

Asynchronous DataWriter와 Flow Controller를 이용하면 사용자 데이터를 전송할 때, 이용 가능한 토큰의 수에 따라서 전송할 수 있는 데이터의 양이 달라진다. 그래서 백그라운드 대용량 데이터 송수신 시험에서 주기적으로 추가되는 토큰 수인 token\_added\_per\_period 값을 조절하며 다른 시험과는 다르게 지연 시간 측정과 함께 네트워크 사용률을 측정하였다.

주기적으로 추가되는 토큰의 수를 UNLIMITED로 설정하면 100MB의 사용자 데이터를 송수신하는데 걸리는 시간은 평균 5,918.4ms이고, 이때의 네트워크 사용률은 12%이다. 그리고 주기적으로 추가되는 토큰의 수를 1로 설정하면 100MB의 사용자 데이터를 송수신하는데 걸리는 시간은 평균 116,765.3ms이고, 이때의 네트워크 사용률은 0.4%이다. 주기적으로 추가되는 토큰의 수를 UNLIMITED에서 1로 줄이면 네트워크 사용률을 12%에서 0.4%까지 줄일 수 있지만 데이터의 송수신 완료까지 소요되는 시간은 약 20배까지 차이가 난다.

백그라운드 대용량 데이터 송수신 시험 결과로 확인할 수 있듯이 Asynchronous DataWriter와 Flow Controller를 이용하여 주기마다 추가되는 토큰 수를 줄인다면 송수신 완료까지 소요되는 시간은 증가되지만, 한 번에 송수신되어 처리해야 하는 데이터 사이즈 및 데이터 개수는 줄어든다. 백그라운드 대용량 데이터를 송수신하는 노드에서는 한 번에 처리해야 하는 데이터가 줄어들고, 해당 노드의 네트워크 사용률까지 줄어들기 때문에 중요한 다른 실시간 데이터 송수신 및 처리에 미치는 영향을 줄일 수 있다.

## 5. Result and Discussion

DDS 미들웨어를 이용할 때 기본 DDS QoS Policy를 이용하여 처리하기 어려운 대표적인 세 가지 사용자 데이터 특성인 고빈도, 대용량, 백그라운드 대용량 데이터에 대하여 각 특성별로 시험을 하고, 그 결과를 확인하였다.

고빈도 데이터는 작은 크기의 사용자 데이터를 초당 수천 번의 빈도로 송수신한다. 10,000번의 송수신 시험 결과에서 알 수 있듯이 메시지 헤더와 같은 공통부분의 송수신 횟수를 줄이기 위하여 coalescing 방법을 이용하고, 버퍼 크기를 충분히 설정한 후 HEARTBEAT 주기를 빠르게 설정한다면 지연 시간이 평균 약 787ms, 최대 872ms로 1초의 시간적인 요구사항을 만족할 수 있다. RTI Connex를 이용하는 경우에는 확장 DDS

QoS Policy로 제공하는 batching 방법을 이용하면 보다 안정적으로 고빈도 데이터 송수신이 가능하다.

대용량 데이터는 수 MB의 사용자 데이터를 매 초 송수신하는 특성을 가지고 있다. 대용량 데이터 송수신 시험 결과에서 알 수 있듯이 수 MB의 사용자 데이터를 DDS로 한 번에 전송하거나 아주 작은 크기의 topic으로 나누어 송수신하는 방법 보다는 적절한 크기의 topic으로 분할/재조립하는 방법을 적용해야 한다.

백그라운드 대용량 데이터는 Asynchronous DataWriter와 Flow Controller를 이용하고, 요구사항과 노드 및 네트워크 상태 등을 고려하여 다른 실시간 사용자 데이터 송수신에 영향을 최소화하도록 다른 속성 값들 보다 token\_added\_per\_period 값의 설정이 중요하다. 설정 값이 작아질수록 송수신에 소요되는 전체 시간은 증가되지만, 백그라운드 대용량 데이터 송수신에 사용되는 네트워크 사용률이 줄어들어 다른 사용자 데이터 송수신에 끼치는 영향은 줄어든다.

## V. Conclusions

하나의 시스템에서 사용자 데이터의 종류와 크기, 송수신되는 주기들은 다양하다. 그리고 같은 특성의 사용자 데이터라도 네트워크 상황과 송수신 노드의 개수에 따라서 성능이 다르게 나올 수 있기 때문에 적절한 데이터 송수신 방법을 찾거나 DDS QoS Policy 튜닝을 하는 과정이 힘들다. 무엇보다 사용자 데이터 특성 별로 어떠한 종류의 DDS QoS Policy를 적용할 수 있으며, 속성 값들은 어떻게 설정할 것인지를 결정해야 하는데 이는 많은 경험이 필요한 작업이다.

본 논문은 DDS 미들웨어를 사용하여 최고 성능을 내기 위하여 단순히 DDS QoS Policy를 튜닝하고, 송수신 방법을 찾아서 그 성능을 증명하는 것이 아니다. 앞에서 시험한 세 가지 특성에 대한 시험 결과 또한, 모든 시스템에서 항상 최적의 성능을 보장하지는 않는다. 특정 환경에서 하나의 사용자 데이터 타입에 대한 throughput이 월등히 좋다고 해서 모든 시스템에 최적화 된 DDS QoS Policy 튜닝은 아니기 때문이다.

본 논문의 목적은 시스템을 구성하는 노드 및 네트워크의 성능과 상태를 고려하여 시스템 요구사항을 만족시킬 수 있도록 송수신 방법, topic 크기, 적절한 DDS QoS Policy를 선택하고 이를 튜닝할 수 있도록 가이드를 제시하는 것에 목적이 있다. 이를 위하여 DDS를 이용하여 다양한 특성을 가진 사용자 데이터를 송수신하기 위한 기본적인 송수신 및 DDS QoS Policy 종류 및 튜닝 방법과 효율적인 추가 방법들을 제시하고, 각각의 방법들을 시험하여 성능을 확인하였다. DDS를 이용하는 시스템을 개발할 때, 본 논문에서 설명한 여러 가지 방법을 적절하게 적용한다면 보다 효율적인 송수신 솔루션을 찾거나 DDS QoS Policy를 튜닝한 프로파일을 만드는 데 기여할 수 있을 것으로 생각된다.

## REFERENCES

- [1] Y. Jeong, Y. Choi, S. Ahn, and C. Kim, "The Study of DDS Batch QoS Policy," IEEK Fall Conference Proceedings, pp. 433-435, November, 2011.
- [2] D. C. Schmidt, A. Corsaro, and H. Hag, "Addressing the Challenges of Tactical Information Management in Net-Centric Systems with DDS," The Journal of Defense Software Engineering, pp.24-29, March, 2008.
- [3] Object Management Group, <https://www.omg.org>
- [4] Real-Time Innovations, <https://www.rti.com>
- [5] Hanwha Systems, <http://hanwhasystems.com>
- [6] OMG, Data Distribution Service for Real-time System Version 1.2, January, 2007.
- [7] OMG, The Real-time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification Version 2.1, June, 2008.
- [8] Y. Jeong, "A Study on Large-Capacity Data Transmission and Reception Method using DDS," KIMST Annual Conference Proceedings, pp. 583-584, November, 2017.
- [9] M. Yu, and M. Kang, "Performance Analysis of DDS," KIMST Annual Conference Proceedings, pp. 516-517, June, 2014.
- [10] Real-Time Innovations, Inc, "RTI Connex Core Libraries and Utilities User's Manual Version 5.0," August, 2012.
- [11] Y. Jeong, and C. Kim, "A Study on DDS Discovery Method for Improving the Performance," KIMST Annual Conference Proceedings, pp. 1043-1044, June, 2015.

### Author



Yeongwook Jeong received the B.S., and M.S. degrees in Computer Science from Kyungpook National University, Korea, in 2004 and 2006, respectively. He is currently working as a senior engineer at Naval R&D Center of Hanwha Systems. He is interested in real-time distributed computing, wireless sensor networks, and national defense.