

WAN 환경에서 전송 효율 개선을 위한 DDS RTPS 메시지 변환

이영석, 최훈*, 조성일**

군산대학교, *충남대학교, **(주)구름네트웍스

leeys@kunsan.ac.kr, *hc@cnu.ac.kr, **sungil@gurum.cc

Transform DDS RTPS messages to improve transmission efficiency in WAN environments

Lee Young Seok, Choi Hoon*, Cho Sung Il**

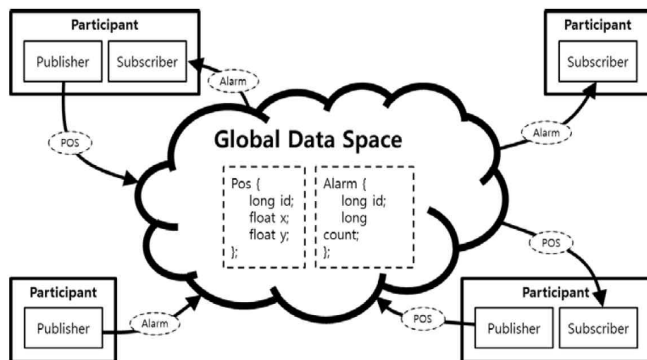
Kunsan National Univ. *Chungnam National Univ., **Gurum Networks

요약

LAN(Local Area Network) 구간을 이어주는 WAN(Wide Area Network) 구간의 네트워크는 LAN 보다 상대적으로 지연시간이 길고, 바이트 당 비용이 크기 때문에 데이터 크기에 민감하다. 본 논문에서는 WAN 구간에서 메시지의 전송 효율을 높이기 위해 DDS(데이터 배포 서비스) RTPS(실시간 발행-구독 프로토콜) 메시지를 변환하는 방식을 제안한다. RTPS 메시지의 헤더 구조를 변경하고, 서브메시지 내의 플래그 필드를 활용하여 서브메시지를 압축하여 전송한다.

I. 서론

DDS(Data Distribution Service, 데이터 배포 서비스)는 OMG(Object Management Group)에서 표준화한 데이터 중심 실시간 발간-구독 통신을 위한 미들웨어 표준이다. DDS는 논리적 데이터 공유 영역인 도메인 네트워크를 동적으로 형성하며, 미들웨어 내부적으로 도메인 네트워크 내에 존재하는 객체들을 검색/관리하기 때문에, 응용프로그램은 통신 대상을 직접 알 필요가 없는 익명성을 기반으로 데이터 송수신을 수행한다. [그림 1]과 같이 DDS의 통신은 데이터 발간자와 데이터 구독자를 통해 이루어지며, 데이터 발간자와 데이터 구독자는 토픽이라는 객체에 의해서 연결되어 데이터의 발간-구독을 수행한다[1].



[그림 1] DDS 통신 구조

DDS는 세부 표준에 따라 DCPS(Data-Centric Publish/Subscribe, 데이터-중심 발행/구독)와 RTPS(Real-Time Publish/Subscribe, 실시간 발행/구독) 프로토콜의 두 표준으로 구성된다. DCPS 표준은 도메인 구성객체, 데이터 발간 인터페이스, 데이터 구독 인터페이스, 데이터 인지 인터페이스 등 데이터를 발간-구독하기 위한 인터페이스를 정의하고, 총 22개의 QoS항목과 QoS 항목들 사이의 관계와 적용 값의 우선순위를 정의한다[2].

본 논문에서는 광역통신망(WAN) 환경에서 메시지의 전송 효율을 높이기 위해 데이터 배포 서비스(DDS) 실시간 발행-구독 프로토콜(RTPS)

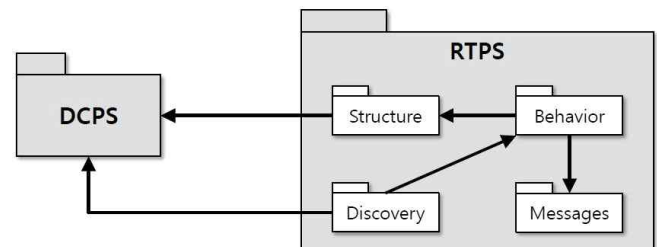
메시지를 변환하는 방법을 제안한다.

II. 본론

2.1 RTPS 개요

RTPS는 도메인을 구성하는 통신 참여자(DomainParticipant) 및 통신단말(endpoint)에 대한 표준 검색 프로토콜을 정의하고, 정의된 QoS에 따라 동작하는 데이터 발간객체와 데이터 구독객체의 동작, 데이터 송수신 상태, 각 객체 내부의 데이터 목록 등을 정의한다. DDS를 구성하는 두 표준에 따라 사용자는 DCPS 표준에 정의된 인터페이스를 이용하여 응용프로그램을 구성하거나 필요한 통신객체들의 동작을 정의할 수 있고, 정의된 DCPS 객체의 정보를 이용하여 DDS 미들웨어 내부에서는 RTPS 표준 정의에 따라 실질적인 통신 기능을 수행한다.

RTPS는 데이터 전송 프로토콜로써 데이터 발행/구독 통신 모델을 지원하며 UDP/IP와 같이 신뢰성 없는 전송 계층 위에서도 동작 가능하도록 설계되었다. RTPS를 구성하는 기본 모듈은 [그림 2]와 같이 4가지로 나눌 수 있다.



[그림 2] DDS RTPS 모듈

- A. Structure Module : Structure Module은 RTPS 표준 규격 상 데이터 교환 시 통신에 참여하게 되는 개체들에 대해 정의
- B. Message Module : Message Module은 RTPS의 Writer와 Reader 간에 정보 교환을 위해 사용될 메시지에 대해 정의
- C. Behavior Module : Behavior Module은 RTPS Writer와 Reader들

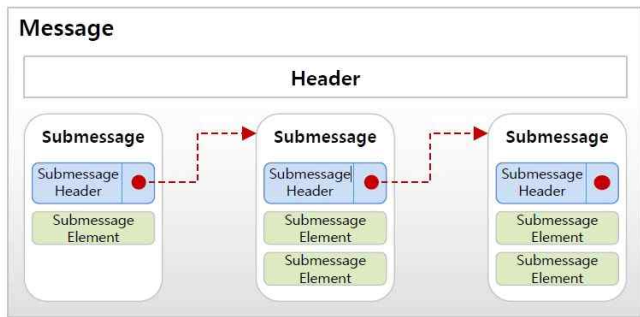
간 상태적, 시간적 조건에 따라 수행되어야 할 메시지 전송 절차에 대해 정의

D. Discovery Module : Discovery Module은 도메인 상에 존재하는 데이터 배포 관련 개체에 대한 정보를 탐색하는 기능을 수행한다. Discovery Module은 RTPS 탐색 프로토콜을 정의 다음과 같은 두 가지 정의된 프로토콜을 사용한다.

- PDP(Participant Discovery Protocol) : 서로 다른 네트워크 상에서의 Participant 탐색을 위한 프로토콜
- EDP(Endpoint Discovery Protocol) : Writer, Reader와 같이 서로 다른 중단점 간의 탐색 정보 교환에 사용되는 프로토콜

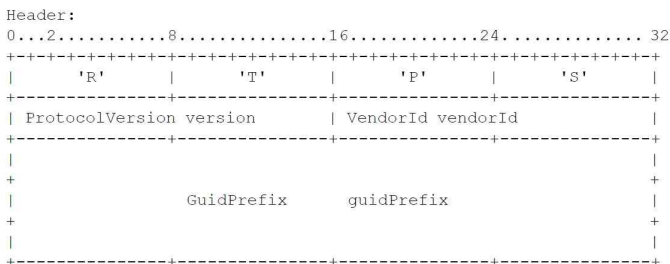
2.2 RTPS 메시지 구조

RTPS 메시지는 [그림 3]과 같이 헤더와 서브메시지의 조합으로 구성되며, 메시지 헤더는 [그림 4]과 같이 'RTPS' 매직(Magic), 프로토콜 버전, 벤더 식별자, guidPrefix 필드로 구성된다.



[그림 3] RTPS 메시지 구조

RTPS 헤더는 총 20바이트 크기이다. RTPS 헤더 내의 첫 번째 필드 4바이트는 'R' 'T' 'P' 'S' 메시지를 표시하는 매직 문자로 구성된다. 두 번째 필드는 현재 프로토콜 버전(2.3)을 표시하며, 메이저(major) 버전(2), 마이너(minor) 버전(3)의 형태로 각 1바이트씩 할당된다. 세 번째 필드는 벤더 식별자(Identifier)로서 [표 1]에 따라 각 DDS 벤더에 할당된 값을 넣을 수 있다. 네 번째 필드는 GuidPrefix로서 hostID, appID, instanceID로 구분되며, 각 4바이트씩 할당된다. hostID 필드는 IPv4 주소를 지정하며, appID는 process ID를 의미하고, instanceID는 카운터(counter)로서 새로운 참여자(participant)에 대해 카운터 값을 증가시키게 된다.

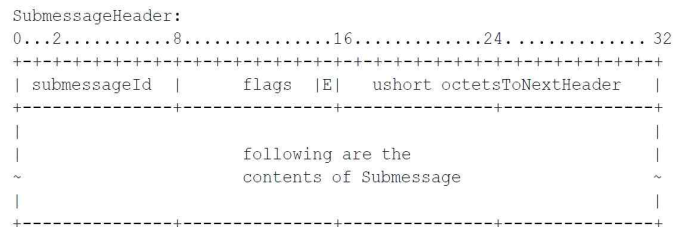


[그림 4] RTPS 헤더 구조

RTPS 메시지 내의 서브메시지는 [그림 5]와 같이 헤더와 서브메시지 내용으로 구성된다. RTPS 서브메시지 내의 서브메시지 헤더는 서브메시지 식별자(ID), 플래그, 다음 헤더, 서브메시지 내용 필드로 구성된다. 서브메시지 식별자 필드는 [그림 6]에 보여지는 것처럼 총 13개의 식별자로 구성되며, 8비트로 표시된다. 데이터 배포 서비스 보안(DDS Security) 표준을 위해 서브메시지 식별자 0x30~3f의 값이 예약되어 있다.

[표 1] DDS RTPS 벤더 ID 및 제품명

RTPS Vendor ID	DDS / RTPS Product Name	Company
{0x01, 0x01}	RTI Connext DDS	Real-Time Innovations, Inc. (RTI)
{0x01, 0x02}	OpenSplice DDS	ADLink Ltd.
{0x01, 0x03}	OpenDDS	Object Computing Inc. (OCI)
{0x01, 0x04}	Mil-DDS	MilSoft
{0x01, 0x05}	InterCOM DDS	Kongsberg
{0x01, 0x06}	CoreDX DDS	Twin Oaks Computing, Inc.
{0x01, 0x07}	Not Active	Lakota Technical Solutions, Inc.
{0x01, 0x08}	Not Active	ICOUP Consulting
{0x01, 0x09}	Diamond DDS	Electronics and Telecommunication Research Institute (ETRI)
{0x01, 0x0A}	RTI Connext DDS Micro	Real-Time Innovations, Inc. (RTI)
{0x01, 0x0B}	Vortex Cafe	ADLink Ltd.
{0x01, 0x0C}	Not Active	PrismTech Ltd.
{0x01, 0x0D}	Vortex Lite	ADLink Ltd.
{0x01, 0x0E}	Geo	Technicolor
{0x01, 0x0F}	FastRTPS, FastDDS	eProsima
{0x01, 0x10}	Eclipse Cyclone DDS	Eclipse Foundation
{0x01, 0x11}	GurumDDS	Gurum Networks, Inc.
{0x01, 0x12}	RustDDS	Afotek

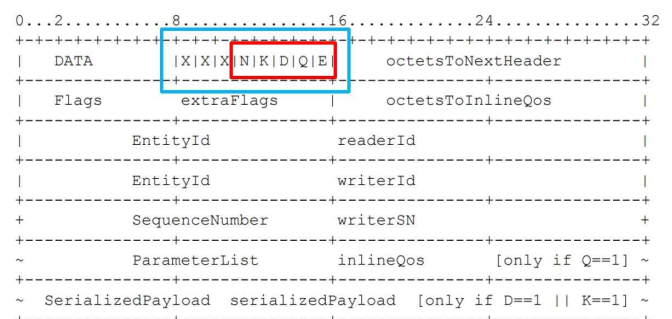


[그림 5] RTPS 서브메시지 구조

```
enum SubmessageKind {
    PAD = 0x01, /* Pad */
    ACKNACK = 0x06, /* AckNack */
    HEARTBEAT = 0x07, /* Heartbeat */
    GAP = 0x08, /* Gap */
    INFO_TS = 0x09, /* InfoTimestamp */
    INFO_SRC = 0x0c, /* InfoSource */
    INFO_REPLY_IP4 = 0x0d, /* InfoReplyIp4 */
    INFO_DST = 0x0e, /* InfoDestination */
    INFO_REPLY = 0x0f, /* InfoReply */
    NACK_FRAG = 0x12, /* NackFrag */
    HEARTBEAT_FRAG = 0x13, /* HeartbeatFrag */
    DATA = 0x15, /* Data */
    DATA_FRAG = 0x16, /* DataFrag */
};
```

[그림 6] RTPS 서브메시지 식별자

플래그 필드는 8비트로 구성되며, 서브메시지 유형에 따라 서로 다르게 사용된다. 예를 들면, [그림 7]에 보여지는 것처럼 데이터 서브메시지에서 8비트의 플래그 중에서 빨간색으로 표시된 5비트만을 사용한다.



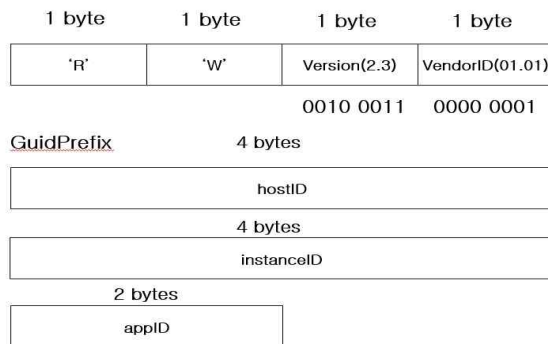
[그림 7] 데이터 서브메시지 내의 플래그

2.3 RTPS 메시지 변환

RTPS 메시지 헤더 내의 첫 번째 필드인 4바이트 길이 'R' 'T' 'P' 'S' 매직 값을 2바이트 'R' 'W'로 변환하여 표시한다. 두 번째 필드인 프로토콜 버전(2.3)은 메이저 버전(2)과 마이너 버전(3)의 2바이트를 1바이트로 줄여서 표현한다.

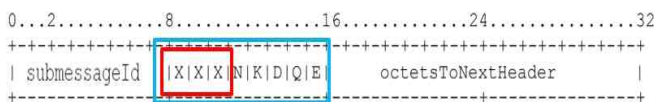
예를 들면, 버전 2.1은 1바이트 형태로 2진수 "0010 0011"로 표시한다. 세 번째 필드인 벤더 ID 필드도 2바이트 크기를 1바이트로 변환하여 표시한다. 2바이트 벤더 ID는 {0x01, 0x03}과 같은 형식으로 표현되는데, 왼쪽 바이트 0x01은 모든 벤더 ID에 동일한 값으로 할당되기 때문에 오른쪽 바이트만으로도 벤더 ID 표현할 수 있다. GuidPrefix 필드 중에서 첫 번째인 hostID는 4바이트 크기로서 IPv4 주소(4바이트)를 표시하므로 바이트 크기를 변경할 수 없다. 두 번째인 appID는 응용 계층의 process ID를 표현한다. 4바이트 크기로 프로세스 ID를 할당한다면, 2^{32} 인 4,294,967,296개의 ID를 할당하는 것이 가능하다. 일반적으로 응용 계층의 process의 개수는 2^{16} 인 65,536개로 충분히 할당할 수 있기 때문에, appID 크기를 2바이트로 줄일 수 있다. 4바이트 instanceID 크기는 변경하지 않는다.

[그림 8]에는 RTPS 헤더 변환 이후의 메시지 크기를 보여준다. RTPS 헤더 크기 20바이트에서 14바이트로 변환되어 6바이트가 줄어드는 효과를 갖는다.



[그림 8] RTPS 헤더 변환

[그림 4]에 보여진 것처럼, 서브메시지 헤더 내의 플래그는 8비트로 구성된다. 서브메시지 유형에 따라 플래그가 사용되는데, [그림 5]에 보여진 데이터 서브메시지가 플래그 5비트를 사용하며, GAP 서브메시지와 NackFrag 서브메시지는 플래그 1비트를 사용한다. 서브메시지 헤더 내의 플래그 8비트 중에서 MSB 3비트는 미사용 비트가 된다. [그림 8]과 같이 미사용 MSB 3비트를 활용하여 서브메시지 내용의 압축 옵션을 설정할 수 있다.



[그림 7] 압축 옵션 설정 플래그

[표 2]에는 3비트 할당 값에 따른 압축 알고리즘의 사용 예를 보여준다. "000" 값은 서브메시지의 압축이 없음을 의미하며, "001" 값은 IPcomp 알고리즘을 이용하여 서브메시지가 압축되었음을 의미한다. "010", "011", "100", "101" 값은 각각 LZ0, LZ4, LZH, Zlib 알고리즘을 이용하여서 서브메시지가 압축되었음을 의미하며, "110"과 "111"은 남겨둔다.

[표 2] DDS RTPS 벤더 ID 및 제품명

압축옵션설정 플래그	압축 알고리즘
000	No compression
001	IP Payload Compression Protocol (IPComp)
010	LZ0
011	LZ4
100	LZH
101	Zlib
110	Reserved
111	Reserved

III. 결론

본 논문에서는 광역통신망(WAN) 환경에서 메시지의 전송 효율을 높이기 위해 데이터 배포 서비스(DDS) 실시간 발행-구독 프로토콜(RTPS) 메시지를 변환하는 방법을 제안하였다. 많은 LAN(Local Area Network)으로 연결된 WAN(Wide Area Network) 환경에서는 네트워크를 통해 전달되는 지연시간이 LAN 환경보다 상대적으로 길고, 데이터 크기와 양에 따라 통신 비용이 높아질 수 있기 때문에, 전송되는 데이터의 양을 줄이는 것이 필요하다. RTPS 메시지의 헤더 구조를 변경하고, 서브메시지 내의 플래그 필드를 활용하여 서브메시지를 압축하여 전송하는 방식을 사용하여 WAN 환경에서의 전송 효율을 개선하는 것이 가능하다.

ACKNOWLEDGMENT

이 논문은 2020년도 정부(중소벤처기업부)의 재원으로 중소기업기술정보진흥원의 지원을 받아 수행된 연구임 (S2894934, 미들웨어 독립적인 DDS Security 기술 개발과 적용)

참 고 문 헌

- [1] Data Distribution Service for Real-time Systems version 1.2, OMG Standard, 2007.
- [2] DDSI-RTPS Specification v2.4, OMG Standard, 2007.
- [3] 이영석, 윤군재, 조성일, 최훈, "데이터 분산 서비스(DDS) 보안을 위한 관리자 참조 모델", 2020년도 한국통신학회 동계종합학술발표회논문집, 2020. 02. 05.