

# 대용량 데이터 전송을 위한 DDS QoS 설정 연구

정현제, 최훈

충남대학교 인공지능학과

충남대학교 인공지능학과

handsome1201@naver.com , [hc@cnu.ac.kr](mailto:hc@cnu.ac.kr)

## Study on DDS QoS Settings for Large Data Transmission

Hyeon Jae Jung, Hoon choi

Chungnam National University, Department of Artificial Intelligence

Chungnam National University, Department of Artificial Intelligence

handsome1201@naver.com, hc@cnu.ac.kr

### 요 약

본 연구에서는 DDS 기반 대용량 데이터 전송의 최적 QoS 설정을 제시하기 위해, RTI Connex DDS를 활용하여 대용량 영상 데이터 송수신 시 QoS 설정이 전송 성능에 미치는 영향을 분석하였다. 내부 시험을 통해, RTI Connex DDS에서 전송 성능에 유의미한 영향을 주는 QoS가 History, Deadline, NACK Delay, Socket Buffer Size 총 네 가지임을 확인했고 네 가지 주요 QoS 설정을 변경하며 400MB 및 500MB 영상이 전송되는데 소요되는 시간을 측정하였다. 실험 결과, QoS를 History=1, Deadline=100ms, NACK Delay=0s, Socket Buffer Size=2MB로 설정했을 때 가장 효율적인 전송 성능을 제공하였다. 또한, keep\_all 설정 시 성능이 저하되고, NACK Delay=0s 설정 시 재전송이 빨랐으며 Socket Buffer Size 증가 시 데이터 전송 안정성이 향상되는 등의 패턴이 확인되었다.

## 1. 서 론

### 1.1 연구 배경 및 필요성

대용량 데이터를 빠르고 안정적으로 송수신하는 것은 다양한 산업 분야에서 필수적이다. 특히, 실시간 데이터 처리가 요구되는 환경에서는 통신 지연을 최소화하고 데이터 손실을 방지하는 것이 핵심 과제이다. 기존의 TCP/IP 기반 클라이언트-서버 방식은 대규모 데이터 전송 시 네트워크 병목 현상과 지연 문제를 초래하며, 확장성이 제한된다는 단점이 있다.

이러한 문제를 해결하기 위해 DDS(Data Distribution Service)가 주목받고 있다. DDS는 발행/구독(publish/subscribe) 모델을 기반으로 하는 분산 통신 프레임워크로서, 실시간 데이터 송수신이 필요한 다양한 산업에서 활용된다. 대표적인 사용 사례는 다음과 같다.

#### • 국방 및 항공 분야

실시간 정보 공유 프레임워크: DDS는 국방 분야의 작전 및 교전 통제 시스템에서 실시간 정보 공유를 위한 소프트웨어 프레임워크로 활용되어 발행/구독 구조의 확장성과 신뢰성을 보장하는 미들웨어로 적용되고 있다 [1].

무기체계 적용: 국산 DDS 미들웨어인 Smart DDS는 국방 무기체계에 도입되어, 기존의 외산 DDS 미들웨어를 대체하고 성능을 검증하였다 [2].

#### • 로봇 공학 및 자동화 분야

다중 모바일 로봇의 작업 계획: Behavior Tree와 DDS를 결합하여 다중 로봇 간의 작업을 조율하는 프레임워크가 제안되었으며, 로봇 간 실시간 데이터 교환이 가능해졌다 [3,4,5].

다중 비용 맵 기반 교통 관리: DDS를 활용한 다중 비용

맵 기반 시스템을 개발하였다. 이를 통해 로봇 간의 충돌을 방지하고, 경로 계획의 효율성을 향상시켰다 [6].

#### • 스마트 그리드 및 에너지 관리

전력 계통 통신 미들웨어 테스트: DDS 기반 통신 미들웨어는 전력 계통의 실시간 데이터 전송에 적용되어, 스마트 그리드의 효율적인 에너지 관리 및 모니터링을 지원한다 [7].

DDS는 QoS(Quality of Service) 설정을 통해 다양한 네트워크 환경에 최적화된 통신을 제공할 수 있으며, 멀티캐스트(multicast)를 지원하여 다수의 노드 간 데이터 전송 효율을 극대화할 수 있다. 그러나 대용량 데이터 송수신 환경에서 DDS의 성능을 극대화하려면 적절한 QoS 설정이 필수적이며, 이에 체계적인 분석이 필요하다.

본 연구에서는 RTI Connex DDS를 사용하여 성능시험을 수행하였다. 해당 기술은 총 56개의 QoS 정책을 지원한다. 이 중 25개는 DDS 표준에 의해 정의되어 있으며, 나머지 31개는 RTI에서 추가한 기능이다. 본 연구의 핵심 목표는 대용량 데이터 전송에 최적화된 QoS 설정을 도출하는 것이며, 이를 위해 다양한 QoS 파라미터를 분석하고 성능을 비교하였다.

### 1.2 연구 목표

본 연구에서는 DDS를 활용하여 대용량 데이터(400MB 및 500MB) 송수신 시 통신 성능에 영향을 미치는 QoS 설정을 분석하고, 최적의 설정 조합을 탐색하는 것을 목표로 한다. 이를 위해 RTI Connex DDS를 이용한 실험을 수행하였으며, 다양한 QoS 설정을 적용한 결과 통신 속도에 유의미한 영향을 미치는 주요 QoS 파라미터 5개

를 도출하였다.

- History 설정: 송신자가 저장하는 데이터 샘플의 개수를 설정하는 파라미터로, 송신자가 저장하는 데이터 개수에 따라 메모리 사용량과 전송 성능이 영향을 받는다. 설정값에 따라 데이터 유실 가능성과 네트워크 부하가 달라진다.
- Deadline 설정: 송신된 데이터가 수신자에게 전달될 수 있는 최대 지연시간을 지정하는 파라미터이다. 해당 시간이 초과되면 DDS는 데이터 손실 가능성을 인지하고 필요한 조치를 수행할 수 있다.
- NACK 지연 설정: 패킷 손실 발생 시 송신자에게 재전송 요청(NACK, Negative Acknowledgment)을 보내는 시간 간격을 조정하는 파라미터이다. 수신자는 손실된 패킷을 감지한 후 즉시 또는 일정 시간 대기 후 송신자에게 NACK을 보내며, 이 설정은 해당 대기 시간을 결정한다.
- 소켓 버퍼 크기: 네트워크 소켓에서 데이터를 버퍼링하는 크기를 결정하는 파라미터이다. 송신자는 데이터를 일정한 크기의 패킷으로 분할 전송하며, 수신자는 이를 네트워크 버퍼를 통해 수신하고 처리한다.
- Reliability 설정 (고정값): 전송의 신뢰성을 보장하는 설정으로서, DDS에서는 BEST\_EFFORT와 RELIABLE 두 가지 모드를 제공한다. BEST\_EFFORT 모드는 최대 전송 속도를 우선하며 패킷 손실이 발생해도 복구하지 않는다. 반면, RELIABLE 모드는 모든 데이터가 반드시 수신되도록 보장하며, 손실된 패킷이 자동으로 재전송된다. 대용량 데이터 전송 환경에서는 데이터 손실을 방지하기 위해 RELIABLE 모드를 사용하는 것이 필수적이다.

본 연구에서는 다양한 QoS 조합을 적용하여 데이터 전송 시간을 비교 분석하였으며, 이를 통해 대용량 데이터 송수신 시 최적의 QoS 설정을 도출하였다.

## 2. 실험 및 분석

### 2.1 실험 환경

RTI Connex DDS를 활용하여 대용량 영상 데이터를 송수신하는 실험을 수행하였다. 실험은 동일 네트워크(LAN) 내 유선 이더넷(Ethernet) 연결(1Gbps) 환경에서 안정적인 데이터 송수신이 가능하도록 설정하였다. 송신자(Publisher)와 수신자(Subscriber)의 하드웨어 및 소프트웨어 사양은 다음과 같다.

- 송신자(Publisher): Intel Core i7-12700F, 32GB RAM, Windows 11 Pro
- 수신자(Subscriber): Intel Core i5-1135G7, 32GB RAM, Windows 11 Pro
- 네트워크 환경: 동일 네트워크(LAN) 내 유선 이더넷 연결(1Gbps), 별도의 네트워크 트래픽이 존재하지 않는 독립적인 네트워크

실험에서 사용된 DDS 솔루션은 RTI Connex DDS이며, 송신자는 Publisher 역할을 수행하여 데이터를 발행하고, 수신자는 Subscriber 역할을 수행하여 데이터를 수신하는 방식으로 진행되었다. 실험 대상 영상은 도로 상황

모니터링 CCTV 영상이며, 해당 영상의 사양은 다음과 같다.

- 해상도: 1920×1080 (Full HD)
- 프레임 속도: 30fps (초당 30 프레임)
- 비디오 포맷: MP4 (H.264 코덱 사용)
- 영상 크기: 400MB, 500MB
- 통신 방식: 송신자(Publisher)와 수신자(Subscriber) 간 1:1 DDS 통신

송신자 및 수신자 코드는 Python으로 작성했으며, RTI Connex DDS의 QoS 설정 파일(QOS\_PROFILES.xml)을 활용하여 실험 조건을 제어하였다.

### 2.2 실험 방법

본 연구에서는 DDS를 이용한 대용량 영상 데이터 전송 실험을 수행하기 위해, 송신 측에서 영상을 1024바이트 크기의 데이터 조각으로 분할하여 전송하고, 수신 측에서 이를 순차적으로 수신한 후 최종적으로 하나의 영상 파일로 복원하는 방식을 적용하였다.

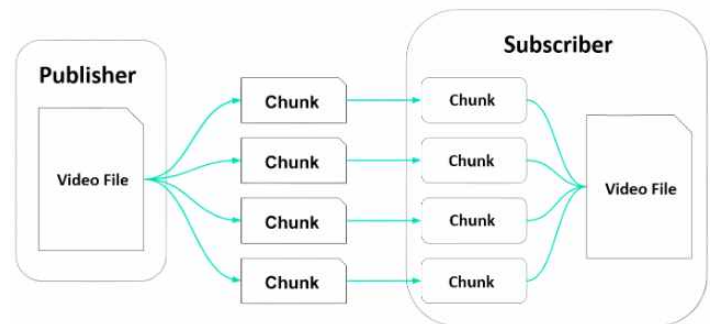


그림 1 대용량 영상 데이터 전송 방식

실험의 신뢰성을 확보하기 위해, 각 실험 조건에서 동일한 영상 파일을 100회 반복 전송하여 평균 전송 시간 및 표준 편차를 측정하였다. QoS 설정값은 DDS가 제공하는 다양한 파라미터 중 대용량 데이터 전송 성능에 가장 큰 영향을 미치는 항목들을 선정하여 실험을 진행하였다. 각 QoS 설정의 의미와 실험에서 적용한 값은 다음과 같다.

- History 설정: 설정값은 1개 저장(History=1), 5개 저장(History=5), 전체 저장(keep\_all)으로 진행했으며, 데이터 저장 개수가 많을수록 메모리 사용량이 증가하고 과도한 버퍼링이 오히려 전송 지연을 초래할 수 있기에 가장 작은 1개 저장과 5개 저장 그리고 가장 많은 양을 저장하는 전체 저장으로 비교 분석하였다.
- Deadline 설정: 설정값은 50ms, 100ms로 했으며, 짧은 Deadline 값이 전송 지연을 줄이는 데 도움이 되지만, 지나치게 짧으면 패킷 손실이나 과부하가 발생할 가능성이 있다. 따라서 네트워크 환경에서 안정적인 성능을 제공하는 적절한 임계값을 찾기 위해 50ms와 100ms 두 가지 값을 비교 분석하였다.
- NACK Delay 설정: 설정값은 즉시 재전송(0s), 1초 후 재전송(1s)로 했으며, 즉각적인 NACK 요청이 데이터 복구 속도를 높일 수 있지만, 너무 잦은 요청은 네트워크 부하를 증가시킬 수 있다. 따라서 즉각적인

재전송이 전송 속도에 미치는 영향과 일정 시간 대기 후 재전송하는 방식의 차이를 비교하기 위해 두 가지 설정을 실험하였다.

- Socket Buffer Size 설정: 설정값은 1MB, 2MB로 했으며, 네트워크 트래픽과 버퍼 크기 간의 관계를 분석하여 최적의 전송 속도를 보장하는 설정을 찾기 위해 실험하였다. 작은 버퍼 크기(1MB)에서 발생하는 데이터 전송의 병목 현상과 보다 큰 버퍼 크기(2MB)가 성능 향상에 미치는 영향을 비교하기 위해 두 가지 값을 선택하였다.

400MB, 500MB 각각 24가지 QoS조합( $3 \times 2 \times 2 \times 2 = 24$ )를 구성하여 총 48가지 실험 케이스 각 QoS 설정 조합이 데이터 전송 성능에 미치는 영향을 비교하였다. 모든 실험 데이터는 텍스트 파일로 저장한 후 분석을 진행하였다.

### 2.3 실험 결과

첫 번째 실험에서는 QoS 정책의 각 옵션값이 대용량 영상 데이터 전송 성능에 미치는 영향을 분석하였다. 400MB와 500MB 영상 파일을 대상으로 각 QoS 정책(Deadline, History, NACK, Socket Buffer)을 개별적으로 변경하면서 전송 시간을 측정하였다. 즉, 3개의 QoS 값은 고정된 상태에서 나머지 1개의 파라미터를 변경하며 측정하는 방식으로 실험을 진행하였으며, 그 결과는 다음과 같다.

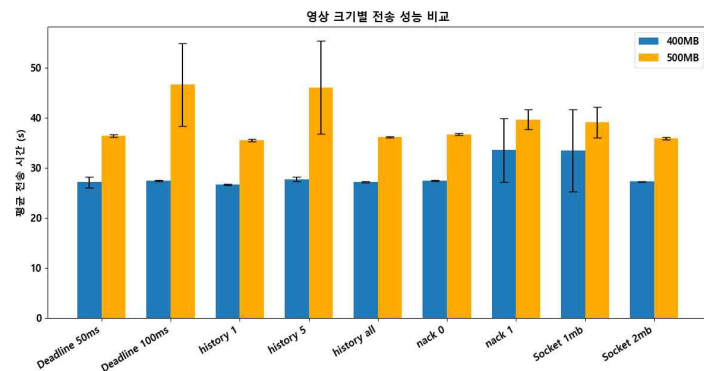


그림 2 QoS 정책 옵션별 전송 시간 비교

실험 결과, 데이터 크기가 증가할수록 즉각적인 패킷 재전송(NACK 0)과 큰 소켓 버퍼(2MB) 설정을 적용할 경우 전송 속도가 빨라지는 경향을 보였다. 또한, History=5 설정에서 가장 긴 전송 시간이 관찰되었다. 반면, keep\_all 설정은 모든 메시지를 저장하지만 송신자가 추가적인 메시지 관리를 수행하지 않으므로 예상보다 큰 지연이 발생하지 않았기 때문에 5보다 더 빠른 결과를 보였다. 따라서, History=1과 keep\_all 설정 간의 성능 차이는 크지 않았던 반면, History=5에서는 불필요한 관리 비용이 추가되면서 전송 시간이 상대적으로 길어지는 현상이 확인되었다.

이러한 QoS 설정이 전송 속도에 미치는 영향을 보다 정밀하게 분석하기 위해, 다양한 QoS 조합을 적용하여 전송 시간을 측정하였다. 아래 그림은 영상 크기별(400MB,

500MB) 평균 전송 시간을 히트맵으로 나타낸 것이다. 색상이 연할수록(파란색) 전송 시간이 길며, 색상이 진할수록(붉은색) 전송 시간이 짧음을 의미한다. 각 행은 History 및 NACK Delay 설정을, 각 열은 Deadline과 Socket Buffer Size 설정을 나타낸다.

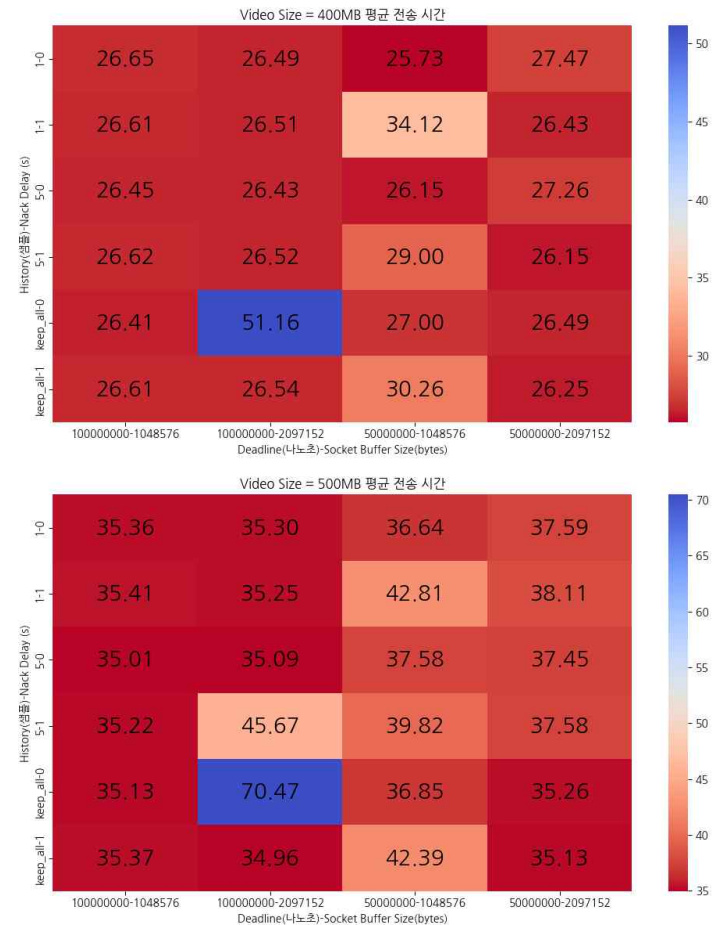


그림 3 QoS 정책 옵션별 전송 시간 비교

위 결과를 분석한 주요 패턴은 다음과 같다.

- History 값 증가(특히 History=5) 시 성능 저하  
History=1과 keep\_all의 전송 속도 차이는 크지 않았으나, History=5에서는 가장 긴 전송 시간이 관찰되었다. 이는 송신자가 일정 개수 이상의 메시지를 저장할 경우, 데이터 관리 부담이 증가하여 오히려 전송 속도가 느려지는 현상이 발생했기 때문으로 해석된다. 따라서, History 값을 불필요하게 증가시키기보다는 최소한(1)으로 유지하는 것이 바람직하다.

- Deadline이 짧을수록 안정적인 성능 제공  
Deadline을 50ms로 설정한 경우, 100ms보다 전송 시간이 전반적으로 짧았으며, 표준 편차도 작아 보다 일관된 성능을 제공하였다. 반면, 100ms 설정에서는 일부 전송 구간에서 전송 속도 변동성이 다소 커지는 경향이 나타났다. 따라서, 실시간 데이터 전송이 중요한 환경에서는 보다 짧은 Deadline(50ms)이 유리하다.

- NACK Delay(0s 설정)가 성능 향상에 기여  
NACK Delay를 0s로 설정한 경우, 패킷 손실이 발생하면 즉시 재전송이 이루어져 전송 시간이 단축되었다. 이에 반해, 1s 딜레이를 적용한 경우에는 재전송이 늦어져

전송 속도가 더 느려졌다. 특히, 대용량 데이터를 송수신하는 환경에서는 지연시간을 최소화하기 위해 NACK Delay를 0s로 설정하는 것이 효과적이다.

- Socket Buffer Size 증가에 따른 성능 변화

Socket Buffer Size를 2MB로 설정한 경우, 1MB보다 전송 시간이 단축되고 변동성이 감소하는 경향이 관찰되었다. 특히, 500MB 데이터 전송 시 Socket Buffer Size를 2MB로 설정했을 때, 보다 안정적인 성능을 제공하였다. 이는 더 큰 버퍼 크기가 데이터 패킷의 처리 속도를 개선하고, 네트워크 전송 과정에서 발생하는 지연을 줄이는데 기여했기 때문으로 분석된다.

실험 결과를 바탕으로, 대용량 영상 데이터를 효율적으로 전송하기 위한 최적의 QoS 설정 조합은 다음과 같다.

- History = 1 (불필요한 메시지 저장 최소화)
- Deadline = 50ms (데이터 전달 주기 조정)
- NACK Delay = 0s (빠른 패킷 재전송 가능)
- Socket Buffer Size = 2MB

이러한 설정은 송신 측에서 불필요한 메시지 저장을 줄이고, 손실된 데이터의 빠른 재전송을 가능하게 하며, 네트워크 버퍼를 효과적으로 활용할 수 있도록 한다. 따라서, 실시간성을 요구하는 대용량 데이터 전송 환경에서 QoS 최적화에 대한 실용적인 가이드를 제공할 수 있다.

### 3. 결 론

본 연구에서는 RTI Connnext DDS를 활용하여 대용량 영상 데이터를 송수신하는 실험을 수행하였으며, 다양한 QoS 설정이 전송 성능에 미치는 영향을 분석하였다. 실험 결과, History 저장 수량을 작게, Deadline을 짧게, 재전송 지연시간을 짧게, Socket Buffer Size를 크게 할수록 효율적인 전송 성능을 제공하는 것으로 확인되었다.

이러한 연구 결과는 DDS를 활용한 대용량 데이터 전송 환경에서 최적의 QoS 설정을 결정하는 데 중요한 기준이 될 수 있으며, 실시간 데이터 처리가 요구되는 다양한 산업 분야에서 QoS 최적화를 위한 실질적인 가이드라인으로 활용될 수 있을 것으로 기대된다.

### 4. 참고문헌

- [1]김기태, 양승현, 정원우. "DDS 기반의 실시간 정보공유 프레임워크 사례 연구." 정보과학회 컴퓨팅의 실제 논문지 26, no.4: 202-210. doi: <https://doi.org/10.5626/KTCP.2020.26.4.202>. (2020)
- [2]최윤석. "Smart DDS 미들웨어 국산화 및 무기체계 적용." 정보과학회지, 34(10): 43-46. (2016)
- [3]Open Robotics. "ROS 2 Documentation (Foxy)." Retrieved February 11, 2025, from <https://docs.ros.org/en/foxy/index.html>.
- [4]AUTOSAR. "AUTOSAR - The Automotive Open System Architecture." Retrieved February 11, 2025, from <https://www.autosar.org>.
- [5]Seungwoo Jeong, Taekwon Ga, Inhwon Jeong, & Jongeun Choi. "Behavior tree driven multi-mobile robots

via data distribution service (DDS)." 제어로봇시스템학회 국제학술대회 논문집, 제주. (2021-10-12)

- [6]Jeong, Seungwoo, et al. "Layered Cost-Map-Based Traffic Management for Multiple Automated Mobile Robots via a Data Distribution Service." arXiv preprint arXiv:2207.08902. (2022)

- [7]김현옥, 최선, 백원길, 강치원. "데이터분산서비스 통신 미들웨어 테스트를 위한 테스터기 설계." 한국정보과학회 학술발표논문집, 제주. (2017-06-18)