자율분산시스템의 통신 메커니즘 QoS 향상에 관한 연구
A study for QoS improvement of communication mechanism on the autonomous decentralized system

이영수*, 안진*, 김병곤* , 홍순흠**, 김영훈**
Lee, Young Su Ahn, Jin Kim, Byoung Gon Hong, Soon Heum Kim, Young Hoon

---

ABSTRACT

자율분산 역 제어시스템의 핵심 통신 미들웨어인 자율분산관리자(ADM; Autonomous Decentalized Manager)는 기존의 피어-투-피어 방식의 수동형 통신보다 브로드 캐스트 및 멀티 캐스트 방식의 능동형 통신을 지원함으로써 제어시스템의 능동성, 자율 제어성, 빠른 처리속도를 확보할 수 있지만 상대적으로 기존의 피어-투-피어 방식에 비해서 신뢰성 및 안정성이 다소 떨어지는 문제점을 가지고 있다. 본 논문에서는 이러한 문제점을 극복하기 위해서 송수신의 신뢰성 및 안정성을 극복하기 위해서 필요한 다양한 통신 메커니즘의 품질 속성(QoS)에 대해서 기술한다.

---

1. 서론

현재 열차운행제어기술은 모든 철도 네트워크에 대한 이산형 중앙집중형 네트워크로 구축되어 있다. 이 방식은 운행밀도가 낮거나 운행패턴이 단순한 경우에 효과적이나 복잡하면서 다양한 형태의 철도를 연계하여 운행해야 하는 경우에는 비효율적인 네트워크 구조를 가지고 있다. 특히 트래픽의 밀도가 높은 대형고밀도 역의 경우 운행 밀도가 매우 높고 운행 패턴 또한 복잡하여 중앙집중형 관리 방식 구조로는 관리상의 어려운 점이 매우 많다. 이런 대형고밀도 역은 중앙집중형 관리 시스템이 구축되어 있다 하더라도 상시 로컬 시스템 형태로 운행하는 경우가 대부분이다. 이러한 문제를 극복하기 위해서 트래픽을 최적화시키고 선로의 용량을 최대한 사용할 수 있도록 하기 위해서 자율분산 역 제어시스템이라는 새로운 시스템을 연구하게 되었다.

자율분산 역 제어시스템의 핵심인 자율분산관리자는 자율분산 역 제어시스템을 구성하는 다양한 서브시스템의 통신 및 생명주기를 다루는 미들웨어로써 다양한 메시지의 송수신 방법을 제공한다. 자율분산관리자는 기존의 시스템에서 보았던 수동형 피어-투-피어 송수신을 지원하는 것 뿐만 아니라 능동형 브로드 캐스트/멀티 캐스트 형태의 송수신 방식을 지원한다. 자율분산관리자의 가장 기본 송수신 방법은 브로드 캐스트/멀티 캐스트 방식은 기존의 피어-투-피어 방식에 비해서 빠른 응답속도, 능동형 통신을 제공하지만 상대적으로 안정성 및 신뢰성이 다소 떨어지는 문제점이 있다. 본 논문에서는 자율분산 역 제어시스템의 자율분산관리자의 통신 모델에 대한 신뢰성 및 안정성 향상을 위한 품질 속성을 살펴본다.

본 논문의 2장에서는 자율분산관리자의 요구사항을 살펴보고, 3장에서는 자율분산 역 제어시스템의 송수신 메시지의 유형과 품질 속성을 정의하고, 4장에서는 이러한 품질 속성을 지원하기 위해서 필요한 자율분산관리자의 품질 속성(QoS; Quality of Service)을 설계하고 5장에서 결론을 맺는다.

---

* 경봉기술
**한국철도기술연구원

2. 자율분산관리자

자율분산관리자는 자율분산시스템을 구현하는데 필수적인 구성요소다. 자율분산관리자는 자율분산시스템을 구성하고 있는 여러 서브시스템이 서로 메시지를 송수신할 때 통신 환경을 제공하고, 또한 서브시스템에 대한 생명주기를 책임지는 역할을 가지고 있다. 또한 자율분산관리자는 시스템이 정지하지 않고 유연한 방법으로 전체 시스템을 확장할 수 있는 환경을 구성할 수 있도록 가장 기본적인 환경을 제공한다. 따라서 이러한 전체 시스템을 구성하기 위해서 다음과 같은 요구사항을 도출할 수 있다.

□□ 유연한 시스템 아키텍처

- 빠른 설치 및 유지보수 비용의 감소
- 시스템의 유연한 온라인 확장
- 매우 단순한 구조
- 설정 및 구축의 용이
- 다양한 OS를 지원하면서 동일한 기능을 제공

□□ 신뢰성 있는 시스템 아키텍처

- 복수의 네트워크 지원
- 다양한 Quality of Service 지원

□□ 뛰어난 성능의 시스템 아키텍처

- 멀티 쓰레드 구조의 병행 처리
- 비동기 방식의 큐잉 처리

자율분산관리자(ADM; Autonomous Decentalized Manager)에게 요구되는 다양한 요구사항 중에서 자율분산 역 제어시스템을 구축하기 위한 가장 중요한 요소가 바로 네트워크 기능이다. 전체 시스템을 구성하는 개별적인 애플리케이션을 프로세스라고 할 때 그림 1은 프로세스간의 메시지 교환을 자율분산관리자가 어떻게 수행하는지 보여주는 그림이다.

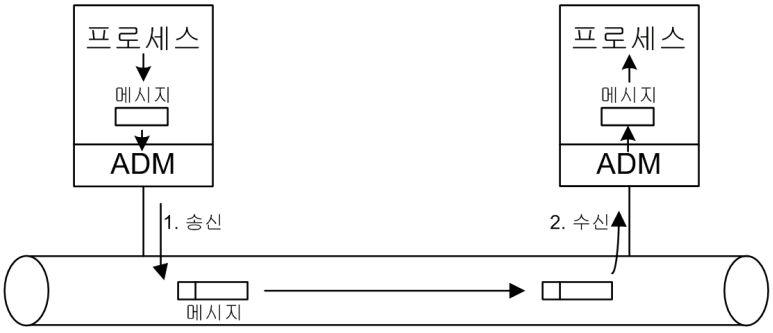

그림 1 프로세스와 자율분산관리자의 메시지 송수신

메시지는 서로 교환하고자 하는 데이터의 추상화된 표현이며 프로세스는 이런 메시지를 서로 교환함으로써 시스템을 동작하게 한다. 프로세스는 그림 1에서 보는 것 처럼 자율분산관리자라는 통신 미들웨어를 통해서 메시지를 서로 주고 받게 된다. 따라서 자율분산관리자가 없다면 자율분산 역 제어시스템을 구축이 불가능해 지며 자율분산관리자의 능력에 따라 전체 시스템의 자율분산 및 제어 능력이 결정된다고 할 수 있다.

3. 메시지와 품질 속성과의 관계

자율분산 역 제어시스템을 구성하는 서브시스템과 프로세스는 다음과 같다.

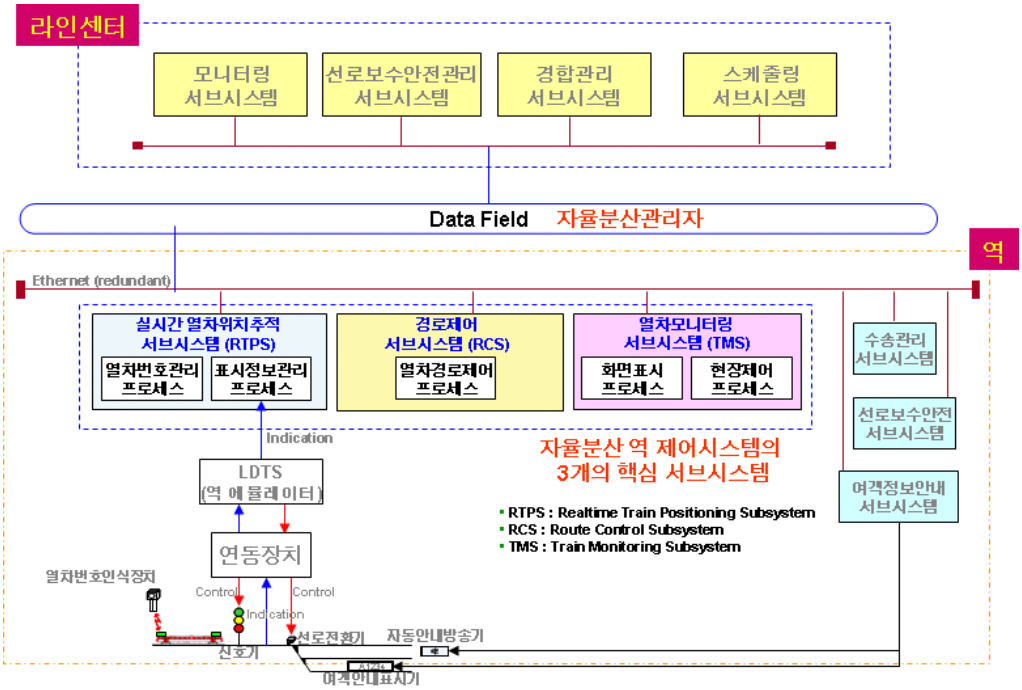

그림 2 자율분산 역 제어시스템의 서브시스템들

개별 서브시스템은 독립적으로 수행하는 프로세스 별로 구성되어 있으며 서브시스템당 하나의 자율분산관리자가 구동하게 된다. 그림 2에서 보는 것 처럼 RTPS, RCS, TMS 서브시스템은 자율분산 역 제어 시스템을 구성하는 가장 핵심 서브시스템으로써 이들 시스템간의 자율분산적인 시스템 운영을 위해서 다양한 형태의 메시지를 서로 교환하여 유기적으로 역 시스템을 구축하게 된다.

이들 핵심 서브시스템이 서로 유기적인 자율분산 역 제어시스템을 구축하기 위해서 사용하는 메시지는 다음과 같다. 메시지는 중요한 메시지만 선별하였다.

도표 1. 자율분산 역 제어시스템의 메시지

메시지	설명	품질 속성
열차 추적 상태 메시지	RTPS가 열차의 추적 상태를 주기적으로(약 1,000ms 예상) 다른 서브시스템과 센터로 전송한다.	주기적 송신, 누락 허용
라인업 메시지	RTPS는 열차 추적 기능을 하면서 TMS에 표시할 라인업을 관리하여 주기적으로 송신한다.	주기적 송신, 누락 허용
현장 상태 메시지	RTPS가 주기적으로 현장 상태를 다른 서브시스템과 센터로 전송한다.	주기적 송신, 누락 허용
현장 상태 이벤트 메시지	현장 상태가 변경될 때 마다 전송한다.	이벤트 발생시 송신, 누락 허용안함
현장 제어 메시지	RTPS는 RCS의 자동 진로 제어 메시지와 TMS의 개별 제어 메시지를 수신하여 LDTS로 제어 명령을 내린다.	이벤트 발생시 송신, 누락 허용안함
열차 번호 제어 메시지	RTPS는 TMS의 열번 제어 메시지를 수신하여 열차 추적에 반영한다.	이벤트 발생시 송신, 누락 허용안함
스케질 전체 송신 메시지	하루의 운행을 시작하기 전 자정에 센터로부터 수신한다.	1일 1회 송신, 누락 허용안함
스케줄 변경 메시지	스케줄이 변경되었을 때 수신한다.	이벤트 발생시, 누락 허용안함
실적 전체 송신 메시지	시스템 기동시에 센터로 부터 수신한다.	시스템 기동시 수신, 누락 허용 안함

메시지	설명	품질 속성
실적 발생 메시지	RTPS가 RCS, TMS, 센터로 송신한다.	이벤트 발생시 송신, 누락 허용안

도표 1를 보면 개별 메시지 마다 품질 속성이 서로 다름을 알 수 있다. 현장 제어 메시지의 경우 이벤트 발생시 마다 송신하지만 절대로 누락되어서는 안되는 메시지다. 이와는 다르게 열차 추적 상태 메시지의 경우는 주기적으로 송신하지만 누락되어도 크게 무방한 메시지다. 이렇듯이 개별 메시지는 메시지가 필요로 하는 고유의 품질 속성이 존재한다. 자율분산관리자가 이러한 메시지의 품질 속성을 고려하여 안정적인 네트워크를 구축하고 송수신하려면 다양한 품질 속성(QoS; Quality of Service)를 지원하는 것을 필수적이라 하겠다.

4. Quality of Service (QoS)

비록 자율분산관리자가 다양한 네트워크 송수신 방법을 제공하기는 하지만 이것만으로 충분히 안정적인 메시지 송수신이 가능한 것은 아니다. 도표 2는 자율분산관리자가 제공하는 송수신 방식에 따른 장·단점을 설명한 것이다.

도표 2. 송수신 방식의 장·단점

송수신 방법	장점 및 단점
브로드 캐스트 / 멀티 캐스트	• 빠른 송수신이 가능하다. • 목적지가 정해져 있지 않아서 수신측을 지정할 필요가 없다. • 안정성이 떨어져서 메시지의 손실 위험이 있다. • 메시지의 순서가 변경되어 일관성이 훼손될 수 있다. • 대용량의 메시지 처리에 적합하지 않다.
피어-투-피어	• 네트워크 연결 시도시 시간소비가 심하다. • 지정한 목적지로만 송수신이 가능하다. • 안정적인 전달이 가능하다. • 송수신 내용의 순서가 보장된다. • 대용량의 메시지 송수신에 적합하다 • 다수의 메시지를 송수신할 때 지연이 심하다.

자율분산 역 제어시스템에서 자율분산관리자는 자율분산 역 제어시스템을 구성하는 다양한 프로세스간의 안정적인 네트워크 송수신을 위해서 위와 같이 송수신 방식을 결정할 수 있기는 하지만 두 네트워크 송수신 기법은 그 차이가 명확하여 어느 한 쪽만 선택하여 사용한다는 것은 어려운 일이라 할 수 있다. 이런 이유로 자율분산 역 제어시스템의 핵심인 자율분산관리자는 안정적인 메시지 송수신 및 높은 수준의 품질 속성을 확보하기 위한 다양한 통신 메커니즘이 필요하다.

이 절에서는 다음의 Quality of Service 속성에 대해서 설명하도록 하겠다.

- 커스텀 헤더 속성
- 메시지 필터링
- 메시지 우선순위 제어
- 메시지 분할 전송
- 메시지 승인
- 메시지 재전송
- 메시지 생명주기

- 트랜잭션 처리
- 메시지 지속성

개별 QoS 속성은 특징이 있으며 하나 이상의 QoS 속성을 사용하여 메시지 송수신을 수행할 수 있다.

4.1 커스텀 헤더 속성

메시지를 송수신하는 주체인 서브시스템의 프로세스들은 보내고자 하는 내용(예;열차 스케줄)을 추상화된 메시지의 본문에 내용을 담아서 송수신하게 된다. 자율분산 역 제어시스템을 구성하는 서브시스템은 본문 내용에 핵심적인 메시지를 넣어서 보낼 수도 있지만 이 방식은 메시지를 별도로 구현해야 하기 때문에 변화에 대한 가능성이 높다. 따라서 이런 문제를 극복하기 위해서 커스텀 헤더 속성을 지원할 수 있다. 커스텀 헤더 속성은 자율분산관리자에게 메시지를 보내기 전에 헤더에 송신자가 원하는 데이터를 헤더에 추가할 수 있는 기능이다.

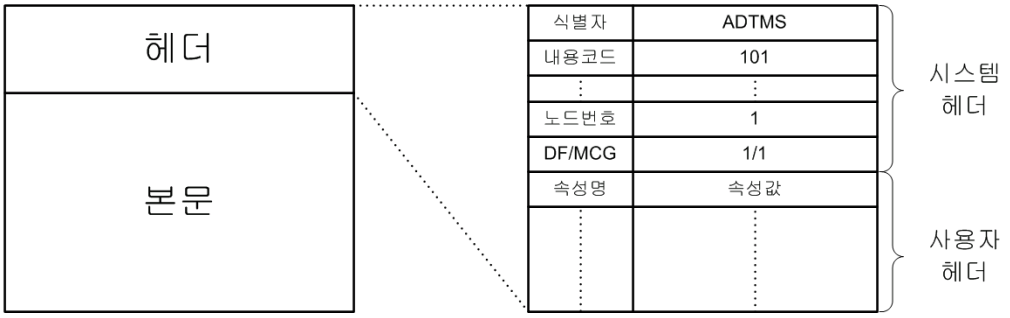

그림 3 커스텀 헤더 속성

그림 3과 같이 헤더는 시스템 헤더와 사용자 헤더로 구분할 수 있다. 시스템 헤더의 경우 자율분산관리자가 직접 설정하는 것으로 응용 소프트웨어 모듈은 이 헤더값을 변경하거나 사용할 수 없다. 다만 사용하 헤더 속성을 통해서 응용 소프트웨어 모듈은 사용하고자 하는 헤더 값을 속성명과 속성값을 통해 지정할 수 있으며 수신측 응용 소프트웨어 모듈은 이 정보를 사용할 수 있게 된다. 이 커스텀 헤더는 다음과 같이 간단한 방식으로 헤더를 설정할 수 있을 것이다.

```
ScheduleMessage msg = new ScheduleMessage(); // 열차 스케줄 메시지 생성
msg.setDate("2006-03-01");
msg.setSchedule(...);
Session session = ADM.getSession(); // 메시지 송신을 위한 세션 획득
session.setCustomHeader("속성명", "속성값"); // 커스텀 헤더 설정
session.send(msg); // 메시지 송신
```

이 QoS 속성은 송신하고자 하는 값을 메시지에 추상화 시키지 않더라도 헤더에 커스텀 속성을 추가할 수 있다는 점에서 유용하게 사용할 수 있는 QoS 속성이다.

4.2 메시지 필터링

이 메시지 필터링 속성은 응용 소프트웨어 모듈이 메시지를 필터링할 때 사용할 수 있다. 일반적인 송수신 메시지의 경우에는 메시지를 필터링 기능을 부가적으로 제공하지 않으며 수신한 메시지를 그대로 프로세스에게 전달하게 된다. 그러나 메시지 필터링 QoS 속성을 활용하면 필요에 따라서 메시지를 선별하여 수신할 수 있게 된다.

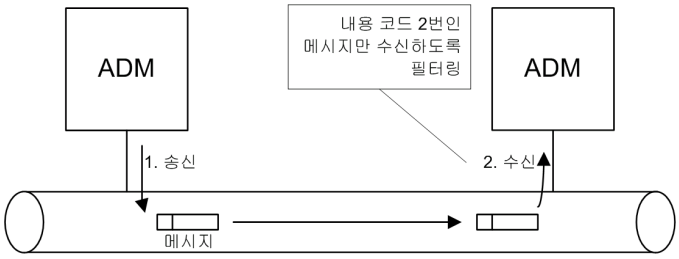

그림 4 메시지의 헤더값을 기반으로 필터링

그림 4에서 보는 것 처럼 응용 소프트웨어 모듈간의 메시지 송수신시 메시지 필터링을 적용하면 수신측 응용 소프트웨어 모듈은 수신하는 모든 메시지에 대해서 처리하는 것이 아니라 필요한 메시지만 처리할 수 있게 된다. 이것은 응용 소프트웨어 모듈의 안정성을 향상시키고 자원의 가용성을 증가시킨다.

```
ServiceContext ctx = ADM.getServiceContext();
ctx.setMessageQuery("ccode = 2"); // 메시지를 SQL 기반으로 필터링
ProcessQueue queue = ctx.getProcessQueue();
Message msg = queue.receive(); // 메시지를 수신
```

위 샘플 코드와 같이 응용 소프트웨어 모듈에서는 SQL 기반의 쿼리문을 지정하면 내용 코드가 2번인 메시지만 선별하여 수신할 수 있게 된다.

4.3. 메시지의 우선순위 제어

자율분산관리자는 응용 소프트웨어의 메시지의 처리 우선순위 기능을 제공할 수 있다. 제어 메시지의 경우 신속하면서 정확하게 수신측 응용 소프트웨어에 전달되어야 한다. 자율분산관리자가 많은 양의 메시지를 동시에 처리하고 있다고 가정할 경우 현재 즉시 제어해야 할 메시지에 대해서 가장 우선순위를 높게 지정하여 가장 먼저 메시지를 처리할 수 있도록 해야 한다. 이런 기능은 메시지의 우선순위 제어 기능을 통해서 구현이 가능하다.

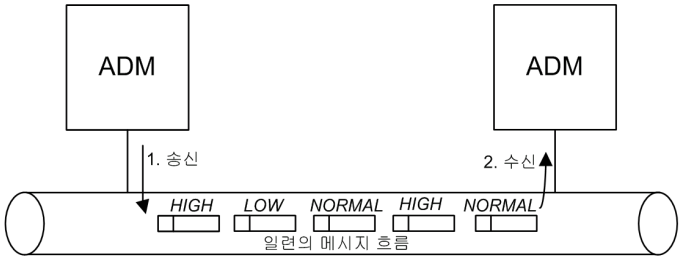

그림 5 메시지의 우선순위 제어

그림 5와 같이 송신측 자율분산관리자는 다양한 우선순위를 가진 메시지를 송신할 수 있다. 가장 나중에 송신한 메시지의 경우 우선순위가 HIGH로 지정되어 있으며 가장 먼저 보낸 메시지는 NORMAL로 지정되어 있다. 만약에 송신한 5개의 메시지가 동시에 수신측 자율분산관리자의 큐에 들어갔다고 가정하면 자율분산관리자는 우선순위가 가장 높은 HIGH로 설정되어 있는 메시지를 순서대로 처리하게 된다.

4.4 메시지의 분할 전송

메시지 분할 전송 기능은 과도하게 큰 메시지를 멀티 캐스트로 송신할 때 안정적인 메시지 전달을 위해서 꼭 필요한 기능이라고 할 수 있다. 일반적으로 대용량의 데이터를 보내는 경우에는 피어-투-피어

방식으로 전송하는 것이 더 적합하지만 환경에 따라서 피어-투-피어 방식을 사용하기 힘든 경우도 발생할 수 있다. 이런 이유로 멀티 캐스트 환경에서 안정적인 메시지 전달을 위해서 일정한 크기를 초과하는 메시지는 분할하여 송신할 수 있다. 보통의 경우 메시지를 분할 전송하는 경우는 자율분산 역 제어시스템에서 전체 열차 스케줄 정보가 해당할 수 있다. 수십 킬로 바이트에서 수 메가 바이트까지 가능한 열차 스케줄 데이터는 크기가 멀티 캐스트 방식으로 한 번에 처리할 수 없으므로 메시지 분할 전송 QoS 정책을 사용하여 안정성을 확보할 수 있다.

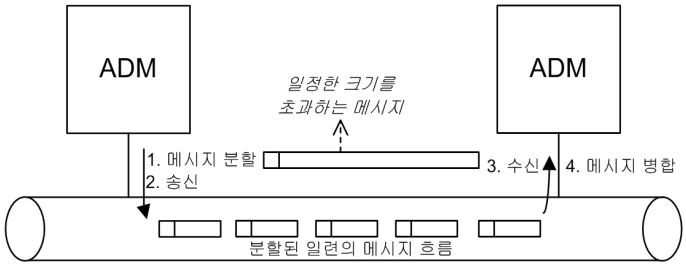

그림 6 메시지의 분할 전송

메시지 분할 전송은 메시지를 단순하게 분할하여 전송하는 것으로 충분한 안정성을 확보하는 것은 아니다. 총 5개의 메시지 중에서 2번째 메시지를 손실하였을 경우 메시지의 안정성을 좀더 유연하게 확보하려면 반드시 재전송을 해야 한다. 이런 이유로 메시지 분할 전송은 반드시 재전송와 함께 사용하여야 한다.

4.5 메시지 승인

메시지 승인은 반드시 메시지가 도착해야 하는 경우 사용할 수 있는 가장 좋은 QoS 정책이다. 예를 들어 열차 제어 메시지의 경우 지정한 시간에 정확하게 전달해야 하기 때문에 이런 제어 메시지의 경우 정확하게 목적지가 송신하려면 피어-투-피어 방식을 사용하거나 또는 멀티 캐스트 방식에 메시지 승인 QoS 정책을 사용하면 가장 좋은 안정성을 확보할 수 있다. 이 메시지 승인은 피어-투-피어 및 멀티 캐스트 방식 모두 사용이 가능한 정책이지만 멀티 캐스트 방식에서 많은 효과를 기대할 수 있다.

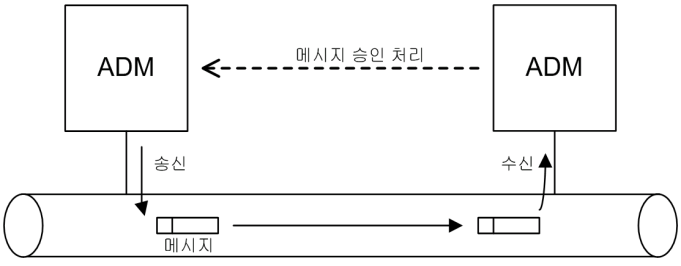

그림 7 메시지 승인 처리

그림 7에서 처럼 자율분산관리자는 메시지를 송신한 이후에 수신측이 제대로 수신했는지 대기하게 된다. 수신측인 메시지를 수신한 이후에는 메시지의 승인 처리를 시도하게 되며 이때 송신측은 메시지가 제대로 전달되었음을 확인하게 된다. 멀티 캐스트 송수신의 경우 메시지의 수신 여부를 확인할 수 있는 장치가 없기 때문에 메시지 승인 정책은 멀티 캐스트 송수신의 근본적은 취약점을 극복할 수 있는 아주 훌륭한 QoS 속성이라 하겠다.

4.6 메시지 재전송

메시지 재전송 기능은 메시지를 수신측에서 승인하지 않았을 경우 메시지를 수신하지 않았음을 확인하

고 재전송 하는 기능이다.

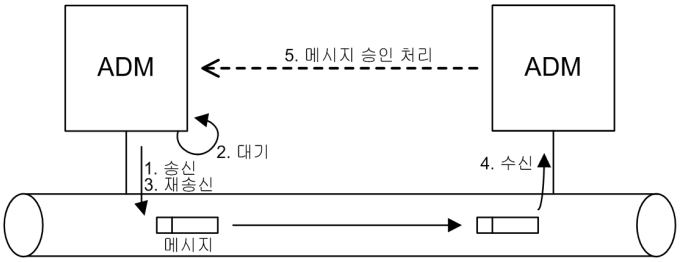

그림 8 메시지 재전송

그림 8와 같이 송신측 자율분산관리자는 메시지를 송신하게 되며 이때 메시지는 승인 처리 QoS 속성을 가지고 있는 메시지다. 송신측 자율분산관리자는 메시지의 송신한 이후에 메시지의 승인 처리가 될 때 까지 일정한 시간동안 대기하게 되는데 일정한 시간동안 승인 처리가 되지 않으면 재전송을 시도하게 된다. 수신측은 네트워크의 문제 또는 여러 가지 문제로 인하여 최초 보낸 메시지에 대해서 수신을 하지 못하여 승인 처리를 하지 못했지만 이후에 재전송된 메시지를 수신하였으므로 승인 처리를 하게 된다. 송신측 자율분산관리자는 메시지의 수신을 확인하고 메시지 송신을 종료하게 된다.

4.7 메시지 생명주기

메시지의 생명주기는 네트워크 지연이나 자율분산관리자의 처리 지연으로 인하여 발생할 수 있는 메시지의 유효성을 극복하기 위한 Qos 속성이다.

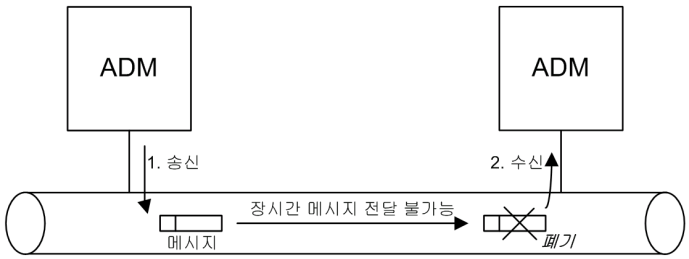

그림 9 기한이 만료된 메시지의 폐기처리

송신측 자율분산관리자가 메시지를 송신하게 되는데 네트워크 지연이나 기타 여러 가지 이유로 메시지가 발생한 시간이 과도하게 오래되었거나 하는 경우 수신측에 오래된 메시지를 수신했을 때 메시지가 과연 유효한가를 판단할 필요가 있다. 예를 들어 신호기를 제어할 때 제어 메시지가 과도하게 오래된 경우 이 메시지에 따라 신호기를 제어하게 되면 치명적인 문제가 발생할 수 있다. 이런 이유로 자율분산관리자는 메시지의 기한만료에 대한 정책에 대해서 명확하게 제시할 필요가 있다.

그림 9에서 보면 장시간 메시지의 전달이 지연되어 수신측 자율분산관리자가 수신하였을 경우 일정한 시간을 초과한 메시지에 대해서 메시지를 폐기 처분하게 된다. 이러한 품질 속성은 치명적인 제어 메시지의 경우 매우 중요한 속성이다.

4.8 트랜잭션 처리

응용 소프트웨어 모듈간의 메시지를 처리하다가 보면 처리의 단위가 하나 이상의 메시지로 구성될 수 있다. 이런 메시지의 경우 하나의 단일 트랜잭션으로 처리되어야 한다.

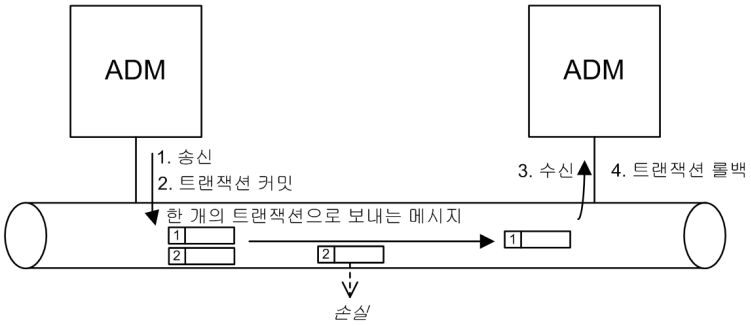

그림 10 메시지의 트랜잭션 처리

그림 10과 같이 총 2개의 메시지를 한꺼번에 송신하고자 할 때 수신측에서는 반드시 모든 메시지를 수신해야 한다. 그렇지 않은 경우 모든 메시지의 송신은 처리가 되지 않은 것으로 간주하게 된다. 그림 x와 같이 2번째 메시지를 손실하고 메시지를 수신했을 경우 수신측은 트랜잭션을 롤백처리하게 된다. 트랜잭션이 롤백되면 자율분산관리자는 수신한 1번 메시지에 대해서 수신을 처리하지 하지 않고 버리게 된다.

```
ControlMessage msg1 = new ControlMessage();
ControlMessage msg2 = new ControlMessage();
ControlMessage msg3 = new ControlMessage();
Session session = ADM.getSession(true); // 트랜잭션으로 처리하는 세션 획득
session.send(msg1);
session.send(msg2);
session.send(msg3);
session.commit(); // msg1, msg2, msg3가 한번에 트랜잭션 처리가 된다.
```

4.9 메시지 지속성

메시지 지속성은 자율분산관리자의 장애 상황이 발생했을 때 대처할 수 있는 중요한 수단이 된다. 예를 들어 메시지를 송신하던 중에 현재 자율분산관리자가 구동중인 시스템에 장애가 발생하여 자율분산관리자를 갑자기 중단했을 경우 송신중이던 모든 메시지는 손실하게 된다. 이런 문제를 극복하기 위해서 자율분산관리자는 다양한 지속성 정책을 제공할 필요가 있다.

도표 3 송신자의 유형

송신자의 유형	특 징
지속 불가능한 송신자	• 자율분산관리자의 장애 발생시 메시지가 손실된다. • 자율분산관리자가 재기동하더라도 송신하던 메시지를 모두 손실한다. • 지속 가능한 송신자 보다 매우 빠르게 메시지를 처리한다.
지속 가능한 송신자	• 자율분산관리자가 장애가 발생하더라도 메시지 손실을 방지할 수 있다. • 자율분산관리자가 재기동 하면 송신하던 메시지를 다시 보낼 수 있다. • 상대적으로 성능이 느리다. • 별도의 persistence storage가 필요하다.

도포 3에서 보는 것 처럼 송신자는 두 가지 유형으로 구분할 수 있다. 일반적인 경우는 지속성이 없는 송신자를 이용하여 메시지를 송신하지만 지속성이 필요로 할 때 지속성이 가능한 송신자를 사용할 수도 있다. 지속성의 지원에서 가장 중요한 것은 장애상황이 발생하여 자율분산관리자가 다시 기동하더라도 송신하던 메시지의 재전송 여부다. 지속성을 지원하면 송신하던 모든 메시지를 1차적으로 persistence

storage(예; DBMS, file system)에 저장하고 난 후 송신하기 때문에 자율분산관리자에 장애가 발생하더라도 송신하려는 메시지의 손실을 방지할 수 있다. 다만 지속성을 지원하는 송신자의 경우 지원하지 않는 송신자에 상대적으로 성능이 저하되는 문제가 발생할 수 있다. 따라서 이러한 QoS 속성은 충분히 대상을 선별한 이후에 사용해야 할 것이다.

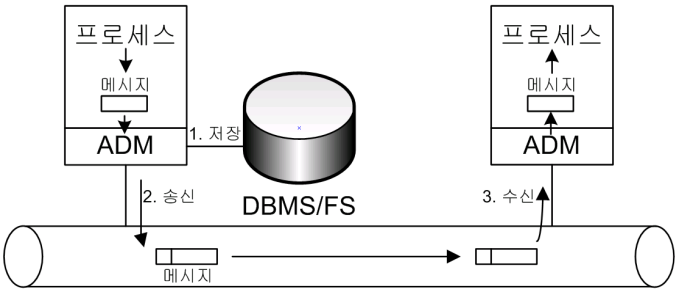

그림 11 지속성을 지원하는 송신자

지속성이란 메시지의 지속적 유지의 가능 여부를 설정하여 통신 메커니즘의 신뢰성을 크게 향상시킬 수 있는 QoS 정책이라고 할 수 있다. 다만 지속 가능한 송신자의 경우 송신자가 메시지를 무한정 유지할 수 없기 때문에 기한만료 정책을 따로 지정할 필요가 있겠다.

5. 결론

지금까지 신뢰성 있고 안정성을 향상시킬 수 있는 자율분산관리자의 다양한 품질 속성(QoS)에 대해서 알아보았다. 철도 신호 제어시스템에서 무엇보다 확보되어야 할 것은 바로 네트워크의 신뢰성 및 안정성이라고 하겠다. 물리적인 부분을 제외하고 소프트웨어적인 부분만 보더라도 자율분산관리자의 피어-투-피어, 브로드 캐스트/멀티 캐스트 송수신 모델에 대한 신뢰성 확보는 상당히 중요한 문제라 할 수 있다. 이러한 신뢰성 확보를 위해서 품질 속성을 적절하게 선택하여 적용시키는 것 또한 매우 중요한 문제라고 할 수 있다. 그러나 이러한 품질 속성이 안정성과 신뢰성을 향상시킨다 하여 무조건 다 적용하는 것은 심각한 성능 문제를 야기시키므로 적절하고 효과적인 선택이 필요하다. 자율분산 역 제어시스템의 자율성 및 능동성을 최대한 확보하기 위해서 이러한 다양한 품질 속성에 대한 이해와 충분한 시스템 테스트가 수반되어야만 원하는 목표를 달성할 수 있을 것이다. 현재 자율분산관리자 또한 이러한 품질 속성의 상당수가 구현되어 있고 설계되어 있지만 보다 높은 수준의 신뢰성 및 안정성을 확보하기 위해서 본 논문에서 설명한 품질 속성의 추가적인 설계와 구현이 좀더 필요하다. 본 논문에서 설명한 다양한 품질 속성이 제대로 구현되면 진정한 자율분산 역 제어시스템의 구현에 한발짝 다가설 수 있을 것이다.

(참고문헌)

[1] 이영수, 안진, 김은희, “자율분산시스템 단계적 구축방식의 철도역 적용에 관한 연구”, 2005.5, 춘계철도학회
[2] Open Management Group, "Data Distribution Service", 2005
[3] Sun Microsystems, "Java Message Service", 2002
[4] Open Management Group, "CORBA Event Service", 2004
[5] Open Management Group, "CORBA Notification Service", 2004