

# EPC Network Architecture에 바탕을 둔 RFID 미들웨어

## 시스템의 가용성 관리 방안

하성호<sup>o</sup> 박진욱 채흥석

부산대학교 컴퓨터공학과

(shha<sup>o</sup>, jwpark, hschae)<sup>o</sup>@pusan.ac.kr

### Availability Management Methods of RFID Middleware System Based on EPC Network Architecture

Sungho Ha<sup>o</sup> Jinwook Park, Heung Seok Chae

Department of Computer Science an Engineering, Pusan National University

#### 요 약

EPC Network Architecture는 일반적으로 RFID 미들웨어 시스템의 표준안으로 응용되고 있다. RFID 미들웨어 시스템은 그 사용에 있어 서브시스템들의 장애 발생이 예상되지만 이를 방지하고 복구하는 가용성(Availability)에 대한 지원은 부족하다. 따라서 본 논문은 EPC Network Architecture를 응용한 미들웨어 시스템을 개발함에 있어 가용성을 보장하는 방안을 제시한다. 그리고 가용성의 특성 가운데 장애 복구 기법에 더욱 초점을 두고 일반적인 수준의 가용성 또는 높은 수준의 가용성을 제공하는 다양한 기법들을 제시한다.

#### 1. 서 론

RFID 미들웨어 시스템은 물품에 부착된 EPC 태그 정보를 읽는 리더(Reader)와 비즈니스 정보시스템 사이에서 실시간으로 태그 정보의 필터링(filtering)과 비즈니스 이벤트 처리 등의 중요한 역할을 수행하는 시스템이다. 이에 대해 EPCglobal에서 RFID 시스템에 대한 표준안으로 EPC Network Architecture[1]를 제시하고 있다. EPC Network Architecture는 2002년 Savant Version 0.1을 시초로 2003년 Savant Version 1.0[2]로 발전하였고 2004년엔 ALE(Application Level Event)[3]를 포함하는 동인터페이스 중심의 모델로 발전해 나가고 있다. 본 논문에서는 현재 EPC Network Architecture를 바탕으로 RFID 미들웨어의 가용성 관리를 위한 방안을 소개한다.

가용성[4]이란 사용자로부터 요구되는 서비스를 시스템의 중단 없이 지속적으로 제공하는 특성을 말한다. RFID 미들웨어 시스템은 실시간으로 EPC 태그 정보를 읽고 이벤트에 대한 동작을 수행함으로써 일반적인 시스템보다 높은 가용성이 요구된다. 그러나 EPC Network Architecture는 가용성에 대한 고려가 고 가용성(High Availability)을 처리하기엔 부족하다. 또한, 시스템 구성에 따라서 고 가용성을 요구하는 시스템이 있고 일반적인 수준의 가용성만으로 충분한 시스템이 있을 수 있다.

가용성 향상을 위한 표준안으로 Service Availability Forum(SA)의 Application Interface Specification(AIS)[5]가 있다. AIS는 소프트웨어의 고가용성을 성취하기 위한 표준적인 API를 제공을 목표로 한다. 이 API를 사용하는 모든 응용 소프트웨어는 거의 동등한 수준의 가용성을 보장 받게 될 수 있다. 하지만 AIS는 범용적인 통신 시스템에 초점을 두고 있으므로 비즈니스 시스템으로 RF를 이용하는 RFID 시스템의 특수성은 고려하지 않았다. 따라서 본 논문에서는 EPC Network Architecture를 바탕으로 개발되는 RFID 미들웨어 시스템의 가용성을 보장하기 위한 다양한 시스템 구성 모델을 제시한다.

본 논문은 다음과 같이 구성된다. 2절에서는 가용성에 대한 정의와 장애 복구 방법을 설명한다. 3절에서는 EPC Network Architecture의 가용성을 높이기 위한 방안을 제시한다. 마지막으로 4절에서는 결론을 맺는다.

#### 2. 연구 배경

이 논문은 교육인적자원부 지방연구중심대학육성사업(차세대물류IT기술연구사업단)의 지원에 의하여 연구되었음.

#### 2.1 가용성

가용성이란 사용자로부터 요구되는 서비스를 사용자가 원하는 시점에 서비스가 제공될 확률을 뜻한다. 가용성의 중요한 두 가지 속성엔 MTBF(Mean Time Between Failure)와 MTTR(Mean Time To Repair)이 있다. MTBF는 시스템에서 발생하는 장애와 그 다음 장애 간의 평균 시간을 뜻한다. 그리고 MTTR는 장애가 발생한 시스템을 복구하는 데 소요되는 평균 시간을 뜻한다. 가용성은 MTBF에 비례한다. 장애 발생간의 평균 시간이 클수록 시스템이 지속적으로 동작을 하게 된다. 또한, 가용성은 MTTR에 반비례한다. 복구시간이 빠르면 그만큼 가용성은 높아진다. 따라서 MTTR의 값이 클수록 시스템의 가용성은 낮아진다.

가용성을 향상시킬 수 있는 방법은 두 가지다. 첫 번째는 MTBF를 극대화시키는 것이다. MTBF가 매우 크다는 것은 시스템이 매우 많은 시간동안 동작을 하다가 장애가 발생하는 것을 뜻한다. 따라서 사용자 입장에서는 높은 수준의 가용성을 달성하게 된다. 두 번째는 MTTR를 최소화함으로써 높은 수준의 가용성을 달성할 수 있다. 장애가 발생하면 즉각적인 복구를 통해서 높은 수준의 가용성을 달성하는 것이다. 장애를 복구하기 위해서는 장애 발생 여부를 파악하고 파악된 장애에 대한 복구를 수행한다. 이를 장애 탐지(Fault Detection)와 장애 복구(Fault Recovery)[6]라 부른다.

MTBF의 최소화는 신뢰도 또는 결함 포용(Fault tolerance)[7]에 관련된 연구에 초점을 두며 일반적으로 가용성이라고 할 때는 MTTR에 초점을 둔다. 그러므로 본 논문에서도 미들웨어의 장애 발생에 대한 예방, 포함, 제거 기법 보다는 발생된 장애에 대한 즉각적인 복구를 통하여 높은 가용성을 성취하는 방법을 제시한다.

#### 2.2 장애 복구

가용성을 보장하는 방법은 시스템의 장애를 찾아내는 장애 탐지와 파악된 장애로부터 시스템을 복구하는 장애복구가 있다. 본 논문은 장애 탐지 보다는 MTTR을 최소화하기 위한 장애 복구를 중심으로 기술하였다. 장애 복구는 시스템의 동작 중에 적용되어야 한다. 그러므로 시스템이 정지된 상태에서 원인을 파악하고 장애를 복구하는 방법으로는 가용성을 높일 수 가 없다. 따라서 일반적으로 장애 발생이 예상되는 컴포넌트를 중복(Redundancy)하여 신속한 복구를 시도한다. 즉, 현재 실행되고 있는 주 컴포넌트(primary component, Active component)의 기능을 대신할 보조 컴포넌트(standby component, backup component)를 둔다. 그리고 장애 발생 시에 보조 컴포넌트가 주 컴포넌트를 대신해서 동작하게 함으로써 서비스는 연속으로 제공하고 주 컴포넌트의 장애복구를 시도한다. 이때 주 컴포넌트의 장애로 보조 컴포넌트가 주 컴포넌트의 역할을 대신 수행하기까지의 과정을 Fail over라 하며 주 컴포넌트의 장애가 복구되어

보조 컴포넌트에서 원래의 주 컴포넌트로 작업을 수행하게 하는 과정을 Fail back이라 한다.

(1) 중복기법

- 수동적 중복기법

수동적 중복 기법에서는 정상시에 주 컴포넌트만 동작을 하다가 주 컴포넌트에 장애가 발생한 경우에 보조 컴포넌트가 주 컴포넌트의 동작을 대신하게 된다. 보조 컴포넌트가 주 컴포넌트의 동작을 대신하기 위해서는 주 컴포넌트와 동일한 상태로 동기화가 되어야 한다.

▪ 2N 중복 기법

서비스를 수행하는 주 컴포넌트와 주 컴포넌트에 장애가 발생하였을 경우 즉시 대처할 수 있는 보조 컴포넌트로 구성되어 있다. 구현이 단순하며 차후에 N+M, N-way 중복 기법으로 확장이 용이하다.

주 컴포넌트가 완전히 복구되어 Fail over 되기 전에 보조 컴포넌트까지 장애가 발생하였을 경우는 대처할 수 있는 방법이 없다. 따라서 빈번한 장애가 예상되는 컴포넌트에 적합하지 않다. 반면 장애발생 빈도가 극히 낮은 컴포넌트에 2N의 방법을 적용하면 항상 보조 컴포넌트가 필요하므로 자원의 낭비가 예상된다.

▪ N+M 중복 기법

N개수만큼의 주 컴포넌트에 M개수만큼의 보조 컴포넌트를 구성하는 방법이다. 각 컴포넌트 마다 중복을 두는 2N기법을 사용함으로써 발생하는 보조 컴포넌트의 낭비를 줄일 수 있다.

▪ N-way 중복 기법

- 능동적 중복기법

능동적 중복 기법은 보조 컴포넌트가 주 컴포넌트와 동일하게 동작을 한다. 즉, 동일한 입력을 처리한다. 만약 주 컴포넌트에 장애가 발생하면 보조 컴포넌트가 주 컴포넌트 역할을 대신한다. 보조 컴포넌트가 주 컴포넌트와 지속적으로 동기화가 되고 있기 때문에 빠른 작업 전이가 가능하다.

▪ N-way Active 중복 기법

N-way 중복 기법과 유사하나 가장 큰 차이점은 standby가 없다는 것이다. 짧은 순간의 장애도 인정하지 않는 고 가용성이 요구되는 시스템에 적합한 모델이다. 능동적 중복기법은 보조 컴포넌트가 주 컴포넌트와 동일하게 동작한다. 즉, 동일한 입력을 처리한다. 만약 주 서버에 장애가 발생하면 보조 서버가 주 서버의 역할 즉각적인 복구가 이루어지긴 하나 네트워크의 부하 발생 비율이 높다.

(2) 비 중복기법

비 중복기법(No Redundancy)은 중복을 가지지 않는 모델이다. 장애 발생 컴포넌트가 시스템 전반에 걸쳐 심각한 영향을 끼치지 않고 고 가용성을 요구하지 않을 때, 그리고 복구에 시간적 여유가 있을 경우 사용한다. 중복을 가지지 않는 기법임으로 보조 컴포넌트에 대한 비용의 손실이 적다. 하지만 복구되기 까지 시간의 소비가 많으므로 즉각적인 복구가 필요하지 않은 경우에 사용할 수 있다.

2.3 EPC Network Architecture

그림1은 EPCglobal이 정한 최근의 EPC Network Architecture의 모습이다.

-RFID Reader

리드 존(read zone) 안에서 RFID Tag정보를 읽어오는 역할을 수행한다.

-Reader Management

리더를 관리하고 통제하는 역할을 수행한다.

-Filtering & Collection

리더에서 읽어온 Raw 태그 정보를 필터링 및 컬렉션 하는 역할을 한다.

-EPCIS Capturing Application

Event에 해당하는 EPC정보를 제공하는 역할을 한다.

-EPCIS Repository

EPCIS 정보를 저장하는 저장소이다.

-EPCIS Accessing Application

EPCIS의 정보를 직접 액세스하여 이용하는 애플리케이션이다.

본 논문에서는 RFID 미들웨어의 가용성을 보장하는 대상으로 리더와 미들웨어로 한정한다. 이는 RFID 미들웨어 시스템의 특성상 태그 정보를 직접적으로 읽는 리더는 외부환경에 노출되어 있을 수 있고 미들웨어도 정확한 필터링과 컬렉션을 위해 노출될 수 있다. 따라서 RFID 미들웨어 시스템을 구성하는 리더와 미들웨어는 장애가 발생할 확률이 높다.

3. 가용성 관리 방안

가용성 관리를 위해서 EPC Network Architecture에 가용성관리자(Availability Manager)를 둔다. 그림2의 Monitored Component는 가용성 관리자가 가용성을 제공하기 위해 관심을 두고 있는 컴포넌트를 뜻한다.

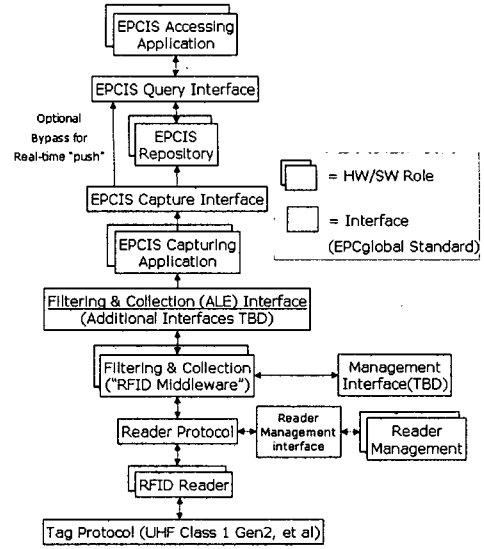


그림 1 EPC Network Architecture

즉, RFID Reader, Filtering&Collection(RFID Middleware) 그리고 EPCIS Capturing Application 등이다. 본 논문은 각 컴포넌트에 대해 가용성 관리자가 장애탐지 및 복구 시도를 하는 주체로 가정한다.

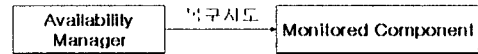


그림 2 가용성 관리자를 통한 복구시도

3.1 수동적 중복 기법

▪ 2N 중복 기법

그림3은 Reader에 대한 가용성을 2N 중복 기법의 모습으로 보여주고 있다. RFID Network의 특성상 태그 정보는 라디오 주파수를 통해 전달됨으로 리더 존안에 있는 리더는 모든 태그 정보를 읽게 된다. 따라서 보조

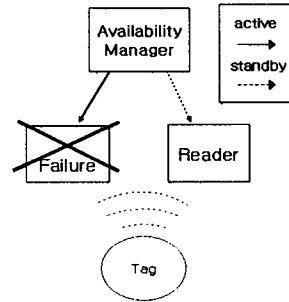


그림 3 2N중복기법(Reader)

컴포넌트 없이 모두 주 컴포넌트가 되므로 리더의 가용성 제공은 2N기법으로는 부적합하다. 리더의 가용성은 이후 설명할 N-way Active중복 기법으로 구현이 가능하다.

그림4는 미들웨어에 대한 가용성을 2N 중복 기법의 모습으로 보여준다. 필터링과 컬렉션을 담당하는 주 RFID 미들웨어 하나당 하나의 보조 미들웨어를 둬으로써 주 미들웨어의 장애 발생 시 보조 미들웨어로 Fail over 된다. N개의 미들웨어가 필요하다면 같은 N개수의 보조 미들웨어가 필요함으로 가용자원에 대한 낭비가 심한 단점이 있다. 하지만 가용자원이 충분한 환경에서는 높은 가용성을 보장할 수 있다. 장애가 복구된 이후엔 원래의 역할로 돌아오는 Fail back을 수행한다.

▪ N+M 중복 기법

그림5는 N+M기법은 응용한 3+1 중복 기법이다. N+M 중복 기법을

사용할 경우 N개의 주 미들웨어와 M개의 보조 미들웨어를 운영한다.

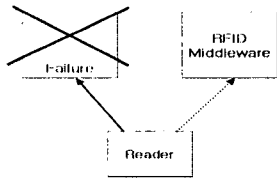


그림 4 2N중복기법(Middleware)

하나의 미들웨어에 단순히 하나의 미들웨어만 두는 2N방식에 비해 주 미들웨어의 수(N)와 무관하게 보조 상태의 미들웨어의 수(M)를 결정할 수 있다. 이와 같은 구성의 단점은 여러 미들웨어가 장애를 일으킬 경우 특정 보조 미들웨어에 부하가 집중될 수 있다는 문제점이 있다. 그리고 2N 구성의 경우와 마찬가지로 보조 미들웨어는 장애 발생 이전까지 대기상태 이므로 시스템 낭비를 초래한다. 하지만 장점은 2N의 구성에 비해 보조 미들웨어의 수를 줄여 시스템 가용 자원의 낭비를 줄일 수 있다.

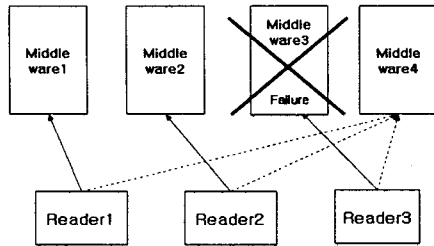


그림 5 3+1 중복기법을 통한 미들웨어의 가용성

▪ N-way 중복 기법

미들웨어의 장애 복구를 위해 N-way 중복 기법을 적용할 경우 리더는 하나의 주 미들웨어를 가지고 다수개의 보조 미들웨어를 가질 수 있다. 2N, N+M 중복 기법과 다른 점은 각 미들웨어들이 주 컴포넌트로서의 역할과 보조 컴포넌트로서의 역할을 동시에 수행한다는 것이다. 여분의 미들웨어를 거의 확보하지 못한 환경에서 능동적 중복 방법을 사용하기에는 미들웨어가 처리해야 할 이벤트의 양이 너무 많을 때 적합하다. 그러나 2N, N+M 가용성에 대한 안정성이 떨어지며, 둘 이상의 미들웨어가 동시에 장애가 발생하고 보조로 지정된 미들웨어가 부족할 경우 복구 전략이 차질이 생길 가능성이 있다. 그림6은 미들웨어의 가용성을 2-way 중복기법을 통해 구현하고 있다.

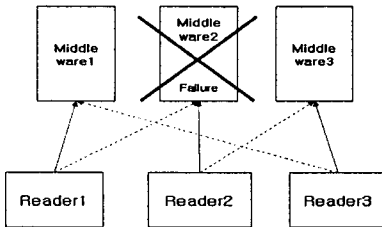


그림 6 2-way 중복기법을 통한 미들웨어의 가용성

3.2 능동적 중복 기법

▪ N-way Active 중복 기법

미들웨어의 장애 복구를 위해 N-way Active 중복 기법을 적용하였을 경우 모든 미들웨어가 주 컴포넌트 상태에 놓이게 된다. 특정 미들웨어에 장애가 발생하더라도 리더는 여러 주 미들웨어를 가지고 있으므로 장애 발생에 대한 오버헤드가 없으며 다운타임(downtime)을 거의 느끼지 못한다. 이러한 작업의 효율성이 높다는 장점에도 불구하고 미들웨어들은 다른 복구 전략에 비해 리더들 간의 이벤트를 처리해야 함으로 이에 대한 오버헤드가 있다. 그림7은 2-way active기법을 적용하여 미들웨어의 가용성을 보장하는 방식을 보여주고 있다. 미들웨어2에 장애가 발생하면 미들

웨어2가 처리하던 리더1과 리더2를 미들웨어1과 미들웨어3이 처리하게 된다.

그림8은 리더 존안에 있는 리더기의 가용성을 보장하기 위해 N-way Active 중복기법을 사용한 모습을 보여준다. Tag정보는 라디오 주파수 이므로 리더 존안에 있는 모든 태그는 리더기로 읽혀오게 된다. 따라서 리더기의 가용성을 보장하기 위해서는 리더 존의 범위를 예측하고 리더 존안에 리더를 중복하여 두는 방법으로 가용성을 보장하고 있다. 그림8에서 리더2에 장애가 발생하면 3-way Active 방식으로 리더1과 리더3이 태그 정보를 처리하게 된다.

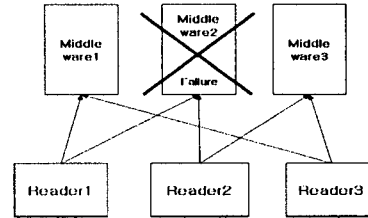


그림 7 2-way Active 기법을 통한 미들웨어의 가용성

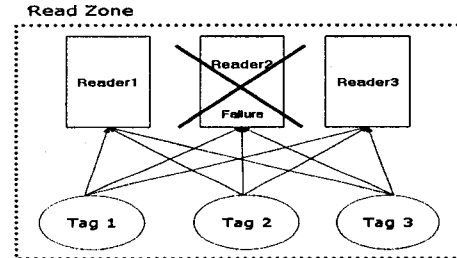


그림 8 3-way Active중복 기법을 통한 리더의 가용성

4. 결론 및 향후 연구 방향

본 논문에서는 EPC Network Architecture에 바탕을 둔 RFID 미들웨어 시스템의 가용성 관리방안에 대해서 연구하였다. 시스템 구성에 따라 높은 가용성을 보장하는 기법도 있으며 가용성은 낮지만 비용이 적게 드는 기법도 존재하였다. 따라서 RFID 미들웨어 시스템을 구현하려 할 때 비용과 복구 대상 컴포넌트의 중요도를 고려하여 시스템을 구성할 수 있을 것이다.

향후 연구에서는 다양한 중복기법에 따른 가용성 보장을 위한 효율적인 설계방안에 대해 연구를 할 계획이다.

5. 참고문헌

- [1] Ken Traub, Greg Allai and Henri Barthel The EPCglobal Architecture Framework, EPCglobal, 2004
- [2] Sean Clark, Ken Traub, Dipan Anarkat and Ted Osinski Auto-ID Savant Specification 1.0, Auto-ID Center, 2003
- [3] Ken Traub, Sylvanus Bent, Ted Osinski, Samuel Niles Peretz, Steve Rehling, Steve Rehling and Bryan, The Application Level Events Specification, EPCglobal, 2005
- [4] Evan Marcus & Hal Stern, Blueprints for High Availability, Wiley, 2003
- [5] Service Availability Forum Application Interface Specification, <http://www.saforum.org>
- [6] Len Bass, Paul Clements and Rick Kazman, Software Architecture in Practice Second Edition, 2003
- [7] Jalote and Pankaj, Fault tolerance in distributed systems, Englewood Cliffs, 1994I