

멀티 에이전트 시스템에서 메시지 전송 성능 향상을 위한 개선된 RIP 적용 방안

김효원*, 윤희용*

*성균관대학교 정보통신공학부

e-mail : khwsuper@yahoo.co.kr, youn@ecc.skku.ac.kr

A method of Improved RIP for Performance Improvement of Message Transport in Multi-Agent System

Hyo-Won Kim*, Hee-Yong Youn*

*School of Information and Communication Engineering, SungKyunKwan University

요 약

최근 유비쿼터스 시대에 고품질의 빠른 서비스를 제공하기 위한 에이전트 시스템의 연구가 활발히 진행 중이다. 멀티 에이전트 시스템은 플랫폼 위의 에이전트들이 서로 메시지를 전송하며 자율적이고 지능적으로 일을 처리하는 시스템이다. 에이전트간 메시지 전송을 최적화 하기 위해서는 전체 에이전트와, 네트워크 토폴로지에 대한 정보 공유가 필요하다. RIP는 전체 네트워크 상태를 공유하는 라우팅 프로토콜로서 에이전트 메시지 전송 성능 개선에 적합한 기법이다. 하지만 RIP 라우팅 프로토콜은 여러 취약점이 있다. 에이전트 메시지 전송 성능 개선을 위한 에이전트의 운용은 위 같은 문제를 해결하고 효과적인 메시지 전송을 가능하게 할 것이다

1. 서론

최근 유비쿼터스 시대를 맞아 분산환경에서 고품질의 빠른 서비스를 제공하기 위한 많은 연구가 이루어지고 있다. 그 연구의 중심에 멀티 에이전트 시스템이 있다. 에이전트는 지식과 추론 능력, 학습, 계획 능력 등을 가지고 사용자를 대신해서 사용자가 원하는 작업을 자동적으로 할 수 있다. 이러한 특성은 여러 분야와 결합하여 지능적이고 자율적인 프로세스를 가능하게 한다. 특히 네트워크 기술과의 결합은 기존의 기술의 한계를 뛰어넘어 새로운 에이전트 패러다임을 제시할 수 있다.

멀티 에이전트 시스템은 플랫폼 위의 에이전트들이 서로 메시지를 전송하며 자율적이고 지능적으로 일을 처리하는 시스템이다. 따라서 메시지 전송은 시스템의 성능을 결정하는 중요한 요소이다. 전체적인 에이전트와 플랫폼의 상태 공유는 메시지 전송에 최적의 에이전트를 검색할 수 있게 한다. RIP(Routing Information Protocol)는 라우팅 프로토콜로서 전체 네트워크의 상태 공유에 우수한 성능을 갖고 있는 프로토콜이다.[1-3]

하지만 기술이 고도화 될수록 각각의 네트워크들의 상태는 실시간으로 다양하게 바뀐다. 이러한 환경에서 멀티 에이전트 시스템의 메시지 전송 개선을 위한 알고리즘으로 일반적인 RIP를 채택하게 되면 몇 가지 취약점이 발생하게 된다. 예를 들면 빈번한 에이전트플랫폼 삭제 삽입 와 같이 물리적으로 네트워크 상태가 바뀌기도 하고, 특정 링크의 장애가 발생하거

나, 특정 플랫폼에 트래픽이 많이 집중되어 혼잡상황을 유발하는 등의 내부의 유동적 상황이 발생할 수 있다. 게다가 메시지를 전송할 때 네트워크 루프 같은 심각한 문제로 메시지가 목표 에이전트에 전송되지 않을 수도 있다.

본 논문에서는 멀티 에이전트 시스템의 메시지 전송 성능 향상을 위해 개선된 RIP 알고리즘 적용하는 기법을 제안한다. 에이전트와 플랫폼에 대한 전체적인 상태 공유는 메시지 전송에 가장 적합한 에이전트 선정에 중요한 정보이다. 에이전트 플랫폼 위에서 메시지 전송 성능 향상을 위한 작업을 수행하는 에이전트들을 구성하여 개선된 RIP 알고리즘을 적용하게 되면, 시스템은 메시지 교환을 위해 RIP 라우팅 테이블 형식의 플랫폼과 에이전트의 상태를 기록하는 테이블을 구성하고, 자동적으로 네트워크 상태의 변화에 의해서 업데이트 되게 할 수 있다. 이를 기반으로 전체 에이전트와 플랫폼에 대한 테이블의 정보들은 에이전트들이 가장 적합한 환경의 에이전트로 메시지를 전송할 수 있게 할 것이다. 이를 통해 개선된 메시지 전송 효과를 얻을 수 있다.

본 논문의 구조는 다음과 같다. 2 장에서는 RIP 프로토콜의 동작 원리를 소개하고, 멀티 에이전트 시스템에 대해 기술한다. 3 장에서는 메시지 전송 성능 향상을 위해 RIP의 원리를 채택함으로써 나타나는 취약점과 이를 해결하고 메시지 전송 성능을 높이는 방안 에 대해 다루고 있으며, 끝으로 4 장은 논문의 결론을 담고 있다.

2 관련연구

2.1 RIP

기본적으로 모든 라우팅은 특정 정책에 따른 네트워크 비용을 기준으로 최소비용이 드는 경로를 최적의 경로로 설정한다. RIP 는 distance vector 라는 라우터 사이의 거쳐야 할 네트워크 개수를 기준으로 라우팅 테이블을 구성해나간다. 그리고 네트워크에 변동이 생기면 30 초를 간격으로 갱신된 네트워크 토폴로지를 포함한 모든 라우팅 정보를 인접 라우터에 전송한다[2]. RIP 를 구성하는 라우팅 테이블의 기본형식은 다음과 같다.

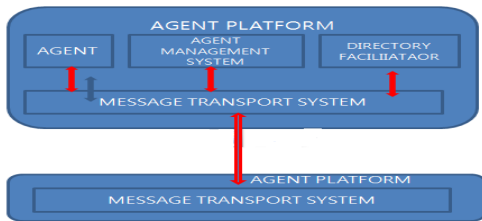
<표1> 라우팅 테이블

DESTINATION	HOP COUNT	NEXT ROUTER	Other information
163.5.5.0	7	200.3.4.0	
213.54.2.0	5	111.2.3.0	
122.34.0.0	2	110.0.0.2	

패킷은 목적지까지의 홑수를 기준으로 작은 홑수를 최적의 경로 판단하고 다음 라우터로 이동하기 위해 포트를 결정한다.

2.2 멀티 에이전트 시스템

멀티 에이전트 시스템은 자율적이고 지능적인 에이전트들의 협업에 의해 효과적으로 동작되는 시스템을 말한다.

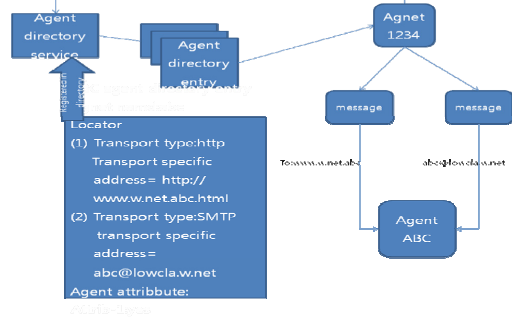


(그림 1) 에이전트 시스템

그림 1과 같이 기본적으로 에이전트는 하나의 에이전트 플랫폼 위에 하나 혹은 그 이상이 생성된다. 에이전트를 위해 이용되는 모든 서비스는 Agent Management System에 의해 통제, 제어되며, 일종의 에이전트인 DF(Directory Facilitator)는 옐로우 페이지 기능을 통해 다른 에이전트 플랫폼과, 에이전트를 등록하고 있어서 협업할 에이전트를 빠르게 찾을 수 있다. 에이전트간에는 메시지를 주고 받으며 작업을 수행하게 된다[1].

아래 그림 2 는 에이전트의 동작의 예를 단순화 한 것이다. 우선 Agent1234 가 Agent directory service 로 협업에 적합한 에이전트 검색을 의뢰하면, 에이전트 플랫폼에 등록된 에이전트 중에서 가장 적합한 에이전트를 Agent directory entry 에서 찾아서 Agent ABC 를 반환한다. 그러면 Agent1234 는 HTTP 형식 혹은 SMTP

형식으로 검색된 Agent ABC 와 통신을 하게 되며 메시지를 주고 받으며 작업을 수행한다[4].



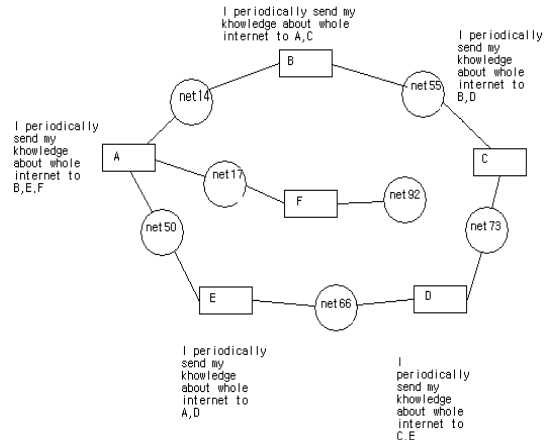
(그림 2) 에이전트의 동작

3. 개선된 RIP 기법을 적용한 메시지 전송

3.1 알고리즘 메시지 전송을 하는 멀티 에이전트 시스템의 취약성

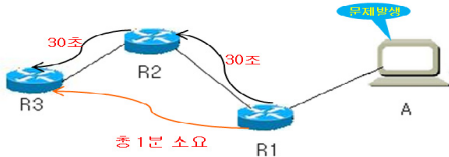
본 논문에서는 RIP 라우팅 프로토콜의 알고리즘을 적용하는 기법을 통해 전체 에이전트 플랫폼 간의 네트워크 상태를 테이블을 구성해 공유하고, 이를 기반으로 멀티 에이전트 시스템의 메시지 전송 성능을 개선을 제안한다. 하지만 일반적인 RIP방식은 몇 가지 취약성을 가지고 있어서 전체적인 시스템에 문제를 가져오게 된다.

1) 메시지 전송 성능 개선을 위한 테이블 구성 및 운영을 방법에 RIP를 적용하면 인접한 플랫폼의 에이전트에 자신의 플랫폼간 통신에 대한 모든 정보를 특정 주기(30초)마다 전달하는 방법, 즉 ripple carry 에 의한 라우터간 정보 공유 방식을 통해 라우팅 테이블을 주고 받으며 전체 에이전트 플랫폼 네트워크의 상태를 갱신한다. 이때 변경된 정보뿐만 아니라 모든 정보를 교환하는 불필요한 overhead가 발생한다. 라우팅 테이블 형성 프로세스는 그림 3 과 같다



(그림 3) 라우팅 테이블 생성

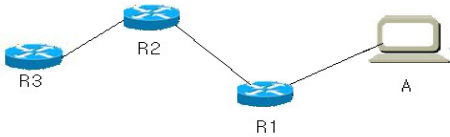
위와 같은 라우팅 테이블 구성 원리는 여러 개의 hop을 가지고 있는 네트워크간의 정보 공유는 정보 전달 주기의 hop수의 배만큼 시간이 소요된다는 것을 의미한다



(그림 4) Hop 소요시간

그림 4와 같이 만약 A 플랫폼에서 문제가 발생했다면 R1에서 문제를 파악하고 R2에 A와 통신 불가능한 테이블 갱신을 한 후, R2까지 30초, 또 다시 R3까지 30초의 시간이 소요된다. 즉 R3가 A플랫폼의 상태 변화를 갱신한 테이블이 도착하는데 1분이란 시간이 걸리는 문제가 발생한다.

2)Slow convergence time은 플랫폼 위의 에이전트 통신간 루프를 생성하게 된다. 루프의 생성원리는 다음과 같다.



(그림 5) 라우팅 루프

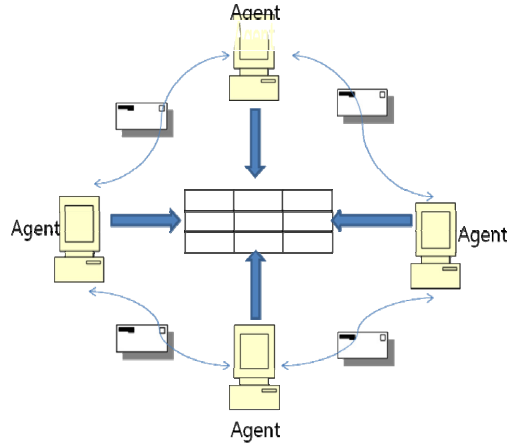
그림 5에서 A 에이전트 플랫폼이 다운되면 R1은 라우팅 테이블에 A 에이전트 플랫폼에 대한 possibly down을 기록하게 된다. 하지만 R2의 라우팅 테이블에는 A플랫폼이 여전히 살아 있는 것으로 되어 있고 R2는 R1으로 라우팅 업데이트를 하게 된다. R1은 R2로부터 라우팅 업데이트를 받고 A 플랫폼으로 가려면 R2를 통한 경로를 인식하여 라우팅 테이블에 다시 기록을 하게 된다. 이 과정에서 A네트워크로의 경로에 대한 HOP값이 계속해서 바뀌게 되는데 이러한 상으로 인해 에이전트 플랫폼간 루프가 형성되게 되는 것이다.

- 3) 인터넷 등의 다양한 네트워크들의 결합에서 distance vector라는 하나의 개념을 사용한 통신량 산정 방식은 다양한 policy를 요구하는 현재 네트워크의 특성을 잘 반영하지 못한다고 할 수 있다. 멀티 에이전트 시스템의 메시지 전송 최적화를 위해서 Cost는 물론 네트워크간 hop수가 될 수도 있지만, delay time, 트래픽 등의 다양한 기준에 의해 산정될 수 있어야 한다. 그리고 양방향 cost가 다른 경우도 빈번히 발생한다. 따라서 RIP의 distance vector를 메시지 전송 향상을 위한 전체 네트워크 상태 확보를 위한 알고리즘으로 사용하기에는 개선이 필요하다.

요하다.

3.2 개선된 RIP를 이용한 에이전트간 메시지 전송

아래 그림 6은 에이전트 메시지 전송에 개선된 RIP 알고리즘 기법을 도식화한 것이다. 우선 Agent 플랫폼들간에 연결을 한다. 그리고 메시지 전송 성능을 위해 RIP 기법을 적용해 테이블을 형성한다.



(그림 6) RIP 이용한 메시지 전송

메시지 전송 성능 향상을 전담한 에이전트들은 표 2와 같은 에이전트 라우팅 테이블을 주고 받는다.

<표 2>메시지 전송 성능 개선을 위한 테이블

Agent Platform ID	STATUS	HOP COUNT	NEXT PLATFORM	Others Info.
A	Congestion	1	C	
B	Good	2	D	
C	Good	5	A	
D	Delete	2	B	

이를 기반으로 전체 RIP를 적용한 메시지 전송 에이전트들의 협업에 의해 전체적인 에이전트 플랫폼 네트워크의 상태가 공유되게 된다. 각 에이전트들은 DF에 등록된 에이전트의 정보를 활용하여 빠른 검색으로 에이전트의 상태를 체크하고 에이전트 라우팅 테이블을 채워 나가며, RIP의 ripple carry 기법에 의해 테이블을 공유한다. 이를 통해 전체 멀티 에이전트 시스템의 에이전트 네트워크에 대한 전체적인 상태를 파악 가능하게 됨으로써 메시지 전송 시 가장 통신에 상태가 적합한 에이전트들에 협업 에이전트로서 우선 순위를 줄 수 있게 된다.

테이블을 구성하기 위한 4가지 전담 에이전트를 생성, 동작시키면 라우팅 루프뿐만 아니라 여러 가지 개선된 성능의 에이전트 메시지 전송을 할 수 있게 된다. 필요 에이전트들은 다음과 같다.

- ① 네트워크 토폴로지 변화를 인지하는 에이전트
- ② 최대 HOP을 체크하는 에이전트

③송신자를 체크하는 에이전트

④에이전트의 혼잡 정도를 체크하는 에이전트

4 개의 메시지 전송 성능 개선을 위한 에이전트들은 각 에이전트 플랫폼 위에 존재하게 된다. DF 에 에이전트 플랫폼마다 위치한 모든 에이전트들의 접속 여부 등록하고 있다. 따라서 위의 4 개의 에이전트가 DF 에 등록하면 에이전트의 호출 시 필요한 에이전트에 빠르게 접근할 수 있다. 1:1, 1:n 혹은 n:m 관계의 에이전트간의 작업이 발생하면 DF 에 의해 빠르게 검색된 관련 에이전트 플랫폼의 에이전트들과 메시지를 주고받고, 에이전트 라우팅 테이블에 결과를 반영시킨다.

②번 에이전트는 최대 HOP 지정을 통한 라우팅 루프 방지 기법을 구현시키며 15 이상의 HOP 을 요구하는 패킷 발생시 에이전트는 이를 인식하고 DF 를 통해 접속된 모든 에이전트 플랫폼의 에이전트들에 메시지를 전송하여 목적지 망은 끊어졌다고 인식을 하게 하고 각 에이전트들은 라우터에 통신거부 요청을 하여 더 이상 라우팅을 발생시키지 않고 루프를 방지한다.

에이전트의 메시지에는 송신자의 정보를 포함하고 있다. ③번 에이전트는 에이전트 메시지에 상대방의 정보를 체크하고 수신 에이전트가 비정상적인 반복적 같은 송신 에이전트로부터 수신을 할 경우 일정 수준에 이르면 이를 에이전트 라우팅 루프로서 인식하고 해당 에이전트로의 경로 설정을 중지 요청하고 다른 라우터의 ⑤에이전트를 통해 통신량이 적은 플랫폼의 에이전트를 검색하고 해당 플랫폼의 에이전트로 메시지를 보내 우회로의 역할을 하게 한다.

또한 필요 시에는 특정 라우터의 에이전트 플랫폼에 라우팅 테이블 갱신 시간을 유보시키도록 특정 에이전트에 메시지를 보내고 메시지를 받은 에이전트는 라우터에 유보요청을 하여 네트워크 상태 변화가 전체 네트워크에 전파될 수 있을 정도의 충분한 시간을 제공하여 slow convergence 에서 오는 라우팅 루프를 막을 수 있다[5].

위와 같은 전체 네트워크에 대한 정보를 공유하기 위한 메시지 전송개선 에이전트들의 라우팅 루프 방지 외에도 네트워크 토폴로지의 변화가 발생할 경우 이를 인지한 ①번 에이전트가 DF 에 등록된 모든 에이전트에 메시지를 Broadcasting 할 수 있기 때문에 기존 RIP 의 라우팅 테이블 갱신 방식인 ripple carry 방법보다 훨씬 빠르게 전체 네트워크의 변화를 에이전트 라우팅 테이블에 반영 할 수 있다. 그리고 에이전트 시스템은 S/W 기반으로 생성되기 때문에 쉽게 특정 정책에 맞춰 Cost 산정 기준을 유연성 있게 바꿀 수 있다. 뿐만 아니라, 트래픽이나 혼잡 발생시에도 해당 플랫폼의 에이전트가 혼잡 사실을 Broadcasting 하면 통신량이 곧고루 분산되도록 혼잡에서 정상적인 통신량 상태로 떨어질 때까지는 에이전트들이 경로요청을 해도 혼잡 플랫폼의 에이전트는 거절을 하고, 요청 에이전트는 그 인근의 통신량이 적은 에이전트에 경로를 할당하도록 할 수 있다. 결국 에이전트들의 협업을 통해 라우터는 가장 안정적인 네트워크를

구성, 운용함으로써 메시지 전송을 최적화 한다.

4. 결론

본 논문에서는 멀티에이전트 시스템의 메시지 전송 성능을 개선하기 위해 전체적인 플랫폼의 상태를 모든 에이전트들이 공유하여, 협업 대상 에이전트 선정시 통신량이 적은 에이전트에 우선순위를 두어 메시지를 전송함으로써 전체적 시스템의 성능 향상을 할 수 있는 기법을 제안하였다. 그리고 파생되는 취약성 알아보고, 메시지 전송 관리를 위한 전담 에이전트들의 협업을 이용한 문제 해결 방안을 다루었다.

DF 의 에이전트 참조에 대한 정보에 본 논문에서 제안한 메시지 전송 관리 에이전트들을 통한 전체 네트워크 상태 파악능력은 메시지 전송시 가장 적합한 에이전트 선택에 중요한 부가 정보가 되어 준다. 게다가 메시지를 서로 주고 받으며 협업하는 에이전트들의 장점은 distance vector 라는 단순한 네트워크 통신량 산정 기준을 넘어 다양한 네트워크의 특징을 반영, 보다 최적의 경로를 제공할 수 있다. Agent 의 Broadcasting 에 의해 네트워크 토폴로지 및 트래픽 상황에 기민하게 대처할 수 있는 장점도 있다.

이상적인 유비쿼터스 환경 구축을 위한 멀티 에이전트 시스템에 대한 더욱 깊고 다양한 연구가 필요하다[6]. 향후 연구 과제로 우리가 제안한 멀티에이전트 시스템 기반의 라우팅 지원 기법에 대한 보다 구체적인 모델의 정립과 분석이 필요하다.

참고문헌

- [1] FIPA (Foundation for Intelligent Physical Agents), web resource available at <http://www.fipa.org/>
- [2] 유기훈, "라우팅 프로토콜:라우팅의 기본적 개념", 방송과 기술 통권 142 호, October2007
- [3] M. Luck, P. McBurney and C. Priest, "Agent Technology: Enabling Next Generation Computing", (AgentLink community) (2003)
- [4] B. Horling and V. Lesser, "A Survey of Multi-Agent Organizational Paradigms", The Knowledge Engineering Review, Volume 19, Number 4, Cambridge University Press, 2005, pp. 281-316.
- [5] 김태현, "효율적인 네트워킹을 위한 라우팅 루프 해결 기법"February2007
- [6] 최정화 외, "유비쿼터스 환경의 사용자 서비스를 위한 지능형 에이전트 기술", 한국정보 과학회지, September2007