

무선 환경에서 부하 균등화를 고려한 모바일 노드 관리 알고리즘

*최영호, 최재혁, 신광식, 최상방
인하대학교 전자공학과
e-mail : elitecyh@naver.com

Mobile Node Management Algorithm considering Load Balancing in Wireless Environment

*Young-Ho Choi, Jae-Hyeok Choi, Kwang-Sik Shin, Sang-Bang Choi
Dept. of Electronic Engineering
Inha University

Abstract

This paper presents the load balance of Mobile IP in wireless systems. Mobile IP can support wireless users with continuous network connections while changing locations. If a failure occurs in a mobility agent, the wireless users located in the coverage area of the fault mobility agent will lose their network connections. To tolerate the failure of mobility agents, this paper proposes the method of an efficient approach to reduce severe overload.

I. 서론

노트북, 핸드폰, PDA 등과 같은 이동 단말기를 통해 인터넷 서비스를 받고자 하는 인구가 늘어남에 따라 사용자의 이동성을 지원하기 위한 다양한 기술들이 개발되어 왔으며 이동 중의 서비스 제공을 위한 연구가 매우 활발히 이루어지고 있다. IETF에서는 LAN 환경의 무선 통신 환경을 이용하면서 통신망을 위한 데이터그램 교환은 기존의 유선 인터넷 망을 그대로 이용할 수 있도록 하기 위하여 Mobile IP[1]를 제안하였다. 본 논문에서 제안하고자 하는 점은 이전 논문[2]의 알고리즘을 이용하면 부하가 한쪽으로만 몰리는 결과가 나타난다. 이러한 문제점을 풀기 위해 에이전트에서 Erlang's loss B 공식[3]을 이용해 인접해 있는 에이전트들의 차단되는 비율(Blocking rate)이 적은 쪽으로

모바일 노드를 연결해 줌으로써 위의 문제점을 풀 수 있고 부하가 잘 분배된 에이전트를 사용할 수 있는 알고리즘을 제시한다.

II. 배경 지식

2.1 Mobile IP

인터넷 사용자들은 언제 어디서나 고품질의 인터넷 서비스를 사용하기를 바라고 있으며, 휴대용 컴퓨터나 PDA와 같은 이동 단말기들의 성능 향상과 무선 통신 기술의 발전으로 인하여 그 사용자 수가 크게 증가하고 있다. 그러나 자신의 홈 네트워크를 떠나 있을 때 이동성이 있는 단말기로 인터넷을 사용하는 것은 그리 쉽지 않다. 즉, 이동한 곳에서 모델을 이용하거나 새로운 IP 주소를 할당받아야 하는데, 이는 기존의 IP protocol이 이동성을 보장해주지 못하기 때문이다. 이런 문제를 해결하기 위해 제안된 방법이 Mobile IP이다. Mobile IP 이용자는 자신의 홈 네트워크가 아닌 다른 곳에서도 기존의 IP 주소를 이용해, 인터넷을 사용할 수 있다. 상대방 역시 해당 노드가 어느 곳에 위치해 있는지를 인식할 필요 없이 해당 노드의 기존 IP 주소로만 연결을 설정하면 된다. 즉, Mobile IP란 노드의 이동을 검출하고, 이동 후의 네트워크에서도 이동전의 네트워크에서와 동일하게 통신할 수 있게 해주는 프로토콜이다.

2.2 시스템 모델

이 논문에서 나타내는 시스템 모델은 3세대[4] 무선

시스템으로 간주한다. 시스템 모델은 그림1에 나타나고 있다. 이 시스템 모델은 세 가지 중요 구성요소를 포함한다. 이 세 가지 요소는 MN(Mobile node), RAN(Radio Access Network) 그리고 Core Network이다. MN는 RAN과 무선 네트워크로 연결이 되고 RAN은 홈 또는 외부 에이전트들과 상호작용을 한다. 홈 에이전트와 외부 에이전트들은 Mobile IP기능을 통해서 무선 데이터 세션을 제공한다. 라우터들은 홈 에이전트와 외부 에이전트에서 패킷 전송을 돕는다.

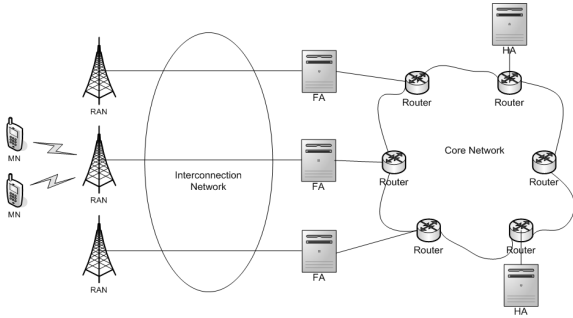


그림1. 무선 데이터 네트워크 모델

2.3 에이전트의 고장 허용방법

본 논문에서 실패는 단지 모바일 에이전트에서만 일어난다고 가정한다. 홈 에이전트와 외부 에이전트는 주기적으로 위치해 있는 서버넷에 광고 메시지를 전송한다[1]. 여기서 각 에이전트는 그림2와 같이 인접해 있는 두 에이전트의 정보를 저장한다고 가정한다. C 에이전트에서 실패가 일어났을 경우 C 에이전트는 더 이상 에이전트 메시지를 전송할 수 없다. 실패가 일어나지 않은 인접한 에이전트는 한 주기동안 C 에이전트에서 보내는 에이전트 메시지를 받지 못하고 C 에이전트에서 실패가 일어났다는 것을 인지하게 된다. 한 에이전트에서 실패가 일어나면 진행중인 데이터 요청은 잃게된다. 그러나 이런 잃어버린 데이터 요청과 응답 패킷은 신뢰성 있는 전송 계층에서 end-to-end를 이용해서 재전송 할 수 있다[5].

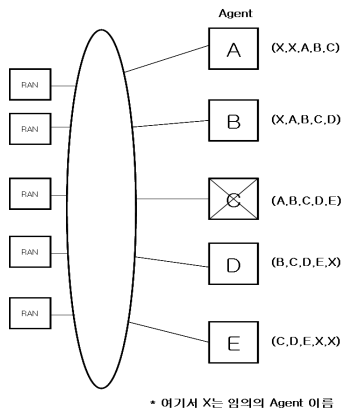


그림 2. 인접 에이전트의 정보 저장상황

하지만 이 논문에서는 어떻게 잃어버린 데이터를 요청하고 응답하는지에 대해서는 다루지 않는다.

2.4 관련 연구

이전 논문에서 제시한 Fault Tolerance 방법은 에이전트에서 실패가 일어나면 인접해 있는 실패 없는 에이전트가 실패가 일어난 에이전트의 모바일 노드를 연결시켜 준다. 연결 시켜줄 경우 에이전트는 실패가 일어난 에이전트의 연결된 모바일 노드의 개수를 자기 자신이 연결 시켜줄 수 있는 모바일 노드의 개수와 비교한 다음 충분히 연결 시킬 수 있는 모바일 노드의 수면 바로 연결 시키고 연결 시켜야 할 모바일 노드의 수가 초과하게 되면 인접한 실패없는 에이전트에게 연결시켜 주게 된다. 이렇게 연결이 될 경우 부하가 적은 쪽으로 연결이 될 수도 있지만 부하가 높은 쪽으로도 연결이 될 수 있는 문제점을 가지고 있다.

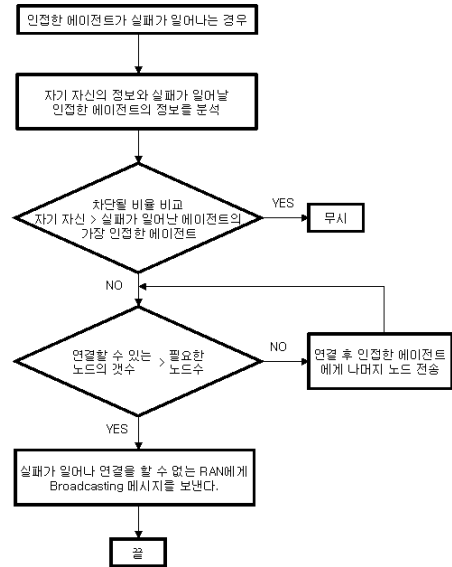


그림 3. 부하 균등화를 위한 흐름도

III. 제안된 부하 균등화 방법

3.1 부하 균등화

이전 논문에서 제시한 알고리즘은 에이전트에서 실패가 일어날 경우에 실패가 일어난 에이전트에 연결되어 있던 노드들을 인접해 있는 에이전트로 바로 연결 후 노드의 개수를 비교해서 연결을 할 수 있는만큼 노드를 연결시키고 다시 인접한 에이전트에게 남은 노드를 넘겨서 연결을 시켜 주는 방법을 제안했다. 이렇게 연결을 할 경우는 부하가 많이 걸려있는 쪽의 에이전트로 연결이 될 가능성이 있다. 하지만 본 논문에서는 인접한 에이전트에 실패가 일어날 경우 각 에이전트들이 인접한 에이전트의 차단되는 비율을 저장해 놓고

인접해 있는 에이전트의 실패를 인식하게 될 경우에는 실패한 에이전트에서 가장 가까운 두 에이전트가 자기 자신의 차단되는 비율과 상대방의 차단되는 비율을 비교한 뒤 차단될 확률이 높으면 무시하고 차단될 확률이 낮으면 광고 메시지를 전송해서 모바일 노드들을 연결시켜 준다. 만약 부하가 적음에도 불구하고 실패된 에이전트에 연결되어 있던 모바일 노드의 수가 초과하게 된다면 인접한 에이전트에게 남은 모바일 노드를 전송시키도록 넘겨준다. 이 과정은 그림3에 흐름도로 나타내고 있다. 차단되는 비율을 구하기 위해서는 잘 알려진 큐잉 시스템으로부터 Erlang's loss B 공식을 이용한다. 이렇게 함으로써 부하가 높은쪽으로 연결되는 것을 막을 수 있고 부하가 고르게 분포되는 부하 균등화를 만들 수 있다.

표 1. 매개 변수의 의미

매개 변수	의 미
N_{Agent}	무선 시스템에서 에이전트의 수
F_{Agent}	무선 시스템에서 실패가 일어난 에이전트의 수
C_{Agent}	에이전트에서 데이터를 다룰 수 있는 유닛의 수
λ_{Agent}	에이전트에 데이터 도착 비율
μ_{Agent}	에이전트에서 서비스 될 데이터의 비율
$W_{FAk}(W_{HAk})$	실패가 일어난 에이전트에서 실패 없는 에이전트로의 부하 비율 $(\sum_{k=1}^{N_{FA}-F_{FA}} w_{FA_k}=1, \sum_{k=1}^{N_{HA}-F_{HA}} w_{HA_k}=1)$

3.2 에이전트에서 실패가 일어난 경우의 블로킹 확률

3.2.1 외부 에이전트

외부 에이전트에서 실패가 일어났을 경우는 각 에이전트에서 차단될 비율을 구한 뒤 비교해야 한다. 우선 외부 에이전트에서 차단될 비율은 다음과 같이 구한다. 우선 외부 에이전트에서 차단될 확률을 $P_{FA_Blocking}$ 이고 데이터 요청에 의해 실패가 없는 에이전트에 도착할 비율을 λ_{FA} 로 놓는다. 큐잉 시스템의 Erlang's loss B 공식을 살펴보면 다음과 같다.

$$\frac{\left(\frac{\lambda_{FA}}{\mu_{FA}}\right)^{c_{FA}}}{c_{FA}!} \quad (1)$$

$$\sum_{i=0}^{c_{FA}} \frac{\left(\frac{\lambda_{FA}}{\mu_{FA}}\right)^i}{i!}$$

어떤 외부 에이전트에서 실패가 일어나면 실패가 일어난 외부에이전트에 있던 노드들이 인접해 있는 실패 없는 외부 에이전트로 데이터 요구를 하게 된다. 인접해 있는 실패 없는 외부 에이전트는 비율이 점점 커지고 $\lambda_{FA}+F_{FA}\times\lambda_{FA}\times W_{FAk}$ 의 값을 갖게 된다. 이 식을 (1)

식에 대입하면 (2)번과 같은 차단될 확률이 나온다.

$$\frac{\left(\frac{\lambda_{FA}+F_{FA}\times\lambda_{FA}\times w_{FA_k}}{\mu_{FA}}\right)^{c_{FA}}}{c_{FA}!} \quad (2)$$

$$\sum_{i=0}^{c_{FA}} \frac{\left(\frac{\lambda_{FA}+F_{FA}\times\lambda_{FA}\times w_{FA_k}}{\mu_{FA}}\right)^i}{i!}$$

(1)과 (2)번식을 이용해서 (3)번 외부 에이전트에서 차단될 확률($P_{FA_Blocking}$)을 다음과 같이 이끌어 낼 수 있다.

$$\frac{\left(\frac{\lambda_{FA}+F_{FA}\times\lambda_{FA}\times w_{FA_k}}{\mu_{FA}}\right)^{c_{FA}}}{c_{FA}!} - \frac{\left(\frac{\lambda_{FA}}{\mu_{FA}}\right)^{c_{FA}}}{c_{FA}!} \quad (3)$$

$$\sum_{i=0}^{c_{FA}} \frac{\left(\frac{\lambda_{FA}+F_{FA}\times\lambda_{FA}\times w_{FA_k}}{\mu_{FA}}\right)^i}{i!} - \sum_{i=0}^{c_{FA}} \frac{\left(\frac{\lambda_{FA}}{\mu_{FA}}\right)^i}{i!}$$

각 에이전트는 에이전트 메시지를 보낼때 메시지 안에 차단되는 비율을 저장해서 인접한 에이전트들에게 전송을 하고 메시지를 받은 에이전트들은 차단되는 비율을 저장하고 있다가 인접한 에이전트가 실패가 일어날 경우 자기 자신의 차단되는 비율과 비교해서 3.1절의 부하 균등화와 같은 방법으로 실행한다.

3.2.2 홈 에이전트

홈 에이전트에서 실패가 일어났을 경우 외부 에이전트와 마찬가지로 인접한 에이전트들은 실패가 일어난 에이전트 메시지를 한 주기동안 받지 못하게 된다. 인접한 에이전트들은 실패가 일어난 에이전트를 알게 되고 (4)번 식과 같이 홈 에이전트에서 걸리는 차단되는 비율($P_{HA_Blocking}$)을 구한다. 홈 에이전트의 차단되는 비율은 외부 에이전트의 차단되는 비율과 유사하다.

$$\frac{\left(\frac{\lambda_{HA}+F_{HA}\times\lambda_{HA}\times w_{HA_k}}{\mu_{HA}}\right)^{c_{HA}}}{c_{HA}!} - \frac{\left(\frac{\lambda_{HA}}{\mu_{HA}}\right)^{c_{HA}}}{c_{HA}!} \quad (4)$$

$$\sum_{i=0}^{c_{HA}} \frac{\left(\frac{\lambda_{HA}+F_{HA}\times\lambda_{HA}\times w_{HA_k}}{\mu_{HA}}\right)^i}{i!} - \sum_{i=0}^{c_{HA}} \frac{\left(\frac{\lambda_{HA}}{\mu_{HA}}\right)^i}{i!}$$

CN(Correspondent Node)는 모바일 노드와 통신을 하기 위해서는 우선 모바일 노드가 처음 등록되어 있던 홈 에이전트를 거쳐서 통신을 하게 된다. 하지만 실패가 일어나서 홈 에이전트 대신 인접한 에이전트가 대행할 경우에는 모바일 노드가 등록되어 있지 않기 때문에 모바일 노드가 있는 위치를 전부 검색해야 한다. 단, 모바일 노드가 이동을 해서 홈 에이전트에게 등록을 하게 되면 등록된 노드와는 바로 통신이 가능하다.

IV. 시뮬레이션 및 성능분석

시뮬레이션은 제안된 접근방법의 부하 균등화 비율

을 비교하기 위해서 여러 가지 제안상황을 제시하고 비교 분석하였다. 실험 시뮬레이션을 위해 사용된 프로그램은 모바일 IP가 기반이 된 NS2(Network Simulator 2)[6]를 이용하였다. 사용된 시스템은 CPU 1.6G, DDR SDRAM 256M, 운영체제는 Fedora Core로 구성되어 있다. 시뮬레이션 환경은 그림 2와 같이 에이전트 다섯 개가 인접해 있는 상황을 이용한다.

결과값을 비교하기 위해 에이전트 세 곳에 차단되는 비율을 B=25%, C=50%, D=90%로 설정한다. 세 에이전트 중에서 C 에이전트가 실패가 일어난다고 설정하고 제안된 알고리즘을 이용하면 그림 4와 같이 나타난다.

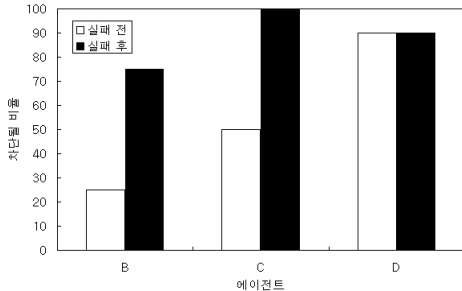


그림 4. 에이전트 부하율

실패가 일어난 후의 부하율을 살펴보면 부하가 적은 쪽으로 연결이 되면서 부하가 균등화 됨을 알 수 있다.

다른 접근 방법으로는 한 에이전트에서 실패가 일어나서 인접한 에이전트가 그 일을 대행하지만 실패가 일어난 에이전트에서 연결중인 모바일 노드의 개수가 자신이 연결 시켜줄 수 있는 모바일 노드의 수를 감당할 수 없을 때 다시 인접한 에이전트에게 모바일 노드를 연결 시켜줌으로써 원활한 통신을 할 수 있는 방법을 제시한다. 에이전트 다섯 곳에 차단되는 비율을 A=50%, B=70%, C=50%, D=90%, E=85%로 설정한다. 다섯 에이전트중 C 에이전트가 실패가 일어났다고 설정하고 제안된 알고리즘을 사용하면 그림 5와 같이 나타난다.

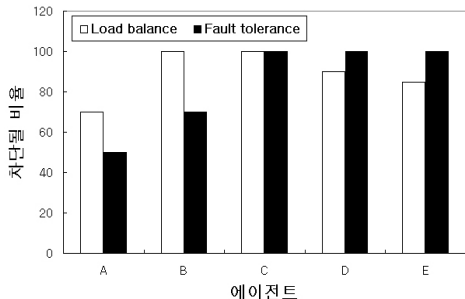


그림 5. 본 논문과 이전 논문의 부하율 결과표

여러번의 실험결과 평균적으로 에이전트 전부 차단될 비율이 높거나 전부 낮을 경우에는 큰 효과가 없지만 그 이외의 경우에는 본 논문에서 제시한 알고리즘

이 더욱더 부하를 균등화 시킴을 알 수 있다.

V. 결론 및 향후 연구 방향

이 논문에서는 무선 네트워크 상황에서 어떤 에이전트가 고장을 일으켜서 모바일 노드를 연결 시켜 줄 수 없을 때 실패가 일어난 에이전트에 인접해 있는 에이전트 중에서 부하가 적은 에이전트를 통해 모바일 노드들을 연결해 줌으로써 부하가 적은 쪽으로만 이동하게 되는 부하 균등화의 장점이 있다. 이전 논문과 비교한 결과값을 살펴보면 부하가 균일하게 분포됨을 알 수 있다. 또한 소프트웨어적으로 처리되기 때문에 하드웨어의 지원이 필요하지 않게 됨으로써 큰 비용이 추가되지 않는다.

향후 연구 방향은 홈 에이전트에서 실패가 일어났을 때 외부 CN가 모바일 노드와 통신을 하고자 할 때 실패가 일어난 홈 에이전트 역할을 대신 해주는 에이전트에는 모바일 노드의 위치정보가 존재하지 않기 때문에 모바일 노드가 어디에 위치하고 있는지 전부 검색을 해야한다. 효율적인 통신 환경을 위해서 이런 문제점을 보완하는 연구가 필요하다.

참고문헌

- [1] C. Perkins, "IP Mobility Support for IPv4", Technical Report IETF RFC 3220, Jan. 2002
- [2] Jenn-Wei Lin, Joseph Arul, "An Efficient Fault-Tolerant Approach for Mobile IP in Wireless Systems", IEEE TRANSACTION ON MOBILE COMPUTING, vol.2, no.3, pp.207-220, Aug. 2002.
- [3] MICHAŁ PIÓRO, DEEPANKAR MEDHI, "Routing, Flow, and Capacity Design in Communication and Computer Networks", MORGAN KAUFMANN, p.18-19, 2004
- [4] B. Sarikaya, "Packet Mode in Wireless Networks : Overview of Transition to Third Generation", IEEE Comm. Magazine, vol.38, no.9, pp.164-172, Sept. 2000.
- [5] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", IEEE/ACM Trans. Networking, vol.5, no.6, pp.756-769, Dec. 1997.
- [6] NS-2, <http://www.isi.edu/nsnam/ns/>