

이동 코드를 이용한 네트워크 관리 모델링

권혁찬, 유관중

충남대학교 컴퓨터과학과

{hckwon, kjyoo}@nspplab.chungnam.ac.kr

Modeling for Network Management with Mobile Code

Hyeok-Chan Kwon, Kwan-Jong Yoo

Dept. of Computer Science, ChungNam Nat'l. University

요약

네트워크 관리 시스템은 네트워크의 효율성과 생산성을 최대화하기 위해 복잡한 네트워크를 통제하는 시스템이다. 네트워크 관리에 사용된 기존의 방식은 클라이언트 서버 방식으로, 대표적으로 SNMP(Simple Network Management Protocol), CMIP(Common Management Information Protocol) 등의 프로토콜을 사용한다. 최근 들어서는 이동 에이전트를 이용한 네트워크 관리에 대한 연구도 여러 기관에서 진행 중에 있다. 본 논문에서는 네트워크 관리 시스템 개발 시 기존의 RPC 방식을 이용한 경우와 이동 에이전트를 이용한 경우 각각의 성능을 평가하기 위한 평가 모델을 제시하고자 한다. 본 모델을 기초로 차후 네트워크 관리시스템 개발 시 적합한 수행 구조를 선택하는데 도움이 될 수 있을 것이다.

1. 서론

네트워크 관리 시스템은 네트워크의 효율성과 생산성을 최대화하기 위해 복잡한 네트워크를 통제하는 시스템이다. 네트워크 관리에 사용된 기존의 방식은 클라이언트 서버 방식으로, 대표적인 프로토콜로 SNMP(Simple Network Management Protocol), CMIP(Common Management Information Protocol) 등이 있다[2].

최근 들어서는 네트워크 관리를 위해 이동 에이전트를 이용하는 연구가 활기를 띠고 있다. 기존의 RPC 나 멀티 에이전트가 글로벌(global) 통신에 의해 작업을 수행했던 반면, 이동 에이전트 시스템은 에이전트 자신이 서버를 돌아다니며 주어진 작업을 수행하는 구조를 갖는다.

이동 에이전트의 서버로 이동하는 이러한 특성은 글로벌 통신비용을 줄일 수 있다는 장점이 있기는 하지만, 통신에 사용되는 메시지, 반환되는 결과 값, 누적되는 데이터 크기, 통신 횟수 등의 여러 요소에 의해 오히려 성능이 저하되는 경우도 있다. 따라서 정확한 성능을 비교 평가하기 위해서는 직관적인 평가보다 실제 수행 능력을 평가할 수 있는 수행 모델이 필요할 것이다.

본 논문은 네트워크 관리를 위한 이동 에이전트와 기존의 RPC 방식의 수행을 평가하기 위한 평가모델을 제안한다. RPC 방식은 기존에 네트워크 관리에 대표적으로 사용되는 프로토콜인

SNMP(Simple Network Management Protocol)[2]를 사용한다고 가정한다. 이동 에이전트는 데이터를 누적 시키며 이동하는 전형적인 구조와 locker 패턴이 적용된 구조[6] 각각에 대해 모델링 한다. 본 모델은 네트워크 트래픽에 주안점을 두었으며 기타 CPU, 메모리의 비용 등은 고려하지 않았다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 네트워크 관리의 개요를 살펴보고 3장에서 네트워크 관리를 위한 모델을 제안한다. 마지막으로 추후과제와 결론은 4장에서 맺는다.

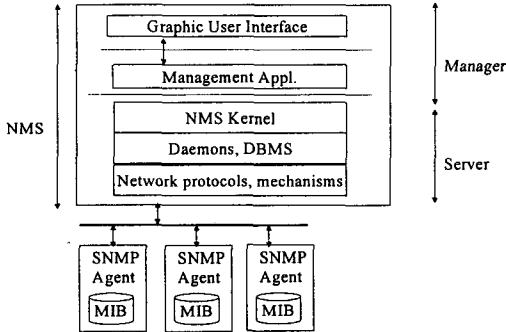
2. 네트워크 관리

일반적인 네트워크 관리 시스템의 구조는 [그림 1]과 같다.

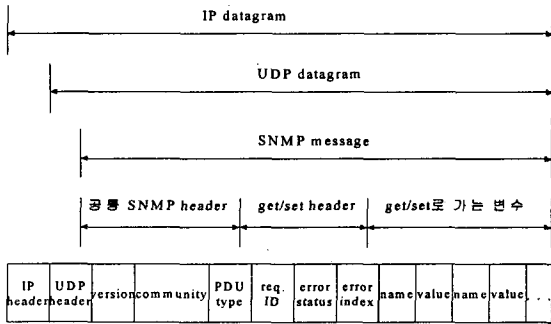
NMS(Network Management Station)는 네트워크 관리자이다. 네트워크 관리자(NMS)와 SNMP 에이전트 사이에 특정한 정보를 주고 받는 것이 네트워크 관리의 기본이다. 관리되어야 할 특정한 정보, 자원 등을 모아놓은 집합체가 MIB(Management Information Base)이다. 네트워크를 관리한다는 것은 관리 대상인 장비들이 제공하는 MIB 중에서 특정 값을 얻어와 그 장비의 상태를 파악하거나 그 값을 변경함을 의미한다.

NMS와 SNMP 에이전트간에 통신을 위해 사용되는 프로토콜이 SNMP이며 get-request, get-response, get-next-request, set-request, trap 등의

5가지 primitive가 제공된다. SNMP는 데이터 전송 시 UDP를 사용한다. SNMP 메시지의 형식은 [그림 2]와 같다



[그림 1] 전형적인 네트워크 관리 구조



[그림 2] SNMP 메시지 형식

3. 수행 평가 모델

[표 1]에서 기본 모델에 사용되는 파라미터들을 볼 수 있다.

[표 1] 모델링을 위한 파라미터

파라미터	설명
N	관리되는 노드의 수
R _i	i 번 노드로의 request 횟수
S _{nr}	n 번 노드에서 r 번째 request 시 요구되는 SNMP variable 의 수
get_req	request 메시지 헤더의 크기 (IP+UDP+SNMP header)
get_res	response 메시지 헤더의 크기 (IP+UDP+SNMP header)
s _{va}	variable binding 의 크기 (SNMP variable + variable name)
μ	Marshalling / Unmarshalling 비용
σ	네트워크 딜레이(delay)
β	네트워크 대역폭(bandwidth)

모델링을 위해 다음과 같은 가정을 둔다.

- 데이터를 marshalling 하고 unmarshalling 하는 시간은 데이터의 크기에 비례한다
- 네트워크 환경은 동일한 대역폭을 갖는다.
- RPC 방식은 SNMP 프로토콜을 사용한다.

3.1 RPC를 위한 모델

RPC[4]는 클라이언트 서버 모델로 네트워크 관리자가 관리되는 노드로 SNMP 변수를 요구하거나 변수의 값을 변경 할 것을 요구하면 관리되는 노드는 관리자로부터의 요구에 응답한다. 관리자와 서버는 작업 수행 시 글로벌 통신을 사용하게 된다.

RPC에 대한 전체 network load는 식 (1)과 같다.

$$LRPC = \sum_{n=1}^N \sum_{r=1}^{Rn} (get_req + get_res + (Snr * s_va)) \quad (1)$$

RPC에 대한 전체 수행 시간은 식 (2)과 같다.

$$TRPC = 2 \sum_{n=1}^N \sum_{r=1}^{Rn} \delta + LRPC \left(\frac{1}{\beta} + 2\mu \right) \quad (2)$$

수행 시간을 계산하기 위해 네트워크 지연시간(delay), 대역폭(bandwidth), marshalling 시간 등이 추가되었다.

본 모델에서 가정된 네트워크 환경은 균일한 대역폭을 갖는다. 만일 각기 다른 대역폭을 갖는 서브네트워크(sub network)를 관리하는 경우엔 조금 다른 모델이 적용될 것이다. 만약 관리자와 서브네트워크간의 대역폭이 β₀이고 서브 네트워크 내의 대역폭이 β₁ 이라면 실제 적용되는 대역폭은 두 대역폭의 최소값인 min(β₀, β₁)이 될 것이다.

3.2 이동 에이전트를 위한 모델

실제 전송되는 이동 에이전트의 크기 S_{MA}는 식 (3)과 같다.

$$S_{MA} = D_{MA} + (D_{MA} / \text{payload}) \quad (3)$$

$$* (IPheader + TCPheader)$$

$$D_{MA} = \text{agent_header} + MA$$

식 (3)의 payload는 TCP payload이다. 일반적인 이동 에이전트의 전송 데이터 형식은 (ATP header + payload)이다. 전송 데이터는 다시 TCP 메시지로 분할된다. 분할된 각각의 조각들은 TCP헤더와 IP 헤더로 싸여 TCP/IP 방식으로 전송된다.

데이터를 계속 누적시키며 이동하는 전형적인 이동 에이전트의 전체 network load는 식 (4)와 같다.

$$LMA = \sum_{n=0}^N (S_{MA} + \sum_{i=1}^n \sum_{r=1}^{Rn} (Snr * s_va)) \quad (4)$$

각각의 노드 방문 시 검색된 SNMP 변수들과 상태들은 에이전트의 이동 시 데이터 영역에 계속 누적된다.

이동 에이전트의 전체 실행 시간은 식 (5)와 같다.

$$TMA = 2\delta + LMA\left(\frac{1}{\beta} + 2\mu\right) \quad (5)$$

3.3 locker pattern을 적용한 이동 에이전트를 위한 모델

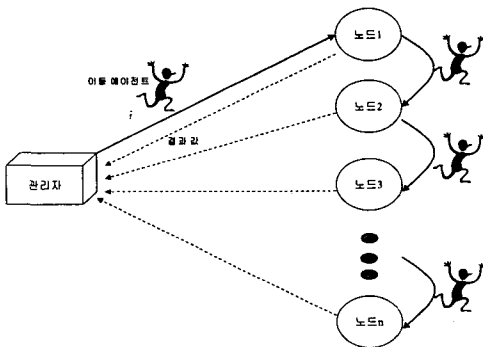
[그림 3]과 같이 Locker pattern[6]은 이동 에이전트가 방문하는 각각의 노드에 자신만의 영역을 갖고 있어서 그 노드에서 얻은 결과를 그곳에 보관 시키고 다음 노드로 이동하는 구조이다. 이때 결과는 또 다른 이동 에이전트를 파견하여 수집하거나 각각의 노드에서 RPC 방식으로 관리자에게 전달해준다. 본 모델에서는 후자의 경우로 가정하였다.

Locker pattern을 적용한 이동 에이전트의 경우 network load는 식 (6)와 같다.

$$LMA(L) = \sum_{n=0}^N (S_MA + \sum_{r=1}^{Rn} (get_res + (Snr * s_va))) \quad (6)$$

식 (6)에서 보듯이 이동하는 에이전트의 크기와, 관리자에게 전달하는 reply 메시지의 크기만 고려하면 된다. 이는 이동 에이전트가 이동 시 검색된 데이터가 누적되지 않기 때문이다. 전체 수행 시간은 식 (7)과 같다.

$$TMA = 2N\delta + LRPC\left(\frac{1}{\beta} + 2\mu\right) \quad (7)$$



[그림 3] locker pattern

4. 결론

이동 에이전트의 가장 큰 장점은 네트워크 통신 비용을 줄일 수 있다는 점이다. 이동 에이전트는 코드가 직접 이동하여 글로벌 통신이 아닌 방문한 노드에서의 지역적인(local) 통신에 의해 작업을 처리하고 그 결과를 가져오기 때문이다. 최근 들어 이동 에이전트를 데이터 마이닝(data mining), 네트워크 관리, 전자 상거래(Electronic

Commerce) 등에 응용하기 위한 연구가 활성화 되고 있다.

본 논문에서는 이동 에이전트를 응용할 수 있는 분야 중 하나인 네트워크 관리에서의 RPC와 이동 에이전트의 수행을 평가하기 위한 모델을 제안하였다. 이동 에이전트는 데이터가 누적되는 전형적인 구조와 Locker pattern이 적용된 구조 각각에 대해 모델링 하였다. 이러한 수행 평가 모델은 차후 분산 애플리케이션 개발 시 적합한 수행 패턴을 결정하는데 참고 자료가 될 수 있을 것이다. 현재 본 모델에 샘플 데이터를 적용하여 실제 성능을 평가, 비교해 보기 위한 시뮬레이션(simulation) 작업이 진행중이다.

추후 과제로서 네트워크 bandwidth가 혼합된 구조, 이를테면 wireless network와 LAN이 혼합된 경우와 같은 다양한 환경에서 적용될 수 있는 모델링을 하는 일이 있다. 또한 추가로 고려할 수 있는 파라미터들을 추출하는 일과 그것을 모델에 적용하는 일, 새로운 이동 에이전트 수행 패턴을 분석하여 그에 맞는 모델을 만드는 작업등이 필요하다.

참고문헌

- [1] A. Bieszczad, T. White and B. Pagurek, "Mobile Agents for Network Management," IEEE Communications Surveys, September 1998
- [2] W. Stallings, SNMP, SNMPv2 and RMON, Addison Wesley, 1996
- [3] Harrison, C. G., D. M. Chess, A. Kershenbaum, Mobile Agents: Are they a good idea ?, IBM Watson Research Center, Mar. 1995.
- [4] A. Carzaniga, G. P. Picco, G. Vigna, "Designing Distributed Applications with Mobile Code Paradigms," Proceedings of the 19th International Conference on Software Engineering, Boston, 1997.
- [5] M. Strasser, M. Schwehr, "A Performance Model for Mobile Agent Systems," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications PDPTA'97, Volume II, pp. 1132-1140, 1997.
- [6] Y. Aridor, D. B. Lange, "Agent Design Patterns: Elements of Agent Application Design," Second International Conference on Autonomous Agents (Agents 98), 1998.
- [7] A. Sahai, C. Morin, "Enabling a Mobile Network Manager through Mobile Agents," In Proceedings of the Mobile Agents'98, Stuttgart, Germany, Sept 1998.
- [8] H.C.Kwon, J.H.Lee, H.J.Park, B.N.Yoon and K.J.Yoo, "A Performance Evaluation Model for Mobile Agent System," Proc. of the 1'st International Conference on Advanced Communication Technology (ICACT'99), Muju resorts, Korea, pp.303-306, Feb. 1999