

이동 적응망 응용을 지원하기 위한 서비스 브라우저 구조

이은주 정현구 김세훈 김성완 윤현주^o 이준원 김진수
한국과학기술원, 한국정보통신대학교*

{leeej, hkjeong, kimsh, swkim}@camars.kaist.ac.kr, juyoon@icu.ac.kr^o, {joon, jinsoo}@cs.kaist.ac.kr

A Service Browser Architecture for Supporting MANET Applications

Eunju Lee Hyunku Jeong Sehoon Kim Sungwan Kim Hyeon-Ju Yoon^o Joon-Won Lee Jin-Soo Kim
Information and Communication University*, Korea Advanced Institute of Science and Technology

요 약

우리는 현재 유선망에서 웹 브라우저와 여러 검색엔진을 통해 필요한 서비스를 찾고 다운로드 받을 수 있다. 그러나 이동 적응망에서는 사용자가 쉽게 사용할 수 있는 브라우저가 아직 개발된 것이 없으며 유선망에서와 같이 서비스 검색을 담당할 중앙의 서버를 가정할 수도 없다. 본 논문에서는 이동 적응망의 노드들이 망에 존재하는 서비스와 주변 노드들을 검색, 관리하기 위한 서비스 브라우저와 이를 뒷받침할 프로토콜을 소개한다.

1. 서 론

최근 노트북이나 PDA와 같은 휴대용 컴퓨터가 발달함에 따라 무선망에 대한 연구가 활발해지고 있다. 이동 적응망(MANET: Mobile Ad Hoc Network)도 그러한 흐름 중에 하나이다. 이동 적응망의 특징은 모든 노드들이 기간 망에 의존하지 않고 라우터로서 동작하며, 무선의 멀티 홉 링크를 통해 서로 통신한다는 것으로 요약될 수 있다. 이러한 이동 적응망에 대한 연구는 여러 방향에서 진행되고 있는데, 토폴로지가 유동적으로 변하는 환경에서 효율적으로 통신을 하기 위한 라우팅 프로토콜의 개발, 휴대용 컴퓨터의 효율적 전력 소비를 위한 메커니즘, 노드 사이 통신의 품질 보장을 위한 대역폭 확보 등에 관한 연구가 그것이다. 이들 중에서 실제 이동 적응망에서 활용될 수 있는 응용프로그램들을 고려한 연구는 드물다. 본 논문에서는 이동 적응망에서 꼭 필요한 응용프로그램 중의 하나인 서비스 브라우저를 구현하고, 이를 통해 앞으로 행해져야 할 연구의 방향을 제시하고자 한다.

이번 정보과학회에 참가한 모든 사람들이 노트북이나 PDA 등 휴대용 컴퓨터를 소유하고 있고, 이들 컴퓨터는 이동 적응망 모드에서 동작하고 있다고 가정해보자. 참가자 중 발표자들은 자신의 컴퓨터에 발표 자료와 논문을 저장하고 있고 다른 사람들에게 배포할 의사가 있다. 학회 주최측의 도우미들의 컴퓨터는 전체 일정 안내문과 길 안내 서비스, 참가자 간에 의사소통을 위한 채팅 서비스 등을 제공한다. 이 때, 학회에 참가한 네트워크를 연구하고 있는 A는 네트워크 분야의 발표가 어디에서 진행되는지를 알기 위해 도우미의 컴퓨터에 있는 일정 안내문을 보아야 한다. 이를 위해서 기본적으로 각 컴퓨터의 주소가 고정되어 있어야 한다. A는 미리 도우미 컴퓨터의 주소를 알고, 해당 컴퓨터에 접속하여 일정 안내문을 찾고 다운로드 받아야 할 것이다. 일정 안내문에는 발표 위치와 함께 각 발표자의 컴퓨터 주소를 담고 있어서 A가 각 컴퓨터에 접속해서 발표자료를 다운로드 받아야 한다. 만약 이러한 번거로운 과정을 거치지 않고 몇 번의 클릭만으로 자동으로 원하는 자료나 서비스를 찾고 다운로드 받아서 실행할 수 있는

응용프로그램이 있다면 편리할 것이다. 기간망이 갖추어진 환경이라면 웹 브라우저를 통하여 서비스가 위치한 url을 알아내고, 서비스 제공자와 직접 통신할 수 있을 것이다. 그러나 이동 적응망에서는 노드의 이동이 빈번하기 때문에 웹 서버를 가정할 수 없고, 서비스 제공자의 주소가 유동적으로 변하기 때문에 고정된 url을 사용할 수 없다. 따라서 이동 적응망의 특성을 고려하여 망에 존재하는 서비스를 검색하는 방법과 이를 기반으로 사용자에게 편의를 제공할 수 있는 응용프로그램에 대한 요구는 절실해 진다. 또한 이 응용프로그램이 망에 있는 노드들 중에서 자주 통신을 할 노드들을 그룹으로 묶어서 관리한다면 더욱 효율적일 것이다. 즉 위 예시의 경우, 연구 분야에 따라 그룹을 나누고, 관심 있는 그룹에 가입하여 해당 그룹의 발표자료들을 찾고 그룹의 구성원들끼리 모여 채팅 등 필요한 서비스를 할 수 있도록 한다면, 주로 그룹 내에서만 노드나 서비스를 찾기 때문에 망의 부하와 서비스 검색 시간을 줄일 수 있을 것이다.

이를 위하여 본 논문에서는 이동 적응망에 존재하는 서비스와 노드를 관리하여, 사용자에게 편의를 제공하는 서비스 브라우저(Service browser)를 제안하였다. 서비스 브라우저의 개발은 사용자가 망에 존재하는 서비스를 사용할 때 편의를 제공하는 응용프로그램으로서뿐만 아니라 이동 적응망에서의 실제 응용프로그램을 개발함으로써 응용프로그램 계층의 관점에서 이동 적응망에 실제 필요한 연구의 방향을 제시한다는 것에서도 의미가 있다.

논문은 다음과 같이 구성되어 있다. 2장에서 서비스 검색 프로토콜과 서비스 브라우저에 대한 기존의 연구들을 살펴보고, 3장에서 서비스 브라우저가 제공하는 기능과 이러한 기능을 수행하기 위한 시스템 구조 및 프로토콜을 설명할 것이다. 4장에서는 결론과 향후 과제를 제시할 것이다.

2. 관련 연구

2-1. 서비스 검색 프로토콜

서비스 브라우저의 주요 기능은 망에 존재하는 서비스를 검색하는 것이다. 서비스 검색 프로토콜은 서비스 목록을 관리하는 서버를 이용하는 중앙관리 방식과 중앙의 서버를 이용하지 않는 분산적 방식으로 나눌 수 있다. Jini[1], UPnP[2], SLP[3] 등의 기존의 프로토콜은 서버 기반의 서비스 검색 프로토콜로서 각 노드는 자기가 가지고 있는 서비스를 인덱스(index) 서버에 등록하고 서비스가 필요할 때 중앙의 서버를 통하여 찾는다. 그러나 이동 적용망은 노드들의 이동이 빈번하므로 중앙의 안정적인 서버를 가정하고 있는 이러한 프로토콜은 적합하지 않다.

이와 같은 이유로 이동 적용망에 적합한 서비스 검색 방법은 분산적인 방식이다. 분산적인 방식은 기본적으로 풀(Pull), 푸쉬(Push), 하이브리드(Hybrid) 모델로 나눌 수 있다. 풀 모델은 각 노드가 서비스가 필요한 시점에 서비스 요청 메시지를 브로드캐스트(broadcast)하고, 해당 서비스를 가지고 있는 노드가 응답을 보냄으로써 서비스 검색을 하는 방법으로, 서비스를 찾을 때까지의 시간이 길다는 단점이 있다. 반대로 푸쉬 모델은 각 노드가 자신이 가지고 있는 서비스 목록을 주기적으로 브로드캐스트하여 서비스가 필요로 할 때 자신이 받은 서비스 목록을 바탕으로 그 서비스를 가지고 있는 노드와 직접 통신하는 방식이다. 이 방법은 주기적으로 목록을 보내주어야 하기 때문에 망에 부하를 많이 일으키게 된다는 단점이 있다. 풀 모델과 푸쉬 모델의 단점을 보완하기 위한 방법인 하이브리드 모델은 각 노드가 처음에 망에 들어갔을 때나 새로운 서비스가 있을 때 전체 망에 자신의 서비스 목록을 브로드캐스트하고, 서비스를 필요로 하는 노드는 서비스 요청 메시지를 브로드캐스트하는 방식을 취한다. 이 방식은 서비스 목록을 주기적으로 브로드캐스트하지 않으므로 푸쉬 모델 보다 망에 부하를 덜 주고, 서비스 목록이 망 전체에 적절히 분산 되어 있으므로 풀 모델에 비하여 서비스 요청에 대한 응답이 빠를 수 있다. 이러한 기본적 방식에서 좀 더 성능을 향상 시키기 위한 연구로 [4], [5]가 있다. 우리가 개발하는 서비스 브라우저는 개발을 간단히 하기 위해서 풀 모델의 서비스 검색을 가정하였고, 전체 구조에서 서비스 관리 프로토콜만 수정하면 어떤 분산적인 방식의 서비스 검색 모델에서도 동작할 수 있다.

2-2. 서비스 브라우저

[6]은 SLPv2를 위한 서비스 브라우저인 UaBrowser의 기능과 특징, 사용자 인터페이스 등을 설명하고 있다. UaBrowser는 SLPv2를 기반으로 하고 있기 때문에 중앙에 서비스 목록을 유지하고 있는 디렉토리 에이전트(directory agent)를 사용하고, 주변 노드에 대한 정보를 관리하지 않는 반면, 본 논문에서 제안한 서비스 브라우저는 분산적인 방식의 서비스 검색 모델을 사용하고 그룹이라는 개념을 도입하여 망에 있는 노드들의 정보를 관리한다. 또한 [6]은 서비스 브라우저가 수행해야 할 기능들을 사용자 인터페이스를 중심으로 설명하고 서비스 검색 프로토콜로 SLPv2를 사용한다고 할 뿐, 내부 구조나 프로토콜에 대한 언급을 포함하지 않고 있다. 본 논문에서는 [6]와는 다른 관점에서, 즉 내부 구조와 프로토콜 측면에서 서비스 브라우저를 설명할 것이다.

3. 서비스 브라우저

이번 장에서는 서비스 브라우저가 제공해야 할 기능들을 짚어보고, 이들 기능을 수행하기 위한 서비스 브라우저의

구조 및 프로토콜에 대해 설명한다.

3-1. 기능

서비스 브라우저의 주요 기능은 노드와 서비스를 관리하는 것이다. 이를 위하여 망에 존재하는 노드를 관리하기 위해서 그룹의 개념을 도입했다. 그룹은 물리적인 토폴로지와는 상관없는 논리적인 개념으로서, 각각의 노드는 한 개 이상의 그룹에 속하고, 자신이 속한 그룹의 이름과 비밀 번호는 오프라인 상으로 이미 알고 있다고 가정하였다.

서비스 브라우저가 제공하는 주요 기능은 다음과 같이 요약할 수 있다.

- 이웃 노드(자신이 속한 그룹에 있는 노드)를 보여준다.
- 이웃 노드들의 서비스들을 보여준다.
- 전체 망에서 특정 조건을 만족하는 노드나 서비스들을 찾는다.
- 찾은 서비스들 다운로드 하고, 실행시킨다.
- 새로운 그룹을 생성하거나, 기존 그룹에 참여할 수 있다.

이 밖에도 사용자 등록 및 탈퇴 기능, 서비스 등록 및 삭제 기능 등을 제공한다.

3-2. 구조

위에서 언급한 기능을 수행하기 위해서 서비스 브라우저는 서비스 관리자(service management protocol manager), 사용자 관리자(user management protocol manager), 그룹 관리자(group management protocol manager), 전송 관리자(transfer management protocol manager) 등의 모듈로 구성되어 있다. 또한, 그룹 정보, 사용자 정보, 서비스 정보 등의 데이터를 저장하기 위한 MIB(Management Information Base)와 노드 간의 통신에서 사용될 메시지의 형식인 XML 문서로의 변환 및 해석을 담당하기 위한 XML Manager도 필요하다. 그림1은 서비스 브라우저의 내부 구조를 나타낸 것이다.

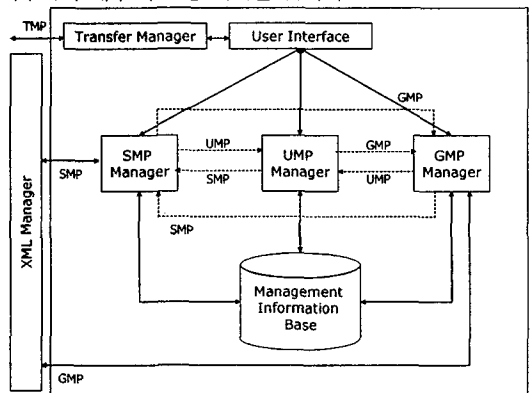


그림1. 서비스 브라우저 내부 구조

사용자가 유저 인터페이스를 통해 요청한 명령들은 유저 인터페이스에 의해 적절한 관리자에게 할당되고, 각 관리자들은 주어진 일을 각자 수행하거나, 여러 관리자가 협업하여 수행하고 그 결과를 유저 인터페이스에 돌려준다. 서비스 브라우저가 필요로 하는 여러 데이터는 MIB에 저장되는데, 그림2는 MIB를 도식화한 것이다. MIB는 멀티 유저 기능을 지원하기 위하여 사용자 목록(user list)을, 각 사용자마다 속한 그룹을 관리하기 위해서 그룹 목록(group list)을, 그룹에 속한 노드들을 관리하기 위하여 (사용자,

노드) 쌍인 이웃 노드 목록(neighbor list)을, 서비스 검색을 위하여 자신이 제공하는 서비스의 설명서(service description)를 담고 있는 목록(my service list)과 다른 노드의 서비스 설명서를 담고 있는 서비스 캐시(advertised service list)를 저장하고 있다.

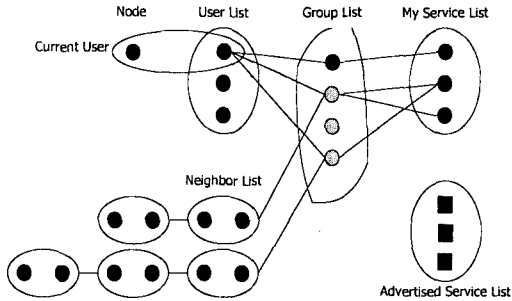


그림2 Management Information Base

3-3. 프로토콜

앞에서 언급한 각 관리자는 서비스 관리 프로토콜(Service Management Protocol-SMP), 그룹 관리 프로토콜(Group Management Protocol-GMP), 사용자 관리 프로토콜(User Management Protocol-UMP), 전송 관리 프로토콜(Transfer Management Protocol-TMP) 등의 프로토콜에 따라 서비스 브라우저가 제공하는 기능을 수행한다.

Service Management Protocol (SMP): 서비스를 관리하기 위한 규약으로 서비스 등록, 서비스 검색 등을 담당한다. 서비스는 사용자가 속한 그룹 중에서 원하는 그룹들에게만 공개된다. 사용자가 새로운 서비스를 등록하면, 서비스 목록에 추가하고 그룹 목록에서 사용자 서비스를 공개하기를 원하는 그룹과 서비스를 매핑한다. 사용자가 서비스를 요청하면, 먼저 로컬의 서비스 목록에서 조건에 맞는 서비스 설명서를 찾아서 응답한다. 앞에서 언급했던 것처럼 우리는 풀 모델을 가정했으므로, 로컬에서 적절한 서비스를 찾지 못할 경우 서비스 요청 메시지를 브로드캐스트한다. 외부로부터 서비스 요청 메시지를 받은 노드는 마찬가지로 로컬의 MIB를 찾아보고 있으면 결과를 보내주고, 없으면 요청을 다시 포워딩한다. 풀 모델이 아닌 푸쉬 모델이나 하이브리드 모델 등의 다른 서비스 검색 모델을 사용하기 위해서는 전체 구조를 바꿀 필요 없이 SMP만 각 모델에 맞게 수정하면 된다.

User Management Protocol (UMP): 사용자를 관리하기 위한 규약으로 새로운 사용자의 등록 및 탈퇴, 사용자 정보 관리, 망에서 임의의 노드 찾기 등의 기능을 담당한다. 사용자 등록은 유저 인터페이스에서 새로운 사용자의 정보를 받아서 사용자 목록에 추가하고, 탈퇴 시에는 사용자 목록에서 해당 엔트리를 제거하고, 그룹 정보에서 사용자를 삭제한다. 사용자가 임의의 사용자 아이디를 이용하여 사용자를 찾고자 할 때에는 망 전체에 사용자 검색 요청을 브로드캐스트한다. 사용자 검색 요청을 받은 외부 노드의 UMP는 자신의 사용자 목록을 찾아보고 있으면 응답하고 없으면 요청을 포워딩한다.

Group Management Protocol (GMP): 브라우저에 등록된

사용자들이 속한 그룹의 정보를 관리하는 규약으로 MIB의 그룹 목록과 함께 이웃 노드 목록을 이용하여, 새로운 그룹의 등록 및 탈퇴, 기존 그룹에의 참여, 그룹에 속한 노드 보여주는 기능을 수행한다. 어떤 노드가 새로운 그룹을 생성하면, 그룹 목록에 추가하고, 현재 사용자를 해당 그룹의 이웃 노드 목록에 추가한다. 노드 A가 그룹G에 참여할 때에는, 그룹 이름 G와 비밀번호를 담아서 그룹 조인 요청 메시지를 브로드캐스트한다. 이 메시지를 받은 외부 노드 B는 그룹 목록에 G가 있으면 비밀번호 확인 후, G의 이웃 노드 목록에 A를 추가하고 자신이 가지고 있는 G의 이웃 노드 목록을 보내준다. A 노드가 응답을 받으면, G를 그룹 목록에 추가하고 B로부터 받은 이웃 노드 목록을 연결한다. 그룹에서 탈퇴할 때에는, 그룹 목록 중 해당 그룹에 연결된 이웃 노드 목록에서 해당 사용자를 삭제하는데, 이 때 이웃 노드 목록이 비게 되면 그룹 이름도 그룹 목록에서 삭제한다. 사용자가 특정 그룹에 속한 노드를 보고자 할 때에는, 해당 그룹을 그룹 목록에서 찾고, 그 그룹에 연결된 이웃 노드 목록을 보여준다.

Transfer Management Protocol (TMP): 두 노드 사이의 실제 서비스의 전송에 관한 규약으로 서비스를 주고 받는데 이용된다. 외부 노드가 서비스 다운로드를 요청하는 경우, TMP를 통하여 해당 서비스를 로컬 메모리에서 찾아서 보내준다. 서비스를 다운로드 받은 노드는 서비스를 SMP를 통해 서비스를 등록한다.

4. 결론 및 향후 과제

이동 적응망은 기존 유선 망과는 달리 노드들의 이동이 빈번하기 때문에 중앙 집중적인 서비스 검색 모델은 적합하지 않다. 또한 아직 이동 적응망에서 활용되는 응용프로그램도 많이 제안되지 않았다. 서비스 브라우저는 이동 적응망에 있는 서비스들을 쉽게 찾을 수 있는 응용프로그램으로서, 이동 적응망에서 없어서는 안될 응용프로그램 중에 하나이다.

본 논문은 이동 적응망에 있는 디바이스에 올려질 수 있는 서비스 브라우저의 구조와 프로토콜을 제시하였고, 현재 서비스 브라우저는 구현 과정 중에 있다. 논문에서 제시한 서비스 브라우저는 간단한 서비스 검색 모델을 사용하고 있는데, 향후에는 좀 더 빠르고 정확한 검색을 위하여 이동 적응망 환경에 적합한 효율적인 서비스 검색 메커니즘을 개발할 것이다.

5. 참고 문헌

- [1] Arnold, Wollrath, et al. The Jini Specification. Reading, MA, USA: Addison-Wesley, 1999.
- [2] UPnP White Paper. <http://upnp.org/resources.htm>.
- [3] Service Location Protocol Version 2, Internet Engineering Task Force(IETF), RFC 2608, June 1999.
- [4] Dipanjan Chakraborty, Anupam Joshi, Yelena Yesha, Tim Finin. GSD: A Novel Group-based Service Discovery Protocol for MANETS. *Proceedings of MWCN*, 2002.
- [5] Michael Klein, Birgitta KonigRies. Multi-layer Clusters in Ad-hoc Networks - An Approach to Service Discovery. *Proceedings of International Workshop on Peer-to-Peer Computing*, 2002.
- [6] Eivind Jasund, Christian Bettstette, Christian Schwingschlogl. A Service Browser for the Service Location Protocol Version2. *Proceedings of the IFIP workshop on IP and ATM Traffic Management*, Sept. 2001.