

유비쿼터스 컴퓨팅을 위한 이동 애드혹 네트워크의 효율적인 리소스 발견 기법

(An Effective Resource Discovery in Mobile Ad-hoc Network for Ubiquitous Computing)

노 동 건[†] 신 현 식^{**}
(Donggeon Noh) (Heonshik Shin)

요 약 유비쿼터스 컴퓨팅 환경이 점차 확산되면서, 그 안에서 제공되는 다양한 리소스에 대한 효율적인 발견 전략들이 그 중요성을 더하고 있다. 특히, 유비쿼터스 환경을 구성하는 다양한 네트워크 중에서, 이동 애드혹 네트워크의 효율적인 리소스 발견 전략이 근래에 주목받고 있는데, 이는 이 네트워크를 구성하는 무선 노드들의 높은 이동성과 이들이 소유한 자원들의 제약 사항 등 고려해야 할 요소들이 많기 때문이다. 이 논문에서는 이러한 이동 애드혹 네트워크를 위한 적용적이고 효율적인, 유비쿼터스 환경의 이기종 네트워크와 연동 가능한 리소스 발견 전략을 제시한다.

키워드 : 유비쿼터스 컴퓨팅, 이동 애드혹 네트워크, 리소스 발견, 하이브리드 네트워크

Abstract Rapid advances in ubiquitous computing and its pervasive influence over our society demand an efficient way to locate resources over the network. In mobile ad-hoc networks (MANETs) which is a special type of the sub-networks in ubiquitous environment, effective resource discovery is particularly important, due to their dynamics and the resource constraints on wireless nodes. In this paper, we propose an adaptive and efficient resource discovery strategy for MANETs. Our strategy also provides a solution to bridge different types of networks which coexist in ubiquitous environment.

Key words : ubiquitous computing, mobile ad-hoc network, resource discovery, hybrid network

1. 서 론

리소스란 '사람이나, 소프트웨어 프로그램 또는 다른 리소스에 의해서 사용 되어질 수 있는 개체'를 일컫는다. 그것은 서비스 객체, CPU, 저장 공간, 통신 채널, 소프트웨어 필터, 데이터, 하드웨어 기기 또는 다른 사용자가 될 수도 있다. 이들 여러 가지 리소스들을 사용자가 자연스럽게 이용하기 위해 리소스 발견 기법이 필요한데, 이러한 기법은 네트워크 환경에서 자신이 제공하는 리소스 정보를 다른 노드들에게 인식 시키고 또한 자신이 필요한, 다른 노드에서 제공하는 리소스 정보를 찾아내어 그것을 사용하게 하는 방법을 제공한다.

컴퓨터 네트워크와 그와 관련한 응용들이 획기적으로

발전하여 일상적인 삶에서 매우 중요한 위치를 차지하면서, 사용자들은 전체 네트워크에서 자신이 필요한 리소스를 찾는 효율적인 발견 전략을 필요로 하게 되었다. 특히, 다양하고 복잡한 다수의 리소스가 동시에 제공되는 유비쿼터스 환경에서 효율적인 리소스 발견 전략은 매우 필수적이고 중요한 요소라 할 수 있다.

그림 1에서 보듯이 유비쿼터스 환경은 다수의 이기종 네트워크들이 결합된 하이브리드 네트워크 형태로 구성되어 있다. 이들 망 구성 요소들 중에서 유선 네트워크를 위한, 또는 무선 랜이나 무선 홈 네트워크와 같은 기반 구조형 무선 네트워크를 위한 리소스 발견 전략은 이미 많은 선행 연구들이 존재하고, *Jini*, *SLP*, *UPnP* 와 같은 산업 표준으로 정착되기도 하였다. 따라서 유비쿼터스 환경에서의 리소스 발견에 대한 최근 이슈는 이동 애드혹 네트워크(mobile ad-hoc network, *MANET*)에 집중되고 있다. 특별히, *MANET* 환경은 매우 동적인 시간적 공간적 환경 변화, 즉 구성 노드들의 높은 이동성 및 빈번한 회선 장애(link failure)로 인한 잦은 네트워크 토폴로지 변화와 제한된 리소스를 가진 노드들, 중

[†] 학생회원 : 서울대학교 컴퓨터공학부
dgnoh@mobisys.snu.ac.kr

^{**} 종신회원 : 서울대학교 컴퓨터공학부 교수
shinhs@snu.ac.kr
논문접수 : 2006년 5월 24일
심사완료 : 2006년 8월 17일

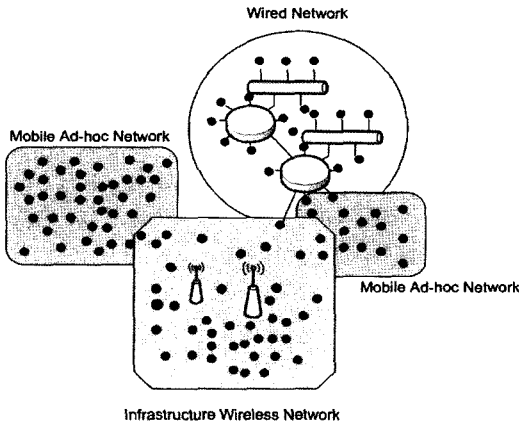


그림 1 유비쿼터스 컴퓨팅 환경을 구성하는 하이브리드 네트워크

양 집중형 관리의 부재, 비통제형(unauthorized) 네트워크 등의 특징으로 인해 구조화된(well-structured) 네트워크를 기반으로 하는 기존의 리소스 발견 전략들이 적용되기에는 적당치 않은 특징을 갖고 있고, 따라서 다른 각도의 새로운 접근 방식이 필요하다.

유비쿼터스 환경의 한 구성 망으로서 MANET을 위한 리소스 발견 전략이 가져야 할 주요 속성들을 나열하면 아래와 같다.

- 리소스가 제한된 무선 이동 디바이스들이 오버헤드(트래픽 또는 계산량)와 지연시간을 최소화 하면서 동적으로 리소스를 찾을 수 있어야 한다.
- 잠재적인 참여 노드 수 증가에 대비하여 수 백 개의 노드로 구성된 큰 규모의 네트워크에서 동작 가능하도록 확장 가능해야 한다.
- 하드웨어와 소프트웨어 플랫폼에 상관없는 장치 독립적인 인터페이스를 제공해야 하고, 사용자의 개입이 최소화되어야 한다.
- 이 기존 망들이 혼재된 유비쿼터스 환경과의 상호 리소스 발견 연동을 위하여 브리징(bridging) 기능을 제공하여야 한다.

위의 요구 조건들을 만족시키기 위해 우리는 RADIZ(resource advertisement and discovery with independent zones)라는 유비쿼터스 환경에서 MANET을 위한 새로운 리소스 발견 프로토콜을 설계하였다.

이 논문의 구성은 다음과 같다. 2장에서는 지금까지 제시된 MANET에서의 리소스 발견 전략에 대해 살펴보고, 3장에서는 우리가 설계한 새로운 유비쿼터스 컴퓨팅을 위한 MANET의 리소스 발견 전략인 RADIZ의 특징을 분석한다. RADIZ의 성능 분석을 위한 시뮬레이션 모델과 실험 결과에 대해 분석이 4장에서 이어지고, 마지막으로 5장에서 결론 및 향후 과제로 마무리한다.

2. 관련연구

일반적으로 MANET에서는 디렉터리 모델에 비해 무디렉터리 모델이 더 적합하다고 여겨지는데, 이는 그것이 완전히 분산적인 모델이고, 리소스 브로커(디렉터리)와 같은 기반 구조의 필요성이 전혀 없기 때문이다.

이러한 무 디렉터리 리소스 발견 전략은 초기에는 애드혹 라우팅 계층과 독립적인 미들웨어 계층으로 구현되었다[1,2]. 따라서 리소스의 발견과 라우팅이 동시에 이루어 지지 못하고 개별적으로 이루어지게 된다. 즉, 먼저 자신이 원하는 리소스를 찾기 위해 질의 메시지를 보내어 원하는 리소스 정보를 얻고, 그 리소스 서버로의 연결을 위해 네트워크 계층에서 다시 라우팅 컨트롤 메시지를 보내게 된다. 이 때, 리소스 발견 프로토콜과 라우팅 프로토콜에서 공통적으로 보이는 비효율적인 플러딩(flooding) 메커니즘으로 인해서 불필요한 중복적인 네트워크 오버헤드가 발생하게 된다. 또한 이러한 독립적인 계층 구성은 이식성은 뛰어나지만, 구현이 복잡하고, 모듈의 사이즈가 크게(heavyweight) 된다.

이러한 문제점을 해결하기 위해 라우팅 프로토콜과 결합된 리소스 발견 기법이 나오게 되었다. 이 방법은 라우팅 패킷과 리소스 발견 패킷을 통합하여 비효율적인 플러딩의 중복을 피한다. 리소스 발견 기법을 프로액티브(proactive) 방식의 라우팅 프로토콜과 통합시킨 연구[3]와 리액티브(reactive) 방식의 라우팅 프로토콜과 통합시킨 연구[4-6]가 수행되었고, 간단한 하이브리드 라우팅 프로토콜과 연동시킨 연구[7,8]도 진행되었다.

그러나 현재의 무 디렉터리 방식의 리소스 발견 전략은 아래와 같은 약점이 있으며, 우리는 이러한 도전 과제를 해결하고자 RADIZ를 제안하게 되었다.

- 동적인 네트워크 변화(예를 들면, 이동성이나, 리소스의 인기도 등)에 대한 적응력 부족
- 리소스 발견 알고리즘의 효율성 부족
- 큰 규모의 네트워크로의 확장성 부족
- 이 기존 망과의 연동성 부족

3. RADIZ

RADIZ(resource advertisement and discovery with independent zone)는 효율적인 라우팅 프로토콜 [9]과 연동된 적응형 하이브리드 리소스 발견 기법이다. 그림 2는 다른 발견 기법과 구분되는 RADIZ의 전체적인 특징들을 보여주고 있다. 각 노드들은 네트워크 상태와 리소스의 인기도, 남아있는 파워 양 등을 고려하여 자동적(autonomic)이고 동적(dynamic)으로 적응함으로써 자신의 영역 크기를 조절한다. 이렇게 조절된 각 노드마다의 다양한 영역을 기반으로 푸시 기반 리소스 발

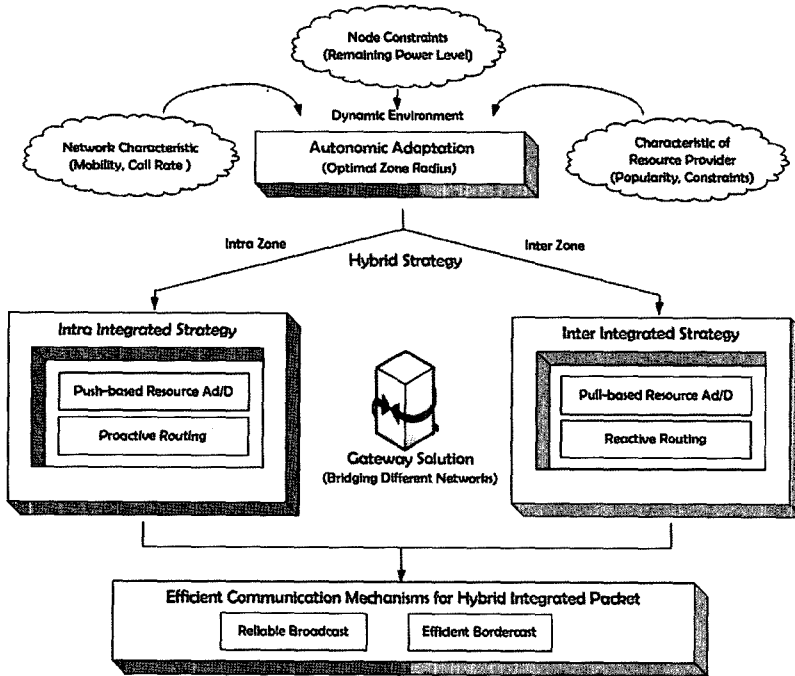


그림 2 RADIZ 기법의 개요

견 전략과 풀 기반 리소스 발견 전략을 결합한 하이브리드 전략을 사용하는데, 이 때 각각의 리소스 발견 전략은 신뢰성 있는 브로드캐스트(broadcast) 기법과 효율적인 보더캐스트(bordercast)기법을 사용하여 구현된다. 또한, 경량화 된 구현을 가능하게 하고 불필요한 플러딩을 줄이기 위하여 리소스 발견 프로토콜이 네트워크 계층과 연동되었다. 또한 이 기종 망과의 연동을 위한 게이트웨이 브리징 기법을 제공한다.

3.1 네트워크 계층과 연동된 하이브리드 발견 기법

네트워크 계층의 라우팅 프로토콜과 미들웨어 계층의 리소스 발견 프로토콜은 공통적으로 플러딩 기법을 사용한다. 이 두 프로토콜의 반복적인 플러딩 기법은 그렇지 않아도 부족한 MANET 구성 노드의 자원에 심각한 영향을 주게 된다. 따라서 RADIZ는 이 두 프로토콜을 서로 연동함으로써, 즉 라우팅 제어 패킷에 리소스 관련 정보를 포함시킴으로써 원하는 리소스에 대한 정보와 그 리소스를 제공하는 서버에 대한 라우팅 정보를 동시에 얻을 수 있고, 따라서 소모되는 시스템 및 네트워크 오버헤드를 줄이고, 경량의 구현이 가능하게 된다.

또한 RADIZ는 자신의 영역 내에서는 푸시 기반 발견 기법을 사용하고, 영역들 사이에서는 풀 기반 기법을 사용한다. 이것은 리소스 발견 프로토콜을 하이브리드 라우팅 프로토콜과 연동시킴으로서 가능하다.

그림 3은 RADIZ에서 라우팅 프로토콜과 리소스 발

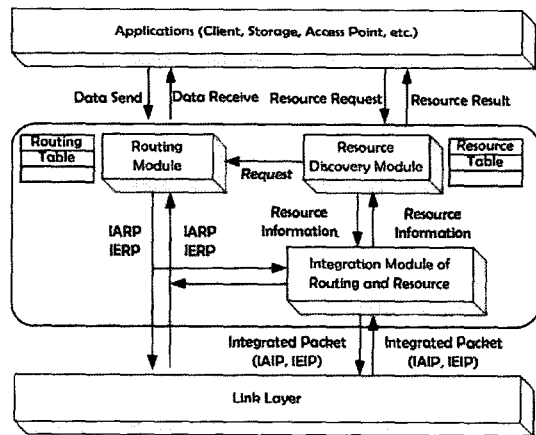


그림 3 네트워크 계층과 연동된 하이브리드 리소스 발견 전략 구조도

견 프로토콜이 어떻게 서로 연결되어 하이브리드 연동 프로토콜이 구현되는지에 대한 구조도를 보여주고 있다. 이 연동 프로토콜은 리소스 발견이 아닌 순수하게 라우팅 목적으로만 독립적으로도 수행될 수 있도록 설계되었다.

이 그림에서 나타나는 IAIP(intra integrated protocol) 패킷은 IARP(intra routing protocol) 패킷에 리소스 관련 정보를 추가시킨 확장 패킷을 의미하고, IEIP

(inter integrated protocol) 패킷 역시 *IERP*(intra routing protocol) 패킷을 확장한 것을 일컫는다. 주지할 사항은, 여기에 추가되는 리소스 관련 정보의 형식과 언어는 *RADIZ*가 적용어질 서비스 아키텍처에 따라 달라져야 한다는 것이다. 예를 들어, 타깃(target)이 서비스 지향적(service-oriented)인 웹 기반의 서비스라면 리소스 정보는 *WSDL*(web services description language)의 형식으로 표현 될 수 있을 것이다. 본 논문은 리소스 발견 기법에 초점을 두고 있고, 리소스 혹은 서비스 레벨이나 기기 레벨의 호환성(interoperability)에 목적을 두고 있지 않기 때문에, 리소스 정보를 간단한 형식으로 표현하였다.

3.2 자율적으로 적응하는 노드의 영역 변경

*RADIZ*에서는 각 노드의 간단한 알고리즘의 수행을 통해 동적으로 현재의 환경에 적응하여 최적의 하이브리드 상태가 유지한다. 즉, 각 노드는 독립적으로 현재 네트워크 환경과 파워 양에 최적화된 자신의 영역 반경을 결정하고, 또한 자신이 리소스를 제공하고 있다면 그 리소스의 인기도를 고려하여 영역 반경을 결정한다.

3.1장에서 설명한 것처럼, *RADIZ*는 네트워크 계층과 연동되어 있기 때문에 노드의 영역이라는 것은 리소스 발견의 측면에서는 푸시 기반 발견 영역을 의미하고, 라우팅의 측면에서는 프로액티브 라우팅 영역을 의미한다.

3.2.1 노드의 영역 반경 결정 알고리즘

*RADIZ*에서는 영역 반경 결정 방법으로서 최소 점 찾기(min searching, *MS*) 알고리즘과 적응형 트래픽 추정(adaptive traffic estimation, *ATE*) 알고리즘을 결합한 하이브리드 알고리즘[10]을, 연동된 리소스 발견 프로토콜에 맞게 수정 및 확장하여 사용한다.

푸시 기반 리소스 발견 트래픽과 프로액티브 라우팅 트래픽은 영역 반경에 대해 비 감소(non-decreasing) 함수이고, 반대로 풀 기반 리소스 발견 트래픽과 리액티브 라우팅 트래픽은 영역 반경에 대해 비 증가(non-increasing) 함수이다. 따라서 전체 트래픽은 이 두 함수의 합, 즉 볼록 형(convex) 함수가 된다. 그림 4는 영역 반경에 따라 *IAIP*와 *IEIP* 트래픽이 변화하는 모양을 보여주고 있는데, 이 그림에서는 영역 반경이 3 홉(hop)일 때에 그 노드를 통과하는 트래픽의 양이 최소가 된다. *MS* 알고리즘은 이렇게 전체 연동 패킷 트래픽이 최소가 되는 영역 반경을 찾기 위한 알고리즘이다.

그림 4의 화살표로 표시된 것처럼, 이 알고리즘은 각각의 노드에서 반복적으로 영역 반경을 1 홉씩 증가 혹은 감소시키면서 연동 패킷 트래픽의 최소 점을 찾아낸다. 즉, 정해진 시간 주기 내에서 그 노드를 통과하는 연동 컨트롤 패킷의 트래픽 양을 측정 하고, 이것이 이전 주기의 트래픽 양보다 작아졌다면 이전 주기에 수행

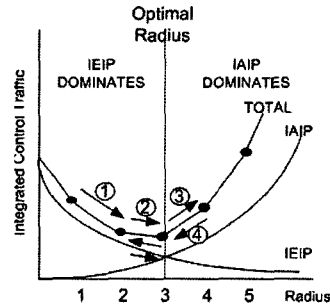


그림 4 최소 점 찾기 알고리즘

했던 것과 동일하게 영역 반경을 1 홉만큼 조절하고, 반대로 이전 주기의 트래픽의 양보다 커졌다면 이전 주기에 수행했던 것과 반대로 영역의 크기를 조절한다. 주의해야 할 것은 *MS* 알고리즘은 그것이 동작하는 동안 네트워크의 상태가 안정되어 있을 때 효율적으로 최소 점을 찾을 수 있다는 것이다. 즉, 네트워크 상태가 자주 변화하는 환경에서 *MS* 알고리즘으로 최소 점을 찾는 것은 매우 비효율적일 수 있다. 따라서 *MS* 알고리즘을 적용할 때에는 시간 주기의 선택이 중요한데, 그것이 너무 작다면 측정되어 지는 트래픽의 의미가 없어지고, 너무 크다면 네트워크 환경이 바뀔 가능성이 높아지기 때문이다.

ATE 기법은 각 노드에서 각각 초기에 한번 수행한 *MS* 알고리즘에 의해 얻어낸 영역 반경을 기준으로, 이후에 지속적으로 간단하게 영역 반경을 조절하는 알고리즘이다. *ATE*는 우선 *MS*로 알아낸 최적 영역 반경에서의 *IAIP* 트래픽에 대한 *IEIP* 트래픽의 비율을 Γ_{thres} 값으로 세팅한다. 그리고 주기적으로 $\Gamma(R)$ 과 Γ_{thres} 의 값을 비교하여 영역 반경을 조절한다. $\Gamma(R)$ 은 노드의 영역 반경이 R일 때 그 노드를 통과하는 *IAIP* 트래픽에 대한 *IEIP* 트래픽의 비율이다. 만약 $\Gamma(R) < \Gamma_{thres}$ 이면 그림 4의 오른쪽 영역으로서 *IAIP* 우위(domination) 상태를 의미하므로 영역 반경을 1홉 줄이게 되고, 그 반대라면 *IEIP* 우위 상태이므로 영역 반경을 1홉 늘리게 된다. 그러나 너무 자주 영역 반경을 변화시키는 것은 네트워크의 불안정을 유발할 수 있으므로 지연 유도 기법(delayed triggering mechanism)을 사용하여 이를 방지하는데, $\delta(\delta \geq 1)$ 가 유도(hysteresis) 인자로 사용된다. 따라서 $\Gamma(R) > \Gamma_{thres} \times \delta$ 이면 영역 반경을 1만큼 늘이고, $\Gamma(R) < \Gamma_{thres} / \delta$ 이면 영역 반경을 1만큼 줄인다.

ATE 알고리즘이 리소스 서버에서 돌아가고 있다면, 네트워크의 상태뿐만 아니라 리소스의 인기도도 고려되어야 한다. 따라서 리소스 서버는 주기적으로 그 리소스

의 호출 빈도를 모니터해야 한다. 만약 P_{new} (현재 주기 동안 호출된 빈도)가 P_{old} (지난 주기 동안 호출된 빈도) 보다 크다면, 영역 반경을 1 홉 증가시키고, 반대의 경우는 감소시킨다. 이 경우에도 마찬가지로 너무 잦은 영역 반경 변화를 방지하기 위해 $\epsilon(\epsilon \geq 1)$ 인자를 사용한 지연 유도 기법이 사용된다. 주목해야 할 것은 리소스 서버 노드나 리소스 사용자 노드나 동일한 ATE 알고리즘이 동작하고 있다는 것이다. 단지 리소스 사용자의 측면에서는 인기도와 관련된 파라미터가 변하지 않기 때문에 의미가 없게 된다.

만약 노드의 남은 파워양이 특정 임계치(threshold) 미만이라면 영역 결정 알고리즘의 수행을 중단하고, 자신의 영역 크기를 0으로 고정한다. 그리고 정해진 방식으로 절전(power down) 모드로 진입하여 자신을 지나는 패킷들이 없게 하고, 꼭 필요한 리소스가 있을 때에만 깨어서 풀 기반 리소스 발견 기법을 수행하여 찾는다.

3.3 푸시 기반 리소스 발견

RADIZ에서 리소스 서버는 자신의 동적인 영역 내에서 푸시 기반 리소스 발견 전략을 수행한다. 즉, 리소스 서버는 주기적으로 자신의 영역 내의 모든 노드들에게 자신의 리소스를 광고(advertisement)하는 브로드캐스트 메시지를 보내고, 영역 내의 모든 노드들은 수동적으로 그 광고 메시지를 받아 리소스에 대한 정보를 저장하고 유지한다. 또한 푸시 기반 광고 메시지를 받은 노드들은 자신이 제공하는 리소스가 있다면 이에 대한 정보를 담은 메시지를 생성하여 소스 노드에게로 다시 보낸다(backtracking). 물론 이 메시지는 연동 메시지가므로 리소스 정보와 함께 라우팅 정보도 포함되어 있다.

3.3.1 푸시 기반 리소스 발견을 위한 메시지 형식

연동된 푸시 기반 리소스 발견을 위해서는 IAIP 패킷과 IAIP_REP 패킷이 필요한데, 이 중 IAIP 패킷의 형식이 그림 5에 나타나 있다. 그림에서 보듯이, IAIP

패킷은 두 개의 부분으로 구성되어 있는데, 하나는 IARP에서 사용되는 라우팅 컨트롤 정보 부분이고, 다른 하나는 리소스에 대한 정보 부분이다. 이 리소스 정보 부분은 리소스 타입, 리소스 존속 기간(lifetime), 부가 정보 등이 포함되어 있다. 다시 강조하지만, 본 논문은 서비스 레벨의 호환성이 목적이 아니고, 리소스 발견 기법에 대한 연구이므로, 웹 서비스 등의 특정 서비스 모델을 정하지 않고, 간단한 리소스 정보 형식을 사용하였다.

리소스 타입은 네트워크 내의 모든 노드들이 알 수 있도록 미리 정의 되어 있다고 가정한다. 또한 리소스의 갱신 주기를 정하여 존속 기간 필드를 통해 전달하는데, IAIP 패킷을 받은 노드는 이 존속 기간이 지나도록 다시 그 리소스에 대한 IAIP 패킷이 도착하지 않으면 그 리소스에 대한 정보를 무효화(invalidation) 한다. 따라서 리소스 서버가 임의로 옮겨지거나 꺼지더라도 자동으로(autonomous) 그 리소스의 부재를 인지할 수 있다. 부가 정보에는 리소스 인터페이스나 QoS 속성이 포함된다.

IAIP_REP 패킷 형식은 IAIP 패킷 형식과 같으나 라우팅 정보 부분이 IARP_REP 형식으로 대체된다.

3.3.2 푸시 기반 영역의 트래픽 분석

각 노드의 영역 내의 트래픽은, 자신의 리소스를 광고하기 위한 푸시 기반 발견 전략의 트래픽과, 프로젝트브 라우팅을 위해 영역 내의 노드에 대한 라우팅 정보를 업데이트하는 트래픽, 두 가지로 구성된다. 그러나 RADIZ에서는 리소스 발견과 라우팅을 연동하였기 때문에 라우팅 정보를 업데이트 하는 동시에 자신의 리소스를 홍보할 수 있다. 따라서 RADIZ의 영역 내에서의 전체 트래픽은 라우팅 업데이트 트래픽과 같고, 리소스 발견 기법을 위한 추가적인 트래픽은 발생하지 않는다.

업데이트 트래픽은 시간에 따라(time driven) 주기적으로 발생할 수도 있고, 이웃 노드와의 연결성(con-

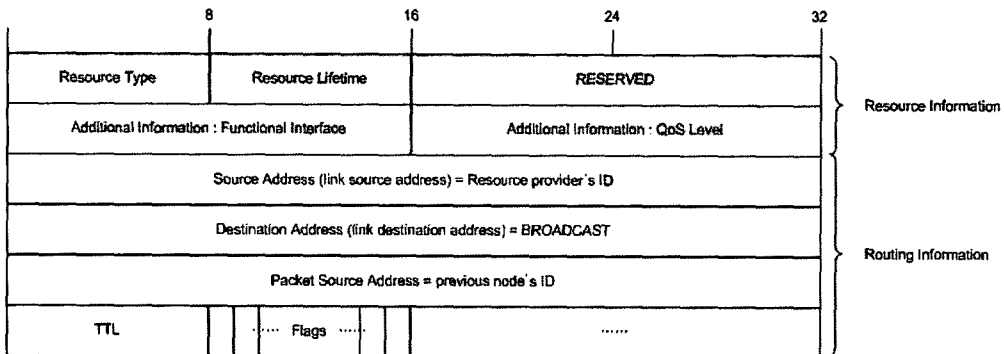


그림 5 푸시 기반 리소스 발견을 위한 IAIP 패킷 형식

nectivity)이 바뀔 경우처럼 사건에 기반(event driven)하여 발생할 수도 있다. 회선 비용이 일정하다고 가정하면, IAIP 트래픽은 아래와 같다.

$$C_{IAIP}[\text{traffic/node/s}] = T_{\text{time_driven}} + T_{\text{event_driven}}$$

$$= F_{\text{update}} \times UT_{IAIP}/N_{\text{zone}}(\rho, \sigma)$$

$$+ v \times UT_{IAIP}/N_{\text{zone}}(\rho, \sigma)$$

여기서 $T_{\text{time_driven}}$ 은 주기적인 트래픽을 뜻하고, $T_{\text{event_driven}}$ 은 사건 기반 트래픽이다. F_{update} 는 업데이트 빈도[1/초], UT_{IAIP} 는 IAIP 패킷 하나에서 비롯되는 업데이트 트래픽, ρ 는 영역 반경[홉], σ 는 노드의 밀집도 [노드 당 이웃 노드의 수], $N_{\text{zone}}(\rho, \sigma)$ 는 영역 내의 노드의 수, 그리고 v 는 노드의 속도[노드가 이동하면서 초당 새로 만나는 이웃 노드의 수] 이다.

주의해서 볼 사항은 IAIP 트래픽의 양이 전체 노드의 개수에 따라 달라지는 것이 아니라, 속도와 업데이트 주기에 의존적이라는 것이다. 따라서 이동성이 높고, 회선 장애가 많은 네트워크에서는 영역 반경을 작게 유지하는 것이 유리하다. 또한, 리소스의 인기도가 낮은 경우에도 업데이트 주기를 낮추거나, 영역 반경을 작게 하는 것이 효율적일 것이다.

3.4 풀 기반 리소스 발견

노드가 찾고자 하는 리소스가 있으나, 현재 자신의 리소스 테이블에 자신의 요구 조건을 만족시키는 리소스가 존재하지 않을 때, 즉 푸시 기반 리소스 발견 기법으로 알게 된 리소스들 중에 해당 리소스가 존재하지 않을 경우에, 자신이 적극적으로 풀 기반의 리소스 발견 전략을 사용하여 리소스 질의 메시지를 보내게 된다.

3.4.1 효율적인 보더캐스트(bordercast) 기법

RADIZ는 풀 기반 리소스 발견 전략으로서 BRP(bordercast resolution protocol)[11]를 사용한다. BRP는 자신의 영역 내의 노드들 중에서 특정한 요건을 갖춘 몇몇의 노드들에게만 질의 메시지를 라우팅하는 기법을 제공하고, 결과적으로 그 질의를 소스 노드로부터 최대한 멀리까지 효율적으로 보내는 기능을 수행한다. 또한 질의 중복(overlap)을 막기 위하여 중복 질의 발견 기법도 제공한다. 여기서 말하는 특정한 요건을 갖춘 노드들은 리보더캐스팅(rebordercasting) 노드라고 일컫는데, 소스 노드 영역 내의 노드들 중에서 자신의 영역이 소스 노드 영역을 벗어나는 새로운 영역을 포함하고 있으며, 질의 중복이 되지 않은 노드를 의미한다.

RADIZ는 각 노드마다 영역 반경이 매우 다양하므로, 한 노드의 영역 내에 다른 노드의 영역이 완전히 포함되는 경우가 발생한다. 이런 경우, 포함하는 노드의 입장에서는 포함되는 노드에게서 얻을 수 있는 새로운 라우팅 정보나 리소스 정보가 하나도 없으므로, 이러한

노드에게로 질의를 전달하는 것은 아무런 의미가 없다. 따라서 BRP는 이러한 상황을 피하여 리보더캐스팅 노드들을 선택하여 그 노드들에게만 질의 메시지를 유니캐스트(unicast) 한다.

그림 6은 영역 반경이 3홉인 소스 노드에 의해서 수행되는 BRP의 예를 보여주고 있다. 우선 BRP가 시작되면 소스 노드로부터 각 노드의 최 외곽 노드들까지의 트리가 만들어 진다. 그 후에, 각 경로상의 노드들에 대한 영역 크기와 질의 중복 발견 기법을 이용하여 리보더캐스팅 노드를 선택하게 된다. 이 그림에서는 노드 B와 노드 D가 선택된다. 노드 A와 노드 C는 자신의 영역이 소스 노드의 영역에 완전히 포함되므로 질의 메시지를 보낼 대상에서 제외된다. 리보더캐스팅 노드를 선택한 후 소스 노드는 이들 노드 B와 노드 D에게 질의 메시지를 유니캐스트하고, 노드 B와 노드 D는 이에 대한 질의 처리(query processing)를 하게 된다. 즉, 해당 질의를 만족 시키는 리소스가 있으면 응답 메시지를 보내고, 없으면 다시 질의 메시지를 리보더캐스팅 한다.

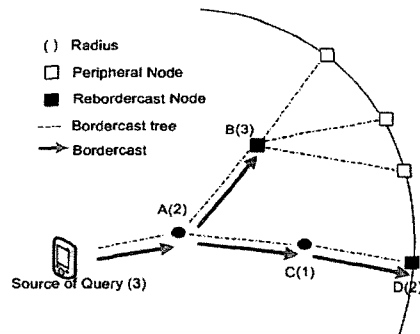


그림 6 BRP(bordercast resolution protocol)의 예

3.4.2 풀 기반 리소스 발견의 위한 메시지 형식

풀 기반 리소스 발견을 위하여 RQM(resource query message)과 RRM(resource reply message)이 필요하다. 리액티브 라우팅 요청 패킷에 리소스 정보를 포함한 패킷을 RQM으로 정의하였고, 리액티브 라우팅 응답 패킷에 리소스 정보를 포함한 패킷을 RRM으로 정의하였다. RQM의 포맷은 그림 5의 리소스 정보 부분에 리소스 서버 주소 필드가 추가되고, 라우팅 컨트롤 부분에는 IERP 패킷이 들어간 형식이다. 리소스 정보 부분의 존속 기간(lifetime)과 리소스 서버의 주소 필드는 초기에는 아무 값도 포함하고 있지 않다가, 해당 리소스를 찾은 후에 RRM이 최종적으로 만들어지기 전까지 임시로 해당 값들이 채워진다. 리소스 타입과 부가 정보 필드는 사용자가 찾고자 하는 리소스의 정보들로 초기화한다. 라우팅 정보 부분의 목적지 주소 필드는 BRP에

의해서 정해진 리보더캐스팅 노드의 주소를 포함하고 있다.

*RRM*의 형식은 *RQM*과 같지만, 리소스 정보 부분에는 *RQM*이 찾은 리소스의 정보가 포함되어 있다. *RQM*이 찾은 리소스 정보와 그 리소스 서버의 라우팅 정보를 가진 노드에서 *RRM*을 생성하고, 그 후에, *RRM*은 *RQM*을 받은 경로의 반대 경로를 따라서 (backtracking) 소스 노드까지 도달하게 된다.

3.4.3 영역들 간의 트래픽 분석 및 정확성 증명

각 노드의 영역들 사이의 트래픽은 두 가지 부분, 즉 풀 기반 리소스 발견을 위한 트래픽과 리액티브 라우팅을 위한 트래픽으로 나눌 수 있다. 따라서 아래의 수식과 같이 표현된다.

$$C_{\text{ondemand}}[\text{traffic/node/s}] = C_{\text{pullbased_rd}} + C_{\text{reactive_routing}}$$

$$= RQM_{\text{traffic}}[\text{traffic/query/node}] \times N_{rp} / MTNR$$

$$+ IERP_{\text{traffic}}[\text{traffic/query/node}] \times N_{\text{total}} / MSID$$

여기서 $C_{\text{pullbased_rd}}$ 는 풀 기반 리소스 발견에 의해서 유발되는 트래픽, $C_{\text{reactive_routing}}$ 은 리액티브 라우팅에 의해서 유발되는 트래픽, RQM_{traffic} 과 $IERP_{\text{traffic}}$ 은 각각 하나의 리소스 발견 질의와 라우팅 질의에 의해서 유발되는 각 노드의 트래픽을 의미한다. N_{rp} 는 리소스를 제공하는 서버의 개수, N_{total} 은 전체 네트워크의 노드 수, 그리고 $MTNR$ (mean time to next request)은 리소스 발견 질의들 사이의 평균 시간, $MSID$ (mean session interarrival delay)는 라우팅 질의 사이의 평균 지연 시간으로서 호출 속도(call rate) 값의 역수 값이다.

우리는 궁극적으로 $C_{\text{pullbased_rd}}$ 값을 줄이는 것, 즉 RQM_{traffic} 값을 줄이는 것이 목적이다. 리소스 발견 프로토콜과 라우팅 프로토콜이 각각 독립적으로 구현되었다면 각 계층에서 중복되는 플러딩 작업이 발생하기 때문에, 연동 되었을 경우 보다 이 값이 매우 크게 된다. 또한, *BRP*는 효율적인 질의 플러딩으로 이 값을 훨씬 작게 한다.

위의 수식은 회선 장애(link failure)로 인하여 리소스 질의의 라우팅이 실패하는 경우는 고려하지 않고 있다. 회선 장애로 인한 트래픽 오버헤드를 분석하려면 *NLFF*(normalized link failure frequency of node) 등 몇 가지 추가적인 인자가 고려되어야 하지만, 간단한 분석을 위하여 이를 생략하였다. 주목할 점은 영역들 간의 트래픽은 전체 노드의 개수에 의존적이라는 점과, 리소스를 제공하는 서버가 많을수록, 또 리소스의 인기가 높아질수록 커진다는 점이다.

3.5 유비쿼터스 환경을 위한 이 기종 망과의 연동

유비쿼터스 네트워크에는 유선 망 혹은 구조화된 무선 망, *MANET* 등 여러 가지 타입의 네트워크들이 공

존할 수 있고, 이들은 서로 다른 관리 도메인(administrative domain)에 의해서 운영 되어질 수도 있다. 유선 망과 무선 망을 연동하는 가장 쉬운 방법은 무선 *AP*(access point)를 사용하여 무선 노드들을 유선 네트워크에 연결하는 것이다. 그러나 *MANET* 환경과 같이 *AP*를 사용하지 않는 네트워크의 경우에는 여러 개의 네트워크 인터페이스를 가지고 있어서 서로 다른 관리 도메인을 연결(bridge)시켜주는 게이트웨이로서 동작하는 노드들을 필요로 한다. 즉, 게이트웨이 노드는 네트워크 인터페이스들 중 하나로 *MANET*의 구성 노드들과 통신 하고, 다른 인터페이스들로 유선 망이나 *AP*를 사용한 무선 망 등의 이 기종 망과 통신을 유지하면서, 이들을 서로 연결 시켜주는 역할을 한다. 유비쿼터스 환경에서 이러한 게이트웨이 노드들은 기반 환경으로서 곳곳에 내장되어 존재할 수 있다. 게이트웨이 노드에 의해서 서로 연결된 네트워크 들은 새로운 하이브리드 네트워크를 생성하게 된다.

비록 *MANET* 내의 일반 노드들이 게이트웨이 노드를 통하여 이 기종 네트워크들과 통신을 할 수 있다 하더라도, 리소스 발견 프로세스는 단지 *MANET*에서 사용가능한 리소스들로 제한되게 된다. 따라서 하이브리드 네트워크로 구성된 전체 유비쿼터스 네트워크에서 리소스 발견을 가능케 하려면 게이트웨이 노드가 아래와 같은 부가적인 *RADIZ* 연동(bridging) 리소스 발견 서비스를 수행하여야 한다.

- 연동 서비스는 *RADIZ* 프로토콜과 다른 망의 리소스 발견 프로토콜과의 인터페이스를 제공하여, 프로토콜 사이의 변환 기능을 제공한다. 이 같은 기능은 각 프로토콜의 메시지에 대한 간단한 해석과 메시지 간의 매핑을 통하여 이루어질 수 있다.

- 게이트웨이 노드는 이러한 연동 리소스 발견 서비스를 네트워크마다 적용되는 고유의 리소스 발견 프로토콜을 사용하여 각각의 연동될 네트워크에 광고한다.

예를 들어 그림 7과 같이 *AP*를 사용한 구조화된 무선 망에서 *SLP*(디렉터리 기반 발견 기법 중 하나)가 사용되고 있을 경우, 게이트웨이 노드는 연동 리소스 발견 서비스를 *SLP*의 디렉터리에 등록하고, 또한 푸시 기반 발견 기법을 사용하여 주변의 *MANET* 노드들에게 광고한다. *MANET* 노드가 이 서비스를 사용하여 연동된 망에서 리소스를 찾고자 할 경우, 먼저 *RADIZ* 프로토콜을 사용하여 연동 리소스 발견 서비스를 찾고, 이 서비스를 제공하는 게이트웨이 노드에게 해당 리소스 질의 메시지(*RQM*)를 보낸다. 그러면 게이트웨이 노드는 이를 해석하고 *SLP* 질의 형태로 변환하여 *SLP*를 사용한 리소스 발견을 수행하고, 다시 그 결과를 받아 *RRM*의 형태로 변환하여 소스 노드에게 보낸다. 반대

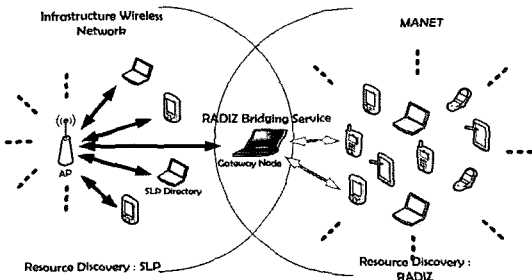


그림 7 이 기종 망 간의 리소스 발견 연동 서비스

로 구조화된 무선 망의 노드에서 MANET의 리소스를 찾고자 하는 경우 SLP 디렉터리에서 연동 리소스 발견 서비스를 찾아 그 정보에 맞는 형식으로 리소스 질의 메시지를 보낸다. 그러면 게이트웨이 노드는 이를 해석하여 RADIZ를 사용한 리소스 발견을 수행한다.

이러한 방법으로 MANET 내의 리소스들이 밖의 네트워크들에게 보여 질 수 있고, 반대의 경우도 가능하다. 유비쿼터스 하이브리드 망을 아우르는 연동 리소스 발견은, 구조화 된 인트라넷과 여러 개의 MANET이 연결된 엔터프라이즈 환경에서도 유용할 수 있다.

연동 서비스의 구현에서 중요한 것 중 하나가 권한의 문제인데, 리소스 사용에 관한 권한이나, 네트워크 접근에 관한 권한의 문제는 본 논문에서 다루지 않는다.

4. 성능 평가

우리는 RADIZ의 성능을 평가하기 위하여 코넬(Cornell) 대학의 NS2 모바일 확장 버전을 사용하여 시뮬레이션을 디자인 하였다. IEEE 802.11 MAC 프로토콜 위에서 푸시 기반 리소스 발견과 연동될 프로액티브 라우팅기법으로 DSDV[12]를 사용하였고, 풀 기반 리소스 발견과 연동될 리액티브 라우팅 기법으로 AODV[13]를 사용하였다.

4.1 시뮬레이션 모델

1000m×1000m의 사각형 영역에 무작위로 각각 50개, 100개, 150개, 200개의 노드를 위치시킨 네트워크를 생성하였다. 그 중 다섯 개의 리소스 서버 노드가 각각 다른 종류의 리소스를 제공하고 있고, 네트워크 내의 모든 노드들은 이들이 제공하는 리소스 타입에 대한 정보를 미리 알고 있으며, 남아있는 파워의 양은 모두 임계치 이상으로 가정하였다. 각각의 시뮬레이션은 500 초 동안 실행하였고, 같은 환경으로 30번씩 실행하였다.

트래픽 모델로서 우선 리소스 발견 트래픽 부하를 시뮬레이션 하기 위해서, 무작위로 선택된 노드가 하나의 리소스에 대한 리소스 질의 메시지를 보내게 하였는데, 각각의 리소스에 대한 질의를 사이의 간격 시간은 주어진 MTNR(1초, 10초)을 평균으로 하는 지수 분포

(exponential distribution)를 따르게 하였다. 또한 RADIZ가 라우팅과 연동된 리소스 발견 프로토콜이므로 라우팅 트래픽 부하를 시뮬레이션 하는 것도 필요하다. 따라서 한 세션에서 한 노드가 일정 수의 데이터 패킷을 무작위로 정해진 목적 노드에 보내도록 하였다. 한 세션에서 보내는 데이터 패킷의 양은 10 패킷을 평균으로 푸아송 분포를 따르고, 각 데이터 패킷의 크기는 1 Kbit, 전송 속도는 초당 16 패킷이다. 각 세션들 사이의 시간은 주어진 MSID(3초, 150초)를 평균으로 지수 분포를 따른다.

4.2 시뮬레이션 결과

RADIZ의 성능 평가를 위하여 우리는 다섯 가지의 서로 다른 리소스 발견 전략을 직접 구현하여 비교하였다. ZRP-RDP는 ZRP[14] 하이브리드 라우팅 프로토콜과 연동된 리소스 발견 전략이고, AODV-RDP는 AODV 라우팅 프로토콜과 연동된 리소스 발견 전략이다. 또한 네트워크 레이어와 연동되지 않은 푸시 기반 그리고 풀 기반 리소스 발견 전략을 구현하였다.

4.2.1 트래픽 오버헤드와 지연 시간 분석

그림 8은 이들 다섯 가지 전략과 RADIZ의 평균 트래픽과 지연시간을 비교한 결과를 보여준다. 이 실험에서는 단지 리소스 발견 트래픽만을 부하로 주어 테스트 하였으며, 라우팅 발견 트래픽은 모델링하지 않았다. 노드들의 평균 속도는 10m/s 이고 평균 정지 시간은 10 초, 그리고 각 리소스 서버의 MTNR은 1초로 하였다. 영역 크기의 지연 변화를 위한 δ 와 ϵ 의 값은 각각 10, 1.5로 하였다. 그림 8(a)에서 보듯이 라우팅 레이어와 연동하지 않은 풀 기반, 푸시 기반의 발견 기법은 상대적으로 큰 트래픽을 보이는 것을 알 수 있는데, 이로써 레이어 연동(cross-layer)의 효율성을 알 수 있다. RADIZ는 노드 50개의 작은 네트워크에서도 20%~65%의 리소스 발견 트래픽을 줄였고, 노드의 개수가 커질수록 이 차이는 매우 커졌다.

그림에서 알 수 있듯이, 영역 크기가 1 홉인 ZRP-RDP 전략의 성능이 RADIZ 다음으로 좋았는데, 이 방법은 모든 노드의 영역의 크기가 항상 1 홉으로 고정되어 있기 때문에 환경이 바뀌면 성능이 나빠질 가능성이 매우 크다. 모든 노드의 영역 크기가 3 홉으로 고정된 ZRP-RDP의 성능이 현재의 환경에서는 좋지 않음을 고려할 때, 현재 환경에서 각 노드의 최적의 하이브리드 상태는 평균 1 홉과 3 홉 사이라고 추정할 수 있는데, RADIZ에서의 리소스 제공 노드들의 평균 영역 크기는 2.21 홉, 모든 노드의 평균 영역 크기는 1.65 홉으로서 추정 값과 비슷함을 알 수 있다. 또한 노드의 개수에 따른 RADIZ 트래픽의 추이를 살펴보면 지수적으로 증가하지 않고 거의 일직선으로 일정하게 증가하는

것을 알 수 있다. 이는 RADIZ의 확장성을 보여주고 있으며, 이는 적응적 영역 크기 변화와 리소스 발견의 효율성으로 인한 것이다. 그림 8(b)는 RADIZ가 지연 시간 면에서도 가장 좋은 성능을 나타냄을 보여주고 있다.

표 1 변화하는 시뮬레이션 환경

		Before (0s~250s)	After (250s~500s)
Network Environment	Average Speed	15 m/s	0.5 m/s
	Pause Time	10 s	100 s
Resource Discovery Traffic	MTNR	10 s	1 s
Routing Traffic	MSID	150 s	3 s

4.2.2 환경에의 적응 능력 분석

RADIZ의 성능을 좀 더 현실적인 환경에서 실험하고자, 이번에는 리소스 발견 트래픽 부하와 라우팅 부하를 함께 적용하여 실험하여 보았고, 시뮬레이션 기간 도중에, 즉 시작 후 250초 후에 네트워크 환경을 변화시켜 보았다. 라우팅과 연동하지 않은 풀 기반 리소스 발견 기법에는 라우팅을 위하여 AODV 라우팅 기법을 같이 사용하였고, 푸시 기반 리소스 발견 기법에는 DSDV 라우팅 기법을 함께 사용하였다. 우리가 실험한 네트워크 환경과 트래픽 모델은 표 1에 나타나 있고, 50개 노드의 네트워크에서 환경이 변하기 전과 후에 각 전략의 트래픽을 분석하여 보았다.

그림 9는 그 결과를 나타낸다. 환경 변화 전에는 RADIZ와 영역 크기가 1로 고정된 ZRP-RDP, AODV-RDP가 상대적으로 좋은 성능을 보였는데, 이는 노드의 이동성이 높아 토폴로지가 빠르게 변하고 회선 장애의 가능성이 높은 상황에서 노드의 영역 크기가 커지면 영역을 유지하는 오버헤드가 매우 증가하기 때

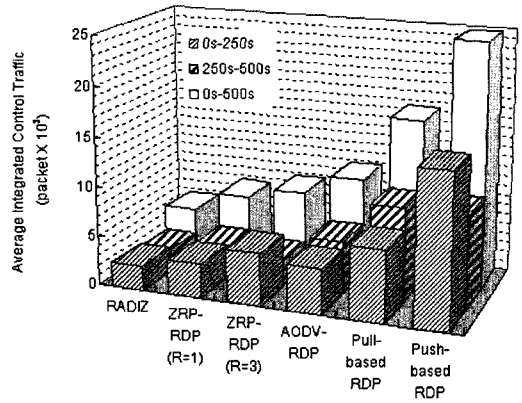
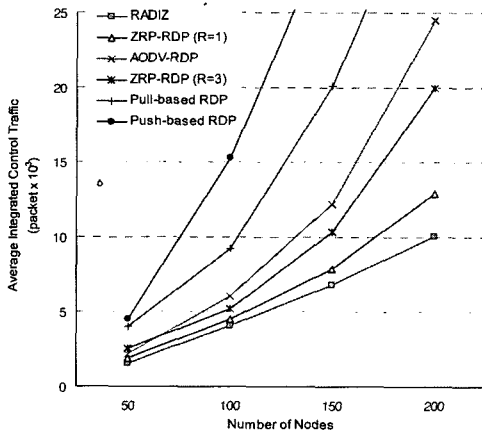


그림 9 환경 변화에 따른 연동 패킷 트래픽 비교

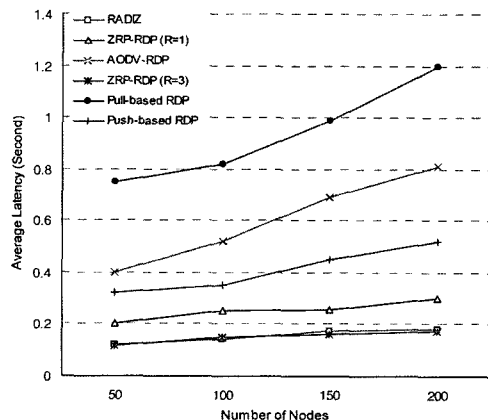
문이다. 더욱이 리소스가 자주 불리지 않기 때문에 푸시 기반 영역을 크게 하여 트래픽을 줄일 수가 없기 때문이다. 반대로 환경이 변한 후에는 RADIZ와 영역 크기가 3 홉으로 고정된 ZRP-RDP가 좋은 성능을 보였다. 이것은 상대적으로 안정된 네트워크와 리소스들이 매우 자주 호출되는 환경에서는 상대적으로 큰 영역을 유지하여 메시지의 플러딩을 최대한 방지하는 것이 더 유리하기 때문이다. 결과적으로 RADIZ는 이들 환경에 잘 적응하여 환경이 변하기 전과 변한 후에, 그리고 전체에 걸친 트래픽의 양에서 가장 좋은 결과를 보였다. 환경 변화 전에 리소스 제공 노드들의 평균 영역 크기는 2.16 홉, 모든 노드의 평균 영역 크기는 1.25 홉 이었고, 환경 변화 후에는 리소스 제공 노드들의 평균 영역 크기는 3.8 홉, 모든 노드의 평균 영역 크기는 2.91 홉 이었다.

4.2.3 이 기종 망과의 연동 능력 분석

마지막으로 우리는 이 기종 망과의 연동 테스트를 위



(a) 연동된 패킷 평균 트래픽



(b) 리소스 질의 평균 지연시간

그림 8 리소스 발견 트래픽 부하를 주었을 경우 RADIZ의 성능

하여 간단한 실험을 추가적으로 실시하였다. 디렉터리 기반 리소스 발견 기법을 사용하는 무선 인트라넷을 시뮬레이션 하기 위해 세 개의 노드를 생성하여, 한 노드는 디렉터리 노드로서 다섯 가지 새로운 리소스 정보들을 담고 있고, 한 노드는 리소스를 찾는 사용자 노드의 역할을, 마지막 노드는 RADIZ의 리소스 발견 연동 서비스를 수행시키는 게이트웨이 노드로 설정하였다.

간단한 실험을 위하여 디렉터리 기반 발견을 위한 질의 메시지와 리소스 정보는 SLP의 형식을 단순화 하여 설계하였다. MANET 노드의 크기는 50 노드이고, 평균 정지 시간은 2초, 평균 속도는 각각 1 m/s, 10 m/s, 20 m/s, 30 m/s 이다. 10개의 리소스(MANET에 5개, 디렉터리에 5개)에 대한 MTNR은 2초이고, 총 시뮬레이션 시간은 500초이다. 이러한 환경에서 두 이 기종 망 사이의 리소스 발견 성공률과 질의 지연 시간에 대한 실험 결과가 표 2에 나타나 있다.

표 2에서와 같이 리소스 발견 성공률이 95% 이상으로, 이는 연동하기 전의 동일 환경의 MANET에서의 성공률과 비슷하고, 지연 시간 역시 평균 0.125 초로 연동 전의 값과 비슷하다. 따라서 게이트웨이 브리징 서비스의 높은 정확성을 확인할 수 있다. 평균 속도가 커질수록 리소스 발견 성공률은 약간 낮아지는데, 이것은 노드 이동에 따른 회선 장애와 토폴로지 변화에서 기인한다.

이 실험에서는 Jini나 UPnP, SLP 등 다소 복잡한 디렉터리 기반 프로토콜을 사용하지는 않았지만, 두 망 사이의 간단한 브리징 시뮬레이션을 통하여 RADIZ의 연동 서비스를 통한 이 기종 망 간의 연결 가능성을 확인할 수 있다.

표 2 이 기종 연동 리소스 발견 실험 결과

Average Speed	Success Ratio	Average Latency
1 m/s	97.2 %	0.12 s
10 m/s	96.3 %	0.11 s
20 m/s	95.7 %	0.14 s
30 m/s	94.1 %	0.13 s

5. 결론

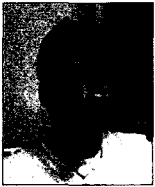
MANET의 특성들을 고려하여 설계한 RADIZ는 경량의 하이브리드 적응형 리소스 발견 전략으로서 네트워크 계층과 연동(cross-layer)되어 동작한다. 리소스 정보를 라우팅 컨트롤 패킷에 포함시켜 보냄으로서 중복되는 플러딩 동작을 줄이고, 푸시 기반 모델과 풀 기반 모델의 하이브리드 리소스 발견 기법을 사용함으로써 네트워크와 시스템 오버헤드를 줄였다. 아울러 각 노

드는 변화하는 환경(네트워크 특성 및 리소스의 인기도, 사용 가능한 파워 등)에 스스로 적응하여 영역 반경을 조정함으로써 최적의 하이브리드 상태를 유지할 수 있다. 이러한 RADIZ의 여러 가지 특성으로 인해 감소되는 네트워크 및 시스템 오버헤드는 적은 파워 소모와 네트워크 혼잡도, 적은 메모리와 프로세서의 사용으로 연결되는데, 이들은 MANET 환경에서 매우 필요한 요구 조건이다. 더불어 RADIZ는 유비쿼터스의 다양한 망들에서 동작하는 리소스 발견 전략들과 연동될 수 있도록 게이트웨이 솔루션을 제공함과 동시에 수백 개의 노드들로 이루어진 큰 규모의 MANET에서도 동작 가능한 확장성을 지녔는데, 이는 다른 발견 전략들에서는 찾아볼 수 없는 RADIZ 만의 특징이다.

참고 문헌

- [1] S. Helal, "Konark - a service discovery and delivery protocol for ad-hoc networks," in Proceedings of WCNC'03, New Orleans, USA, Mar. 2003.
- [2] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, and A. Schade, "DEAPspace: transient ad-hoc networking of pervasive devices," The International Journal of Computer and Telecommunications Networking, Vol. 35, P. 411-428, Mar. 2001.
- [3] U. C. Kozat and L. Tassioulas, "Network layer support for service discovery in mobile ad hoc networks," in Proceedings of INFOCOM'03, San Francisco, USA, Apr. 2003.
- [4] W. Ma, B. Wu, W. Zhang, and L. Cheng, "Implementation of a light service advertisement and discovery protocol for mobile ad hoc network," in Proceedings of IEEE GLOBECOM'03, San Francisco, USA, Dec. 2003.
- [5] L. Cheng, "Service advertisement and discovery in mobile ad-hoc networks," in Proceedings of CSCW'02, New Orleans, USA, Nov. 2002.
- [6] R. Koodli and C. E. Perkins, "Service discovery in on-demand ad-hoc networks," Internet draft, MANET Working Group, draft-koodli-manet-servicediscovery-00.txt, Sept. 2002.
- [7] C. Oh, Y. Ko, and Y. Roh, "An integrated approach for efficient routing and service discovery in mobile ad hoc networks," in Proceedings of CCNC'05, Las Vegas, USA, Jan. 2005.
- [8] R. Harbird, S. Halies, and C. Mascolo, "Adaptive resource discovery for ubiquitous computing," in Proceedings of the 2nd Workshop on Middleware for Pervasive and Ad-Hoc Computing, Toronto, Canada, Oct. 2004.
- [9] P. Samar, M. R. Pearlman, and Z. J. Hass, "Independent zone routing: an adaptive hybrid

- routing framework for ad hoc wireless networks," IEEE/ACM Transactions on Networking, Vol. 12, No. 4, Aug. 2004.
- [10] M. R. Pearlman and Z. J. Haas, "Determination of the optimal configuration for the zone routing protocol," IEEE Communications., Vol. 17, pp.1395-1414, Aug. 1999.
- [11] Z. J. Haas, M. R. Pearlman, and P. Samar, "The bordercast resolution protocol (BRP) for ad hoc networks," IETF, MANET Internet Draft, July 2002.
- [12] C. E. Perkins, P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," Proceedings of ACM SIGCOMM, pp.234-244, Aug. 1994.
- [13] C. E. Perkins, E. M. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF, RFC 3561, July 2003.
- [14] Z. J. Haas, M. R. Pearlman, and P. Samar, "The zone routing protocol (ZRP) for ad hoc networks," IETF, MANET Internet Draft, July 2002.



노 동 건

2000년 서울대학교 컴퓨터공학과 졸업
 2002년 서울대학교 컴퓨터공학부 석사
 졸업. 2004년 서울대학교 컴퓨터공학부
 박사 수료. 2004년 3월~현재 서울대학
 교 컴퓨터공학부 박사과정. 관심분야는
 Ad-hoc Network, Sensor Network,

Mobile Computing



신 현 식

1980년 미국 텍사스대학교 석사. 1985년
 미국 텍사스대학교 박사. 1986년~현재
 서울대학교 교수. 관심분야는 Mobile
 System, Real Time System, Embed-
 ded System