

인터넷 호스트 이동성을 지원하는 DNS기반 위치수정 방안

(DNS Based Location Management Scheme Supporting
Route Optimization for Mobile Host)

조 인 준 [†] 조 기 환 ^{**}

(In-June Jo) (Gi-Hwan Cho)

요 약 인터넷 프로토콜은 호스트 식별에 IP주소를 사용하는 반면에 응용프로그램에서는 인지의 편의성을 고려하여 호스트 이름을 사용한다. 따라서 사용자가 사용하는 호스트 이름을 IP주소로 변환해 주는 DNS(Domain Name Server)사용은 인터넷 응용 환경에서 절대적이다. 본 논문에서는 그 중요성이 증대되고 있는 인터넷상에서 호스트 이동성을 지원코자 할 때, DNS기능을 활용하여 호스트의 이동성에 적응적인 위치수정(Location Update)과 경로설정(Routing) 방안(DNSLR : DNS based Location Registration)을 제안한다. DNSLR은 통신이 빈번하게 이루어지는 국지 DNS(Local DNS)를 중심으로 이동 호스트의 위치수정을 제한하기 때문에 위치수정의 부담을 줄이는 반면에 대부분의 통신이 이동 호스트의 홈 에이전트(HA : Home Agent)를 거치지 않고 이루어지기 때문에 최적에 가까운 경로설정을 제공한다.

Abstract While Internet protocol makes use of the IP address to identify a computer, application program utilizes the domain name for the sake of easy to use on the application level. Therefore, the use of DNS(Domain Name Server), which converts a domain name into its corresponding IP address, becomes inevitable to the normal application users. In this thesis, we propose an adaptive location update and routing scheme, named DNSLR (DNS based Location Registration), for Internet host mobility environment. DNSLR decreases the location update overhead by limiting the update effects into local DNS area, but provides nearly optimal routing for most communications with bypassing a moving destination hosts home agent.

1. 서 론

90년 중반에 들어서 인터넷워크 환경에서 호스트의 이동성을 지원하는 다양한 프로토콜이 제안되었다[1, 2, 3, 4, 5, 6, 7, 8]. 이들 사이에서 중요한 차이점은 이동 호스트의 현재의 위치 추적을 위해서 네트워크내로 이동 호스트의 위치정보를 전파하는 방법과 이동호스트에게 패킷을 전달하기 위해서 경로를 설정하는 방법에서 찾을 수 있다. 즉, 호스트가 이동하는 환경에서 가장 중

요한 기술적인 요소는 네트워크내로 위치정보를 어떻게 효율적으로 분산시키고 이를 패킷의 경로설정에 어떻게 최적으로 활용할 것인가 하는 점이다[3].

이동호스트가 이동하면 이동호스트로 향하는 패킷은 방향전환(Forwarding)지원을 받아야 한다. 이를 위해 이동호스트는 일반적으로 2개의 주소를 유지한다. 하나는 이동호스트가 가입된 지역을 식별하기 위한 논리적인 위치 식별자로 홈 주소(영구주소 : *home_IP*)를 유지하고, 다른 하나는 이동호스트가 이동한 지역을 식별하기 위한 물리적인 위치 식별자로 COA(Care-Of-Address)를 유지한다. COA는 이동호스트에게 서비스를 제공하고 있는 FA(Foreign Agent)의 주소이거나 FA가 지정한 임시주소이다. 이와 같이 유지되는 {*home_IP*, COA} 쌍을 이동바인딩정보라 하고 이를 특정 이동호스트의 현 위치식별에 사용한다. 호스트들이 이동호스트와 통신을 하기 위해서는 이동호스트의 이동

본 연구는 배재대학교 연구비를 지원받아 수행되었음.

[†] 정 회 원 : 배재대학교 정보통신공학부 교수
injune@mail.paichai.ac.kr

^{**} 정 회 원 : 전북대학교 전자정보공학부 교수
ghcho@cs.chonbuk.ac.kr

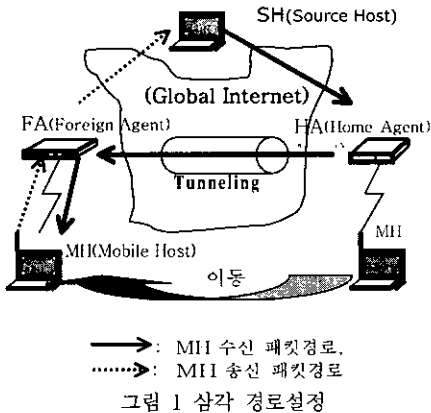
논문접수 : 2000년 1월 3일

심사완료 : 2000년 10월 9일

바인딩정보 얻기 위한 전략이 필요하다. 이를 위한 가장 일반적인 접근방법은 기존의 네트워크 기반구조에 이동성을 지원하는 개체(홈 영역내의 라우터들 혹은 라우터 사이의 어떤 장소)를 정의하고 이에 이동바인딩정보를 저장하여 패킷 방향전환에 이를 활용할 수 있도록 기존의 인터넷 프로토콜을 조정하는 방법이다.

이러한 접근법에 따라 정의된 이동성 지원개체가 자신에게 저장된 이동바인딩정보를 목적으로 한 패킷을 수신하게 되면 방향전환기법에 따라 이 정보를 참조하여 지정된 위치(즉, COA)로 패킷을 전송한다. 이는 패킷의 경로설정 결정이 이동성지원 개체에 저장된 가용한 이동바인딩정보를 기반으로 이루어짐을 의미한다. 따라서 호스트 이동에 따라 형성되는 패킷의 경로는 이동호스트의 이동바인딩정보가 네트워크내의 어느 곳에 저장되어 있는가에 따라 결정적으로 영향을 받게 된다.

이동 IP에서 진보된 위치관리 전략을 사용하지 않으면 송신호스트로부터 이동호스트로 향하는 패킷이 홈 네트워크를 경유하는 기본경로를 통해서 전달되게 된다. 이는 불필요하게 장거리 경로가 형성됨을 의미한다. 하지만, 이동호스트로부터 송신되는 패킷은 홈 네트워크를 경유하지 않고 직접경로를 통해서 전달된다. 상기와 같은 경로설정을 삼각경로설정(Triangle Routing)이라고 부른다([그림1]). 이 방안은 1997년에 IETF RFC2002 [4] 인터넷 표준으로 확정되어 오늘에 이르고 있다.



삼각경로설정의 문제점은 패킷 전달 지연 현상과 이동호스트의 이동율이 높을 경우 과도한 위치설정 현상이다. 또한 동일 서브넷에 이동호스트와 송신호스트가 동시에 존재할 경우, 두 컴퓨터간에 송수신 시간의 불균형(Asymmetric)이 심각하게 발생한다[1,3]. 특히 호스트가 광 범위하게 인터넷을 이동하게 될 경우나 실시간

멀티미디어 응용을 수행할 경우에는 이러한 현상이 더욱 심각하다. 따라서, 이동사용자에게 심각한 성능 투명성 문제를 제기하게 된다.

이러한 문제점의 기본 해결책은 송신호스트로 하여금 현재 MII가 접속중인 FA를 인식할 수 있게 하는 방안이다. 이렇게 되면 패킷은 MH의 HA를 거치지 않고 직접 FA로 터널전달된다. 즉, 송신호스트에서 이동호스트의 FA까지 직접경로를 설정하게 된다. 이와 같이 HA를 우회하는 최적에 가까운 경로설정(직접경로설정)방안에 관한 연구가 참고문헌[1, 3, 8] 등에서 제안되었다.

상기의 참고문헌 등에서 제안된 직접경로설정 방안의 공통적인 문제점은 송신호스트가 사용할 것이라는 가정을 전제로 MII의 위치정보가 전파된다는 점이다. 따라서, 불필요한 위치정보가 전파됨에 따라 네트워크에 부담을 가중시키고, 호스트가 불필요한 위치정보를 캐쉬에 유지하기 때문에 캐쉬 크기에 대한 부담과 캐쉬의 정확성 저해로 우회경로 선택 가능성을 증가시킨다.

본 논문에서는 이러한 문제의 해결을 위해 새롭게 고안된 직접경로 설정방안으로 DNSLR을 제안한다. 고안된 DNSLR은 인터넷 응용프로토콜인 DNS프로토콜 [10,11]과 기본 이동 IP[4, 5]의 조합으로 설계되었다.

2. 본 연구의 동기 및 내용

인터넷 프로토콜에서 호스트와 호스트간의 통신은 수신자의 명시방법에 따라 2가지 형태로 시작된다. 즉, 인터넷 응용이 수신지를 (1) IP주소로 명시하는 경우와 (2) 호스트 이름으로 명시하는 경우이다.

오늘날 인터넷이 급속하게 성장함에 따라 전문가의 전유물처럼 인식되었던 인터넷이 일반대중으로 확대되어 Cyber지구촌을 형성해 가고 있다. 이러한 동향에 따라 인터넷 프로토콜 사용행태도 앞 단락의 (1)의 형태에서 (2)의 형태로 급속하게 전환되고 있다. 따라서, IP주소라는 전문적인 호스트 지정방법 보다는 손쉬운 호스트 이름의 사용은 절대적이다. 이러한 패러다임으로 변화 때문에 DNS는 인터넷 대중화에 필수적인 핵심요소가 되었음은 물론 이를 이용하여 새로운 인터넷 서비스를 추가하기 위한 연구가 활발히 진행중이다[12, 13].

본 논문에서 DNSLR방안을 연구하게 된 동기도 상기와 같은 인터넷 프로토콜의 사용행태의 패러다임 변화에 기인한다. 즉, 서론에서 연구된 직접(최적)경로설정 방안 모두는 DNS의 사용빈도가 낮은 (1)의 경우를 대상으로 고안된 것이기 때문에 DNS프로토콜이 직접경로 설정에 전혀 관여하지 않는다. 하지만, (2)의 경우에는 대부분의 인터넷 통신이 호스트 이름으로 시작되기 때

문에 DNS프로토콜의 사용빈도가 매우 높게 된다. 따라서 사용빈도가 높은 DNS프로토콜을 이동호스트의 이동바인딩정보의 전파수단으로 활용하면 앞에서 설명한 직접경로설정 방안 모두를 대체할 수 있다는 전제가 성립된다. 즉, 빈번하게 호출되는 DNS데이터베이스에 이동호스트의 이동바인딩정보를 저장하고, 송신호스트가 이동호스트 이름 해결을 위해 이에 접근하였을 때, 저장된 이동바인딩정보를 피기백(Piggyback)시켜 송신호스트에게 전파한다. 송신호스트는 이를 활용하여 직접(최적)경로 설정한다. 따라서 DNS가 직접경로 설정에 적극 관여하는 방안이다. 이렇게 되면 기 제안된 직접경로 설정방안에서 지적된 문제를 효과적으로 해결할 수 있다.

본 논문의 구성은 다음과 같다. 3장에서는 DNSLR을 위한 수정된 기반구조를 정의하고, DNSLR의 설계 내용으로 위치수정 절차와 경로설정 절차를 다루었다. 4장에서는 제안된 DNSLR을 시뮬레이션으로 구현한 내용을 다루었고, 5장에서는 구현된 DNSLR의 시뮬레이션 결과를 다른 방안(RFC 2002[4, 5], 참고문헌[1,8])과 비교하여 분석하였다. 6장에서는 본 연구를 통하여 얻어진 향후 연구분야를 정리하였고, 7장에 결론을 맺었다.

3. DNS기반 위치수정 및 경로설정(DNSLR)

3.1 DNSLR의 기본 개념

호스트 이동 패턴에서 국지성 이용은 효과적인 위치수정과 경로설정에 결정적인 역할을 한다[3]. 이는 이동호스트가 대부분 자신의 홈 서버 네트워크 혹은 현재 위치하고 있는 서버 네트워크가 속해 있는 지정된 영역 주위로 이동한다는 가정을 기반으로 한다. 그리고 이동호스트와 통신하고자 하는 송신호스트는 전체 호스트 중에서 극히 제한적이다. 이와 같은 상황을 근거로 각 이동호스트를 위한 DNS국지지역과 필요기반 호스트개념을 유도해 낼 수 있다. DNS국지지역은 호스트가 등록된 지역의 DNS집합으로 정의하고, 필요기반 호스트는 이동호스트와 통신이 필요한 송신호스트로 정의한다. 따라서, 이동호스트의 위치정보 전파를 DNS국지지역 개념에 따라 이동호스트가 등록된 지역의 DNS내로 제한하고, 필요기반 호스트개념에 의해 이동호스트와 통신이 필요한 송신호스트로만 제한할 수 있다.

이러한 2가지 고안 즉, DNS국지성과 필요기반 호스트개념이 본 논문에서 제안한 DNSLR의 기본방안에 접목될 수 있다. 각 국지 지역내의 DNS가 자신의 영역에 등록된 이동호스트들의 이동바인딩정보를 유지함으로써 이동호스트의 현재 위치정보 분배기 역할을 하고, 이동

호스트와 통신이 필요한 송신호스트로 제한하여 이동바인딩정보를 분배시켜 직접경로 설정에 활용할 수 있다. 이렇게 되면 기존의 직접경로설정 방안에서 송신호스트에게 행해졌던 별도의 위치정보 알림 절차가 제거된다. 이는 DNS프로토콜에 의해 위치정보가 송신호스트에게 피기백되기 때문이다. 또한 기존의 최적경로설정 방안에서 송신호스트가 사용 가능성을 전제로 위치정보를 캐쉬하는 문제가 해결된다. 이는 필요기반 호스트 개념에 따라 송신호스트가 통신이 필요한 호스트의 위치정보만을 전파 받기 때문이다. 따라서 캐쉬에 불필요한 호스트의 위치정보가 유지될 가능성이 낮아져 우회경로 설정 가능성이 줄어들 뿐만 아니라 캐쉬 크기에 대한 부담도 제거된다. 이를 지원하기 위해 이동호스트는 이동을 행할 때 마다 DNS국지지역개념에 따라 자신의 위치정보를 자신이 등록된 DNS로 제한하여 알려주어야 한다. 이렇게 되면 DNS가 호스트 이동성 지원을 목적으로 호스트 관련 정보(호스트 이름, IP주소, 이동바인딩정보 등)를 전 인터넷에 효과적으로 제공하게 되어 분산형태의 이동성지원 인터넷 기반구조 역할이 가능하다.

3.2 DNSLR의 기반구조

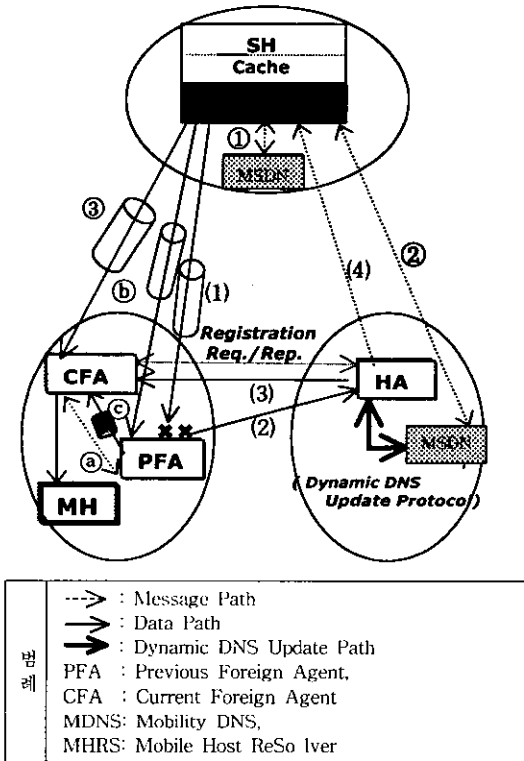
호스트가 네트워크 연결을 그대로 유지하면서 다른 네트워크로 이동하기 위해서는 이동성 기능을 지원하는 새로운 네트워크 개체들의 집합이 정의되어야 한다. DNSLR은 MH(MH: Mobile Host), MA(MA: Mobile Agent), MDNS(MDNS: Mobility Domain Name Server), MHR(MH ReSolver)라는 이동성 지원 개체들로 구성된다. [그림2.1]은 이들에 대한 구성도를 보인 것이다.

이들 개체에 대해 설명을 단순화하기 위해 참고문헌[4,8]과 비교하여 DNSLR에서 각 개체에 대해 추가적으로 정의된 기능만을 중심으로 설명을 한다. 여기에서 설명되지 않은 내용은 참고문헌[4, 8]과 동일하다.

MH개체는 자신이 가입된 DNS(본 논문에서는 MDNS)에게 자신의 이동바인딩정보 전파를 위해서 MDNS_IP를 추가적으로 유지한다는 점을 제외하고는 기존의 방안과 동일한 기능을 가진다.

MA개체는 HA와 FA(CFA, PFA)로 대별된다([그림 2]). HA개체에 추가된 기능은 MH의 FA로부터 등록요구메시지를 수신하게 되면 동적DNS수정프로토콜[14]을 이용하여 MDNS에게 MH_IP수정요구메시지(다음 단락에서 설명)를 보낸다는 점이다. 다음으로 FA개체는 HA에게 MH의 등록요구메시지를 보낼 때 그 메시지내에 MH로부터 수신한 MDNS_IP를 포함시킨다는 점을 제외하고는 기존의 기능과 동일하다.

MDNS개체는 DNSLR방안에서 새롭게 정의된 개체로 기존의 DNS에 MH의 이동성 지원기능을 추가한 특수한 DNS개체이다. DNSLR에서는 MH의 국지영역에 최소한 한 개 이상의 MDNS를 위치시킨다. 이에 는 자신이 관할하는 영역에 존재하는 모든 MH들이 마 리 등록된다. MDNS는 MH의 이동성을 지원하기 위해 MH의 이동바인딩정보를 저장할 수 있는 DNS RR(Resource Record)을 새롭게 정의하고 있다. 이러한 용도로 정의된 DNS RR을 *MH-RR*로 명명하고 이에 대한 구조는 3.3.1절에 자세히 설명된다. MDNS의 기능은 다음과 같다.



- ①: Local MDSN Query/Response
- ②: Remote MDNS Query/Response
- ③: Direct Tunneling
- [MH 이동시 On-going 패킷경로]
- ④: Registration Request/Response
- ⑤: On-going Packet
- ⑥: Direct tunneling
- [코야(Orphan)패킷경로:특수터널]
- (1): 코야패킷 (2):HA로 터널전달
- (3): CFA로 터널전달
- (4): Binding Update

그림 2 DNSLR기반구조

첫째, HA로부터 동적DNS수정프로토콜[14]를 통하여 *MH-RR* 수정요구메시지를 수신하게 되면 해당하는 MH의 DNS RR(즉, *MH-RR*)에 MH의 이동바인딩정보를 갱신하는 기능을 수행한다. 둘째, 송신호스트 혹은 수신호스트의 MHRS(다음 절에서 설명함)개체로부터 호스트 이름에 대응하는 DNS RR 질의메시지를 수신하게 되면 해당하는 DNS RR(*MH-RR*포함)을 응답메시지에 포함시켜 송신자에게 보내는 기능을 한다.

MHRS개체 역시 DNSLR방안에서 새롭게 정의된 개체로 송신호스트의 응용으로부터 요청된 호스트 이름에 관련된 정보를 MDNS로부터 추출하여 호스트 응용에 제공하는 이름해결자 프로그램이다. 어의 동작형태는 반복적 혹은 재귀적 유행을 갖는다[10, 11]. DNSLR에서는 후자인 재귀적 이름해결자를 선택한다. 이에 대한 자세한 내용은 3.3.2절에 설명한다. 기본적으로 MHRS는 MII가 요구한 이름에 대응하는 위치정보를 획득하기 위해 MDNS에게 질의메시지를 보내고 응답메시지를 수신하는 기능을 한다.

[그림2]의 기반구조에 표시된 위치정보 수정 및 경로설정이 이루어지는 과정은 3.3.2 - 3.3.3절에서 자세히 설명된다.

3.3 DNSLR의 위치수정과 경로설정 설계

3.3.1 이동IP와 DNS 프로토콜간의 인터페이스 설계

DNSLR방안의 위치수정 및 경로설정 절차의 설계내용을 설명하기 위해 먼저 이동IP와 DNS프로토콜간의 인터페이스 설계내용을 기술한다. 제안된 방안에서 MH는 자신이 가입된 국지 MDNS에 자신의 이동바인딩정보를 유지하고, SH는 이를 직접경로 설정에 활용하기 위해서 정규적인 DNS프로토콜을 이용한다. 따라서 DNS프로토콜이 이동호스트의 특성을 반영할 수 있도록 다음과 같은 조정 및 변경이 필요하다.

첫째, 기존의 DNS 데이터베이스에 MH의 이동정보(위치정보 및 TTL)를 저장하기 위한 데이터구조로 DNS RR이 새롭게 정의되어야 한다. 이는 RFC 1034[10], RFC 1035[11]에 근거하여 다음과 같이 정의하였다.

```
{ owner(domain-name), Type(MH-RR), class(IN),
TTL(TTL 값), RDATA(home_IP, COA, M-TTL,
stay-TTL) }
```

*MH-RR*은 새롭게 정의한 DNS RR유형을 의미하고, RDATA(Resource DATA)내의 *COA*에는 MH의 위치주소가 저장되고, *M-TTL*에는 HA가 동적DNS수정 프로토콜을 통해서 제공한 캐쉬시간이 저장된다. 그리고 *stay-TTL*에는 MH의 사용자가 입력한 방문 예정기간

이 저장된다.

둘째, 이동 컴퓨팅환경에서는 이동호스트의 위치정보 (COA)가 빈번하게 변경되기 때문에 이를 반영하는 캐쉬정책이 정의되어야 한다. 이를 위해 본 논문에서는 참고문헌[1, 8]에서의 호스트 캐쉬방안과 DNS프로토콜 [10, 11]에서의 캐쉬방안을 조합하여 이동성을 지원하는 캐쉬방안을 제안한다. 설명의 혼돈을 피하기 위해서 참고문헌[1,8]의 이동호스트 캐쉬시간을 M_TTL 로 표기하고, 참고문헌[10,11]의 DNS RR 캐쉬시간을 D_TTL 로 표기한다. 그리고 본 방안에서 새로운 캐쉬변수로 $stay_TTL$ 을 도입한다. 이에는 사용자가 예측하여 입력한 방문시간이 저장된다.

이와 같이 정의된 변수를 이용하여 참고문헌[1,8]에 정의된 송신호스트의 캐쉬엔트리 구조가 DNSLR에서는 다음과 같이 새롭게 정의되어야 한다.

참고문헌[1,8] 캐쉬엔트리 구조 :

{ *home_IP, COA, M_TTL* }

DNSLR 캐쉬엔트리 구조 :

{*domain-name, MDNS_IP, home_IP, COA, M_TTL, stay_TTL* }

새롭게 정의된 캐쉬 항목들은 이동IP프로토콜과 DNS 프로토콜에 의해 생성되고 관리된다. 즉, *domain_name, MDNS_IP, home_IP, stay_TTL* 항목은 송신호스트의 이름 해결자(DNS프로토콜)만이 생성 가능하고, *COA*와 *M_TTL* 항목은 이동호스트의 HA(이동IP프로토콜)만이 생성 가능하다. 이렇게 정의된 송신호스트의 캐쉬엔트리를 대상으로 DNSLR은 다음과 같이 캐쉬를 운용,관리한다.

(1) 이동호스트 캐쉬엔트리 생성 : 캐쉬 내에 통신하고자 하는 이동호스트의 엔트리가 존재하지 않으면, 송신호스트가 MDNS에게 이를 요구하여 정의된 캐쉬엔트리의 모든 항목들을 생성한다.

(2) 이동호스트 캐쉬엔트리 수정 : 다음의 2가지 경우에 수정이 일어난다. 첫째, 해당 이동호스트의 캐쉬엔트리내의 $M_TTL = 0$ 이고 $stay_TTL = 0$ 이면서 캐쉬영역이 가용한 경우, 혹은 $M_TTL = 0$ 이고 $stay_TTL \neq 0$ 일 경우(사용자가 방문을 계속하고 있는 경우)에는 캐쉬엔트리내의 *MDNS_IP*를 이용하여 직접 MDNS로부터 새로운 이동정보(위치정보, TTL정보)를 취득하여 캐쉬엔트리를 수정한다. 이는 새로운 COA를 취득하여 캐쉬 적응율을 제고하기 위한 것이다. 둘째, 특수터널을 사용하는 경우에는 HA로부터 *COA*와 *M_TTL*를 제공

받아 수정이 이루어진다.

(3) 이동호스트 캐쉬엔트리 삭제 : 다음과 같은 2가지 경우에 삭제가 일어난다. 첫째, $M_TTL = 0$ 이고 $stay_TTL = 0$ 이면서 가용한 캐쉬영역이 없을 경우 자동 삭제된다. 둘째, 가용한 캐쉬영역이 없을 경우에는 캐쉬관리 알고리즘(예, LRU(Least Recently Used) 기법 등)에 따라 삭제된다.

다음으로 상기와 같은 DNSLR의 캐쉬정책을 지원하기 위해 DNS캐쉬정책이 조정될 필요가 있다. DNS는 효율적인 이름해결을 위해 *NS(Name Server) RR*과 호스트 *RR*(본 논문에서는 *MH_RR*)을 대상으로 캐쉬를 행한다. DNSLR측면에서 보면 *NS RR*정보는 변경이 자주 일어나지 않는 정적인 특성을 갖는데 반해 *MH_RR*정보는 변경이 자주 일어나는 동적인 특성을 갖는다. 따라서 DNS프로토콜이 이를 반영하기 위해서는 *NS RR*에 대해서는 정규적인 DNS캐쉬정책을 그대로 유지하고 *MH_RR*에 대해서는 캐쉬하지 않는 정책(이동호스트를 MDNS에 등록 시 $D_TTL = 0$ 으로 초기값 설정)을 사용한다. 즉, 변경 가능성이 없는 *NS RR*을 캐쉬함으로써 DNS서버 탐색 부담을 감소시키고, 변경 가능성이 높은 *MH_RR*의 캐쉬는 자신이 행하지 않고 상기에서 설명한 DNSLR프로토콜의 캐쉬정책에 위임시켜 이원화된 캐쉬정책을 단일화 하자는 것이다. 따라서 DNS프로토콜의 이름해결자가 *MH_RR*을 취득하게 되면 DNSLR프로토콜에 직접전달만 하고 자신의 역할을 종료하게 된다. 즉, 이름해결자와 DNS서버는 *MH_RR*에 대해서 캐쉬하지 않음을 뜻한다. 이를 반영한 DNSLR프로토콜의 기본적인 캐쉬 동작절차는 다음과 같다. 이름해결을 요구 받은 송신호스트가 먼저 자신의 캐쉬에서 이동호스트의 엔트리를 찾는다. 발견되었을 경우에는 M_TTL 과 $stay_TTL$ 값에 따라 최적경로를 설정하거나 혹은 캐쉬내의 *MDNS-IP*를 이용하여 MDNS로부터 새로운 위치정보를 취득한다. 취득된 정보는 DNSLR캐쉬정책에 따라 송신호스트가 캐쉬한다. 발견되지 않았을 경우에는 송신호스트가 DNS프로토콜에게 이동호스트의 위치정보 획득을 전가한다. 이때 DNS프로토콜은 *NS RR*만 캐쉬하고 *MH_RR*정보는 DNSLR프로토콜에 전달만 하고 캐쉬하지 않는다.

이의 결과로 DNS 프로토콜과의 양립성이 제고되고 참고문헌[1,8]에서 제시한 기본 캐쉬정책을 유지하면서 DNSLR에 적응적인 캐쉬정책의 도출이 가능하였다.

셋째, 이동호스트는 부족한 컴퓨팅자원으로 구성된다. 이는 휴대성을 제고하기 위해 저전력 및 소형화를 추구

하기 때문이다. 따라서 이동호스트에서 동작되는 소프트웨어는 경량화 되어야 한다. DNS 프로토콜의 이름 해결자는 2가지 형태가 있다. 이 중에서 재귀적 방식으로 동작하는 이름 해결자(스터브 이름해결자)를 선택한다. 이는 컴퓨팅자원이 부족한 호스트에서 동작하도록 이름 해결기능의 대부분을 MDNS서버에게 전가시켜 동작하기 때문에 이동호스트에 적합하다.

넷째, 기존의 DNS에서 DNS RR의 변경은 주 파일(Master File)을 통해서 일괄수정처리 방식으로 이루어진다. 하지만 본 제안 방안에서는 MH가 이동하면 즉시 위치관련 정보가 DNS데이터베이스에 반영되어야 하기 때문에 이러한 방법이 적합하지 않다. 이에 대한 해결책으로 DNS RR을 동적으로 수정할 수 있는 동적DNS수정(Dynamic DNS Update) 프로토콜[14]을 채택한다.

3.3.2 위치수정 절차 설계

3.2절에서 정의한 기반구조에서 DNSLR의 위치수정은 패킷 운반 지연 및 네트워크 자원의 소비를 줄이기 위해서 2단계 수정절차를 거치도록 설계하였다.

제 1단계의 위치수정은 MH가 새로운 위치로 이동했을 경우 다음과 같은 프로토콜에 의해 이루어진다.

[MH→FA] : { *home_IP*, *MDNS_IP*, *COA*, *stay_TTL* }

MH가 FA로부터 COA를 할당 받아 FA에게 등록요구메시지를 보낸다.

[FA→HA]: { *home_IP*, *MDNS_IP*, *COA*, *stay_TTL* }

FA가 자신의 방문자목록에 MH주소를 등록한 후, MH의 HA에게 이동IP프로토콜[4,5]을 통해서 등록요청한다.

[HA→MDNS] : { *home_IP*, *COA*, *M_TTL*, *stay_TTL* }

HA가 자신의 가입자목록에 MH가 가입되었는지를 확인(*home_IP*사용)하고, 방향전환목록에 COA를 등록한다. 그 다음에 HA가 제공하는 *M_TTL*를 추가하여 DNS수정메시지를 제작하고, 이를 *MDNS_IP*를 사용하여 동적DNS프로토콜[14]을 통해서 MDNS에게 보낸다.

[MDNS→HA] : { 수정 확인메시지 }

MDNS는 *MH_RR*의 *COA*, *M_TTL* 및 *stay_TTL* 항목을 수정한다. 그 다음에 수정확인메시지를 제작하여 HA에게 보낸다.

[HA→FA] : { 등록 확인메시지 }

HA가 등록확인메시지를 FA에게 전송.

[FA→MH] : { 등록확인메시지 }

FA는 등록확인메시지를 수신하고 이를 MH에게 전송한다.

이 메시지를 MH가 수신하면 자신에게 전송되는 패킷을 수신 받기 위한 등록절차가 완료된 상태가 된다.

다음으로 제 2 단계의 위치수정은 송신호스트가 MH와 통신을 시작하는 시점에서 다음과 같은 2가지 경우의 절차에 따라 이루어진다.

[경우1] SH에 통신하고자 하는 MH가 캐쉬 되어 있지 않을 경우

[SH→MHRS] : { *MHs domain-name* }

SH가 MHRS에게 *MH_RR* 취득을 요청한다.

[MHRS→MDNS] : { *Query(MHs Domain-name)* }

이를 수신한 MHRS는 MH의 MDNS에게 *MH*의 *MH_RR*을 질의한다.

[MDNS→MHRS] : { *Reply(COA, Stay_TTL, MDNS_IP)* }

이를 수신한 MDNS는 자신의 데이터베이스에서 *MII*의 *MH_RR*를 추출하여 이를 응답메시지에 포함시켜 MHRS에게 송신한다.

[MHRS→SH] : { *COA*, *M_TTL*, *stay_TTL*, *MDNS_IP* }

이를 수신한 MHRS는 *MH_RR*로부터 *MDNS_IP*를 캐쉬하고 SLIST를 수정한다. 그리고 *MH_RR*를 SH에 보낸다.

[SH→SHs appl.] : { *MHs COA* }

SH는 수신한 *MH-RR*로부터 (*COA*, *M_TTL*, *stay_TTL*, *MDNS_IP*)를 추출하여 자신의 캐쉬에 이를 저장한다. 그리고 나서 요구한 응용에게 COA를 보낸다. 이 상태가 2단계 위치수정이 완료된 상태이다.

[경우2] SH캐쉬에 MH의 *MH-RR*이 캐쉬는 되어 있으나 *M_TTL* = 0, *stay_TTL* = 0 or ≠ 0 일 경우

[SH→MDNS] : { *Query(MHs Domain-name)* }

SH가 캐쉬로부터 MH의 *MDNS_IP*를 추출하여 직접 MDNS에게 MH의 *MH_RR*을 질의한다.

[MDNS→SH] : { *Reply(COA, Stay_TTL, MDNS_IP)* }

MDNS는 MH의 *MH_RR*이 포함된 응답메시지를 SH에게 송신한다.

[SH→SHs appl.] : { *MHs COA* }

SH는 수신한 *MH-RR*로부터 (*COA*, *M_TTL*, *stay_TTL*, *MDNS_IP*)를 추출하여 자신의 캐쉬에 이를 저장한다. 그리고 나서 요구한 응용에게 COA를 보낸다.

상기와 같이 기본적인 2단계 위치수정이 이루어지고, 기타의 부가적인 위치수정이 본 논문에서 제안한 방안의 효율성 재고를 위해 다음의 경우처럼 발생한다.

[경우1] MH가 홈으로 돌아오면 다음과 같은 위치수정이 일어난다.

[HA→MDNS] : { *MHs domain-name* }

HA는 자신의 방향전환목록에서 MII주소를 삭제한다. 그리고 자신의 MDNS에게 동적DNS수정프로토콜[14]로 위치수정요구메시지를 보낸다.

[MDNS→HA]: { *위치수정 응답메시지* }

MDNS는 MII의 *MII_RR*의 위치정보를 수정한다. 그리고 나서 HA에게 위치수정확인메시지를 보낸다.

[경우2] MH가 현재의 FA에서 또 다른 FA로 이동하면 현재의 FA는 PFA가 되고 이동한 FA가 CFA가 된다. 이러한 MH의 이동과정에서 PFA로 향하는 패킷(On-going 패킷이라 함)은 MH가 이동했기 때문에 고아패킷이 되는 데 이를 해결하기 위해서 다음과 같은 위치수정이 발생한다[8].

[CFA→PFA] : { *MIIs Current COA* }

CFA가 MH로부터 PFA의 IP주소를 받아 PFA에게 등록요구메시지를 보낸다.

[PFA→CFA]: { *on-going packet* }

PFA가 방문자목록에서 MII를 삭제하고 방향전환목록에 COA를 추가한다.

[경우 3] SH가 MH의 위치정보를 갖지 않은 PFA에게 패킷을 보냈을 경우에는 다음과 같은 위치수정 절차가 이루어진다[7].

[PFA→HA] : { *Orphan Packet* }

PFA는 자신의 방문자목록과 방향전환목록에 MH가 등록되어 있지 않음을 확인하고 HA에게 고아 패킷을 보낸다.

[HA→CFA/SII]: { (*Orphan Packet*) / (*COA*) }

HA가 방향전환목록의 COA(CFA)로 고아 패킷을 보낸다. 그리고 SH에게 바인딩수정메시지를 보내어 위치를 수정한다.

3.3.3 경로설정 절차 설계

SH와 MH간의 경로설정절차를 설명하기 위해서 SH가 유지하는 캐쉬엔트리 구조를 3.3.1절에서 설명하였다. 그리고 MHRs가 제귀적 이름해결 방법을 사용하기 때문에 SH의 국지 MDNS는 IETF RFC 1034의 정의에 따라 다음과 같은 테이블구조로 요구상태를 표시한다.

{*SNAME, STYPE, SCLASS, SLIST, SBELT,*

CACHE }

이들 변수들의 의미는 RFC 1034[10]와 동일하게 정의되고 동작된다.

DNSLR의 경로설정절차는 3.2절에서 정의된 기반구조를 바탕으로 3.3.2절의 이동바인딩정보 전파전략을 수행한다는 전제 하에서 다음과 같이 설계하였다.

[단계0] SH의 응용이 MH의 호스트 이름 해결을 요청한다.

[단계1] SH가 자신의 캐쉬에서 MH의 캐쉬엔트리를 탐색한다.

존재 :

(*M_TTL* ≠ 0, *stay_TTL* ≠ 0) : go to [단계 2]

(*M_TTL* = 0, *stay_TTL* ≠ 0) : go to [단계 2]

(*M_TTL* ≠ 0, *stay_TTL* = 0) : go to [단계 2]

(*M_TTL* = 0, *stay_TTL* = 0) : go to [단계 3]

· 부존 : go to [단계 3]

[단계2] SH가 MH에게 직접 혹은 HA를 경유하는 경로를 설정하여 패킷을 전송한다..

(1) SH의 ARP엔트리에 MII가 존재 : go to (2), 부존 : go to (3)

(2) 다음의 2가지 경우로 패킷이 경로설정된다.

MH가 자신의 HA에 존재할 경우 :

LAN Broadcasting으로 MII에게 직접패킷 전송

MH가 HA로부터 이동 했을 경우 :

LAN Broadcasting된 패킷을 HA가 수신하여 MH의 COA로 패킷전송

(3) COA로 직접 패킷전송 : go to [단계 8]

[단계3] SH가 MH의 COA취득을 MHRs에게 요구한다. 이를 수신한 MHRs는 SH의 국지MDNS에게 이를 전달한다. 이를 수신한 SH의 MDNS는 다음과 같은 선행작업을 행한다.

(1) *SNAME*과 *CACHE*내의 *NS RR*과의 일치성을 검사한다.

일치된 *NS RR* 존재 : 발견된 NS에게 COA획득 절의를 요청한다. : go to [단계 4]

일치된 *NS RR* 부존 : go to (2)

(2) *SNAME*과 자신의 *SLIST*내의 *NS RR*과의 일치성을 검사한다.

일치된 *NS RR* 존재 : 발견된 NS에게 COA획득 절의를 요청한다. : go to [단계 4]

일치된 *NS RR* 부존 : go to (3)

(3) *SBELT*목록을 *SLIST*에 복사하고 이 목록 중에

서 NS를 선택하여 COA 획득을 절의한다. : go to [단계 4]

[단계 4] 질의를 수신한 원격MDNS는 자신의 데이터베이스로부터 요청한 DNS RR을 탐색한다.

존재 :

SH의 MDNS를 통하여 SH의 MHS에게 MH의 MH_RR 응답메시지를 보낸다 : go to [단계 5]

부존 :

다음에 질의할 NS가 포함된 응답메시지를 SH의 MDNS에게 보낸다. : go to [단계 6]

[단계 5] MHS는 수신한 MH_RR로부터 MH의 COA를 SH에게 보낸다. : go to [단계 7]

[단계 6] 이를 수신한 SH의 MDNS는 자신의 캐쉬에 이를 저장한다. : got to [단계 3]

[단계7] SH가 MH의 캐쉬엔트리를 캐쉬하고, 이를 이용한 패킷 송신절차를 행한다. :

go to [단계 2]

[단계 8] 패킷을 수신한 FA는 다음의 경로설정 절차를 거친다.

(1) 방문자목록에서 MH를 찾는다.

존재 : MH로 직접 패킷을 전송한다.

부존 : go to (2)

(2) 방향전환목록에서 MH를 찾는다.

존재 : 방향전환목록의 COA(CFA)로

패킷을 보낸다. : go to [단계 8]

부존 : 패킷을 MH의 HA에게 보낸다.

: go to [단계9]

[단계 9] 패킷을 수신한 HA는 다음의 처리를 한다.

(1) 방향전환목록의 COA로 CFA에게 패킷을 전송한다. : go to [단계 8]

(2) 방향전환목록의 COA를 SH에게 전달한다. : go to [단계 10]

[단계 10] 이를 수신한 SH는 COA를 자신의 캐쉬에 저장한다. : go to [단계 1]

4. 시뮬레이션

본 장에서는 제안된 DNSLR방안의 우수성을 입증하기 위해 시뮬레이션을 통해 기존의 직접 경로설정 방안 [8]과 비교하여 분석한 내용을 다룬다.

4.1 시뮬레이터

시뮬레이션을 위해 사용한 시뮬레이터는 1993년에 MIT LCS ANAG(Advanced Network Architecture Group)에서 개발된 netsim[9]을 기반으로 하였다. 이는 네트워크 개체들간에 패킷교환 및 프로세스 관리를 사

건중심기법(Event driven techniques)으로 행한다.

본 시뮬레이션을 위해 3가지 형태의 구성요소를 제작하였다. 즉, (1) 네트워크 개체 구성요소로 인터넷, 이더넷, 무선넷을 제작하였고, (2) 이동성 지원개체 구성요소로 RT(RouTer), MA, DNS(MDNS)를 제작하였고, 그리고 (3) 이동성 단말개체로 이동호스트(MH)를 제작하였다. 동일형태의 모든 구성요소는 공통된 동작루틴과 네트워크 매개변수를 공유하지만, 그들은 자신의 특징들을 인지할 수 있는 서로 다른 네트워크 매개변수 값(본 시뮬레이션에서 사용된 값은 주로 기존의 연구[15, 16]로부터 사용된 것을 이용하였다.)을 가진다.

4.2 DNSLR 시뮬레이션 환경

DNSLR의 시뮬레이션을 위해서 정의된 네트워크 모델은 1개의 인터넷과 23개의 이더넷, 4개의 MDNS, 23개의 RT, 25개의 무선네트워크, 25개의 MA, 그리고 60개의 MH로 구성하였다. 이는 시뮬레이터 및 컴퓨터이션 부담 때문에 제한한 것이다. 이렇게 제한된 네트워크 모델을 대상으로 다음과 같은 5가지 프로토콜이 동작되도록 구현하였다. (1) 현재 MH가 위치한 MA에게 패킷을 전달하는 터널링 프로토콜(Tunneling protocol), (2) 이동바인딩정보를 전파하는 등록프로토콜(Registration protocol), (3) MA가 MH에게 자신을 인지시켜 주고 양자간에 네트워크 연결을 설정해 주는 Beaconsing / Solicitation 프로토콜, (4) MDNS에 MH의 이동바인딩정보를 전파하는 동적DNS수정 프로토콜(Dynamic DNS update protocol), (5) SH가 MH의 위치정보를 취득하기 위한 DNS질의/응답 프로토콜(DNS query/reply protocol) 등이다. 이들 프로토콜구현에 대한 상세한 구현내용은 지면 관계상 참고문헌[17]을 참조하기 바란다.

4.3 시뮬레이션 매개변수

이동 컴퓨팅환경에서 2가지 중요한 사건이 발생하게 된다. 즉, 주어진 시간에 이동호스트가 전송한 데이터 패킷 수를 나타내는 호출(Call)사건과 이동호스트가 행한 이동횟수를 가리키는 이동(Move)사건이다. 호출과 이동의 비율을 나타내는 CM_{ratio} 는 호출비 대 이동비의 비율로 정의된다. 이 변수는 각 이동호스트가 한번 이동하여 행한 평균 호출 수로 정의한다. 따라서 이 변수로 각 이동호스트가 행한 이동과 호출 빈도를 정규화 할 수 있다. 사실 이 변수는 실험적인 매개변수 이지만 본 논문의 시뮬레이션에서는 각 시뮬레이션 수행마다 명시적인 값을 지정하였다.

각 이동 호스트는 랜덤하게 패킷 호출과 이동사건을 생성한다. 일반적으로 단위시간당 발생한 사건의 수는

포아송 분포를 가진 랜덤값으로 모델화 한다. 고정된 시간 범주에서 발생한 평균사건의 수와 연속적인 호출사이의 시간 간격이 주어지면 이들은 포아송 값으로 산정할 수 있다. 유사하게 연속적인 이동사이의 시간간격은 CM_{ratio} 에 포아송 값을 곱하여 계산될 수 있다. 각 이동 호스트는 계산된 시간범위에서 호출과 이동을 반복적으로 행한다. 이러한 사건 때마다 대응되는 루틴이 수행된다.

다음으로 이동과 패킷 호출 방향의 형식화를 위한 중요한 변수로 S_{rate} (Symmetric rate)를 정의한다. 이 변수는 패킷 호출과 이동이 서로 다른 목적지를 향해서 발생함을 지시하는 역할을 한다. 즉, 패킷 호출의 목적지는 MH이고, 이동의 목적지는 무선네트워크이다. 따라서 이동과 패킷 호출의 규칙은 서로 다른 매개변수에 의해 선택된다. 호스트 이동방향에 대해서 S_{rate} 는 MH가 자신의 국지 MDNS지역에 머무는 시간과 다른 국지 MDNS지역에서 보낸 시간 사이 비율로 정의된다. 또한, 이 변수는 자신의 국지 MDNS지역 내에서 MH가 자신의 HA에 머무는 시간과 다른 MA에서 보낸 시간 사이의 비율에도 반영된다. 그리고 패킷 호출의 경우에 이 변수는 자신의 국지 MDNS지역내의 MH에게 행하는 패킷 호출수와 다른 국지 MDNS지역내의 MH에게 행하는 패킷 호출 수 사이의 비율로 정의된다.

상기에서 정의된 매개변수 이외의 것은 참고문헌[17]에 기술된 정의를 그대로 사용하였다.

4.5 이동과 패킷 호출 시나리오

시뮬레이션 수행 시 가장 중요한 측면은 MDNS의 국지성을 고려하여 이동과 패킷 호출을 형식화하는 작업이다. 형식화된 이 시나리오가 인터넷네트워크 통신환경에서 대부분의 호스트 이동과 패킷 호출을 일반화하고 이의 수용이 가능하도록 분명해야 한다. 현실적으로 이동과 패킷 호출이 실 생활에서는 몇 가지 형태를 갖지만 한 호스트가 얼마나 자주 패킷을 보내고 얼마나 오랫동안 이동 에이전트에 머물 것인가 등에 대해 특정 지우기는 어렵다. 하지만, 시뮬레이션을 위해서는 문제영역을 형식화하기 위한 대표적인 시나리오의 정의가 필요하다. 이 시나리오를 기반으로 각 방안을 수행하여 결과를 도출함으로써 각 방안의 비교 검토를 위한 객관적인 기준을 제공할 수 있다. 본 논문에서 사용한 이동과 호출 시나리오는 참고문헌[3]에서 정의한 시나리오를 그대로 사용하였다. 이는 DNSLR의 시뮬레이션 환경이 참고문헌[3]에서 전제조건 환경과 동일하기 때문이다.

5. 시뮬레이션 결과

논문에서 제안된 방안의 효과를 분명히 보이기 위해 참고문헌[4,5]의 기본방안에 DNS를 추가한 DNS2002 방안과, 참고문헌[8]에서 제안된 직접경로설정 방안(직접경로)에 DNS를 추가로 구현한 방안(편의상 DNSIMHP방안이라 명명함)을 대상으로 고안된 DNSLR방안을 비교 분석하였다. 시뮬레이션이 여러 랜덤 Seed값으로 수행되었지만 52763817로 수행된 결과만 여기에 기술한다. 다시 말하면, 시뮬레이션은 서로 다른 랜덤값을 적용해도 동일한 결과를 보였다. 몇 개의 그래프가 다소 불안정 상태를 보였지만 각 방안의 일반적인 행위의 검토에는 충분하였다. 본 시뮬레이션에서는 0.1단위로 S_{rate} 값을 증가시켜 가면서 실행하였다. 이 논문에서는 지면의 제한 때문에 4개의 S_{rate} (0.4, 0.5, 0.6, 0.7)에 대한 결과만을 정리하였다.

5.1 등록결과

[그림3]은 정의된 시뮬레이션 환경에서 DNSLR, DNSIMHP, DNS2002방안 각각에 대해 등록 발생 수를 보여주고 있다. 이 그림에서 관찰할 수 있는 핵심 내용

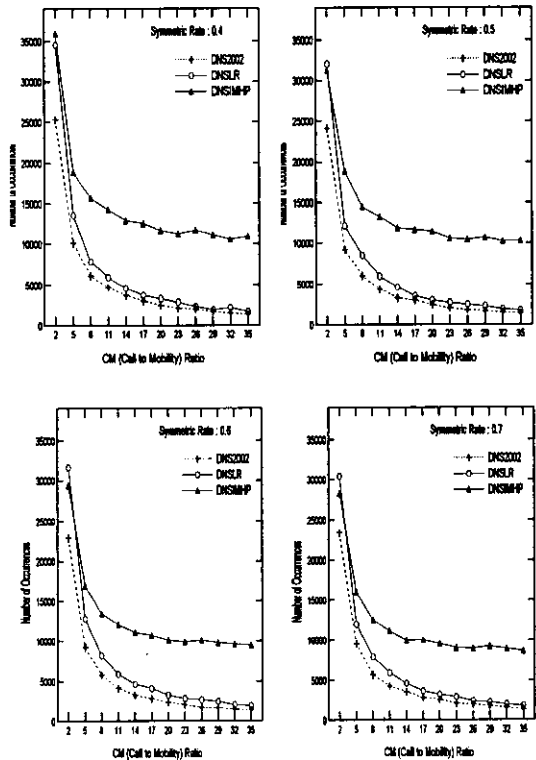


그림 3 등록횟수

은 DNSLR이 DNSIMHP보다 상대적으로 매우 낮은 등록빈도를 보인다는 것이다. 이는 DNSLR방안에서 송신 호스트가 통신초기에 DNS질의/응답 과정에서 대부분의 이동호스트 위치정보(COA)를 동시에 전달 받는데 반해, DNSIMHP방안에서는 부가적인 위치알림 메시지를 통해서 HA 혹은 PFA로부터 이동 호스트의 위치정보를 전달 받기 때문이다. 따라서 DNSLR의 등록비용이 DNSIMHP보다 저렴하다. DNSLR과 DNSIMHP의 등록부담은 CM_{ratio} 가 증가함에 따라 그 편차가 일정한 비율임을 관찰할 수 있다. 이는 이동비율이 낮아짐에 따라 두 방안 모두 안정된 위치수정이 이루어지기 때문이다.

5.2 네트워크 점유시간

[그림 4]는 각 방안별로 데이터 패킷 점유시간을 보여주고 있다. 이 그림에서 본 바와 같이 DNSLR이 DNSIMHP와 DNS2002보다 월등하게 네트워크 점유시간이 짧게 나타났다. 이는 DNSLR의 사전위치알림방식의 우수성 때문이다. 이의 결과로 DNSLR의 정당성이 입증된 것이다. 그림에서 보듯이 이동과 패킷 호출 방향이 국지성을 유지하게 되면 3 방안 모두 네트워크 점유

시간이 감소함을 보이고 있다. 이러한 현상은 이동 및 패킷 호출 횟수가 증가할 경우에도 마찬가지로 보이고 있다. 본 시뮬레이션 환경이 일정한 시간에 고정된 패킷 생성 수를 기반으로 수행 되었기 때문에 네트워크 점유시간이 곧 이동 시스템의 성능을 나타낸다고 볼 수 있다.

5.3 직접경로설정 결과

[그림 5]는 두 방안의 직접경로 설정 시뮬레이션 결과를 보여준다. 그림에서 본 바와 같이 DNSIMHP방안이 DNSLR에 비해 현저하게 직접경로 설정 횟수가 낮다. 이는 DNSIMHP에서는 사후 위치정보 캐쉬방법(첫 번째 패킷을 MA(HA) 혹은 PFA)로 보낸 다음 이로부터 위치정보를 전달 받는 방법을 사용한 반면에, DNSLR은 사전 위치정보 캐쉬방법(DNS 질의/응답 시 위치정보를 캐쉬하는 방법)을 사용했기 때문이다. 따라서, DNSIMHP에서는 지연된 위치 정보 전파로 인하여 직접경로 설정 횟수가 감소되고, 이는 캐쉬 적중율(캐쉬가 가지고 있는 COA에 이동호스트가 현재 위치할 확률) 낮아지는 요인이 된다. 반면에, DNSLR에서는 DNS

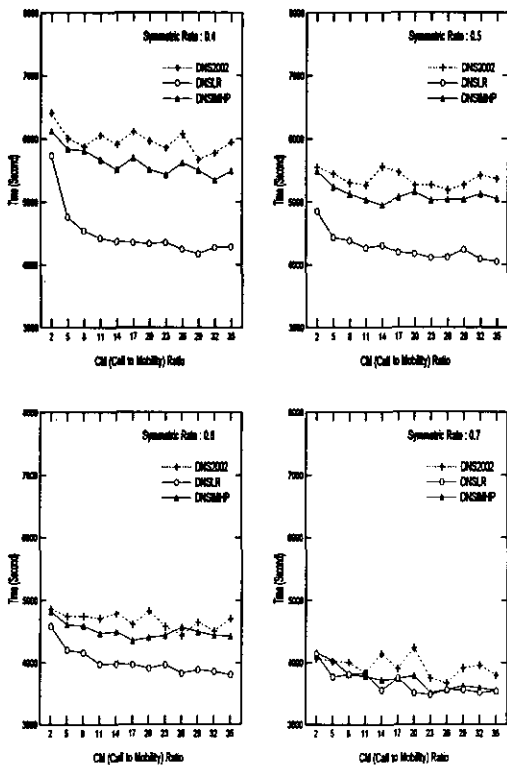


그림 4 각 방안별 네트워크 점유시간

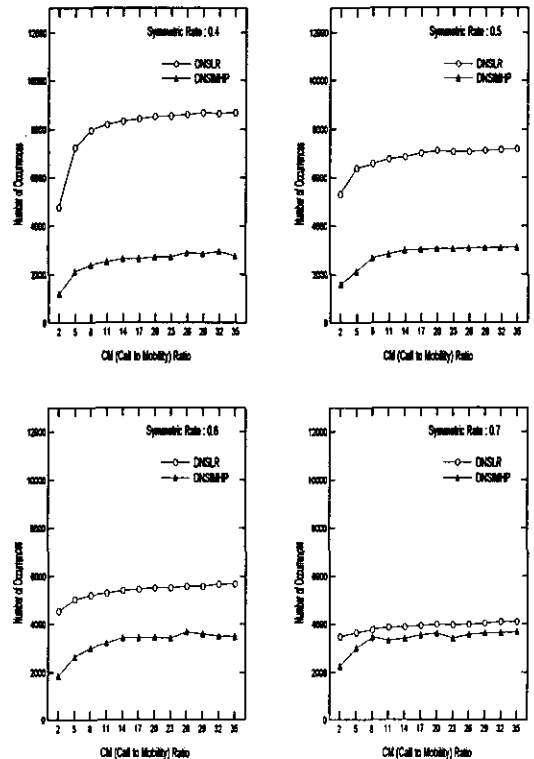


그림 5 직접경로 설정 횟수

질의/응답시 캐쉬정보가 피기백 되기 때문에 상대적으로 적절경로 설정 횟수가 증가하고 캐쉬 적중율이 DNSI-MHP보다 높다.

6. 향후 연구분야

본 연구에서는 인터넷 환경에서 DNSLR이 이동호스트의 위치정보전달과 경로설정을 최적화 시키는 개선된 해결책이라는 사실만을 보여주었다. 하지만, 향후 연구되어야 할 분야가 이 연구를 통해서 분명해 졌다.

- DNSLR을 실제로 구성된 네트워크에 구현하여 시험할 필요가 있다. 시뮬레이션에서 이동컴퓨팅환경을 고려하고자 노력을 했지만, 해결되어야 할 더 많은 현실적인 문제와 평가에 대한 요구가 존재할 수 있다. 특히, 실 네트워크에 DNSLR 적용에 따른 DNS부하 분석 및 평가 모델에 대한 연구가 보다 구체적인 DNSLR의 현실성 검증을 위해 필요하다.

- 이동성을 지원하는 DNS프로토콜에 대한 더 심도 있는 연구가 필요하다. 현재의 DNS프로토콜은 고정호스트를 전제로 설계된 것이다. 따라서, 현재의 DNS프로토콜은 호스트에게 현실적인 성능 제공을 위해 장시간의 TTL을 기반으로 정적 캐쉬기법을 근간으로 동작한다. 이는 동적특성을 지닌 이동컴퓨팅환경에는 적합하지 않다. 결과적으로 DNS가 호스트 이동성을 지원하기 위해서는 이동호스트의 동적특성을 반영하여 보다 견고한 DNS인프라 구조정의 및 이에 적용적인 프로토콜 연구가 추가적으로 필요하다.

7. 결론

본 논문에서는 이동IP가 위치수정 비용을 최소로 줄이면서 기존의 고정 인터넷에 근접하도록 대부분의 통신에 대해 최적경로를 제공할 수 있는 최적화된 방안을 개발하고자 하였다. 제안된 DNSLR은 기본방안[4,5]에 위치와 경로설정 최적화를 위해 이동성을 지원하는 DNS개념을 추가하였다. 이는 모든 통신이 호스트 이름으로 시작한다는 가정이 전제가 된다. 이 논문에서 행해진 연구로부터 얻은 결론을 다음과 같이 요약할 수 있다.

- 인터넷 응용을 동작시킬 때 절대적으로 사용되는 DNS기능을 이동호스트의 위치설정 과정에 적용함으로써 탁월하게 위치수정과 경로설정 부담을 완화 시켰음을 보였다.

- 송신호스트가 패킷 호출을 시작할 시점에 DNS 프로토콜을 이용하여 위치정보를 전과 받는 방식(Call-Initiated)이 불필요한 위치수정을 감소시키고 효율적인

경로설정에 결정적인 역할을 하였다.

- 네트워크 개체의 특성을 고려하여 네트워크 기반구조와 이동호스트 사이에 위치정보와 경로설정처리 의무를 적절하게 분배 시키는 것이 확장성 있고 효율적인 시스템을 생성하는데 도움이 되었다.

- 위치정보 캐쉬는 유효시간이 지나면 재설정되어야 한다. 그렇지 않으면 패킷이 잘못 방향전환되어 전달되었다.

- 본 논문에서 제안한 방안(DNSLR)이 RFC 2002[4, 5]와 참고문헌[8]보다 시스템 성능측면에서 탁월하게 우수함을 보였다.

참고 문헌

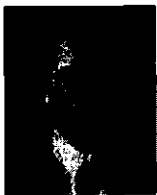
- [1] Andrew Myles, David B. Joheson, and Charles Perkins, A Mobile Host Protocol Supporting Route Optimization and Authentication, *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 5, pp. 839-849, Jun. 1995
- [2] Charles Perkins, Andrew Myles, David B. Johnson, IMPH: A Mobile Host Protocol for the Internet, *Computer Network and ISDN Systems*, Vol. 27, No. 3, pp. 479-492, Dec. 1994
- [3] Gihwan cho, A Location Management Scheme Supporting Route Optimization for Mobile Hosts, *Journal of Network and Systems Management*, Vol. 6, No. 1, pp. 31-50, March. 1998
- [4] C. Perkins, IP Mobility Support, *Internet Request for Comments RFC2002*, 1996
- [5] C. Perkins, IP Mobility Support Version 2, *Internet Draft draft-ietf-mobileip-v2-00*, November 1997
- [6] P. Calhoun, Charles Perkins, Tunnel Establishment Protocol, *Internet Draft, draft-ietf-mobileip-calhoun-01*, 1998
- [7] Charles Perkins, David B. Johnson, Special Tunnels for Mobile IP, *Internet Draft, draft-ietf-mobileip-spectun-00*, 1997
- [8] David. B. Johnson, Charles Perkins, Routing Optimization in Mobile IP, *Internet Draft draft-ietf-mobileip-optim-08*, 1999
- [9] A. Heybey, The Network Simulator Version 2.1, MIT LCS Advanced Network Architecture Group, Sept. 1990
- [10] P. Mockapetris, Domain Names-Concepts and Facilities, RFC 1034, 1987
- [11] P. Mockapetris, Domain Names-Implementation and Specification, RFC 1035, 1987
- [12] C. Everhart, L. Mamakos, New DNS RR Definitions, RFC 1183, October 1990
- [13] Gulbrandsen, P. Vixie, A DNS RR for specifying

- the location of services(DNS SRV), October 1996
- [14] P. Vixie, S.Thomson, Y. Rekhter, J. Bound. Dynamic Updates in the Domain Name System(DNS UPTADE), RFC2136, April 1997
- [15] David. B. Johnson, Ubiquitous Mobile Host Internetworking, in Proc. 4th Workshop on Workstation Operating Systems, Oct. 1993
- [16] J. Ioannidis, D. Duchamp, and G. Q. Maguire. IP-based Protocols for Mobile Internetworking. *In Proc. of ACM SIGCOMM symposium on Communication, Architecture and Protocols.* pp. 893-907, Sept.1991
- [17] I.J. Jo, DNS Based Location Management Scheme Supporting Route Optimization for Mobile Host, Ph.D. Thesis, University of Ah-Ju,1999



조 인 준

1982년 전남대학교 계산통계산학과 졸업(학사). 1985년 전남대학교 계산통계학과 대학원졸업(석사). 1999년 아주대학교 컴퓨터 공학과 대학원 졸업(박사). 1990년 정보처리 기술사(전산 조직 응용). 1983년 ~ 1994년 한국전자통신연구원(선임연구원). 1994년 3월 ~ 현재 배재대학교 컴퓨터 공학과(부교수). 관심 분야는 정보.통신 Security, 컴퓨터 네트워크(이동컴퓨팅), 전산조직응용



조 기 환

1985년 전남대학교 계산통계학과 졸업(학사). 1987년 서울대학교 계산통계학과 졸업(석사). 1996년 영국 Newcastle 대학교 전산학과 졸업(박사). 1987년 ~ 1997년 한국전자통신연구원(선임연구원). 1997년 ~ 1999년 목포대학교 컴퓨터과 학과 전임강사. 1999년 ~ 현재 전북대학교 전자정보공학부 조교수. 관심분야는 이동컴퓨팅, 무선인터넷, 분산처리시스템, 컴퓨터통신