

主題

초고속 인터넷 서비스를 위한 통합 네이밍 서비스 기술

한국정보통신대학원 최준균

차례

- I. 서론
- II. 네이밍 서비스 관련 표준화 현황
- III. ATM 네이밍 서비스 요구사항
- IV. ATM-DNS 구조 및 망 구성 방안
- V. ATM-DNS를 이용한 Client/Server 통신 방식
- VI. IP-DNS와 연동구조
- VII. ATM-DNS에서 정보 검색 절차
- VIII. ATM-DNS를 이용한 네이밍 서비스 제공 방안
- IX. 결론

요약

본 고에서는 초고속 인터넷 서비스를 위해 기존 IP 도메인 네임 서비스 체계를 초고속 ATM 망에 확장하는 방안에 대하여 설명한다. 이를 위해 먼저 ATM 네이밍 서비스 구조와 통신방식에 대하여 검토하며, IP-DNS와 연동 방안을 설명한다. 다음으로 ANS (ATM Name System)를 이용한 정보 검색과 네이밍 서비스 제공 방안을 설명한다.

I. 서론

본 고에서는 단말 호스트를 식별하기 위해 그 호스트가 ATM 통신망 환경에 있는 것일 지라도 IP 주소만으로 단말을 식별할 수 있던 제약에서 벗어나

ATM 단말의 경우 IP 주소만이 아니라 ATM주소로도 식별을 위한 호스트 네임을 알 수 있도록 해주며, 또한 호스트 네임으로 단말의 ATM 주소를 바로 알 수 있도록 해 주는 ATM-DNS(ATM Domain Name Service)를 이용해 인터넷 서비스를 제공하는 방안에 대해 기술한다.

정보화 사회에서 국가의 정보통신망은 물리적 허부망으로 ATM 기술을 사용하여 고속의 통신 인프라를 제공하고 그 위에 다양한 상위 계층 프로토콜을 필요에 따라 수용함으로써 음성, 데이터 및 실시간 화상 정보까지 각각에 대한 QoS(Quality of Service)를 만족시키면서 서비스를 제공한다.

이러한 ATM 통신 기술은 국제표준화 기구인 ITU-T의 관리 하에 90년대 초에 12개의 기초 권고안이 작성된 이후 약 10년간의 표준화 활동과 '91년부터 진행된 ATM Forum에서의 표준화 활동에 의해 대부분의 기본 권고안들이 완성되었으며 많은

제품들이 생산되어 LAN(Local Area Network) 환경과 초고속 기간망에서 PVC(Permanent Virtual Circuit) 형태로 사용되고 있다.

현재 ATM 통신망은 그 성공 및 확산을 위해 신규서비스의 창출과 기존서비스의 가격경쟁력 있는 수용 등을 필요로 하는데 이를 위해 Native ATM API (Application Programming Interface), ATM 망에서의 인터넷 수용 방안, ATM 망에서의 음성 전화서비스 수용 방안, ATM 망에서의 화상서비스 수용 방안 등에 대해 집중적으로 연구가 진행되고 있으며 첫번째 대상으로 인터넷 서비스를 고려하고 있다.

컴퓨터 통신을 위해 사용되기 시작한 인터넷은 '90년대 초 WWW(World Wide Web)의 등장에 힘입어 최근 폭발적인 성장률을 보이고 있는데, 인터넷에 대한 표준화를 담당하는 IETF를 중심으로 단순한 비실시간 데이터 전송 위주였던 인터넷 서비스에서 탈피하여 음성 및 실시간 화상 데이터 등 QoS를 요구하는 서비스를 포함하고자 하는 노력이 진행되고 있으며, 응용면에서도 디지털 도서관, 가상환경, 원격교육, 분산컴퓨팅, 원격진료, 전자상거래 등 다양한 서비스가 생겨남에 따라 트래픽 양에서 음성 전화서비스를 능가할 것으로 평가되고 있다.

이러한 사용자 및 트래픽의 증가는 현재의 인터넷의 네트워크 측면에서 새로운 하부구조를 요구하는데 그 구축 방안으로 크게 두 가지 방법이 거론되고 있다. 첫째는 고속 라우터를 이용해 데이터를 신속히 처리하고 이들 라우터 간에 고속의 전용선을 이용해 연결하는 방법인데 이 방법은 기존의 인터넷 구성 방안과 동일하므로 현재 인터넷 환경에 대한 커다란 변화가 필요 없는 반면 현재 구축되고 있는 ATM 기반의 초고속정보통신망의 활용 측면에서 별 연관을 갖지 못한다. 둘째로 ATM 기반의 초고속정보통신망을 활용하여 통신 하부구조를 제공 받으며 그 위에서 인터넷을 운용하는 방안으로 ATM

계층 스위칭 기능을 이용해 고속으로 IP 패킷을 전달할 수 있고 QoS 보장이 용이하며, 전용선 아닌 ATM의 가상회선을 사용하므로 비용면에서 더 저렴한 가격으로 융통성 있게 ATM 통신망에서 제공하는 다양한 전송속도를 지원 받을 수 있는 점 등 많은 장점을 가지나 IP 주소를 가지고 ATM 주소를 찾아내는 주소해석(address resolution) 기능 또는 IP 전송을 위한 새로운 신호방식을 필요로 한다. 따라서 종단 단말 인식을 위해 IP 또는 ATM 주소에 상관없이 호스트 네이밍으로도 인터넷 서비스가 가능하도록 해 주는 ATM-DNS 기능에 대한 요구가 계속 확대될 것이다.

II. 네이밍 서비스 관련 표준화 현황

IP-DNS (Domain Name Service) 체계를 초고속 ATM 망에 확장하기 위한 ATM-DNS의 표준화는 1996년 ATM Name System (ANS) 규격 1.0의 발표로 시작되었다. ANS는 Internet Engineering Task Force (IETF)의 Domain Name System (DNS)를 확장한 것으로 zone 보안, 동적 update, prompt notification과 증가하는 zone 전달 능력 확장을 위한 내용이 포함되어 ANS 2.0이 완성되었다.

ANS는 native ATM 서비스를 위해 호스트 이름을 통한 ATM 망 접근을 허용하기 위한 서비스이다. ANS 서비스 대상은 native ATM 서비스를 이용하는 데스크탑 사용자(예, ATM NIC 또는 ADSL로 접근하는 사용자)를 위해 서비스를 제공한다. ANS는 native ATM 응용에서 착신 호스트 이름을 통한 ATM 주소 정보 추출을 위해 ANS 서버로 질의를 발생하는 것으로 시작되며, 서버로부터의 주소 결과를 통해 호 설정을 하여 응용 서비스가 개시된다. 이와 같은 서비스 절차 및 호스트들의 이름 체계는 인터넷에서 사용되는 IP-DNS에 기반한

다.

현재 네임 서비스의 빠른 개발을 지원하고 더 높은 수준의 디렉토리 서비스를 위한 하부구조를 제공한다. ATM 어플리케이션은 다양한 종류의 디렉토리 서비스를 요구한다.

ANS는 native ATM 응용이고 API를 이용한다. ATM 클라이언트는 ANS 서버와 통신하기 위하여 ATM SVC를 이용한다. ANS 1.0은 query와 reply할 때 ATM SVC 혹은 UDP를 이용하고 zone 전달 메시지는 TCP를 이용한다. 도메인 네임을 ATM 어드레스로 전환은 현재 top level domain name아래서 수행되고 ATM address를 domain name으로의 전환은 atm.int로 끝나는 새로운 도메인을 이용한다. 한편 IETF 에서 도메인 네임 서비스와 관련된 표준화 현황을 보면 먼저 DNS Incremental Transfer, Notification, and Dynamic Update (dnsind) Working Group은 미래 DNS 프로토콜을 위해 추가할 기능으로 다음의 3가지 사항에 대하여 연구를 진행하고 있다.

- 증가하는 Zone 전달 : 일부 zone 파일의 크기가 아주 커짐에 따라서, zone의 변화된 subset 만의 전달을 부가적으로 허용하는 프로토콜은 결정적인 서버에 전체 트래픽과 부하를 감소한다.
- Notification 변화 : 큰 시간 간격동안 적어도 한 개의 secondary가 primary와 일치하지 않는 데이터를 가지고 있을 수 있다. 제안된 "notify" 메커니즘(여기서 primary가 알려진 secondary에게 메시지를 전달한다)은 zone에 있는 데이터의 연속성과 비교하여 서버의 빠른 수렴을 돕는다.
- 동적 update : 큰 zone의 작은 부분을 자주 갱신하고 이런 갱신 증가되도록 할 필요성을 인식한다.

다음으로 DNS Operations (dnsop) Working Group은 DNS name 서버의 기능과 DNS zone 파일의 관리를 위한 지침을 연구 중이다. 이러한 지침은 DNS 도메인의 운영자와 관리자에 의해 DNS 프로토콜의 구현에 관계한 기술적 정보를 제공할 것이다.

Domain Name System Security (dnssec) Working Group은 DNS의 동적 update 기능을 보호하기 위한 안전한 DNS 프로토콜에 대하여 연구하며, 특별히, 갱신 처리의 응답을 감지하고 갱신 처리를 명령할 수 있는 방안을 검토 중이다. 현재 활발히 검토 중인 이슈는 다음과 같다.

- name과 zone 모두를 생성, 삭제, 갱신하는 범위
- zone 전달 동안 동적으로 갱신해야 하는 name의 보호
- wildcard scope 안에서 더 명백한 name을 위한 KEY resource record의 범위
- 제안된 만료 resource record의 이용 혹은 관계

1998년 11월에 만들어진 ICANN(Internet Corporation for Assigned Names and Numbers)은 미국 정부 혹은 계약자와 지원자들에게 의해 지금까지 수행되었던 일련의 인터넷 기술적 관리 기능을 조정하는 비영리, 사적 부문 단체이다. 여기서는 프로토콜 파라미터의 할당, 도메인 네임 시스템의 관리, IP 어드레스 공간의 할당, 루트 서버 시스템의 관리를 조정하는 기능을 수행한다.

III. ATM 네이밍 서비스 요구사항

ANS(ATM Name System)은 DNS secu-

urity(DNSSEC)와 동적 update [DNSDYN] 확장을 포함하는 인터넷 도메인 name 시스템 (DNS)(RFC1034, RFC1035)을 기반으로 하며 다음과 같은 특징을 갖고 있다.

- DNS가 end system에서 광범위하게 쓰이고 있다.
- 대부분의 국가에서 DNS domain 등록 절차를 구현하고 있다.
- 사용자들이 DNS를 잘 알고 있다.
- Name space가 적절히 할당되면 확장 용이하다.
- DNS 구현이 CPU와 메모리 관점에서 효율적이다.
- secondary server 메커니즘을 통해 복구 능력이 있다.
- query에 대한 대기 시간이 짧다.
- 동적 update와 보안 부분에 대한 확장이 진행 중이다.
- ATM을 위한 새로운 데이터베이스 객체의 증설이 용이하다.
- Dual host에게 IP와 native ATM간의 선택을 허용한다.

도메인 네임에서 ATM 주소, ATM 주소에서 도메인 네임으로 서비스 변환은 여러 구현 방법에 의해 제공 가능하다. 하나는 서비스 요구 단말이 IP나 ATM에 대한 공통 주소를 갖고 서비스를 제공하도록 하고, 또 다른 방법은 망에서 DNS/ANS로의 변환을 각 단말에게 투명하게 제공하도록 공통 DB를 구축하는 것 등이다. ANS에서 디렉토리 서비스를 위한 요구조건을 보면 다음과 같다.

- 복구 기능. Single point failure가 없어야 한다. (회복력)
- 저장 효율을 포함하는 효율적인 구현

- 계층 naming 구조(확장성)
- 질의에 대한 짧은 대기 시간
- 순간적인 데이터 불일치가 발생하더라도 ANS에 있는 정보에 손상이 없어야 한다.
- 일시적 서비스 중단 지원 ? "busy signal"

다음으로 ATM 네이밍 서비스를 위한 요구사항을 보면 다음과 같다.

- 지리적 영역, global 서비스, 네트워크 사업자들을 지원하기 위한 계층적 구조
- 네임, 키워드, 넘버 및 또는 그들의 조합으로 사용자, 터미널, 그리고 서비스 식별을 지원
- 사용자, 터미널 장비, 그리고 end-user 서비스는 이름, 키워드, 그리고 넘버로서 식별될 수 있어야 한다.
- 네임, 키워드, 그리고 넘버를 이용한 multi-thread query 기술
- 사용자, 터미널, 그리고 서비스의 각각 혹은 그들 그룹을 위한 Multiple assignments
- 액세스 제어 혹은 security 지원
- 액세스 제어와 security를 위한 일부 추가적인 기능이 필요하다. 예를 들면, virtual private network 혹은 closed user group을 지원하는 것.
- 만약 security가 구현되면, ATM client는 적어도 하나의 계층 구조 네임 도메인의 zone의 정확한 키가 필요하다.
- ATM 네임 서비스 데이터 베이스는 데이터 베이스 안에 인증된 public key를 저장할 수 있어야 한다.
- 로컬, 지방, 국가, 국제 도메인에 따라서 어드레스와 네임 등록의 Granularity
- ATM DNS는 만약 네임 공간이 적절히 할당되어지면 잘 조정될 수 있다.
- ATM name 서비스의 확장성은 계층적 네임

- 구조에 의해 수행된다.
- 계층적으로 분산된 ATM 네임 서비스 데이터 베이스는 ATM 네임 공간의 granularity를 줄 수 있다.
 - query의 짧은 대기시간과 효율적 저장
 - 상대적인 디렉토리 서비스 기능은 query의 짧은 대기시간과 효율적 저장을 위해 동적으로 재구성 될 수 있어야 한다.
 - 디렉토리 서비스를 위하여, end client는 하나 이상의 ATM 네임 서비스 데이터 베이스의 하나 이상의 ATM 어드레스를 알아야 한다
 - ATM 네임 서비스 데이터 베이스의 동적 updating과 자동 설정 방법
 - ATM end system 어드레스가 바뀔때, ATM name 서비스 데이터 베이스 안에 어드레스정보를 동적으로 체크하고 갱신할 수 있어야 한다.
 - ATM client는 ATM 네임 서비스 데이터 베이스로부터 바뀐 ATM 어드레스 정보를 동적으로 학습한다.
 - 또한, ATM 네임 서비스 데이터 베이스 정보에 일시적 불일치는 묵인되어야 한다.
 - 사용자, 터미널, 그리고 서비스 이동성의 지원
 - 이동중인 사용자, 터미널, 그리고 서비스는 그들의 위치 데이터 베이스를 동적으로 갱신할 수 있어야 한다.
 - 이것은 ATM 어드레스에 따라 ATM 네임 서비스 데이터 베이스에 의해 연속적으로 체크된다.
 - uni-cast, multicast, and broadcast와 같은 통신 환경 지원
 - 인터넷의 클래스 D address에 보듯이, ATM 네임 서비스는 multicast 와 broadcast 통신 환경을 식별할 수 있어야 한다.
 - ATM 네임 서비스 데이터 베이스의 적절한 등록과 회원 분배 메커니즘은 multicast와

broadcast 지원에 적용될 수 있다.

- 현존 IP-DNS 포맷과 등록 절차를 지원
 - ATM 네임 서비스는 IP 기반 네트워크와 X.500의 디렉토리 서비스 체계의 이미 현존하는 도메인 네임 기법을 이용해야만 한다
 - IP-DNS는 접속되어 있는 IP end system의 ATM address에 따라서 ATM 네임 서비스를 제공한다.
 - ATM address와 ATM 네임의 등록 절차는 IP address와 도메인 네임 서비스의 등록 절차와 유사하다.
- 현존하는 전화와 데이터 네트워크를 포함하는 non-B-ISDN으로 확장을 지원
 - ATM 응용은 ATM 네임 서비스를 이용한다.
 - 이것은 비 ATM 네트워크와 독립적이다.
 - 그러나, ATM 네임 서비스는 현존하는 전화 네트워크와 데이터 네트워크 처럼 non-B-ISDN으로 반드시 확장될 것이다.
 - 예를 들면, 현존 전화 사용자는 인터넷 유닛의 도움으로 ATM 네임에 의해 ATM client에게 전화할 수 있다.

IV. ATM-DNS 구조 및 망 구성 방안

이름 계층을 구현하는 ATM-DNS는 개념적으로 독립되어 있는 두가지 측면을 지닌다.

첫째는 추상적인 것으로서, ATM-DNS는 이름에 대한 권한을 위임하기 위한 이름 구문과 규칙을 지정한다. 둘째는, 구체적인 것으로서, ATM-DNS는 이름을 주소로 효율적으로 매핑해 주는 분산 컴퓨팅 시스템을 구현하는 것에 대해 지정한다.

IP-DNS와 마찬가지로 ATM-DNS는 도메인 이름(domain name)이라 알려진 계층적 이름 할당법을 사용한다. 도메인 이름은 구분자인 마침표에 의해서 분리된 부이름(subname)들의 열로 구성

된다. 도메인 이름에서 지역 레이블은 처음에 그리고 최상위 도메인은 마지막에 적는다. 개념적으로, 최상위 수준의 이름들은 두 가지 완전히 다른 이름 계층들 (즉, 지리적 또는 조직적 이름 계층)을 허용한다. 지리적 기법의 경우는 기계들을 국가별로 구분한다.

한국에 있는 기계들은 KR이라는 최상위 수준 도메인에 속한다. 어떤 국가가 도메인 이름 시스템에 기계를 등록하고자 하면, 중앙 관장 기관에서는 그 국가의 레이블로서 국제 표준의 두 문자로 구성된 식별자를 갖는 새로운 최상위 수준 도메인을 그 국가에 할당해 준다.

client들로 하여금 여러 종류의 엔트리들 중에서 구분할 수 있도록 하기 위해서, 시스템에 저장되어 있는 이름이 부여된 각 항목(item)에 그 항목이 기계, 우편함, 또는 사용자의 주소인지 여부를 지정하는 종류를 할당해 준다. client가 ATM-DNS에게 이름을 찾아달라고 요청할 때 원하는 종류를 지정해야만 한다. 예를 들어, E-mail프로그램에서 이름을 찾기 위해서 ATM-DNS를 사용할 때는 원하는 답이 mail exchanger주소여야만 한다는 것을 지정해 준다. 원격 로그인 응용의 경우에는 목적지의 ATM주소가 찾고자 하는 것이라고 지정해 준다. 이와 같이 주어진 이름은 ATM-DNS에서 하나 이상의 항목으로 매핑될 수 있다. 따라서 client는 이름을 찾고자 할 때 원하는 객체의 종류를 지정하고, server는 그러한 종류의 객체들을 결과로서 돌려준다.

이름을 주소로 매핑하는 도메인 기법은 네임 서버라 불리는 독립적이고 협력적인 시스템들로 구성된다. 네임 서버는 도메인 이름에서 ATM주소로 매핑해주는 이름-주소 번역을 제공하는 서버 프로그램이다. 흔히 서버 소프트웨어는 전용 프로세서에서 수행되며 이와 같은 기계를 네임 서버라 한다. name resolver라 불리는 client소프트웨어는 이름을 번역할 때 하나 이상의 네임 서버들을 사용

한다.

ATM 네임 서버 들을 트리 구조로 배열할 수 있는데, 이 트리의 루트는 최상위 도메인을 인식하고 어떤 서버가 그 도메인을 해결하는지를 알고 있는 서버이다. 해결할 이름이 주어지면, 루트는 그 이름에 대한 정확한 서버를 선택할 수 있다. 다음 수준에서 각 네임 서버는 최상위 도메인(예, kr)에 대한 답을 제공한다. 이 수준의 각 서버는 그 도메인의 서브 도메인들을 해결할 수 있는 서버들을 알고 있다. 트리의 세번째 수준에서 네임 서버는 서브 도메인(예, kr밑의 ac)에 대한 답을 제공한다. 이와 같은 개념적인 트리는 서브도메인이 정의되어 있는 각 수준에서의 서버에서 계속된다.

개념적 트리의 링크들은 물리적 망 연결을 의미하지 않는다. 대신 어떤 서버가 알고 있으며 접근할 수 있는 다른 네임 서버들을 나타낸다. 서버들은 ATM 네트워크 상의 임의의 장소에 위치할 수 있다. 만일 도메인 시스템내의 서버들이 이와 같은 간단한 모델이 제안하는 것처럼 동작한다면, 연결성과 권한간의 관계는 단순화 될 수 있다. 즉 권한이 서브도메인에 위임될 때, 이를 요청한 조직은 그 서브 도메인에 대한 ATM 네임 서버를 설정하고 서버 트리에 이 서버를 연결해 준다. 그러나 실질적으로 한 물리적 서버가 이름 계층의 많은 부분에 대한 정보를 가질 수 있기 때문에 서버 트리는 적은 수의 수준만으로 구성될 수 있다.

V. ATM-DNS를 이용한 Client/Server 통신 방식

ATM-DNS의 Client 쪽은 Resolver라고 불리어지며, 일반적으로 서버 쪽의 도메인 네임의 정보를 조회한다. ATM-DNS 서버는 Resolver의 조회에 응답한다. ATM-DNS의 일반적인 구성도는 (그림 1)과 같다.

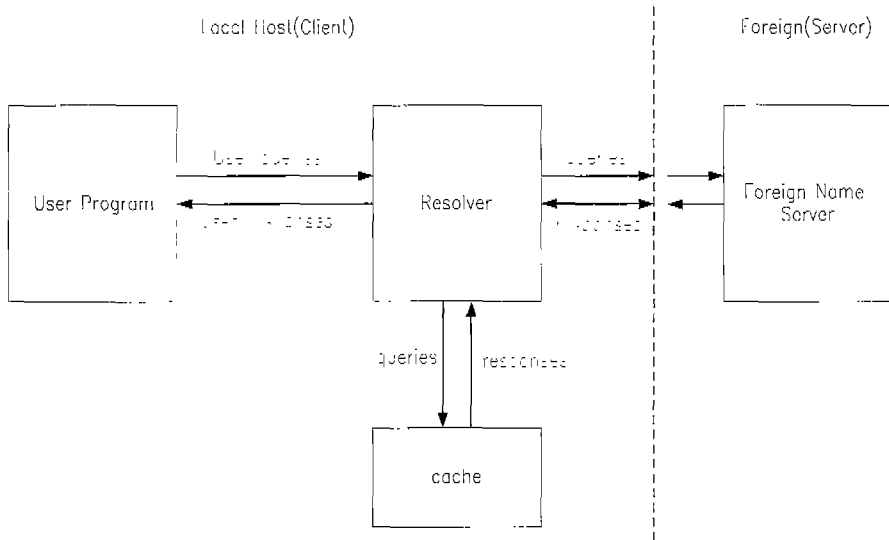


그림 1. ATM-DNS 구성도

User program은 resolver를 통하여 domain name space와 상호 작용한다. Resolver는 user queries에 의해 foreign name server와 local cache로 queries를 경유하여 획득된 정보로 응답한다. Resolver는 여러 개의 질의를 서로 다른 여러 foreign name server에게 특유의 user query에 대한 응답으로 만들수 있고, user query에 대한 응답은 여러 network access와 임의의 시간을 포함한다. Foreign name server에 대한 질의와 일치하는 응답은 표준화된 포맷(datagram)을 갖는다. 확실히, name server는 특정한 machine이나 프로세스, large timeshared host에서의 프로세스들에 독립적이다.

ATM-DNS clients는 ATM server와 통신하기 위해서 ATM SVC를 사용한다. 또한 ATM-DNS server사이의 질의와 응답을 위해서는 ATM SVC또는 UDP를 사용하고 신뢰성 있는 zone transfer message의 전송을 위해 TCP를 사용한다. 실제 ATM-DNS를 이용하는 서비스 절

차는 다음 (그림 2)와 같다.

ATM-DNS 호스트는 native ATM 서비스를 이용하는 ATM응용과 ATM 응용에서 발생하는 이름 서비스 요구를 처리하는 ATM-DNS Client 기능으로 구분된다. ATM-DNS 응용에서 웹 브라우저와 같이 통신하고자 하는 착신 단말의 이름을 사용자로부터 받고, 이 이름과 상응하는 ATM 주소를 얻기 위해 ATM-Client를 활성화 한다. ATM-DNS Client는 해당 이름을 입력 파라미터로 하는 질의를 발생하여 미리 설정된 ATM-DNS 서버와의 연결을 통해 전송한다. 이를 수신한 ATM-DNS 서버는 자신이 유지하는 주소 정보를 통해 이를 추출하려 하며, 해당 정보가 있을 경우 바로 ATM-DNS Client로 응답하게 된다. 만일 주소 정보가 없는 경우 해당 ATM-DNS 서버는 다시 ATM-DNS Client로 동작하며 상위 서버에게 관련 정보에 대한 질의를 발생한다. ATM-DNS 상위 서버는 최종적으로 착신 단말이 속한 zone의 ATM-DNS 서버가 될 것이며, 해당 주소 정보를 질의가 전송된 경로의 역순으로 응답을 전송하게 된

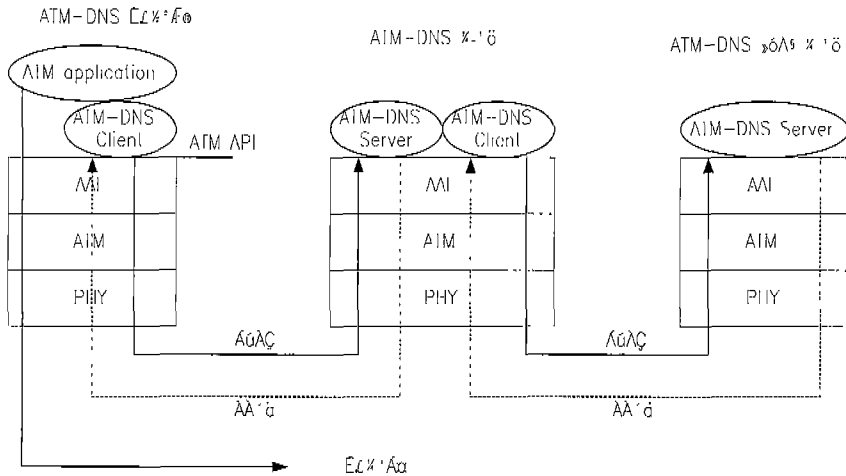


그림 2. ATM-DNS 수행 절차

다. 최종적으로 이를 수신한 ATM-DNS Client는 해당 정보를 ATM 응용에게 전달하며, ATM 응용은 이 주소 정보를 통해 착신 호스트와 호 설정을 시도하게 된다.

Basic directory service는 도메인 이름을 ATM 주소로 변환하고 ATM 주소를 도메인 이름으로 변환한다. 도메인 이름에서 ATM 주소로의 변환은 현재의 top level domain name하에서 수행되고 ATM 주소에서 도메인 이름으로의 변환은 ATMA.INT의 새로운 도메인 이름을 사용하여 수행된다. 또한 ATM 이름과 주소이외의 정보가 ATM-DNS 데이터베이스에 추가될 수 있다. ATM환경에서 DNS프로토콜이 사용될 때, ATM 네트워크는 datagram 서비스를 지원하지 않기 때문에 ATM-DNS 질의와 응답은 client와 server 사이에 설정되는 ATM 가상채널 상에서 수행된다. 기본적으로 ATM-DNS client는 질의를 할 때 (query의 header안에 RD bit를 1로 설정하는 것에 의해서) query의 recursive processing을 요구하므로 ATM-DNS name server는 recursion을 지원해야 한다. client가 처음 동작을 시작할 때는 한 개 또는 그 이상의 ATM-DNS

server의 ATM 주소를 갖고 있어야 하므로, ATM-DNS client는 ATM-DNS server의 ATM 주소를 찾기 위해 다음 2개의 메커니즘 중 하나를 사용한다.

- ILMI(Integrated Local Management Interface)에 의해 ATM-DNS server 주소 획득
- Well-known ATM-DNS 주소 획득

만일 ATM-DNS server 주소가 ILMI로부터 얻어질 수 없거나 client가 ILMI가 제공하는 server 주소로 VC를 설정할 수 없을 때 well-known address가 사용될 것이다. Well known address는 02C500790000000000000000000000.00A03E000002.00. 이다.

한 개 또는 그 이상의 ATM-DNS server의 ATM 주소를 얻은 후에 ATM-DNS client는 ATM-DNS request를 보내고 ATM-DNS response를 받기 위해서 server중 어느 한 개로 연결을 설정할 것이다. client는 [ATMNAS]에 의해 규정된 API에 의해 접근할 수 있는 UNI

signaling protocol을 사용하여 동적으로 연결을 설정하고 해제한다. client는 어떤 기간동안의 휴지 상태 이후에 연결을 해제한다. server도 다른 client에게 서비스를 제공하기 위해서 사용되지 않는 연결을 해제한다.

ATM directory service로 접속하기 위한 API의 기본 기능은 도메인 이름으로 ATM 주소를 알아내고 ATM 주소로 도메인 이름을 알아내는 서비스를 포함하는 API로의 call을 수행하는 것이다. Starting point에서 사용될 수 있는 API의 한가지 예로서 UNIX operating system[BSD4.3]에 의해 제공되는 것을 들 수 있다. 이 API의 예는 입력으로서 host 이름을 받아서 그 host 이름에 관한 ATM-DNS 데이터베이스 정보를 리턴하는 gethostbyname_atmnsap와 gethostbyname_E.164서비스를 제공한다. 수신된 메시지에 host address type, host address length, host address와 host의 alias name 목록과 같은 기타 정보가 포함될 것이다. 또한API의 예로서 입력으로 host address type, host address length와 host address를 받는 gethostbyaddr을 제공한다. 이 서비스는 host address name과 선택적으로 기타 ATM-DNS 정보를 되 돌려준다.

ATM-DNS와의 인터페이스는 IP protocol family address를 결정하기 위해 현재 사용되는 BSD 4.3 인터페이스와 동일하게 될 수 있다. 이것은 뒤따르는 call들과 같이 구성된다. (여기서는 가장 일반적인 'C'-syntax로서 정의되었다.)

```
extern int h_errno;
```

```
struct hostent *gethostbyname_atmnsap
    (name)
    char *name;
```

```
struct hostent *gethostbyname_e164
    (name)
    char *name;

struct hostent *gethostbyaddr(addr,
    len, type)
    char *addr: int len, type;

struct hostent *gethostent()

sethostent(stayopen)
int stayopen;

endhostent()

herror(string)
char *string;
```

where

```
struct hostent
{
    char *h_name; /* official name of host */
    char **h_aliases; /* alias list */
    int h_addrtype; /* host address type */
    int h_length; /* length of address */
    char **h_addr_list; /* list of addresses from
        nameserver */
};
```

```
#define h_addr h_addr_list[0] /* address,
for backward compatibility*/
```

이 구조체의 멤버:

```
h_name 호스트의 공식 이름
h_aliases 호스트에 대한 다른 이름의 배열
```

h_addrtype 리턴되는 주소형식
 h_length 바이트단위의 주소길이
 h_addr_list 호스트에 대한 네트워크 주소의 배열.
 호스트 주소는 네트워크 바이트 순서로 리턴된다.
 h_addr h_addr_list에 있는 처음 주소

sethostent, endhostent, gethostent와 herror call들은 수정되지 않았다. 현재 사용되는 h_addrtype은 항상 응답으로서 AF_INET 수반하고 gethostbyaddr function의 type argument는 또한 모든 경우에 있어서 AF_INET가 되는 것이 가정되기 때문에 남아있는 두 개의 루틴을 지원하기 위해서 Structure hostent의 의미는 최근에 지원되는 주소 형식까지 확장되어야 한다. 필요한 새로운 사양은 다음과 같이 정의된다.

```
#define AF_ATMNSAP      3
#define AF_ATME164     8
```

이러한 값들은 일반적으로 socket.h안에 규정된다. gethostbyname_e164는 E.164 address를 리턴하고 gethostbyname_atmnsap는 ATM-nsap address를 리턴한다.

h_length는 h_addr_list내의 일련의 주소의 길이로 설정된다.

소켓 형태의 주소 변환이 지원된다면 일부 새로운 structure와 function들은 현재 대부분의 BSD 4.3구현에서 나타나는 것처럼 제공될 것이다.

```
struct atmnsap_addr
{
    char atmnsap[20];
}
```

```
struct atmel64_addr
{
    char atmel64[20];
}
```

```
struct atmnsap_addr *atmnsap_addr(const
char *cp);
char *atmnsap_ntoa(struct atmnsap_addr
in);
```

```
struct atmel64_addr *atmel64_addr(const
char *cp);
char *atmel64_ntoa(struct atmel64_addr
in);
```

VI. IP-DNS와 연동구조

인터넷 가입자가 예를 들어 jkchoi@icu.ac.kr 주소로 Email 이나 ftp, www를 할 경우 해당 명령어에 대한 목적지 IP 주소를 찾는 작업을 한다. 이때 ATM 망을 통할 경우 ATM 주소를 찾아야 하므로 해당 IP 이름에 대하여 목적지 IP 주소와 ATM 주소를 얻을 수 있어야 한다. 이는 IP-DNS와 ATM-DNS의 연동으로서 실현될 수 있다. 연동된 IP/ATM-DNS server에서는 기존 DNS 절차와 동일한 방식을 사용하여 IP이름과 주소, ATM 이름과 주소가 서로 연계되도록 하여야 한다. 그리하여 목적지에 이르는 ATM 주소를 사용하여 ATM 망에서 스위치형 가상채널이 설정되어 해당 응용 서비스가 연결된다.

특정 망 구성 요소에 독립적이면서 다음과 같은 서비스 요구사항을 만족시켜야 한다.

도메인 네임으로 ATM address와 IP address를 응답받고 ATM address와 IP

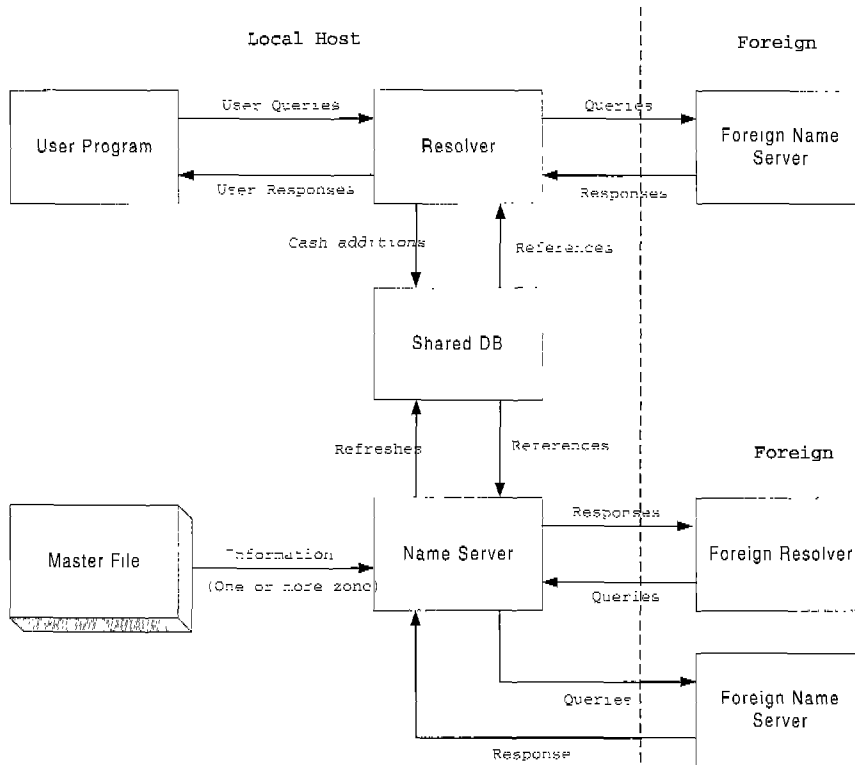


그림 3. ATM-IP 연동 시스템 구성

address로 질의하면 도메인 네임으로 응답받을 수 있는 기본적인 디렉토리 서비스가 가능해야 한다.

(그림 3)의 ATM-IP 연동 구조는 기존의 DNS와 같으나, 사용자 프로그램에서 resolver로 질의하는 부분, resolver에서 네임 서버로 질의하는 부분, 데이터 베이스에 저장하는 부분등의 포맷이 확장되어 IP 주소 및 ATM 주소까지 포함하도록 하며, local 네임 서버에서 찾을 수 없는 zone에 대한 정보는 foreign 네임 서버에 역시 의존하도록 한다.

공유 데이터 베이스는 local name server와 name server를 위해 도메인 공간 데이터를 유지한다. 공유 데이터 베이스의 내용은 name server의 주기적인 refresh 절차에 의해 관리되는 독자적인 데이터와 이전 resolver request로부터 캐쉬된 데이터의 혼합이다. 도메인 데이터의 구조와 name server와 resolver 사이의 동기화를 위한

필요성은 이 데이터 베이스의 일반적인 특성을 내포하나 실제 포맷은 로컬 구현자에 의존된다.

(그림 4)에서 정보 흐름은 역시 호스트 그룹이 최적화된 활동을 수행할 수 있도록 구성한다. 때때로 이러한 정보의 흐름은 단말 호스트가 full resolver를 구현하지 않도록 한다.

이것은 요구되는 새로운 코드의 양을 최소화하기 원하는 PC나 호스트에 적절하다. 이런 방법은 centralized 캐쉬가 높은 hit ratio를 갖는다는 조건하에서 호스트 그룹이 많은 별개의 캐쉬를 유지하기 보다는 작은 수의 캐쉬를 공유하도록 한다. 다른 경우에 resolver들은 recursive server에 위치한 resolver의 앞에 있는 stub resolver로 대체될 수 있다. 어떤 경우에도 도메인 구성요소가 항상 가능할 때 마다 신뢰성을 유지하기 위하여 중복된다.

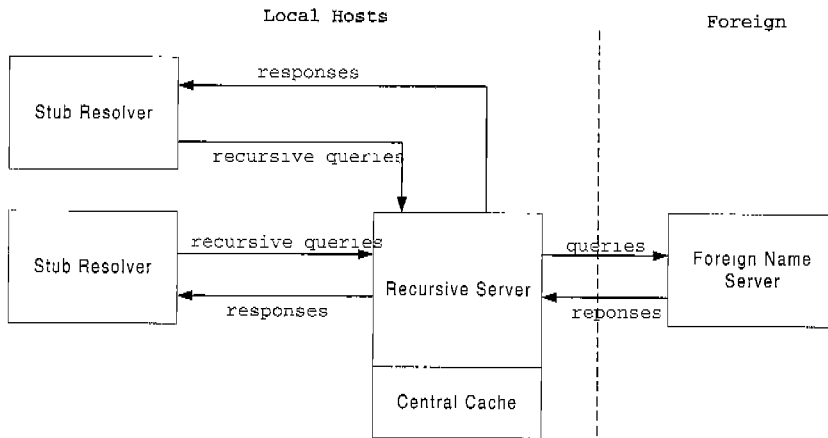


그림 4. recursive server 정보 흐름

기존 IP-DNS 호스트 표는 단순히 IP 주소와 연관된 호스트 이름의 단순한 분서 파일이다. 대부분의 유닉스 시스템에서 호스트 표는 /etc/hosts 파일 내에 존재하게 되며 표의 각 항목은 IP 주소 그리고 그 IP 주소에 해당하는 호스트 이름들을 포함하였다. ATM-DNS와 연동을 위해서는 ATM address와 ATM Name 항목을 추가하여 구성하여 준다.

이러한 연동은 기존의 IP-DNS체계에서 사용되는 hosts file, zone address data file, reverse zone address data file에 ATM 주소와 이름필드를 추가함으로써 이루어질 수 있다.

VII. ATM-DNS에서 정보 검색 절차

ATM-DNS는 기본적으로 DNS구조를 따르기 때문에 데이터베이스 구조도 역시 DNS를 기반으로 이루어져야 한다. 따라서 ATM-DNS 데이터베이스라고는 하나 이 데이터베이스의 구조는 기존의 DNS 데이터베이스를 위해 제안 구조와 요구사항을 만족시켜야 할 것이다.

개념적으로, 도메인 이름의 해결은 하향식(top-

down), 즉 root name server에서 시작해서 트리의 리프(leaf)에 위치한 server들로 전개된다. ATM domain name system을 사용하는 방법에는 두 가지가 있다. 어떤 경우든 client 소프트웨어는 해결할 이름과 그 이름의 클래스에 대한 선언, 원하는 답의 종류, 그리고 name server가 이름을 완전히 번역해야만 하는지 여부를 지정하는 코드를 포함하는 도메인 이름 질의를 만들어서 server에게 보낸다.

ATM Domain Name Server가 질의를 받으면, 자신이 권한을 갖고 있는 서브 도메인에 그 이름이 있는지 검사한다. 만일 있다면, 데이터베이스를 참조하여 그 이름에 해당하는 주소로 번역한 뒤, 질의에 덧붙여 client에게 되돌려 보낸다. 만약 이름 server가 그 이름을 완전히 해결할 수 없다면, client가 어떤 동작을 지정했는가를 검사한다. 만일 client가 recursive resolution을 요청했다면, server는 그 이름을 해결할 수 있는 ATM Domain Name Server를 접근해서 그 답을 client에게 돌려준다. client가 iterative resolution을 요청한 경우는, name server가 답을 제공할 수 없다. 대신에, client가 그 이름을 해결하기 위해서 다음에 접근할 name server를 지정

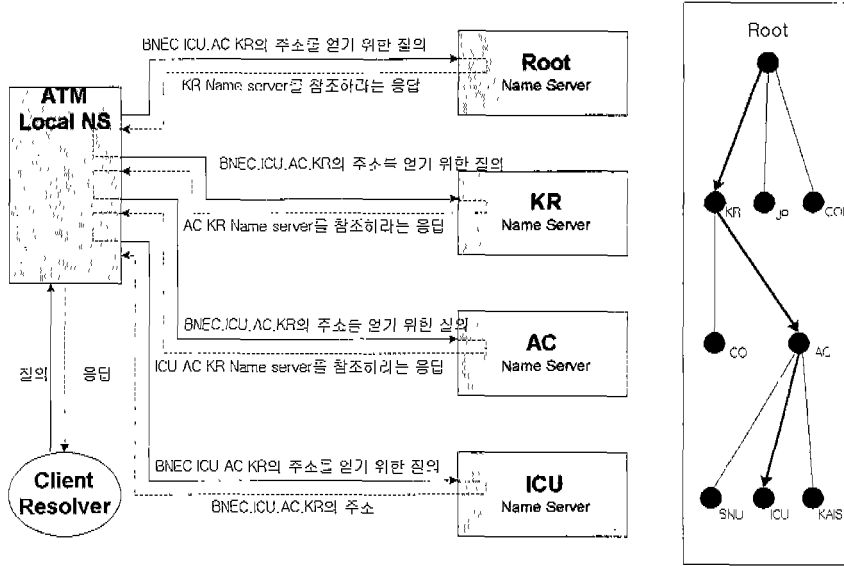


그림 5. ATM-DNS 데이터 흐름도

하는 응답을 생성한다.

client는 최소한 하나의 name server를 어떻게 접근하는지를 알고 있어야 한다. ATM Domain Name Server가 다른 server들을 접근할 수 있도록 하기 위해서, 도메인 시스템은 각 server가 최소한 하나의 root server의 주소를 알고 있을 것을 요구한다. 또한 각 server는 바로 위 도메인의 server주소를 알고 있어야 한다.

(그림 5)는 ATM-DNS에서의 데이터의 흐름과정을 보여준다. 먼저 client는 Local Name Server (Local NS)로 질의를 전달한다. Local Name server는 요청된 데이터를 도메인 내에서 찾으면 그에 해당하는 ATM 주소를 돌려주게 된다. 만약 Local Name server가 요청된 데이터를 찾지 못하면 다른 Name server로 질의를 전달한다. 최악의 경우 Local Name server의 질의는 ATM-DNS tree의 최상위에 있는 Root Name server에서 시작하여 요청된 데이터가 발견될 때까지 아래로 전달되게 된다. 매번 최상위 server에게로 질의가 전달된다면 그 server는 부하를 견디지

못하게 될 것이므로 모든 server는 이전에 발견된 데이터들을 cache영역에 등록하여 이후 동일한 요청이 있을 경우에는 상위로 질의를 전달하지 않고 내부 cache를 참조하게 된다.

기본적인 API 기능을 사용하여 도메인 네임으로 ATM 주소와 IP 주소를 얻거나 ATM 주소와 IP 주소로서 도메인 네임을 얻을 수 있다.

(그림 6)은 실제 어플리케이션이 검색하는 과정을 나타내었다. 어플리케이션이 resolver를 통하여 로컬 네임 서버에게 질의를 하면, 로컬 네임 서버는 도메인 네임으로부터 데이터를 검색하고 remote 네임 서버가 어디에 있는지 알고서 각 로컬 네임 서버에 질의한다. Name server가 remote 네임 서버에게 질의를 보내지만 실제 remote 네임 서버는 즉시 그 질의를 받지 못한다. 질의는 지연이 생기거나 하부 네트워크에 의해 손실될 경우도 고려해야 한다. Time out에 의한 중복된 질의를 포함한다.

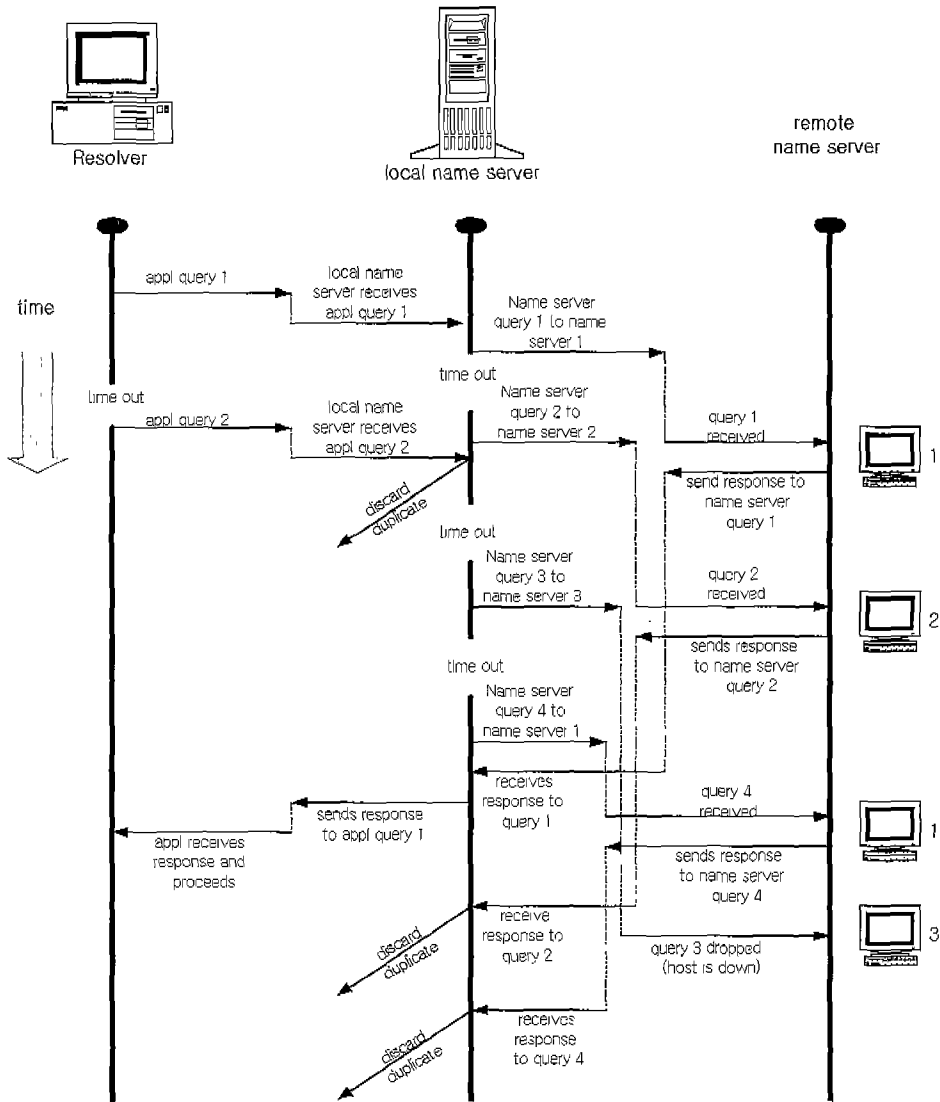


그림 6. ATM-DNS 동작 절차

VIII. ATM-DNS 를 이용한 네이밍 서비스 제공 방안

본 절에서는 호스트를 식별하기 위해 그 호스트가 ATM 통신망 환경에 있는 것일 지라도 IP주소만으로 단말을 식별할 수 있던 제약에서 벗어나 ATM 단말의 경우 IP 주소만이 아니라 ATM.로도 식별을

위한 호스트 네임을 알 수 있도록 해주며, 또한 호스트 네임으로 단말의 ATM 주소를 바로 알 수 있도록 해 주는 ATM-DNS를 이용해 인터넷 서비스를 제공하는 방안에 대해 기술한다.

프로토콜의 개방성과 인터넷으로의 접속을 용이하게 해 주는 네이밍 서비스를 사용함으로써 인해 인터넷의 사용이 폭발적으로 증가하고 있다. 반면

native ATM 서비스에 대한 요구가 점차적으로 증가하고 있지만 ATM 주소가 너무 길고 외우기 힘들면서도 인터넷에서 제공하고 있는 네이밍 서비스를 제공하지 않고 있기 때문에 native ATM 통신망으로의 접속에도 상당한 문제점을 안고 있는 실정이다. 이러한 문제에 대한 해결책으로 대두되고 있는 방안이 바로 목적지를 쉽고 간단하게 식별하고 접속할 수 있는 ATM-DNS를 사용하는 것이다. 기본적으로 ATM-DNS는 인터넷의 DNS를 기반으로 하지만 인터넷이 아닌 native ATM 통신망을 대상으로 하는 서비스라는 점에서 차이가 난다.

ATM 응용 프로그램은 다양한 형태의 디렉토리 서비스를 필요로 하며 ATM에서의 디렉토리 서비스라는 것이 다름 아닌 ATM-DNS를 사용하여 ATM 종단 단말의 단말기 이름에 대응하는 ATM 주소를 찾는 것이라고 할 수 있다. ATM-DNS는 IETF의 DNS를 ATM 망까지 확장한 개념이다.

ATM-DNS의 기능 구조는 ATM-DNS 사용자와 ATM-DNS 서버, 두 가지 구성 요소로 이루어진다. ATM-DNS 사용자는 ATM 종단 시스템의 이름에 대응하는 ATM 주소를 얻기 위해 ATM-DNS 서버에 질의를 한다. ATM-DNS 사용자와 ATM-DNS 서버 간에 질의, 응답을 교환하기 위해서는 SVC나 PVC를 사용한다. 또한 .ATM-DNS 사용자는 이러한 연결 설정을 위해 ATM API를 사용한다. ATM-DNS 서버에서는 ATM 종단 시스템의 이름과 주소 간의 매핑 정보를 관리해야 하며, 동적으로 업데이트 정보를 교환하기 위해 인접한 ATM-DNS 서버와의 연결을 유지해야 한다.

현재로서 ATM-DNS를 사용하면 인터넷 통신망과 ATM 통신망이 각각 독자적으로 네이밍 서비스를 제공하는 것이 아니라 ATM 망과 인터넷 망과의 연동이 가능하다. 이 경우 DNS를 기반으로 한 구조와 절차에 B-ISDN 서비스를 위한 추가적인 데이터 레코드와 절차가 필요하다. 또한 인터넷과의

연동을 위해서 연결형 서비스인 ATM과 비연결형 서비스를 제공하는 인터넷 사이의 통신을 위해서는 별도의 고려가 있어야 할 것이다.

현재까지 이러한 제약을 극복하기 위한 노력으로 IP over ATM에 대한 연구가 진행되었으며 그 연구방향은 크게 overlay model(또는 layered model)과 intergrated model(또는 peer model)로 나눌 수 있다. Overlay model에서는 ATM 망을 독립적인 링크 계층으로 사용하여 IP 패킷을 전송하는 것을 의미한다. 따라서 IP 계층(layer 3)과 ATM 계층(layer 2)에 각각 독립적인 라우팅 프로토콜이 존재하게 된다. 가장 전형적인 방법은 상위 계층(IP) 라우팅은 OSPF(Open Shortest Path First)나 RIP(Routing Information Protocol)와 같은 IP 라우팅 프로토콜을 사용하고 IP 라우터 연결은 PNNI와 같은 ATM 라우팅 프로토콜을 사용하는 것이다. 이 경우 ATM 종단점은 ATM 주소와 IP 주소를 가져야 하고 모든 IP 패킷은 ATM 망을 통하여 ATM 라우팅 기능을 이용하여 전송된다. 따라서 IP 패킷에 있는 IP 목적지 주소를 해당하는 ATM 주소로 변환해주는 ATM 주소해석 프로토콜이 존재해야 한다. 이러한 단점을 해결해 주는 것이 바로 ATM-DNS의 기능이라고 말할 수 있겠다.

또 한 가지 방법으로는 Integrated Model 방안이 있다. Integrated model은 ATM 계층과 IP 계층이 하나의 라우팅 프로토콜을 사용하는 것이다. 이 방법의 가장 큰 장점은 두 라우팅 프로토콜간의 상호작용의 복잡성과 비효율성을 피할 수 있다는 것이다. 이 모델에서는 ATM망을 통하여 IP 패킷을 전송하기 위하여 기존의 OSPF나 RIP와 같은 IP 계층 라우팅 프로토콜을 사용하게 된다. 그리고 연결설정을 위하여 LDP(Label Distribution Protocol)와 같은 표준화되지 않은 ATM 신호 프로토콜을 사용할 수도 있다. Integrated model에서는 주소해석 프로토콜을 필요하지 않게 된다. 그

러나 ATM 스위치가 멀티프로토콜 라우터와 같은 기능을 수행해야 하기 때문에 ATM 스위치의 복잡도가 증가할 수 있다.

IETF에서 표준화가 진행되고 있는 MPLS는 이러한 integrated model의 대표적인 예가 된다. MPLS는 기존에 제안되었던 도시바의 셀스위치 라우터(CSR), Epsilon의 IP 스위치, Cisco의 tag 스위치, IBM의 ARIS 등의 여러 레이블 스위치 개념을 결합하여 IP 전송에 적합한 integrated model 프로토콜을 정의해 나가고 있다.

오버레이 모델에서 목적지는 DNS와 IP over ATM 환경에서 ATM ARP에 의해 식별한다. 이러한 환경에서 목적지 이름으로부터 IP 주소를 식별하기 위해 DNS를 사용하며, ATM ARP는 IP 주소를 이용해 ATM 주소를 알아 내기 위해 사용된다. 이러한 절차는 DNS 패킷 교환과 ARP의 두 번의 과정을 거쳐야 비로소 단말에 접속할 수 있으므로 비효율적이다.

그러나 DNS가 ATM-DNS에 통합되어 ATM-DNS 서비스를 제공하는 네임 서버가 데이터베이스에 ATM 단말의 IP주소와 ATM주소를 모두 저장하고 있어서, 서로 이름과 주소 정보를 교환할 수 있게 된다면 목적지 ATM 주소를 식별하는 절차가 매우 간단하고 효율적이 될 것이다.

IX. 결 론

정보화 사회에서 ATM과 인터넷은 각각 초고속 정보전달 하부 망과 응용 기반을 제공하면서 초고속 중형정보통신 환경 구축의 쌍두마차로서의 역할을 해나가고 있다.

인터넷과 ATM은 상호 필요성에 의해 긴밀하게 결합될 필요성이 있으나 Connection-oriented 특성의 ATM과 Connectionless특성의 인터넷이 결합하기 위해서는 많은 기술적인 문제점이 남아 있

다. 특히 망 규모가 작은 사설 ATM망 영역에서의 인터넷 수용기술은 IETF 및 ATM Forum을 위주로 많은 해결방안을 제시하고 있지만 망 규모가 커짐에 따라 Scalability, QoS 보장, 보안, VPN(Virtual Private Network) 서비스의 제공 등 사설망에서 심각하게 고려되지 않던 새로운 문제들이 등장한다.

따라서 두 서비스 간의 연동을 위한 네이밍 서비스 제공이 무엇보다도 필요한 시점이며 이 분야에 대한 활발한 연구가 진행되고 있으나 아직까지 괄목할 만한 성과를 얻지 못하고 있다. 이 분야의 중요성을 고려할 때 국내에서도 많은 관심을 가져 재산권 확보와 연계하여 표준화 활동에 적극적으로 참여하는 것이 요구된다.

※참고문헌

- [1] ATM Forum, "ATM Name System Specification Version 1.0", af-saa-0069.000, Nov, 1996
- [2] ATM Forum, "ATM Name System V2.0 Baseline Text", BTD-SAA-ANS-02.00, Jan, 1999
- [3] Straw Ballot, "ATM Forum Addressing : User Guide", STR-RA-ADDR-USER-001.02, Oct, 1998
- [4] M. Lotttor, "Domain Administration Operation Guide", (RFC 1033), Nov, 1987
- [5] P. Mockapetris, "Domain Names ? concepts and facilities", (RFC 1034), Nov, 1987
- [6] P. Mockapetris, "Domain Names - implementation and specification", (RFC 1035), Nov, 1987
- [7] P. Vixie, "Dynamic Updates in the Domain Name System", (RFC 2136), April, 1997

[8] Paul Albitz, Cricket liu, "DNS and BIND",
O'Relly , 3rd ed, Sep, 1998



최준균

1988 한국과학기술원 (박사 - 데이터통신)
1985 한국과학기술원 (석사 - 통신)
1982 서울대학교 공과대학 (학사 - 전자공학)
1998년~현재 한국정보통신대학원대학교 부교수
1986년~1997년 한국전자통신연구원, 책임연구원
1990년~1991년 캐나다 토론토 대학, 교환연구원
1993년~1996년 ITU-T SG13, Associate
Rapporteur (AAL) 및 국내 대표
1997년~현재 ITU-T SG13, Rapporteur
(Q2: B-ISDN Network
Capability)
1997년~현재 한국정보통신기술협회 (TTA), 통
신망구조연구반 의장
관심분야 : 통신망, 네트워크 시스템 및 프로토콜, ATM