

# 이동 네트워킹 환경을 위한 프록시 기반 적용 기법

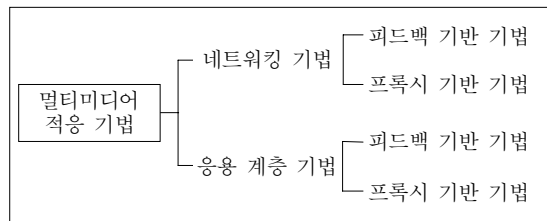
## Proxy Based Adaptation Techniques for Mobile Networking Environments

손지연(J.Y. Son)                      휴대클라이언트연구팀 연구원  
원유재(Y.J. Won)                    휴대클라이언트연구팀 선임연구원  
박준석(J.S. Park)                    휴대클라이언트연구팀 선임연구원  
김명규(M.G. Kim)                    휴대클라이언트연구팀 선임연구원  
한동원(D.W. Han)                    휴대클라이언트연구팀 선임연구원, 팀장  
황승구(S.K. Hwang)                  멀티미디어연구부 책임연구원, 부장

이동 네트워킹 환경에서 프록시의 적용은 메모리 요구량의 감소, 성능 향상, 비용과 활용도 측면에서의 효율성 등 많은 점에서 효용성이 입증되고 있다. 따라서 본 고에서는 프록시를 기반으로 하는 다양한 적용 기법들을 조사·분석하고, 아울러 프록시를 적용하는 데 있어서의 이슈가 되는 문제들을 살펴봄으로써 이동/무선 환경에서 프록시를 보다 효과적으로 활용할 수 있기를 기대한다.

### I. 서론

최근 급속한 진전을 보이고 있는 무선/이동 망 기술 및 각종 이동 단말 기술들은 궁극적으로 멀티미디어 데이터 서비스의 지원을 목표로 하고 있다. 이동 멀티미디어 서비스는 이동이 가능하다는 점이 유선 서비스와 가장 큰 차이점이며, 무선 환경에서의 전파 강도나 도달 시간 등을 이용해서 위치 정보도 얻을 수 있으므로 이를 이용한 새로운 서비스의 창출도 가능할 것으로 보고 있다. 그러나 유무선망간의 심한 이질성과 망 특성, 단말의 이동성으로 인해 기존의 멀티미디어 서비스를 이동/무선 환경에 그대로 적용하는 경우에는 예기치 못할 데이터 손실 및 지연시간 증가, 심지어는 서비스 중단 등의 현상을 초래할 수 있다. 이와 같이 네트워크 조건의 변화가 심한 이동 환경에서 기존의 멀티미디어 서비스들을 원활하게 지원하기 위해서 현재까지 여러 가지 적용



(그림 1) 이동 환경을 위한 적용 기법 분류

방안들이 제안되어 졌는데, 이들을 적용 대상에 따라서 분류해 보면, (그림 1)과 같이 크게 네트워킹 기법과 응용 계층 기법으로 나눌 수 있다[1].

이들은 다시 적용의 주체에 따라서 피드백 기반 적용과 프록시 기반 적용으로 분류되어 질 수 있는데, 피드백 기반 적용 기법은 제3자의 개입 없이 순수하게 송신자와 수신자 일련의 제어 정보 교환으로 환경에 적응하는 방식인 반면, 프록시 기반 적용 기법은 유무선 환경의 명백한 환경 차이를 극복하고자 프록

시로 지칭되는 제3자를 개입시켜 유무선 환경의 중재 역할을 통해 적응 서비스를 지원하는 방식이다 [2]. 본 고에서는 상기 (그림 1)의 멀티미디어 적응 기법들 가운데 네트워킹 측면에서의 프록시 기반 적응 기법들에 대해 초점을 맞추어 조사하고자 한다.

먼저, 제II장에서는 프록시의 일반적인 개요에 관해 언급하며, 제III장에서는 구체적인 프록시 적용 기법들을 기술한다. 이어서 제IV장에서는 프록시가 적용되고 있는 구현 환경들의 특성 및 사례들을 기술하며, 제V장에서는 프록시를 적용할 때에 공통적인 이슈들을 살펴보고, 마지막으로 제VI장에서는 결론을 맺기로 한다.

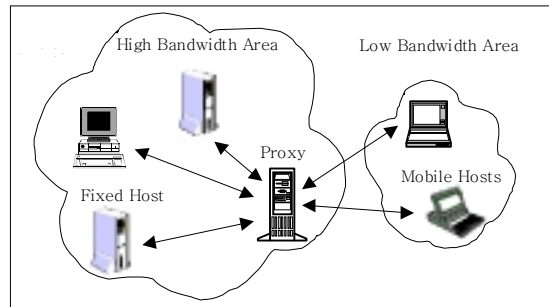
## II. 프록시 개요

프록시는 유선과 무선의 여러 환경(대역폭, 무선 클라이언트의 능력, 기타 조건의 잦은 변화 등)의 차이 때문에 발생하는 응용 서비스 저하에 대한 중재 역할을 하기 위한 것으로 기본적인 프록시 기반 구조는 (그림 2)와 같다. 게이트웨이, 필터, 에이전트라는 이름으로도 사용되지만, 본 고에서는 이들을 총칭하여 프록시라고 명명한다.

사실, 프록시는 이미 기존의 유선 망에서도 이질적인 요소들을 해결하기 위해서 방화벽(firewall)이나 각종 응용 계층의 게이트웨이의 용도로 활용되어져 왔다. 이러한 프록시 구조를 이동/무선 환경에서 특히 선호하는 이유는 다음과 같은 장점들을 가질 수 있기 때문이다[3].

- 파워 절약
- 메모리 요구량 감소
- 특화된 H/W로 접근
- 이동 플랫폼을 위한 유지 노력 감소
- 성능 향상
- 링크 단절에 대한 핸들링
- 비용과 활용도 면에서의 효율성

프록시를 적용하는 것의 가장 큰 특징 중의 하나는 사용자들에게 투명성(transparency)을 제공한다



(그림 2) 프록시 적용 환경

는 점이다. 이를 투명성의 정도에 따라 보다 세분화시켜 보면, 종단 시스템에 완전히 투명하게 운용됨으로써 기존 시스템이나 응용의 수정이 필요 없는 방식이 있을 수 있으며, 두 종단 시스템 모두를 수정하는 구현 방식(opaque implementation)이나 한 쪽 종단 시스템만 수정을 요하는 구현 방식(translucent implementation)이 존재할 수 있다[4]. 또한 프록시의 위치에 따라서 통합 방식과 분산 방식으로 분류할 수 있는데, 통합 방식은 유무선 접속 지점과 같이 성능 증가가 기대되는 곳이 특정 한 지점에 프록시를 두는 경우를 말하며, 분산 방식은 위성 통신에서와 같이 성능 증가를 기대하는 특정 링크 주변에 주로 프록시를 사용하는 방식을 말한다. 분산 방식은 다시 끝단에 있는 프록시들의 기능이 동일한가의 여부에 따라 대칭적 방식(symmetric)과 비대칭적 방식(asymmetric)으로 분류될 수 있다.

## III. 프록시 적용 기법

제III장에서는 이동 환경의 네트워킹 측면에서 보다 구체적인 프록시의 적용 기법들을 기술하기로 한다.

### 1. 동적 재경로(re-routing) 설정 기법

이동성을 지원하기 위해서 프록시를 사용하는 기법으로서 대표적인 프로토콜이 IETF(The Internet Engineering Task Force) 2002 Mobile IP(Internet Protocol)이다[5]. Mobile IP는 이동 호스트의 영구

IP 주소를 바꾸지 않고 이동성 지원을 가능하도록 하기 위해서 홈 에이전트(home agent)와 외부 에이전트(foreign agent)라고 하는 프록시를 활용한다. 기존 IP의 경우에는 사용자가 원래의 네트워크 접속 점을 떠나 다른 네트워크에 접속하기 위해서는 그 네트워크의 IP 주소를 새로 할당받아 수동으로 IP 주소를 변경시켜 주어야 하지만, Mobile IP는 사용자의 호스트 IP 주소를 바꾸지 않고 다른 곳의 네트워크에 접속하여 인터넷 서비스를 받을 수 있는 기능을 제공한다.

이때, 홈 에이전트는 이동 호스트의 홈 네트워크에 존재하면서 홈 IP 주소와 COA(Care of Address)를 관리하는 프록시로서, 다른 네트워크에 접속했을 경우 이동 호스트로 향하는 데이터 패킷을 외부 네트워크로 전달해 주기 위한 터널링 기능을 담당한다. 반면, 외부 에이전트는 이동 호스트가 외부 네트워크에 속할 때 COA를 부여하는 프록시로서, 이동 호스트가 자신이 서비스하는 네트워크에 들어왔을 경우 이들에 대한 라우팅 서비스를 제공한다. 홈 에이전트로부터 이동 호스트로 터널링을 통해 캡슐화되어 전달된 데이터 패킷은 외부 에이전트에서 역캡슐화되어 이동 호스트에게 전달된다. 이동 호스트로부터 나가는 데이터 패킷의 경우에는 단지 기본 게이트웨이 기능만을 수행한다.

## 2. 연결 분리 기법

프록시를 이용한 연결 분리 기법은 한 쪽 종단 시스템으로부터의 연결을 끊고 반대쪽 시스템과 연결을 설정함으로써 이루어진다. Indirect TCP(I-TCP)가 대표적인 예이다[6]. 이는 두 종단 시스템 사이에 TCP(Transmission Control Protocol) 능력의 차를 해소하기 위해 사용되는데, 주목할 점은 프록시를 통한 연결 분리 기법을 사용한다 하더라도 종단간의 연결(end-to-end connectivity)을 위한 IP의 동시 사용을 배제할 필요는 없다는 점이다. 즉, 연결 분리 기법은 응용 프로그램이나 연결별로 관리되어야 하며, 최종 사용자에 의해 제어 가능해야 하고, 또한 사용자로 하여금 특정 연결이나 응용 프

로그램에 대해서 연결 분리 기법을 적용할지 혹은 말지를 결정할 수 있어야 한다.

## 3. ACK 핸들링 기법

TCP의 경우 프록시를 이용한 ACK(Acknowledgement) 처리 기법들이 다양하게 존재하며, 이들은 독립적으로 또는 병행하여 적용되어 질 수 있다.

- ACK Spacing 기법

ACK spacing 기법은 TCP의 버스트(burst) 데이터들에 대해 ACK를 이용하여 데이터의 흐름을 조절하는 방식으로, 버스트의 크기를 줄일 수 있다는 점에서 유용하다[7].

- Local ACK 기법

Local ACK 방식은 프록시가 대신 ACK를 해주는 방식으로 TCP의 경우 이를 통해서 Slow Start의 속도를 향상시킬 수 있으며, 송신측의 전송 윈도우가 빨리 열리도록 한다. 또한 “proxy ARP”와 같이 ARP(Address Resolution Protocol) 요청에 대해서 프록시가 대신 ARP 응답을 해주는 경우도 있다[8].

- Local 재전송 기법

데이터 손실이 감지되었을 때 프록시에서 송신자 대신 데이터 재전송을 수행하는 기법으로 local ACK 기법과 같이 사용될 수도, 아닐 수도 있다. Local ACK를 사용하지 않으면서 local 재전송 기법을 사용하는 대표적인 예가 Snoop이다[9]. Snoop 프로토콜은 프록시를 사용하여 TCP 데이터를 캐쉬하면서 수신자로부터의 ACK를 감시하고 있다가 중복 ACK를 감지하면, 데이터 손실이 발생하였다고 간주하고 local 재전송을 수행한다.

- ACK 필터링 기법

비대칭적인 환경에서 데이터 역방향 링크의 대역폭 관리를 위해 프록시를 사용하는 방법으로서, 역방향 채널로 보내는 ACK 빈도 수를 줄이는 방법의 하나이다. 이는 라우터나 기지국(Base Station: BS)의 링크 계층에서 동작하며, 동일 연결에 대해 수신측으로부터 새로운 ACK가 도착하면, 링크 계층 큐

에서 이보다 오래된 ACK를 찾아 제거함으로써 송신자로 가는 전체 ACK 수를 줄이는 기법이다[10, 11].

- ACK 재구성 기법

이 기법은 비대칭적 네트워크 환경에서 역방향의 ACK 빈도 수를 줄인 것이 TCP 성능에 부정적 영향을 끼치지 않도록 하기 위해서 역방향의 병목 링크를 지난 후에 ACK 스트림을 재구성하는 데 프록시를 활용하는 기법이다[10, 11]. 이는 TCP 성능에 영향을 미치는 폭주 윈도우가 ACK 수에 의해서 증가되는데 반해, 역방향 링크에서의 병목현상을 줄이기 위해 ACK 필터링 기법과 같이 ACK 수를 줄이는 기법들이 제안되면서 이의 부작용(폭주 윈도우가 크게 증가되지 않음)을 없애기 위해 만들어진 기법이다.

“ACK Reconstructor”라고 불리는 soft-state 에이전트(프록시)가 하는 기능은 ACK 순서에서의 차이를 깨우며, 종단간 의미를 깨지 않으면서 송신 측에 보여지는 ACK 스트림을 조절하는 역할을 한다. ACK reconstructor에서는 *ack\_thresh*와 *ack\_interval* 파라미터를 가지고 ACK 스트림을 조절하는데, 전자는 출력 단에서 산재되어 있는 ACK들의 간격(일반적으로 2)을 결정하는 파라미터이며, 후자는 재구성된 ACK들 사이에 일시적인 간격을 결정하는 파라미터이다.

- ACK-우선 스케줄링 기법

비대칭적인 환경에서 특히 양방향 데이터 전송이 일어나는 경우에는 데이터 역방향에서도 ACK 패킷과 데이터가 동일 자원을 공유해야 한다. 이 경우, 데이터와 ACK에 대해 하나의 FIFO(First In First Out) 큐를 두는 것은 문제를 일으킬 수 있다. 즉, 상대적으로 적은 양의 ACK 패킷 처리가 길이가 긴 데이터들로 인해 상당시간 지체(block)될 수 있다는 것이다. 따라서 FIFO로부터 데이터와 ACK 패킷을 서로 다르게 스케줄링하는 기법이 제안되어졌는데, 이는 프록시에서 데이터 패킷보다 ACK 패킷의 우선 순위를 훨씬 높여 처리함으로써 전체적인 데이터 성능을 높이는 기법이다[10, 11].

## 4. 압축 기법

성능을 개선하기 위해서 프록시를 이용한 다양한 형태의 압축 기법들이 제안되어 지고 있다. 압축 기법은 링크를 통해서 전달되어야 할 데이터량을 줄임으로써, 대역폭이 제한된 링크 상에서 특히 유용한 기법이다. 따라서 링크의 효율성을 증대시키며, 처리 지연시간과 에러율이 높은 링크 상에서의 패킷 손실률을 줄일 수 있고, 즉각적인 반응시간 개선 및 시스템의 오버헤드를 줄일 수 있는 장점이 있다.

널리 사용되어 지고 있는 압축 기법들로는 링크 계층 압축, 또는 TCP/IP 헤더 압축 등이 있다. RFC (Request for Comments)1144는 TCP 헤더를 압축하기 위해서 널리 사용되는 방식들을 기술하고 있으며, 이 밖의 헤더 압축 기법들이 RFC2507, RFC2508, RFC2509 등에 기술되어져 있다[12-15]. 반면, Payload 압축은 인터넷 보안문제가 부각되면서 그 비중이 커지고 있다. IP 계층의 보안 메커니즘은 IP Payload를 랜덤 비트 스트림(random bit stream)으로 바꾸어 주기 때문에, Payload 압축은 IP 보안 메커니즘이 적용되기 전에 적용될 필요가 있다. RFC 2393의 경우, 임의의 IP 세그먼트 Payload들에 적용되어질 수 있는 일반적 압축 알고리즘에 대한 프레임워크를 정의하고 있지만, IP Payload들의 대다수 타입(image, audio, video이고 zipped file)들이 이미 압축되어 있기 때문에 항상 적용 가능하지는 않다[16].

## 5. 터널링 기법

프록시를 이용한 터널링 기법은 TCP 메시지들을 캡슐화(encapsulation)/역캡슐화(decapsulation)하거나, 연결 분리 기법을 사용하는 분산형태의 프록시들을 서로 연결하는 데 사용되어 질 수 있다.

## 6. 링크 단절에 대한 핸들링 기법

무선 링크 상에서 링크 단절은 흔히 발생될 수 있는 일이다. TCP에서는 링크 단절이 되면, 그 기간

동안 송신자가 ACK를 전혀 받지 못하므로 이를 복구 발생으로 간주하여 잘못 제어하게 된다. 따라서 이를 해결하기 위한 방법으로 M-TCP에서는 프록시에서 타이머에 의해 링크 단절 사실을 감지하여 송신측에 알리는 방식을 제안하였다[17]. 반면, TCP 연결 분리 기법을 적용하는 경우에는 단절 사실을 프록시 반대쪽에는 알지 못하도록 하고, 링크가 재연결되고 나면, 로컬 복구 절차를 수행하는 방법도 가능하다.

## 7. 멀티캐스팅

단말의 하드웨어적인 제약 등의 이유로 멀티캐스팅 기능을 지원하지 않는 셀 또는 네트워크가 존재하는 경우, 이를 지원하는 네트워크와의 접속 지점에 프록시를 두어 멀티캐스트 데이터를 유니캐스트 데이터로 변환하는 기능을 부여할 수 있다.

## 8. QoS 적용 기법

이동 환경에서의 적응형 응용 프로그램을 지원하기 위해 프록시를 이용하여 QoS(Quality of Service) 측면에서 보다 완전한 해결책을 제공하고자 했던 시도가 Mobware이다[18]. Mobware는 플로우들의 적응과 핸드오프를 지원하기 위해 두 개의 프록시 오브젝트를 사용하고 있다. 하나는 QoS 적용 프록시(QoS Adaptation Proxy: QAP)이고, 다른 하나는 RAP(Routing Anchor Proxy)이다. QAP는 이동 단말이 무선 링크 상에서 변동이 심한 자원의 이용가능성에 대해 검사(probe)하고 적응(adapt)하는 일들을 지원하는 반면, RAP는 Mobware의 주 기능 중의 하나인 플로우 번들링(flow bundling) 기능을 지원함으로써 이동 단말로부터 빠르고 효율적이며 확장 가능한 핸드오프(hand-off)를 지원하도록 한다.

## IV. 프록시 적용 환경

제IV장에서는 프록시가 적용되어 지고 있는 환경의 사례들을 기술한다.

## 1. VSAT 환경

현재 VSAT(Very Small Aperture Terminal) 네트워크는 인공 위성으로 구현되어 있으며, 전형적으로 중심에 지구 기지국이 존재하는 스타 형태로 구현되어 있다. VSAT으로부터 보내진 데이터는 바깥 경로(out-route)를 통해 원격지로 전송되며, 원격지로부터의 데이터는 안쪽 경로(in-route)를 통해 전달된다. 대개의 경우 안쪽 경로보다 바깥 경로가 훨씬 크다.

일반적으로 VSAT 네트워크는 사실망 구축에 사용되며, 대역폭(bandwidth) × 지연(delay) 시간이 크고, 피드백이 오래 걸리며, 비대칭성과 낮은 에러율을 주요 특징으로 한다[19]. 이러한 VSAT 환경에 프록시를 활용하는 주 목적은 처리량 향상과 크기가 작은 트랜잭션(transaction)에 대한 즉각적인 응답 시간을 향상시키는 데 있다. 따라서 VSAT 환경을 위해 구현되는 프록시들은 공통으로 TCP 성능 향상에 초점을 맞추고 있으며, 비대칭적인 분산 형태로 구현하고 있고, local ACK과 local 재전송을 갖는 연결 분리 접근 방법을 주로 채택하고 있다. 또한 대역폭 요구량을 최소화하기 위한 압축 기법들이 공통적으로 많이 사용되고 있다. 반면, 지원하고자 하는 최대 처리량이나 위성 링크를 통해서 사용되는 프로토콜(수정된 TCP 혹은 UDP(User Datagram Protocol) 등등) 또는 사용하는 압축 형태 등은 각 프록시마다 개별적으로 결정되어진다.

## 2. 원거리 무선 망 환경

원거리 무선 망의 네트워크 특성을 열거해보면, 저 대역폭, 긴 지연시간, 높은 에러율, 링크상에서 데이터 수용을 위한 버퍼 요구량 증가, 예기치 못하는 연결 단절 현상 및 링크 연결의 (재)설정 시간의 오랜 지연현상, 단말의 이동성 등이 있다. 이와 같은 무선 원거리 망 환경에서 프록시를 활용한 사례로는 대표적으로 Mowgli 모델이 있다[20]. 이 모델에서 채택한 프록시 적용 기법들로는 TCP 연결 분리 기법과 함께 응용 계층 프록시로서 다양한 형태의 압

축 기법, 대응되는 프로토콜의 수정, 왕복 지연시간 감소, 무선 원거리 링크를 통한 우선 순위 기반의 데이터 멀티플렉싱 기법, 링크 계층의 흐름 제어 기법 등이 있다.

### 3. 근거리 무선 망 환경

근거리 무선 망은 전형적으로 하나의 BS가 하나의 셀을 관장하는 셀 방식으로 구성되어 있다. 이때 BS는 직접 유선 망에 연결되어 있으며, 각각 담당할 셀 내에 위치하고 있는 호스트로부터 또는 호스트에게 패킷들을 전달할 책임이 있다. 무선 근거리 수신기를 장착한 이동 호스트는 한 셀에서 다른 셀로 이동할 때는 유선 망과 이동 호스트 사이에 전달되던 패킷들에 대한 책임이 새로 이동한 셀의 BS로 이전되어야 한다. 이를 핸드오프라고 부른다.

근거리 무선망은 현재 링크의 대역폭이 1Mbps에서 2Mbps 정도이며, 최고 20Mbps까지를 목표로 하며, 왕복 처리지연시간은 수 ms에서 수십 ms이다. 또한 무선 링크의 물리적 특성상 에러 확률이 높고 특히, 핸드오프로 인해 연속된 패킷 손실의 우려도 존재한다. 이에 대해 종단간의 최종 신뢰성 책임을 맡고 있는 TCP에서는 이들 무선에 의한 패킷 손실을 망의 폭주현상으로 오인함으로써 급격한 성능 저하를 야기시킬 수 있다.

이를 극복하기 위해 프록시를 활용한 구현 사례로서 대표적인 프로토콜이 Snoop이다[9]. 버클리의 Snoop 프로토콜은 Snoop 에이전트라는 프록시를 유무선 접속 지점에 위치시켜, 종단간의 의미를 유지하면서 데이터 캐칭과 중복 ACK를 이용한 로컬 재전송 기법을 적용함으로써 무선 구간의 에러 문제를 해결하려고 하였다. 또한 Snoop 에이전트는 이동 호스트가 송신자인 경우의 무선 구간 에러에 대한 NACK(Negative Acknowledge) 처리 기능과 ELN(Explicit Loss Notification) 기능을 갖는다. 이는 이동 호스트가 송신자인 경우 패킷 손실이 발생했을 때 로컬 재전송을 하지 않는 대신, 유선 망에서의 폭주로 인한 패킷 손실이 아님을 알리기 위해서이다.

## V. 프록시 적용의 이슈들

이제까지는 이동 네트워킹 환경에 프록시를 활용한 각종 기법들을 구체적으로 살펴보았다. 이어서 제V장에서는 이렇게 다양하게 적용되고 있는 프록시 기반 기법들에 있어 공통적으로 대두되고 있는 이슈들을 기술하고자 한다.

### 1. 종단간 의미

RFC1958에서 언급하고 있는 바와 같이, 종단간의 의미(end-to-end semantic)는 인터넷 구조의 기본 원칙 중의 하나이다. 종단간 의미가 깨지기 쉽다는 것이 프록시의 사용을 일반적으로 권장하지 않는 주 이유이지만, 모든 프록시 적용이 종단간의 의미를 깨는 것을 의미하지는 않는다. 주로 종단간 의미가 깨어지기 쉬운 경우는 연결 분리 기법에서이며, 커널 영역에서 주로 동작하는 네트워크 계층 이하 프로토콜에 사용자 영역에서 동작하는 필터를 적용하는 경우에도 그렇다. 종단간 의미가 깨어짐으로써 부정적인 영향을 끼치는 점들을 나열해 보면, 다음과 같다.

- 보안

종단간 의미가 깨어짐으로써 가장 나쁜 영향을 끼치는 것이 IP 계층 보안(IP layer security)의 종단간 사용이 불가능하게 된다는 점이다. 즉, IPsec을 채택하는 경우, IPsec의 ESP(Encapsulating Security Payload) 헤더를 통한 IP 패킷의 암호화는 프록시에서 TCP 헤더와 Payload와 관련한 작업들을 어렵게 하기 때문이다.

비투명(non-transparent)한 프록시 구현에서 종단 시스템에서 프록시를 신뢰할 수 있다면, 각 종단 시스템과 프록시 사이에 각각 IPsec을 사용할 수 있다. 그러나 대부분의 경우에 종단 시스템들이 프록시를 신뢰할 수 없다는 점 때문에 이를 적용하는 것이 바람직하지 않다. 더구나 이 방법은 종단간의 보안 방식처럼 안전하지 않으며, 프록시를 사이에 두고 종단 시스템들이 서로 다른 보안 레벨로 협상되어 질

수도 있으며, 이에 따른 부작용이 발생할 수 있다.

반면, 투명한(transparent) 프록시 구현에서는 종단 시스템 측에서 프록시의 존재(정보)를 모르기 때문에 신뢰하기 어렵다. 하지만, 분산 방식에서는 두 개의 프록시 간에는 IPsec이 구현되어 질 수 있으며, 프록시 구현이 사용자들에게 투명하지 않다면, 사용자 측에서 IPsec 터널의 종단점으로서 프록시를 사용하도록 지정할 수도 있다. 그러나, 결론적으로 보면, 프록시의 구현이 종단간 의미를 깨지 않는다 할 지라도 IPsec과 함께 사용되기는 어렵다고 할 수 있다.

- Fate Sharing

종단간 의미 문제로 인해 영향을 받는 것이 다수의 단말들이 프록시를 공통으로 사용함으로써 발생하는 문제들이다. 예를 들어 프록시 고장 가능성을 포함하여 네트워크에 이상이 발생할 경우, 프록시에 대한 의존도와 라우팅 경로 변경 여부에 따라 회복 가능성이 크게 좌우된다는 점이다. 따라서 사용자에 대한 위험 감수를 해야 하며, 피해를 최소화하기 위해 종단간 IP 서비스를 선택할 수 있는 옵션을 가지고 있어야 한다. 이 밖에도 특정 클라이언트의 소프트웨어 업그레이드 시에 프록시 측의 업그레이드 작업을 필요로 하는 경우도 있을 수 있다.

- 종단간 ACK

종단간에 응용 프로그램은 데이터의 최종 전달에 대한 확인의 의미로 TCP ACK를 사용한다는 점에서 보면 프록시가 중간에 개입되는 것이 그리 좋은 방법이 아닐 수 있다. 그러나 종단간의 TCP ACK이라도 상대쪽 응용 계층에 데이터를 제대로 전달했다는 것을 보장하지는 않으며, 이를 위해서는 응용 계층 별도의 ACK 방식이 필요하다는 점에서 보면, 프록시 사용이 데이터 전달을 최종 확인하는 데 방해가 되지 않는 것으로 보인다.

## 2. 성능 상의 병목현상

프록시의 적용은 그 구조 상으로 대규모 네트워크 서비스를 제공하는 데 어려운 점이 있다. 이는 프

록시의 사용이 네트워크의 병목을 일으키는 요인이 될 수 있기 때문이다. 병목현상을 일으킬 수 있는 요소들로는 프록시에서 데이터 copy가 빈번하게 발생하는 경우, 메모리 대역폭 상에 병목을 일으킬 수 있으며, 프록시에서 패킷 스케줄링이나 프로토콜 변환 기능을 수행하는 경우에도 처리 오버헤드로 인한 병목현상이 발생할 수 있다.

## 3. 프록시 위치

프록시의 위치는 보안이나 견고성 측면에서 보면, 안전이 보장된 사이트에 위치시키는 것이 바람직하다. 반면, 물리적으로는 유무선이 접속되어 있는 마지막 지점에 위치시키는 것이 링크 특성 및 프록시와 이동 호스트 사이에 사용될 수 있는 특별한 링크 계층 프로토콜에 대한 정보를 수집하기 쉽다는 점에서 유리하다.

프록시의 위치 뿐만 아니라 프록시의 수도 하나를 둘 것인지, 두 개 또는 그 이상의 N개를 둘 것인지를 결정하는 것도 중요하다. 만일 두 개의 프록시를 두는 경우에는 이들 둘 간의 프로토콜 최적화가 가능하다. 즉, 데이터 경로가 클라이언트-프록시 1-프록시 2-서버일 때, 클라이언트와 서버 측에서는 프록시들간에 발생되고 있는 일들에 대해 관여할 필요는 없다. 반면, N개의 프록시들을 사용하는 경우에는 프로그램 가능한 라우터 모델과 일치되는데, 이는 몇 개의 프록시라도 라우팅 경로를 따라서 어디든 위치할 수 있다[21]. 즉, 취약한 링크가 있는 곳마다 프록시를 둬으로써 라우팅 경로에 대한 re-engineer가 가능할 수 있다는 의미이기도 하다.

## 4. 프록시 이동

클라이언트(단말)가 이동함에 따라 프록시와의 물리적인 경로가 멀어지게 되면, 라우팅 경로에 있어서 비효율적인 요인이 발생할 수 있다. 이를 위해서는 프로세스 이전(transition), 체크포인트, 이전 가능 상태(transferable state)와 관련한 효율적인 프록시 변경 메커니즘에 대한 연구가 필요하다.

## 5. 확장성(Extensibility) 및 유연성

프록시를 기반으로 하는 대부분의 접근 방식들은 특정 데이터 타입 또는 특정 응용을 위해 개발되어 졌기 때문에, 새로운 응용과 데이터 타입들에 대해 확장하여 적용하기 어렵다는 점에서 확장성에 문제가 있다. 반면, 대부분의 시스템에서는 프록시의 기능이 정적으로 고정되어 있기 때문에 네트워크의 변동이 심한 경우에 잘 대처하기가 어렵다는 점에서 유연성이 부족하다[22].

## VI. 결론

이상으로 이동 네트워킹 환경에서 프록시를 기반으로 하는 적응 기법들의 종류와 구체적 내용에 대해 살펴보았다. 이 밖에도 프록시를 활용한 많은 적응 기법들이 존재할 수 있으며, 그 응용 분야 또한 매우 다양하다. 아울러 본 고에서는 이동/무선 환경에서 프록시를 적용하는 데 공통적으로 이슈가 되고 있는 문제들을 지적하였다. 각 이슈들에 대해서는 부분적으로 해결 방안들이 모색되어 지고 있으나, 그 성과는 아직 미진한 실정이다. 이동 환경에서 프록시의 적용에 대해 이미 그 효용성이 입증되고 있는 만큼, 공통적인 이슈들에 대해 보다 많은 연구가 이루어져야 할 것으로 보인다.

## 참고문헌

- [1] Bobby Vandalore, Wu-chi Feng, Raj Jain, Sonia Fahmy, "A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia," submitted to *the Journal of Real Time Systems* (Special Issue on Adaptive Multimedia), Apr. 1999. <http://www.cis.ohio-state.edu/jain/papers.html>.
- [2] Aruna Seneviratne, Behcet Sarikaya, "Cellular Networks and Mobile Internet," *Computer Communications*, Vol. 21, 1998, pp. 1244 - 1255.
- [3] Charles E. Perkins, "Mobile Networking in the Internet," *Mobile Network Application (MONET)*, Vol. 3, No. 4, 1998, pp. 319 - 334.
- [4] J. Border, M. Kojo, Jim Griner, and G. Montenegro, "Performance Enhancing Proxies," Internet-Draft, June 25, 1999.
- [5] C. Perkins, Editor, "IP Mobility Support," IETF RFC 2002, Oct. 1996.
- [6] A. Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," *Proc. of the 15<sup>th</sup> Int'l Conf. on Distributed Computing Systems*, Vancouver, Canada, June 1995, pp.136 - 143.
- [7] C. Partridge, "ACK Spacing for High Delay-Bandwidth Paths with Insufficient Buffering," Internet-Draft draft-rfced-info-partridge-01.txt (work in progress), Sept. 1998.
- [8] James D. Solomon, *Mobile IP - The Internet Unplugged*, Prentice-Hall, 1998.
- [9] Hari Balakrishnan, *Challenges to Reliable Data Transport over Heterogeneous Wireless Networks*, PhD Thesis, University of Berkeley, 1998.
- [10] Eric A. Brewer, Randy H. Katz *et al.*, "A Network Architecture for Heterogeneous Mobile Computing," Invited Paper, *IEEE Personal Communications Magazine*, Oct. 1998.
- [11] Hari Balakrishnan, Venkata N. Padmanabhan, "TCP Performance Implications of Network Asymmetry," IETF Internet Draft, Sept. 1999.
- [12] V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links," RFC 1144, Feb. 1990.
- [13] M. Degermark, B. Nordgren, and S. Pink, "IP Header Compression," RFC 2507, Feb. 1999.
- [14] S. Casner, V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links," RFC 2508, Feb. 1999.
- [15] M. Engan, S. Casner, and C. Bormann, "IP Header Compression over PPP," RFC 2509, Feb. 1999.
- [16] A. Shacham, R. Monsour, R. Pereira, and M. Thomas, "IP Payload Compression Protocol (IPcomp)," RFC 2393, Dec. 1998.
- [17] K. Brown and S. Singh, "M-TCP: TCP for Mobile Cellular Networks," *ACM Computer Communications Review*, Vol. 27, No. 5, 1997.
- [18] Oguz Angin, Andrew T. Compbell, Michel E. Kounavis, and Raymond R.F. Liao, "The Mobiware Toolkit: Programmable Support for Adaptive Mobile Networking," *IEEE Personal Communications*, Aug. 1998, pp. 34 - 43.
- [19] M. Allman, D. Glover, and L. Sanchez, "Enhancing



- TCP Over Satellite Channels Using Standard Mechanisms,” BCP28, RFC 2488, Jan. 1999.
- [20] M. Kojo, K. Raatikainen, T. Alanko, “Connecting Mobile Workstations to the Internet over a Digital Cellular Telephone Network,” *Proc. Workshop on Mobile and Wireless Information Systems (MOBIDATA)*, Rutgers University, NJ, Nov. 1994. <http://www.cs.helsinki.fi/research/mowgli/>, revised version published in *Mobile Computing*, Kluwer, 1996, pp. 253 – 270.
- [21] D.L. Tennenhouse and D.J. Wetherall, “Towards an Active Network Architecture,” *ACM Computer Communications Review*, Vol. 26, No. 2, Apr. 1996, pp. 5 – 18.
- [22] J. Seitz, K. Cheverst, N. Davies, M. Ebner, A. Friday, “Management of Proxy Objects Providing Multimedia Applications in the Mobile Environment,” *Proc. of the 6<sup>th</sup> IFIP/IEEE Int’l Symp.*, 1999, pp. 915 – 928.