

# IPv6-IPv4 변환시스템의 모델링 및 성능분석

서쌍희\*, 공인엽\*\*

\*부산대학교 산업자동화 및 정보통신분야 인력양성사업단

\*\*부산대학교 컴퓨터공학과

e-mail : shseotwin@pusan.ac.kr

e-mail : leafgirl@pusan.ac.kr

## Modeling and Performance Analysis of IPv6-IPv4 Translation System

Ssang-Hee Seo\*, In-Yeup Kong\*\*

\*Dept. of Brain Korea 21, Pusan National University

\*\*Dept. of Computer Engineering, Pusan National University

### 요 약

IPv6-IPv4 변환시스템은 기존의 IPv4 네트워크와 신규 구축되는 IPv6 네트워크 간의 통신을 가능하게 하는 게이트웨이 기반 기술이다. 이러한 IPv6-IPv4 변환시스템에서는 네트워크 간의 모든 트래픽을 변환해야 하므로 높은 성능을 요구된다. 이에 본 연구에서는 이전 연구에서 구현된 게이트웨이 기반 IPv6-IPv4 변환시스템과 변환기의 성능분석에 적용될 수 있는 큐잉 모델을 제시하고 부과되는 트래픽에 따른 처리 성능을 산출하는 분석적인 방법을 제시하였다. IPv6-IPv4 변환시스템의 분석 모델의 경우, 도착간격은 지수분포를 따르고, 서비스시간은 M/M/1/K 모델 기반의 일반분포를 따른다. 또한 IPv6-IPv4 변환시스템의 변환기는 트래픽에 대한 변환 처리를 담당하는 핵심 모듈로서, 순차적인 단계로 이루어진다. 즉, 변환시스템의 변환기 자체의 분석 모델의 도착간격은 지수분포를 따르고, 서비스시간은 M/G/1/K 모델 기반의 일반분포를 따른다. 이렇게 제안된 모델에 대해 상세하게 설명하였으며, 이를 검증하기 위해서 모델을 적용하여 근사한 결과와 실제 측정 결과를 비교하였다.

### 1. 서론

인터넷의 빠른 성장에 따라 32 비트의 IP 주소는 2008 년경에는 모두 고갈될 것이라고 전망되고 있다 [1]. 이에 IETF 는 IPv6 라 불리는 차세대 프로토콜을 개발하기 시작했다. IPv6 는 128 비트 주소체계를 사용하기 때문에 풍부한 주소공간을 활용하여 많은 수의 네트워크 장치들에 각각의 유일한 IP 주소를 제공할 수 있다[2]. 또한 플러그앤플레이 방식의 편리한 인터넷 제공과 이동성 지원, 서비스 품질보장, IPSec 을 통한 보안기능의 강화[3] 등 다양한 기능을 제공할 수 있다.

그럼에도 불구하고, IPv6 도입에 따른 여러가지 제약사항들 때문에 IPv4 에서 IPv6 로의 전환은 꽤 오랜 기간이 소요될 것이다. 이러한 단계에서 필요한 IPv6 라우팅, IPv6 주소, DNS 등의 여러 이슈들이 논의되고 있으며, 무엇보다 필요한 것이 기존에 동작되고 있는 IPv4 호스트 및 라우터와의 호환성과 기존 IPv4 에서 IPv6 로의 자연스러운 이전을 지원해 주는 IPv6-IPv4

변환 메커니즘이다. 이 문제를 해결하기 위해 제안된 여러 기술 중에서 게이트웨이 레벨의 변환시스템은 FreeBSD 와 Linux 상에서 구현되어 있다[4,5,6,7].

이러한 변환시스템은 네트워크 간 트래픽 변환 처리를 위해 고성능을 요구한다[8]. 이에 본 논문에서는 이전의 연구[8]에서 구현된 Linux 기반 소프트웨어 변환시스템 및 변환기 자체의 성능을 분석하기 위한 수학적인 모델을 제시하였다.

본 논문의 구성은 다음과 같다. 제 2 절에서 IPv6-IPv4 변환시스템의 구성과 동작특성을 기술하고, 제 3 절에서 IPv6-IPv4 변환시스템과 변환기 자체에 대한 큐잉 모델을 제시하였다. 제 4 절에서 제시된 큐잉 모델에 대한 근사해를 구하였으며, 제 5 절에서 시뮬레이션 결과에 대한 성능을 분석하였다. 마지막으로 제 6 절에서 결론과 향후 연구 방향을 제시하였다.

### 2. IPv6-IPv4 변환시스템

#### 2.1 IPv6-IPv4 변환시스템 구조

IPv6-IPv4 변환시스템의 전체적인 구조는 그림. 1 과

같다.

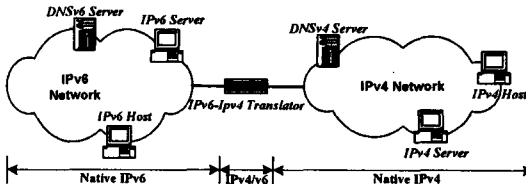


그림 1. IPv6-IPv4 변환시스템 구조

전체 시스템은 IPv6 네트워크, IPv4 네트워크 그리고 가운데에 IPv6-IPv4 변환시스템으로 구성되어 있다. IPv6 네트워크는 IPv6 호스트는 단지 IPv6만을 지원하는 호스트이며, DNSv6 서버는 IPv6 네트워크의 DNS 서비스를 담당한다. IPv4 네트워크 역시 IPv4만을 지원하는 호스트와 DNS 서비스를 담당하는 DNSv4로 구성되어 있다. IPv6-IPv4 변환시스템은 변환기술로서 NAT-PT(Network Address Port Translator-Protocol Translation)[9]메커니즘을 사용한다.

2.2 IPv6-IPv4 변환시스템의 변환기 구조

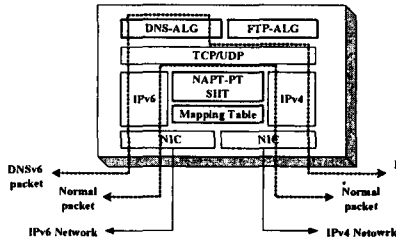


그림 2. 변환기의 프로토콜 구조

IPv6-IPv4 변환기는 IPv4 스택과 IPv6 스택을 모두 포함하고 있으며, NAT-PT/SIIT, 맵핑 테이블, DNS-ALG, FTP-ALG로 구성되어 있다.

NAPT-PT/SIIT는 IPv4 헤더와 IPv6 헤더간의 변환 뿐만 아니라, IPv6 IP 주소/포트와 IPv4 IP 주소/포트 간의 변환을 수행하는 모듈이다. 맵핑 테이블은 IP 주소/포트에 대한 정보를 가지고 있으며, DNS-ALG는 DNSv6 헤더 필드와 DNSv4 헤더필드간의 변환을 담당한다. FTP-ALG는 IPv4와 IPv6가 다른 FTP 명령을 변환하며, DNG-ALG와 FTP-ALG는 IP 페이로드 내의 IP 주소를 변환한다.

2.3 IPv6-IPv4 변환시스템의 기능 구조

그림 3에서는 IPv6-IPv4 변환시스템의 기능적인 구조와 내부적인 패킷의 흐름을 나타내었다[10,11].

- 1) 네트워크로부터 도착한 패킷은 네트워크인터페이스 카드의 하드웨어 메모리에 저장된다. 카드는 하드웨어 인터럽트를 발생시키며, 프로세서는 디바이스 드라이버를 실행하고 카드로부터 주메모리로 패킷을 복사한다.
- 2) 패킷은 입력 큐내에 선입선출 방식으로 저장된다. (프로세서 당 한 개의 입력 큐가 존재)
- 3) 인터럽트가 반환되기 전, 디바이스 드라이버는 소프트웨어 인터럽트를 발생시키며 필터가 호출된다.

소프트웨어 인터럽트가 일단 수행되면, 입력 큐의 패킷들은 OSI 2계층에서 해석되어 타입 필드의 값에 따라 적절한 3계층으로 넘겨진다.

- 4) 목적지 IP 주소가 추출되고 경로 캐시가 조사되며, 만약 캐시에 엔트리가 존재하지 않으면 주라우팅 테이블에 대한 탐색이 이루어진다. 다음 홉에 대한 정보가 포함된다.
- 5) 만약 패킷이 IPv6 호스트와 IPv4 호스트간의 통신을 바라는 패킷이라면, 변환기는 패킷을 변환하는데, 이 과정에서 맵핑 테이블의 탐색이 이루어진다.
- 6) 라우팅 결정에 따라 포워딩되는 패킷을 적절한 인터페이스로 큐잉시킨다. (인터페이스당 하나의 출력 큐가 존재)
- 7) 패킷은 하드웨어 메모리로 전달되며, 인터페이스 카드는 네트워크 상으로 패킷을 내보낸다.

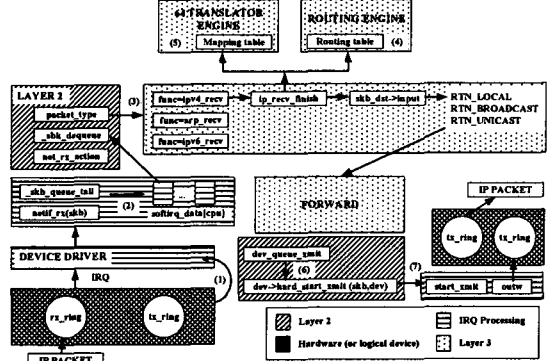


그림 3. IPv6-IPv4 변환시스템의 구조 및 패킷 흐름

3. IPv6-IPv4 변환시스템의 특성

3.1 IPv6-IPv4 변환시스템의 큐잉모델

IPv6-IPv4 변환시스템에서 패킷의 도착은 시스템의 활동과 관계없이 이루어지므로, 도착간격은 독립적이고 동일한 분포를 따른다고 볼 수 있다. 그러므로 도착과정은 도착간격의 지수확률분포로 표현할 수 있다. 또한 서비스 과정 역시 서비스 시간에 대한 확률분포로 표현 가능하며, 이전 서비스 시간과 무관한 것으로 가정하므로, 지수확률분포로 표현할 수 있다. 서비스를 지원하는 서버는 단일서버이며, 서비스 규칙은 선입선출(First Come First Served)의 방식을 취하고 있다. 또한 네트워크 인터페이스 큐에 도착하는 패킷들은 버퍼가 모자라는 경우 상실되므로 큐의 크기는 제한적이며, 여러 개의 네트워크인터페이스 큐가 존재하므로 복수행렬 시스템인 M/M/1/K 모델이라 볼 수 있다.

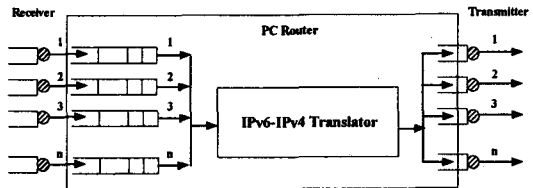


그림 4. IPv6-IPv4 변환시스템 큐잉 모델

그림 4에서  $i(i=1, \dots, N)$  번째 인터페이스를 통한 수

신 작업을 수행하는데 소요되는 시간은 확률변수  $RT_i$ , 평균  $rt_i$ 로 정하였다. 또한  $j$  ( $j=1, \dots, N$ ) 번째 인터페이스의 출력버퍼의 한 전송하는데 소요되는 시간은  $TT_j$ , 평균  $tt_j$ 로 정하였다. 따라서 한 패킷을 꺼내어 전송하는데 소요되는 평균시간  $tt$ 는 식(1)과 같다.

$$tt = \sum_{j=1}^N p_j tt_j \quad (1)$$

여기서  $p_j$ 는  $j$  ( $j=1, \dots, N$ )는  $j$  번째 인터페이스를 통해서 한 패킷이 전송될 확률이다. 또한,  $i$  번째 네트워크 인터페이스를 통해 들어오는 패킷들의 서비스 시간은 독립적이고 동일한 분포를 가지는 확률변수  $S_i$ , 평균은  $E(s_i)$ 로 정하였다.

### 3.2 IPv6-IPv4 변환시스템 변환기의 큐잉모델

IPv6-IPv4 변환시스템의 변환기 부분에 집중하여 분석하는 경우, 변환 서비스를 받기 위한 패킷들은 복수개의 네트워크 인터페이스 버퍼로부터 서버의 라운드 로빈 스케줄링 방식에 의해 일정한 시간단위로 도착하게 된다. 따라서 도착하는 패킷들의 도착간격은 포아송 과정을 따르며, 지수분포를 나타내게 된다. 또한 변환기의 서비스 과정은 필터링과 주소변환 그리고 포워딩의 작업이 순차적으로 수행된다. 따라서 서비스를 받고 있는 패킷 처리의 완료는 이미 받은 서비스의 량에 의존하게 된다. 그러므로 변환기의 서비스분포는 일반분포를 따르는 것으로 볼 수 있다. 그림 5는 IPv6-IPv4 변환시스템의 변환기 자체에 대한 큐잉 모델을 나타낸 것이다.

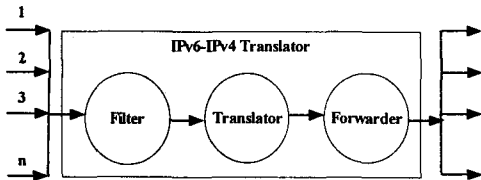


그림 5. IPv6-IPv4 변환시스템 변환기의 큐잉 모델

변환기의 서버는  $N(Q_1, \dots, Q_N)$  개의 큐들을 순차적으로 돌아가면서 서비스하게 된다. 또한 큐 내에서는 선입선출(FCFS) 방식으로 서비스를 수행한다.  $i$  번째와  $i+1$  번째 큐간의 서버교환시간은 독립적이고 동일한 분포를 가지는 확률변수  $s_i$ 로 하였다. 서버의 한 사이클 동안 전체 교환시간의 평균  $E(sw)$ 는 다음과 같다.

$$E(sw) = \sum_{i=1}^N sw_i \quad (2)$$

### 4. IPv6-IPv4 변환시스템의 근사해

시스템의 성능을 근사화하는 몇 가지 방법들이 있는데, 본 논문에서는 시스템이 가지고 있는 요소들을 수학적으로 근사적으로 표현하는 방법을 택하였다. 시

스템의 성능을 분석하기 위하여 성능 척도로서 평균 체제 시간을 사용하였으며, Jirachiefpattana[12]와 Boxma[13]의 방법을 따랐다.

#### 4.1 IPv6-IPv4 변환시스템의 평균체제시간

변환시스템의 평균체제시간은 대기시간과 서비스시간의 합으로 구성된다. 제안된 변환시스템은 패킷들의 수신과 송신을 위한 시간이 포함되어 있다. 따라서 만약  $y_i$ 를  $i$  번째 인터페이스를 통해 시스템에 들어온 각 패킷의 체제시간이라 가정하면 변환시스템의 평균 체제시간  $E(y_i)$ 는 다음 식(3)과 같다[12].

$$E(y_i) = rt_i + E(w_i) + E(s_i) + tt \quad (3)$$

평균서비스시간  $E(s_i)$ 는 실제 변환작업이 이루어지는 변환기의 평균체제시간에 해당된다. 우선,  $i$  번째 인터페이스를 통해 한 패킷이 도착하기 전  $Q_i$ 에서의 큐 길이를  $x_i$ 라 하고, 서버가  $Q_i$ 에서 서비스를 시작해서  $Q_i$ 로 돌아올 때까지의 서버의 한 사이클의 길이를  $c_i$ ,  $i$  번째 인터페이스를 통해 한 패킷의 도착으로부터 서버가  $Q_i$ 로 돌아올 때까지의 추가적인 시간을  $rc_i$ 라 가정한다. 이 패킷의 평균대기시간은 추가적인 시간  $rc_i$ 와 대기하고 있는 패킷 수 만큼의 사이클이 필요하다. 따라서 평균대기시간  $E(w_i)$ 는 Boxma[13]에서 언급한 것처럼 다음의 식과 같다.

$$E(w_i) = E(rc_i) + E(x_i)E(c_i) \quad (4)$$

$$E(w_i) = \frac{1 - \rho + \rho_i}{1 - \rho - \lambda_i E(sw)} \frac{1 - \rho}{(1 - \rho) \rho + \sum_{j=1}^N Var(\rho_j)} \quad (5)$$

$$\cdot \left[ \frac{\rho}{2(1 - \rho)} \sum_{j=1}^N \lambda_j Var(s_j) + \frac{\rho}{2s} \sum_{j=1}^N Var(sw_j) + \frac{s}{2(1 - \rho)} \sum_{j=1}^N \rho_j (1 + \rho_j) \right]$$

$i, j = 1, \dots, N$

그러나, 본 논문에서의 모델은 인터페이스 큐 버퍼 크기가  $k$ 로 한정되어 있으므로, 도착하는 패킷 중에서 입력되지 않는 패킷은 시스템이 상태  $k$ 에 있는 것을 보는 패킷들이다. PASTA(Poisson Arrivals See Time Averages)에 의해 이 확률을  $P_k$ 라 하면 입력률은 식(6)과 같다.

$$\lambda_e = \lambda(1 - P_k) = \frac{\lambda(1 - P_k)}{1 - \rho^{k+1}}, (\rho \neq 1) \quad \frac{\lambda k}{k+1}, (\rho = 1) \quad (6)$$

따라서 최종적인 평균대기시간은 식(6)을 식(5)에 대입하여 식(7)처럼 구할 수 있다.

$$E(w_i) = \frac{1 - \rho + \rho_i}{1 - \rho - \lambda_i (1 - P_k) E(sw)} \frac{1 - \rho}{(1 - \rho) \rho + \sum_{j=1}^N Var(\rho_j)}$$

$$\cdot \left[ \frac{\rho}{2(1 - \rho)} \sum_{j=1}^N \lambda_j (1 - P_k) Var(s_j) + \frac{\rho}{2E(sw)} \sum_{j=1}^N Var(sw_j) \right]$$

$$+ \frac{E(sw)}{2(1-\rho)} \sum_{j=1}^N \rho_j (1+\rho_j) \quad i=1, \dots, N \quad (7)$$

#### 4.2 IPv6-IPv4 변환시스템의 변환기 평균체제시간

Jirachiefpattana[12]의 모델과 달리 우리는 변환시스템의 변환기 부분을 하나의 시스템으로 보았다. 전체 변환시스템의 서비스 시간은 변환기의 평균체제시간에 해당된다고 볼 수 있다.

시스템내 전체 패킷수에 대한 확률함수는 식(8)과 같다[14].

$$P[N=n] = \frac{(1-\rho)\rho^n}{(1-\rho^{K+1})} \quad (8)$$

시스템내 K 개 패킷이 존재할 확률  $P_b$  는 식(9)와 같다.

$$P_b = P[N=K] = \frac{(1-\rho)\rho^K}{(1-\rho^{K+1})} \quad (9)$$

처리율 H 는 식(10)과 같다.

$$H = \lambda(1-P_b) \quad (10)$$

평균 응답시간 T 는 한 패킷의 평균체제시간이다.

따라서 Little 의 공식을 이용하면 변환기의 평균체제시간은 식(11)과 같다.

$$T = \frac{E[N]}{H} = \frac{\rho^{K+1}(K\rho - K - 1) + \rho}{\lambda(1-\rho^K)(1-\rho)} \quad (11)$$

평균체제시간 T 는 결국 IPv6-IPv4 변환시스템의 평균 서비스시간  $E(s_i)$  에 해당된다.

#### 5. 성능분석

패킷 전송은 Ipv6 호스트에서 Ipv4 호스트로 한쪽 방향으로 진행되었다. Ipv6 호스트는 FreeBSD, 500MHz 펜티엄 PC 이며, IPv4 호스트는 WindowXP, 866MHz 모바일 펜티엄 m 이며, Ipv6-Ipv4 변환시스템은 Linux, 333MHz, 펜티엄 PC 이다. IPv6 호스트와 IPv4 호스트는 100Mbps Ethernet 으로 연결되었다. 근사해를 구하는데 필요한 상수값들은 측정값들에 근거하여 선택되었다. 그림 6 은 IPv6-IPv4 변환시스템의 성능을 분석하기 위해 32, 1440 바이트의 패킷에 대한 식(4)를 이용한 근사해와 네트워크 상에서의 측정결과를 나타낸 것이다.

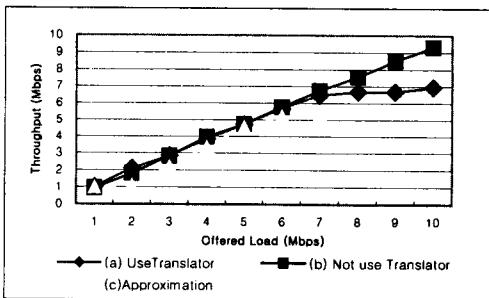


그림 6. 근사와 측정에 의한 처리율  
그림 6 에서 보는 바와 같이 라우터는 제공되는 부

하가 증가할수록 처리율도 증가한 반면, 변환기는 로드가 증가할수록 포화상태에 이르는 것으로 나타났다. 근사화된 변환기의 평균체제시간은 입력률이 증가함에 따라 포화상태에 이르는 것으로 나타났으며, 결국 이는 전체 변환시스템의 성능에 영향을 미치는 것으로 나타났다. 향후 변환기의 특성에 따른 시간 지연의 주요 요인에 관한 보다 구체적인 연구가 요구된다.

#### 6. 결론

IPv6 의 도입을 위해서는 IPv6 네트워크와 IPv4 네트워크가 서로 통신할 수 있는 변환시스템의 개발이 필수적이며, 많은 변환시스템들이 개발되어 왔다. 본 연구에서는 게이트웨이 레벨의 IPv6-IPv4 변환시스템의 성능을 분석하는 방법을 제시하는 것을 목표로 하여 변환시스템에 대한 큐잉 모델에서의 평균체제시간을 근사화하는 방법을 제시하였다.

타당성 검증을 위해 변환시스템에 대한 큐잉 모델을 대상으로 평균체제시간을 계산하고 이를 측정 결과와 비교하였다.

본 연구에서는 IPv6-IPv4 변환시스템의 성능 분석에 필요한 방법론적인 토대를 제시하였으며, 향후 추가적인 서비스 기능이 지원되는 변환시스템에 대한 성능 분석 모델에 대한 연구가 진행중이다.

#### 참고문헌

- [1] F.Solensky, IPv4 Address Lifetime Expectations in Ipnng, Addison Wesley, 1996.
- [2] K.P. Worrall, "The impact of IPv6 on Wireless Networks", IEEE Conference Publication NO.477, pp.323-329, 2001.
- [3] M. Dobrucki, "The Effects of the transition to IPv6 on Internet Security", pp.1-18, 1999.
- [4] E. Nordmark, "Stateless IP/ICMP Translation Algorithm (SIIT)", IETF RFC 2765, 2000.
- [5] G. Tsirtsis and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", IETF RFC 2766, 2000.
- [6] M. E. Fiuczynski, V. K. Lam, and B. N. Bershad, "The Design and Implementation of an IPv6-IPv4 Network Address and Protocol Translator", Proc USENIX Conference, 1998.
- [7] IPv6 Forum Korea, "TM2001-008 : Linux-based Userspace NAT-PT", <http://www.ipv6.or.kr/>, 2001.
- [8] I.Y. Kong, J.L. Lee, J.T. Lee, "Design and Implementation of IPv6-IPv4 Protocol Translation System using Dynamic IP Address", LNCS 2713, pp.496-509, Springer-Verlag, Berlin, 2003.
- [9] P. Srisuresh, and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", IETF RFC 3022, 2001.
- [10] D. P. Bovet and M. Cesati, Linux Kernel, O'Reilly, 2001.
- [11] P. Gotmarker, Linux Ethernet-howto, 2000.
- [12] Jirachiefpattana, P. County, T.S. Dillon, R. Lai, "Performance evaluation of PC routers using a single-server multi-queue system with a reflection technique", in computer communications '97, Elsevier Science, 1997.
- [13] O.J. Boxma and B. Meister, "Waiting-time approximations for cyclic-service systems with switch-over times", in Performance '86, Elsevier Science, 1986.
- [14] Jianhua Cao, Mikael Andersson, Christian Nyberg, Maria Kihl. "Web Server Performance Modeling Using an M/G/1/K\*PS Queue", in Telecommunications IEEE, 2003