# IPv6상의 TCP 패킷 모니터링을 위한 사용자 친화적 도구 개발

설순욱*(suseol@icu.ac.kr), 이종국, 김명철, 마중수
한국정보통신대학원대학교

# Development of User Friendly Tool for Monitoring TCP Packet on IPv6

Soonuk Seol*, Jongkuk Lee, Myungchul Kim, Joongsoo Ma
Information and Communications University

## 요 약

현재 인터넷 프로토콜인 IPv4의 주소 부족 등의 문제를 해결하기 위해 차세대 인터넷 프로토콜 IPv6에 대한 연구가 활발히 진행되고 있다. 그러나 새로운 프로토콜 IPv6 상에서 기존의 다른 상위 프로토콜 및 응용이 어떠한 영향을 받게 될지는 의문이다. 한편, IPv6 상에서 새로운 프로토콜 및 응용을 개발할 때에 많은 시험이 요구된다. 이와 같은 이유로 인해 새로운 통신 프로토콜에 있어서 모니터링은 필수적이다. 그러나 지금까지의 많은 모니터링 라이브러리들은 텍스트에 기반하고 있으며, 그래픽 데이터를 제공하는 도구들도 대부분 통계정보의 제공에 초점을 두고 있다. 이로 인해 프로토콜이 그 표준에 따라 정확한 원리에 맞게 동작하는 지를 파악하는 데는 큰 도움이 되지 않고 있다. 이에 본 연구에서는 대상 프로토콜의 실제적인 동작 과정이나 원리를 알 수 있도록 그래픽 기반의 사용자 친화적 모니터링 도구를 제작한다. 이를 위해, 먼저 IPv6가 다른 프로토콜에 영향을 미칠 수 있는 사항들을 분석한다. 다음으로 본 교에 구축되어 있는 IPv6 호스트에 모니터링에 필요한 환경을 구축하고, Java Applet을 이용한 모니터 프로그램을 제작한다. 현재 개발하는 모니터 프로그램은 TCP의 슬라이딩 윈도우(Sliding Window) 기법에 관련된 사항을 모니터링 해주는 것으로 그 범위를 한정한다. 개발된 도구를 이용하여 IPv6 상에서 FTP가 TCP를 이용하여 파일을 전송하는 경우의 모니터링을 실시하고, 그 결과를 분석 제시한다. 이로써, 개발된 사용자 친화적 모니터링 도구가 얼마나 쉽게 슬라이딩 윈도우 기법을 이해시켜 주고, 내포된 의미를 파악할 수 있게 해주는지를 알 수 있다

## 1. 서론

인터넷은 수 년 동안 지속적인 규모의 증가를 보였고, 이에 따라 주소 공간의 부족, 부하의 급증과 같은 문제가 대두되었다. 또, 실시간 음성 및 동화상 전송과 같은 새로운 요구가 일어남에 따라 인터넷 프로토콜 변경의 필요가 절실해 졌다. 이에 인터넷 프로토콜 (Internet Protocol)의 새로운 버전인 IPv6가 출현하였고, 지금도 연구가 활발히 진행되고 있다.

새로운 인터넷 프로토콜 IPv6는 IPv4에 비하여 128bit 네트워크 주소 사용, 융통성 있는 헤더 양식의 사용, 향상된 선택사항, 자원 할당의 지원, 프로토콜 확장을 위한 준비, 송신지에서만의 프래그맨테이션(Fragmentation) 허용, 헤더체섬(header checksum)의 제거와 같은 차이를 갖는다[6].

언제든지 새로운 프로토콜의 개발 단계에서는 많은 시험을 수행하게 된다. 여러 벤더에서 프로토콜을 구현하면, 그 구현물이 표준에 맞는지, 또 서로 다른 벤더들 간의 상호 운용이 잘 되는지를 확인하는 절차가 필요하다. 특히 초기 단계에서는 표준 자체가 문제가 없는지 확인하는 것이 필수적이다. 현재 많은 인터넷 프로토콜 및 응용에서 사용되고 있는 TCP (*Transmission Control Protocol*)의 경우도 초기 구현 당시 실리 윈도우 증후군 (Silly Window Syndrome)과 같은 문제가 발견되기도 하였다[6].

IPv6가 등장함에 따라 동일한 문제가 야기될 수 있다. IPv6로의 변화에 있어서 몇 가지 고려해볼 사항으로 증가된 주소 처리와 라우팅 테이블 관리로 인한 많은 트래픽, 새로운 흐름 제어의 활용, 송신지에서만의 프래그맨테이션, IPv4에서 IPv6로의 전환기 등에서 야기되는 문제점들이다.

이와 같이 프로토콜의 개발 단계에서 오류를 찾는데 유용하게 사용되는 것이 모니터링이다. 모니터링을 통해 네트워크 상으로 오가는 패킷을 포획(capturing)하여 각 패킷에 설정되어 있는 값을 알 수 있다. 이를 지원하는 프로그램으로서 Libpcap v6[1]와 같은 라이브러리 및 TCPdump[2]가 널리 사용된다. 그러나, 이들은 패킷의 각 필드 내용을 수치적 혹은 문자적 데이터만으로 표현하고있어 쉽게 각각의 필드가 뜻하는 바를 확인할 수 없다. 뿐만 아니라, 선행되는 패킷과 뒤따르는 패킷간의 관계에 대한 정보도 제공하지 않고 있다

한편, 그래픽 정보를 제공하는 기존의 모니터링 도구들은 특정 서버에 접속하는 호스트의 시간별 접속 정도, 사용하는 응용의 파악이라는 데 초점을 두고 있다. 이는 프로토콜의 작동 과정이나 원리를 파악하거나 오류를 발견하고자 하는 목적과는 부합되지 않는 것이다.

이에 본 연구에서는 IPv6 상에서 동작하는 TCP의 슬라이딩 윈도우 기법과 관련된 사항을 모니터링하여 정상적인 동작을 하는지 확인하고, 실질적인 동작을 알기 쉽게하는 그래픽 기반의 정보를 제공하는 모니터링 도구를 제작한다.

본 논문의 구성은 다음과 같다. 2장에서 본 논문에서 사용하는 네트워크 모니터인 TCPdump의 동작 원리와, TCP의 슬라이딩 윈도우 기법(Sliding Window Mechanism)에 대해 기술한다. 3장에서는 실제 개발에 대해 설명하고, 4장에서 모니터링 실험을 통해 결과를 분석하고, 끝으로 5장에서 결론을 맺는다.

## 2. 관련 연구TCPdump

이 절에서는 모니터링 도구에서 사용하는 TCPdump의 동작 원리에 대하여 알아보고, 구현된 모니터링 도구가 보여주게 될 TCP의 슬라이딩 윈도우에 관련한 사항에 대하여 기술한다.

### 2.1 TCPdump

TCPdump는 UNIX와 TCP/IP환경에서 사용되는 잘 알려진 네트워크 모니터이다[3]. TCPdump및 그와 유사한 패키지들은 다양한 네트워크 파라매터들을 표현하는 수많은 양의 데이터를 모을 수 있는 능력이 있다 이들 패키지들이 특정 파라매터나

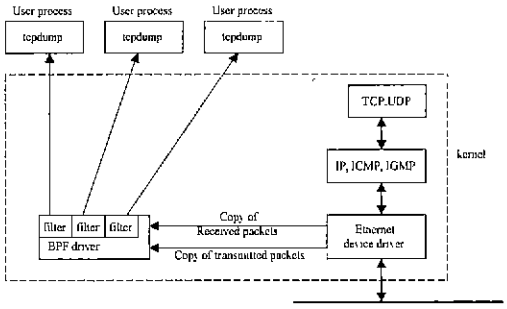


그림 1. BSD Packet Filter[4]

트래픽의 타입에 초점을 맞추어 데이터를 필터링할 수 있는 유용한 기능이 있는 반면, 데이터가 지니는 의미를 보여주거나 이해 시키는 데는 한계가 있거나 존재하지 않는다 많은 양의 데이터를 분석해야 하는 시뮬레이션에서는 Visualization이 요구된다 Visualization은 또한 네트워크 실험으로부터의 데이터를 이해시키는 것을 확실히 보장할 수 있다[3].

현재 BSD(Berkeley Software Distribution)에서 나온 커널은 TCPdump가 네트워크 인터페이스로부터 패킷을 포획하고 필터할 수 있도록 하는 한 방법인 BSD Packet Filter(BPF)를 제공한다 그림 1은 Ethernet에서 사용된 BPF의 모습이다[4].

그림 1에서와 같이 TCPdump는 Ethernet으로 전송되어 오는 것과 나가는 모든 패킷을 BPF를 통해 복사 본을 받는다 이 과정에서 Libpcap 이라는 패킷 캡춰 라이브러리를 사용한다 이 라이브러리는 필터링을 요하는 명령을 해석하고 파싱하여 필요한 정보만을 필터링 해주는 기능을 한다

TCPdump는 캡춰된 각각의 패킷에 대해서 다양한 정보를 제공하는데, 이를테면, 도착 시간, 근원지 주소, 목적지 주소, 기타 다양한 TCP/IP 플래그, 데이터의 크기 같은 것들이다 TCPdump 이외의 유사한 프로그램은 Solaris 2.2의 snoop. AIX 3.2.2에서 제공하는 iptrace 등이 있다[4]

### 2.2 TCP와 슬라이딩 윈도우 (Sliding Window) 기법

모든 TCP 패킷은 동일한 기본 헤더를 갖는다 TCP는 패킷의 제어 필드 (Control Field)에 의해 서로 다른 기능을 하게 된다 예를 들면, 최초의 연결 설정 시에 제어 필드의 'SYN'을 의미하는 비트가 1로 설정되고, 패킷의 확인 응답을 위해서는 'ACK'를 의미하는 비트가 세팅된다.

슬라이딩 윈도우는 스트림 전송을 효율적으로 할 수 있게 한다. 송신자와 수신자가 신뢰성을 확보하기 위해 확인응답을 하는데, 보내지는 매 패킷마다 확인응답을 한다면 망의 대역폭을 충분히 활용하지 못하게 된다 반면 슬라이딩 윈도우 프로토콜을 이용하면 송신자가 확인응답을 기다리기 전에 다중의 패킷을 전송하는 것이 가능하기 때문에 효율적이다 그림 2와 같이 슬라이딩 윈도우를 일련의 패킷이 전송되는 것으로 생각하면 된다. 이 프로토콜은 순서대로 작은 고정 크기의 윈도우를 통과 이 윈도우 내에 속하는 모든 패킷을 전송한다

전송되었으나 확인응답이 수신되지 않으면 이 패킷은 비확인응답(unacknowledged)되었다고 한다[6]. 그림 2에서 패킷 세그먼트 4, 5, 6이 바로 그것이다. 기술적으로, 어떤 주어진 시간에 비확인응답 될 수 있는 패킷의 수는 윈도우 크기(그림에서 사각형 상자)에 의해 작고 고정된 수로 제한된다. 이 윈도우 사이

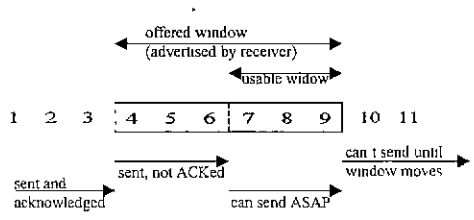


그림 2. TCP Sliding window[4]

```
tcpdump -i eth0 -l port 20 | tee dat
23:53:17.029627 ipv6-1.etri.re.kr.ftp-data > mum.icu.ac.kr.1819: S
2027222321:2027222321(0) win 16384 <mss 1460> (DF) [tos 0x8]
23:53:17.029728 mum.icu.ac.kr.1819 > ipv6-1.etri.re.kr.ftp-data: S
132440322:132440322(0) ack 2027222322 win 31856 <mss 1460> (DF)
23:53:17.042711 ipv6-1.etri.re.kr.ftp-data > mum.icu.ac.kr.1819: .
ack 1 win 17376 (DF) [tos 0x8]
```

그림 3. TCPdump의 실행 예

쯔는 수신자의 여건에 따라 변화되기도 한다. 수신 쪽에서는 버퍼가 차게 됨에 따라 작은 윈도우 크기를 ACK을 통해 알려오게 된다

## 3. 모니터링 도구 개발

이 절에서는 모니터링 도구의 개발에서의 범위, 제작 과정과 동작 원리, 그리고 정보의 표현에 관하여 기술한다

### 3.1 범위

본 논문에서 구현하려고 하는 모니터는 TCP 패킷에서 슬라이딩 윈도우와 관련한 일부 정보만을 나타내기로 한다. 기존의 텍스트에 기반한 모니터에 대해 그래픽에 기반한 모니터의 제작에 초점을 두고 있기 때문에 수치적인 데이터나 문자 정보 등의 표현은 제외한다. 여기서 중심적으로 나타낼 내용은 두 가지로 나누어 볼 수 있다 첫 번째는 모니터링 하려는 컴퓨터에 오가는 패킷의 시간 간격을 선 그래프로 표현을 해 주는 것이고, 두 번째는 TCP의 슬라이딩 윈도우 기법을 파이 그래프로 나타내는 것이다. 이를 위해 살펴보아야 할 TCP의 관련 필드는 다음과 같다

- Source/Destination Port
- Sequence Number
- Acknowledgment Number
- Control Filed(URG, ACK, PSH, RST, SYN, FIN)
- Window Size

### 3.2 제작 과정과 동작 구조 및 원리

모니터링 도구를 제작하기 앞서 우리는 모니터링 시험 환경을 구축하였다. IPv6 호스트간의 파일 전송을 대상으로 모니터링 하기 위해 IPv6 환경을 구축하였다. 이에 관한 설명은 본 논문의 범위 밖이므로 생략한다. 제작 과정의 개요는 다음과 같다. 먼저, 패킷을 포획하는 TCPdump에서 활용하게 될 Libpcapv6[1]를 설치한다. 다음으로, TCPdump의 결과를 우리가 사용하기 좋은 형태로 나타내기 위해 TCPdump 프로그램을 수정한다. 그림 3에 TCPdump의 실행 예가 나타나 있다. 이제 포획된 데이터를 분석하여 그래픽 정보를 제공하도록 하는 프로그램을 제작한다. 개발자로 하여금 언제, 어디서, 어떤 플랫폼에서나 모니터링 정보를 볼 수 있도록 하기 위해 Java 애플릿을 이용하여 프로그램을 개발하고 웹 브라우저를 통해 확인할 수 있도록 한다. 이렇게 개발되는 모니터링 도구는 그림 4와 같이 동작하게 된다

먼저 ICU/IPv6에서 TCPdump가 실행되고, ETRI/IPv6에서는 FTP 어플리케이션을 통해 파일을 전송한다. 그러면 TCPdump가 포획을 하고 그 결과를 로컬 디스크에 파일(Data)로 저장한다. 이런 이후에 인터넷에 접속되어 있는 컴퓨터가 ICU의 Web

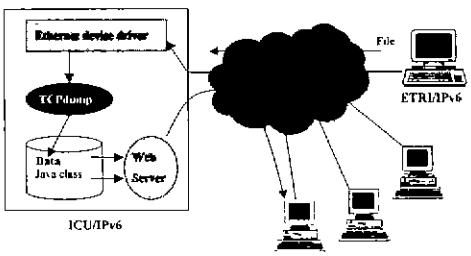


그림 4 모니터의 동작 과정

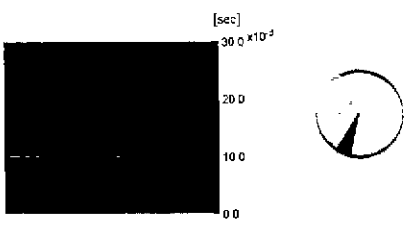


그림 5. 파일 전송 초기 단계

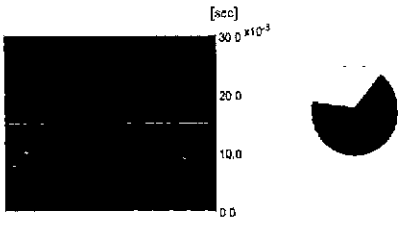


그림 6. 재전송 요구 중인 상태

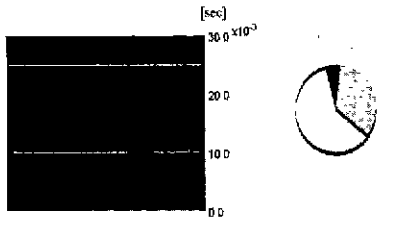


그림 7. 재전송 받은 후의 상태

Server에 접속하면 홈페이지에서 Java class가 전송되어 실행된다. 이 Java class는 ICU에 있는 Data를 가져와서 그것을 그래피컬하게 보여 주게 된다.

### 3.3 정보의 표현

모니터링하여 제시할 정보는 앞서 설명한 바와 같이 Ethernet 인터페이스에 오가는 패킷 사이의 시간간격(Interarrival Time [3])과 슬라이딩 윈도우이다 시간간격은 인터페이스로 들어오는 패킷뿐만 아니라 나가는 패킷까지 고려한다는 사실을 주목하여야 한다. 현재의 시간간격 값은 최근에 도착한 패킷과 현재 도착한 패킷 사이의 차이가 된다. 이 값이 적다는 것은 이 컴퓨터가 바쁘다는 것을 의미한다 즉, 적은 시간간격동안 계속해서 패킷을 처리해야 하기 때문이다. 반대로 이 값이 크다는 것은 패킷을 처리하는데 여유가 있다는 것이다. 네트워크의 입장에서 바라볼 경우, 시간간격 값이 크다는 것은 그만큼 그 패킷이 전송되는데 오래 걸렸다는 것이고, 이것은 상대 컴퓨터에서의 지연이 적다고 가정할 경우 네트워크의 트래픽이 높다는 것을 의미한다. 제작하는 모니터에서는 시간간격을 나타내기 위해 2차원 그래프를 사용하였다 x축은 패킷의 순열을 의미하는 패킷 Number, y축은 시간간격을 표시하도록 했고 그 단위는 $10^{-3}$ sec로 했다. 그래프의 시작은 오른쪽부터이고 패킷이 도착할 때마다 그래프는 왼쪽으로 이동한다.

슬라이딩 윈도우 개념을 비주얼하게 보여주는 방법은 여러가지가 있다. 한 방법은 앞서 보았던 그림 2에서와 같이 패킷의 열에 시각형 윈도우 박스가 이동하는 것을 보여주는 방법이 있을 것이다 이 경우 전송할 데이터의 전체 크기가 얼마나 큰가에 따라 문제가 생길 수 있다 큰 데이터의 경우 한 화면에 나타내기 위해서는 상대적으로 각 패킷의 크기가 작아져야만 한다 그렇지 않으려면 윈도우를 이동하는 대신 패킷 열을 이동시켜야 한다. 이것은 슬라이딩 윈도우의 이름에 맞지 않아 오히려 혼동을 야기할 수 있다. 따라서 본 논문에서는 파이 그래프로 표현하기로 한다. 파이그래프에서는 패킷이 일정한 크기를 유지할 수 있고 부채꼴의 창이 있는 원이 윈도우를 대신할 것이다. 자세한 모습은 그림 5를 참조하기 바란다.

윈도우의 크기는 상대 컴퓨터에서 알려져 온 값인데, 이것은 곧 상대 컴퓨터의 버퍼를 의미하기도 한다. 결국 파이 그래프로 표현한 슬라이딩 윈도우는 수신자의 상황을 반영하는데 초점을 두는데 의의가 있다

## 4. 모니터링 실험 결과

이 절에서는 모니터링을 실시하여 그 결과를 분석하여 제시한다. 실험은 ETRI(ipv6-1.etri re kr)로부터 ICU(mum.icu ac.kr)로 375599bytes 크기의 파일을 전송하는 과정에서 TCP 패킷을 포획하고 모니터링하는 것이다. 파일을 완전히 전송하는데는 3.76초가 소요됐다. 이 때, 송수신단에 다른 응용을 실행하여 트래픽을 가했다. 총 428개의 패킷이 수신되었다.

웹브라우저에서 나타난 그래픽 기반 모니터링 결과를 분석하여 보자. 그림 5를 보면, FTP를 통해 파일을 전송 받을 때 최초에는 전송되는 파일의 도착 시간이 길다가 점점 줄어서 파일 전송이 종료될 때까지 비슷한 값을 갖게 됨을 알 수 있다. 시간간격이 30 msec를 웃도는 수준에서 10msec 주위의 값으로 내려오게 된다. 이것은 초기에 긴 경로로부터 점점 짧은 경로를 찾다가 이후는 같거나 비슷한 거리의 경로를 택한 것을 의미한다 그림에서 보면 0 msec에 가까운 값들이 사이에 존재하는데 이것은 수신지에서 ACK을 보내는 패킷이 자신의 Ethernet 인타페이스에 도착한 시간이다

다음으로 주목할만한 사실은 10msec 주위의 값을 갖는 연속된 패킷 수는 1에서 3개 정도라는 사실이다. 이것은 파이 그래프를 통해서도 알 수 있는데 패킷이 두개 정도 체워지면 이내 ACK을 받아 패킷 영역만큼 윈도우가 이동하게 된다. 64.6%의 패킷에 대한 확인응답이 있었다. 패킷 2개 정도를 받을 때마다 ACK를 보낸 다는 것이다. 하지만 실제 중간에 재전송을 위해 패킷을 하나 받을 때마다 계속해서 재전송 ACK를 보내는 경우가 있었다. 그림 6을 살펴보면, 매 패킷 마다 ACK을 보냈음을 알 수 있다. 하지만 슬라이딩 윈도우가 이동하지 않고 패킷이 체워진 것을 보면 보내진 ACK Number가 동일한 값임을 알 수 있다. 즉, 수신지에서는 그 ACK Number를 갖는 패킷이 오기까지 기다리고 있는 것이다. 그림 7에서 보면 30msec이상의 지연을 갖는 패킷이 하나 있음을 알 수 있다. 이것이 바로 수신지에서 요구하던 패킷이다. 재전송 이후에는 계속해서 평균 두개의 패킷마다 확인 응답이 있었다. 문제는 왜 송신지에서 수신지에서 요구한 ACK Number를 갖는 패킷을 빨리 재전송하지 않았는가 하는 것인데, 아마도 ACK 패킷이 송신지에 도착되지 않았거나 늦게 도착했을 가능성이 크다. 즉, 이미 윈도우 사이즈만큼 패킷을 다 보낸 이후에 그 패킷을 받았다는 것이다. 이 것을 확인하기 위해서는 송신지 쪽에서도 캡춰를 수행해서 양쪽의 상황을 같이 비교해야 할 것이다.

앞으로 점점 증가할 트래픽에 대해 네트워크에 부하를 줄이는 것이 중요해 지는데, 본 논문의 실험 결과에서 알 수 있듯이 수신지에서는 즉시 ACK을 보냄으로써 네트워크 트래픽을 가중하고 있다. 그리므로 네트워크의 트래픽에 따라 ACK를 보내는 빈도를 달리하는 TCP가 일반화될 필요가 있을 것이다

## 5. 결론

차세대 인터넷 상에서의 많은 프로토콜을 시험하기 위해서 프로토콜의 동작 과정과 원리의 이해를 돕는 그래픽 기반의 네트워크 모니터가 요구되는 가운데, 본 논문에서는 IPv6 상에서 TCP 패킷을 포획해서 그래픽 정보로 나타내는 진보된 네트워크 모니터링 도구를 개발하였다.

기존의 덱스트로만 보아오던 모니터링 결과(그림 3)보다 개발된 도구를 이용하여 모니터링 한 결과가 훨씬 이해하기 쉬웠고 패킷의 전후 관계를 잘 파악할 수 있었다. 파일전송과 관련해서 수행된 실험에서, TCP의 슬라이딩 윈도우 프로토콜이 어떻게 동작하는가, 실제 수신지에서 얼마나 자주 ACK을 보내는가, 재전송은 어떻게 일어나는가 등을 명확히 알 수 있었다. 또한, IPv6에서 실행되는 TCP는 문제없이 동작함을 알 수 있었다. 이것은 실험에서의 IPv6 간의 패킷은 IPv4로 인캡슐되어 중간 전송은 IPv4의 라우팅 기법에 따랐기 때문일 것으로 판단된다 향후 복잡한 IPv6가 늘어나고, IPv6 위에 새로운 프로토콜 및 응용이 개발되는 경우 본 논문에서 제시한 것과 같은 사용자 친화적 모니터링 도구가 유용할 것으로 기대된다.

## 참고문헌

[1] D Y.Kang, Libpcap v6, The Packet Capture library, http://dutch.antd.nist.gov/~dykang
[2] Van Jacobson and etc, The protocol packet capture and dumper program, http://www-nrg.ee lbl gov/nrg html
[3] Russel O Cleaver, and Scott F. Midkiff, "Visualization of Network Performance using the AVS Visualization System", IEEE, 1994, pp 407-408.
[4] W. Richard Stevens, "TCP/IP Illustrated, Volume 1:The Protocols", 1994.
[5] Douglas E. Comer, "Internetworking With TCP/IP Vol I: Principles, Protocols. and Architecture 3rd ", 1997.
[6] Stephen A. Thomas, "IPng and the TCP/IP Protocols: Implementing the Next Generation Internet", 1996