

종단 간 성능 측정을 위한 인프라*

이승준[†] 이경진[‡] 송왕철[‡] 전길남[‡]
 한국과학기술원[†]
 제주 대학교[‡]

sjlee@an.kaist.ac.kr indigo@nmlab.cheju.ac.kr philo@cheju.ac.kr chon@cosmos.kaist.ac.kr

eTOP: End-to-End Performance Measurement Infrastructure

Seungjun Lee[†] Kyung Jin Lee[‡] Wang-Cheol Song[‡] Kilnam Chon[‡]
 Dept. of Computer Science, Korea Advanced Institute of Science and Technology[†]
 Division of Telecommunication Engineering, Cheju National University[‡]

요 약

차세대 네트워크 인프라에서 높은 대역폭을 필요로 하는 어플리케이션의 품질 보장에 대한 요구가 증가함에 따라, 네트워크의 성능을 측정하고 분석하여 특정 네트워크 지역의 병목현상이나 네트워크 장애에 대해 사전에 대비하도록 하는 장기적, 주기적으로 성능을 측정하는 인프라에 대한 필요성이 대두되고 있다. 우리는 본 논문에서, 기가급의 대역폭을 서비스를 제공하는 차세대 네트워크인 연구망의 성능을 사용자 입장에서 측정할 수 있는 인프라를 제안한다.

1. 서 론

연구망은 차세대 인터넷 기술 및 서비스 개발에 필요한 네트워크 환경을 제공해 주는 비영리 네트워크로 기가급의 대역폭을 서비스한다. 우리나라의 첨단연구개발망(KOREN)[1], 초고속 연구망(KREONET), 미국의 Internet2가 대표적인 예이다.

지금까지 대학과 기업을 중심으로 한 우리나라의 차세대 인터넷 연구를 살펴보면 인터넷에서 사용할 수 있는 응용 서비스의 개발과 연구, 응용 서비스를 제공할 수 있는 하드웨어적 인프라를 구축하기 위한 네트워킹 기술 분야에 대한 연구로 편중되어 있었다. HDTV(High Definition TV) over IP, 고에너지 물리학(High Energy Physics)의 large file transfer 등 높은 대역폭을 필요로 하는 어플리케이션의 요구가 증가함에 따라, 차세대 네트워크의 성능을 측정하고 분석하여 특정 네트워크 지역의 병목현상이나 네트워크 장애에 대해 사전에 대비하도록 하는 매저먼트(measurement) 연구에 대한 필요성이 대두되고 있다. 매저먼트 연구를 위해 필수적인 것이 장기적, 주기적으로 성능을 측정하는 매저먼트 인프라이다. 그러나 우리나라 연구망에서의 매저먼트 인프라는 사용자의 네트워크 사용량 파악을 위한 수동적인 트래픽 측정(passive measurement)으로 제한적이라서, 연구망이 사용자 어플리케이션에게 어느 정도의 서비스 품질을 제공할 수 있는지에 대한 정보는 전무하다.

사용자 입장에서의 인터넷의 성능을 측정하기 위한 기존의 인프라인 Skitter[2], Surveyor[3], PingER[4] 등은 연구망의 성능 측정을 위한 인프라로는 적합하지 않다. 왜냐하면 측정하는 정보가 지연(delay)과 손실률(loss)로 제한적이기 때문에, 어플리케이션 종류에 따른 실제 사용자가 느끼는 성능과는 거

리가 멀다.

본 연구에서는 높은 대역폭을 제공하는 연구망을 사용하는 종단 간의 성능 측정을 위한 인프라(infrastructure)인 eTOP에 대해 소개한다. 2장에서 제안하는 시스템에 대해 상세히 설명하고, 3장에서 향후 계획에 대해 언급하면서 마무리한다.

2. 시스템 개요

2.1 성능 측정 지표

우리는 연구망의 성능의 지표로 사용될 수 있는 정보에 대해 <표 1> 과 같이 정의하였다.

분류	성능 정보	측정 툴
Perf_Basic	Round Trip Time (ms)	ping
	Packet Loss (%)	
Perf_TCP	Bandwidth (kbps)	lperf
Perf_UDP	Bandwidth (kbp)	lperf
	Packet Loss (%)	
	Jitter (ms)	

<표1> 측정 성능

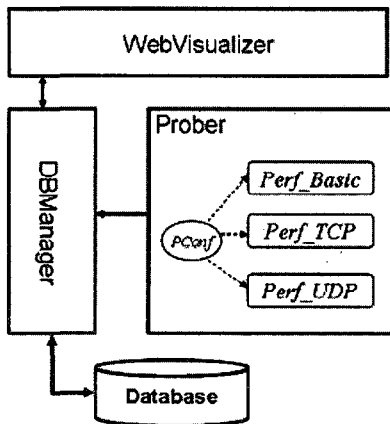
네트워크의 기본적 성능(Perf_Basic)을 나타내는 지표로써 지연(RTT)과 손실률(packet loss)을 정의하였다. 이것의 측정을 위해서는 ping을 사용하였다. 네트워크 어플리케이션은 TCP나 UDP를 사용하고, 두 프로토콜에 따라 요구되는 성능 정보가 다르기 때문에 우리는 어플리케이션을 위한 성능을 TCP의 성능을 나타내는 Perf_TCP와 UDP의 성능을 나타내는 Perf_UDP로 구분하였다. Perf_TCP의 성능 정보로는 사용가능한 대역폭(bandwidth)를 Perf_UDP의 성능 정보로는 사용가능한 대역폭, 손실률, 지터(jitter)를 정의하였다. 그 둘은 트래픽 생성기

* 본 연구는 한국전산원과 첨단정보기술연구센터를 통한 과학재단의 지원을 받아 수행되었음.

(traffic generator)인 Iperf[5]를 이용하여 측정하였다. UDP는 송신자가 보내려고 하는 만큼 보내는 것이기 때문에 실제로 사용가능한 대역폭이라는 개념이 존재하지 않는다. 송신자가 보내려고 하는 패킷의 얼마를 잃어 버렸는가를 나타내는 손실률(loss)이 더 중요한 정보이다. 따라서 우리는 TCP의 사용가능한 대역폭에 100Kbyte를 더한 크기의 트래픽을 UDP 성능 측정을 위해 사용하였다. 100Kbytes라는 수치는 iperf로 TCP의 사용 가능한 대역폭을 측정 한 후, 그 값에 계속 100bytes씩 더해가면서 UDP의 손실률이 0에 가까우면서 UDP 대역폭은 최대가 되는 값을 찾아서 나온 수치이다.

2.2 시스템 구조

본 시스템은 성능을 측정 하는 Prober, 측정결과를 시각화하여 보여주는 WebVisualizer, 데이터베이스와 관련된 작업을 처리해주는 DBManager로 구성되어 있다.



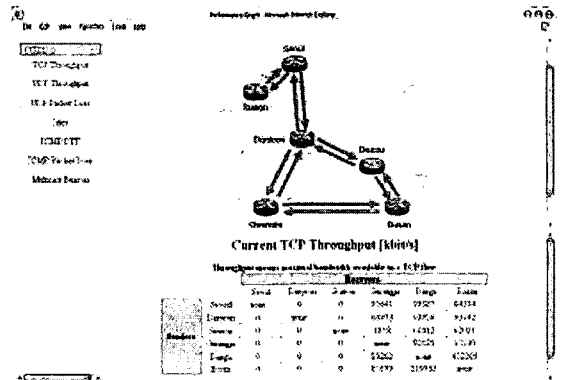
<그림 1> 시스템 구조

1) Prober

각 Prober는 Iperf를 서버를 데몬(daemon)으로 수행하고 있고, cron에 측정 주기를 지정해 준다. Prober는 성능을 측정할 상대방의 IP 주소가 적힌 파일을 읽어, 목적지 리스트를 만든다. 그 다음에 각각의 호스트에게 다음과 같은 순서로 측정 작업을 수행한다. 맨 먼저, 56bytes짜리 ICMP 패킷을 보내 상대방 호스트가 살아있는지를 확인한다. 살아 있으면 Perf_Basic, Perf_TCP, Perf_UDP를 순서대로 실행한다. Perf_Basic은 ping의 측정 결과를 파싱하여 RTT(Round Trip Time)과 손실률(Packet Loss)의 최대, 최소, 평균값을 뽑아낸다. ping을 구동시킬 때 실행 간격(time interval), 패킷 크기(packet size), 보내는 패킷 갯수(count)를 사용자가 지정하여 보내게 된다. 이러한 ping의 옵션들은 PConf 에서 설정하면 된다. Iperf[5]는 NLANR에서 개발하였고, TCP/UDP 트래픽을 생성하여 각각의 대역폭을 측정한다. Perf_TCP와 Perf_UDP는 이 툴을 사용한다. Perf_TCP는 iperf를 구동하여 TCP 트래픽 전송하여 TCP 어플리케이션이 사용할 수 있는 대역폭(bandwidth)를 측정한다.

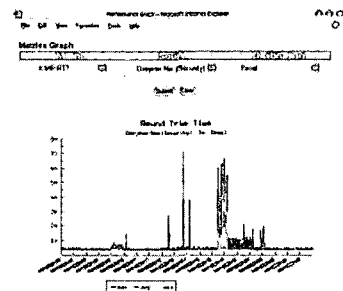
Iperf를 TCP 윈도우 크기, 트래픽 전송 시간, parallel stream 수 값을 사용자가 직접 PConf 에 설정한 값으로 구동시킬 수 있도록 되어있다. Perf_UDP는 Iperf를 구동하여 UDP traffic을 전송하여 UDP 어플리케이션이 사용할 수 있는 대역폭과, 손실률, 지터(jitter)를 측정한다. 측정된 파싱된 결과는 DBManager를 호출하여 데이터베이스에 저장한다.

2) WebVisualizer



<그림 2> 초기화면

WebVisualizer는 데이터베이스에 저장된 값을 웹에서 사용자에게 편리한 형태로 보여준다. <그림 2>는 초기화면으로 좌측에 측정되어진 성능 정보들이 나오고, 중앙 위쪽에 Prober들의 토폴로지를 보여준다. 중앙 아래쪽은 해당 성능 정보의 현재 값을 행렬 형태로 보여준다. 토폴로지 그림에서 화살표를 클릭하면 <그림 3>과 같이 측정값을 선 그래프 형태로 보여줌으로써 시간에 따른 성능 값의 변화를 알 수 있다. Prober가 설치된 장소의 이름을 클릭하면 <그림 4>와 같이 현재 측정값을 막대그래프 형식으로 보여준다. <그림 5>와 같이 측정된 값을 raw data 형태로 볼 수 있다. raw data 형태로 보는 경우 툴의 옵션 값과 같은 상세한 정보도 볼 수 있다.



<그림 3> 선 그래프

