

# 가상 토폴로지와 지역 조정 항을 이용한 네트워크 거리 추정

이상환  
국민대학교 컴퓨터 학부

## Network Distance Estimation Scheme with Virtual Topology and Local Adjustment Term

Sanghwan Lee  
Kookmin University, School of Computer Science  
E-mail : sanghwan@kookmin.ac.kr

### 요 약

네트워크 거리 (Network distance : 일반적으로 ping이나 traceroute 등으로 측정 가능한 round trip time 등 네트워크 상에서 패킷의 전송 시간) 추정 기법은 인터넷상의 많은 응용프로그램들에서 유용하게 사용된다. 예를 들면 다수의 서버를 인터넷상에 설치하고자 하는 경우 사용자들 간의 네트워크 거리를 알고 있다면 서버와 사용자간의 왕복 전송 시간 (Round Trip Time)등을 최소화할 수 있도록 서버를 분산하여 설치하는 구성을 도출해 낼 수 있을 것이다. Peer to Peer 응용 프로그램들에서도 이 네트워크 거리 정보는 매우 유용하다. 기존에 존재하는 추정 기법들은 대부분 유클리드 공간 좌표 기반 기법들로서 유클리드 좌표 상의 거리가 실제 네트워크 거리와 유사하도록 유클리드 공간 좌표를 지정한다. 그러나 이런 방법들의 문제점은 인터넷 상의 네트워크 거리가 삼각 부등식을 만족하지 않는 경우가 존재하는 등 유클리드 공간의 기본적인 가정을 만족하지 못한다는 데 있다. 이런 문제점 때문에 새로운 모델이 필요하고, 이 논문에서는 가상 토폴로지 (Virtual Topology) 모델과 지역 조정 항 (Local Adjustment Term) 모델을 제시하고, 기본적인 성능 분석을 시도하였다.

### 1. 서론

현재 사용되는 많은 인터넷 응용 프로그램들의 성능은 각 인터넷 호스트 간의 물리적인 거리 뿐만 아니라 네트워크 거리 (Network distance : 일

반적으로 ping이나 traceroute등으로 측정 가능한 round trip time 등 네트워크 상에서 패킷의 전송 시간)에 많이 의존한다. 예를 들면 웹 서버나 peer to peer 프로그램들에서의 파일 다운 로드

시간은 기본적으로 서버와 클라이언트 간의 왕복 시간 (Round Trip Time)에 크게 달려있다. 또한 복수의 서버를 인터넷 상에 설치하고자 할 때도 예상되는 사용자들로부터 서버까지의 네트워크 거리를 최소한으로 줄일 수 있는 장소에 서버들을 설치하고자 할 것이다. 이러한 방식을 사용함에 있어서, 각 호스트 들간의 네트워크 거리에 관한 정보는 매우 유용하다. 하지만 각 호스트 들간의 네트워크 거리를 직접 측정하는 방식은  $N^2$ 의 측정을 요하기 때문에 사용하기가 실제적으로 불가능할 경우가 많다. 따라서 실제의 측정 양을 줄이면서 네트워크 거리를 추정 (Estimation)할 수 있는 기법들이 요구되는데 몇 해 전부터 유클리드 공간에 호스트들을 대응 시키는 기법들이 연구자들 사이에서 많이 논의 되었다.

Eugene Ng등이 발표한 GNP (Global Network Position) 기법은 각각의 호스트에서 미리 정해진 20여 개 정도의 랜드마크 (Landmark) 로의 네트워크 거리를 측정한 다음 이 정보를 바탕으로 해서 각 호스트에게 가상 유클리드 공간의 한 좌표를 부여하는 것이다 [1]. 각 호스트들간의 이러한 좌표를 이용하여 유클리드 거리를 계산하면 이 거리가 실제 네트워크 거리의 추정 값이 된다. 이런 유클리드 공간 대응 기법은 그 방식이 매우 단순하기 때문에 많은 연구자들에 의해 발전되었는데, Virtual Landmark 기법이나 ICS 기법들에서는 좌표 계산 시간을 획기적으로 줄일 수 있는 방법을 제시하였고, Vivaldi 기법은 랜드마크가 없이도 좌표를 계산할 수 있는 방식을 제시하였다 [2,3].

하지만 이런 유클리드 공간 대응 기법은 여러 가지 문제점을 가지고 있다. 첫 번째로 네트워크 거리가 유클리드 공간에서 발생하지 않았다는 것이다. 그 예로 Triangle Inequality를 만족하지 않는 호스트들이 존재한다는 점이다. 또한 이런 추정 방식이 실제 시스템에서 사용될만한 정확도를 갖추지 못했다는 점이다.

이에 따라 다른 연구자 들은 유클리드 공간 모델

대신에 비 유클리드적인 모델을 사용하려는 시도를 해왔다 [4]. 하지만 [4]에서 제시된 방식은 매트릭스의 분할을 이용하는 것인데 별로 직관적이지 않다는 문제점이 있다.

이 논문에서도 비슷한 문제 의식에서 출발하여, 네트워크 거리의 비 유클리드적인 특성을 모델링할 수 있는 두 가지 모델을 제시하고, 성능 분석을 하였다. 기본적으로 가까운 네트워크 거리들이 삼각 부등식을 많이 만족하지 못하는데 이를 해결하기 위해 각 호스트 별로 가까운 거리를 좀 더 자세히 표현 할 수 있는 방법들을 제시하는 것이다. 그 예로 가상 토폴로지 기법에서는 호스트들을 클러스터로 분할 한 후 호스트 내부와 외부의 거리 추정을 달리 했고, 지역 조정 항 모델에서는 각 호스트의 추정 성능에 따라 추정 값을 보완하는 기법을 제시한다.

이 논문의 이후 전개는 다음과 같다. 2절에서 기본적인 배경 지식으로 유클리드 공간 좌표 기반 네트워크 거리 추정 기법들과 클러스터 이용 기법들에 대해 살펴본다. 3절에서는 새로 제안하는 2가지 비 유클리드 모델을 제시한다. 4절에서는 이 두 기법의 성능을 분석하고, 5절에서 논문의 내용을 정리한다.

## 2. 배경 및 관련 연구

인터넷의 거리 추정 기법은 여러 가지 방식으로 이루어진다. 이 절에서는 좌표 기반 거리 추정 기법에 대해 알아보고, 호스트들의 클러스터를 이용하는 방법들에 대해 알아본다.

### 2.1 좌표 기반 거리 추정 기법

좌표 기반 거리 추정 기법의 가장 기본적인 아이디어는 다음과 같다. 각 호스트들이 가상의 유클리드 공간에서 하나의 좌표를 할당 받는다. 좌표를 할당하는 방식은 두 호스트사이의 유클리드 거리가 실제의 네트워크 거리가 되도록 한다. 오차가 없이 정확하게 좌표를 할당하는 것은 불가능하

기 때문에 일반적으로 오차를 최소한으로 하는 것이 각 좌표 부여 방식의 목표가 된다.

좌표를 어떻게 부여할 것인가에 대한 연구가 많이 이루어져왔다. 우선 GNP (Global Network Positioning)이라는 기법은 랜드마크라고 불리는 특별한 호스트들은 기준점으로 사용한다 [1]. 이 랜드마크들 간에 서로 네트워크 거리를 측정한다. 다음 simplex downhill 기법을 사용하여 좌표를 각 랜드마크에 부여하는데 실제 거리와 추정된 거리 (유클리드 거리에 의해 정해짐) 간의 오차가 최소한이 될 때까지 계속 좌표를 수정해 나가는 것이다. 랜드마크의 좌표가 정해지면 일반 호스트들은 각 랜드마크와 자신간의 거리를 측정하고, 이 실제 거리를 바탕으로 simplex downhill 기법을 이용하여 자신의 좌표를 정한다. 일단 무작위로 자신의 좌표를 지정한 후 그 좌표와 랜드마크들의 좌표에 의해 계산된 추정 거리를 실제 거리와 비교하여, 오차가 줄어드는 방향으로 자신의 좌표를 계속 수정하여 local minimum에 도달하면 그 좌표를 자신의 좌표로 설정한다. 이 GNP 방법은 우선 랜드마크로의 거리 측정 패킷이 물리게 되어 랜드마크의 부하를 초래하게 된다. 또 다른 문제점으로는 simplex downhill 기법 같은 iterative 기법을 사용함으로써 해서 좌표 결정 시간이 매우 오래 걸린다는 점이다.

이에 대한 문제점을 해결하기 위해 Virtual Landmark 기법과 ICS 기법이 도입되었다 [2,3]. 이 방법들은 공히 Lipschitz Embedding이라는 방식을 사용하여 초기 좌표를 계산한 다음 Principal Component Analysis라는 기법을 사용하여 좌표의 차원을 줄이게 된다. Lipschitz Embedding에서는 차원이 랜드마크의 수와 동일 한데 PCA를 사용함으로써 해서 이 차원을 2-3 차원으로 줄일 수 있게 된다. 이 두 가지 방식에서는 simplex downhill과 같은 시간이 많이 걸리는 방식을 사용하는 대신에 형렬 곱셈에 의해 좌표를 계산하게 된다.

위에서 언급한 세가지 방식은 공히 랜드마크를 사용한다. 즉 미리 정해진 10-20개 정도의 랜드마크로의 거리를 측정하여, 그 거리를 바탕으로 자신의 좌표를 결정하는 것이다. 하지만 이 방식은 랜드마크의 부하를 초래하는 문제가 있는데 이 문제점을 해결하는 방식으로 Vivaldi 기법이 제안되었다 [5]. 이 기법에서는 각 호스트들이 무작위로 선택된 다른 호스트들과의 거리를 측정하여 자신의 좌표를 계속 수정해 나가는 것이다. 좌표를 수정하기 위해 spring relaxation 기법을 사용하였다. 한가지 문제점은 이 방식을 사용할 경우 좌표를 결정하는데 시간이 오래 걸리고, 시스템 전체적으로 안정화가 되는 데 시간이 오래 걸린다는 점이다.

## 2.2 클러스터 기반 기법

IDMaps 기법은 좌표를 기반으로 하지 않고, 클러스터를 이용한다 [6]. Tracer라 불리는 호스트들을 인터넷에 배치 시킨 다음, 각 tracer들이 가까운 곳에 존재하는 다른 호스트들로의 거리를 측정한다. 각 tracer에 가까운 호스트들이 한 클러스터를 이루게 된다. 즉 인터넷의 호스트들을 클러스터로 분할을 한다는 것이다. 각 tracer들은 또한 tracer들간의 거리를 측정한다. 이에 따라 멀리 떨어진 두 호스트 사이의 거리는 한 호스트로부터 가까운 tracer 까지의 거리와 상대 호스트에서 가까운 다른 tracer로의 거리와 이 두 tracer간의 거리의 합으로 추정된다. 이 방식의 문제점은 두 호스트가 매우 가까운 거리임에도 불구하고, 가까운 Tracer가 서로 달라서 추정 거리가 매우 커질 수 있다는 점이다. 이 문제점이 이 논문에서 제시하는 가상 토폴로지 기법의 주요 연구 동기 중 하나이다.

## 3. 비 유클리드 모델

앞에서 언급했다시피 이 논문에서는 유클리드 모델 대신에 비 유클리드 모델을 네트워크 거리 추

정 기법에 도입하고자 한다. 그 가장 기본적인 동기는 네트워크 거리가 유클리드 공간의 가정을 만족시키고 있지 못하고, 그로 인해 정확도가 떨어지기 때문이다. 다음에서는 두 가지의 새로운 모델을 소개한다.

### 3.1 가상 토폴로지 (Virtual Topology) 모델

가상 토폴로지를 구성하기 위해서 우리는 거리 행렬  $D = (d_{ij})$ 을 사용한다. 거리 행렬이란 각 호스트 들간의 거리를 행렬로 표현한 것으로  $d_{ij}$ 는 호스트  $i$ 로부터 호스트  $j$ 까지의 거리를 의미한다. 왕복 지연 시간을 거리로 사용하는 경우에는 이 거리 행렬은 symmetric matrix가 된다. 이 절에서의 설명은 우리가 이 거리 행렬을 알고 있다고 가정한다. 하지만 실제적으로는 이 거리 행렬을 미리 알지 못함으로 표본 추출 방식에 의해 이 거리 행렬의 일부 원소만을 알고 있다고 해도 비슷한 방식으로 가상 토폴로지를 구성할 수 있음을 이후에 보이도록 한다.

우리는 호스트들이 클러스터로 분류되었다고 가정한다. 이 분류는 AS 번호를 이용할 수도 있고, 지역적인 정보를 이용할 수도 있다. 각 클러스터는 가상의 tracer를 가지고 있고 가정한다. 우리는 클러스터 내부의 각 호스트 들로부터 이 가상의 tracer까지의 거리를 추정해 내야 한다. 우선  $c(i)$ 를 호스트  $i$ 의 클러스터라고 하자.  $v(c)$ 를 클러스터  $c$ 의 가상 tracer라고 하자.  $x_i$ 를 호스트  $i$ 로부터 가상 tracer  $v(c(i))$ 까지의 거리라고 하자. 그리고  $l_{v(c)v(c')}$ 을 클러스터  $c$ 와  $c'$ 의 가상 tracer들간의 거리라고 하자. 이에 따라 호스트  $i$ 와 호스트  $j$ 간의 거리는  $x_i + l_{v(c(i))v(c(j))} + x_j$ 로 추정된다.

여기에서 가장 큰 문제는 어떻게  $x_i$ 들과  $l_{v(c)v(c')}$ 을 측정하는가 하는 점이다. 우리는 이 과정을 두 단계로 구분한다. 첫번째 단계에서는 우선  $x_i$ 를 구하고, 그 다음에  $l_{v(c)v(c')}$ 을 구하게 된다. 우선 클러스터 내부의 각 호스트 사이에서는 다음과 같은

등식을 만족한다.

$$x_i + x_j = d_{ij}, \text{ for } i < j \quad (1)$$

이를 행렬로 표현하면 다음과 같다.

$$Ax = d \quad (2)$$

여기에서  $A$ 의 각 행은 호스트  $i$ 와  $j$ 의 짝을 의미하는데  $i$ 번째 열과  $j$ 번째 열만 1이고, 나머지 열은 모두 0이다.  $d$ 의 각 행은 호스트  $i$ 와  $j$ 간의 실제 거리이다. 등식 (2)는 실제로 변수에 비해 행 수가 많기 때문에 해답이 존재하지 않을 가능성이 많은데 이를 위해 linear least square 기법을 이용하여 오차가 최소가 되는 근사 해를 구하게 된다. 이는 다음과 같이 표현할 수 있다.

$$A^T A \hat{x} = A^T d \quad (3)$$

$$\hat{x} = (A^T A)^{-1} A^T d \quad (4)$$

이 방식을 통해  $x_i$ 들을 구한 다음,  $l_{v(c)v(c')}$ 도 비슷한 방식으로 구할 수 있다. 다행히도, 그 해답은 다음과 같이 표현된다.

$$l_{v(c)v(c')} = \frac{\sum_{i \in c} \sum_{j \in c'} (d_{ij} - x_i - x_j)}{|c||c'|} \quad (5)$$

여기에서  $|c|$ 는 클러스터  $c$ 에 존재하는 호스트들의 수이다.

### 3.2 지역 조정항 (Local Adjustment Term) 모델

좌표 기반 기법의 문제점을 해결하는 방법으로 여기에서는 지역 조정 항 (Local Adjustment Term, LAT) 모델을 제안한다. LAT기법의 기본적인 아이디어는 다음과 같다. 우선 좌표 기반 기법을 이용하여 호스트들은 유클리드 공간에 대응시킨다. 그 다음에 각 호스트들은 자신의 좌표에 의해 추정되는 거리의 정확도에 따라 유클리드 거리를 조정하는 값을 결정해서 유클리드 거리를 수정하게 된다. 어떻게 이 조정 항을 결정하는 지가 가장 중요한 문제가 된다. 일단 이 조정 항이 결정되었다면 다음과 같은 성질을 가지게 된다. 각 호스트는  $(x_1, x_2, x_3, \dots, x_d; e_x)$ 의 좌표를 가지게 되는데

$e_x$  가 지역 조정 항이다. 이러한 좌표를 가지고 두 호스트 간의 거리는  $\hat{d}_{xy} = d_{xy}^E + e_x + e_y$  이다.

여기서  $d_{xy}^E = \sqrt{\sum_{k=1}^d (x_k - y_k)^2}$  는 호스트  $x$ 와  $y$  사이의 유클리드 거리이다.  $e_x$ 는 Vivaldi에서의 height vector와 비슷하게 보이지만 실제적으로는 매우 다른데, 그 차이점은 조금 후에 다시 설명한다. 어쨌든 위와 같은 설정에서 한가지 해결해야 할 점은  $x$ 와  $y$ 의 지역 조정 항인  $e_x$ 와  $e_y$ 를 어떻게 구하는가 하는 점이다. 이상적으로는  $e_x$ 가 호스트들 간의 거리에서 비 유클리드적인 성질을 표현할 수 있으면 좋는데 그렇게 하기 위해서는 많은 양의 거리 측정이 필요함으로 우리는 다음과 같은 방식을 이용하여 쉽게 구하도록 한다. 우선  $\epsilon_x$ 를 구한다. 이  $\epsilon_x$ 는 다음과 같은 식을 최소화 하는 파라미터이다.

$$E(x) = \sum_y (d_{xy} - (d_{xy}^E + \epsilon_x))^2 \quad (6)$$

여기서  $d_{xy}$ 는 호스트  $x$ 와  $y$ 의 실제 거리이다.  $\epsilon_x$ 는 다음과 같이 쉽게 구해진다.

$$\epsilon_x = \frac{\sum_y (d_{xy} - d_{xy}^E)^2}{n} \quad (7)$$

여기서  $n$ 은 다른 호스트의 수이다.

지역 조정항  $e_x$ 는  $\epsilon_x/2$ 로 설정하면 된다. 우리는 아래와 같은 정리를 통해 이 지역 조정항을 이용하는 기법이 항상 좌표 기반 기법에 비해 더 나은 stretch 값을 가짐을 보여준다.

정리 1. The hybrid model using a  $d$ -dim Euclidean space and the adjustment term defined above reduces the squared stress of a pure  $d$ -dim Euclidean embedding by

$$\frac{4n \sum_x e_x^2 + 2n^2 \text{Var}(e_x)}{\sum_{x,y} d_{xy}^2} \geq 0, \quad (8)$$

$$\text{where } \text{Var}(e_x) = \sum_x e_x^2 / n - \left( \frac{\sum_x e_x}{n} \right)^2.$$

정확도에 대한 기준으로 이 후 성능 분석에서 스트레스 (stress)와 상대 오차 (relative error)를 사용하는데, 이 각각의 정의에 대해서는 다음 절에서 다룬다. 다만 위 정리는 정확도를 stress로 비교 할 경우 정확도가 증가함을 보여주고 있다. 특히 각 조정항의 크기가 클수록, 이 지역 조정항 기법을 이용할 경우 정확도가 더 높아진다.

지금까지의 설명은 모든 호스트 들간의 거리를 이미 알고 있다는 가정하에서였다. 하지만 실제에서는 이 방식을 있는 그대로 사용할 수가 없다. 대신에 우리는 표본 추출에 의한 방식을 사용한다. 즉 한 호스트에서 다른 모든 호스트로의 거리를 측정하는 대신에 무작위로 추출된 작은 수의 호스트들에 대해서만 실제 거리를 측정하고, 이를 바탕으로 지역 조정항을 계산하도록 한다.  $S$ 를 무작위로 추출된 호스트들이라고 하자.

$$\tilde{e}_x = \frac{\sum_{y \in S} (d_{xy} - d_{xy}^E)^2}{|S|} \quad (9)$$

따라서 실제에서는 이 지역 조정항 기법은 다음과 같이 동작한다 : a) 각 랜드마크를 통해  $d$ -차원의 유클리드 좌표를 계산한다. b) 무작위로 선택된 호스트 들에 대해 각 호스트에서 거리를 측정한 후 표본 추출된 지역 조정항을 계산한다.

이 조정항은 Vivaldi의 height vector와 비슷한 모양을 띠고 있다. 하지만 몇가지 주요한 차이점이 존재한다. 우선, 지역 조정항을 계산하는 방식이 매우 단순하고 다른 호스트의 지역 조정항에 의존하지 않는다. 하지만 Vivaldi에서는 그 분산

계산 과정에 의해 한 height vector에서의 조그만 변화가 다른 많은 height vector들을 변화하게 한다. 따라서 시스템이 전체적으로 안정화가 되는데 시간이 걸린다. 두번째로, 위의 정리 1에서 보였 다시피 지역 조정항은 정확성을 증가 시킨다는 점 이 수학적으로 검증되었다. 또다른 장점으로 이 지역 조정항은 다른 어떠한 거리 추정 기법들과도 연동하여 사용할 수 있다는 점이다. 즉 기반이 되 는 거리 추정 방식이 유클리드 공간 좌표 방식이 아니더라도 이 지역 조정항은 사용할 수 있다. 이 러한 점에서 이 지역 조정항은 다른 기법을 대체 하는 새로운 기법이라기 보다는 다른 기법들을 보 완하는 선택 사항으로 사용될 수 있다. 심지어는 Vivaldi에서조차도 이 지역 조정항을 사용할 수 있다.

#### 4. 성능 분석

이 절에서는 앞에서 제시한 두 가지 방식의 거리 추정 기법의 성능을 분석한다. 첫째로 중앙 집중 방식 가상 토폴로지 (CVT : Centralized Virtual Topology) 의 성능을 GNP, 가상 랜드마크, IDMaps 방식과 비교한다.

성능 비교를 위한 기준으로 stress와 상대 오차 (relative error)를 사용한다. Stress는 다음 식으 로 표현된다.

$$Stress = \sqrt{\frac{\sum_{xy} (d_{xy} - \hat{d}_{xy})^2}{\sum_{xy} d_{xy}^2}} \quad (10)$$

여기서  $d_{xy}$  는 호스트 x와 y 사이의 실제 네트워

크 거리이고,  $\hat{d}_{xy}$  는 추정 거리이다.

상대 오차는 다음과 같이 정의된다.

$$\frac{|d_{xy} - \hat{d}_{xy}|}{\min(d_{xy}, \hat{d}_{xy})} \quad (11)$$

성능 분석에서는 여러 가지 거리 추정 기법들을 실제 데이터 집합들에 적용하여 나온 정확도의 stress와 상대 오차를 비교하게 된다.

우선 사용한 데이터 집합을 설명한다. 그림 1의 예에서 보듯이 4개의 데이터 집합을 주로 사용 한다. 데이터 집합 "PL"은 Planetlab 의 202 호 스트들 간의 측정 거리를 바탕으로 만든 202 \* 202 네트워크 거리 행렬이다 [7]. 데이터 집합 "King"은 2000 여개의 DNS 서버들 사이에서 King method를 이용하여 측정한 거리 중 오류가 있는 부분을 제외하고 최대한의 square matrix를 얻어 냈는데 그 크기는 462 \* 462 이다 [8]. 즉 462 호스트 간의 거리 행렬이라고 볼 수 있다. "NLNR"은 83개의 AMP project에 존재하는 호 스트 들간의 거리이다 [9]. 마지막으로 "18CL"은 2-level 토폴로지 (Synthetic topology)에서 계산 된 거리를 바탕으로 형성된 데이터 집합이다.

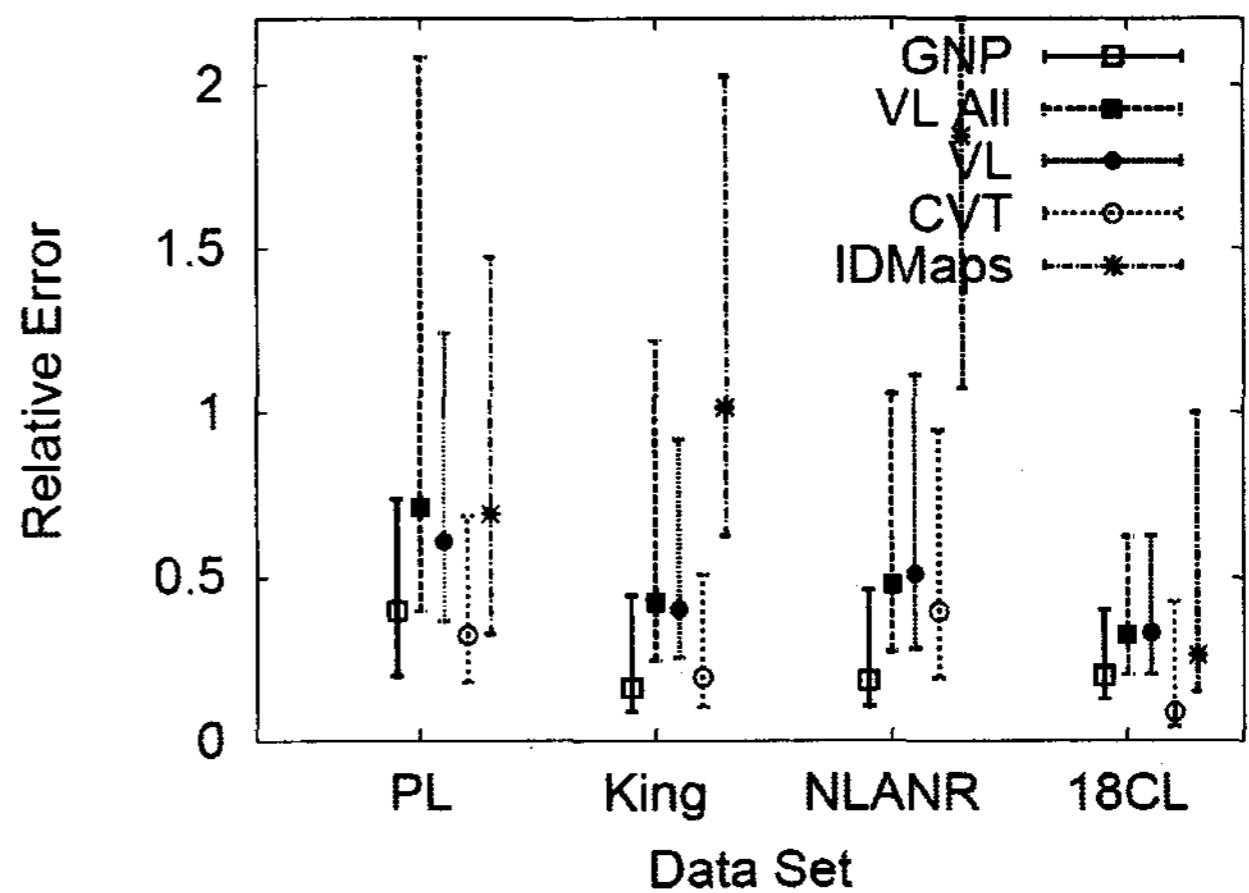


그림 1. 18-20개의 랜드마크를 사용하였다. 각 데이터 집합에 대해 상대 오차의 50, 70, 90 % 값을 보여준다.

그림 1 은 이 데이터 집합들에 대해 위에서 언급 한 방식들을 적용하여 나온 결과에서 상대적 오차 (relative error)를 측정하고, 50 %, 70 %, 90%의 값들을 보여주고 있다. 일반적으로 GNP가 가장 작은 상대 오차를 보여주고 있다. 즉 GNP가 매우

좋은 추정 정확도를 가지고 있다는 점을 알 수 있다. 하지만 CVT의 경우는 IDMaps 기법보다는 훨씬 좋은 성능을 보여주고 있고, GNP와도 비슷한 성능을 보여주고 있다. 특히 CVT는 18CL 데이터 집합에서 가장 좋은 성능을 보여주고 있는데, 이는 호스트 들이 강한 클러스터를 이루고 있을 때 매우 효과적임을 알 수 있다. 실제로 우리가 얻을 수 있는 데이터 집합은 매우 강한 클러스터를 보여주지 못한다. 그 이유는 대체로 DNS 서버나 Planetlab 호스트 들이 전체 인터넷 상에서 골고루 흩어져 있기 때문이다. 만약 실제 사용자들의 호스트를 무작위로 선택하고, 그 수가 많아진다면 클러스터 현상이 매우 강하게 나타나게 될 것이다. 이 경우 Virtual topology는 매우 훌륭한 성능을 보일 것으로 예상된다.

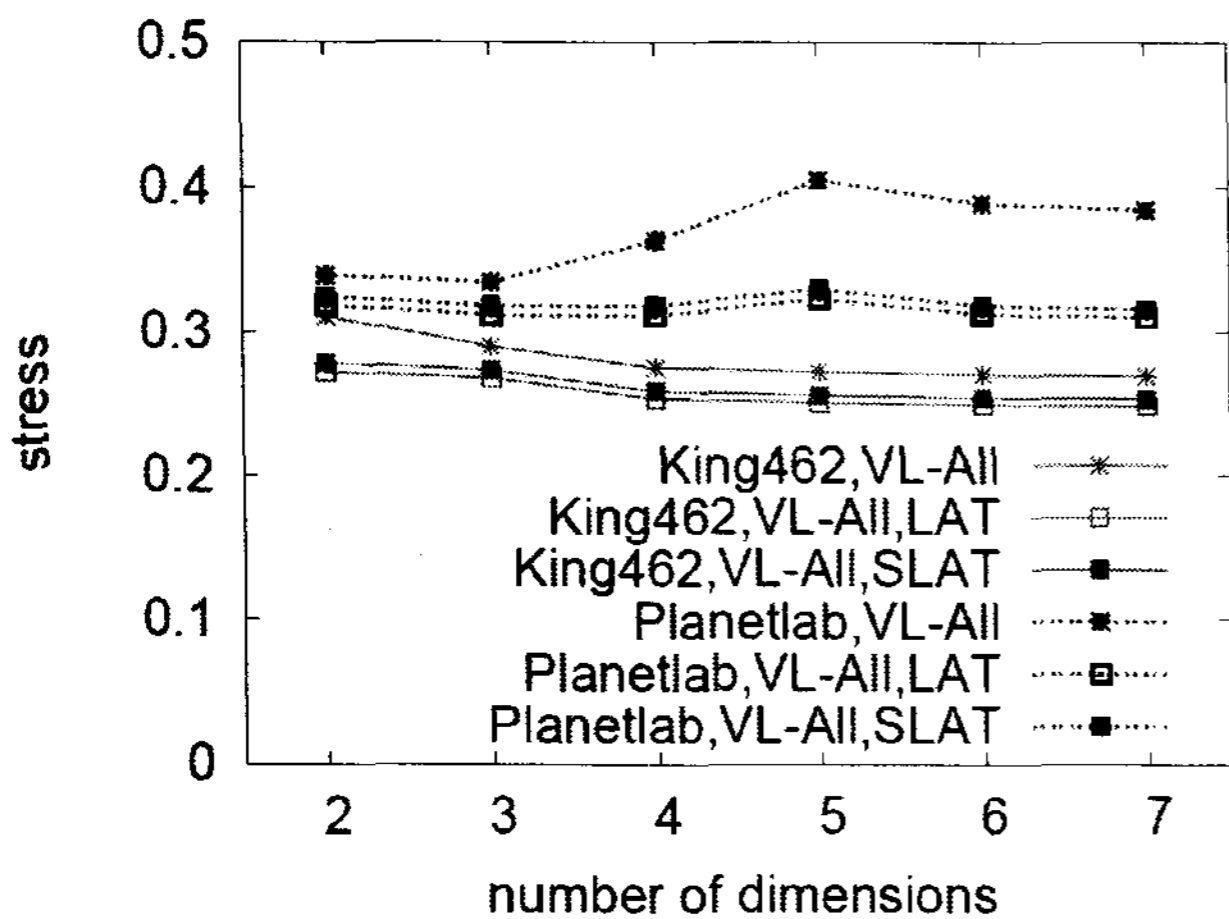


그림 2. 각 차원 별 가상 랜드마크 기법의 stress. LAT 기법과 SLAT 기법, 그리고 아무 기법도 안 사용했을 경우 등 세 종류의 실험 결과를 보여준다.

다음은 지역 조정 항 (LAT) 기법에 대한 성능을 분석해 보도록 한다. 이를 위해 가상 랜드마크 기법에서 LAT를 사용하지 않는 방식과 사용하는 방식의 차이를 보도록 한다. 여기에서는 LAT는 모든 노드에 대한 거리 측정을 이용하여 계산한다. 우리는 차원을 2에서 7까지 변화시킨다. 그림 2

에서는 그 실험 결과를 보여준다. 사용한 데이터 집합은 King462 와 Planetlab이다. 각 데이터 집합에서 “VL-All”, “VL-All, LAT”, “VL-All,SLAT”의 세가지 다른 기법을 이용하였다. “VL-All”은 가상 랜드마크 기법 자체만을 사용한 것을 의미하며, “VL-All, LAT”는 가상 랜드마크 기법에 LAT 옵션을 사용한 것이고, “VL-All,SLAT”은 가상 랜드마크 기법에 표본 추출 지역 조정 항 (Sampled Local Adjustment Term)을 사용한 것이다. 그림 2에서 보듯이 지역 조정 항을 사용하면 stress 값을 상당히 줄일 수 있음을 볼 수 있다. 특히 원래의 유클리드 공간 좌표 기법의 stress가 매우 클 경우, LAT의 사용은 더 많은 stress의 감소를 가져오게 된다. 또한 유클리드 공간 좌표 기법에서 차원을 증가시키는 것은 별로 도움이 되지 않음을 알 수 있다. 낮은 차원의 유클리드 공간 좌표와 LAT 옵션을 사용하는 것이 더 나은 성능 향상의 방법이라고 할 수 있다.

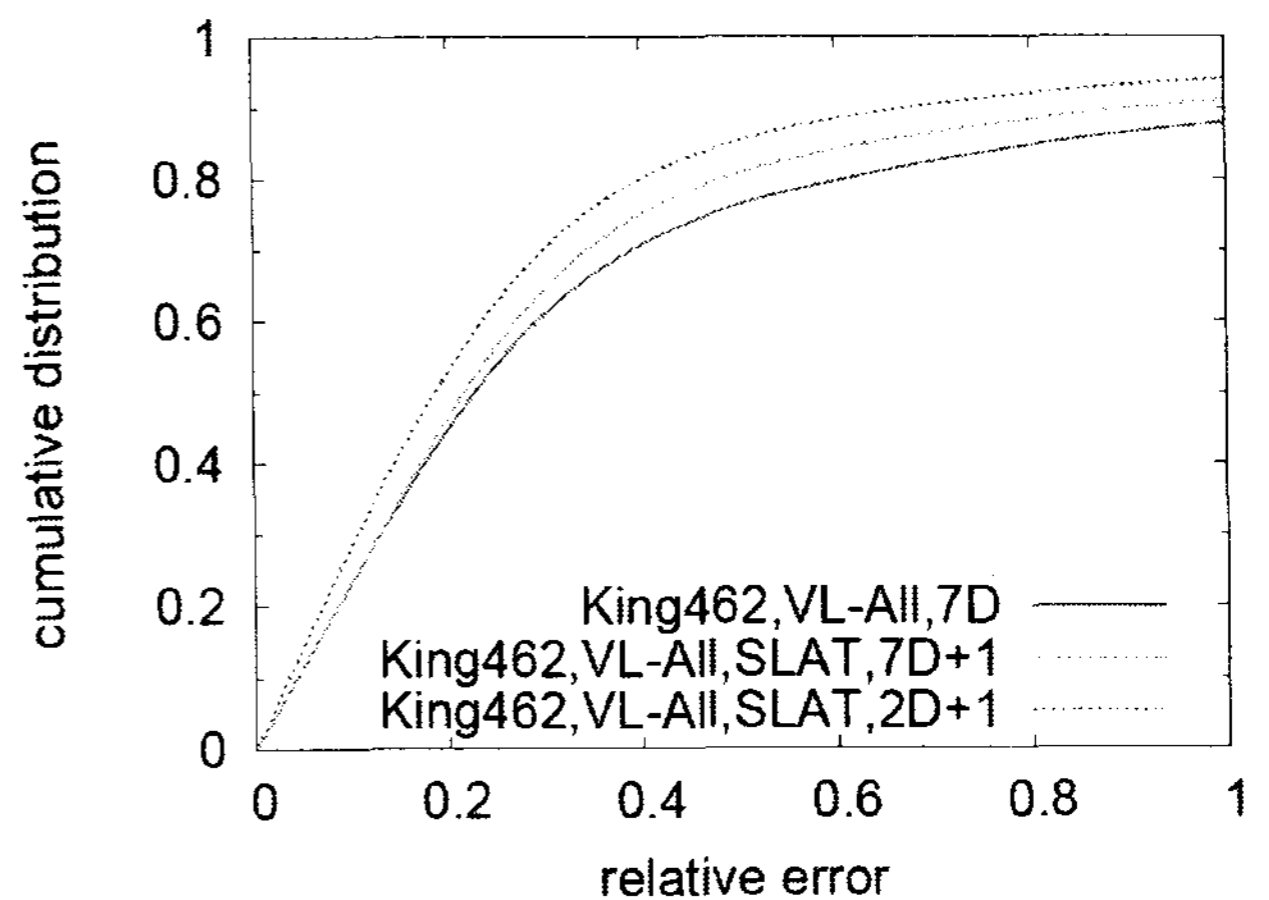


그림 3. 462 호스트 킹 데이터 집합에 적용한 여러 추정 기법들의 상대 오차의 누적 분포

LAT 옵션이 stress도 줄이지만 상대 오차도 줄인다. 그림 3에서는 462노드의 King 데이터 집합을 이용하여 상대 오차의 누적 분포를 계산하였다. 기본 유클리드 공간 좌표 기법은 가상 랜드마크 기법이고, SLAT 옵션을 추가한 경우와 추가하지

않은 경우가 있는데 옵션을 추가 한 경우 가상 랜드마크 기법에서 사용된 차원 수를 2와 7 두가지로 사용하였다. 그림 3에서 볼 수 있다시피, SLAT 옵션을 2차원 가상 랜드마크 기법에서 사용한 결과가 7차원의 가상 랜드마크만을 사용한 결과에 비해 더 작은 상대 오차를 보여준다.

## 5. 결론

이 논문에서는 기존에 많이 연구되어온 유클리드 공간 좌표 기반 네트워크 거리 추정의 정확도를 개선할 수 있는 두 가지 기법들을 소개했다. 실제 인터넷에서 측정된 데이터를 바탕으로 CVT나 LAT를 적용했을 경우, 상당한 정확도의 향상을 가져옴을 보여준다. 그 원인으로는 인터넷 상의 네트워크 거리가 가지는 비 유클리드적인 성질을 표현할 수 있는 모델링을 하였기 때문으로 보여지는데 그에 대한 좀더 심도 깊은 분석이 이후 과제로 연구할 계획이다.

## [참고문헌]

- [1] T.S. Eugene Ng and Hui Zhang, "Predicting internet network distance with coordinates-based approaches," in *Proc. IEEE INFOCOM*, New York, NY, June 2002.
- [2] Liying Tang and Mark Crovella, "Virtual landmarks for the internet," in *Proceedings of the Internet Measurement Conference(IMC)*, Miami, Florida, Oct. 2003.
- [3] Hyuk Lim, Jennifer C. Hou, and Chong-Ho Choi, "Constructing internet coordinate system based on delay measurement," in *Proceedings of the Internet Measurement Conference(IMC)*, Miami, Florida, Oct. 2003.
- [4] Yun Mao and Lawrence K. Saul, "Modeling distances in large-scale networks by matrix factorization," in *Proceedings of the Internet Measurement Conference(IMC)*, Taormina, Sicily,

Italy, Oct. 2004.

- [5] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris, "Vivaldi: A decentralized network coordinate system," in *Proceedings of ACM SIGCOMM 2004*, Portland, OR, Aug. 2004.
- [6] Paul Francis, Sugih Jamin, Cheng Jin, Yizin Jin, Danny Raz, Yuval Shavitt, and Lixia Zhang, "Idmaps: A global internet host distance estimation service," *IEEE/ACM Transactions on Networking*, 2001.
- [7] Jeremy Stribling, "Rtt among planetlab nodes," [http://www.pdos.lcs.mit.edu/~strib/pl\\_app/](http://www.pdos.lcs.mit.edu/~strib/pl_app/).
- [8] "Vivaldi data set," <http://pdos.lcs.mit.edu/p2psim/kingdata>.
- [9] "Nlanr amp data set," <http://amp.nlanr.net/Status/>.