트래픽 부하 측정을 위한 적응성 있는 랜덤 패킷 샘플링 기법

박 재 성*, 최 백 영**, Zhi-Li Zhang**

Adaptive Random Packet Sampling for Traffic Load Measurement

Jaesung Park*, Back-Young** Choi, Zhi-Li Zhang**

요 약

트래픽 부하 측정은 네트웍 트래픽 엔지니어링의 기반이 된다. 그러나 고속 링크에서 트래픽 부하 정보를 얻기 위해 모든 패킷을 측정하는 것은, 라우터의 패킷 포워딩 성능을 저해시키므로 확장성이 결여된다. 이에 따라 샘플링 기법이 트래픽 측정의 대안으로 제시 되었다. 샘플링은 라우터의 성능 저해를 최소화 시킬 수 있으나 샘플링으로 예측되는 트래픽 부하는 실제 트래픽 부하와 차이를 보이게 되며, 이와 같은 오류가 제한되지 못한다면 측정 값을 기반으로 하는 응용들에 부영향을 미치게 된다. 본 논문에서는 샘플링 오류를 오류 허용 범위내로 제한 시킬 수 있는 적응성 있는 패킷 샘플링 기법을 제안한다. 제안 기법은 수학적 분석을 통해 얻어진 부하 예측 오류에 영향을 미치는 주요 트래픽 파라메터를 각 블록의 시작마다 예측하여 샘플링 확률을 동적으로 적용 시킨다. 본 논문에서는 또한 실제 측정된 인터넷 트래픽을 이용하여 제안 기법의 확장성과 성능을 검증하였다

Key Words : 트래픽 측정, 샘플링, 오류 제한, 트래픽 엔지니어링

ABSTRACT

Exactly measuring traffic load is the basis for efficient traffic engineering. However, precise traffic measurement involves inspecting every packet traversing a link, resulting in significant overhead on routers with high-speed links. Sampling techniques are proposed as an alternative way to reduce the measurement overhead. But, since sampling inevitably accompany with error, there should be a way to control, or at least limit, the error for traffic engineering applications to work correctly. In this paper, we address the problem of bounding sampling error within a pre-specified tolerance level. We derive a relationship between the number of samples, the accuracy of estimation and the squared coefficient of variation of packet size distribution. Based on this relationship, we propose an adaptive random sampling technique that determines the minimum sampling probability adaptively according to traffic dynamics. Using real network traffic traces, we show that the proposed adaptive random sampling technique indeed produces the desired accuracy, while also yielding significant reduction in the amount of traffic samples.

Key Words : Traffic measurement, Sampling, Bounded Error, Traffic Engineering

* LG전자 UMTS 연구소 무선 HW Gr.(4better@lge.com), ** Department of Computer Science, University of Minnesota at Twin City (choiby@cs.umn.edu)

논문번호 : 020467-1023, 접수일자 : 2002년 10월 23일

Ⅰ. 서 론

네트워 트래픽 부하를 정확하게 측정하는 것은 네트워의 구성, 관리, 폴리싱, 과금 등 네트워 전반에 걸친 문제를 발견하고, 해결하기 위해 매우 중요하다. 현재 링크 상의 트래픽을 측정하기 위한 방법은 해당 링크 상의 모든 패킷을 수집하는 방식을 취하고 있다.[15][16] 그러나 링크가 고속화 되고 있는 상황에서 이와 같은 방법은 라우터의 패킷 처리 성능을 감소시키거나, 측정을 위한 특별한 장치의 사용을 필요로 하며, 또한 측정 트래픽 저장을 위해 매우 큰 저장 공간을 요구한다. 더욱이, 측정된 데이터의 양이 많아 질수록 이를 처리하기 위한 시간이 길어지게 되며, 이로 인해 네트워의 문제를 검출하여 이를 해결하기 위한 적절한 방법을 결정하기 전에 네트워 상황이 회복 불능의 상태가 되는 경우가 발생할 수 있다. 따라서 효과적인 트래픽 부하 측정을 위해서는 패킷을 샘플링하여 측정하는 것이 보다 적절한 대안이 된다. 그러나 샘플링은 필수불가결하게 오류를 수반하게 되며, 이와 같은 오류는 측정 데이터를 기반으로 하는 모든 응용에 부정적인 영향을 미칠 수가 있으므로, 샘플링으로 인한 오류를 정량화하고 제어하는 것이 필요하다.

네트워 트래픽 측정과 관련하여 샘플링에 대한 연구는 다양한 응용에 대하여 광범위하게 이루어져 왔다. 네트워 트래픽의 통계적 샘플링에 대한 연구는 1990년대 초반에 NSFNET 백본에서 처음으로 시작되었다[1]. Claffy et. al.은 패킷 크기의 분포, 패킷 도착 시간 간격에 대한 통계값을 얻기 위해 전통적인 event-driven, time-driven 정적(static) 샘플링 방법을 NSFNET 백본 트래픽에 적용하여 그 성능을 분석하였다[1]. [2]에서 제안된 Trajectory 샘플링은 네트워 도메인을 통하는 모든 트래픽을 직접적으로 관측하고 네트워 트래픽의 공간적 관계에 대한 통계값을 추론하였다. 플로우의 크기에 따른 플로우 샘플링은 플로우의 볼륨에 따른 차등적인 과금을 위해 사용되었다 [3]. [4]에서는 확률적 패킷 샘플링 방법을 사용하여 elephant 플로우를 식별하고, 각 패킷의 헤더를 검사하여 패킷 크기에 따라 샘플링 확률을 결정하였다. 그러나 이와 같은 방법들은 패킷 샘플링시에 발생하는 오류를 특정값 이하로 제한 하거나 이를 제어하기 위한 방법에 대해서는 다루지 않고 있다.

본 논문에서는 트래픽 부하 측정을 위한 적응성 있는 랜덤 패킷 샘플링 기법을 제안한다. 본 논문에서 제안하는 기법은 패킷 샘플링을 위해 패킷 내용을 검사하지 않으며, 패킷 샘플링을 통한 트래픽 부하 예측시 발생하는 오류를 수학적으로 분석하고 정량화 하고, 또한 사용자가 지정한 오류 허용 수준 내에 트래픽 부하 예측 오류를 제한할 수 있다는 점에서 이전 연구들과 차별화 된다.

본 논문은 다음과 같이 구성된다. 우선 2장에서는 본 논문에서 다루고자 하는 문제를 수학적으로 정의하고, 정적 샘플링의 문제점을 보임으로써 적응성 있는 샘플링의 필요성을 제시한다. 3장에서는 적응성 있는 패킷 샘플링 방법을 제안하고 수학적으로 제안 기법의 성능을 분석한 후, 4장에서는 실제 측정 트래픽을 이용한 결과를 제시하여 제안 기법의 타당성을 검증하고, 5장에서 결론을 맺는다. 본 논문의 실험에 사용된 데이터들은 NLANR[5]과 Auckland-II[6]에서 얻었으며 다양한 링크 속도와 입력율, 측정 시간을 가지는 데이터를 선별하여 사용하였다. 실험에 사용된 데이터의 특성은 표 A.1로 정리하였다.

Ⅱ. 패킷 샘플링을 통한 트래픽 부하 예측 문제

본 장에서는 우선 트래픽 부하 측정을 위한 샘플링 기법의 문제를 정의 한 후 주어진 오류 허용 수준 내에서 트래픽 부하를 정확하게 예측하기 위해 필요한 최소한의 샘플 수에 대한 하한값을 유도하고, 이를 기반으로 최적 샘플링 확률을 결정한다. 본 논문에서 최적 샘플링 확률은 최소한의 패킷 샘플로 원하는 정확도를 보장하는 샘플링 확률을 의미한다. 특히 본 장에서는, 최적 샘플링 확률은 트래픽 관측 구간 동안 입력 패킷 수와 이들의 크기의 변이에 의한 함수라는 것을 보이며, 네트워 트래픽은 시간에 따라 패킷 수와 패킷 크기가 변화하기 때문에 고정 샘플링 확률을 사용하는 정적 패킷 샘플링 방법은 패킷의 과다 측정으로 인한 샘플링 효율의 감소와 필요 이하의 패킷 측정으로 인한 오류 발생을 가져 온다는 것을 보인다.

1. 패킷 샘플링에 의한 트래픽 부하 예측 오류

인터넷 링크에서 패킷의 도착 시간 간격은 불규칙 적이므로, 패킷 샘플간의 간격 역시 불규칙적이 된다. 그러나 트래픽 부하의 특성 변화점 검출과 같이, 측정된 트래픽 부하 값을 이용하는 대부분의 응용들은 시계열 분석 방법을 사용하며, 시계열 분석

을 위해서는 각 관측 값들 사이의 간격은 동일해야 한다. 이를 위해 트래픽 부하는 고정 길이의 관측 구간별로 측정되어 진다. 본 논문에서 관측 구간은 타임 블록, 혹은 블록이라 부르기로 하며 B로 나타내기로 한다. 타임 블록은 특정 엔지니어링 목적에 맞게 임의로 구성 가능하다.

한 블록에 입력된 패킷의 수를 m이라 하고, X_i를 i번째 입력 패킷의 크기라고 하면 이 블록에서의 트래픽 부하는 $V=\sum_{i}^{m}X_i$가 된다. 즉, 트래픽 부하는 특정 시간 동안 입력되는 패킷 수와 그들의 크기에 따라 결정된다. 이제 블록의 트래픽 부하를 예측하기 위해 입력 패킷 중 n개 $(1\le n\le m)$를 랜덤하게 샘플링 했다고 가정해보자. 즉, 각 패킷은 $p=n/m$의 동일 확률로 샘플링 된다. X_j를 $(j=1,\ldots,n)$ j번째 샘플의 패킷 크기라고 하면, 블록에서의 실제 트래픽 부하는 다음과 같이 예측된다 ($\hat{V}$).

$$\hat{V}=\frac{m}{n}\sum_{j=1}^{n}X_j \tag{1}$$

$E[\hat{V}]-V=0$이므로, 예측된 트래픽 부하 $\hat{V}$는 실제 트래픽 부하 V의 unbiased estimator가 된다. 패킷 샘플링을 통한 트래픽 부하 예측에서 발생하는 오류를 $|\hat{V}-V|/V$로 정의하면, 오류를 지정된 허용 범위 내로 제한하기 위한 문제는 다음과 같이 정의 할 수 있다.

$$\Pr\{|\frac{\hat{V}-V}{V}|>\varepsilon\}\le\eta \tag{2}$$

여기서 (ε,η)는 오류 허용 범위를 나타낸다. 예를 들어 $(\varepsilon,\eta)=(0.2,0.1)$은 부하 예측 오류가 0.2 이하가 되도록 확률 90%로 보장한다는 것을 의미한다. 이와 같은 문제 정의에 대해 이제 문제는 샘플링 오류를 주어진 오류 허용 범위 내로 보장하는 최소한의 샘플링 확률을 결정하는 것이다.

2. 최적 샘플링 확률

랜덤 샘플의 중심 극한 이론에 (central limit theorem)[7] 따라 샘플 크기가 무한대 $(n\to\infty)$가 되면 샘플 데이터의 평균은 모 집단의 분포 함수와 관계없이 모 집단의 평균에 접근하게 된다. 따라서 식 (2)는 다음과 같이 다시 정리할 수 있다.

$$\Pr\{|\frac{\hat{V}-V}{V}|>\varepsilon\}=$$

$$\Pr\{\frac{\sqrt{n}}{\sigma}|\frac{1}{n}\sum_{i=1}^{n}X_i-\mu|>\frac{\varepsilon\mu\sqrt{n}}{\sigma}\}$$

$$\approx 2(1-\Phi(\frac{\varepsilon\mu\sqrt{n}}{\sigma}))<\eta \tag{3}$$

여기서 μ와 σ는 각각 블록내 입력 패킷들의 크기의 평균과 표준편차를 나타내며 $\Phi(\cdot)$는 표준 정규 분포의 누적 확률 분포(c.d.f), 즉 $N(0,1)$를 나타낸다. 따라서 주어진 오류 허용 범위를 만족하기 위해 필요한 최소한의 샘플 패킷 수는 다음과 같이 주어진다.

$$n\ge n^*=(\frac{\Phi^{-1}(1-\eta/2)}{\varepsilon}\cdot\frac{\sigma}{\mu})^2=z_p\cdot S \tag{4}$$

여기서 $z_p=(\frac{\Phi^{-1}(1-\eta/2)}{\varepsilon})^2$는 오류 허용 범위 (ε,η)로 정의되는 상수이며 $S=(\frac{\sigma}{\mu})^2$는 블록내의 패킷 크기 분포의 SCV (squared coefficient of variance)를 나타낸다. 식 (4)에서 보는 바와 같이 샘플링 오류는 샘플의 수에 의해 조절이 가능하며, 특히 주어진 오류 허용 범위를 만족하기 위해 필요한 최소한의 샘플 수 (n^*)는 패킷 크기 분포의 SCV와 정비례 관계에 있음을 볼 수 있다. 이제 식 (4)로부터 최적 샘플링 확률, p^*는 다음과 같이 주어진다.

$$p^*=\frac{n^*}{m} \tag{5}$$

즉, 주어진 샘플링 오류 허용 범위 (ε,η)를 만족시키기 위해서는, 블록내에 입력되는 패킷들은 최소 p^*의 확률로 랜덤하게 샘플링되어야 한다.

3. 정적 샘플링의 문제점

정적 샘플링 기법은 고정된 샘플링 확률로 입력 패킷을 샘플링 하는 방법으로서, 가장 간단한 형태는 "one-out-of-N" 방식으로 N개의 입력 패킷 마다 한 개의 샘플을 취하는 것이며 Cisco Sampled NetFlow[8]등에 사용되어 지고 있다. 그러나 이와 같은 방법은 트래픽 특성이 시간에 따라 동적으로 변화한다는 것을 고려하고 있지 않으므로 트래픽

부하 예측에 사용되는 경우, 매 블록마다 샘플링에 의해 예측된 트래픽 부하가 주어진 오류 범위 내로

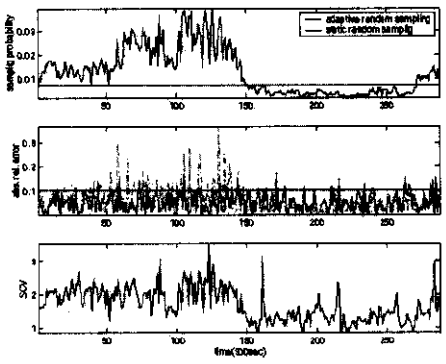


그림 4 정적 샘플링의 문제점

제한된다는 것을 보장하지 못한다. 더욱이 이와 같은 방법에서는 모든 블록에서 사용될 고정 샘플링 확률을 어떻게 결정할 것인가가 명확하지 않다.

정적 샘플링의 문제점을 통해 패킷 크기의 변이가 샘플링에 미치는 영향의 중요성을 보이기 위해 그림 1에서는 Auckland 데이터 Π_1을 이용하여 최적 랜덤 샘플링(adaptive random sampling)과 정적 샘플링(static random sampling)의 성능을 비교하였다. 최적 랜덤 샘플링에서는 각 블록 별로 최적 샘플링 확률을 미리 계산하여 샘플링을 수행하였으며, 공정한 비교를 위해 정적 샘플링 확률은 전체 패킷 데이터에 대한 샘플의 개수가 최적 랜덤 샘플링과 같아지도록 정하였고 블록 크기는 300초로, 오류 허용 범위는 $(\varepsilon, \eta) = (0.1, 0.1)$로 정하였다. 그림 1의 x축은 블록 크기 단위의 시간을 나타내며, 맨 위 그림은 시간에 따라 최적 랜덤 샘플링 확률과 정적 랜덤 샘플링 확률의 변화를 보여 준다. 그림 1의 두 번째 그림은 각 블록별로 샘플링 오류를 나타내며 그림에서 사선은 $\varepsilon = 0.1$을 나타내고, 맨 아래 그림은 각 블록별 패킷 크기의 SCV를 나타낸다.

이 그림에서 보는 바와 같이 블록의 패킷 크기의 변이가 크게 되면, 정적 샘플링은 필요한 수보다 적은 양의 패킷을 샘플링하게 되어 (언더 샘플링) 트래픽 부하 예측 오류가 커지게 된다. 반대로, 블록의 패킷 크기 변이가 작은 경우, 정적 샘플링은 필요 이상의 패킷을 샘플링하여 (오버 샘플링), 측정 장비의 효율을 메모리와 패킷 처리 능력 측면에서 감소시키게 된다. 더욱이 이와 같은 오버 샘플링과 언더 샘플링의 잦은 천이로 인해 정적 샘플링은 트래픽 예측 오류의 분산을 증가시키게 된다. 즉, 불필요한 오버 샘플링 없이 샘플링을 통한 트래픽 부하 예측 오류를 지정된 오류 허용 범위 내로 제한하기 위해서는, 패킷 샘플링 확률은 트래픽 부하의 동적인 변화에 따라 적응성 있게 변화되어야 한다.

최적 랜덤 샘플링에서는 각 블록별로 최적 샘플링 확률을 미리 정하였다. 그러나 실제 네트워크 상황에서는 최적 샘플링 확률을 정하기 위한 트래픽 파라메터인 블록 내의 실제 패킷 크기 분포 SCV와 입력 패킷 수를 블록이 시작하는 시점에서 미리 알 수 없다. 따라서 다음 장에서는 이와 같은 문제를 해결하기 위한 방법과 이로 인한 트래픽 부하 오류에 관한 분석을 수행한다.

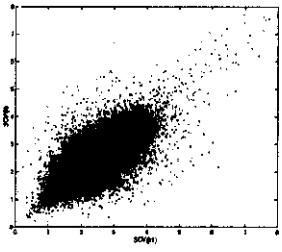
III. 적응성 있는 랜덤 패킷 샘플링 기법

본 장에서는 이전 블록들에서 샘플 패킷들을 통해 얻어진 값들을 이용하여 현재 블록의 샘플링 확률 결정에 핵심이 되는 두 인자인 패킷 크기 분포 SCV와 입력 패킷수를 AR (Auto-Regressive) 모델을 사용하여 예측하는 방법을 제시한다. 또한 AR 모델의 타당성은 실제 네트워크 트래픽 데이터를 통해 실험적으로 검증한다. 이로 인해 각 블록별로 예측되는 트래픽 부하에는 샘플링에 의한 오류와 AR 모델을 통한 트래픽 파라메터 예측에 의한 오류가 포함되게 된다. 본 장에서는 이와 같은 오류를 수학적으로 분석하여 이와 같은 오류들이 트래픽 부하 예측에 미치는 영향을 정량화하고 이를 제어하기 위한 방법에 관해 논한다.

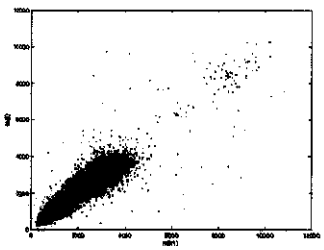
1. 트래픽 파라메터 예측을 위한 AR 모델

예측하고자 하는 파라메터의 과거 값과 현재 값 간에 연관 (correlation)이 클수록 예측의 정확성 및 효용성은 증대된다. 따라서 우리는 예측 하고자 하는 트래픽 파라메터의 예측 가능 여부를 알아보기 위해, 다양한 인터넷 트래픽 측정 데이터를 대상으로 패킷 크기 분포 SCV와 입력 패킷 수, m, 각각에 대하여 연관 관계를 조사하였으며, 이들 각각은 모두 연속적인 두 블록간에 강한 연관 관계를 나타낸다는 것을 검증하였다. 한 예로써 그림 2에 표 A.1의 Π_4에 대하여 블록 크기를 60초로 하였을 경우 SCV와 m의 scatter plot을 보였다. 그림에서 보는 바와 같이 연속적인 두 블록 간에 SCV와 m

의 값은 강한 연관 관계를 가지며 특히, 이들 간에는 강한 선형적인 관계를 나타낸다는 것을 볼 수 있다. 따라서 본 논문에서는 이와 같은 강한 선형적인 관계에 착안하여, 이전 블록의 트래픽 파라메터



(a) 블록당 입력 패킷 크기의 SCV



(b) 블록당 입력 패킷 수
그림 2. 패킷 크기 SCV와 입력 패킷 수의 연관 관계

값을 이용하여 현재 블록의 트래픽 파라메터 값을 예측하기 위해 선형 회기 (Linear Regression) 기법 중 하나인 AR 모델을 이용하였다[9].

다른 시 계열 모델과 비교하여 AR 모델은 직관적으로 이해 가능하며, 모델 파라메터들을 간단한 선형 방정식으로 구할 수 있으므로 온라인 트래픽 부

표 1. 분석에 사용된 기호들

S_k; k번째 블록의 실제 패킷 크기 SCV
S_k^s; k번째 블록의 샘플 패킷 크기 SCV
$\widehat{S_k^s}$; k번째 블록에서 예측된 샘플 패킷 크기 SCV
n_k^*; k번째 블록에서 필요한 최소한의 샘플 수
$\widehat{n_k^*}$; k번째 블록에서 예측된 최소한의 필요 샘플 수
$\widehat{n_k}$; k번째 블록에서 실제 샘플된 패킷 수
m_k; k번째 블록에서 실제 입력 패킷 수
$\widehat{m_k}$; k번째 블록에서 예측된 입력 패킷 수

하 예측에 매우 효율적이다.[9] 트래픽 파라메터 예측을 위해 본 논문에서 제시하는 AR 모델은 다음과 같이 동작하며 앞으로 전개될 수식들을 위해 사용되는 기호들을 표 1에 정리 하였다.

1.1. SCV 예측

S_k를 k번째 블록의 패킷 크기 분포 SCV라고 하고, S_k^s를 k번째 블록에서 랜덤하게 샘플된 패킷들의 크기 분포의 SCV라고 하면 이들 간에는 다음과 같은 관계가 성립한다.

$$S_k^s = S_k + Z_k \tag{6}$$

여기서 Z_k는 샘플 패킷으로부터 실제 패킷 크기 SCV를 예측할 때 발생되는 오류를 나타낸다. 이제 $AR(u)$ 모델[9]에 의해 위의 식은 다음과 같이 표현된다.

$$S_k^s = \sum_{i=1}^{u} a_i^s S_{k-1}^s + e_k^s \tag{7}$$

여기서 u는 예측에 사용된 이전 블록의 수를 나타내며, a_i는 모델 파라메터, e_k^s는 예측 오류로서 AR 모델의 특성에 따라 평균 0이고 분산 $var(e_k^s) = \sigma^2_{S_k^s}(1-\sum_{i=1}^{u} a_i^s \rho_{S_k^s,i})$인 정규 분포를 따르며 이전 블록의 예측 오류들과 연관이 0인 특성을 가진다. ($\rho_{S_k^s,i}$는 S_k^s의 lag-i 자기 연관을 나타낸다.) a_i는 식 (8)과 같이 과거 ν개의 S_i^s 값을 이용하여 선형 방정식을 풀면 얻을 수 있다. $\nu \geq 1$는 구성 가능한 값으로서 u와는 무관하며 일반적으로 메모리 크기라고 불린다.

$$\rho_h = \sum_{i=1}^{u} a_i \rho_{h-i} \tag{8}$$

$h = \nu, \ldots, \nu - u + 1$, ρ_h; 데이터의 lag-h 자기 상관

위의 $AR(u)$ 모델에 따라 (k-1)번째 블록의 끝에서 k번째 블록의 SCV는 u개의 이전 블록의 샘플 패킷의 SCV를 이용하여 다음과 같이 예측된다.

$$\hat{S}^{s}_{k} = \sum_{i=1}^{u} a_i^s S^s_{k-i} \quad (9)$$

이제 위의 식 (6), (7), (9)에 따라 k번째 블록에서 $AR(u)$ 모델에 의해 예측된 SCV, $\hat{S}^s_k$는 다음과 같이 주어진다.

$$\hat{S}^{s}_{k} = S_k + Z_k + e^s_k \quad (10)$$

식 (10)에서 보는 바와 같이 이전 블록의 샘플 패킷들의 SCV, S^s_{k-i}를 이용하여 k번째 블록의 실제 패킷 분포 SCV, S_k를 예측하는 경우에는 두 가지 오류가 수반되며, 하나는 랜덤 샘플링에 의한 오류인 Z_k이고 다른 하나는 예측 모델 $AR(u)$에 의한 예측 오류 e^s_k이다.

1.2. 블록당 입력 패킷 수 (m) 예측

k번째 블록의 입력 패킷 수를 m_k라고 하면 위와 비슷한 방법에 의해 AR(u) 모델에 의해 $m_k = \sum_{i=1}^{u} b_i m_{k-i} + e_{m,k}$로 예측된다. (여기서 b_i는 모델 파라메터이며 $e_{m,k}$는 예측 오류로서 평균 0인 정규분포를 가지며 이전 블록의 예측 오류들과는 연관을 가지지 않는다.) 이제 $\hat{m}_k$를 k번째 블록에서 예측된 입력 패킷 수라고 하면 $AR(u)$모델에 따라 $\hat{m}_k = \sum_{i=1}^{u} b_i \hat{m}_{k-i}$가 된다.

SCV의 예측에서와 마찬가지로 이전 블록의 샘플 패킷 수를 기반으로 현재 블록의 패킷 수를 예측하는 경우에는 샘플링에 의한 오류와 예측에 의한 오류가 수반된다. 그러나 현재 상용 라우터는 라인 카드에 패킷 카운트를 지원하고 있고 패킷 카운터를 매 패킷 입력마다 하나씩 증가시키는 것은 입력 패킷의 헤더를 검사하지 않으므로 라우터의 성능에 많은 영향을 주지 않는다. 따라서 각 블록의 끝에서 실제 블록 내의 입력 패킷 수를 알 수 있고 이 경우에는 현재 블록의 패킷 수는 이전 블록의 실제 패킷 수로부터 예측 가능하다. 즉, $\hat{m}_k = \sum_{i=1}^{u} b_i m_{k-i}$이 되어 입력 패킷 수 예측에는 오직 $AR(u)$ 모델에 의한 예측 오류만이 포함된다.

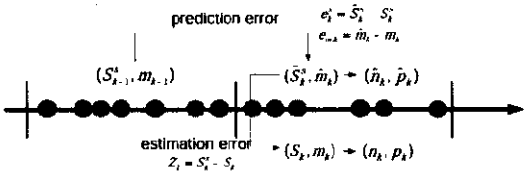


그림 3. 트래픽 파라미터 예측 오류

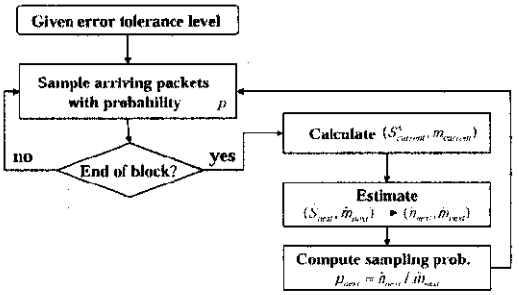


그림 4. 적응성 있는 랜덤 패킷 샘플링 플로우 차트

1.3. 샘플링 확률

이제 SCV와 입력 패킷 수가 예측되었으므로 k번째 블록에서 오류 허용 범위를 만족시키기 위한 최소한의 샘플 패킷 수와 샘플링 확률은 다음과 같이 결정된다.

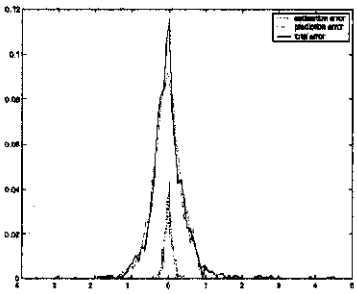
$$\hat{n}_k = z_p \hat{S}^{s}_{k}, \quad \hat{p}_k = \frac{\hat{n}_k}{\hat{m}_k} \quad (11)$$

그림 3은 트래픽 파라메터 SCV와 m을 예측하는 전체 과정을 보여주며, 그림 4는 적응성 있는 랜덤 샘플링 과정의 플로우 차트를 나타낸다. AR 예측 모델을 이용하면 모델 파라메터들 (a_i, b_i)은 각 블록의 끝에서 계산 되어져야 한다. 앞서 언급한 바와 같이 이 값들은 선형 방정식에 의해 얻어지므로 이들을 계산하기 위한 계산상의 복잡성은 $O(\nu)$가 된다. (ν는 메모리 크기) 실제 트래픽 측정 데이터를 사용하여 시험한 결과 원하는 성능을 얻기 위한 ν 값은 그리 크지 않아도 된다(대략 5)는 것을 알 수 있었다.

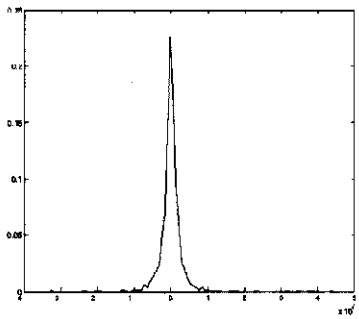
2. 트래픽 부하 예측시 발생하는 오류 분석

이 절에서는 샘플링과 예측을 통해 트래픽 부하 예측시 발생되는 오류를 분석한다. 식 (9)에서 보는

바와 같이 SCV 예측에는 샘플링에 의한 오류 Z_k와 예측에 의한 오류 e_k^s가 야기된다. 실제 네트웍 트래픽 측정 데이터를 이용한 결과 이들 오류는 이론에서와 마찬가지로 평균 0인 정규분포를 따른다는 것을 확인할 수 있었다. 예로써 그림 5-(a)는 Π_1에 대해 Z_k, e_k^s와 $Z_k + e_k^s$를 나타내며, 그림에서 보는 바와 같이 이들 모두는 중심이 0인 종형태를 나타낸다. 보다 정량적인 분석을 위해 skewness 검사와 kurtosis 검사[11]를 한 결과 이들은 모두 정규 분포를 가짐을 확인하였다. 같은 방법을 입력 패킷 수 예측시 발생하는 오류 $(e_{m,k})$에 대하여 적용한 경우에도 같은 결과를 얻었다 (그림 5-(b)).



(a) 샘플링과 예측에 의한 패킷 크기 SCV 오류



(b) 패킷 수 예측 오류

그림 5. 정규분포 형태의 오류

이와 같은 결과에 따라 제안 기법에 의해 발생되는 오류는 모두 평균 0인 정규분포를 가진다고 가정하면, 적응성 있는 샘플링에 의한 오류은 다음과 같이 정량화 된다. 분석의 편이를 위해 $AR(1)$ 모델을 가정하면, 예측 오류, e_k^s의 분산은 $\sigma^2_{S_k^s}(1-a_1^s \rho_{S_k,1})$이 되므로 식 (6)과 (7)에 의해 S_k 예측시 발생되는 전체 오류의 분산은 다음과 같이 주어진다.

$$var(Z_k) + var(e_k^s) = \sigma^2_{S_k^s}(1 - 2a_1^s \rho_{S_k,1} + (a_1^s)^2) \quad (12)$$

이제 $\tilde{n}_k = m_k \cdot \hat{n}_k / \hat{m}_k$를 k번째 블록에서 샘플링 확률 $\hat{p}_k = \hat{n}_k / \hat{m}_k$로 입력 패킷을 샘플링 한 경우 블록의 끝에서 실제로 얻어지는 샘플 패킷 수라고 하면 k번째 블록에서 예측되는 트래픽 부하는 다음과 같이 주어진다.

$$V = \frac{m_k}{\tilde{n}_k} \sum_{j=1}^{\tilde{n}_k} X_j \quad (13)$$

여기서 X_j는 k번째 블록에서 랜덤하게 샘플된 j번째 패킷의 크기를 나타낸다. 랜덤 변수를 랜덤한 갯수만큼 합한 값에 대한 중심 극한 정리[12]에 의해 다음과 같은 lemma와 theorem을 얻을 수 있다.

Lemma 1. $\lim_{n_k^* \to \infty} \frac{\tilde{n}_k}{n_k^*} \to 1$

(증명) $\lim_{\hat{m}_k \to \infty} \frac{m_k}{\hat{m}_k} = 1$ 이고, $n_k \to \infty$ 일 때 $S_k^s \to S_k$, $\hat{n}_k \to n_k$가 된다. 또한 $\tilde{n}_k = \hat{n}_k \cdot \frac{m_k}{\hat{m}_k}$ 이므로 $\lim_{n_k^* \to \infty} \frac{\tilde{n}_k}{n_k^*} \to 1$ ■

Theorem 1. 각 블록에서 트래픽 부하 예측에 의해 발생되는 오류는 확률 $1-\eta$로 다음과 같이 제한된다.

$$\left|\frac{V - \hat{V}}{\hat{V}}\right| \le \varepsilon + \frac{1}{\sqrt{z_p}}(1+\varepsilon) Y + o(\frac{1}{m})$$

$$\approx \varepsilon + \frac{1}{\sqrt{z_p}}(1+\varepsilon) Y$$

여기서 $z_p = (\frac{\Phi^{-1}(1-\eta/2)}{\varepsilon})^2$이며 Y는 평균 0 분산 1인 표준 정규 분포를 따르는 랜덤 변수이다.

(증 명)

$$\hat{V} = \frac{m}{\tilde{n}} \sum_{i=1}^{\tilde{n}} X_i = V + \frac{(\sigma^s \sqrt{n} Y + o(\sqrt{n})) \cdot \hat{m}}{\hat{n}}$$

$$\approx V+\frac{\widehat{m}\mu^s}{\sqrt{z_p}}\cdot\frac{\sqrt{S(S+Z)}}{S+Z+e^s}Y$$ 이고

$\frac{\sqrt{S(S+Z)}}{S+Z+e_s}\le\frac{S+Z/2}{S+Z}\le 1$이며, 확률 $1-\eta$로 $|\frac{V-V}{V}|<\varepsilon$이므로 확률 $1-\eta$로 다음 식

$|V-V|<V\varepsilon+\frac{V}{\sqrt{z_p}}(1+\varepsilon)Y+\frac{e_m\mu^s}{\sqrt{z_p}}Y$가 성립

하므로 $|\frac{V-V}{V}|\le\varepsilon+\frac{1}{\sqrt{z_p}}(1+\varepsilon)Y+o(\frac{1}{m})$

$$\approx\varepsilon+\frac{1}{\sqrt{z_p}}(1+\varepsilon)Y$$ ■

Theorem 1에 의해 적응성 있는 랜덤 샘플링에 의한 오류의 이론적 한계값은 확률 $1-\eta$로 다음과 같이 주어진다.

$$Var(|\frac{V-V}{V}|)\le\frac{(1+\varepsilon)^2}{z_p}\qquad(14)$$

즉, 식 (14)에 따라, 트래픽 파라메터 예측과 샘플링을 통한 오류를 보상하고 원하는 오류 허용 범위 내로 오류값을 제한하기 위해서는 다음과 같이 보다 엄격한 오류 허용값 (ε')을 설정해야 한다.

$$\varepsilon'=\varepsilon-s\frac{(1+\varepsilon)}{\sqrt{z_p}}\qquad(15)$$

여기서 s는 트래픽 부하 예측 오류의 분산 값을 제어하기 위한 제어 파라메터이다.

Ⅳ. 실제 측정된 트래픽을 통한 성능 평가

본 장에서는 제안 기법인 적응성 있는 랜덤 패킷 샘플링의 성능을 실제 측정 데이터를 사용하여 정적 샘플링과 비교함으로써 검증한다. 실험에 사용된 데이터는 표 A.1에 정리되어 있으며, 모든 데이터에 대하여 동일한 결과를 얻었으나 논의의 일관성을 위해 본 장에서는 Π_1 적용시 얻은 결과만을 나타내기로 한다.

그림 6은 ε 값을 변화시켜가며, 이에 따른 트래픽 부하 예측 오류 확률을 $1-\eta$로 나타낸 것이다. 정적 샘플링과의 공정한 비교를 위해서는 두 기법간 동일한 샘플링 효율을 가지도록 해야 하며, 이를 위해 정적 샘플링 확률은 전체 데이터에 대한 샘플의 수가 제안 기법을 적용한 경우와 같아지도록 정하였다. 그림에서 보는 바와 같이 모든 ε값에 대하여 정적 샘플링은 주어진 오류 허용 확률 범위 $1-\eta$보다 큰 오류를 발생시키는 반면 제안 기법은 지정된 오류 값의 한계를 따른다는 것을 볼 수 있다.

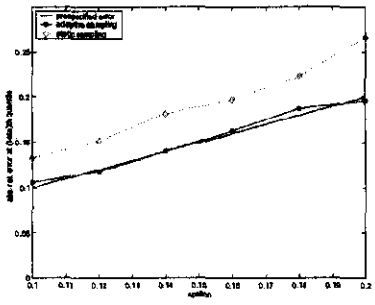


그림 6. ε에 따른 트래픽 부하 측정 예측 오류 ($\eta=0.1, B=60\text{sec}$)

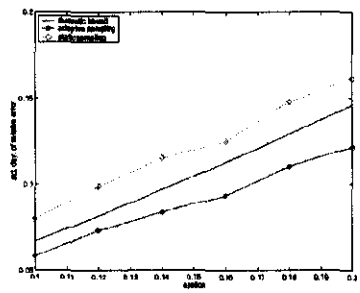
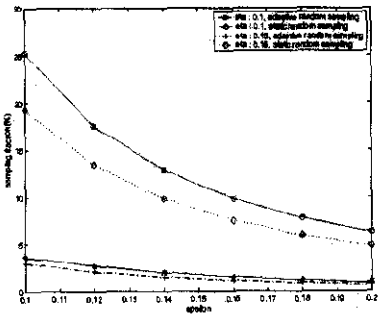


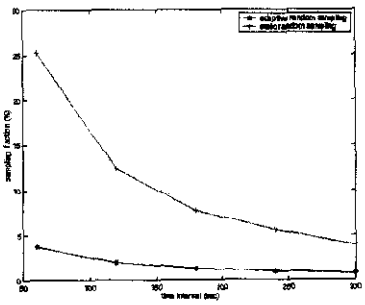
그림 7. 트래픽 부하 예측 오류의 표준 편차 ($\eta=0.1, B=60\text{sec}$)

그림 7은 제안 기법과 정적 샘플링 간에 트래픽 부하 예측시 발생하는 오류의 표준 편차를 비교하고 있다. 발생 오류의 분산이 작을수록 샘플링을 통해 얻은 값이 실제 값에 보다 근사하므로 발생 오류의 표준편차는 샘플링 기법을 평가하는 중요한 요소가 된다[18]. 그림에서 보는 바와 같이 제안 기법에 의한 오류는 항상 식 (14)로 주어지는 이론적인 한계값 이하로 주어진다. 이와는 달리 정적 샘플링에 의한 오류는 2장에서 언급한 바와 같이 잦은 오버 샘플링과 언더 샘플링으로 인해 분석된 이론적 한계값 보다 커지게 된다.

그림 8은 정적 샘플링과 제안 기법을 자원의 효율성 면에서 비교하고 있다. 자원의 효용성은 전체 데



(a) ε에 따른 SF



(b) 타임 블록 간격에 따른 SF

그림 8. Sampling Fraction $(\varepsilon, \eta) = (0.1, 0.1)$

이터에 대한 샘플 데이터의 수의 비로 정의되는 SF(sampling fraction)으로 측정하였다. SF는 샘플링 기법의 처리 요구 사항과 샘플 저장 공간 요구사항을 평가하기 위한 간접적인 척도가 된다. 비교를 위해 정적 샘플링 확률은, 정적 샘플링에 의한 오류가 제안 기법이 제공하는 오류와 같아지도록 설정하였다. 그림 8-(a)에서 보는 바와 같이 오류 한계값이 작아질수록 보다 많은 패킷을 샘플해야 한다. 그러나 제안 기법은 정적 샘플링 기법에 비해 훨씬 적은 양의 패킷 샘플로 같은 오류 한계 값을 만족시킬 수 있다. 그림 8-(b)는 타임 블록이 SF에 미치는 영향을 나타내며, 일반적으로 타임 블록이 커질수록 SF는 감소한다. 이와 같은 이유는 샘플링의 정확도는 입력 패킷 수가 아니라 오직 필요한 최소한의 샘플 수에 의해 결정되기 때문이다. 즉, 블록 크기가 커질수록 동일 오류 한계 값을 만족시키기 위해 필요한 샘플 패킷의 수가 전체 입력 패킷에 비해 상대적으로 작아지기 때문에 블록이 커질수록 SF는 작아진다. 이와 같은 의미로, 제안 기법은 입력율이 매우 높은 경우에도 같은 양의 패킷만을 샘플링하므로 확장성을 가진다. 즉, 그림 8은 Π_1에 의해 얻어졌으며 Π_1의 평균 데이터 입력율은 1Mbps 이하였음에도 불구하고 5% 이하의 SF를 보였다. 제안 기법을 평균 입력율이 35.36Mbps인 Π_6에 적용한 경우(블록 크기 30초, $(\varepsilon, \eta) = (0.1, 0.1)$) SF는 단지 0.022%였다.

오류 제어에 필요한 샘플의 수는 패킷 크기 SCV에 의해 결정되므로, 패킷 크기 SCV가 SF에 미치는 영향을 고찰하였다. 표 2는 실험에 사용한 측정 트래픽들의 간단한 SCV 통계값을 나타낸다. 표 2에서 보는 바와 같이 인터넷 트래픽은 측정 시간, 링크 속도 등에 따라 광범위한 차이를 보인다. 표 2는 또한 $(\varepsilon, \eta) = (0.1, 0.1)$의 경우 SF 값을 보여주며, SCV가 SF에 직접적인 영향을 준다는 것을 볼 수 있다. (일반적으로 SCV가 작을수록 SF도 작아진다.) 또한 Π_6와 Π_8의 비교에서 보는 바와 같이 동일한 SCV에 대해서는 패킷 입력율이 SF에 영향을 준다. 즉, 이들간의 SCV는 비슷하나 표 A.1.에서 보는 바와 같이 Π_6의 입력율은 Π_8의 입력율에 비해 약 5배 정도 빠르기 때문에 Π_6의 SF가 다섯배 정도 작게 나타남을 볼 수 있다. 그림 9는 SCV와 트래픽 부하 간의 scatter plot을 나타낸다. 그림에서 보는 바와 같이 링크의 이용율이 높은 경우 패킷 크기 SCV는 작아지는 경향을 보인다. 이는 대부분의 인터넷 트래픽은 TCP 응용에 의한 것이며[17], TCP 패킷의 데이터 크기는 거의 비슷하기 때문이다[14]. 패킷 SCV가 작을수록 같은 오

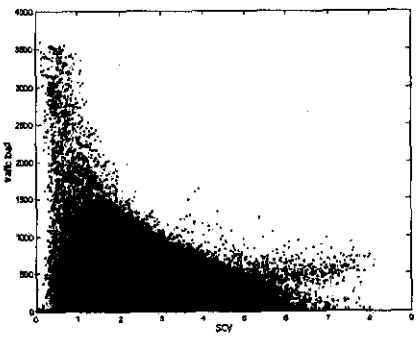


그림 9. 트래픽 부하와 패킷 크기 SCV와의 관계

류 허용 값에 대해 제안 기법의 SF는 낮아진다. 즉, 고속 링크의 이용율이 높더라도 같은 오류 허용 값에 대해 제안 기법에 의해 필요한 샘플 패킷 수는 동일하므로, 상대적으로 라우터의 트래픽 측정으로 인한 부담은 줄어들게 된다.

표 2. SCV와 SF와의 관계 $(\varepsilon, \eta) = (0.1, 0.1)$

	평균		최소		최대		SF(%)	
	B=60s	B=300s	B=60s	B=300s	B=60s	B=300s	B=60s	B=300s
Π_1	1.70	1.68	0.69	0.85	5.21	3.52	5.48	0.67
Π_2	2.11	2.09	0.39	0.60	5.29	4.32	5.85	0.83
Π_3	2.59	2.48	0.47	0.90	7.39	6.44	5.07	1.01
Π_4	1.18	1.12	0.32	0.55	2.55	1.38	1.67	0.30
Π_5	0.89	0.89	0.75	0.78	1.13	0.99	0.91	0.25
	B=30s		B=30s		B=30s		B=30s	
Π_6	1.26		1.25		1.27		0.022	
Π_7	1.35		1.35		1.35		0.022	
Π_8	1.22		1.16		1.32		0.12	

V. 결론 및 추후 연구 과제.

링크가 고속화되고 있는 상황에서 트래픽 부하 측정을 위해 모든 패킷을 측정하여 패킷의 내용을 검사하는 것은 라우터의 성능 저하를 가져 온다. 또한 사용자의 사용 형태 변화, 새로운 응용의 등장, 네트워 제어 프로토콜, 링크 고장 등 다양한 이유로 인해 네트워 트래픽 부하가 변화하는 상황에서 고정 확률로 랜덤하게 패킷을 샘플링하는 정적 샘플링 기법은 샘플링으로 인한 오류를 특정 값 이하로 제한하기 어려우며, 고정 샘플링 확률을 어떻게 정의할 것인가에 관한 과학적 접근방법이 어렵다. 따라서 본 논문에서는 트래픽 부하 측정을 위해 적응성 있는 샘플링 기법을 제안하고, 수학적으로 성능을 분석하였으며, 실제 측정 트래픽을 이용하여 성능을 평가하였다.

본 논문에서는 분석을 통해 오류를 특정 값 이하로 제한하기 위해 필요한 샘플의 수는 입력 패킷 크기의 SCV와 정비례 한다는 것을 밝혔다. 실제 샘플링을 수행하기 전에 샘플링 확률을 미리 정하기 위해, AR 모델을 이용하여 패킷 크기 SCV와 입력 패킷 수를 예측하였고, 분석과 실험을 통해 AR 모델의 타당성을 검증하였다. 실제 측정된 인터넷 트래픽을 이용한 실험을 통해 제안 기법은 사용자가 지정한 오류 범위 내로 트래픽 부하 예측 오류를 제한하며 입력 패킷 수에 비해 실제 측정해야 하는 패킷의 비가 매우 낮기 때문에 자원의 사용 면에서도 매우 효과 적이라는 것을 검증하였다.

본 논문에서 제안한 적응성 있는 샘플링 기법은 오류를 지정 범위 내로 제한 할 수 있으며, 확장성을 가지므로 트래픽 부하 측정을 기반으로 하는 여러 응용들의 기반으로 사용될 것으로 기대된다.

추후 연구 과제로는 제안된 패킷 샘플링을 이용하여, 트래픽 전체 부하 뿐만이 아니라 현재 이슈가 되고 있는 각 플로우별 크기, 지속시간 등의 측정 방법을 제시하고, 보다 엄격한 샘플링 오류 제어를 위한 효율적 제어 기법에 대한 정량적 분석 및 효과적인 메커니즘의 개발이 진행 중에 있다.

부록 A. 실험 데이터

본 논문에서는 다양한 속도, 종류의 링크에서 다양한 시간에 측정된 인터넷 트래픽 데이터를 사용하였으며, 표 A.1은 본 논문의 실험에 사용된 데이터를 나타낸다.

AIX 데이터는 NASA Ames, Moffet Field, MAE_WESR를 연결하는 링크에서 측정된 데이터

표 A.1. 사용 데이터 요약 표

Trace Name	Trace File	입력율	지속 시간
Π_1	Auckland-II 19991201-192548-0	92.49KBps	24h 2m 58s
Π_2	Auckland-II 19991201-192548-1	55.16KBps	24h 2m 57s
Π_3	Auckland-II 19991209-151701-1	49KBps	23h 11m 38s
Π_4	Auckland-II 20000117-095016-0	168KBps	2h 23m 15s
Π_5	Auckland-II 20000114-125102-0	222.14KBps	21m 37s
Π_6	AIX 989950026-1	25.36MBps	90s
Π_7	AIX 20010801-996689287-1	21.60MBps	90s
Π_8	COS 983398787-1	4.95MBps	90s

이다. 링크 종류는 OC12c POS 프레임을 교환하며 PPP encapsulation을 사용한다. 이로 인해 이곳에서 얻은 데이터는 TCP/IP 헤더의 일부 정보가 누락되어 있다.

CoS 데이터는 Colorado 주립 대학과 Front Range GigaPop (FRGP) 간의 링크로서 속도는 OC3c이고 ATM/AAL5 형태의 프레임을 교환한다. 이 링크는 각각 I2트래픽, CSU 자체 트래픽, 공중망 트래픽을

지원하는 3개의 PVC로 구성된다.
Auckland-II 데이터는 Auckland 대학의 인터넷 업링크(OC3c ATM 링크)에서 측정된 GPS로 동기화된 IP 헤더 트래이스이다. 다른 MOAT 데이터와는 달리 이들은 1시간 가량의 긴 측정기간 동안 측정되었다.

참고 문헌

[1] Kimberly C. Claffy, George C. Polyzos and Hans-Werner Braun, "Application of sampling methodologies to network traffic characterization" , in Proceedings ACM SIGCOMM '93, San Francisco, CA, September 13--17, 1993.

[2] Nick G. Duffield and Matthias Grossglauser, " Trajectory sampling for direct traffic observation" , Proceedings of ACM SIGCOMM 2000 pp271-28.

[3] Nick Duffield, Carsten Lund, and Mikkel Thorup, "Charging from Sampled Network Usage" , ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA , Nov. 2001

[4] Cristian Estan and George Varghese, "New Directions in Traffic Measurement and Accounting" , ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA , Nov. 2001

[5] PMA Traces Archive http://moat.nlanr.net utilization

[6] Auckland Traffic Traces http://pma.nlanr.net/Traces/long/

[7] Donald A. Berry and Bernard W. Lindgren, Statistics theory and Methods, 2nd ed., Duxbury Press, ITP, 1996

[8] Sampled NetFlow. http://www.cisco.com/univercd/cc/tc/doc/product/software/ios120/120newft

[9] John M. Gottman, "Time-series analysis" ', Cambridge University Press, 1981

[10] - " Adaptive Random Sampling for Load Change Detection" , Department of Computer Science and Engineering at University of Minnesota, Technical Report Nov. 2001

[11] A.K. Bera and C.M. Jarque, "An efficient large-sample test for normality of observations and regression residuals" , Working Papers in Economics and Econometrics, 40, Australian National University, 1981

[12] P. Billingsley, "Convergence of Probability Measures" , New York Wisley, 1968 (p.369)

[13] C. R. Rao, ''Sampling Techniques'' 2nd ed., N.Y., Wiley. 1973

[14] W. Richard Stevens, "TCP/IP Illustrated, vol. 1" , Addison-Wesley, 1994

[15] http://www.caida.org/tools/measurement/coralref/doc/doc/index.html

[16] Daniel W. McRobb, "cflowd Design" , http://www.caida.org/tools/measurement/cflowd/

[17] K Claffy, Greg Miller, and Kevin Thompson, "the nature of the beast: recent traffic measurements from an Internet backbone" , INET' 98 Geneva, Switzerland, 1998

[18] C. R. Rao, "Sampling Techniques" , 2nd Ed., N.Y., Wiley 1973

박 재 성 (Jaesung Park)
1997년 2월: 연세대학교 전자 공학과 석사
2001년 2월: 연세대학교 전기,전자 공학과 박사
2001년 5월 ~ 2002년 4월: University of Minnesota 연구원
2002년 6월 ~ 현재: LG 전자 안양 연구소 UMTS 연구소 선임 연구원

<주관심분야> 인터넷 트래픽 측정 및 분석, IP-based RAN, Mobile IP, Fast Handover

최 백 영 (Choi-Back Young)
1995년: 포항공과대학교 전산과학과 석사
2003년 8월: Dept. Computer Science and Engineering, University of Minnesota 박사 졸업
2002년 ~ 현재; Sprint Lab. IP Group 연구원

<주관심분야> 네트웍 트래픽 측정 및 분석, 인터넷 토폴로지, QoS Routing, QoS

Zhi-Li Zhang
1997년 2월: Dept. Computer Science, University of Massachusetts at Amherst 박사 졸업
1997년 ~ 현재: Dept. Computer Science and Engineering, University of Minnesota 조교수

<주관심분야> 컴퓨터 네트웍, 실시간 분산 멀티미디어 시스템, 컴퓨터 시스템 성능평가 및 모델링