

# 정확한 고정대역폭 추정을 통한 새로운 가용대역폭 측정 기법

## (A New Available Bandwidth Measurement Technique with Accurate Capacity Estimation)

조 성 호 <sup>†</sup>      최    한 <sup>†</sup>      김 종 권 <sup>\*\*</sup>  
(Seongho Cho)    (Han Choe)    (Chong-kwon Kim)

**요 약** 인터넷에서 단대단 (end-to-end) 가용대역폭 측정은 분산된 응용 서비스, 단대단 QoS 보장 등에 유용하게 이용할 수 있다. 일반적으로 단대단 가용대역폭 측정에는 단일-홉 갭 모델에 기반한 패킷 트레인 (packet train) 전송을 통한 측정이 많이 사용되고 있다. 하지만 패킷 트레인의 출력 간격은 네트워크 망 구성과 경쟁 트래픽에 의해서 영향을 받으므로, 이 때 발생한 오차에 의해서 가용대역폭 측정에도 오차가 발생하게 된다. 본 논문에서는 기존의 방법들이 가지고 있는 연이은 (back-to-back) 패킷 트레인을 전송하여 링크의 고정대역폭을 측정에 의한 부정확성을 해결하기 위해서 패킷 트레인의 초기 갭에 영향을 받지 않는 가용대역폭 측정 알고리즘을 제안한다. 또한 빠른 측정을 위해서 패킷 트레인 전송 시 적절한 초기 간격을 예측하는 방법을 제안한다. 제안하는 방법의 성능을 검증하기 위해서 다양한 망구조에서 실험을 통해서 기존의 기법들과 비교하였다.

**키워드** : 가용대역폭 측정, 대역폭 측정, 패킷 트레인, 단대단 QoS (Quality-of-Service)

**Abstract** Measuring the end-to-end available bandwidth in the Internet is a useful tool for distributed application services or QoS (Quality-of-Service) guarantee. To measure the end-to-end available bandwidth, Single-hop Gap model-based packet train measurement techniques are well-known. However, the error of packet train output gap can happen by network topologies. This error of the output gap causes the inaccuracy of the available bandwidth measurement. In this paper, we propose a new end-to-end available bandwidth measurement technique with accurate capacity measurement and fast convergence methods. To solve the erroneous capacity measurement problem of the back-to-back packet train transmission, we propose a new available bandwidth measurement method by decoupling the capacity measurement with the initial gap of the packet train. Also, we propose a new technique to predict the proper initial gap of the packet train for faster convergence. We evaluate our proposed method by the simulation in various topologies comparing with previous methods.

**Key words** : Available Bandwidth Measurement, Bandwidth Measurement, Packet Train, End-to-end QoS (Quality-of-Service)

### 1. 개 요

기존 인터넷은 데이터 전송을 위해서 최선형(best-

effort) 서비스에 기반하고 있다. 하지만 네트워크가 고도화됨에 따라서 실시간 멀티미디어 전송, 상호간 (Peer-to-peer) 파일 공유, 멀티캐스트와 같은 새로운 응용 서비스뿐만 아니라, 단대단(end-to-end) QoS (Quality-of-Service), 혼잡 제어, SLA(Service Level Agreement) 확인, 트래픽 분석(Traffic Engineering)과 같은 다양한 공학적 문제들이 제기되고 있다.

위와 같은 문제들은 가용대역폭(Available Bandwidth) 측정을 통해서 해결할 수 있다. 예를 들면, 선택 가능한 인터넷 경로들의 가용대역폭을 측정함으로써 사

· 이 논문은 2005년도 두뇌한국21사업에 의하여 지원되었음

<sup>†</sup> 학생회원 : 서울대학교 전기.컴퓨터공학부  
shcho@popeye.snu.ac.kr  
hchoe@popeye.snu.ac.kr

<sup>\*\*</sup> 종신회원 : 서울대학교 전기.컴퓨터공학부 교수  
ckim@popeye.snu.ac.kr

논문접수 : 2004년 12월 1일  
심사완료 : 2005년 4월 29일

용자가 요구하는 대역폭을 보장할 수 있는 링크를 선택하거나, 효율적인 파일 전송 트리를 구성할 수 있다. 따라서, 가용대역폭 측정은 여러 ISP(Internet Service Provider)를 통해 네트워크가 구성되고 BcN (Broadband Convergence Network)이 현실화됨에 따라서 서로 다른 네트워크 환경에 있는 노드들에게 멀티미디어 서비스를 제공하는데 있어서 중요한 기반 기술 중의 하나이다.

가용대역폭을 측정하는 방법으로는 SNMP[1]와 같은 프로토콜을 이용하여 라우터에서 직접 정보를 얻어서 분석하는 방법과 사용자 측에서 정보를 직접 수집하는 방법 두 가지가 있다. 네트워크로부터 얻은 SNMP 정보를 가공하여 분석하는 도구로는 Multi Router Traffic Grapher (MRTG)[2], Round Robin Database(RRD) [3]와 같은 툴들이 있다. 이런 방법들은 네트워크 관리자가 사용할 수 있는 방법으로 라우터로부터 얻어진 데이터를 통해서 통계적 분석을 할 수 있기 때문에 정확한 데이터를 얻을 수 있는 장점이 있다. 하지만 위 방법들은 라우터에 직접 접근할 수 있어야 하기 때문에 네트워크 관리자에게만 권한이 부여되며, 관리 도메인이 다른 여러 ISP로 구성된 망에서는 현실적으로 적용하기 힘들다. 또한 상호간 파일 공유와 같은 경우는 별도의 중앙 관리자의 도움 없이 개별사용자들 간에 파일 공유 트리 등을 구성하기 때문에 단말이 네트워크의 정보를 수집하는 것이 필요하다. 따라서 단대단으로 네트워크 경로에 대한 정보를 수집할 수 있는 새로운 방법이 필요로 하게 되었다.

단대단으로 네트워크 경로의 정보를 측정하는 연구는 초기에는 단대단으로 병목 링크의 고정대역폭 (Bottleneck Link Capacity)을 측정하는 연구들[4-7]로 시작하여 현재는 가용대역폭 측정[8-13], 병목 링크의 위치 분석[14-16] 등으로 확장되고 있다. 특히 가용대역폭을 측정하는 분야가 현재까지 활발하게 연구되고 있다.

위의 단대단 고정대역폭과 가용대역폭 측정 기법들에서 사용되는 기본적인 개념들은 다음과 같이 정의할 수 있다. 네트워크 경로 L은  $i = 1, \dots, N$ 개의 개별 링크의 연결로 이루어져 있다. 각각의 링크  $i$ 는 최대 전송할 수 있는 초당 비트 전송률을 가지며, 이를  $C_i$  bps라고 한다. 이 때 경로 L 상에서 전송률이 가장 낮은 링크를 병목 링크라고 정의하고 링크의 비트 전송률을 해당 경로 L의 고정대역폭  $C_L$ 로 정의하며 다음과 같다.

$$C_L = \min \{C_i\}, i = 1, \dots, N \quad (1)$$

또, 링크  $i$ 에 대해서 시간  $t$ 에 존재하는 다른 플로우들의 전송률의 합을 경쟁 플로우의 양  $B_i(t)$  ( $0 \leq B_i(t) \leq C_i$ )로 정의한다. 여기서 링크  $i$ 의 시간  $[t, T]$  동안의 링크 이용률  $u_i(t, T)$  ( $0 \leq u_i(t, T) \leq 1$ )는

$$u_i(t, T) = \frac{1}{T} \int_t^{t+T} \frac{B_i(t)}{C_i} dt \quad (2)$$

로 정의한다. 이 때 링크  $i$ 의 시간  $[t, T]$  동안의 가용대역폭  $A_i(t, T)$ 는

$$A_i(t, T) = \frac{1}{T} \int_t^{t+T} (C_i - B_i(t)) dt \quad (3)$$

로 정의한다. 여기서 경로 L의 시간  $[t, T]$  동안의 가용대역폭  $A_L(t, T)$ 는

$$A_L(t, T) = \min(A_i(t, T)) \quad (4)$$

로 정의할 수 있다.

위에서 정의한 경로 상의 고정대역폭이나 가용대역폭을 단대단으로 측정하기 위해서 많은 방법들이 제안되었고, 측정 패킷의 형태로 패킷 쌍(packet pair) 혹은 패킷 트레인(packet train)이 많이 사용되었다. 이 기법은 송신 측에서 길이가 같은 패킷을 일정한 간격으로 전송하고 수신 측에서 변화된 간격을 측정하여 이를 기반으로 경로 상의 고정대역폭이나 가용대역폭을 측정하는 기법이다. 특히 가용대역폭 등을 측정하기 위해서는 네트워크를 병목 링크를 갖는 하나의 단일 홉으로 모델링하거나 수신 측에서 측정한 변화된 간격을 히스토그램 분석, 회귀분석 등을 사용하여 통계적으로 분석하는 방법들이 사용되고 있다. 하지만 기존 기법들은 가용대역폭 측정에 병목 링크의 고정대역폭을 이용함으로써 부정확하게 측정된 고정대역폭이 측정값의 정확도에 영향을 주거나, 측정에 많은 시간이 걸리는 문제 등이 존재한다.

본 논문에서는 기존의 단일-홉 갭 모델 (Single-hop Gap Model)[9]에 기반하여 병목 링크의 고정대역폭과 패킷 트레인의 초기 간격과의 관계를 분리하여 가용대역폭을 측정할 수 있는 새로운 기법인 DCIG (Decoupled Capacity estimation and Initial Gap of the packet train)을 제안한다. 이 방법을 통해서 얻어진 패킷 트레인에 의한 고정대역폭 측정 오차에 따른 가용대역폭의 오차를 줄일 수 있다. 또한 기존의 방법들이 패킷 쌍의 적절한 초기 간격을 찾는 데 많은 시간이 걸리는 문제를 해결하기 위한 방법으로 FC (Fast Converging) 기법을 제안한다. FC 기법은 패킷 쌍의 초기 간격에 대해서 출력 간격이 병목 링크의 고정대역폭에 반비례하고 경쟁 트래픽의 양에 비례해서 선형적으로 증가하는 점에 착안하여 그 변화량을 측정하여 적절한 초기 간격을 빠르게 찾는 기법이다. 이를 통해서 가용대역폭을 찾는 데 필요한 패킷 트레인의 숫자와 측정 시간을 줄일 수 있다.

논문의 구성은 다음과 같다. 2장에서는 기존 연구 기법을 고정대역폭 측정, 가용대역폭 측정, 병목 링크 위

치 파악기법들로 분류하고 각각의 기법이 갖는 특징을 분석하였다. 3장에서는 단일-홉 갭 모델에 대해서 살펴보고 이 모델에서 패킷 쌍과 패킷 트레인이 갖는 특성을 분석하였다. 4장에서는 제안하는 DCIG(Decoupled Capacity estimation with the Initial Gap of the packet train)와 FC(Fast Converging) 기법들을 설명하고, 5장에서는 이를 이용하여 가용대역폭을 측정하는 알고리즘을 제안하였다. 6장에서는 기존의 기법들과 비교하기 위해서 다양한 네트워크 환경에서 실험한 결과를 제시하고 알고리즘의 성능을 비교 분석하였다. 7장에서는 결론을 내리고 관련된 차후 연구 주제들을 간략히 살펴보았다.

## 2. 기존 연구

관련된 기존 연구들은 크게 병목링크의 고정대역폭을 측정하는 기법과 가용대역폭을 측정하는 기법, 경로 상 병목 링크의 위치를 찾는 기법으로 나눌 수 있다.

먼저 병목링크의 고정대역폭을 측정하는 기법에는 bprobe[4], Packet Bunch Modes (PBM)[5], Asymptotic Dispersion Rate (ADR)[6], Capacity Probe (CapProbe)[7] 등이 있다. bprobe는 서로 다른 패킷 길이를 갖는 많은 패킷 쌍(packet pair)을 전송하여 최소 전송 시간을 갖는 샘플을 구하고 이를 기반으로 패킷의 길이 당 전송률을 얻어서 경로 상의 고정대역폭을 측정하는 기법이다. PBM은 기존의 기법들이 패킷 쌍의 샘플이 단일형태(unimodal)임을 가정하고 측정하는 방법인 것과 다르게 패킷 쌍의 샘플이 다중형태(multimodal)일 수 있음을 증명하고 이에 따른한계를 극복하기 위해서 서로 다른 패킷 트레인을 전송하여 이를 통해서 다양한 모드를 처리할 수 있는 방법을 통해서 경로 상의 고정대역폭을 측정하였다. ADR은 네트워크의 망 구성 형태에 따라서 고정대역폭에 대한 측정이 높게 측정되거나 낮게 측정될 수 있음을 증명하고, 이를 이용하여 히스토그램의 빈도를 분석하여 고정대역폭을 측정할 수 있는 방법을 제시하였다. CapProbe는 패킷 쌍이 전달될 때 라우터의 대기열에서 겪는 지연에 의한 효과를 줄이기 위해서 패킷 쌍의 최소전송 지연을 갖는 샘플의 분산된 간격을 추출하여 병목링크의 고정대역폭을 측정하는 기법이다.

이런 기법들은 송신 측에서 패킷 쌍을 초기 간격 없이 연달아(back-to-back) 전송하는 경우 수신 측에서 측정된 샘플의 간격이 병목링크의 고정대역폭에 의해서만 영향을 받는 점에 기반하고 있다. 따라서 오차를 줄이고 정확한 측정을 위해서는 패킷 쌍이 링크 상의 경쟁 패킷에 영향을 받는 효과를 최소로 줄이는 것이 중요하고, 이를 위해서는 작은 크기의 패킷을 사용하는 것이

유리한 것으로 알려져 있다.

또한 가용대역폭을 측정하는 분야가 현재까지 활발하게 연구되고 있으며, 크게 단일-홉 갭 모델 (Single-hop Gap Model)에 기반한 연구들[8-10]과 자가 혼잡유발(Self-induced Congestion) 기법들[11-13]로 나눌 수 있다.

단일-홉 갭 모델에 기반한 연구들로는 Delphi[8], Initial Gap Increasing(IGI) / Packet Transmission Rate (PTR)[9], Spruce[10] 등이 있다. 위와 같은 단일-홉 모델에 기반한 기법들은 네트워크를 병목 링크를 갖는 단일 홉으로 축소하여 모델링 하고 여기에 일정한 초기 간격(initial gap)을 갖는 패킷 쌍이나 패킷 트레인 등을 전송했을 때 수신 측에서 관찰되는 출력 간격(output gap)의 변화를 측정하여 가용대역폭을 측정하는 기법이다. Delphi는 Multifractal Wavelet Modal (MWM)에 기반하여 경쟁 트래픽의 양을 측정하는 기법이다. IGI/PTR은 네트워크를 처음으로 단일-홉 갭 모델을 사용하였고, 패킷 쌍을 전송했을 때 입력 간격과 경쟁 트래픽에 의해서 바뀌는 출력 간격의 관계를 분석하여 가용대역폭을 측정하는 기법이다. Spruce는 여러 개의 패킷 쌍 간의 간격이 포아송 분포를 갖도록 하여 경쟁트래픽의 랜덤한 표본을 추출하여 가용대역폭을 측정하는 기법이다.

이런 기법들은 적은 수의 패킷을 이용해서 빠르게 가용대역폭을 찾을 수 있다는 장점이 있다. 따라서 네트워크의 상황이 변화하더라도 가용대역폭을 빠르게 측정할 수 있다. 또한 측정 패킷이 네트워크에 부하를 적게 주는 장점이 있다. 하지만 초기에 측정된 병목링크의 고정대역폭을 이용하여 가용대역폭을 측정하기 때문에 병목링크의 고정대역폭을 정확하게 측정하지 못하면 예러가 크게 발생하는 단점이 있다. 또한 혼잡한 링크가 병목링크가 아닌 경우 초기 측정된 병목링크의 고정대역폭의 오차에 의한 가용대역폭 측정에 오차가 생길 수 있는 문제[6]가 있다. 이런 문제는 혼잡 링크(tight link)와 좁은 링크(narrow link)가 경로 상에서 서로 다른 링크일 경우에 발생하게 된다.

자가 혼잡유발 기법들로는 대표적으로 Train of Packet Pair(TOPP)[11], pathChirp[12], Pathload[13] 등이 있다. 이와 같은 자가 혼잡유발 기법들은 전송하는 패킷의 전송속도를 높여가면서 전송하는 패킷의 속도보다 수신 측에서 관찰되는 속도가 큐잉에 의해서 작아지는 순간을 관찰하여 가용대역폭을 측정하는 기법이다. TOPP는 네트워크에 일시적으로 많은 부하를 주지 않기 위해서 일정한 간격을 갖는 여러 개의 패킷 쌍을 사용한다. 패킷 쌍의 초기 간격을 점차 감소시키면서 패킷 쌍의 트레인을 여러 번 전송을 통해서 얻어진 출력 간격

의 변화를 측정하여 혼잡이 발생하는 시점을 통해서 가용대역폭을 측정하게 된다. pathChirp는 하나의 패킷 트레인 안에서 초기 간격을 일정한 비율로 감소시키고 이를 패킷 처프(packet chirp)라고 정의한다. 이런 패킷 처프를 여러 번 전송하여 패킷 처프 안에서 출력 간격이 급격히 증가하기 시작하는 패킷 쌍을 찾고 이 때의 전송률이 네트워크 경로 상에 혼잡을 유발하는 것으로 판단해서 가용대역폭을 찾는 방법이다. Pathload는 송신 측에서 전송하는 속도가 가용대역폭보다 큰 경우에 라우터의 대기열에서의 지연이 증가하여 패킷 트레인 전체의 지연이 전체적으로 증가하는 점에 착안하여 초기 전송률을 이진 트리 검색과 비슷한 방식으로 변화 시키면서 패킷 트레인의 지연이 전체적으로 증가하기 시작하는 전송률을 찾아내는 기법이다.

이런 기법들은 경로 상 병목의 위치와 무관하게 가용대역폭을 찾을 수 있는 장점이 있다. 즉 혼잡 링크와 좁은 링크가 서로 다른 경우에도 혼잡 링크에서 측정 패킷들이 겪게 되는 지연을 통해서 가용대역폭을 측정하기 때문에 병목의 위치와 무관하게 가용대역폭을 찾을 수 있다. 하지만 혼잡을 유발하기 때문에 측정 패킷 자체가 네트워크에 많은 부하를 주고 측정하는데 많은 시간이 걸리는 단점이 있다.

그 밖에 경로 상의 병목 링크의 위치를 찾기 위한 BFind[14], Pathchar[15], Pathneck[16]와 같은 기법들도 연구되고 있다.

지금까지 다른 대역폭 측정 관련 연구들을 분류하면 그림 1과 같다.

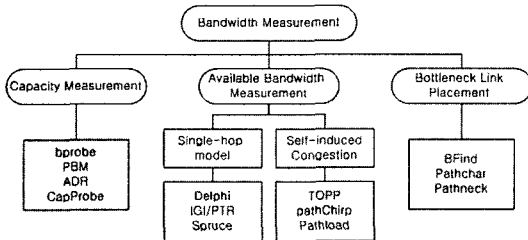


그림 1 대역폭 측정 기법의 분류

### 3. 단일-홉 갭 모델과 패킷 트레인의 인자에 따른 효과

#### 3.1 단일-홉 갭 모델

본 논문에서 제안하는 방법은 단일-홉 갭 모델[9]에 의한 분석에 기반하고 있다. 따라서 이 장에서는 단일-홉 갭 모델의 특성을 살펴본다. 단일-홉 갭 모델은 패킷 쌍이 전달될 때 병목 링크에 의해서 간격이 크게 영향을 받는다는 점에 착안하여, 병목 링크를 갖는 네트워

크의 경로를 병목 링크 하나로 축소하여 단순한 모델로 만들었다. 이 때 측정 시 패킷 쌍의 두 개의 패킷이 전송되는 동안에는 라우팅 경로가 변경되지 않아서 병목 링크는 고정되고 측정하고자 하는 경쟁 트래픽의 양은 일정한 것으로 가정한다.

이 모델에서는 패킷 쌍의 출력 간격은 초기 간격과 경쟁 트래픽의 양에 의해 변하는 성질을 이용한다. 패킷 쌍의 출력 간격이 초기 간격에 비해서 변화가 없는 경우는 경쟁 트래픽의 영향을 받지 않은 경우이므로 가용대역폭 이하의 속도로 전송되고 있음을 의미한다. 이 모델에서 패킷 쌍의 출력 간격이 초기 간격에 비해서 증가하는 경우는 라우터의 대기열에서 패킷 쌍 사이에 경쟁 트래픽이 들어와서 출력 간격에 영향을 준 경우이다. 이 때 초기 간격  $g_i$ 에 대해 증가한 출력 간격  $g_o$ 는 다음과 같다.

$$g_o = \frac{p + B_i g_i}{C} \tag{5}$$

여기서  $p$ 는 측정 패킷의 길이,  $B_i$ 는 병목 링크의 경쟁 트래픽의 전송률,  $C$ 는 병목 링크의 고정대역폭을 의미한다. 여기서 초기 간격에 비해서 증가한 출력 간격은 가정한 바와 같이 병목 링크의 고정대역폭이 일정하고 경쟁 트래픽의 양이 측정하는 동안 일정하다면 초기 간격 동안 라우터의 대기열에 들어온 경쟁 트래픽의 양을 반영하게 된다.

반면 패킷 쌍의 출력 간격이 초기 간격에 비해서 감소하는 경우는 라우터의 대기열에 있던 경쟁 트래픽의 패킷과 첫 번째 측정 패킷  $P_1$ 이 처리된 이후 경쟁 트래픽이 들어오지 않으면 두 번째 패킷  $P_2$ 가 대기열에서 지연을 겪지 않고 바로 처리되게 되고, 이 때 패킷 쌍의 출력 간격은 초기 간격에 비해 감소하게 된다. 즉, 출력 간격은 첫 번째 측정 패킷  $P_1$ 이 라우터의 대기열에서 겪는 지연만큼 감소하게 된다. 이 때 패킷 쌍을 통한 측정을 통해서 경쟁 트래픽의 양을 측정하지 못하게 된다. 이 때 출력 간격  $g_o$ 와 초기 간격  $g_i$ 는 첫 번째 패킷  $P_1$ 이 도착했을 때 라우터의 대기열에 있는 경쟁 트래픽의 패킷이  $Q$ 만큼 있을 때 다음 식 (6)과 같은 관계를 갖는다.

$$g_o = g_i - \frac{Q}{C} \tag{6}$$

식 (5)에서와 같이 수신 측에서 관찰되는 패킷 쌍의 증가된 출력 간격  $g_o$ 는 초기 간격  $g_i$ 가 고정되어 있을 때 병목 링크의 고정대역폭에 대한 경쟁 트래픽의 양에 비례해서 증가하는 특징을 갖는 것을 알 수 있다. 하지만 초기 간격이 작아서 패킷 쌍의 전송속도가 가용대역폭보다 상대적으로 높게 설정된 경우에는 수신 측에서

관찰되는 출력 간격에 경쟁 트래픽의 양이 충분히 반영되지 못한다. 반면 초기 간격이 증가하여 출력 간격에 가까워 질수록 경쟁 트래픽이 충분히 반영된다. 하지만 초기 간격을 계속 증가시켜서 패킷 쌍의 전송 속도가 가용대역폭보다 낮게 전송하는 경우에는 패킷 쌍의 출력 간격이 초기 간격에 비해서 감소할 수 있다. 따라서 초기 간격과 출력 간격이 같아지는 초기 간격을 찾았을 때 수신 측에서 출력 간격을 관찰함으로써 경쟁 트래픽의 양과 가용대역폭을 측정할 수 있게 된다. 즉, 경로 상의 병목 링크의 고정대역폭을 측정할 수 있다면, 패킷 쌍의 초기 간격에 대해서 측정된 출력 간격을 통해서 식(5)로부터 경쟁 트래픽의 양을 측정할 수 있다. 또한 식 (3), (4)를 이용하면 측정된 병목 링크의 고정대역폭과 경쟁 트래픽의 양으로부터 경로 상의 가용대역폭도 측정할 수 있다. 여기서 병목 링크의 고정대역폭과 가용대역폭을 측정하는 방법은 3.2절에서 설명하도록 한다.

**3.2 패킷 쌍의 초기 간격에 따른 고정대역폭 및 가용대역폭 측정**

단일-홉 겹 모델에서 패킷 쌍의 초기 간격은 링크의 고정대역폭이나 가용대역폭을 측정하는데 중요한 요소이다. 먼저 패킷 쌍을 연이어 전송하게 되면, 단일-홉 겹 모델에서 병목 링크의 고정대역폭을 구할 수 있다. 만약 패킷 쌍의 간격이 아주 작은 - 현실적으로 운영체제의 커널에서 패킷 간의 간격을 0으로 만들어서 전송할 수 없으므로 - 연이은(back-to-back) 패킷 쌍을 전송하는 경우, 라우터의 대기열에 두 개의 패킷은 연속적으로 도착하게 된다. 따라서 출력 간격은 첫 번째 패킷이 처리되는 데 걸리는 시간이 되며 이는 측정 패킷의 크기와 병목 링크의 대역폭에 의해서 결정된다. 즉, 크기  $p$ 이고 초기 간격  $g_i$ 가 0인  $N$ 개의 패킷 쌍을 전송하여 측정되는 출력 간격  $g^k_o$  ( $k = 1, \dots, N$ )을 통해서 병목 링크의 고정대역폭  $C$ 를 다음과 같은 식 (7)을 통해서 측정할 수 있다.

$$C = \frac{N \times p}{\sum_{k=1}^N g^k_o} \tag{7}$$

따라서 조사 패킷의 크기와 출력 간격을 통해서 병목 링크의 고정대역폭을 측정할 수 있다. 이와 같은 원리를 이용하여 IGI/PTR, Spruce 등의 기법들은 초기 간격  $g_i$ 가 아주 작은 연이은 패킷 쌍을 전송함으로써 경로 상 병목 링크의 고정대역폭을 측정하게 된다.

반면 패킷 쌍의 초기 간격을 증가시키면, 라우터의 대기열에서 경쟁 트래픽과 측정 패킷은 서로 경쟁하게 되어 출력 간격은 초기 간격에 비해 증가하게 된다. 하지만 초기 간격이 증가함에 따라서, 경쟁 트래픽의 양이 출력

간격에 미치는 영향이 점차 감소되고 출력 간격과 초기 간격 간의 차이는 감소하게 된다. 이는 초기 간격  $g_i$ 가 증가함에 따라서 점차 패킷 쌍이 라우터의 대기열에 도착하는 속도가 가용대역폭과 비슷한 속도가 되므로, 패킷 쌍의 첫 번째 패킷 이후에 도착하는 경쟁 트래픽에 의해서 두 번째 패킷이 라우터의 대기열에서 걸게 되는 상대적인 지연이 점차 감소하고 출력 간격  $g_o$ 와 초기 간격  $g_i$  간의 차이 역시 점차 감소하게 된다.

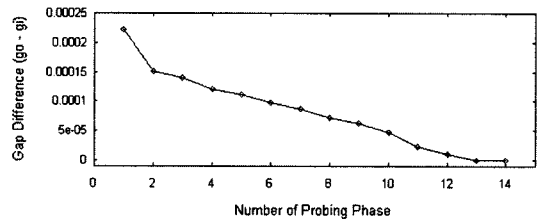


그림 2  $g_i$ 의 변화에 따른  $g_o$ 와  $g_i$ 의 차이의 변화

그림 2는 75%의 경쟁 트래픽이 있는 단일 홉 링크에서 패킷 쌍 20개씩 전송하여 증가하는 초기 간격에 대하여 평균 출력 간격  $g_o$ 와 초기 간격  $g_i$  간 차이의 관계를 도식한 그림이다. 초기 간격  $g_i$ 가 0인 연이은 패킷 쌍을 전송한 이후 패킷 쌍의 초기 간격  $g_i$ 를 0.00002 초씩 증가시키기에 따라  $g_o$ 와  $g_i$ 간의 차이가 변화하는 것을 측정하여 도식하였다. 그림 2에서 맨 첫 샘플은 연이은 패킷 쌍을 전송한 경우에 해당된다. 첫 번째 샘플을 제외하고는 전반적으로 초기 간격이 증가함에 따라서 출력 간격과 초기 간격의 차이가 선형적으로 감소하는 것을 알 수 있다. 또한 초기 간격에 의한 패킷 쌍의 전송 속도가 가용대역폭과 같아지는 경우에는 출력 간격과 초기 간격 간의 차이가 0이 된다. 이 때 관찰되는 패킷 쌍에는 현재 링크 상에 존재하는 경쟁 트래픽의 양이 가장 잘 반영된 것으로 판단할 수 있다. 따라서 출력 간격과 수신 간격의 차이가 0이 되는 샘플을 이용하여 경쟁 트래픽의 양을 측정할 수 있다. 앞서 연이은 패킷 쌍의 전송을 통해 얻은 샘플을 이용하여 식 (7)에서 측정된 링크의 고정대역폭과 경쟁 트래픽의 양을 이용하여 식 (3), (4)로부터 경로 상의 가용대역폭을 측정할 수 있다.

**3.3 측정 패킷: 패킷 쌍과 패킷 트레인**

단일-홉 겹 모델에서는 패킷 쌍의 초기 간격과 출력 간격의 변화를 통해서 경쟁 트래픽의 양을 측정할 수 있는 모델을 제시하였다. 이 때 경쟁 트래픽의 양을 측정하기 위한 충분한 샘플을 얻기 위해서는 여러 개의 패킷 쌍을 전송해야 한다. 이와 같이 여러 개의 샘플을 얻기 위한 방법으로 다양한 형태의 측정 패킷 형태들이 제시되었다.

측정 패킷의 형태로는 크게 패킷 쌍과 패킷 트레인 두 가지로 나눌 수 있다. 패킷 쌍은 길이가 같은 두 개의 패킷을 초기 간격을 설정하여 전송하는 방식이다. 3.1절에서 설명한 바와 같이 패킷 쌍이 도착하는 시점에 따라서 경쟁 트래픽의 영향을 받아서 초기 간격이 증가하기도 하고 감소하기도 한다. 특히 라우터의 대기열에 쌓인 패킷의 숫자가 급격히 변화하는 경우에는 도착하는 시점에 따라서 경쟁 트래픽을 잘 반영하지 못할 수 있다. 따라서 샘플 간의 오차가 심하고 적절한 샘플을 얻기 위해서는 많은 패킷 쌍을 전송해야 하는 문제가 있다. 이와 같은 문제를 해결하기 위해서 특정한 분포를 갖는 간격을 두고 여러개의 패킷 쌍을 전송하는 Train of Packet Pair(ToPP)[11], Poisson-distributed Packet Pair(PoPP)[10]와 같은 형태를 사용하고, 특히 PoPP는 경쟁 트래픽의 임의의 샘플을 측정하는데 유리하다. 하지만 적절한 샘플을 얻는데 필요한 측정 패킷의 양이 많이 필요하고 측정 시간이 길어지는 단점이 있다.

반면 패킷 트레인은 두 개 이상의 패킷의 모든 쌍이 일정한 초기 간격을 갖도록 설정하여 전송하는 방식이다. 각각의 패킷은 앞 뒤 패킷과 패킷 쌍을 이루게 되어 패킷 트레인에 N개의 패킷을 포함하는 경우 모두 N-1개의 패킷 쌍을 이루게 된다. 하지만 패킷 트레인은 내부의 연이은 모든 패킷 쌍이 샘플이 되기 때문에 경쟁 트래픽이 영향을 주는 위치에 따라서 앞뒤 패킷쌍에 영향을 주게 된다[17]. 따라서 초기 간격이 작고 경쟁 트래픽의 양이 많다면 패킷 트레인 내부의 한 개의 패킷이 영향을 받은 경우 뒤에 이어오는 패킷 간의 간격이 영향을 받게 되는 문제가 있다.

#### 3.4 패킷 트레인과 경쟁 트래픽 간의 관계

3.1절에서 설명한 단일-홉 겹 모델에서는 네트워크 상의 경로를 병목 링크를 갖는 단일홉으로 단순화했다. 하지만, 실제 다중 홉으로 이루어진 네트워크에서 측정 패킷을 전송한 경우 측정된 출력 간격과 모델 사이에는 오차가 발생하게 된다. 또한 패킷 트레인은 단순히 패킷 쌍의 모음이 아니기 때문에 경쟁 트래픽의 형태에 따라서 패킷 트레인이 포함하고 있는 패킷 쌍의 출력 간격에 미치는 영향이 서로 다르다. 따라서 이번 절에서는 이런 특성을 고정대역폭 측정의 부정확성, 혼잡 링크와 좁은 링크 차이에 따른 문제, 패킷 트레인과 경쟁 트래픽의 형태에 따른 패턴으로 나누어 그 관계를 분석한다.

먼저 실제 네트워크는 경로 상에 여러개의 링크가 존재하기 때문에 병목 링크를 지난 이후에도 다른 링크의 라우터에서 패킷 트레인 사이에 경쟁 트래픽의 영향을 받을 수 있기 때문에 패킷 트레인의 출력 간격이 변하게 된다. 특히 연이은 패킷 트레인에서 병목 링크의 고정대역폭을 반영한 출력 간격이 이후의 링크에서 경쟁

트래픽의 패킷으로 인해 증가하는 경우에는 병목 링크의 고정대역폭이 낮게 측정된다. 이와 같이 낮게 측정된 고정대역폭은 식 (5)를 이용하여 경쟁 트래픽의 양을 측정할 때 경쟁 트래픽의 양을 상대적으로 낮게 측정하는데 영향을 준다. 따라서 낮게 측정된 고정대역폭은 가용대역폭이 높게 측정하는 문제를 유발한다.

또한 경로 상에 혼잡 링크(tight link)가 좁은 링크(narrow link)와 다를 수 있기 때문에[6] 단일 병목링크를 가정한 단일-홉 겹 모델에서는 좁은 링크의 고정대역폭을 이용하여 가용대역폭을 측정하는 경우 오차가 발생할 수 있다. 따라서 IGI와 같이 가용대역폭 측정을 위해 연이은 패킷을 전송하여 병목 링크의 고정대역폭을 측정하는 기법에서는 가용대역폭을 높게 측정하고 네트워크 경로의 상황이 변하여 가용대역폭이 급격히 낮아지는 경우 변화된 가용대역폭을 측정하지 못하는 문제[10]가 발생할 수 있다.

그리고 패킷 트레인은 각각의 패킷이 앞 뒤 패킷과 패킷 쌍을 이루는 것으로 가정하지만, 실제로는 경쟁 트래픽이 영향을 미치는 위치와 경쟁 트래픽의 패킷 크기에 따라서 패킷 쌍의 집합과는 다른 특징을 갖는다. 특히 패킷 트레인의 패킷 쌍 샘플 중에는 앞의 패킷 쌍 사이에 경쟁 트래픽의 패킷이 들어오는 경우 해당 패킷 쌍의 출력간격은 증가시키지만 경쟁 트래픽의 패킷에 의한 라우터 대기열에서의 지연이 다음 패킷 쌍의 출력간격을 감소시키는 경우가 발생할 수 있다. 이런 형태를 경쟁 트래픽에 의한 거울 패턴(mirror pattern)이라고 한다[17]. 또한 경쟁 트래픽의 패킷이 겪는 라우터 대기열에서의 지연이 패킷 트레인의 여러 개의 패킷 쌍에 연속적으로 영향을 주는 경우 패킷쌍의 출력 간격이 감소하는 효과를 겪게 되는데, 이런 형태를 체인 패턴(chain pattern)이라고 한다[17]. 그 밖에도 경쟁 트래픽의 여러 개의 패킷이 여러 개의 패킷쌍 사이에 들어오므로써 패킷 쌍의 출력 간격이 왜곡되는 현상이 나타날 수 있는데, 이런 형태를 수량 패턴(quantification pattern)이라고 한다[17]. 위와 같은 현상들은 패킷 트레인의 초기 간격이 작아서 전송률이 링크의 고정대역폭에 비해서 높은 경우에 많이 발생하고 특히 감소된 패킷 쌍의 샘플이 많이 나타나는 것을 관찰할 수 있다[17]. 따라서 패킷 트레인에서 출력 간격이 감소한 패킷 쌍을 모두 포함하여 평균적인 전송률을 이용하는 PTR과 같은 기법은 네트워크 링크 상의 가용대역폭이 높은 경우에 측정값의 오차가 커진다.

#### 4. 대역폭 측정 및 빠른 수렴 기법

이 장에서는 패킷 트레인의 초기 간격에 영향을 받지 않고 고정대역폭과 경쟁 트래픽의 양을 측정할 수 있는

DCIG 알고리즘을 제안한다. 특히 충분히 경쟁 트래픽이 반영된 샘플을 통해서 고정대역폭과 경쟁 트래픽을 동시에 얻을 수 있는 방법을 제시한다. 이를 통해서 연이은 패킷 트레인 전송을 통해 측정된 병목 링크의 고정대역폭으로 인하여 발생하는 가용대역폭 측정의 오차를 줄일 수 있다. 그리고 패킷 트레인이 경쟁 트래픽에 의해서 영향을 받은 출력 간격이 초기 간격에 비해 증가한 패킷 쌍 샘플만을 이용함으로써 패킷 트레인이 포함하고 있는 잘못된 샘플을 제외시킬 수 있다. 또한 패킷 쌍의 초기간격에 대한 출력 간격과 초기 간격 간의 차이의 변화 특성이 병목 링크의 고정대역폭과 경쟁 트래픽의 양에 따라서 선형적으로 변화한다는 점에 기반하여 가용대역폭에 가까운 초기 간격을 빠르게 찾는 FC 기법을 제시한다. 이를 통해서 가용대역폭을 빠르면서도 정확하게 측정할 수 있다.

4.1 DCIG(Decoupled Capacity estimation with the Initial Gap of the packet train)

3.1절에서 살펴본 바와 같이 패킷 트레인의 초기 간격을 증가시키면 경쟁 트래픽의 양에 의해서 출력 간격이 점차 감소하게 된다. 특히 초기 간격을 증가시키면서 패킷 트레인을 전송하다가 초기 간격과 출력 간격이 같게 되면 측정된 출력 간격을 통해서 경쟁 트래픽의 양을 측정하게 된다. 이 때 혼잡 링크의 고정대역폭에 의해서 패킷 트레인의 지연이 결정되므로, 식 (5)에서 패킷 트레인의 출력 간격에 반영된 고정대역폭  $C$ 와 경쟁 트래픽의 양  $B_i$ 는 혼잡 링크의 상황을 반영하게 된다. 따라서 패킷 트레인의 초기 간격이 가용대역폭에 해당하는 속도로 패킷 트레인을 전송하는 샘플, 즉 경쟁 트래픽의 양이 패킷 트레인의 출력 간격에 반영된 샘플을 이용하면, 자가 혼잡유발과 같은 방법을 사용하지 않고도 고정대역폭 측정 오차에 따른 가용대역폭 측정 시 발생하는 오차를 해결할 수 있다.

패킷 트레인을 전송하여 링크의 고정대역폭을 측정하기 위해서 가용대역폭과 가까운 초기 간격을 갖는 두 개의 패킷 트레인을 전송하여 얻은 출력간격의 샘플  $(g_i^1, g_o^1), (g_i^2, g_o^2)$ 를 통해서 수식 (5)에서 병목 링크의 고정대역폭과 경쟁 트래픽의 양을 구할 수 있다. 수식 (5)에서 패킷 트레인을 전송할 때 송신 측에서 정하는 패킷 길이  $p$ 와 초기 간격  $g_i$ 과 출력 간격  $g_o$ 를 제외한 두개의 변수인 병목 링크의 고정대역폭  $C$ 와 병목 링크에 존재하는 경쟁 트래픽의 양  $B_i$ 를 얻기 위해방정식을 풀면 수식 (8)과 같다.

$$C = \frac{P(g_i^1 - g_i^2)}{g_i^1 g_o^2 - g_i^2 g_o^1}, \quad B_i = \frac{P(g_o^1 - g_o^2)}{g_i^1 g_o^2 - g_i^2 g_o^1} \quad (8)$$

이를 통해서 수식 (3)과 같이 패킷 트레인의 길이에

해당되는 시간  $T$  동안에 해당되는 링크  $L$ 의 고정대역폭  $C$ 와 가용대역폭  $A_L = C - B_i$ 를 얻을 수 있다.

4.2 FC(Fast Converging) Technique

그림 2와 같이 연이은(back-to-back) 패킷 트레인을 전송한 첫 번째 경우를 제외하고는 패킷 쌍의 초기 간격에 대해서 출력 간격과 초기 간격의 차이는 선형적으로 변화한다. 이런 특성은 식 (5)에서 경쟁 트래픽의 양이 패킷 트레인을 전송하는 동안 일정하다고 가정하면 출력 간격이 초기 간격에 따라서 선형적으로 변화한다는 것을 통해서도 알 수 있다. 따라서 이를 이용하여 좀더 빠르게 출력 간격과 초기 간격이 같아지는 지점을 찾을 수 있다. 초기 간격에 따른 출력 간격과 초기 간격의 차이 간의 관계를 방정식으로 나타내면 수식 (9)과 같다.

$$g_o - g_i = \frac{g_{diff}^1 - g_{diff}^2}{g_i^1 - g_i^2} g_i + K \quad (9)$$

이 때,  $g_{diff}^1$ 와  $g_{diff}^2$ 는 각각  $g_o^1 - g_i^1, g_o^2 - g_i^2$ 으로 첫 번째 샘플과 두 번째 샘플의 출력 간격과 초기 간격 간의 차이이고, 상수  $K$ 는 연이은 패킷 트레인을 전송했을 때 얻은 출력간격과 초기 간격 간의 차이이다. 수식 (9)를 통해서 초기에 구한 두 개의 샘플을 이용하여  $g_o - g_i$ 가 0이 되는  $g_i$ 를 빠르게 예측할 수 있다. 여기서 구한 값에서 가까운 두 개의 초기 간격  $g_i$ 로부터 샘플 두 개를 얻어서 DCIG 알고리즘을 사용하여 고정대역폭과 가용대역폭을 측정한다.

FC 기법을 IGI/PTR에서 사용하는 점진적으로 수렴하는 기법과 비교하면 그림 3과 같다. IGI/PTR은 초기 간격을 얻기 위해 연이은 패킷 트레인 전송을 통해서 측정된 출력 간격의 1/2을 하한으로 하고 출력 간격을 처음 얻은 출력 간격의 1/8씩 증가시키면서 초기 간격과 출력 간격이 같은 초기 간격을 검색한다. 이러한 IGI/PTR이 일반적으로 가장 적은 측정패킷을 사용하는 방법으로 알려져 있다[9,10]. 하지만 FC 알고리즘을 이용하면 초기 간격과 출력 간격이 같은 초기 간격을 더

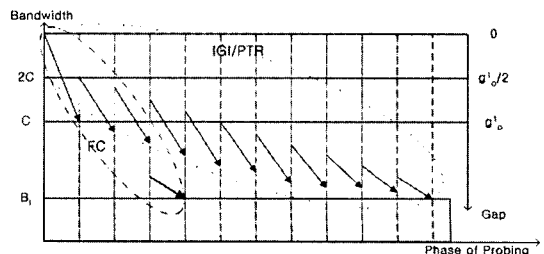


그림 3 패킷 트레인의 간격 변화에 따른 경쟁 트래픽의 양 측정과정

빠르게 찾을 수 있다.

출력 간격과 초기 간격이 같아지는 초기 간격을 찾는 과정은 그림 3과 같다. 그림 3에서 화살표는 패킷 트레인이 전송되는 사건을 의미하며, 화살표의 시작점은 패킷 트레인의 초기 간격과 이에 해당되는 패킷 트레인의 전송률을 의미하고 화살표의 끝점은 패킷 트레인이 수신된 출력 간격과 관찰된 패킷 트레인의 전송속도를 의미한다. 병목 링크의 고정대역폭은  $C$ 이고 암영 부분은 경쟁 트래픽의 양  $B_i$ 를 의미한다. IGL/PTR의 경우에는 초기 값  $g_i$ 를 점차 증가시키면서 선형적으로 가용대역폭에 해당되는 적절한 초기 간격을 찾아간다. 이에 반해서 제안하는 FC 알고리즘은 처음 전송한 연이은 패킷 트레인을 통해서 초기 값을 구하고 이 샘플을 통해서 가용대역폭 측정을 위해서 패킷 트레인에 경쟁 트래픽의 양을 반영할 수 있는 초기 간격을 찾게 된다.

### 5. 알고리즘

패킷 트레인 전송을 통해 DCIG 알고리즘과 FC 기법을 이용하여 가용대역폭을 측정하는 과정은 다음과 같다. 먼저 초기 간격의 고정대역폭에 의한 최대값을 측정하기 위해서 연이은 패킷 트레인을 전송한다. 이를 통해 측정된 출력 간격의 절반을 첫 번째 측정 패킷트레인의 초기 간격  $init\_g_i$ 로 설정한다. 그리고 증가되는 초기 간격  $gap\_search\_diff$ 는 초기 출력 간격의 1/8로 한다. 연이은 패킷 트레인 전송을 통해서 얻은 고정대역폭이 부정확할 수 있는 점을 고려하고 급격하게 패킷 트레인의 간격을 늘이지 않음으로써 초기 간격과 출력 간격이 같아지는 지점을 놓치지 않기 위해 위와 같이 정하였다. 또한 패킷 트레인의 전송한 회수는  $num\_probe$ 를 이용하여 매 패킷 트레인을 전송할 때마다 1씩 증가시킨다. 이와 같은 설정을 통해서 두개의 패킷 트레인을 전송하여 FC 알고리즘을 위한 두 개의 샘플을 얻는다. 이 때 얻어진 샘플을 이용하여 초기 간격과 출력 간격이 같아지는 초기 간격을 예측하는 FC 알고리즘  $estimate\_init\_gap$ 을 적용한다. 그리고 예측된 초기 간격을 이용하여 패킷 트레인을 전송한다. 이 때 초기 간격과 출력 간격을 비교하여 그 크기가 같으면 측정 패킷의 전송을 중단하고 DCIG 알고리즘  $cal\_avail\_bw$ 를 적용하여 가용대역폭을 측정한다. 이와 같은 과정은 초기 간격과 출력 간격이 같아지는 시점까지 초기 간격을  $gap\_search\_diff$ 만큼씩 증가시키면서 계속해서 반복하게 된다. 이 때 초기 간격과 출력 간격을 비교하는 함수  $gap\_comp$ 에서는 두 값의 비가 10% 이내인 경우 같은 것으로 판단하였다. 위에서 제안한 DCIG와 FC를 적용한 의사코드는 그림 4와 같다.

```

double probing_available_bw() {
    /* initial back-to-back packet transmission */
    gi = 0;  init_go = go = send_packet(gi);
    num_probe = 1;
    prev_gi = gi;  gi = go/2;  gap_search_diff = go/8;

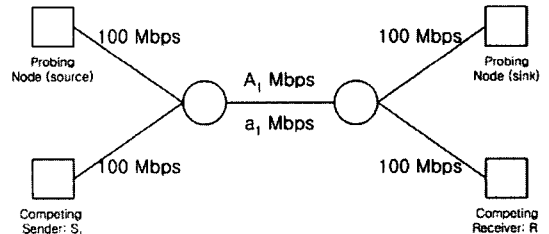
    while (1) {
        prev_go = go;  go = send_packet(gi);
        if (gap_comp(gi, go) != 0) break;
        prev_gi = gi;  gi += gap_search_diff;
        num_probe++;

        if (num_probe == 2) /* FC algorithm */
            gi = estimate_init_gap(init_go, prev_gi, prev_go - prev_gi, gi, go - gi);
    }
    return cal_avail_bw(prev_gi, prev_go, gi, go); /* DCIG algorithm */
}

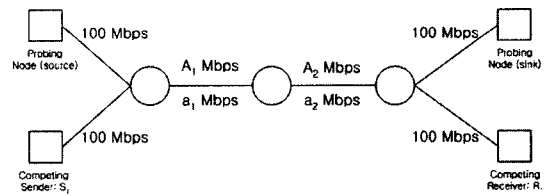
double estimate_init_gap(init_go, prev_gi, prev_diff, gi, diff) { /* FC algorithm */
    return -init_go / ((prev_diff - diff) / (prev_gi - gi));
}

double cal_avail_bw(prev_gi, prev_go, gi, go) { /* DCIG algorithm */
    C = p * (prev_gi - gi) / (prev_gi * go - gi * prev_go);
    B = p * (prev_go - go) / (prev_gi * go - gi * prev_go);
    return C - B;
}
    
```

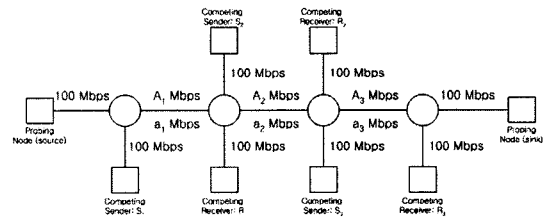
그림 4 DCIG와 FC를 적용한 의사코드



(a) 단일 홈 실험을 위한 망 구조



(b) 다중 홈 실험을 위한 망 구조



(c) 다중 홈 실험을 위한 망 구조  
그림 5 실험을 위한 망 구조

### 6. 실험결과

#### 6.1 실험 환경 및 망 구성

실험을 위해서 ns-2[19]를 사용하였다. 먼저 단일 홈 망 구조를 그림 5 (a)와 같이 구성하여 실험하였다. 병목 링크의 고정대역폭은  $A_1$  Mbps로 경쟁 트래픽의 양



을  $a_1$  Mbps로 정의하고 경쟁 트래픽의 양을 변화시키면서 기존 프로토콜과 제안하는 프로토콜의 성능을 비교하였다. 또한 FC 알고리즘을 적용한 경우와 IGI 기법과 같이 선형적으로 수렴하는 기법을 사용하는 경우에 대해서 가용대역폭 측정에 필요한 패킷 트레인의 개수를 비교하였다.

다중 홉에서의 DCIG 알고리즘과 기존의 다른 기법들의 성능을 비교하기 위해서 그림 5 (b), (c)와 같은 망 구조를 가지고 실험하였다. 여기서 경로 상에 혼잡 링크(tight link)와 좁은 링크(narrow link)가 따로 존재하는 경우를 고려하기 위해서 첫 번째 링크와 두 번째 링크의 고정대역폭을 서로 다르게 설정하고 각 링크에 존재하는 경쟁 트래픽의 양을 조절하면서 실험하였다. 또한 좀더 복잡한 망 상황에서의 성능을 비교하기 위해서 다양한 상황에 따른 실험을 하였고 기존의 다른 기법들의 수렴하는 속도를 비교하는 실험을 수행하였다. 여기서도 마찬가지로 첫 번째, 두 번째, 세 번째 링크의 고정대역폭을 각각  $A_1, A_2, A_3$  Mbps라고 정의하고, 각 링크에 경쟁 트래픽의 양을  $a_1, a_2, a_3$  Mbps로 정의하였다. 여기서 측정 노드(Probing Node)는 가용대역폭을 측정하기 위한 알고리즘을 이용하여 패킷 트레인을 전송하고 그에 대한 결과를 응답하는 노드이다. 그리고 경쟁 노드(Competing Node)는 링크에 로드를 증가시키기 위해서 경쟁 트래픽을 발생시키는 노드이다.

실험에서 패킷 트레인의 주요 설정값인 측정 패킷의 길이와 패킷 트레인의 길이에 대한 설정은 다음과 같다. 패킷 트레인에서 측정 패킷의 길이는 측정하고자 하는 목적에 따라 길이가 갖는 효과가 다르다. 측정 패킷의 길이가 길면 길수록 패킷 처리시간이 길어지기 때문에 경쟁 트래픽의 영향을 많이 받게 된다. 반면 길이가 짧으면 그 영향이 작아진다. 따라서 연이은 패킷을 전송하여 링크의 고정대역폭 측정하는 경우에는 측정 패킷의 길이가 짧을수록 경쟁 트래픽에 의한 오차를 줄일 수 있다. 하지만 가용대역폭을 측정할 때는 적절하게 경쟁 트래픽의 영향을 받는 것이 중요하다. 특히 인터넷 패킷은 40, 500, 1500 Bytes에 집중되어 있기 때문에 [18] 가용대역폭을 측정할 때는 중간 정도의 크기인 500~700 Bytes를 갖는 패킷 크기가 적절한 것으로 알려져 있다 [9]. 따라서 이 실험에서는 패킷 트레인의 측정 패킷의 길이를 700 Bytes로 정하였다.

또한 패킷 트레인의 길이에 따라서 원하는 시간 동안 경쟁 트래픽이 영향을 미치는 라우터의 대기열 상태를 반영할 수 있다. 패킷 트레인의 길이가 길면 가용대역폭을 측정할 때 측정하고자 하는 라우터의 큐 상태를 반영하는 시간을 늘일 수 있다. 즉, 식 (4)에서 T에 해당되는 시간을 늘일 수 있다. 하지만 긴 패킷 트레인은 빠

른 측정에 불리하고 네트워크 상황이 빠르게 변하는 경우는 적합하지 않다. 따라서 적절한 패킷 트레인 길이를 선택하는 것이 필요하다. 여기서는 측정을 위해서 사용되는 패킷 트레인에 사용한 패킷의 개수를 60개로 하였다. 패킷 트레인에 사용한 패킷의 개수는 IGI/PTR 논문에서 제시한 결과 [9]를 기반으로 여러 번의 실험을 통해서 측정값의 정확도가 높으면서도 측정에 걸리는 시간이 길지 않도록 정하였다.

6.2 단일 홉

실험은 두 가지로 나뉘어서 진행하였다. 먼저 FC 알고리즘을 통해서 측정 패킷 트레인을 효과적으로 줄일 수 있는지 확인하기 위해서 FC를 이용할 때와 IGI 알고리즘을 적용해 측정할 때 필요한 패킷 트레인의 개수를 비교하였다. 그 다음으로는 각각의 알고리즘이 가용대역폭을 측정하는데 미치는 영향을 비교하기 위해서 IGI/PTR 알고리즘과 DCIG, DCIG와 FC를 동시에 적용한 경우로 나뉘어서 비교하였다.

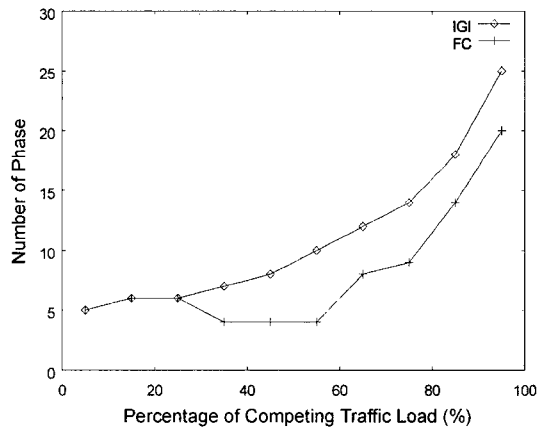


그림 6 경쟁 트래픽의 증가에 따른 패킷 트레인 전송 회수 비교

그림 6은 경쟁 트래픽의 양을 증가시키면서 가용대역폭을 측정하는데 필요한 패킷 트레인의 개수를 비교한 그래프이다. 전체적으로 경쟁 트래픽의 영향이 커짐에 따라서 출력 간격이 커지기 때문에 가용대역폭이 작은 경우에 비해서 가용대역폭이 큰 경우 필요한 패킷 트레인이 증가하는 것을 볼 수 있다. 또한 FC 알고리즘을 적용한 경우 초기 간격을 연이은 패킷 트레인을 전송해서 얻은 출력 값의 절반을 초기 간격으로 설정했기 때문에 가용대역폭이 50%인 경우를 중심으로 빠르게 수렴한다. 가용대역폭이 큰 경우에는 IGI나 FC 모두 경쟁 트래픽의 영향을 크게 받지 않기 때문에 출력 간격이 크게 증가하지 않아서 필요로 하는 패킷 트레인의 개수는 차이 나지 않는다. 하지만 가용대역폭이 작아질수록 출

력 간격과 초기 간격이 같아지는 초기 간격을 찾는 데 수렴하는 시간이 IGI나 FC가 유사하게 증가하지만 언제나 FC가 적은 수의 측정패킷 만으로도 적절한 초기 간격을 찾을 수 있음을 알 수 있다.

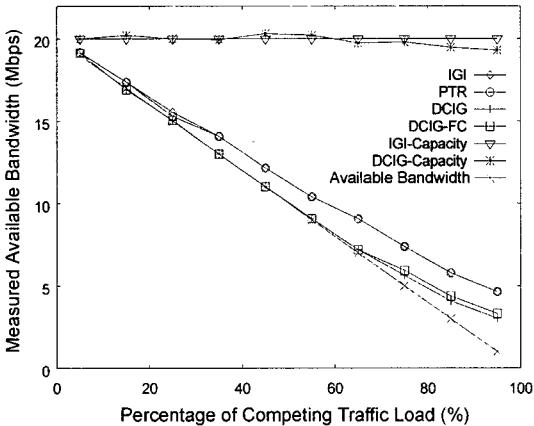


그림 7 단일 홉에서 경쟁 트래픽의 증가에 따른 가용대역폭 측정 비교

그림 7은 경쟁 트래픽의 양이 증가함에 따른 가용대역폭의 정확도를 IGI/PTR과 비교한 그래프이다. 단일 홉에서 IGI와 DCIG 알고리즘 상에서 측정하는 링크의 고정대역폭은 크게 차이가 나지 않는다. 하지만 IGI/PTR은 가용대역폭이 감소함에 따라서 전체적으로 가용대역폭을 높게 측정하는 것을 볼 수 있다. IGI의 경우는 경쟁 트래픽이 증가함에 따라 패킷 트레인의 초기 간격이 충분히 크더라도 출력 간격에 반영된 경쟁 트래픽이 많기 때문에 경쟁 트래픽의 양이 높게 측정되기 때문이다. 반면 DCIG만 적용한 경우에는 IGI/PTR에 비해서 정확하게 가용대역폭을 측정하는 것을 볼 수 있다. 좀 더 정확하게 가용대역폭을 측정하는 이유는 해당 샘플에 반영된 링크의 고정대역폭을 반영하여 가용대역폭을 측정하기 때문이다. 또한 DCIG 알고리즘에 FC 기법을 적용한 경우에는 빠르게 가용대역폭을 측정할 수 있지만 순차적인 샘플이 아닌 적은 샘플을 이용하기 때문에 DCIG만 적용한 경우에 비해서 좀 더 높게 측정하는 경향이 있다. 하지만 FC 기법을 적용한 경우에도 IGI에 비해서는 가용대역폭을 정확하게 측정하는 것을 볼 수 있다.

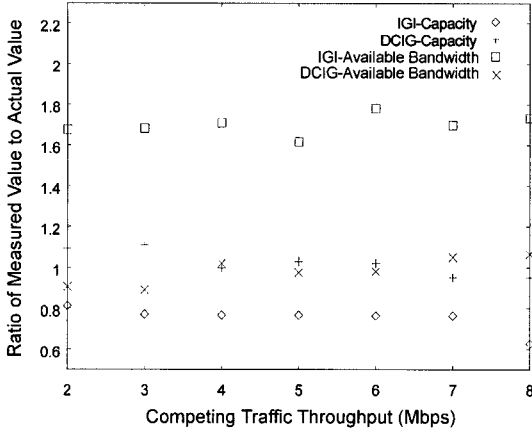
### 6.3 다중 홉

다중 홉 실험에서는 그림 6 (b), (c)와 같은 다중 홉의 실험 망에서 각각의 프로토콜이 갖는 두 가지 특성에 대해서 실험하였다. 먼저 혼잡 링크(tight link)와 좁은 링크(narrow link)가 따로 존재하는 경우 [6]의 가용

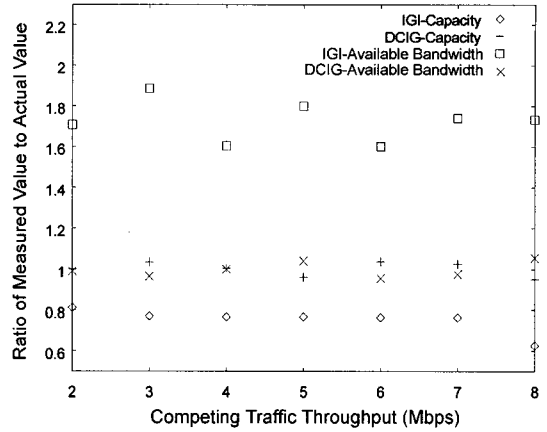
대역폭의 정확도에 대해서 실험하였다. 그림 6 (b)와 같은 망 구조에서 첫 번째 링크와 두 번째 링크의 고정대역폭이 서로 다른 경우에 대해서 IGI와 DCIG 기법들의 가용대역폭을 측정 정확도를 실험하여 비교하였다. 여기서 IGI 경우는 병목 링크의 고정대역폭을 측정하여 가용대역폭을 얻게 되는데, 이 실험에서는 각각의 기법에서 링크의 고정대역폭 측정의 정확도에 따른 가용대역폭 측정에 대한 효과를 살펴본다.

그리고 두 번째로 그림 6 (c)에서와 같은 다중 홉 경로 상에서 단일-홉 갭 모델에 기반한 IGI, DCIG 기법과 자가 혼잡 유발 기법 중 하나인 pathChirp 기법이 가용대역폭이 변함에 따라서 그 값에 얼마나 빠르게 수렴하여 측정하는지 비교하였다.

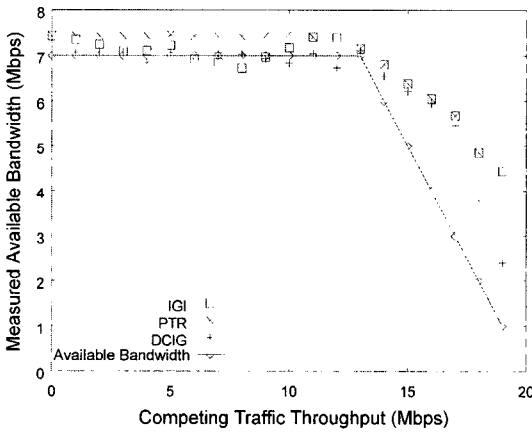
그림 8 (a), (b)는 다중홉 링크에서 IGI/PTR과 DCIG를 이용하여 고정대역폭과 가용대역폭을 측정된 값의 정확도를 비교하는 그래프이다. 그림 8 (a)는 첫 번째 링크고정대역폭  $A_1$ 를 10Mbps, 두 번째 링크고정대역폭  $A_2$ 를 20Mbps로 설정하고, 첫 번째 링크의 경쟁 트래픽의 양  $a_1$ 를 8Mbps로 고정시키고, 두 번째 링크의 경쟁 트래픽의 양  $a_2$ 를 2Mbps에서 8Mbps로 변화시키면서 IGI와 DCIG 알고리즘의 고정대역폭과 가용대역폭의 측정이 얼마나 정확한지를 분석한 그래프이다. 그림 8 (b)는 첫 번째 링크 고정대역폭  $A_1$ 를 20Mbps, 두 번째 링크 고정대역폭  $A_2$ 를 10Mbps로 설정하고, 두 번째 링크의 경쟁 트래픽의 양  $a_2$ 를 8Mbps로 고정시키고, 첫 번째 링크의 경쟁 트래픽의 양  $a_1$ 를 2Mbps에서 8Mbps로 변화시키면서 IGI와 DCIG 알고리즘의 고정대역폭과 가용대역폭의 측정이 얼마나 정확할지를 분석한 그래프이다. 두 실험에서 IGI는 10Mbps 링크에 대해서 약 80% 정도로 고정대역폭을 낮게 측정하는 반면에 DCIG 알고리즘의 경우에는 고정대역폭을 95%에서 105% 내외로 IGI보다 정확하게 측정함을 알 수 있다. 또한 가용대역폭에 있어서도 IGI는 160% 이상 높게 측정하는 문제를 보이는데 비해서, DCIG 알고리즘의 경우에는 가용대역폭을 92%에서 108% 범위 내에서 측정하는 것을 볼 수 있다. 이런 IGI에서의 가용대역폭의 과대 측정 현상이 나타나는 이유는 연이은 패킷 트레인을 전송하여 고정대역폭을 측정하는 경우, 병목 링크를 지난 패킷 트레인이 두 번째 링크를 지날 때 경쟁 트래픽에 의해 패킷 간격이 증가하여 고정대역폭을 낮게 측정하기 때문이다. 즉, 경쟁 트래픽의 양을 구할 때 낮게 측정된 고정대역폭을 이용하게 됨으로써 가용대역폭이 상대적으로 높게 측정함을 알 수 있다. 반면 DCIG 알고리즘의 경우에는 정확히 측정된 고정대역폭을 통해서 상대적으로 정확한 경쟁 트래픽의 양을 측정함으로써 가용대역폭도 실제 값에 가깝게 측정됨을 알 수 있다.



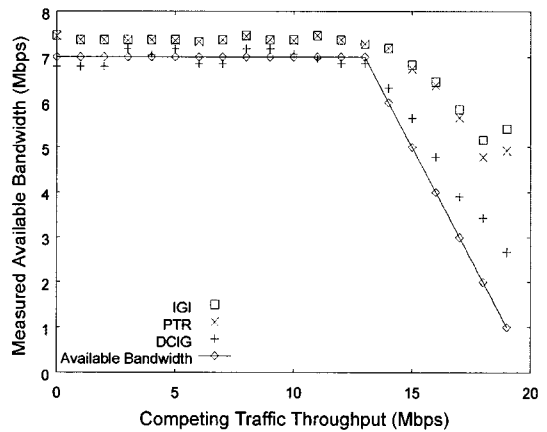
(a)  $A_1 = 10\text{Mbps}$ ,  $A_2 = 20\text{Mbps}$  ( $A_1 < A_2$ ),  $a_1 = 8\text{Mbps}$ ,  $a_2 = 2\text{--}8\text{Mbps}$



(b)  $A_1 = 20\text{Mbps}$ ,  $A_2 = 10\text{Mbps}$  ( $A_1 > A_2$ ),  $a_1 = 2\text{--}8\text{Mbps}$ ,  $a_2 = 8\text{Mbps}$



(c)  $A_1 = 10\text{Mbps}$ ,  $A_2 = 20\text{Mbps}$  ( $A_1 < A_2$ ),  $a_1 = 3\text{Mbps}$ ,  $a_2 = 0\text{--}19\text{Mbps}$



(d)  $A_1 = 20\text{Mbps}$ ,  $A_2 = 10\text{Mbps}$  ( $A_1 > A_2$ ),  $a_1 = 0\text{--}19\text{Mbps}$ ,  $a_2 = 3\text{Mbps}$

그림 8 IGI/PTR과 DCIG의 고정대역폭 및 가용대역폭 측정의 정확도 비교

그림 8 (c), (d)는 혼잡 링크(tight link)와 좁은 링크(narrow link)가 따로 존재하는 경우 IGI/PTR과 DCIG의 가용대역폭 측정의 정확도를 비교한 그래프이다. 그림 8 (c)는 첫 번째 링크 고정대역폭  $A_1$ 를 20Mbps, 두 번째 링크 고정대역폭  $A_2$ 를 10Mbps로 설정하고, 두 번째 링크의 경쟁 트래픽의 양  $a_2$ 를 3Mbps로 고정시키고, 첫 번째 링크의 경쟁 트래픽의 양  $a_1$ 를 0Mbps에서 19Mbps로 변화시키면서 IGI/PTR과 DCIG 알고리즘의 가용대역폭의 측정이 얼마나 정확한지를 분석한 그래프이다. 그림 8 (d)는 첫 번째 링크고정대역폭  $A_1$ 를 10Mbps, 두 번째 링크고정대역폭  $A_2$ 를 20Mbps로 설정하고, 첫 번째 링크의 경쟁 트래픽의 양  $a_1$ 를 3Mbps로 고정시키고, 두 번째 링크의 경쟁 트래픽의 양  $a_2$ 를 0Mbps에서 19Mbps로 변화시키면서 IGI와 DCIG 알고

리즘의 고정대역폭과 가용대역폭의 측정이 얼마나 정확한지를 분석한 그래프이다. 또한 두 실험 모두 고정대역폭이 20Mbps를 갖는 링크의 경쟁 트래픽의 양이 13Mbps를 넘는 시점에 고정대역폭이 10Mbps인 링크가 병목링크이던 상황에서 혼잡 링크(tight link)와 좁은 링크(narrow link)가 서로 달라지는 상태로 변하게 된다. 그림 8 (c)처럼 혼잡 링크가 좁은 링크보다 앞에 있는 경우에는 PTR 알고리즘은 가용대역폭을 과대 측정하고 IGI 알고리즘의 정확도는 떨어지는 것을 확인할 수 있다. 마찬가지로 그림 8 (d)처럼 좁은 링크가 혼잡 링크보다 앞에 있는 경우에는 IGI/PTR 알고리즘은 가용대역폭을 과대 측정하는 것을 볼 수 있다. 어떤 경우든 IGI/PTR 모두 혼잡 링크와 좁은 링크가 별도로 존재하는 경우에는 가용대역폭을 과다 측정하는 것을 볼

수 있다. 하지만 DCIG 알고리즘은 IGI/PTR에 비해서 가용대역폭을 더 정확히 측정하는 것을 알 수 있다.

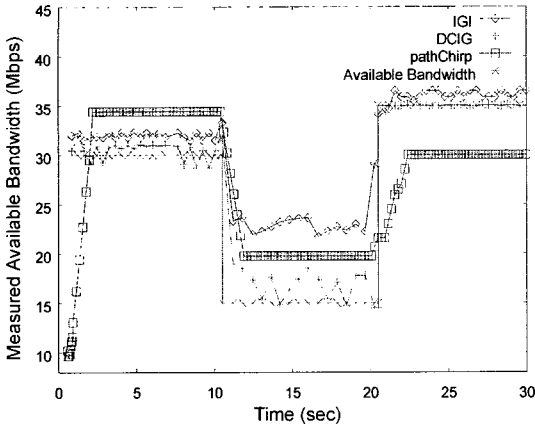


그림 9 가용대역폭 측정의 수렴 속도 비교

그림 9는 그림 6 (c)의 망 구조에서 각 링크 고정대역폭  $A_1, A_2, A_3$ 을 60Mbps 설정하고, 첫 번째 링크와 세 번째 링크의 경쟁 트래픽의 양  $a_1, a_3$ 를 25Mbps로 고정시키고, 두 번째 링크의 경쟁 트래픽의 양  $a_2$ 를 10.5, 20.5초에 30Mbps에서 45Mbps으로 45Mbps에서 25Mbps로 각각 변화시키면서 IGI, DCIG, pathChirp가 변화된 가용대역폭에 얼마나 빠르게 수렴하는지를 비교하기 위해서 그린 그래프이다. 그래프에서 IGI와 DCIG는 가용대역폭이 변하는 시점인 10.5초와 20.5초에서 네트워크의 상태 변화에 빠르게 적응하여 가용대역폭을 찾는 것을 알 수 있다. 하지만 pathChirp는 가용대역폭을 찾는데 약 2~4초 정도의 지연이 발생하는 것을 볼 수 있다. 또한 pathChirp는 패킷 처프(packet chirp)의 간격 비율, 패킷의 최저 전송 속도, 평균 전송 속도, 최대 전송 속도 등과 같은 초기값의 설정에 따라서 수렴 속도 및 측정의 정확도에 많은 영향을 받는다. 특히 자주 변화하는 네트워크 상황에 패킷 처프의 간격 비율 등에 따라서 잘못된 값으로 수렴하는 문제점을 발견할 수 있다. 이와 같은 문제는 회귀분석을 이용하는 자가 혼잡유발 방법들에서도 유사한 문제점을 발견할 수 있다[10]. 또한 IGI는 가용대역폭을 과다 측정하는 문제를 보여주고 있다. 하지만 DCIG 알고리즘은 IGI처럼 빠르게 수렴하면서도 동시에 가용대역폭을 정확하게 측정하는 장점을 가지고 있음을 실험 결과를 통해서 알 수 있다.

## 7. 결론

지금까지 인터넷에서 단단단으로 고정대역폭과 가용대역폭을 측정하는 기법에 대해서 살펴보았다. 특히 IGI

와 Spruce와 같은 단일-홉 겹 모델에 기반한 기법들이 가용대역폭을 측정할 때 이용하는 고정대역폭 측정값이 가용대역폭 측정에 미치는 영향에 대해서 분석하였다. 이와 같은 문제를 해결하기 위해서 두 개의 패킷 트레인 샘플을 이용하여 고정대역폭 측정과 가용대역폭 측정할 수 있는 새로운 기법인 DCIG 알고리즘을 제시하였다. 이를 통해서 경쟁 트래픽에 의해서 패킷 트레인의 초기 간격이 병목 링크의 고정대역폭의 측정에 미치는 오차에 따른 영향을 줄일 수 있다. 또한 측정에 필요한 패킷 트레인의 양을 줄여서 빠르게 고정대역폭과 가용대역폭을 측정함과 동시에 네트워크에 부하를 줄일 수 있는 FC 기법을 제안하였다. 패킷의 초기 간격에 대한 출력 간격과 초기 간격 간의 차이 간의 선형적 관계를 이용하여 경쟁 트래픽의 양을 적절히 반영하는 초기 간격을 빠르게 찾을 수 있다. 이와 같은 방법들을 통해서 다양한 네트워크 환경에서도 가용대역폭을 빠르게 측정할 수 있음을 모의실험을 통해서 제시하였다. 또한 모의 실험을 통해서 제안한 기법의 성능을 IGI/PTR과 비교하였고, 기존의 기법 중에서 IGI가 가지고 있는 가용대역폭을 과다 측정하게 되는 문제점과 자가 혼잡유발 기법 중의 하나인 pathChirp 기법이 가용대역폭이 변화하는 경우 수렴하는데 많은 시간이 걸리는 문제점을 제시하였다.

앞으로는 본 연구와 관련하여 구현을 통해서 실제 네트워크에서 다양한 망 구성 환경에 대한 가용대역폭 측정 기법들의 성능을 비교 분석할 예정이다.

## 참고 문헌

- [1] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and Waldbusser, S., "Version 3 of the Protocol Operations for the Simple Network Management Protocol (SNMP)," IETF RFC 3416.
- [2] Oetiker, T. Multi Router Traffic Grapher (MRTG), <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>
- [3] Oetiker, T. Round Robin Database (RRD), <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>
- [4] Carter, R. and Crovella, M., "Measuring Bottleneck Link Speed in Packet-switched Networks," Performance Evaluation, Vol. 27, No. 28, pp. 297-318, 1996.
- [5] Paxson, V., "Measurements and Analysis of End-to-end Internet Dynamics," PhD thesis, University of California, Berkeley, 1997.
- [6] Dovrolis, C., Ramanathan, P., Moore, D., "What Do Packet Dispersion Techniques Measure?," IEEE INFOCOM, pp. 905-914, 2001.
- [7] Kapoor, R., Chen, L., Lao, L., Gerla M., and Sanadidi, M., "CapProbe: A Simple and Accurate Capacity Estimation Technique," ACM SIGCOMM, 2004.

[8] Ribeiro, V., Coates, M., Riedi, R., Sarvotham, S., Hendricks, B. and Baraniuk, R., "Multifractal Cross-Traffic Estimation," In ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, 2000.

[9] Hu, N. and Steenkiste, P., "Evaluation and Characterization of Available Bandwidth Probing Techniques," IEEE Journal on Selected Areas in Communications, Vol. 21, No. 6, pp. 879-894, 2003.

[10] Strauss, J., Katabi, D., Kaashoek, F., "A Measurement Study of Available Bandwidth Estimation Tools." ACM SIGCOMM Internet Measurement Workshop, 2003.

[11] Melander, B., Bjorkman, M. and Gunningberg, P., "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks," IEEE Global Internet Symposium, 2000.

[12] Ribeiro, V., Riedi, R., Baraniuk, R., Navratil, J. and Cottrell, L., "pathChirp: Efficient Available Bandwidth Estimation for Network Paths," Passive and Active Measurements (PAM) workshop, 2003.

[13] Jain, M. and Dovrolis, C., "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput," ACM SIGCOMM, pp. 295-308, 2002.

[14] Akella, A., Seshan, S. and Shaikh, A. "An Empirical Evaluation of Wide-area Internet Bottlenecks," ACM IMC, 2003.

[15] Jacobson, V., "Pathchar-A tool to Infer Characteristics of Internet Paths," Presented as April 97 MSRI talk, 1997.

[16] Hu, N., Li, L., Mao, Z., Steenkiste, P. and Wang, J., "Locating Internet Bottlenecks: Algorithms, Measurements, and Implications," ACM SIGCOMM, 2004.

[17] Johnsson, A., Melander, B. and Björkman, M., "On the Analysis of Packet-Train Probing Schemes," CIC, 2004.

[18] Claffy, K., Miller, G. and Thompson, K., "The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone," presented at the ISOC INET Conf., 1998.

[19] ns-2 simulator, <http://www.isi.edu/nsnam/ns/>, 2005.



최 한

2003년 서울대학교 수학과 졸업(학사)  
2005년 서울대학교 대학원 전기·컴퓨터공학부 졸업(석사). 2005년~현재 서울대학교 대학원 전기·컴퓨터공학부 박사과정. 관심분야는 차세대 인터넷, 오버레이 네트워크, 프로토콜 디자인 및 성능 분석



김 종 권

1981년 서울대학교 산업공학과 학사  
1982년 미국 조지아 공과대학교 산업공학과 석사. 1987년 미국 일리노이 대학교 전산학과 박사. 1984년~1987년 IBM 산호세 연구소 연구조원. 1987년~1991년 미국 Bellcore 통신 연구소 연구원. 1991년~현재 서울대학교 전기·컴퓨터공학부 교수. 관심분야는 차세대 인터넷, 초고속 라우터, 이동통신

조 성 호

1999년 서울대학교 전산과학과 졸업(학사). 2001년 서울대학교 대학원 전기·컴퓨터공학부 졸업(석사). 2001년~현재 서울대학교 대학원 전기·컴퓨터공학부 박사과정. 관심분야는 차세대 인터넷, 프로토콜 디자인 및 성능분석, 이동성 관리.



이동통신