

# TCP의 수동적 네트워크 용량 측정을 위한 지연 패킷 페어 기법

황재현<sup>○</sup> 유시환 유혁

고려대학교 컴퓨터학과

{jhhwang<sup>○</sup>, shyoo, hxy}@os.korea.ac.kr

## Delayed packet-pair scheme

### for passive network capacity estimation with TCP

Jae-Hyun Hwang<sup>○</sup> See-Hwan Yoo Hyuck Yoo

Department of Computer Science, Korea University

#### 요 약

종단에서 네트워크 용량에 대한 정보는 사용자뿐 아니라 전송 계층 프로토콜 상에서 전송량 등을 조절하기 위한 중요한 용도로 활용된다. 이러한 네트워크 용량(또는 대역폭)을 종단에서 측정하기 위해 활용되는 방법 중 하나가 바로 패킷 페어 기법이다. 패킷 페어 기법은 계산이 간단하고 빠른 측정이 가능하다는 장점이 있으나, 각 프로빙 패킷에 대한 즉각적인 응답 패킷을 필요로 하기 때문에 지연 ACK 방식이 존재하는 TCP 상에서는 적용하기 어렵다는 단점이 있다. 본 논문에서는 이러한 지연 ACK의 효과를 그대로 유지하면서 네트워크 용량의 추정이 가능한 지연 패킷 페어 기법을 제안한다. 시뮬레이션 결과를 통해 제안된 지연 패킷 페어 기법이 기존 방식의 정확성, 빠른 측정 시간 등의 장점을 보존하면서 지연 ACK의 사용 여부에 관계없이 정확히 동작할 수 있음을 보였다.

#### 1. 서 론

자신이 사용하는 네트워크의 용량을 알 수 있다는 것은 네트워크의 효율성이나 관리 측면에서 매우 중요하다. 이러한 정보는 TCP의 혼잡 제어 조절, 멀티미디어 스트리밍에 있어서의 전송량 조절, 또는 P2P에서의 피어(peer) 선택 등에 활용될 수 있다. 종단에서의 네트워크 용량 측정에 대한 연구는 최근까지도 다수 진행되어 왔는데 이들은 크게 별도의 프로빙(probing) 패킷을 이용한 능동적인 방식과 데이터 패킷을 그대로 측정에 이용하는 수동적인 방식으로 나뉜다. 능동적인 방식의 대표적인 기법으로는 Pathrate[1]와 CapProbe[2]가 있다. 이 둘의 차이점은 Pathrate의 경우 패킷 트레인(packet-train) 방식을 사용하는 반면, CapProbe는 패킷 페어(packet-pair) 방식을 사용한다는 것인데, CapProbe 쪽이 보다 가볍고 간단하며 측정 시간이 빠르다고 알려져 있다. 패킷 페어를 이용하여 네트워크의 용량을 측정하는 방법은 기본적으로 두 프로빙 패킷이 병목 지점(bottleneck)을 거쳐오면서 벌어지는 간격을 이용하는 것으로, CapProbe에서는 송신 측에서 프로빙 패킷 페어를 같은 시점에 동시에 전송하여 (즉, 패킷 페어 간 시간 간격을 없앴) 이 둘의 벌어지는 간격을 샘플링한 뒤, 가장 작은 간격을 취하여 네트워크 용량을 측정한다. 별도의 측정용 패킷을 사용하는 능동적인 방식과는 달리, 수동적인 방식은 원래 전송 되는 데이터 패킷을 측정 시 그대로 사용하는 방법으로 별도의 추가 패킷을

요구하지 않는다는 장점이 있다. CapProbe의 저자들은 CapProbe에서 사용한 기법을 이용하여 이것을 각각 TFRC와 TCP에 적용한 수동적 측정 기법(각각 TFRC Probe[3], TCP Probe [4])을 제안한 바 있다. 수동적인 방식에서 중요한 점은 TCP의 ACK와 같이 데이터 패킷에 대한 응답 패킷이 존재해야 한다는 것과 이들이 즉각적으로 반환되어야 한다는 것이다. 그러나 TCP의 경우 지연 ACK(delayed ACK)의 사용으로 인해 ACK가 데이터 패킷 두 개당 하나씩으로 줄어들기 때문에 패킷 페어간 간격을 측정할 수 없다는 문제가 발생한다. TCP Probe에서는 송신 측의 수정만으로 이 문제를 해결하기 위해 두 패킷씩 시퀀스를 바꿔 보내는 방법을 제시하였다. 즉, 순서 외의 시퀀스(out of sequence) 패킷을 수신 측에 먼저 도착하게 함으로써 즉각적인 중복 ACK(dup-ACK)를 발생시키고 뒤이어 복구 ACK를 보내게 함으로써 자연스럽게 지연 ACK의 효과를 없앤 것이다. 그러나 이 방법은 바꾸어 말하면 TCP에 수동적 측정 기법을 적용하기 위해서는 지연 ACK로 인한 이득을 포기해야 한다는 것을 의미한다. 또한 매 패킷 페어마다 중복 ACK가 발생하기 때문에 TCP 구현 내의 빠른 경로(fast path)를 활용하지 못하며, 상대방이 지연 ACK를 사용하지 않는 경우 시퀀스의 순서를 바꿀 필요가 없음에도 이를 반영하지 않는다는 단점이 있다. 이에 본 논문에서는 시퀀스의 순서를 바꾸지 않고 지연 ACK 효과를 그대로 유지하면서도 패킷 페어 방식이 가지는 장점을 보존할 수 있는 지연 패킷 페어(Delayed packet-pair) 기법을 제안한다. 시뮬레이션을

통해 제안된 방식이 TCP Probe와 비교하여 동일한 수준의 정확성 및 빠른 측정 시간을 가지면서도 지연 ACK의 사용 여부에 관계없이 정확하게 동작함을 보일 수 있었다.

본 논문은 다음과 같이 구성된다. 먼저 2장에서는 패킷 페어 기법 및 이를 사용한 네트워크 용량 측정 기법들에 대해 알아본다. 3장에서는 본 논문에서 제안하는 지연 패킷 페어 기법에 대해 설명하고 기존 기법과 비교한다. 4장에서는 시뮬레이션을 통해 주장하는 바를 검증하고 정확성 및 빠른 측정 시간 등의 속성이 유지됨을 보인 후 5장에서 결론을 맺는다.

## 2. 관련연구 및 동기

앞서 설명한 바와 같이 본 논문은 여러 네트워크 용량 측정 기법 중 패킷 페어 기법을 이용한 CapProbe와 그것을 이용한 TCP Probe에서의 측정 방식에 초점을 두고 있다. 기본적으로 패킷 페어 기법은 연속적으로 전송된 두 개의 패킷이 병목 링크에서 큐잉되어 다음의 T 시간만큼의 간격이 벌어진다는 사실에 착안한 것이다.

$$T = L / B$$

여기서 L은 두 번째 패킷의 크기를 의미하며, B는 병목 링크, 즉 경로 상에서 가장 작은 네트워크 용량(network capacity)을 가진 링크의 대역폭을 의미한다. 따라서 T 값을 측정함으로써 B를 구할 수 있게 된다[2]. 다만 패킷 페어 방식은 크로스 트래픽(cross-traffic)에 의한 큐잉 지연(queueing delay)으로 인해 그 정확도가 낮아질 수 있다. 즉, 첫 번째 패킷에 큐잉 지연이 발생하는 경우 패킷 간격이 좁아짐에 따라 상대적으로 과대 측정(over-estimation)이 되며, 두 번째 패킷에 지연이 발생하는 경우 반대로 과소 측정(under-estimation) 현상이 발생한다. CapProbe에서는 이러한 크로스 트래픽에 의한 지연을 구분하기 위해 두 프로빙 패킷의 RTT를 더한 지연 합(delay sum) 정보를 유지한다. 즉, 잘못 측정된 샘플에 대한 지연 합 값에는 크로스 트래픽에 의한 큐잉 지연이 포함되어 있으며, 이는 큐잉 지연이 포함되지 않은 최소 지연 합 값보다 높다는 것이다. 따라서 최소 지연 합을 가지는 패킷 페어를 대상으로 도착 간격을 측정하여 링크의 용량을 계산하게 된다.

이와 같은 내용은 TCP Probe에도 동일하게 적용되며, 다만 별도의 측정용 패킷을 추가적으로 사용하기보다 TCP가 전송하는 데이터 패킷을 그대로 측정에 사용한다는 점이 다르다. 그러나 패킷 페어 방식을 TCP에 적용하기 어려운 가장 큰 이유는 바로 지연 ACK의 사용 때문이다. TCP Probe에서는 역전 패킷 페어(inverted packet-pair) 기법을 제안함으로써 이를 극복하였는데, 이는 패킷 페어 시퀀스의 순서를 바꿈으로써 지연 ACK의 효과를 없앤 것이다. 그림 1은

이를 표현한 것으로, 먼저 수신 측에서 지연 ACK를 사용한다면 그림 1(위)의 그림처럼 두 개의 패킷 페어에 대해 하나의 ACK만이 돌아오기 때문에 패킷 간격을 측정할 수 없게 된다. 반면 패킷 페어의 순서를 'i+1', 'i'와 같이 바꿔서 보내게 되면, 'i'(즉, dup-ACK)와 'i+2' 순서의 연속 ACK를 받을 수 있게 된다.

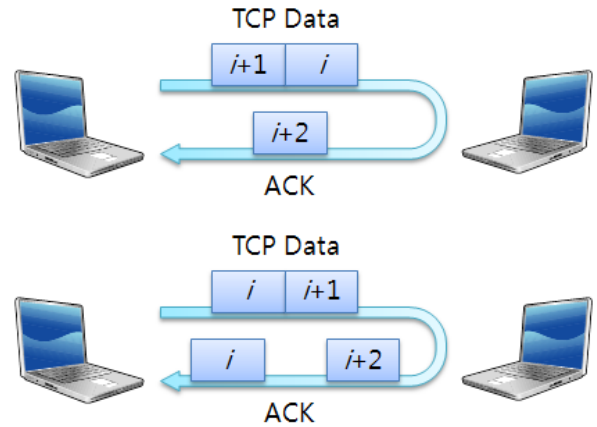


그림 1. Delayed ACK(위)와 inverted packet-pair에 의한 ACK(아래) [4]

이처럼 역전된 패킷 페어 기법이 네트워크 용량 측정 방법을 TCP에 적용시킬 수 있는 하나의 대안이 될 수 있으나, 결과적으로는 지연 ACK를 포기함으로써 TCP에 적용되던 ACK 패킷의 감소 효과를 없애게 된다. 이는 수동적 측정 방식의 장점인 추가적인 패킷이 사용되지 않는다는 장점을 상쇄시킨다. 따라서 지연 ACK의 효과를 유지할 수 있는 패킷 페어 기법이 필요하다.

## 3. 지연 패킷 페어 기법(Delayed packet-pair scheme)

본 논문에서 제안하는 지연 패킷 페어 기법의 요점은 수신 측에서 지연 ACK를 사용하는 경우 지연 ACK간의 간격을 통해 네트워크 용량을 측정하자는 것이다. 먼저 지연 ACK 사이의 간격이 의미 있는 값이 되기 위해서는 TCP Probe의 경우와 마찬가지로 해당 데이터 패킷이 동시에 전송되어야 한다. 여기에는 두 가지 이유가 있는데, 첫 번째는 병목 링크의 대역폭이 얼마인지 알 수 없기 때문이다. 즉, 병목 링크의 대역폭을 반영하는 패킷 간격을  $\Delta t$  시간이라고 했을 때, 패킷 페어 간 전송 간격이  $\Delta t$  보다 큰 경우 측정 값에 오차가 발생한다. 또 하나의 이유는 두 번째 패킷의 큐잉 지연의 발생을 최소화 시키기 위해서이다. 즉, 패킷 페어 간의 간격이 커질수록 크로스 트래픽에 의한 영향을 받기 쉬워지기 때문에 정확한 측정을 위해서는 그림 1에서와 같이 측정에 사용되는 패킷 페어가 동시에 전송이 이루어져야 한다.

그림 2는 수신 측에서 지연 ACK를 사용하는 경우 일반적인 패킷 페어 전송 방법(a)과 본 논문에서

제시하는 지연 패킷 페어 전송 방법(b)을 나타낸 것이다. 그림에서 데이터 패킷 내 숫자는 전송 순서를 의미하며, ACK의 숫자는 대응되는 데이터 패킷에 의해 발생된 것임을 나타낸다. (예를 들면, 2번 ACK는 2번 데이터 패킷에 의해 발생한 것이다.) 또한 회색으로 표시된 것은 ACK를 발생시키는 패킷임을 의미한다. 먼저 그림 2 (a)를 살펴보면, 1번과 2번, 3번과 4번이 쌍을 이루기 때문에 두 쌍 사이에 전송 시간의 차이가 생긴다. 앞서 설명한 바와 같이 이 간격이  $\Delta t$  보다 작으면 병목 지점의 큐잉으로 인해 올바른 샘플 측정이 가능하나, 그렇지 않은 경우 2번과 4번 지연 ACK의 간격이 벌어져 과소 측정이 된다. 따라서 지연 ACK 사이의 간격이 병목 링크의 대역폭을 정확히 반영하기 위해서는  $2\Delta t$ 가 되어야 하며, 이를 항상 보장하기 위해서는 ACK를 발생시키는 데이터 패킷이 동시에 전송되어야 한다. 이때 지연 ACK를 연달아 발생시키는 최소 개의 데이터 패킷 수는 세 개가 된다. 즉, 앞서 홀수개의 패킷이 전송된 상황에서 세 개의 패킷을 동시에 전송하면 첫 번째 패킷과 세 번째 패킷에 의해 지연 ACK가 연이어 발생할 수 있기 때문이다. 그림 2 (b)는 데이터 패킷을 세 개씩 묶어 시퀀스를 바꾸지 않고 전송하는 지연 패킷 페어 방식을 나타낸 것이다. 먼저 전송되는 세 개의 패킷 쌍 내의 2번 ACK는 측정에 영향을 주지 않으나, 뒤이어 전송되는 패킷 쌍에 의해 4번과 6번 ACK가 발생되며, 송신 측에서는 이 두 ACK를 통해  $2 \times packet\_size / 2\Delta t$  로 네트워크 용량을 계산해 낼 수 있다.

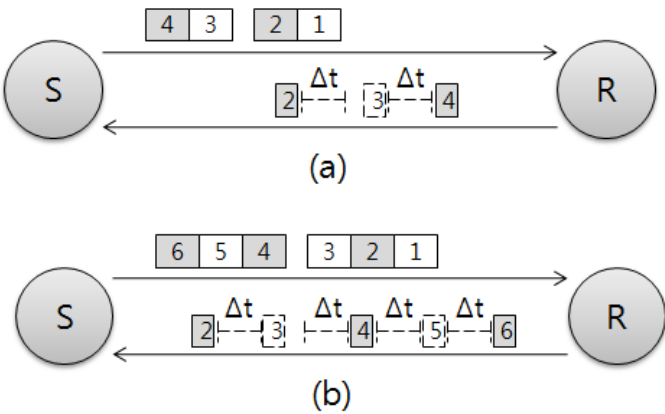


그림 2. Delayed ACK 하의 (a) 일반 packet-pair와 (b) Delayed packet-pair

여기서 한 가지 고려해 볼 수 있는 문제점은 지연 ACK는 200ms의 타임아웃(timeout)을 가진다는 점이다. 즉, 데이터 패킷이 도착한 후 200ms 내에 추가 데이터 패킷이 도착하지 않거나 수신 측에서 보낼 데이터 패킷이 없으면 ACK를 즉시 전송한다. 그림 2 (b)에서 만약 3번과 4번 패킷 사이에 200ms의 지연이 발생한다면 4번과 6번에 의해 ACK가 발생되지 않게 될

것이다. 전송 윈도우 크기가 확보된 상황에서 보낼 데이터가 계속해서 존재한다면 이러한 200ms의 지연은 병목 링크의 대역폭이 가장 큰 원인이라 할 수 있다. 이는 패킷 크기를 1500 바이트라고 했을 때  $1500 \times 8 / 0.2$  로 60Kbps가 된다. 최근의 네트워크 링크들은 대부분 이보다 크기 때문에 큰 문제가 되지 않지만, ISDN과 같이 64Kbps의 전화선을 사용하는 경우  $\Delta t$ 는 0.1875초가 되어 프로빙의 두 번째 패킷의 지연에 따라 첫 번째 패킷에서 ACK가 발생할 가능성이 있다. 이렇게 측정을 실패하는 경우를 구분하기 위해 지연 패킷 페어에서는 ACK가 커버하는 패킷의 수가 동일하게 들어오는 경우에 한해 측정을 수행하도록 하였다. 예를 들면, 지연 ACK가 연속해서 2회 이상 들어오는가를 확인하는 것이다. 또한 60Kbps 이하의 링크 또는 전화 모뎀(56Kbps)을 사용하는 경우처럼 지연 ACK의 효과가 아예 사라지거나 지연 ACK를 사용하지 않는 경우에도 ACK가 커버하는 패킷의 수가 하나씩 연속해서 동일하므로 이 역시 측정 대상에 포함이 되며, 이때는  $packet\_size / \Delta t$  로 네트워크 용량이 측정될 것이다.

4. 시뮬레이션 실험

본 논문에서 제안하는 지연 패킷 페어 기법의 검증을 위해 NS-2 (버전 2.30) 네트워크 시뮬레이터[5] 내의 TCP Reno에 이를 구현하였다. 실험은 다음과 같은 다섯 가지 시나리오에 대해 병목 링크의 용량을 측정하도록 하였다. 또한 TCP Probe와의 비교를 통해 기존의 정확성과 빠른 측정 속도의 특성이 그대로 유지되는지 알아보았다.

- 100Mbps, 10ms의 병목링크와 지연 ACK 사용
- 10Mbps, 50ms의 병목링크와 지연 ACK 사용
- 1Mbps, 50ms의 병목링크와 지연 ACK 사용
- 64Kbps, 50ms의 병목링크와 지연 ACK 사용
- 10Mbps, 50ms의 병목링크와 지연 ACK 미사용

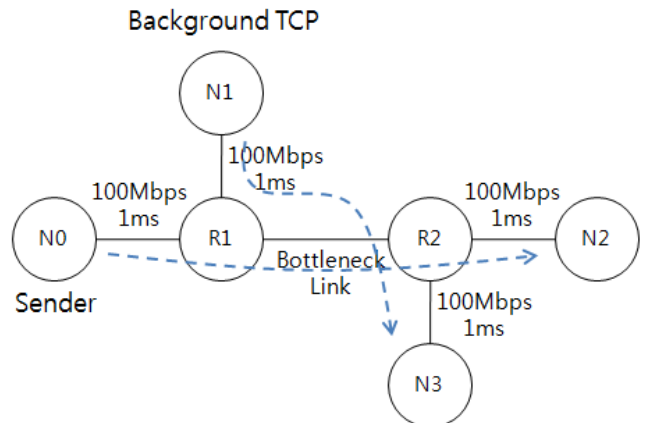


그림 3. 시뮬레이션 토폴로지

표 1. 네트워크 용량 측정의 정확도와 측정 시간 비교

| 측정 기법          | TCP Probe  |          | Delayed Packet-Pair |          |
|----------------|------------|----------|---------------------|----------|
|                | 측정 값(Mbps) | 측정 시간(s) | 측정 값(Mbps)          | 측정 시간(s) |
| 시나리오           |            |          |                     |          |
| 100Mbps, 10ms  | 96.689     | 1.546    | 96.689              | 0.874    |
| 10Mbps, 50ms   | 9.669      | 2.982    | 9.669               | 1.663    |
| 1Mbps, 50ms    | 0.967      | 7.2      | 0.967               | 8.945    |
| 64Kbps, 50ms*  | 62 (Kbps)  | 42.027   | 69 (Kbps)           | 41.977   |
| 10Mbps, 50ms** | 9.669      | 2.988    | 9.669               | 1.064    |

\* background TCP를 사용하지 않았음. \*\* 지연 ACK를 사용하지 않았음.

실험에 사용한 토폴로지는 그림 3과 같으며, 패킷 크기는 1500 바이트로 고정하고 각 실험마다 100초간 시뮬레이션을 수행하였다. 먼저 그림 4는 전송된 데이터 패킷 수에 대해 얼마만큼의 ACK 패킷이 생성되었는지를 비교한 그래프이다. 마지막 지연 ACK를 사용하지 않은 경우를 제외하고, 지연 패킷 페어 기법은 지연 ACK의 사용을 그대로 수용하기 때문에 ACK의 수가 50%정도 생성되고 있음을 알 수 있다. 반면 TCP Probe의 경우, 시퀀스의 꼬임으로 인해 중복 ACK(dup-ACK)가 추가되어 보다 많은 ACK를 사용하게 된다. 뿐만 아니라 지연 ACK를 사용하지 않는 경우에도 중복 ACK를 발생시켜 불필요한 연산을 수행한다.

표 1은 각 시나리오에 대해 네트워크 용량 측정의 정확도와 측정에 걸린 시간을 비교한 결과를 정리한 것이다. 정확도의 경우 64Kbps 환경에서 약간의 과대 측정된 경우를 제외하고는 TCP Probe와 동일한 측정 값을 얻어냈다. 또한 측정에 걸리는 시간도 대부분 보다 빠르거나 비슷한 수준임을 알 수 있다. 결과적으로 본 논문에서 제안하는 지연 패킷 페어 기법이 지연 ACK를 그대로 사용하면서, 동시에 높은 정확도와 빠른 측정 속도 역시 같이 유지시켜준다는 것을 알 수 있다.

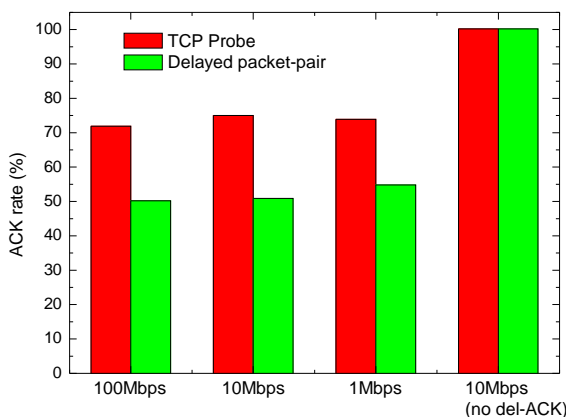


그림 4. 전송된 데이터 패킷 개수 대비 생성된 ACK 개수의 비율 비교

## 5. 결 론

일반적으로 패킷 페어 기법은 종단에서 네트워크의 용량 측정을 위해 사용되는 방법으로, 계산이 간단하고 빠르다는 장점이 있다. 그러나 TCP에 수동적으로 적용 시 지연 ACK로 인해 측정이 어렵다는 문제점이 존재하여 능동적인 측정 방법에 대한 연구가 주류를 이루어온 것이 사실이다. 이에 본 논문에서는 지연 ACK의 효과를 그대로 유지하면서 네트워크 용량의 측정이 가능한 지연 패킷 페어 기법을 제안하였다. 시뮬레이션 결과를 통해 제안된 기법이 TCP Probe의 역전 패킷 페어 방식과는 달리 지연 ACK에 의한 ACK 절감 효과를 그대로 유지하면서 기존의 정확성, 빠른 측정 시간 등의 장점을 같이 취할 수 있음을 보였다. 또한 TCP Probe와 마찬가지로 송신 측의 수정만으로 사용이 가능하며, 전송 순서를 변경하지 않으므로 지연 ACK의 사용 여부에 관계없이 효율적으로 동작한다.

## 참고문헌

- [1] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?", in Proc. of IEEE INFOCOM, 2001.
- [2] R. Kapoor, L. J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "CapProbe: A Simple and Accurate Capacity Estimation Technique", in Proc. of ACM SIGCOMM, 2004.
- [3] L. J. Chen, T. Sun, D. Xu, M. Y. Sanadidi, and M. Gerla, "Access link capacity monitoring with TFRC probe", in The 2<sup>nd</sup> Workshop on End-to-End Monitoring Techniques and Services, 2004.
- [4] A. Persson, Cesar A. C. Marcondes, L. J. Chen, M. Y. Sanadidi, and M. Gerla, "TCP Probe: A TCP with built-in Path Capacity Estimation", in The 8<sup>th</sup> IEEE Global Internet Symposium, 2005.
- [5] NS-2 Network Simulator version 2.30. <http://www.isi.edu/nsnam/ns>