

# 액티브 네트워크 기술과 액티브 큐 매니지먼트를 이용한 혼잡제어

최기현\*, 장경수\*\*, 신호진\* 신동렬\*  
\*성균관대학교 전기전자 및 컴퓨터공학과  
\*\*경인 여자 대학  
e-mail : tomahawk@nova.skku.ac.kr

## Congestion Control Using Active Network and Active Queue Management

Kee-Hyun Choi\*, Kyung-Soo Jang\*\*, Ho-Jin Shin\*, Dong-Ryeol shin  
\*Dept. of Electrical and Computer Engineering, Sungkyunkwon University  
\*\* School of Computer & Information Technology, Kyungin Women's College

### 요 약

기존의 TCP/IP 네트워크 상에서 혼잡을 제어하는 일반적인 방법은 end-to-end 피드백 방식이다. 혼잡이 발생했는지의 여부는 수신측의 ACK 신호로 예측할 수 밖에 없으므로 RTT(round trip time)가 길어질 경우 혼잡처리 시간이 길어지게 된다. Active Network 기술을 이용한 ACC(Active Congestion Control)의 경우 혼잡이 발생한 라우터에서 전송자에게 직접적으로 정보를 전송하기 때문에 보다 빠른 혼잡처리가 가능하다. 비록 코어 라우터에서 패킷을 누락시키는 방법에 있어 상태 정보를 필요로 하는 단점이 있지만 모의 실험 결과 향상된 처리량을 보여주었다. 혼잡처리에 있어 새로운 처리 방식은 AQM(Active Queue Management)이다. 이 방식은 RED 를 기본 모델로 변형된 것이 대부분이며 RED 가 갖는 단점을 개선하고 혼잡처리와 공정성(fairness)을 향상 시키는데 목적이 있다. 본 논문에서는 ACC 의 피드백 메커니즘과 AQM 의 융통성 있는 큐 관리 기법을 이용하여 보다 향상된 혼잡제어 기법을 보여준다.

### 1. 서론

인터넷 사용자의 증가와 멀티 미디어 데이터 양의 증가로 인해 인터넷은 새로운 혼잡처리 방법을 요구 하고 있다. 이러한 요구를 충족시키기 위한 여러 방법론이 대두되고 있으며 대표적인 예로 액티브 네트워크(Active Network)와 Active Queue Management(AQM) 기술 등이 있다.

기존의 네트워크 내부 라우터(core router)들이 주로 패킷(packet) 전송과 경로할당을 처리하는 반면 액티브 네트워크에서는 내부 라우터에서 특정처리에 필요한 프로그램을 실행 할 수 있도록 하여 혼잡이 발생할 경우 보다 신속하게 처리하게 된다. 혼잡 제어에 있어 액티브 네트워크의 이용은 네트워크 혼잡에 보다 신속하게 응답하게 해준다. 기존의 단대단(end-to-end) 피드백 혼잡제어는 오직 단말노드에서만 혼잡을 감지

하고 제어할 수 있다. 따라서 액티브 네트워크 기술을 이용하여 혼잡을 네트워크 내부에서 직접적으로 처리해 줄 수 있다면 네트워크 성능을 향상시킬 것이다[6].

액티브 네트워크 기술은 패킷 내에 프로그램이나 처리에 필요한 정보를 담아 전송하게 되는데 네트워크 상에 있는 노드는 이 패킷(active 패킷 또는 smart 패킷[7])을 읽어 들여 필요한 처리를 할 수 있다. 또한 네트워크 내부 노드들은 서로에게 필요한 네트워크 상의 정보를 공유할 수 있다. 이 기술을 사용할 경우 혼잡이 발생하면 네트워크 내부에서 이를 감지하여 주변 라우터에 혼잡에 대한 정보를 알려줄 수 있으며 직접적으로 혼잡을 제어할 수 있다.[10] 반면 기존의 혼잡제어 방식은 수신측과 송신측과의 지연 즉 RTT(Round Trip Time)가 클 경우 그 만큼 혼잡을 제어

하는데 시간이 더 오래 걸리게 된다. 또 네트워크의 대역폭이 클 경우 혼잡이 감지되어 어떤 처리를 시작할 때까지 이미 전송된 패킷으로 인해 혼잡이 더욱 가중되게 된다.

기존의 FIFO 큐(queue)는 혼잡이 가중될 경우 여러 흐름(flow)들이 동시에 전송률을 줄이고 늘리는 동기화 현상(Global synchronization)이 발생하여 Congestion collapse 을 발생시킨다. 이러한 문제를 해결하기 위하여 AQM[5]이 등장하였으며 RED[12], ARED[4], FRED[8], SRED[2], RIO[9], BLUE[3], CHOCe[1] 등등의 AQM 들이 대표적인 예이다. AQM 은 혼잡처리를 보다 신속하게 처리하거나 미리 처리하여 패킷 누락으로 인한 대역폭의 낭비를 줄이게 된다. 본 논문에서는 SRED 와 CHOCe 그리고 FRED 에서 제안한 혼잡 메커니즘을 이용하여 Chameleon 큐를 구성하고 코어 라우터와 엣지 라우터(edge router)간의 통신을 이용하여 혼잡을 제어해 보겠다.

### 2. 액티브 네트워크 동기와 배경

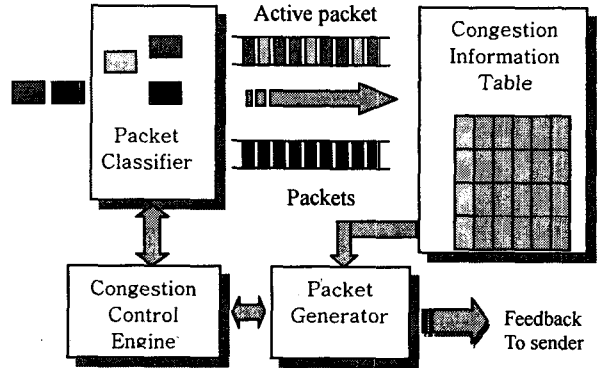
최근에 연구는 네트워크 내부에서 응용프로그램에 관련된 처리에 관심을 갖고 있다. 네트워크 노드나 단말시스템에서 정적으로 구현되어 있는 프로토콜 모듈은 필요에 따라서 동적으로 다른 모듈과 결합하여 새로운 서비스를 제공하게 된다. 네트워크상에서 응용프로그램의 명시적인 처리요구에 대한 처리의 예는 동적 캐싱과 적응 캐싱에 관한 연구이다. 이 접근 방법에서 클라이언트의 요청을 직접적으로 응답할 것인지 아니면 네트워크 상태를 알고 있는 다른 캐시나 원래 서버에 이 요청을 넘길 것인지를 결정하는 캐시에 보내진다.

MPEG Video 같은 멀티미디어 트래픽은 계층화된 형태를 갖는다. 각 계층에 우선 순위를 할당하고, 낮은 우선 순위를 갖는 계층을 포함하는 패킷을 누락(drop)함으로써 네트워크의 혼잡 시간 안에 실시간 연결을 유지할 수 있게 된다. 이와 유사한 방법이 오디오 전송에 사용되는데 기여도에 따른 레벨에 따라서 각 음성 신호 구성요소는 나누어지게 된다. 기여도가 크지 않은 신호 구성 요소들은 혼잡이 발생한다면 무시될 수 있으며 낮은 우선 순위 패킷에 위치한다. 이와 유사하게 Kansas 대학의 Wireless ATM Voice/Data project[11]에서는 두 개 이상의 패킷이 합성되어 하나의 음성 패킷으로 보내진다. 이는 여러 신호를 합성하여 대역폭을 줄이는 것이다. 따라서 응용프로그램은 제한된 방법으로 네트워크가 이런 데이터를 어떻게 다룰 것인지 알려 주어야 한다.

이런 모든 노력들은 응용프로그램의 명시에 따른 처리를 위해 새로운 네트워크 프로토콜을 생성하기 위한 것들이다. 하지만, 기존 네트워크에서 새로운 프로토콜 표준의 설계와 전개는 시간을 필요로 한다. 액티브 네트워크 플랫폼은 오랜 표준화 절차 없이 최소한의 노력으로 네트워크 안에 새로운 프로토콜을 만들고 실험할 수 있는 훌륭한 도구다.

### 3. 액티브 라우터의 구조

코어 라우터는 상태정보를 저장하여 혼잡 발생시 누락을 결정하기위한 정보로 CIT(Congestion Information Table)를 이용한다. 전반적인 구조는 (그림 1)와 같다. 코어 라우터에 패킷이 도착하면 패킷 분류기에 의해서 일반 TCP 패킷과 액티브 패킷을 구별한다. 분

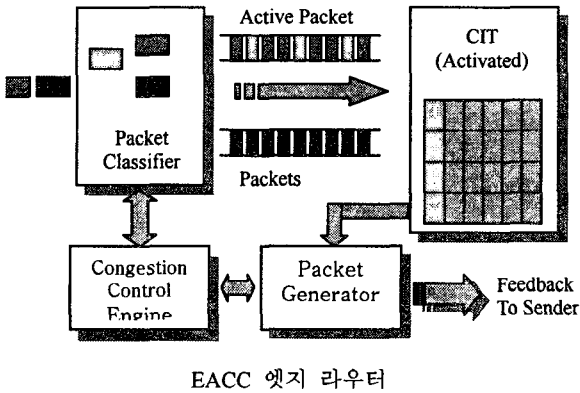
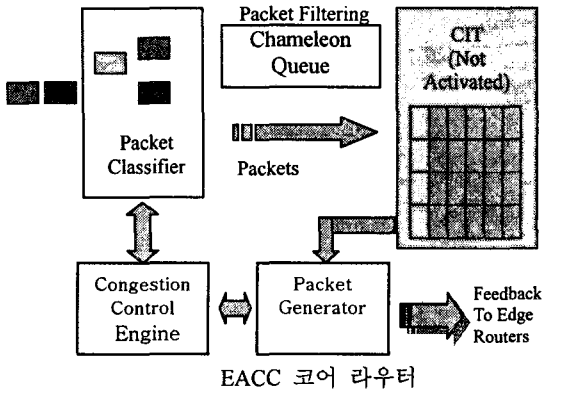


(그림 1) 액티브 라우터의 구조

류된 패킷은 ACC 코어 라우터에서의 실행 환경인 CCE(Congestion Control Engine)에 의해 실행되거나 정보를 얻게 된다. CCE 에서 처리된 패킷은 CIT 에 정보로써 저장되고 CCE 는 패킷 생성기에 패킷을 피드백 할 것인지 여부를 전달하게 된다. CIT 에는 흐름(flow)별로 전송자 주소, 수신자 주소, 윈도우 크기, 패킷 누락 개수 등의 정보가 저장된다. (그림 1)에서 알 수 있듯이 ACC 는 코어 라우터에서 상태 정보를 관리하게 된다. 코어 라우터에 연결된 흐름의 수가 많아질 경우 상태정보의 관리가 어려워지며, 액티브 패킷과 일반 패킷의 처리에 자원을 낭비 하게 된다.

### 4. ACC 와 Chameleon 큐

(그림 2)는 ACC 와 Chameleon 큐를 더한 액티브 라우터의 구조이다. 코어 라우터와 엣지 라우터는 CIT 의 가동 여부와 CCE 에서 처리되는 알고리즘만 차이가 있을 뿐 그 구조상 거의 같음을 볼 수 있다. 액티브 라우터의 장점 중의 하나는 특정 목적에 따라 라우터의 교체나 프로그램의 설정이 필요 없다는 것이다. 단지 패킷 속에 필요한 정보와 프로그램을 담아 전송시키면 실행환경에서 그 목적에 맞게 동작하기 때문이다. 이러한 실행환경이 없는 일반 라우터의 경우에는 네트워크 관리자가 라우터를 직접 설정해 주어야 할 뿐만 아니라 새로운 장비의 교체까지 해야 한다. 코어 라우터는 큐 관리 방식에 있어 RED 의 변형을 사용하였다. 전반적인 구조는 RED 와 거의 유사하지만 문턱 값(threshold) 값에 따라 패킷을 누락시키는 RED 의 누락 메커니즘 이외에 본 논문에서 제안한 RED 의 변형인 Chameleon 큐는 큐 관리에 있어 라우터와의 통신을 이용한다. ECN(Explicit Congestion Notification)을 사용한 RED(이하 RED ECN)와의 차이점은 혼잡지점에서 바로 전송자에게 혼잡 신호를 줄 수 있을 뿐만 아니라 이웃 라우터와 통신하여 보다

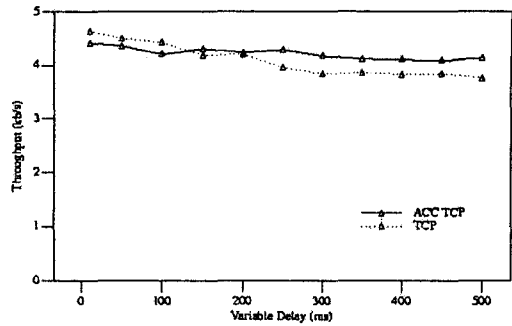


(그림 2) EACC 라우터의 구조

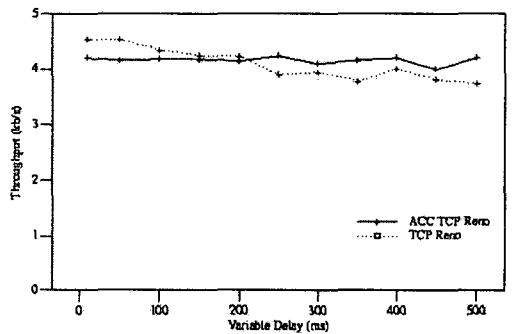
빠른 네트워크 상의 정보를 이용할 수 있다. 또한 RED ECN 은 전송자가 혼잡 신호를 받게 되는데 까지 걸리는 시간이 RTT 에 영향을 받으므로 라우터와의 통신과 전송자에게 직접 혼잡신호를 피드백 하는 Chameleon 큐와 비교해서 신속한 처리가 어렵다.

5. 모의실험 결과

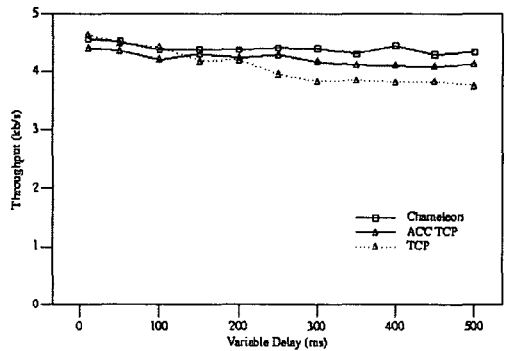
액티브 라우터를 이용해서 혼잡을 제어할 경우 TCP Tahoe 나 TCP Reno 모두 혼잡이 발생한 라우터에서 소스까지 지연시간이 짧을 경우 네트워크상에 전송되어 있는 패킷의 수가 상대적으로 적다. 따라서 누락되는 패킷이 차지하는 비율이 상대적으로 커지게 되므로 (그림 3~4)처럼 성능이 떨어지게 된다. 반면 지연시간이 클 경우에는 혼잡을 발생시키는 트래픽의 양이 상대적으로 많아지게 되므로 성능이 향상되었다. (그림 3~4)은 TCP Tahoe 와 TCP Reno 에 ACC 를 적용한 결과이다. bandwidth \* delay 가 작을 경우에는 기존의 제어방식이 오히려 더 높은 처리량(throughput)을 보였다. 이 점이 ACC 액티브 라우터가 전송자를 대신해서 트래픽의 양을 조절하는 역할에 있어 첫번째 문제점이다. 두 번째 문제점은 ACC 의 액티브 라우터는 혼잡이 발생할 경우 무차별적인 패킷 누락이 발생한다는 것이다.



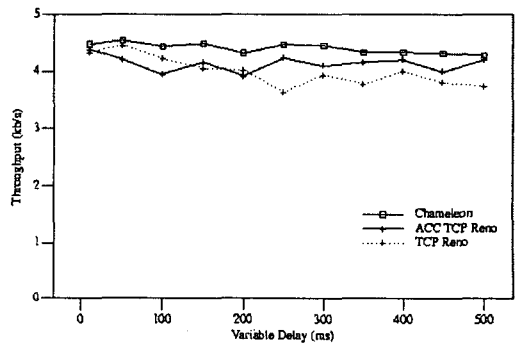
(그림 3) TCP Tahoe 와 ACC TCP Tahoe(RED)



(그림 4) TCP Reno 와 ACC TCP Reno(RED)



(그림 5) TCP Tahoe, ACC, ACC with Chameleon Queue(RED)



(그림 6) TCP Reno, ACC, ACC with Chameleon Queue(RED)

상대적으로 bandwidth \* delay가 클 경우에는 네트워크상에 있는 패킷의 양이 상대적으로 많은 반면 bandwidth \* delay가 작을 경우 액티브 라우터에서의 패킷 필터링이 상대적으로 너무 많게 되며, 혼잡이 발생할 경우 특정 전송자가 계속해서 필터링 될 확률이 높게 된다. 이는 ACC가 RTT의 변화에 따른 처리와 필터링 메커니즘에 전반적인 문제가 있음을 보여준다. 비록 ACC가 위와 같은 문제점은 있으나 (그림 3~4)에서 볼수 있듯이 bandwidth \* delay가 큰 네트워크 환경에서 ACC가 혼잡제어에 성능상의 이점을 제공함을 확인할 수 있다.

Chameleon 큐를 사용한 ACC는 엡지 라우터에서 각 전송자로 피드백된 정보로 전송자는 윈도우 크기를 조절하게 된다. ACC와 비교할 때 엡지 라우터에서 전송자까지의 지연시간이 작더라도 기존의 TCP와 같은 처리량을 나타냈다. ACC는 위에 언급한 문제점 이외에 다른 문제점을 갖는다. ACC는 각 전송자가 네트워크에 보낼 수 있는 패킷의 크기를 알 수 없고, 패킷 누락이 발생한 전송자에 대해서만 코어 라우터에서 피드백이 이루어지게 된다. 이는 어떤 전송자는 지속적으로 처리량에 영향을 받는 반면, 어떤 전송자는 지속적으로 높은 윈도우 크기로 전송 가능하기 때문에 공평성(fairness)에 문제점을 갖는다. Chameleon 큐를 사용한 ACC는 이를 개선하기 위해서 엡지 라우터와 코어 라우터 간의 통신(communication)을 사용하였다. (그림 5-6)는 TCP Tahoe와 TCP Reno에 Chameleon 큐를 사용한 ACC를 적용했을 경우와 ACC를 적용했을 경우, 그리고 기존의 혼잡제어 방식을 적용했을 경우의 결과이다. 작은 bandwidth \* delay의 경우에도 기존의 방식과 거의 비슷한 결과를 보인다. Bandwidth \* delay가 클 경우에도 기존의 방식 뿐만 아니라 ACC보다 성능이 개선되었다.

## 6. 결론

기존의 혼잡제어 방식은 혼잡 발생 시점에서 혼잡에 대한 처리를 할 수 없다. 즉 혼잡이 발생한 후 패킷 누락에 대한 신호가 전송자에게 도착했을 경우에 혼잡을 감지 하게 된다. 혼잡을 감지 하는데 걸리는 이런 제어시간이 길어 지면 길어 질수록 혼잡은 더욱 악화 된다. 혼잡 발생과 동시에 혼잡에 대한 처리할 수 있다면 전송자는 보다 빠르게 혼잡에 대처할 수 있을 것이다.

본 논문에서는 네트워크의 내부 정보를 이용하여 혼잡 발생 시점에서 혼잡을 제어하는 ACC와 Chameleon 큐를 사용한 ACC를 이용하여 혼잡을 제어 하였다. 위의 두 방식 모두 bandwidth \* delay가 클 경우 기존의 혼잡 제어 방식보다 액티브 라우터를 이용해서 혼잡을 제어할 경우 기존의 혼잡제어의 비효율성을 제거하여 성능이 향상됨을 보여준다.

ACC는 TCP 혼잡제어 방식마다 단순히 누락시키고 이 정보를 각 전송자에게 알려주는 'Drop and Notice'를 이용하여 혼잡을 제어하였다[6]. 하지만 이와 같은 방법은 RTT에 따른 처리가 불가능하고 특정 전송자

에게 불리한 처리를 할 가능성이 있으며 패킷 누락 방법에 있어서 문제점을 갖는다. Chameleon 큐를 사용한 ACC는 ACC의 단점을 해결하기 위해 혼잡 처리 방식에 라우터와의 통신을 추가하여 네트워크 내부에서의 정보를 이용하고 이를 각 라우터가 이용하여 ACC의 문제점을 해결 했으며 엡지 라우터의 패킷 복사 기능으로 모든 전송자는 혼잡이 발생할 경우 네트워크 상태를 알 수 있다. 라우터에서 누락된 정보 이외에 라우터 내부의 통계적 정보나 연결된 트래픽에 대한 정보, 큐에 저장된 패킷의 개수 등을 이용해서 혼잡을 제어 했기 때문에 보다 안정적이고 효율적인 제어가 가능하다. 또한, 엡지 라우터에서 실시간 트래픽과 비 실시간 트래픽을 구별하여 처리할 수 있는 구조를 제공함으로써 네트워크 관리자로서 하여금 보다 원활한 네트워크 관리를 제공해 줄 수 있다. 앞으로 본 논문의 모의실험 결과를 토대로 전체적인 라우터의 구성을 확장하여 네트워크의 특성에 따라 큐 관리 기법을 동적으로 변경할 수 있는 라우터 구조에 대한 연구가 있어야 할 것이다.

## 참고문헌

- [1] R. Pan, B. Prabhakar, and K. Psounis, "CHOCkE- A stateless active queue management scheme for approximation fair bandwidth allocation," in IEEE Infocom '2000, March 2000.
- [2] Ott, T., Lakshman, T. and Wong, L., "SRED: Stabilized RED", Proceedings of INFOCOM'99, pp 1346-1355, March 1999.
- [3] W. Feng, D. Kandlur, D. Saha, K. Shin, "Blue: A New Class of Active Queue Management Algorithms", U. Michigan CSE-TR-387-99, April 1999
- [4] W. C. Feng, D. Kandlur, D. Saha, and K. Shin. "A Self-configuring RED Gateway", In Infocom'99, 1999
- [5] Braden, B. et al "Recommendations on queue management and congestion avoidance in the internet", IETF RFC (Informational) 2309, April 1998.
- [6] Theodore Faber, "ACC: Using Active Networking to Enhance Feedback Congestion Control Mechanisms". IEEE Network, IEEE, pp. 61-65, May/June 1998
- [7] S. Alexander, C. Gunter, A. Keromytis, G. Minden, D. Wetherall, B. Braden, A. Jackson, "The Active Network Encapsulation Protocol (ANEP)", 1997. <http://www.cis.upenn.edu/~switchware/ANEP/>
- [8] Lin, D. and Morris, R.(1997) "Dynamics of Random early Detection" Proceedings of SIGCOMM'97
- [9] Clark, D. and Wroclawski, J., "An Approach to Service Allocation in the Internet", IETF Internet Draft draft-clark-diff-svc-alloc-00.txt, July 1997.
- [10] S. Bhattacharjee, K. Calvert, and E. Zegura, "On Active Networking and Congestion", Technical Report GIT-CC-96-02, College of Computing, Georgia Tech., Atlanta, GA 1996.
- [11] E. Lindsley, Narrowband ATM Networks, B.S.C.S thesis, Dept. of Electrical Engg. and Computer Science, University of Kansas, 1994
- [12] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Trans. on Networking, vol.1, no. 4, Aug. 1993.