제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003.5)

TCP/UDP 환경에서의 RED성능평가

김용범*, 채현석**, 조태경***, 최명렬*
*한양대학교 전자전기제어계측학과
**동원대학 인터넷 정보과
***한양 사이버대학교 컴퓨터학과
e-mail:falcon20@asic.hanyang.ac.kr

The Perfomance Evaluation of RED In TCP/UDP Traffic

Yong-Bum Kim*, Hyun-Suk Chae**, Tae-Kyung Cho***,
Myung-Ryul Choi*
*Dept of EECI, Hanyang University
**Dept of internet & information retrieval, Tongwon college
***Dept of Computer Science, Hanyang Cyber University

요 약

인터넷의 발달로 멀티미디어 데이터 처리가 필요하다. 이를 위해 TCP/UDP를 혼합 환경에서 사용하게 되었고 발생하는 혼잡이나 공정성의 문제를 해결하기 위해 많은 논의가 이루어지고 있다. 본 논문에서는 RED방식에서 사용하는 4개의 파라미터가 라우터 큐의 동작에 미치는 영향을 분석한다. RED 파라미터 분석을 통해서 고정된 파라미터 값을 사용할 경우 다양한 트래픽 상황에 적절히 반응하지 못하는 것을 보여준다. 또한 NS2를 이용한 시뮬레이션에서 DT방식과 RED방식을 비교하면서 공정성이 향상됨을 보인다.

1. 서론

인터넷의 대표적인 전송 프로토콜에는 TCP와 UDP가 있다. TCP는 데이터 전송의 신뢰성과 정확성을 보장하는 대신 시간이 많이 걸려 멀티미디어의 데이터 처리에 적합하지 않다. 하지만 다른 하나의 전송 프로토콜인 UDP는 데이터 전송에 신뢰성은 없으나 시간이 적게 걸려 멀티미디어 데이터 전송에 적합하다. 따라서 실제 인터넷 환경에서 TCP와 UDP가 함께 사용되고 프로토콜의 알고리즘이 상이함에 따라 데이터 전송에 대한 공정성에 문제가 발생한다[1,2].

이런 공정성의 문제를 중간 노드인 라우터에서 해결하기 위해 DT(Drop-Tail)와 RED(Random Early Detection)와 같은 알고리즘이 제안되었다. 이 알고리즘들은 또한 현재 컴퓨터 네트워크의 패킷의 전송과 큐 관리를 담당하는 한정된 버퍼를 갖고 있는 라우터에서의 혼잡상황을 해결하기 위해 제안되었다[4,5]. DT방식은 모든 기법 중에서 가장 간단한 알고리즘으로서, 패킷을 선택적으로 폐기시키는 것이 아니라 이용 가능한 버퍼의 공간이 없을 때 곧 바로 폐기시킨다. 하지만 RED방식은 능동적으로 버퍼를 관리함으로써 링크의 사용효율을 높인다[5]. RED는 평균 큐 길이에 따라 망의 혼잡정도를 결정하여 이에 따라 패킷을 폐기할 확률을 결정한다.

본 논문에서는 RED를 사용하는데 RED 파라미터의 값을 어떻게 정하느냐는 근본적인 문제에서부터 출발하여, RED 파라미터 값이 라우터 큐에 미치는 영향을 분석하고 최적의 성능을 위한 파라미터 사용방향을 제시하겠다. 또한 TCP/UDP 혼합 환경에서 DT방식과 RED방식의 성능 평가를 NS2(Network Simulator)의 시뮬레이션을 통해 검증하고 RED방식에서 데이터 처리의 공정성이 향상되는 것을 검증하겠다.

2. 인터넷 프로토콜

2.1 TCP(Transmission Control Protocol)

TCP/IP 인터넷 프로토콜에 의해 제공되는 신뢰성이 보장되는 데이터 전송서비스를 TCP라고 한다. 이 프로토콜은 두 호스트가 어떻게 TCP흐름 전송을 초기화하고 이 초기화가 이루어졌을 때 두 호스트가 어떻게 서로 승인하는지를 보여준다. TCP는 파일 전송, 전자 메일, Telnet접속, Web접속, 클라이언트-서버 등의 응용 프로그램을 지원하며 직접적으로 네트워크의 상태를 파악할 수는 없지만 상대편 종단과의 대화를 통해 네트워크의 상태를 유추하고 이에 따라 흐름 제어 및 혼잡 제어를 수행한다.

2.2 UDP(User Datagram Protocol)

인터넷의 표준 프로토콜인 TCP/IP의 기반이 되는 프로토콜의 하나로 단순하고, 데이터 본위 개념의 전송 계층에 위치한 프로토콜이다. 데이터가 연속적인 흐름으로 이루어지는 TCP와 달리 각각 하나의 UDP 데이터를 외부로 전송하는 것으로서 모든 동작이 마무리된다. TCP에서는 전송하고자하는 호스트와의 접속을 설정한 후에 통신을 개시하지만, UDP에서는 접속을 설정하지 않고 데이터를 상대의 주소로 곧바로 송출한다. UDP의 특징은 접속을 설정하지 않고 일정한 간격으로 데이터를 전송하기 때문에 프로토콜 처리가 고속이라는 점이나, TCP와 같이 오류 정정이나 재전송기능을 가지고 있지 않기 때문에 목적지로 확실하게 전송된다는 신뢰성은 없다. 따라서, 신뢰성보다는 데이터의 고속처리가 요구되는 멀티미디어 응용 등에 적합하다.

3. RED 파라미터에 대한 동작 특성

RED를 적용하기 위해서는 RED내에서 정의하고 있는 4개의 파라미터 (*Wq*, *MAXp*, *MINth*, *MAXth*)가 RED라우터의 동작에 어떠한 영향을 미치고 있는지를 먼저 정확히 이해할 필요가 있다.

표1. RED 파라미터

Qlen	최대로 수용 가능한 Queue의 크기
MINth	폐기 여부를 결정하는 최소 임계값
MAXth	폐기 여부를 결정하는 최대 임계값
Wq	*Qavg*를 계산하는 가중치 값
MAXp	폐기 여부를 결정하는 최대 확률 값

3.1 평균 큐 길이와 큐 가중치

RED 알고리즘은 버퍼에서의 평균 큐 길이(*Qavg*)를 이용하여 네트워크의 혼잡상황을 미리 감지하고 제어하는 기능을 가지고 있다. *Qavg*를 구할 때 실제 큐 길이가 미치는 영향을 나타내는 값이 큐 가중치(*Wq*)이다. *Wq*가 작으면 현재 큐 실제 길이가 *Qavg*값에 미치는 영향이 작게 된다. 이런 경우에는 짧은 시간 동안 큐의 길이가 증가되더라도 *Qavg*가 증가하는데 미치는 영향이 작기 때문에 일시적인 트래픽의 증가로 인한 오류를 피할 수 있다. 하지만 *Wq*를 마냥 작게만 주는 것도 문제가 있다. *Qavg*가 *MAXth*를 넘었을 경우에 도착하는 모든 패킷을 폐기하기 때문에 이 패킷에 해당하는 TCP 연결을 갖는 송신 호스트는 모두 slow start 단계에 들어가기 때문에 전송 효율이 크게 떨어진다. 그래서 *Qavg*가 *MAXth*를 넘었을 때에는 가능한 빨리 *Qavg*가 *MAXth*밑으로 떨어져야 하는데 실제 큐 길이는 감소했다 하더라도 *Wq*가 너무 작으면 올라간 값이 떨어질 때에도 많은 시간이 필요하게 된다. 또한 *Wq*가 너무 클 경우에는 *Qavg*가 실제 큐 길이의 영향을 많이 받게 된다. 따라서 일시적인 버스트 트래픽에 대해서 바로 반응을 하게되고 DT방식과 다를 것이 없게된다.

3.2 최대 확률(*MAXp*)

*MAXp*가 커질 경우에 패킷 폐기 확률이 커져 패킷 폐기하는 횟수가 많아지므로 실제 큐 길이나 *Qavg*가 전체적으로 줄어들게 된다.

3.3 최소 한계값(*MINth*), 최대 한계값(*MAXth*)

평균 큐 길이가 *Qavg* 〈 *MINth* 이면 패킷을 큐에 넣고, 평균 큐 길이가 *Qavg* 〉 *MAXth*이면 패킷을 모두 폐기하고, *MINth* 〈 〈 *MAXth*이면 랜덤하게 패킷을 마킹하거나 폐기한다. 트래픽이 버스트한 특성을 갖을수록 *MINth*는 크게주어 링크의 높은 사용 효율을 유지할 수 있도록 해준다[1]. *MAXth*는 가급적 작게 주는 것이 유리하다. *MAXth*는 DT방식의 최대 버퍼크기와 비슷하게 동작하므로 이 값이 클 경우는 큐길이가 전체적으로 높게 유지되어 큐잉 지연 시간이 길어지게 된다. *MAXth*와 *MINth*의 차이는 왕복시간동안에 계산되는 평균 큐 길이의 추정치 보다 커야 RED 라우터는 효율적으로 동작할 수 있다.

3.4 RED파라미터 적용의 문제점

위와 같이 RED의 4가지 파라미터는 그 값에 따라 RED의 성능에 중대한 영향을 미친다. 하지만 어떠한 트래픽 상황에서도 범용적으로 적용할 수 있는 파라미터를 결정하는 것은 어려운 과제이다. 파라미터의 최적값은 트래픽의 상황에 따라 그 값을 변화할 수 있어야 한다.

4. 시뮬레이션

4.1 시뮬레이션 환경

제안된 기법의 시뮬레이션을 수행하기 위한 시뮬레이션 네트워크는 그림1과 같으며, UNIX 환경을 기반으로 NS2(Network Simulator)를 이용하여 수행하였다. 이 네트워크는 4개의 node(n0,n1,n2,n3)를 그림과 같이 구성한다. n0와 n2, n1과 n2 사이의 duplex link들은 10Mbps의 대역폭과 10ms지연을 가진다. n2와 n3사이의 duplex link는 1Mbps의 대역폭과 1ms지연을 가진다. TCP Agent는 n0에 접속되며 n3에 접속된 TCP Sink Agent와 연결된다. UDP Agent는 n1에 접속되며 연결은 n3에 접속된 Null Agent와 한다. n2와 n3를 연결하는 라우터에서는 최대 크기가 100인 DT 큐 방식과 RED 큐 방식을 사용한다. TCP Agent가 생성하는 최대 윈도우사이즈 크기는 64 이고 패킷의 최대크기는 1 Kbyte이다. FTP와 CBR 트래픽 생성기는 TCP와 UDP Agent에 각각 접속되어 있고 CBR은 1 Mbps의 속도에서 1 KByte 패킷을 생성하도록 구성되어 있다. FTP는 0.5초에 시작하고 CBR은 1초에 시작해서 두 생성기가 함께 15초에 멈추게 설정했다.

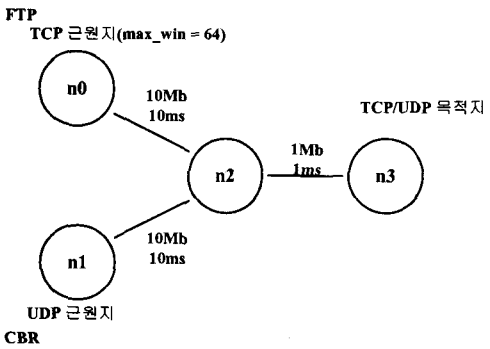

그림1. Network Topology

4.2 시뮬레이션 결과 분석

이 논문에서 큐잉 알고리즘을 DT방식과 RED방식으로 했을 때 데이터의 공정성이 변화하는 것을 검증했다. 그림2의 DT방식에서 TCP가 먼저 데이터를 전송하였지만 이후에 전송을 시작한 UDP가 9초 이후에는 모든 트래픽이 된다. 하지만 그림3의 RED 방식에서는 TCP의 전송량이 증가하여 공정성이 향상되는 것을 볼 수 있다.

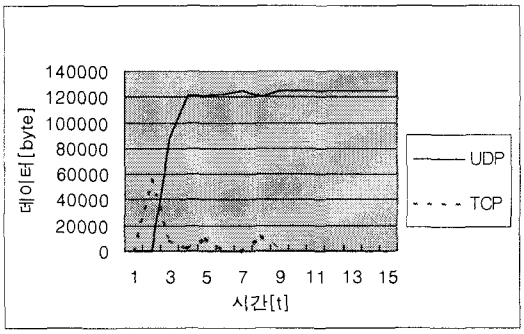

그림2. DropTail 알고리즘

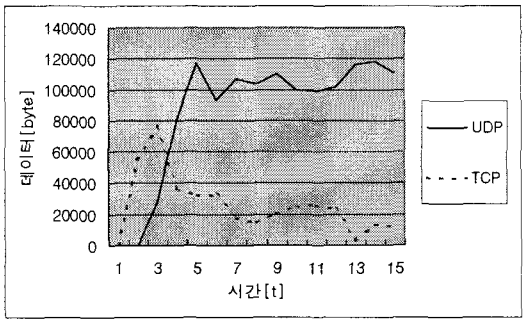

그림3. RED 알고리즘

본 시뮬레이션은 네트워크에서 능동적 폭주 회피 기법인 RED방식을 통해 TCP/UDP 혼합 환경에서 공정성이 향상되는 것을 보여준다. 하지만 연결수가 제한되어있고 RED파라미터 값이 고정되어 있어서 얻어진 결과는 더 다양한 시뮬레이션이 필요하고 문제점을 보완해야한다.

5. 결론 및 향후 연구 방향

폭주 회피 기법은 네트워크 트래픽 부하를 감시하여 병목지점에서의 폭주를 회피한다. 이것은 패킷 드롭핑에 의해 이루어지는데 현재 가장 널리 알려진 방법은 DT(DropTail)방식과 RED(Random Early

Detection)방식이다. 이 방식들은 큐 크기를 낮게 유지함으로써 폭주를 회피한다. 그러나 TCP/UDP의 혼합환경에서 네트워크 특성을 고려하지 않았을 때는 공정성을 떨어뜨리게 된다. 본 논문에서는 TCP/UDP 혼합 환경에서 트래픽의 공정성이 중간 노드인 라우터에서 DT방식을 사용했을 때 보다 RED방식을 사용했을 때 향상되는 것을 시뮬레이션을 통해 검증했다. 하지만 모든 트래픽 상황에 적응하는 고정된 RED 파라미터 값은 존재하지 않는다. 따라서 RED 파라미터 값을 변화하는 다양한 트래픽 환경에서의 시뮬레이션을 통해 분석하여 RED가 가지고 있는 다양한 문제에 접근하고, 해결방안을 연구할 것이다.

참고문헌

[1] 홍석원, 유영석, "체증 제어를 위한 Random Early Detection(RED) 알고리즘의 파라미터 분석", 통신학회 추계학술발표 논문집, 1997. 11

[2] 이정재, 최명렬, "네트워크 및 TCP 알고리즘 변화에 따른 TCP 공정성 분석", 한국통신학회, 2002. 7

[3] S. McCanne and S. Floyd. ns Network Simulator. http://www.isi.edu/nsnam/ns/.

[4] Sally Floyd and Van Jacobson, "Random Early Detection Gateway for Congestion Avoidance," IEEE/ACM Transaction on Networking, Aug. 1993.

[5] Mark Gaynor, "Proactive Packet Dropping Methods for TCP Gateways, "http://www.eecs.harvard.edu/~gaynor/final.ps.