

# 고속 네트워크에서 동적인 타이머 조절 기법을 통한 향상된 TCP-DAD

강동민\*, 박민우\*, 박선호\*, 정대명\*\*

\*성균관대학교 전자전기컴퓨터공학과

\*\*성균관대학교 정보통신공학부

e-mail : {dmkang, mwpark, shpark}@imtl.skku.ac.kr, tmchung@ece.skku.ac.kr

## Dynamic timer Mechanism for Advanced TCP-DAD at High-speed Network

Dong-Min Kang\*, Min-Woo Park\*, Sun-Ho Park\*, Tai-Myoung Chung\*\*

\*Dept. of Electrical and Computer Engineering, Sungkyunkwan University

\*\*School of Information Communication Engineering, Sungkyunkwan University

### 요 약

패킷 재배치(packet reordering)는 전송한 패킷의 순서가 처음과 다르게 뒤섞여 수신지에 도착하는 현상을 의미한다. 패킷 재배치는 불필요한 재전송이나 불필요한 혼잡 제어를 수행하여 TCP의 성능을 저하시킨다. 패킷 재배치에 의한 TCP 성능 저하를 막기 위해 다양한 접근 방법이 소개되었다. TCP-DAD는 기존 TCP에서 3으로 고정된 중복 응답 임계값을 동적으로 조절하여, 잘못된 혼잡 제어를 예방하고 있다. 일반적으로 전송한 패킷의 수가 많을 때, 패킷 재배치를 경험할 확률이 높으며, 패킷 재배치에 의해 잘못된 혼잡 제어가 일어날 가능성 많다. 고속 네트워크 환경에서는 혼잡 윈도우 크기 변화의 폭이 매우 크다는 점을 고려할 때, 중복 응답 임계값을 조절하는 TCP-DAD는 한계가 있다. 본 논문에서는 고속 네트워크 환경에서의 패킷 재배치 현상으로 인한 TCP 성능저하를 완화하기 위한 새로운 알고리즘 TCP-DT를 소개한다. 이는 중복 응답 수에 의존하지 않고 타이머를 통해 혼잡 제어를 수행하는 메커니즘이다. 본 논문은 NS-2를 사용하여 고속 네트워크 환경에서의 시뮬레이션을 통해 제안한 TCP-DT의 성능 향상을 증명하였다.

### 1. 서론

네트워크에서 가장 널리 쓰이는 TCP(Transmission Control Protocol)는 종단 프로세스 간에 연결지향과 신뢰성을 제공하는 전송계층 프로토콜이다. TCP는 흐름 제어, 에러 제어, 혼잡제어 메커니즘을 사용하며, 특히 혼잡제어의 신속 재전송과 신속 복구 메커니즘을 통해 네트워크에서 발생하는 혼잡현상에 유연하게 대응할 수 있는 프로토콜이다.

하지만, 오늘날 네트워크 장비의 발달로 인해 라우터의 병렬처리능력이 높아지고, 망의 부하 분산을 위한 다중 경로 라우팅, 라우트 플루터링 등의 네트워크의 부하 분산을 위한 다양한 기술이 개발되면서 패킷 재배치가 활발히 일어나게 되었다. 패킷 재배치란 패킷의 순서가 송신한 순서와 다르게 뒤바뀌어 수신지에 도착하는 현상을 말한다. 기존의 TCP 메커니즘은 신뢰성 있는 통신을 위해 응답 메커니즘을 사용한다. 이를 통해 무사히 도착한 패킷을 알리고 동시에 out of order나, 패킷 손실이 발생하였음을 송신측에게 알린다. 여러 개의 패킷이 송신한 순서와 다르게 뒤바뀌어 도착하게 되면 수신측은 중복된 응답을 여러 개 전송하게 된다. 이 경우 송신측은 네트워크가 혼

잡이 아님에도 불구하고 혼잡 제어를 수행하여 슬로우 스타트 임계값(Ssthresh)과 혼잡 윈도우(Congestion window)의 크기를 줄여 네트워크의 혼잡을 완화하려 한다. 따라서 해당 세션의 성능은 불필요하게 줄어들게 된다.

패킷 재배치에 의한 성능저하를 완화하기 위한 많은 연구가 있었으며 다양한 접근 방법이 제안되었다. 먼저, 중복 응답의 원인을 패킷 손실로 간주하여 빠른 재전송과 신속 복구 후 패킷 재배치에 의한 중복 응답이었다는 것이 증명되면 혼잡 윈도우의 크기를 복구하는 Eifel 알고리즘, 중복 응답에 대한 판단을 지연시켜 불필요한 재전송을 피하는 TCP-DCR(TCP Delayed Congestion Responds), 중복 응답에 대한 혼잡 제어 메커니즘은 수행하지 않고 오직 타이머를 이용한 패킷 재전송을 통해 패킷 재배치 문제를 해결하는 TCP-PR(TCP for persistent packet reordering)이 있다. 마지막으로 TCP 혼잡 제어의 트리거가 되는 중복 응답 임계값을 동적으로 조절하여 패킷 재배치에 의한 성능 저하를 완화하는 TCP-DAD(TCP dynamically adjusted dupthresh)와 같은 접근 방법이 있다.

TCP-DAD의 경우 중복 응답 임계값과 혼잡 윈도우가 독립된 관계를 가지기 때문에 혼잡 윈도우의 크

기 변화가 심하게 발생하는 고속망에서는 패킷 재배치에 의한 성능 저하를 제대로 해결할 수 없다. 또한 TCP-DCR 과 같이 혼잡 제어 수행 시점을 지연시키는 메커니즘은 패킷 손실에 유연하게 대처할 수 없는 한계점을 지닌다. 본 논문에서는 빠른 재전송과 동적 타이머(dynamic timer)를 통해 TCP-DAD 와 TCP-DCR 의 한계점을 개선하는 TCP-DT 를 제안한다. TCP-DT 는 중복 응답 임계값을 사용하는 대신에 패킷 손실을 판단하기 위해 동적 타이머를 도입한다.

본 논문의 구성은 다음과 같다. 2 장에서 관련연구로 TCP-DCR 과 TCP-DAD 메커니즘을 자세히 살펴보고 3 장에서 본 논문이 제안하는 dynamic timer 메커니즘에 대해서 소개하고 4 장에서 NS-2 를 이용한 시뮬레이션 결과를 통해 제안한 dynamic timer 메커니즘의 성능을 보이고 5 장에서는 결론과 함께 향후 연구 방향에 대하여 기술할 것이다.

2. 관련연구

2.1 TCP-DCR

TCP-DCR 은 RTT(Round Trip Time)가 만료되는 시간까지 신속 재전송과 신속 복구를 지연하는 메커니즘이다. RTT 시간까지는 중복 응답을 여러 개 수신해도 혼잡으로 판단하지 않고 패킷 재배치로 인식하여 혼잡 제어를 하지 않는다. TCP-DCR 은 패킷 손실이 일어나지 않는 환경에서 뛰어난 성능을 보인다. 하지만 단일 패킷 손실이 발생하는 환경에서는 RTT 시간까지 신속 재전송과 신속 복구를 지연시키기 때문에 전송 효율이 떨어질 뿐만 아니라, 혼잡 상황에 늦게 대처하여 다른 세션의 성능에도 영향을 미친다.

2.2 TCP-DAD

TCP-DAD 는 신속한 혼잡 제어를 수행하기 위한 트리거인 중복 응답 임계값을 동적으로 조절하여 혼잡 제어의 발생 시기를 지연하여 패킷 재배치로 인해 늦게 도착하는 응답을 기다리는 메커니즘이다. TCP-DAD 는 단일 패킷 손실이 발생하는 환경에서 TCP-DCR 보다 향상된 성능을 보인다.

TCP-DAD 에서 중복 응답 임계값은 초기에 다른 TCP 와 같이 3 부터 시작하며, 중복 응답 임계값 안에서 긍정 응답을 받으면 중복 응답 임계값을 올려주고 중복 응답 임계값을 넘어가면 신속 재전송과 신속 복구를 사용하여 재전송을 수행하고 중복 응답 임계값을 내려주어 현재 네트워크 상황에 대처 할 수 있는 메커니즘이다.

중복 응답을 받지 않더라도 검증 타이머가 만료되면 지연된 신속 재전송과 신속 복구를 통하여 재전송 및 혼잡 제어를 수행 한다. 이 경우의 중복 응답 임계값은 3 으로 초기화 한다.

TCP-DAD 의 검증 타이머는 RTT 크기로 설정되며 재전송한 패킷의 응답을 수신하는 시점에서 패킷 손실과 패킷 재배치를 구별하기 위해 사용된다.

TCP-DAD 는 중복 응답 임계값이 필요이상으로 커

지거나 작아질 때 불필요하게 재전송 타이머가 만료되거나 혼잡제어가 지연되어 TCP 성능 저하가 일어날 수 있다. 그리고, 중복 응답 임계값과 혼잡 윈도우의 크기가 독립적으로 설정되기 때문에 혼잡 윈도우의 크기 변화가 빈번하며, 그 크기의 폭이 큰 고속망에서는 적합하지 못하다.

3. TCP-DT

이 장에서는 본 논문에서 제안하는 dynamic timer 기법을 적용한 TCP-DT 메커니즘을 소개한다.

3.1 동작과정

아래의 <표 1>은 TCP-DT 의 동작 과정을 보여준다.

<표 1> TCP-DT 동작 과정

Event	Code
Initialization	cwnd = 1 ssthresh = +∞ dupthresh = 3 dynamic timer = RTT
Received dupacks	if(dupacks == 1) set detect timer else if(dupacks == dupthresh) FastRetransmit()
Received acks	if(received ack time <= dynamic timer) Calculate() else if(Received ack time > dynamic timer) FastRetransmit() FastRecovery() PreSsthresh = ssthresh PreCwnd = cwnd ssthresh = ssthresh / 2 cwnd = cwnd / 2
Detect timer expire	if(dynamic timer < Received ack < detect timer expire) ssthresh = PreSsthresh cwnd = PreCwnd dynamic timer = Calculate() else if(Not Received ack) dynamic timer = RTT
Calculate	dynamic timer = (Received ack time + Predynamic timer) / 2

본 논문에서 제안하는 TCP-DT 는 TCP-DAD 가 사용하는 중복 응답 임계값 대신에 기존 TCP 가 사용하는 중복 응답 임계값과 동적 타이머를 사용하는 메커니즘이다. TCP-DT 의 중복 응답 임계값은 3 을 가지며 중복 응답이 3 개 수신되었을 때 재전송을 하게 된다. 하지만 기존 TCP 와 달리 혼잡 제어는 하지 않는다. 동적 타이머는 패킷을 송신할 때 시작되는데 동적 타이머가 만료되기 전에 중복 응답을 수신하고 긍정 응답을 수신한 경우, 긍정 응답이 수신된 시간과 동적 타이머 만기시간의 평균을 계산하여 동적 타이머를 줄이게 된다. 동적 타이머가 만료 될 때까지 중복 응답만을 수신하고 긍정 응답을 수신하지 못한

경우에는 재전송과 혼잡 제어를 하게 된다. 혼잡 제어를 하기 전에 현재의 슬로우 스타트 임계값과 혼잡 윈도우 크기를 저장해 두는데, 이유는 이 후에 패킷 재배치로 판별 되었을 경우 현재의 상태로 다시 되돌리기 위해서이다.

TCP-DT 는 TCP-DAD 가 사용하는 검증타이머를 두는데 검증타이머는 첫 번째 중복 응답이 왔을 때 설정되고 크기는 RTT 시간이다. 검증 타이머는 패킷 손실과 패킷 재배치를 판별하는 기능을 한다.

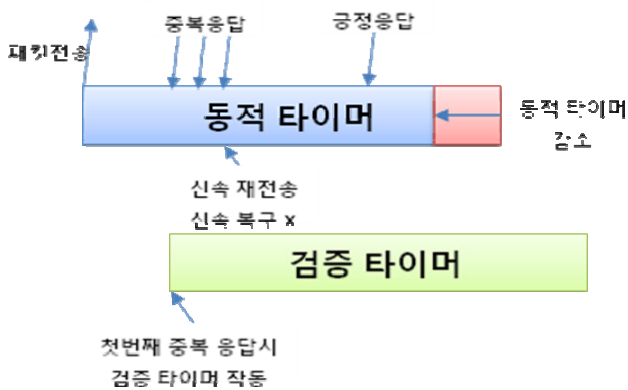
동적 타이머가 만기되어 재전송을 한 후 검증타이머가 만기되기 전까지 긍정 응답을 수신하게 되면 패킷 재배치라고 판별하여 동적 타이머가 만기 되었을 때 수행한 혼잡제어를 취소하고 저장해 놓은 슬로우 스타트 임계값과 혼잡 윈도우 값으로 변경한다.

만약 검증타이머가 만료 될 때 까지도 긍정 응답을 수신하지 못했다면 패킷 손실로 판별하여 기존에 수행했던 혼잡 제어를 그대로 유지하게 된다. 패킷 손실이 일어났을 때 동적 타이머는 초기 값인 RTT 로 변경된다.

### 3.2 시나리오

(그림 1)은 두 가지 시나리오를 보여준다. 먼저 기존의 TCP 와 같이 중복 응답을 3 개 수신하게 되면 곧바로 신속 재전송을 한다. 하지만, 혼잡 제어는 수행 하지 않는다. 그리고 첫 번째 중복 응답을 수신하면 검증 타이머가 작동한다.

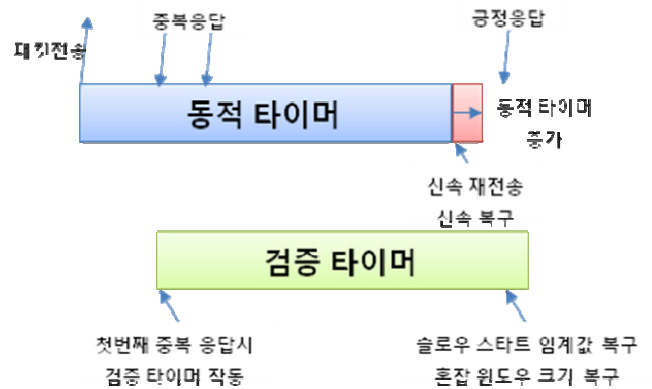
두 번째 시나리오는 동적 타이머가 만기되기 전에 긍정 응답이 수신 된 경우의 시나리오이다. 이 경우 긍정 응답이 수신된 시간과 동적 타이머의 평균시간으로 동적 타이머가 감소 된다. 동적 타이머는 패킷을 전송하고 긍정 응답을 수신 할 때마다 동적으로 변하게 된다.



(그림 1) 신속 재전송과 동적 타이머 슬라이딩

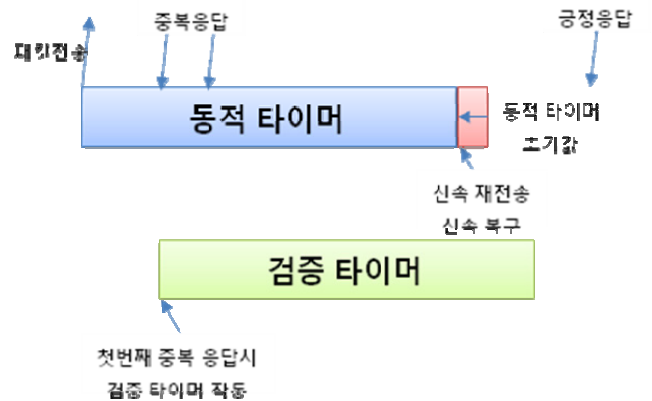
(그림 2)는 패킷 재배치를 패킷 손실로 판단한 경우의 시나리오이다. 이 경우 동적 타이머가 만기되었을 때 신속 재전송과 신속 복구 메커니즘이 수행된다. 하지만 검증 타이머가 만기되기 전에 긍정 응답을 수신하면 패킷 재배치로 판단하고 동적 타이머가 만기되어 줄인 슬로우 스타트 임계값과 혼잡 윈도우의 크

기를 원래의 값으로 복구 한다. 여기에서 동적 타이머는 증가 된다.



(그림 2) 패킷 손실이 패킷 재배치로 검증

(그림 3)은 검증 타이머 시간 안에 긍정 응답이 오지 않은 경우의 시나리오이다. 검증 타이머가 만기되기 전에 긍정 응답이 수신되지 않는다면 패킷 손실로 판단하여 검증 타이머는 과기되고 동적 타이머는 초기값을 갖는다.



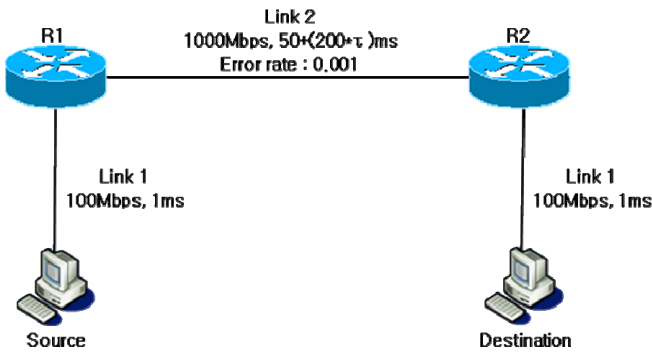
(그림 3) 패킷 손실

## 4. 성능평가

### 4.1 시뮬레이션 환경

NS-2 를 사용하여 본 논문에서 제안한 환경을 구현하고 시뮬레이션을 통하여 성능평가를 시행했다. 공정한 성능평가를 위해 TCP-DAD 와 같은 환경에서 아래의 (그림 4) 와 같이 토폴로지를 구성했다.

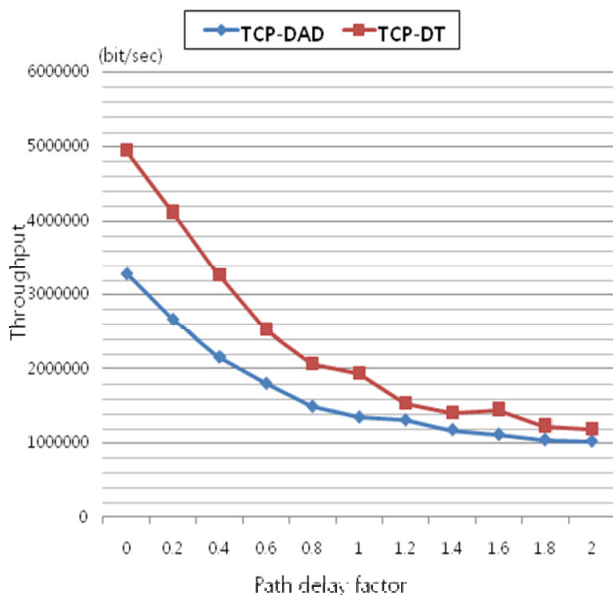
네트워크는 Source, Destination, 라우터 R1, 라우터 R2 로 4 개의 노드와 2 종류의 링크로 구성되어 있다. Link 1 은 100Mbps 대역폭, 1ms 의 전파지연을 가지며, Link 2 는 1000Mbps 대역폭, 평균 50+200 τ ms 의 전파지연을 가진다. 이때 표준 편차는 (200 τ)/3 ms 이며, Path delay factor 는 0 부터 2 사이의 값으로 0.2 간격으로 설정하였다. 오류 모델을 적용하였으며, 이때 오류율은 0.001 을 적용하였다.



(그림 4) 시뮬레이션 환경

네트워크 매개변수로 Source 와 Destination 에 400 세그먼트 크기의 버퍼를 설정 하였으며, 이때 세그먼트의 크기는 1500bytes 로 정의하였다. 또한 혼잡 윈도우의 최대 크기는 500 세그먼트로 설정하였다. 패킷 재배치를 발생시키기 위해 특정 시간 간격으로 패킷의 순서를 바꾸어 보냈고, 특정 시간 간격은 50ms 의 시간으로 시뮬레이션 했다. 10 초에 전송을 시작하여 1010 초에 전송을 종료하는 방법으로 총 1000 초간 시뮬레이션을 진행하였다. 패킷 재배치는 난수에 따라 발생하는데 이때 씨드 값(seed value)은 총 20 가지 다른 값을 적용하였다.

#### 4.2 시뮬레이션 결과



(그림 5) 시뮬레이션 결과

시뮬레이션 결과는 위의 (그림 5) 와 같다. 시뮬레이션 결과 TCP-DT 가 TCP-DAD 에 비해 평균 약 7% 높은 성능을 냈다. 이러한 성능차이는 TCP-DAD 의 경우 중복 응답 임계값으로 패킷 재배치와 패킷 손실을 결정하는데 고속 네트워크 환경에서는 더 많은 패킷 재배치가 발생한다. 그러므로 중복 응답 임계값이 초과 될 때마다 패킷 손실이 아닌 경우에도 패킷 손실

로 판단하고 혼잡 제어를 수행하여 대역폭을 전부 사용하지 못하게 된다. 그에 비해 TCP-DT 는 패킷 재배치 문제를 중복 응답의 수에 의존하지 않고 동적 타이머를 적용하여 고속 네트워크 환경에서의 패킷 재배치 문제를 해결하였으며, 고속 네트워크의 장점인 넓은 대역폭을 모두 사용하여 송신하였기에 시뮬레이션 환경에서 보다 높은 성능을 나타내었다.

이러한 시뮬레이션 결과를 통해서 TCP-DT 가 기존의 TCP-DAD 보다 향상된 성능을 보이는 것을 확인할 수 있다.

#### 5. 결론

본 논문에서는 고속 네트워크 환경에서의 패킷 재배치 현상으로 인한 TCP 성능저하를 완화하기 위해 새로운 알고리즘 TCP-DT 를 제안하였다. 이는 기존의 패킷 재배치로 인한 중복 응답을 혼잡으로 판단하여 불필요하게 신속 재전송과 신속 복구 메커니즘을 수행하여 성능 저하를 일으켰다. 이 점을 보완하기 위해 TCP-DAD 는 동적인 중복 응답 임계값을 사용하여 패킷 재배치 문제를 해결 했다. 그러나 TCP-DAD 의 중복 응답 임계값은 패킷 재배치가 많이 발생하고 중복 응답이 많은 고속 네트워크 환경에서 효율적이지 못했다. 그래서 TCP-DT 는 패킷 재배치 문제를 중복 응답의 수에 의존하지 않고 동적 타이머에 적용하여 고속 네트워크 환경에서도 패킷 재배치 문제를 완화하였으면 NS-2 시뮬레이션을 통해 이를 증명하였다.

향후 연구로는 TCP-DT 의 동적 타이머가 현재 네트워크 상황을 반영하고 최적화 된 성능을 보일 수 있는 동적 타이머 및 검증 타이머의 크기에 관한 연구를 수행 할 것이다.

#### 참고문헌

- [1] K.-C. Leung, O. K. Li and D. Yang, "An Overview of Packet Reordering in Transmission Control Protocol (TCP): Problems, Solutions, and Challenges," IEEE TRANSACTIONS ON PARALLELAND DISTRIBUTED SYSTEMS, vol. 18, no. 4, Apr. 2007.
- [2] 박민우, 김종명, 한영주, 권윤주, 이길재, 정태명, "중복 응답 임계값의 완만한 조절 기법을 통한 향상된 TCP-DAD", 한국정보처리학회 추계학술발표대회, vol15, issue 1, May 2008.
- [3] 박민우, 이제민, 김종명, 한영주, 정태명, "TCP-DAD: 중복 응답 임계값의 동적 조절을 통한 TCP 의 패킷 재배치 내성 향상 기법", 한국정보처리학회 추계학술발표대회, vol14, issue 2, Nov 2007.
- [4] S. Bhandarkar, et al., "TCP-DCR: A Novel Protocol for Tolerating Wireless Channel Errors," IEEE TRANSACTIONS ON MOBILECOMPUTING, vol.4, no.5, Sep.2005.
- [5] E. Blanton and M. Allman, dzik, "Improving the Performance of TCP in the Case of Packet Reordering", Leication Review, vol.32, issue 1, p.20-30, Jan. 2002.