

무선 환경에서 TCP 스푸리어스 타임아웃 방지를 위한 SR-ARQ 재전송 지속성 관리 방안

김 범 준[†] · 한 제 찬^{**}

요 약

무선 링크를 통한 데이터 전송 과정에서 오류 혹은 그 밖의 다른 이유로 빈번하게 발생하는 패킷 손실을 감지하고 재전송하기 위한 기능은 전송의 신뢰성 확보 차원에서 매우 중요하다. 따라서 대부분의 이동통신시스템들은 데이터 링크 계층에서 동작하는 자동 재전송 프로토콜 (Automatic Repeat reQuest; ARQ)을 도입하고 있다. 그러나 자동 재전송 프로토콜의 재전송 기능과 TCP(Transmission Control Protocol)의 재전송 기능 간 원활하지 못한 상호작용은 오히려 TCP의 성능을 저하시킬 수 있는 문제가 있음이 알려져 왔다. 따라서 본 논문에서는 가장 널리 사용되고 있는 자동 재전송 프로토콜인 SR-ARQ(selective repeat ARQ)를 TCP와의 상호 작용 측면에서의 개선하기 위한 방안을 제시하고 OPNET을 이용한 시뮬레이션을 통하여 이로 인한 TCP 성능 향상을 입증한다.

키워드 : 선택적 자동 재전송, 재전송 지속성, TCP 스푸리어스 타임아웃

SR-ARQ Retransmission Persistence Management to Avoid TCP Spurious Timeout in a Wireless Environment

Kim, Beomjoon[†] · Jechan Han^{**}

ABSTRACT

To detect and recover packet losses over wireless links is very important in terms of reliability in packet transmission. Most wireless communication systems adopt an automatic repeat request (ARQ) protocol operating at link layer. However, it has been constantly addressed that the interaction not harmonized sufficiently between ARQ and TCP rather degrades TCP performance. In this paper, therefore, we propose an improved scheme from the aspect of the interaction with TCP loss recovery mechanism that can be applied to selective repeat ARQ (SR-ARQ) protocol and prove that the proposed scheme improves TCP performance significantly by OPNET simulations.

Keywords : SR-ARQ, Retransmission Persistence, TCP Spurious Timeout

1. 서 론

대부분의 이동 통신 시스템에서는 무선 링크를 통한 데이터 전송의 신뢰성을 확보하기 위해서 데이터 링크 계층에서 동작하는 자동 재전송(Automatic Repeat reQuest; ARQ) 프로토콜을 도입하고 있다. 다양한 자동 재전송 프로토콜들 가운데 Selective Repeat ARQ (SR-ARQ) 프로토콜은 그 효율성 측면에서 성능이 우수하기 때문에 실제 이동 통신 시스템에 널리 사용되고 있다[1, 2]. SR-ARQ 프로토콜은 전송에 실패한 패킷들만을 선택적으로 재전송하는 복구 과

정을 수행하는데 이러한 복구 과정으로 인하여 패킷 전송 지연과 지연 변이가 증가할 수 있다. 복구 과정에서 증가한 지연과 지연 변이는 상위 계층에서 동작하는 프로토콜인 TCP (Transmission Control Protocol)의 성능에 상당히 부정적인 영향을 미칠 수 있는데 링크 계층에서의 급격한 전송 지연의 증가로 인해 발생하는 스푸리어스 타임아웃을 가장 대표적인 예로 들 수 있으며 이 외에도 전반적인 TCP와 링크 계층의 재전송 프로토콜간의 상호 동작에도 심각한 문제로서 작용할 수 있다[3, 4, 17]. 최근 인터넷이 무선으로 급속하게 확장되고 있는 현실을 생각했을 때 이 문제는 매우 중요한 문제라 할 수 있다. 따라서 본 논문에서는 TCP의 성능 향상을 궁극적인 목표로 SR-ARQ 프로토콜의 개선 방안을 제안하고 OPNET 시뮬레이션을 통하여 제안된 개선 방안이 TCP 성능 향상에 기여할 수 있음을 입증하고자 한다.

[†] 정 회 원 : 계명대학교 전자공학과 교수(교신저자)

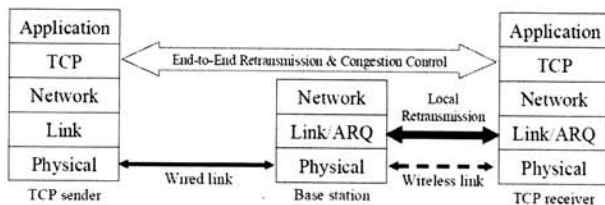
^{**} 준 회 원 : LG전자 MC연구소 선임연구원
논문접수: 2010년 2월 24일
수정일: 1차 2010년 5월 25일
심사완료: 2010년 6월 17일

2. ARQ와 TCP

2.1. ARQ와 TCP 상호 동작

(그림 1)은 사용자 단말과 네트워크의 마지막 접속 지점이 무선 링크로 연결되는 전형적인 이동 통신 시스템 상에 구현된 무선 인터넷의 종단 간 프로토콜 구조를 보여준다. 흔히 기지국이라고 하는 네트워크의 마지막 접속 지점과 사용자 단말은 무선 링크로 연결되어 있기 때문에 기존의 유선 구간에 비하여 전송 조건이 훨씬 더 열악하여 패킷손실률이 상당히 높은 값을 보이게 된다. 이러한 높은 패킷손실률을 극복하기 위해서 무선 링크 상에서는 특화된 ARQ 프로토콜이 동작하여 발생하는 패킷¹⁾ 손실을 지역적으로 재전송하여 복구하는 역할을 수행한다. 이와는 별도로 상위 프로토콜인 TCP의 혼잡 제어 기능은 손실된 패킷을 감지하여 이를 재전송에 의해 복구하는 손실 복구 기능을 포함한다 [5]. 손실된 패킷에 대한 재전송을 수행한다는 점에서 ARQ와 TCP는 유사하다 할 수 있겠으나 ARQ는 동작 범위가 무선 링크에 한정되어 있는 반면 TCP는 종단간 동작한다는 점에서 근본적인 차이가 있다.

(그림 1)에서 볼 수 있는 프로토콜 구조에서 TCP와 ARQ는 상호 보완적으로 동작하는 것이 이상적이다. 이에 대한 이유는 TCP 혼잡 제어의 기본 작동 원리에서 찾을 수 있는데 TCP는 패킷 손실 발생의 여부를 기준으로 네트워크에 혼잡이 발생했는지를 판단하도록 설계되어 있기 때문이다 [5]. 반대로 말하자면 TCP는 패킷 손실이 발생하지 않는 한 네트워크가 혼잡하지 않다고 판단하여 전송량을 지속적으로 증가시키고 언젠가 패킷 손실이 발생하면 그제야 네트워크가 혼잡하다고 판단하여 전송량을 대폭 감소시킨다. 이와 같은 TCP의 혼잡 제어 동작은 전송 링크 상에서 패킷 손실이 거의 발생하지 않는 유선 네트워크에서는 상당히 잘 들어맞을 수 있다. 그러나 무선 링크 상에서는 혼잡 상황이 아니더라도 패킷 손실이 발생할 수 있기 때문에 모든 패킷 손실을 혼잡 상황으로 판단하여 전송량을 감소시킬 수밖에 없는 현재의 TCP 성능은 크게 저하될 수 있는 것이다. 이와 같이 혼잡 상황에서 발생하지 않는 패킷 손실을 흔히 비혼잡 패킷 손실(non-congestion packet loss)이라 하는데 비혼잡 패킷 손실에 의한 TCP 성능 저하를 피하기 위한 많은



(그림 1) 이동 통신 시스템을 통한 무선 인터넷의 종단 간 프로토콜 구조

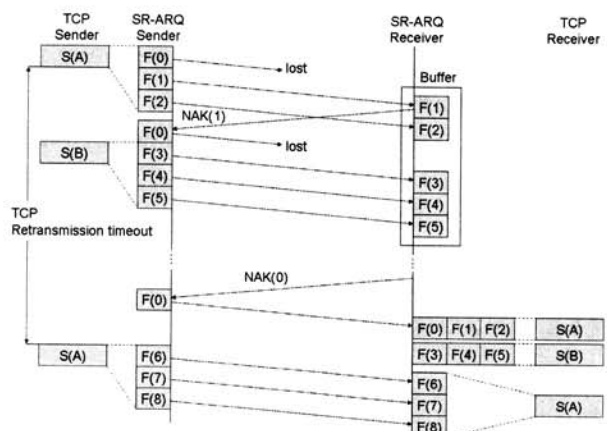
1) ARQ 계층의 전송 단위는 프레임이므로 엄밀하게는 패킷이 아닌 ARQ 프레임이라 하는 것이 옳으나 본 논문에서는 편의상 넓은 의미의 패킷이라는 용어를 사용하였다. 마찬가지로 TCP의 전송 단위는 세그먼트라 하는 것이 옳으나 역시 패킷이라는 용어를 포괄적으로 사용하였다.

연구가 진행되어 왔음에도 불구하고 아직까지 뚜렷한 해결 방안이 나타나지 않은 상태이다[6].

제안된 해결 방안 중 가장 이상적인 하나는 ARQ가 무선 링크에서 발생하는 거의 모든 패킷 손실을 감지하여 복구하도록 하는 것이다. 만약 이것이 가능하다면 TCP는 원래의 동작 방식 그대로 패킷 손실을 네트워크의 혼잡의 발생으로 간주하더라도 큰 무리가 없을 것이다. 이런 관점에서 앞서 비슷한 기능을 수행하는 ARQ와 TCP의 동작이 상호 보완적이어야 한다고 언급한 것이다. 그러나 실제로는 ARQ의 손실 복구 기능을 강화하는 것만이 반드시 좋을 수만은 없는데 그에 대한 이유는 (그림 2)로 설명되어 질 수 있다.

(그림 2)는 TCP 송신원이 두 개의 TCP 패킷 S(A)와 S(B)를 전송하는 과정을 ARQ와 TCP와의 상호 작용 관점에서 나타낸 것이다. 편의상 하나의 TCP 패킷은 세 개의 ARQ 패킷으로 나뉘고 ARQ는 NAK 기반의 SR-ARQ로 동작한다고 가정한다.

- 첫 번째 TCP 패킷을 구성하는 세 개의 ARQ 패킷 가운데 첫 번째인 F(0)가 손실되었고 나머지 두 개의 ARQ 패킷은 정상적으로 SR-ARQ 수신원에게 전달되었다.
- F(1)을 수신한 SR-ARQ 수신원은 F(0)가 손실되었음을 감지하고 이에 대한 재전송을 요청하는 NAK(0)를 SR-ARQ 송신원에게 전달한다.
- SR-ARQ는 재전송이 항상 새로운 전송에 우선하기 때문에 NAK(0)를 수신한 SR-ARQ 송신원은 F(0)를 즉시 재전송하였으나 다시 손실되었다.
- 이후 SR-ARQ 송신원은 두 번째 TCP 패킷을 구성하는 세 개의 ARQ 패킷인 F(3), F(4), F(5)를 순차적으로 전송하고 정상적으로 SR-ARQ 수신원에게 수신되었다.
- F(0)가 수신되지 않은 상태에서 수신된 높은 순번의 ARQ 패킷들은 상위 프로토콜로 전달되지 못하고 SR-ARQ 수신원의 버퍼에 저장되어 있다. 따라서 TCP 수신원은 아직까지 하나의 TCP 패킷도 수신하지 못한 상태이다.
- SR-ARQ 수신원의 재전송 타이머가 만료되어 다시



(그림 2) ARQ와 TCP의 상호 작용

NAK(0)를 SR-ARQ 송신원에게 전송하였고 재전송된 F(0)가 이번에는 정상적으로 SR-ARQ에 도달하였다. 따라서 두 개의 TCP 패킷인 S(A)와 S(B)가 TCP 수신원에 넘겨진다.

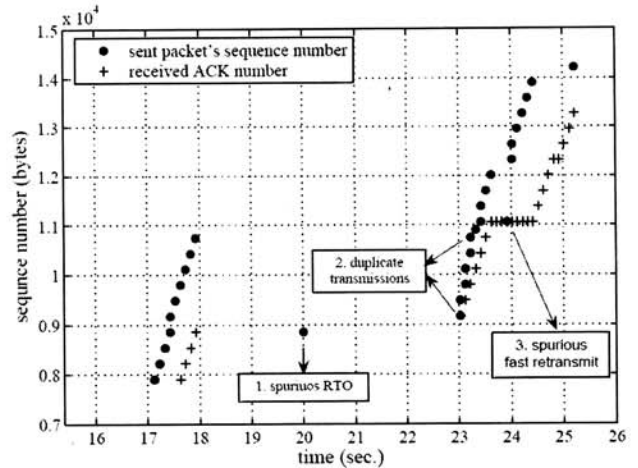
- 그런데 이와 거의 동시에 아직까지 아무런 응답을 받지 못한 TCP 송신원의 재전송 타이머가 만료되면 TCP 송신원은 S(A)를 무조건 재전송하게 되고 이는 세 개의 새로운 ARQ 패킷 F(6), F(7), F(8)로 나뉘어 전송된다.

(그림 2)에 나타난 동작은 다소 극단적이긴 하지만 SR-ARQ의 재전송 기능이 TCP와 조화되지 못하면 비록 ARQ에 의한 재전송이 성공하더라도 불필요하게 S(A)를 중복으로 수신하는 등 결과적으로는 TCP의 성능에는 오히려 나쁜 영향을 미칠 수 있음을 설명하고 있다. 이와 같은 현상을 TCP 스푸리어스 타임아웃이라 하는데 이에 대해서는 다음 절에 상세히 설명한다.

2.2 TCP 스푸리어스 타임아웃

인터넷의 대표적인 전송계층 프로토콜인 TCP는 전송의 신뢰성 확보를 위하여 손실된 패킷을 감지하고 이를 재전송에 의해서 복구하는 손실 복구 기능을 수행하는데 가장 기본적인 복구 방법은 시간에 기초한 방법이다[7]. 이 방법은 패킷을 전송하고 이에 대한 응답이 오기까지의 종단왕복지연(Roundtrip Time; RTT)을 지속적으로 측정하고 이에 근거하여 흔히 재전송 타임아웃이라고 하는 일종의 한계 값을 산출하여 패킷을 전송한 후 재전송 타임아웃 이내에 응답이 수신되지 않으면 손실되었다고 간주하고 이에 상응하는 조치를 취하게 된다. 이 방법은 종단왕복지연의 편차가 그리 크지 않은 상황에서는 무난하게 동작할 수 있는 반면에 종단왕복지연의 편차가 매우 큰 경우에는 불필요한 재전송 타임아웃이 발생하는 문제점이 있다. 이와 같이 패킷 손실이 발생하지 않았음에도 응답이 수신되기까지의 지연이 일시적으로 급격하게 증가하여 불필요하게 발생하는 재전송 타임아웃을 스푸리어스 타임아웃이라 한다[8].

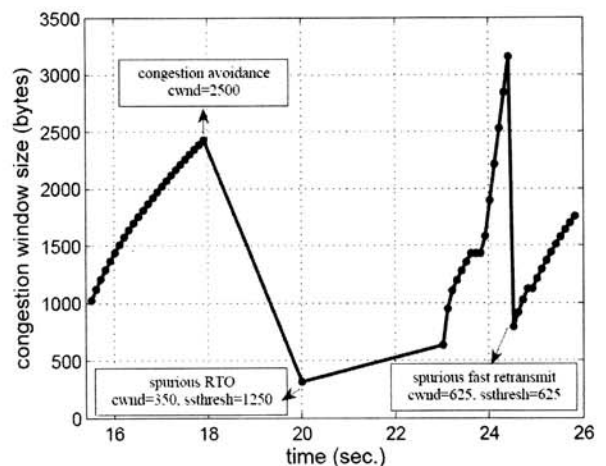
(그림 3)에는 스푸리어스 타임아웃이 발생한 경우 TCP 전송 과정을 나타낸 것이다. 약 17초에서 18초 사이에 총 10개의 패킷이 정상적으로 수신원에 전송되었고 이에 대한 응답이 수신되는 과정에서 다섯 번째부터 마지막 패킷에 대한 여섯 개의 응답의 수신 지연되고 있다. 수신이 계속 지연되면 약 20초경에 스푸리어스 타임아웃이 발생하여 네 번째 패킷이 불필요하게 중복 전송되는 것을 볼 수 있다. 이후 약 23초경부터 앞서 지연된 여섯 개의 응답이 차례로 수신되기 시작하는데 TCP 송신원은 이들 중 첫 번째 응답을 스푸리어스 타임아웃에 의해서 20초경에 재전송한 패킷에 대한 응답으로 해석할 수밖에 없기 때문에 이미 정상적으로 전송된 패킷들이 다시 불필요하게 중복 전송되게 된다. 이와 같이 불필요하게 전송된 패킷들은 잘못된 세 개의 중복 응답이 수신되도록 하기 때문에 이 후에 불필요한 fast retransmit과 fast recovery 과정이 수행되게 된다.



(그림 3) 스푸리어스 타임아웃에 따른 TCP 패킷 전송 과정

(그림 3)을 통해서 설명한 바와 같이 스푸리어스 타임아웃이 일단 발생하게 되면 TCP 송신원은 이미 정상적으로 전송한 패킷들을 불필요하게 중복 전송해야 할 뿐만 아니라 윈도우의 크기를 불필요하게 감소시키게 되어 TCP의 성능이 크게 감소하게 된다. 이를 설명하기 위하여 (그림 4)에는 (그림 3)에 나타난 TCP 패킷 전송 과정 동안 TCP 송신원의 혼잡윈도우(congestion window; cwnd)와 slow-start threshold(ssthresh)의 값의 변화를 나타낸 것이다. 그림에서 볼 수 있는 바와 같이 18초경 congestion avoidance 상태에서 2500바이트에 달했던 혼잡윈도우는 스푸리어스 타임아웃에 의해서 20초경에는 350바이트로 줄어들고 이 후 지연된 응답이 수신됨에 따라서 차츰 증가하다가 다시 잘못된 중복 응답 수신으로 인한 fast retransmit과 fast recovery 과정을 거쳐 결국 25초경에는 혼잡윈도우와 ssthresh의 값이 625바이트로 크게 감소한 상태에서 congestion avoidance가 시작되는 것을 확인할 수 있다.

앞의 두 그림을 통해서 살펴본 바와 같이 스푸리어스 타임아웃은 일단 발생하면 TCP의 성능이 크게 저하된다. 유선 네트워크 등의 안정적인 환경에서 전송이 이루어진다면



(그림 4) 스푸리어스 타임아웃에 따른 TCP 윈도우의 변화

스플리어스 타임아웃의 발생 가능성은 매우 낮지만 무선 전송 환경인 이동 통신 시스템이라면 이야기는 달라질 수 있다. 이동 통신 시스템에서 핸드오버 과정 혹은 앞의 (그림 2)를 통해 설명한 SR-ARQ 프로토콜의 재전송에 의하여 일시적으로 급격하게 증가하는 지연은 스플리어스 타임아웃의 가능성을 증가시킬 수 있는 요인으로 작용할 수 있기 때문이다. 물론 스플리어스 타임아웃을 TCP 송신원이 감지하고 이에 대한 적절한 조치를 취할 수 있도록 하기 위한 방안이 이미 두 개나 제시되어 표준화가 완료된 바 있다[8, 9]. 그러나 이 방안들은 스플리어스 타임아웃이 적어도 한 번 발생한 후에야 이를 감지할 수 있기 때문에 본 논문에서 제시하고자 하는 방안에 비해 엄밀한 의미에서 근본적인 해결책이라 하기 어렵다.

3. 제안하는 재전송 지속성 관리 방안

3.1 기존 방안의 문제점

2장에서 살펴본 문제점과 관련하여 지금까지 SR-ARQ 프로토콜을 개선하고자 하는 많은 연구가 있어왔다[10-14, 17]. 이 연구들은 공통적으로 주어진 상황이나 조건에 따라서 단력적으로 재전송 지속성(retransmission persistence)²⁾을 관리하는 방안을 제안하고 있고 이것이 기존의 재전송 지속성을 고정하는 방식보다 더 좋은 성능을 나타냄을 보였다. 대표적인 예로서 [11, 12]의 연구에서 SR-ARQ는 무선 링크 상에서 발생하는 패킷손실률을 측정하고 이를 기반으로 측정된 패킷손실률이 기준 패킷손실률보다 높은 경우에는 재전송 지속성을 증가시키고 반대로 측정된 비트손실률이 기준 패킷손실률보다 낮은 경우에는 재전송 지속성을 감소시키는 방안을 제안하였고 이를 통하여 TCP의 성능이 향상될 수 있음을 보였다.

그러나 대부분의 기존 연구들은 재전송 지속성을 관리함에 있어서 무선 링크 상에서 발생하는 패킷손실률에 너무 많이 의존한다는 문제점이 있다. 예를 들어 패킷손실률이 증가함에 따라 재전송 지속성을 계속 증가시킨다고 가정해보자. 그러면 SR-ARQ 프로토콜은 손실된 ARQ 패킷에 대해서 많은 수의 재전송을 반복해서라도 전송하고자 할 것이고 그 결과 SR-ARQ 프로토콜의 처리율은 증가할 것이기 때문에 성능 역시 좋아질 것이라 생각할 수 있다. 그러나 위와 같은 SR-ARQ 프로토콜에 의한 과도한 횡수의 재전송 수행은 그 ARQ 패킷이 속한 TCP 패킷에 대한 스플리어스 타임아웃을 유발할 가능성이 높아지게 된다. 더욱이 앞서도 언급한 바와 같이 대부분의 SR-ARQ 프로토콜은 재전송을 항상 신규 전송에 우선하여 수행하도록 설계되어 있으므로 많은 수의 재전송을 수행하는 것은 이 후에 전송되는 신규

ARQ 패킷에 대한 전송 지연 역시 증가할 것이고 이로 인한 스플리어스 타임아웃의 가능성은 더 높아질 수밖에 없음을 의미한다. 오히려 이런 경우에는 비록 무선 링크상의 패킷손실률이 높다고 하더라도 손실된 ARQ 패킷의 SR-ARQ 프로토콜에 의한 재전송을 조기에 포기함으로써 스플리어스 타임아웃을 방지하고 추후에 TCP fast retransmit과 fast recovery에 의한 손실 복구 기능이 동작하도록 하는 편이 TCP 처리율 측면에서는 훨씬 더 유리할 수 있다는 것이다. 이 사실은 2.2절에서 설명한 바와 같이 스플리어스 타임아웃이 발생하면 TCP 송신원은 혼잡윈도우의 값을 1로 줄여야 하는 반면 fast retransmit과 fast recovery에 의해서 복구하게 된다면 패킷 손실 당시의 혼잡윈도우 크기의 절반으로만 줄이면 된다는 점을 생각한다면 쉽게 이해될 수 있다. (그림 2)에 나타난 동작의 예에서는 SR-ARQ에 의한 F(0)의 복구를 포기하고 SR-ARQ 수신원이 두 번째 TCP 패킷인 S(B)를 TCP 수신원에게 전달하였다면 적어도 스플리어스 타임아웃은 피하는 것이 가능하다.

이러한 사실에 착안하여 본 연구과제에서는 재전송 지속성을 제어함에 있어 패킷손실률뿐만 아니라 SR-ARQ 송신원의 큐의 길이 및 트래픽 도착률 등의 종합적인 트래픽 부하를 고려함으로써 스플리어스 타임아웃의 발생을 최소화할 수 있는 SR-ARQ 프로토콜의 재전송 지속성 관리 방안을 제안하고자 한다. 최근에 발표된 연구 가운데 [17]의 제안이 본 논문에서 해결하고자 하는 바와 거의 유사하여 참고할 만하였다. 본 논문에서와 같이 이 연구에서도 ARQ 계층에서의 지나친 재전송 시도가 결과적으로는 TCP 성능을 감소시킬 수 있음을 지적하였다. 다만 [17]은 H-ARQ (Hybrid ARQ)의 사용을 전제로 하고 있기 때문에 제시된 해결책이 재전송 지속성이 아닌 전송 블록에 포함되는 FEC(Forward Error Correction) 코드의 길이를 패킷손실률에 따라서 가변적으로 조절한다는 점에서 본 논문과 차이가 있다. 그러나 ARQ에 의한 재전송 횡수를 제한한다는 점에서는 본 연구의 접근 방법과 일맥상통하는 점이 있다고 할 수 있다.

3.2 제안하는 재전송 지속성 관리 방안

SR-ARQ 송신원이 감당할 수 있는 최대 트래픽 부하를 σ , 신규로 도착하는 ARQ 패킷들의 트래픽 부하를 ρ_{new} , 재전송 대상인 ARQ 패킷들에 의한 트래픽 부하를 ρ_{rtx} 이라 했을 때 본 논문에서 제안하는 재전송 지속성 관리 방안의 원칙은 아래 식 (1)으로 주어진다.

$$\rho_{new} + \rho_{rtx} \leq \sigma \quad (1)$$

식 (1)이 의미하는 바는 신규로 도착하는 패킷들에 의한 부하와 재전송 패킷들에 의한 부하의 합이 SR-ARQ 송신원이 감당할 수 있는 최대 트래픽 부하를 넘지 않아야 한다는 것이다. 만약 두 종류의 트래픽 부하의 합이 σ 를 넘어서게 되면 각 SR-ARQ 패킷이 겪게 되는 지연과 큐의 길이가 지수적으로 급격하게 증가하게 된다.

2) 본 논문에서 재전송 지속성(retransmission persistence)은 ARQ 프로토콜이 손실된 ARQ 패킷을 재전송에 의해서 복구하고자 하는 정도를 의미한다. 예를 들어 재전송 지속성이 높으면 많은 수의 재전송을 통해서라도 손실된 ARQ 패킷을 반드시 복구하겠다는 의지가 강한 것이고 재전송 지속성이 낮다면 허용되는 재전송 횡수가 적어 반드시 이를 복구하겠다는 의지가 약한 것이다.

그런데 일반적으로 ρ_{new} 의 값은 상위 프로토콜로부터 SR-ARQ로 유입되는 데이터양에 의해 결정되기 때문에 SR-ARQ가 제어할 수 있는 값이 아니다. 반면 ρ_{rtx} 의 값은 SR-ARQ에 의해서 손실된 패킷을 얼마의 재전송 지속성을 가지고 재전송할 수행할지의 여부에 따라서 결정되는 값이다. 다시 말해서 높은 재전송 지속성에 따라 손실된 패킷에 대한 재전송 반복 횟수를 증가시킬수록 ρ_{rtx} 의 값 역시 증가한다는 뜻이다. 따라서 식 (1)로 주어지는 두 종류의 트래픽 부하 간의 관계는 다음과 같이 설정되어 질 수 있다.

- 만약 ρ_{new} 의 값이 작은 경우라면 ρ_{rtx} 은 ρ_{new} 과의 합이 σ 를 넘지 않는 한 큰 값을 가질 수 있다.
- 반면 ρ_{new} 의 값이 큰 경우라면 ρ_{rtx} 의 값은 상대적으로 작게 유지되어야 하고 심지어 ρ_{new} 이 거의 σ 와 같거나 일시적이더라도 넘는다면 ρ_{rtx} 은 0이 되어야 한다.

이는 신규로 유입되는 패킷의 수가 적을 때에는 충분한 수의 재전송을 수행하는 것이 가능하지만 신규로 유입되는 패킷의 수가 많을 때에는 비록 무선 링크의 상황이 좋지 못해 손실되는 ARQ 패킷이 많다고 하더라도 허용될 수 있는 재전송 반복 횟수는 제한될 수밖에 없음을 의미하는 것이다. 그렇지 않고 기존의 연구에서처럼 신규로 유입되는 패킷의 수가 많은데도 단지 무선 링크의 상황이 좋지 못하다는 이유로 높은 재전송 지속성을 유지한다면 급격하게 증가하는 지연으로 인하여 스푸리어스 타임아웃의 발생 가능성이 높아지게 된다. 결과적으로 재전송 반복 횟수를 r 이라 했을 때 제안하는 방안이 r 을 결정하기 위한 원칙은 식 (1)을 변형한 식 (2)로 주어진다.

$$\rho_{rtx} \leq \sigma - \rho_{new} \tag{2}$$

SR-ARQ 송신원은 현 시점의 $\sigma - \rho_{new}$ 의 값을 측정하고 이를 기반으로 식 (2)를 만족하는 ρ_{rtx} 의 값을 결정하여 최종적으로 ρ_{rtx} 로부터 r 값을 유도하게 된다. 이를 위해서 무선 채널의 변화에 따른 SR-ARQ의 성능을 수학적으로 분석한 기존의 연구들 가운데 대표적인 두 연구[15, 18]를 주로 참조하였다. 특히 [18]은 M/G/1 모델을 이용하여 무선 링크 상에서 발생하는 패킷손실률을 p 라 했을 때 SR-ARQ의 큐잉 지연과 큐의 길이가 유도되는 과정을 상세히 설명하고 있다. 최종적으로 [18]에서 유도된 ρ_{new} 와 ρ_{rtx} 를 식 (2)에 대입함으로써 얻어지는 최종 결과는 다음의 식 (3)으로 주어진다. 이에 대한 보다 상세한 유도 과정을 위해서는 [18]을 참조하기 바란다.

$$r \leq \log_p \left\{ \frac{\rho_{new} - (1-p)\sigma}{\sigma \cdot p} \right\} \tag{3}$$

4. 성능 분석

4.1 시뮬레이션 환경

제안된 재전송 지속성 관리 방안의 성능을 평가하기 위해서 OPNET을 이용한 시뮬레이션을 수행하였다. 시뮬레이션을 하기 위해서 설정된 변수들을 <표 1>에 정리하였다.

TCP 송신원은 이동성이 없는 고정 단말이고 TCP 수신원은 이동 단말인데 이들은 기지국을 통해서 연결된다. TCP 송신원과 TCP 수신원 사이에는 두 개의 연결이 설정되는데 하나는 FTP 서비스를 위한 연결이고 다른 하나는 백그라운드 트래픽을 위한 연결이다. TCP 송신원은 FTP 트래픽과 백그라운드 트래픽에 대한 TCP 패킷들을 발생시킨다. 이 때 백그라운드 트래픽의 발생을 위해서는 [16]에 소개된 HTTP 트래픽 모델을 참조하였고 제안된 방안의 성능 분석을 위해서 FTP 연결을 통해서 전송되는 TCP 패킷들을 관찰하였다. 그리고 패킷 손실은 무선 구간에서 한정되어 발생하고 손실되는 형태는 랜덤하게 서로 독립적이라고 가정하였다.

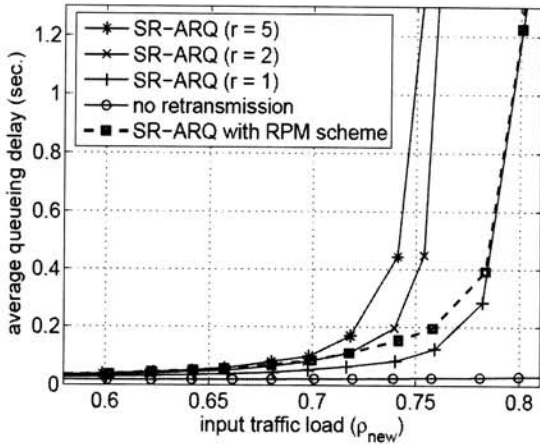
실제 상황을 반영하기 위해서 하나의 SR-ARQ 프로토콜이 상위에서 전달되는 FTP 트래픽의 TCP 패킷과 백그라운드 HTTP 트래픽의 TCP 패킷을 처리하도록 설정하였다. 식 (3)에 의한 재전송 반복 횟수 r 은 5개의 SR-ARQ 패킷을 전송할 때마다 측정되어 갱신된다. 시뮬레이션이 진행되는 동안 TCP 송신원은 4Mbytes에 해당하는 TCP 패킷들을 전송하고 동일한 설정에 대해서 이와 같은 시뮬레이션을 10회 실시하여 평균값을 산출하였다.

<표 1> 시뮬레이션 변수와 설정된 값

변수	설정된 값
TCP 종류	TCP with SACK option
TCP maximum segment size	536bytes
TCP advertisement window size	10kbytes
SR-ARQ packet size	300bytes
SR-ARQ transmitter's queue size	50packets
σ	0.95
유선 구간 전송 속도	10Mbps
무선 구간 전송 속도	240kbps
TCP 송신원과 BS 사이의 유선 구간 RTT	100msec
BS와 TCP 수신원 사이의 무선 구간 RTT	200msec

4.2 ARQ 계층에서의 시뮬레이션 결과

먼저 제안된 재전송 지속성 관리 방안이 정상적으로 동작하는지를 판단하기 위해서 ARQ 계층에서 몇 가지 시뮬레이션을 수행하였고 그 결과를 분석하였다. 제안된 방안이 있어서 재전송 반복 횟수(r)는 1과 5사이의 값을 가지도록 설정되었다. 다음 (그림 5)는 신규 트래픽 부하(ρ_{new})에 대한 SR-ARQ의 송신원 큐에서의 큐잉 지연을 측정된 결과를 보여준다. 패킷손실률은 0.25로 고정된 상태에서 제안된 방



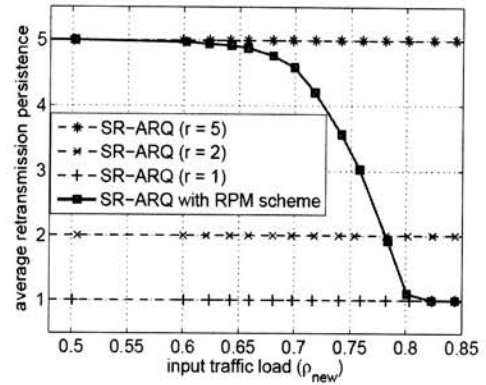
(그림 5) 신규 트래픽 부하(ρ_{new})에 대한 SR-ARQ의 큐잉 지연

안의 동작을 손실된 패킷에 대해서 재전송을 아예 하지 않는 경우와 r 을 1, 2, 그리고 5로 고정한 네 가지 경우와 비교하였다.

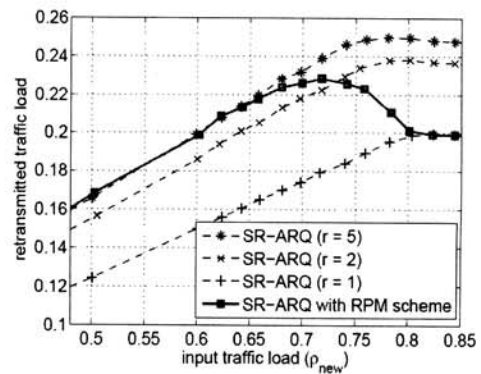
우선 재전송을 아예 하지 않는 경우에 있어서는 ρ_{rtx} 가 항상 0이라는 의미이므로 ρ_{new} 만이 SR-ARQ 송신원에 가해지는 부하로서 작용한다. 그림에서 볼 수 있는 바와 같이 본 실험은 항상 ρ_{new} 가 1보다 작은 값을 갖기 때문에 이 경우의 큐잉 지연은 항상 작은 값으로 일정하게 유지되는 것을 볼 수 있다.

반면 나머지 재전송을 수행하는 경우에 있어서는 ρ_{new} 가 어떤 값에 도달하게 되면 공통적으로 큐잉 지연이 급격하게 증가하는 것을 볼 수 있다. 이는 앞서 설명한 바와 같이 ρ_{new} 와 ρ_{rtx} 의 합이 0.95로 설정된 σ 를 넘게 됨을 의미하는 것이다. 재전송 반복 횟수 r 이 크면 클수록 ρ_{rtx} 의 값이 증가하여 상대적으로 낮은 ρ_{new} 값에서 ρ_{new} 와 ρ_{rtx} 의 합이 0.95를 넘게 되기 때문에 r 이 클수록 큐잉 지연의 값이 급격하게 증가하기 시작하는 ρ_{new} 의 값이 작은 것은 당연한 결과이다. 제안된 방안의 경우 큐잉 지연의 값이 급격하게 증가하는 ρ_{new} 의 값이 최대한 늦추어 지는 것을 볼 수 있는데 이는 ρ_{new} 이 증가함에 따라서 r 값이 계속 감소함을 의미한다. 본 시뮬레이션에서 제안된 방안은 r 이 1과 5사이의 값을 가지도록 하였으므로 최종적으로 r 이 1인 경우의 큐잉 지연과 완벽하게 일치하는 값을 가지게 됨을 확인할 수 있다. 보다 명확한 설명을 위해서 다음 (그림 6)과 (그림 7)에는 ρ_{new} 에 대한 r 과 ρ_{rtx} 의 측정 결과를 나타내었다.

앞에서 설명한 바와 같이 (그림 6)에서 ρ_{new} 가 증가함에 따라서 r 의 값이 5에서 1로 감소하고 그에 따라서 (그림 7)에서 ρ_{rtx} 역시 감소하는 것을 확인할 수 있다. 결과적으로 제안된 재전송 지속성 관리 방안은 ρ_{new} 에 따라서 재전송 허용 횟수의 최댓값을 1과 5사이에서 탄력적으로 조절할 수 있음을 알 수 있다.



(그림 6) 신규 트래픽 부하(ρ_{new})에 대한 r 의 변화

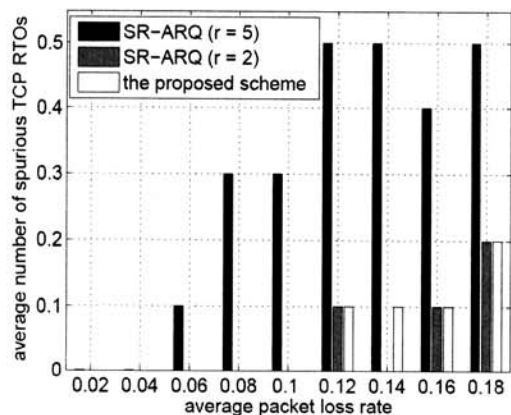


(그림 7) 신규 트래픽 부하(ρ_{new})에 대한 ρ_{rtx} 의 변화

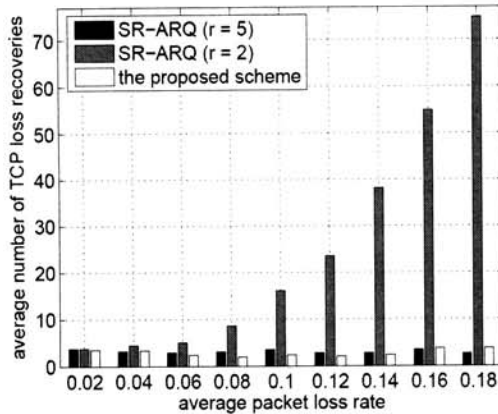
4.3 TCP 계층에서의 시뮬레이션 결과

다음으로 제안된 방안이 실제 TCP의 성능에 어느 정도 기여할 수 있는지를 평가하기 위한 시뮬레이션을 수행하였다. 우선은 과연 제안된 방안이 스푸리어스 발생을 방지하는데 기여할 수 있는지의 여부를 알아보기 위해서 패킷손실률을 증가시키면서 스푸리어스 타임아웃의 발생 횟수를 측정하여 (그림 8)에 나타내었다. 앞서와 마찬가지로 제안된 방안의 결과를 r 을 2와 5로 고정시킨 결과와 비교하였다.

그림에서 볼 수 있듯 r 의 값을 가장 큰 5로 설정한 경우 가장 많은 횟수의 스푸리어스 타임아웃이 발생하고 r 의 값



(그림 8) 패킷손실률(p)에 대한 스푸리어스 타임아웃 발생 횟수



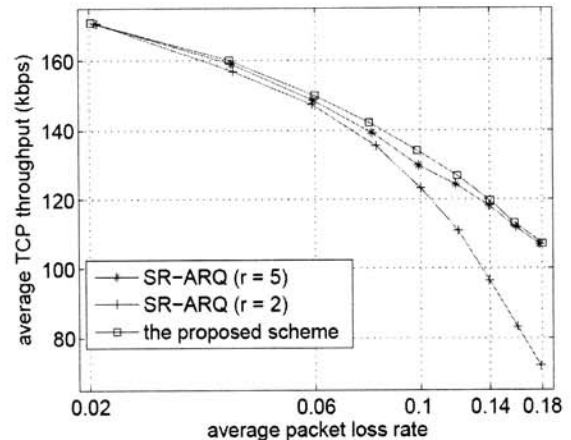
(그림 9) 패킷손실률(p)에 대한 TCP 손실 복구 수행 횟수

을 2로 설정한 경우와 제안된 방안의 경우는 비슷한 횟수의 스푸리어스 타임아웃이 발생함을 확인할 수 있다. 보다 확실한 확인을 위해서 (그림 9)에는 TCP 프로토콜의 손실 복구 과정의 수행 횟수 측면에서 제안된 방안을 r 을 2와 5로 고정시킨 경우와 비교하였다. 여기서 TCP 손실 복구 과정이란 TCP가 감지한 패킷 손실을 fast retransmit에 의한 복구하는 경우와 재전송 타임아웃에 의한 복구하는 경우를 모두 포함한다.

우선 r 의 값이 2인 경우 TCP 손실 복구 수행 횟수가 가장 높은 것을 볼 수 있다. ARQ에 의한 재전송 반복 횟수가 낮다는 것은 손실된 ARQ 패킷이 TCP 패킷 손실로 이어질 확률이 높다는 것을 의미한다. 따라서 TCP 패킷 손실의 수가 증가하기 때문에 이를 복구하기 위한 TCP 손실 복구 과정의 수행이 잦을 수밖에 없다. 반면 r 이 5인 경우와 제안된 경우에는 수행되는 TCP 손실 복구 횟수가 상당히 작는데 이는 ARQ 패킷 손실이 충분히 재전송에 의해 복구되어 TCP 패킷 손실로 이어지지 않았음을 의미하는 것이다.

(그림 8)과 (그림 9)에 나타난 결과를 근거로 제안된 방안은 스푸리어스 타임아웃 측면에서는 작은 값의 r 의 장점을, 그리고 TCP 손실 복구 수행 횟수 차원에서는 큰 값의 r 의 장점을 취한다고 평가하는 것이 가능하다. 즉, 스푸리어스 타임아웃을 방지하는 동시에 TCP 손실 복구 수행 횟수도 낮게 유지하기 때문에 TCP 처리율 역시 당연히 향상시킬 것으로 예상할 수 있다. 이를 확인하기 위해서 다음 (그림 10)에 세 가지 경우의 TCP 처리율을 비교하여 나타내었다.

예상했던 바와 같이 제안된 방안이 적용된 TCP 처리율이 가장 높은 것을 확인할 수 있다. (그림 9)에서 설명했듯이 r 이 2인 경우에는 복구해야 하는 TCP 패킷 손실이 많아지기 때문에 패킷손실률이 증가할수록 TCP의 처리율이 나머지 두 경우에 비해서 많이 떨어지는 것을 볼 수 있다. 그러나 r 이 5인 경우와 제안된 방안의 경우의 TCP 처리율은 그다지 큰 차이가 없음을 볼 수 있는데 그에 대한 이유는 스푸리어스 타임아웃의 발생빈도로서 설명될 수 있다.



(그림 10) 패킷손실률(p)에 대한 TCP 처리율

(그림 9)에서와 같이 TCP 손실 복구 수행 횟수에서 r 이 5인 경우와 제안된 방안의 경우의 차이는 매우 미미하다. 이는 두 경우의 성능 차이는 단지 스푸리어스 타임아웃에만 기인한다는 것을 의미한다. 그런데 4Mbyte를 전송하도록 설정한 시뮬레이션이 진행되는 동안 스푸리어스 타임아웃의 발생 빈도는 그리 높지 않았다. 즉, 스푸리어스 타임아웃은 일단 발생하면 순간적으로 TCP 처리율에 미치는 영향이 매우 크지만 그다지 자주 발생하지 않았기 때문에 그 영향이 전체 시간에 대해 평균화되어 이와 같은 미미한 처리율의 차이를 보인 것으로 분석할 수 있다.

5. 결 론

본 논문에서는 SR-ARQ와 TCP와의 상호 작용을 고려한 재전송 지속성 관리 방안을 제안하였다. 특히 SR-ARQ의 재전송 반복 횟수를 결정함에 있어서 신규 트래픽 부하를 고려함으로써 스푸리어스 타임아웃을 방지할 수 있음을 보였다. 평균화된 TCP 처리율에 일시적인 스푸리어스 타임아웃이 미치는 영향이 수치상으로는 미미하게 보일 수 있으나 연결의 특성 및 지속시간에 따라서 스푸리어스 타임아웃은 TCP의 성능에 상당히 나쁜 영향을 줄 수 있는 것이 사실이다. IETF에서 표준화한 두 개의 방안과는 달리 본 논문에서 제안된 방안은 스푸리어스 타임아웃을 방지하기 위한 근본적인 방안이 될 수 있으며 상당히 간단히 구현될 수 있기에 추후 이동 통신 시스템에서 TCP의 성능 향상에 크게 기여할 수 있을 것이다.

참 고 문 헌

- [1] TIA/EIA/IS-707-A.2 (PN-4145.2), "Data service options for wideband spread spectrum systems: radio link protocol," July, 1998.
- [2] 3GPP2, "cdma2000 high rate packet data air interface specification," C.S0024, ver. 4.0, Oct., 2002.

[3] G. Fairhurst and L. Wood, "Advice to link designers on link automatic repeat reQuest (ARQ)," RFC 3366, Aug., 2000.

[4] P. Karn et al., "Advice for Internet Subnetwork Designers," IETF RFC 3819, June, 2004.

[5] M. Allman, V. Paxson, and, W. Stevens, "TCP Congestion Control," IETF RFC 2581, Apr., 1999.

[6] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," IEEE/ACM Transaction on Networking, Vol.5, No.6, 1997.

[7] V. Paxson and M. Allman, "Computing TCP's Retransmission Timer," IETF RFC 2988, Nov., 2000

[8] Andrei Gurtov and Reiner Ludwig, "Responding to Spurious Timeouts in TCP," in Proc. IEEE Infocom, 2003.

[9] P. Sarolahti and M. Kojo, "Forward RTO-Recovery (F-RTO) : An Algorithm for Detecting Spurious Retransmission Timeouts with TCP and the Stream Control Transmission Protocol (SCTP)," RFC 4138, Aug., 2005.

[10] Carla F. Chiasserini and Michela Meo, "Modeling Interactions between Link Layer and Transport Layer in Wireless Networks," The 12th IEEE International Symposium on PIMRC 2001, San Diego, USA, Sep., 2001.

[11] Francesco Vaciraca, Andrea De Vendictis, and Andrea Baiocchi, "Investigating Interactions between ARQ Mechanisms and TCP over Wireless Links," in Proc. of European Wireless 2004, Barcellona, Spain, Feb., 2004.

[12] A. Mehta, D. Kagaris, and R. Viswanathan, "Throughput Performance of an Adaptive ARQ Scheme in Rayleigh Fading Channels," IEEE Trans. Wireless Comm., Vol.5, No.1, Jan., 2006.

[13] Carla-Fabiana Chiasserini and Michela Meo, "A Recon-gurable Protocol Setting to Improve TCP Over Wireless," IEEE Trans. Vehic. Tech., Vol.51, No.6, Nov., 2002.

[14] Juan J. Alcaraz, Fernando Cerdan, and Joan Garcia-Haro, "Optimizing TCP and RLC Interaction in the UMTS Radio Access Network," IEEE Network, Vol.20, issue 2, Mar., 2006.

[15] Jeong Geun Kim and Marwan M. Krunz, "Delay Analysis

of Selective Repeat ARQ for a Markovian Source Over a Wireless Channel," IEEE Trans. Vehic. Tech., Vol.49, No.5, Sept., 2000.

[16] 3GPP2 C.R1002-0 1.0, "CDMA2000 evaluation methodology," Jan., 2005.

[17] V. Subramanian, K. K. Ramakrishnan and S. Kalyanaraman, "Experimental Study of Link and Transport Protocols in Interference-Prone Wireless LAN Environments," IEEE COMSNETS, 2009.

[18] J. Han et al., "Queuing Analysis Model for the SR-ARQ Protocol with a Finite Retransmission Persistence," 정보처리학회논문지, 제15-C권, 제6호, 2008년 12월.



김 범 준

e-mail : bkim@kmu.ac.kr

1996년 연세대학교 전자공학과(학사)

1998년 연세대학교 전자공학과(공학석사)

2003년 연세대학교 전자공학과(공학박사)

2004년~2006년 LG전자 이동통신기술연구소 선임연구원

2006년~현 재 계명대학교 전자공학과 교수

관심분야 : TCP 혼잡제어, IEEE 802.16, IPTV 및 VoIP 서비스 품질 관리 등



한 제 찬

e-mail : jechan.han@lge.com

2002년 연세대학교 기계전자공학부(학사)

2004년 연세대학교 전기전자공학과(공학석사)

2010년 연세대학교 전기전자공학과(공학박사)

2010년~현 재 LG전자 MC연구소 선임연구원

관심분야 : TCP 혼잡제어 및 무선 TCP, 링크 레벨 오류 제어 및 성능분석