

논문 2014-09-43

인지 무선 환경에서 TCP 성능 향상

(Improving TCP Performance Over Cognitive Radio Networks)

변상선*
(Sang-Seon Byun)

Abstract : In cognitive radio networks (CRNs), SU (secondary user)'s transmissions are frequently disrupted by PU (primary user)'s transmission. Therefore SU experiences consecutive retransmission timeout and its exponential backoff, and subsequently, the TCP of SU does not proceed with the transmission even after the disruption is over or the SU succeeds to hold an idle channel. In order to solve this problem, we propose a cross-layer approach called TCP-Freeze-CR. Moreover we consider a practical scenario where either secondary transmitter (ST) or secondary receiver (SR) detects PU's transmission, which results in the need of spectrum synchronization mechanism. All of our proposals are implemented and verified with a real CRN testbed consisting of 6 software radios called USRP. The experimental results illustrate that standard TCP suffers from significant performance degradation and show that TCP-Freeze-CR greatly mitigates the degradation.

Keywords : TCP, TCP-Freeze-CR, Cognitive radio networks, Universal software radio peripheral (USRP), GNU radio

1. 서론

인지무선 (cognitive radio)기술은 무선 통신의 가장 중요한 문제점으로 대두된 주파수 부족문제를 해결할 수 있는 가장 확실한 방법이다 [1].

표준 TCP (transmission control protocol), 즉, TCP-newReno를 인지무선환경에서 사용했을 때, 심각한 성능저하가 발생되는데, 그 이유는 다음과 같이 기술된다. 인지무선환경에서 2차사용자 (secondary user)가 현재 사용 중인 주파수를 1차 사용자 (primary user)가 사용하려고하면 그 2차 사용자는 즉시, 현재 사용 중인 주파수를 1차 사용자에게 양보하고 다른 사용가능한 주파수를 탐색해야한다. 물론, 탐색을 하는 동안 통신은 중단된다. 하지만, TCP는 이러한 통신 중단 사실을 감지할 수 없기 때문에, 만약, 탐색시간이 길어지게 되면 연속된

패킷 손실이 발생하게 되고, TCP의 혼잡제어 알고리즘은 혼잡원도우를 1로 줄이고 재전송타이머를 지수적으로 증가 (exponentially back-off) 시킨다. 결과적으로 TCP 성능은 심각하게 저하되고, 더 나아가 2차사용자가 사용가능한 새로운 주파수를 찾아 통신을 재개하더라도 TCP는 길어진 재전송 타이머로 인해 패킷을 송신하지 못하게 되는 상황이 발생한다.

우리는 이 논문에서 앞서 제시된 문제점을 크로스레이어 시그널링과 최소한의 TCP 수정을 통해 간단하게 해결하고자 한다. 물리계층 (또는 링크계층)에서 1차전송 (primary transmission)이 감지되면 이를 TCP에게 시그널링 (freeze 시그널을 사용하여 불필요한 혼잡제어와 재전송타이머아웃을 막으면서 전송을 중단시킨다. 그리고, 새로운 주파수로 동기화가 완료되거나 1차전송의 종료를 감지하게 되면, 재차 시그널링 (unfreeze 시그널을 사용하여 중단되기 이전의 상태로 돌아가 전송을 재개한다.

우리는 2차송신자 (secondary transmitter)와 2차수신자 (secondary receiver) 둘 중 하나만 1차전송을 감지하는 경우도 고려한다 (그림 1 참조). 이 상황은, 2차송신자가 1차송신자의 송신범위를

*Corresponding Author(ssbyun@cup.ac.kr)

Received: 28 July 2014, Revised: 27 Aug. 2014, 13 Oct. 2014 Accepted: 29 Oct. 2014.

S.-S. Byun: Catholic University of Pusan

※이 논문은 2014년도 부산가톨릭대학교 교내연구비에 의하여 연구되었음.

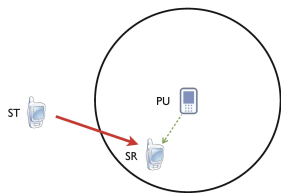


그림 1. 1차전송 (PU)이 오로지 2차수신자 (SR)에 의해서만 감지되는 경우, 2차수신자(ST)는 1차전송을 감지 못함.

Fig. 1 Illustration of the case where the primary transmission is detected by only the SR. The ST cannot detect the primary transmission.

벗어나서 위치하는 경우 발생할 수 있다. 또한, 2차송신자의 전송게인이 상대적으로 1차송신자의 그것보다 큰 경우에도 발생할 수 있다.

협조 감지 (cooperative detection) [2]를 통해 주위에 있는 다른 2차사용자들의 도움을 받아 1차전송을 감지할 수도 있다. 하지만, 이 방법은 모든 2차사용자들과 현재 전송중인 1차사용자들의 전송 및 수신 게인에 대한 정보를 모두 알고 있어야 가능하므로 실제로 구현되기에는 어려움이 있다.

1차전송의 감지는 2차사용자의 물리 또는 링크 계층에서 이루어진다. 2차송신자에 의해서 감지되는 경우 2차송신자는 2차수신자에게 주파수 탐지 및 동기화를 요구하고 자신의 TCP를 크로스레이어 시그널링을 통해 freeze 한다. 2차수신자에 의해서 감지가 되는 경우, 2차수신자는 즉시 이를 2차송신자에게 알려서 2차송신자의 TCP가 freeze 되게 하고 주파수 탐지 및 동기화를 진행한다. 주파수 동기화가 완료되면 2차송신자의 물리계층은 unfreeze 시그널을 자신의 TCP에 전달하여 송신이 재개되게 한다. 주파수 동기화 과정은 2차송수신자의 물리계층에 구현이 되며 이는 송신자와 수신자가 같은 주파수로 동기화 되어 통신을 재개하도록 하기 위해 인지무선에서는 반드시 필요한 기능이다. 우리는 유한상태기계를 통해 이 주파수 동기화 과정을 정의 및 구현한다.

우리가 제안하는 방법은 대표적인 소프트웨어 라디오 플랫폼인 USRP (Universal Software Radio Peripheral)에 구현된다. 그리고, 물리계층으로는 IEEE 802.15.4를 사용한다. IEEE 802.15.4와 IEEE 802.11 그리고 IEEE 802.15.1 모두 2.4

GHz 대역을 사용하므로 인지무선을 통한 위 세 표준의 공존문제가 중요하게 다루어지고 있고 [3], IEEE 802.15.4와 TCP를 통한 소형장치들간의 신뢰적인 통신 프로토콜 구현도 연구가 활발하게 진행되고 있다 [4].

II. 관련연구

인지무선네트워크가 아닌, 일반 무선네트워크에서 TCP 혼잡제어의 성능 개선을 위한 연구는 매우 방대하게 오랫동안 지속되고 있다. 주로, 혼잡에 의한 패킷손실과 무선 통신의 특성 (간섭, 페이딩, 섀도잉 등) 때문에 발생하는 패킷손실을 구별하는 방법들을 제안해왔다 [5]. 일반 무선네트워크에서 발생하는 무선통신 특성에 의한 통신단절은 대부분 일시적인 반면, 인지무선네트워크의 1차전송에 의한 통신단절은 수초에서 수분까지 진행될 수 있어, 더 막대한 TCP 성능저하를 초래 할 수 있다. 따라서, 인지무선환경에서는 반드시 이 점을 고려하여 TCP 를 개선하여야 한다.

Issariyakul [6]에 의해 2차사용자가 표준 TCP 를 사용할 때, 발생하는 성능저하 문제가 처음 제기되었고 이를 해결하기 위해 몇몇 연구가 진행되었다. 하지만, 대부분의 경우는 구현 중심이 아닌 이론 중심이고, 실제 환경에는 맞지 않는 편리한 이론 전개를 위한 가정들을 사용하였다. 예를 들어, 채널 상태의 완전한 파악가능 [7], 에러가 전혀 없는 제어 채널 [8], 1차사용자 전송의 통계적인 예측가능 [8, 9], 단일 RTT (round-trip-time) 동안 변하지 않는 채널 게인 [7] 등이다.

[10]에서 저자들은 무선 멀티홉 네트워크 (인지 무선네트워크는 아님)에서 TCP 성능을 향상시키는 크로스레이어 방법을 제안하고 리눅스 커널과 MadWiFi 드라이버를 사용하여 구현하였다.

우리가 제안하는 방법은 하위계층에서 네트워크의 상태에 따라 전송계층의 상태를 freeze 하고 unfreeze 한다. 따라서, 무선네트워크에서 TCP 성능개선 방법을 제안한 Freeze-TCP [11]와 유사하게 동작한다. 이 Freeze-TCP 역시 크로스레이어 기법을 사용한다. 하지만, Freeze-TCP를 아무런 수정없이 인지무선환경에 적용하고자 하면 다음과 같은 문제점들이 발생된다.

- (1) Freeze-TCP는 무선채널의 저하가 예상되면 수신측 버퍼공간의 고갈을 나타내는 zero window

advertisement를 통해 송신측 TCP를 freeze 하게 되는데, 이는 실제 수신측 버퍼가 고갈되어 발생한 것인지 아니면, 1차전송에 의해서 발생한 것인지 구별할 방법이 없다. 인지무선환경에서 1차전송이 아닌 수신측 버퍼고갈로 인해 zero window advertisement가 발생하였을 경우에는 주파수를 재동기화할 필요가 없다.

- (2) Freeze-TCP에서 지나치게 잦은 freeze가 발생되게 되면 재전송타이머 값이 계속고정되어 unfreeze를 통해 전송이 원활히 재개 되지 않을 수 있다.

위 언급된 2가지 문제를 우리는 다음과 같이 해결한다.

- (1) 1차전송이 발생했을 때, zero window advertisement가 아닌 별도의 크로스레이어 시그널을 통해 TCP를 freeze 한다.
- (2) Freeze 되었다가 unfreeze 되어 통신이 재개될 때, 재전송타이머를 최소값으로 리셋하여, 전송을 신속하게 재개한다.

III. 주파수 동기화 모델

주파수 동기화는 1차전송에 의해서 현재 주파수를 사용할 수 없을 때, 새로운 주파수로 2차송수신자가 동기화되는 과정이다. 이 경우, 주파수 탐색은 2차송신자와 2차수신자 각각 수행하나 최종적으로는 동일한 주파수로 동기화가 되어야 한다. 1차전송이 감지되면 2차송신자 또는 2차수신자에 의해 이 주파수 동기화 과정이 시작되고, 이와 동시에 2차송신자의 TCP가 freeze 된다. 새로운 채널로 동기화가 완료되면 TCP는 unfreeze 된다.

주파수 동기화 과정은 유한상태기계로 모델링 및 구현 되고 (그림 2 참조), 이 유한상태기계는 2차송신자에서만 구현된다. 각 상태의 정의는 다음과 같다.

- TRANSMIT: 2차사용자들이 현재 통신중인 상태
- SENSING: 2차사용자들이 다른 사용가능한 주파수를 감지 (sensing)하고 있는 상태
- COMPARE: 2차수신자는 주파수 감지가 끝나면 자신이 사용가능한 주파수 대역의 목록을 2차송신자에게 알려준다. 그러면, 2차송신자는 2차수

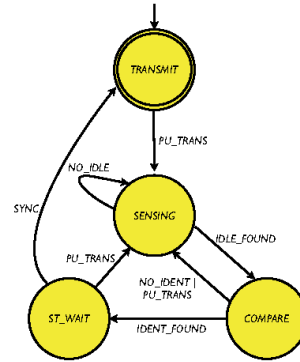


그림 2. 주파수동기화를 위한 유한상태기계모델
Fig. 2 Finite state machine for spectrum synchronization

신자가 알려준 주파수 목록 가운데, 자신도 사용가능한 주파수가 있는지 확인하는 상태를 나타낸다.

- ST_WAIT: 2차수신자가 새로운 주파수로 핸드오버하기를 2차송신자가 기다리는 상태, 2차수신자는 새로운 주파수로 핸드오버를 완료하면 확인 (acknowledgement)을 그 새로운 주파수를 통해 2차송신자에게 전달한다.

그리고, 각 상태전이는 다음과 같이 정의된다.

- PU_TRANS: 2차송신자 또는 2차수신자가 1차송신을 감지. 2차송신자가 감지한 경우에는 바로 SENSING 상태로 전이하지만, 2차수신자가 먼저 감지한 경우에는 이를 2차송신자에게 알려서, 2차송신자가 SENSING 상태로 전이하도록 한다.
- NO_IDLE: 2차송신자가 유휴스펙트럼 (idle spectrum)의 발견 실패
- IDLE_FOUND: 2차송신자가 유휴스펙트럼 발견 성공
- NO_IDENT: 2차송신자는 유휴스펙트럼을 발견했으나, 발견된 대역을 2차수신자가 접근할 수 없는 경우에 이 상태전이가 발생. 2차송신자 또는 2차수신자 둘 중에 하나만 1차전송에 의해서 영향을 받는 경우, 또는 2차송신자와 2차수신자가 감지한 1차전송의 발생 주파수 대역이 서로 다른 경우, 이 두 경우에 있어서, 한쪽에서는 사용가능하다고 판단된 유휴스펙트럼이 다른 한쪽에서는 사용 불가능한 것으로 판단될 수 있다.

- IDENT_FOUND: 2차송신자와 2차수신자가 모두 사용할 수 있는 사용가능 주파수 대역의 발견 성공. 성공이 되면, 2차송신자는 2차수신자에게 발견된 유희스펙트럼으로 동기화 할 것을 요청한다.
- SYNC: 새로 발견된 유희스펙트럼으로 동기화 완료. 새로이 발견된 유희스펙트럼으로 2차수신자가 동기화를 완료했음을 2차송신자가 확인했을 때 발생하는 상태전이다.

IV. 인지무선환경을 위한 TCP 알고리즘

우리가 제안하는 방법은 Freeze-TCP를 응용한 것이므로 TCP-Freeze-CR이라고 명명한다. TCP-Freeze-CR과 크로스레이어 시그널링은 2차송신자에 구현이 되며, 2차수신자에는 표준 TCP가 그대로 사용된다. 물론, 2차송수신자가 따로 지정되지 않고 양방향으로 통신이 발생하는 경우에는 모든 2차사용자가 수정된 TCP와 크로스레이어 시그널링 기능을 가지고 있어야 한다. 이 논문에서는 2차송신자와 2차수신자가 별도로 구분되는 경우를 가정한다. 무선센서네트워크와 같이 통신이 한방향으로만 주로 발생하는 경우가 좋은 예이다.

1차전송은 2차송신자와 2차수신자 모두가 개별적으로 감지하며, 2차수신자가 감지한 경우에는 이를 2차송신자에게 알려서 TCP-Freeze-CR이 동작하도록 해야한다. 1차전송의 감지는 하위계층(물리 또는 링크계층)에서 수행되기 때문에, 2차수신자가 먼저 감지한 경우에는 이 사실을 2차송신자의 하위계층으로 전달해야 한다. 이 전달은 별도의 TCP 연결을 통해서 또는 하위계층간 직접 통신을 통해서 가능하다.

1차전송이 감지되면 하위계층은 freeze 신호를 TCP에 보낸다. 그리고, 새로운 주파수로 2차송수신자가 동기화를 완료하게 되면 unfreeze 신호를 보낸다. Freeze 신호는 주파수 동기화를 위한 유한상태기계의 (TRANSMIT, PU_TRANS) → SENSING 상태전이와 동시에 전달되고, unfreeze 신호는 유한상태기계의 (ST_WAIT, SYNC) → TRANSMIT 상태전이시 전달된다.

TCP-Freeze-CR은 기존의 표준 TCP (즉, TCP-newReno)에 freeze/unfreeze 크로스레이어 신호를 처리하기 위한 모듈이 알고리즘 1과 같이 추가된다.

알고리즘 1의 동작을 간략히 요약하면, freeze

```

procedure OnCrossLayerSignal (signal =
{freeze, unfreeze})
  if signal == freeze then
    Cancel impending RT;
    old_cwnd = cwnd;
    old_rtt = rtt;
    cwnd = 0;
  else if signal == unfreeze then
    rtt = old_rtt;
    if old_cwnd > 0 then
      cwnd = old_cwnd;
    else
      phase = SLOW_START;
      cwnd = 1;
    end if
    RTO_value = min_RTO_value;
    Restart RT;
  end if
end procedure

```

알고리즘 1

Algorithm 1

신호를 받으면 (즉, if signal == freeze then) 제일 먼저 재전송 타이머 (RT) 를 취소한다. 그리고, 현재의 혼잡 윈도우 크기와 RTT를 저장하고 (즉, old_cwnd = cwnd; old_rtt=rtt), 혼잡 윈도우 크기를 0으로 만듦으로써 (즉, cwnd=0;) 아무런 전송이 일어나지 않도록 한다. 즉, 현재 TCP의 상태를 열리는 (freeze) 것이다. Unfreeze 신호를 받게 되면 freeze 될 때 저장된 RTT와 cwnd 값으로 환원하고 (즉, rtt=old_rtt; cwnd=old_cwnd;), 재전송타이머를 재개 한다.

추가적으로 우리는 다음의 두 가지 사항을 고려한다.

1. 만약 freeze 단계에서 저장된 cwnd 값 (freeze 되기 이전의 cwnd 값, 즉, old_cwnd)이 0인 경우, unfreeze 되어도 전송이 개시되지 않기 때문에 cwnd 값을 1로 해서 바로 전송이 개시되게 한다. 여기서, 빠른 패킷 전송을 위해 전송 단계를 slow start로 한다 (즉, phase = SLOW_START; cwnd = 1;)
2. Freeze 되기 전에 재전송 타임아웃을 기다리고 있는 상태이고, freeze 당시의 재전송 타이머가 길게 세팅되어 있다면 unfreeze가 되어도 전송

이 바로 일어나지 않게 된다. 따라서, unfreeze 될 때는 항상 재전송 타임아웃 값을 최소값으로 재정의하고 (즉, $RTO_value = \min_RTO_value$; Restart RT;) 타이머를 개시한다.

위 두 추가사항으로 인한 성능변화는 실험을 통해 평가된다.

V. 구현

1. 시스템 플랫폼

우리는 대표적인 소프트웨어 라디오 개발 플랫폼인 USRP E100을 하드웨어로 사용한다. USRP E100은 ARM Cortex A8 코어를 가지고, 임베디드 리눅스를 운영체제로 사용한다. 그리고, IEEE 802.15.4는 USRP에서 동작되도록 구현된 프로그램 (IEEE 802.15.4 PHY 모듈) [12]을 설치하여 타겟 하위계층으로 사용한다.

2. 주파수 동기화과정의 구현

주파수 동기화 과정은 IEEE 802.15.4 PHY 모듈내에 추가된다. 주파수 감지 모듈은 물리계층 표준과 독립적으로 동작해야 한다. 하지만, 그 위에서 동작하는 TCP의 성능을 향상하는 것이 우리의 주목적이므로, 우리는 1차사용자 역시 IEEE 802.15.4 PHY를 사용하여 전송하게 한다. 2차송수신자는 1차사용자의 MAC 주소를 미리 알고 있도록 해서 1차사용자로부터 오는 패킷을 해독 함으로써 1차전송을 감지 하도록 한다. 이는 인지무선환경에서 1차전송의 개시와 종료만이 TCP 성능에 영향을 미치도록 하기 위함이다. 즉, 1차전송 감지 능력에 의해서 겪게 되는 TCP 성능 저하 문제는 이 논문의 범위를 벗어난다.

주파수 동기화를 위해서는 2차송수신자 간에 몇 번의 메시지 교환이 이루어진다. 즉, 가용채널 목록과 새로이 선택된 채널, 그리고, 수신확인이 2차송수신자 사이에 전달된다. 이러한 메시지 교환은 2차송수신자 가운데 1차전송을 먼저 감지한 쪽에서 다른 쪽의 동기화 과정을 유발하는 방식으로 진행된다. 예를 들어, 2차송수신자가 1차전송을 감지하면 2차수신자에게 주파수 감지를 요청하고 그 결과로 가용채널 (유휴스펙트럼)의 리스트를 반환받는다. 즉, 한쪽에서 요청을 하면 다른 한쪽에서 요청 결과를 반환하는 식으로 진행되기 때문에, 원격프로시저 호출 (RPC: Remote Procedure Call)을 사용하여 구현하면 구현의 복잡도를 낮출 수 있다. 또한, 이

메시지 교환역시 무선채널을 통해서 이루어지므로 신뢰성의 확보가 중요하다. 우리는 Python RPC 확장 라이브러리인 Pyro를 이용, 2차송수신자간 주파수동기화 과정을 구현한다.

우리는 TCP-Freeze-CR의 성능에 영향을 미치는 부가적 요소들을 최소화 하기 위해 TCP 자체 내에서 사용자 데이터를 임의로 생성하게 하고, IP 계층은 두지 않는다. 따라서, Pyro RPC는 TCP-Freeze-CR을 사용하도록 구성될 수 없고, 대신 TUN 인터페이스를 통해 리눅스 운영체제 내의 TCP/IP 스택을 사용하도록 한다.

3 TCP-Freeze-CR

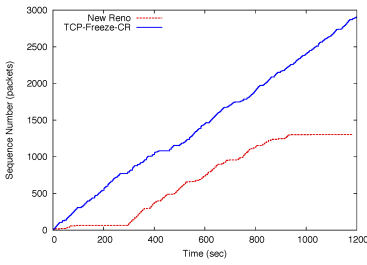
TCP-Freeze-CR은 표준 TCP (즉, TCP-NewReno)에 기반한다. 따라서, 표준 TCP에 기술된 모든 혼잡제어 관련 기법들, 즉, fast retransmit과 fast recovery 기능, RTO에 의한 재전송, Jacobson 알고리즘, cumulative acknowledgement 는 모두 구현한다.

TCP-Freeze-CR은 C언어로 구현되고, 개별 프로세스로 동작한다. USRP E100에서는 RF 이후의 하위 계층 프로토콜들 역시 사용자 프로세스로 동작하기 때문에, 우리는 UNIX domain socket (UDS) 을 통해 TCP-Freeze-CR 프로세스가 Python으로 구현된 802.15.4 PHY 프로세스와 통신하게 한다. 사용자 데이터 송수신 뿐만 아니라 크로스레이어 시그널링도 UDS를 통해 이루어지도록 한다.

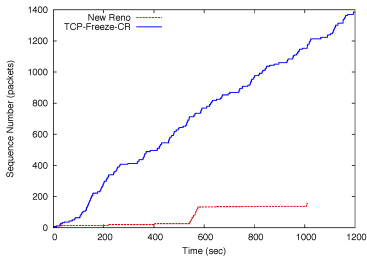
VI. 성능 평가

1. 실험환경 구축

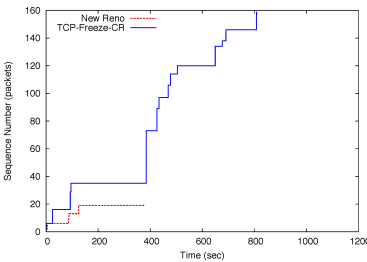
우리가 제안하는 TCP 성능 평가를 위해서 6개의 USRP 장비를 사용하여 인지네트워크를 구성한다. 두 대의 USRP는 2차송수신자를 위해, 나머지 4대의 USRP는 1차사용자로 동작되게 한다. 2차송수신자는 두 개의 채널을 사용하여 통신하고, 하나의 채널에서 1차전송이 감지되면 다른 채널로 전환한다. 각 채널은 1차사용자로 사용되는 USRP 각 2대가 간섭하도록 한다. 모든 1차 사용자는 T_{on} 초 동안 전송을 하고, T_{off} 초 동안 전송을 중단하도록 하는데, 이 T_{on} 과 T_{off} 는 각각 T_{on} 과 T_{off} 를 평균으로 하는 지수분포를 따르도록 한다. 2차사용자는 500ms 동안 1차전송이 감지되지 않으면 해당 채널이 유휴상태라고 인지한다.



(a) $\langle T_{on}, T_{off} \rangle = \langle 2, 8 \rangle$



(b) $\langle T_{on}, T_{off} \rangle = \langle 5, 5 \rangle$



(c) $\langle T_{on}, T_{off} \rangle = \langle 8, 2 \rangle$

그림 3. 시간에 따라 전송되는 패킷 일련번호 증가 추이

Fig. 3 Transition of the sequence number of received packets

그리고, 2차송수신자 가운데 하나만 1차전송을 감지하는 상황을 만들기 위해, 2차송수신자가 자신을 간섭하도록 지정된 1차사용자의 MAC 주소를 미리 알고 있게 하여, 그 MAC 주소를 송신자로 하는 패킷이 감지되면 1차전송의 감지로 간주하고, 그렇지 않으면 무시하도록 한다.

2. 표준 TCP vs. TCP-Freeze-CR 성능비교

1차전송의 발생 빈도에 따른 성능의 차이를 관찰하기 위해 우리는 세 가지 형태의 1차전송패턴을

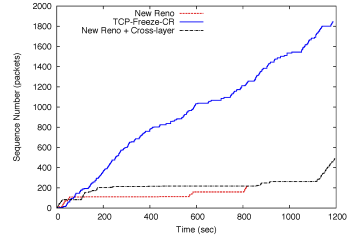


그림 4. 추가고려사항의 효과

Fig. 4 Effect of additional enhancement scheme

적용한다: $\langle T_{on}, T_{off} \rangle = \langle 2, 8 \rangle$, $\langle T_{on}, T_{off} \rangle = \langle 5, 5 \rangle$, $\langle T_{on}, T_{off} \rangle = \langle 8, 2 \rangle$. 측정 계량은 시간에 따른 패킷수신 추이이다. 이를 위해, 우리는 2차수신자가 수신하는 패킷의 일련번호 (sequence number)를 시간에 따라 추적하고, 그 결과를 그림 3에 나타낸다. 그림에서 보여지듯이, TCP-Freeze-CR이 모든 1차전송 패턴에 대해 더 빠른 packet 수신 속도를 보인다. $\langle T_{on}, T_{off} \rangle = \langle 5, 5 \rangle$ 인 경우, 실험이 끝난 시점에 TCP-Freeze-CR이 10배 정도의 패킷을 더 수신했고, 만약, 실험을 더 지속했을 경우에는 이 차이가 더 커질것으로 쉽게 예측된다. 그림 3(c)에서 보듯이, $\langle T_{on}, T_{off} \rangle = \langle 8, 2 \rangle$ 인 경우, 표준 TCP는 18개의 패킷만을 성공적으로 수신한다. 게다가, 그림 3(a)에서 보이듯이, 비교적 낮은 1차전송 빈도에도 300초 정도까지 패킷 수신이 전혀 안되는 것을 관찰할 수 있다.

이는 재전송타임아웃 연속적인 backoff로인해 유휴 채널을 발견, 동기화를 완료했음에도 불구하고 전송을 하지 않고 있기 때문이다. 300초 이후에는 우연히 재전송타임아웃이 유휴채널 사용가능 시점에서 발생하여 전송이 재개된 것이다. 결과적으로, 표준 TCP는 가용 채널을 발견 및 동기화를 완료했음에도 전송이 재개되지 않는 경우가 매우 빈번하고, 반면에 TCP-Freeze-CR은 새로운 채널로 동기화를 완료함과 동시에 대부분 즉각적으로 전송이 재개한다. 또한, 이 측정을 통해 우리는 1차전송이 빈번해질 수록 표준 TCP의 성능 저하는 더욱 심각해짐을 확인한다.

TCP-Freeze-CR은 하위계층이 새로운 채널로 동기화를 완료하여 전송을 재개할 때, cwnd, RTT, RTO 값을 복구하고 재전송타이머를 리셋한다. 그리고, 만약, old_cwnd = 0 일 경우, cwnd = 1로 하고 전송단계를 slow start로 한다. 이는 전송 재

개 직후 전송된 패킷이 손실되거나 acknowledgement가 손실되어 재전송타임아웃이 발생 할 때까지 전송이 진행되지 않는 문제를 완화 하기 위해서이다. 우리는 이 추가고려사항 (4장 말미에 기술되어있음)이 TCP-Freeze-CR에 미치는 영향을 분석한다. 그림 4에 우리는 TCP-Freeze-CR, 표준 TCP, 표준 TCP 에 단순히 크로스레이어 시그널을 사용했을 때 (그래프에서는 New Reno + Cross-layer로 표기) 각각의 패킷 수신 추이를 측정하여 나타낸다. 'New Reno + Cross-layer' 에서 TCP는 freeze 신호를 받았을 때, cwnd, RTT, RTO 값을 저장하고, 전송을 중단 하고, unfreeze 신호를 받았을 때, 저장된 값들을 단순히 복구하고 전송을 재개한다. 참고로 이 실험은 $\langle T_{on}, T_{off} \rangle = \langle 5, 5 \rangle$ 로 하고, 2차송신자만이 1차전송으로부터 영향 받도록 하고 진행한다.

그림에 나와있듯이, TCP-Freeze-CR이 표준 TCP와 'New Reno+Cross-layer' 방법에 비해 월등히 좋은 성능을 보이는 것으로 관찰된다. 더구나, 'New Reno + Cross-layer' 방법은 표준 TCP와 거의 비슷한 성능이 얻어짐이 관찰된다. 따라서, 우리는 추가사항이 인지무선환경에서 TCP 성능을 향상시키는데 매우 중요한 요소임을 확인한다.

VII. 결론

이 논문에서, 우리는 TCP가 인지무선환경에서 겪을 수 있는 성능저하 문제를 논하고, 그 문제를 해결하기 위해 TCP-Freeze-CR을 제안한다. TCP-Freeze-CR은 하위계층으로부터 전달되는 크로스레이어 신호들 (freeze, unfreeze)에 의해 동작을 정지 (freeze)하거나 재개 (unfreeze)한다. 하위계층의 크로스레이어 신호는 1차전송의 감지와 유희스펙트럼의 발견시에 전달된다.

우리는 우리의 제안을 실제 소프트웨어 라디오에 구현하고, 소규모 인지무선네트워크를 구성하여 성능을 검증한다.

우리는 이 논문에서 단일 (single) 홉 인지무선 네트워크만을 고려하는데, 차후, 더 많은 장비를 확보하여 다중 (multi) 홉 인지무선네트워크를 고려한 확장된 TCP-Freeze-CR을 연구할 계획을 가지고 있다. 또한, 성능이 검증된 주파수 감지기법을 연동시켜 1차전송이 다른 물리계층 프로토콜 (예를 들어, IEEE 802.11)을 통해서 이루어지는 경우에 대해서도 고려한다. 또한, 기존의 Freeze-TCP에 동일한

인지무선 기능 (즉, 동일한 하위계층 프로토콜)을 적용하여 TCP-Freeze-CR 과의 성능분석도 고려할 수 있다.

References

- [1] J. Mitola G.Q. Maguire Jr., "Cognitive radio: making software radios more personal," IEEE Personal Communications, Vol. 6, No. 4, pp. 13-18, 1999.
- [2] R. Paul, W. Pak, Y.-J. Choi, "Slectively triggered cooperative sensing in cognitive radio networks," IET Communications, Vol. 8, No. 15, pp. 2720-2728, 2014.
- [3] S. Chepuri, R. de Fransisco, G. Leus, "Performance evaluation of an IEEE 802.15.4 cognitive radio link in the 2360-2400 MHz band," Proceedings of IEEE Wireless Communications and Networking Conference, 2011.
- [4] A. Ayadi, P. Maille, D. Ros, "TCP over low-power and lossy networks: tuning the segment size to minimize energy consumption," Proceedings of IFIP International Conference on New Technologies, Mobility and Security, 2011.
- [5] Y. Tian, K. Xu, N. Ansari, "TCP in wireless environments: problems and solutions," IEEE Communications Magazine, Vol. 47, No. 7, pp. 57-63, 2009.
- [6] T. Issariyakul, L. Pillutla, V. Krishnamurthy, "Turning radio resource in an overlay cognitive radio network for TCP: Greed isn't good," IEEE Communications Magazine, Vol. 47, No. 7, pp. 57-63, 2009.
- [7] D. Chen, H. Ji, V. Leung, "Distributed optimal relay selection for improving TCP throughput over cognitive radio networks: A cross-layer design approach," Proceedings of IEEE International Conference on Communication, 2011.
- [8] K. Chowdhury, M. Di Felice, I. Akyildiz, "TP-CRAHN: a transport protocol for cognitive radio ad-hoc networks," IEEE Transactions on Mobile Computing, Vol. 12, No. 4, pp. 790-803, 2013.

- [9] C. Luo, F. Yu, H. Ji, V. Leung, "Cross-layer design for TCP performance improvement in cognitive radio networks," IEEE Transactions on Vehicular Technology, Vol. 59, No. 5, pp. 2485-2495, 2010.
- [10] S. El Rakabawy C. Lindemann, "A practical adaptive pacing scheme for TCP in multihop wireless networks," IEEE/ACM Transactions on Networking, Vol. 19, No. 4, pp. 975-988, 2011.
- [11] T. Goff, J. Moronski, D. Phatak, V. Gupta, "Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments," Proceedings of IEEE INFOCOM, 2000.
- [12] T. Schmid, "GNU Radio 802.15.4 En- and Decoding," NESL Technical Report, 2006.

저 자 소 개

변상선



1996년 고려대학교 전산학
학사.

2002년 고려대학교 통신
시스템기술 석사.

2007년 고려대학교 전산학
박사.

현재, 부산가톨릭대학교
컴퓨터공학과 조교수.

관심분야: 인지무선네트워크, 임베디드시스템
기반 의료기기

Email: ssbyun@cup.ac.kr