

Wireless Ad-Hoc Network 에서 TCP 성능 향상을 위한 TCP-SF 방안

정성재*, 김문기**, 이병호***
*한양대학교 정보통신학과
e-mail:arpons@ihanyang.ac.kr

TCP-SF : Improving TCP Performance in Wireless Ad-Hoc Networks

Sung-Jea Jung*, Mumgi Kim**, Byungho Rhee***

*Dept. of Information and Communication, HanYang University

요 약

Ad Hoc 무선망은 기존 인프라에 영향을 받지 않으면서, 노드들이 네트워크를 구성하기 때문에 TCP 의 혼잡(Congestion) 제어가 나쁜 영향을 미친다. 기존의 TCP 에 무선망의 요소와 거리에 따른 노드 홉수의 영향을 반영해서 새로운 환경에 적합한 TCP 를 개선해야한다. 본 논문에서는 혼잡의 원인을 구별하여, 패킷 손실을 최소화할 수 있는 알고리즘을 제안한다.

1. 서론

Ad-Hoc 무선망 네트워크(Mobile ad-hoc network)는 기존에 있는 인프라를 사용하지 않고, 임시적으로 네트워크 망을 형성할 수 있다. 특히, 네트워크단말의 이동에 따른 제약이 비교적 적고, 중앙과의 접속점이 필요하지 않기 때문에 네트워크 구성이 빠르고 구축시 비용이 저렴하다. 뿐만아니라 Ad-Hoc 네트워크는 전시장, 생산 현장, 병원, 응급 구조, 군 통신망의 재해, 대규모 무선 회의, 센서 통신망등 여러 분야에서 사용이 가능하다. Ad-Hoc 무선망에 대한 연구는 노드의 이동에 의한 네트워크의 동적변화에 따른 경로 관리를 위한 라우팅 방식에 관한 연구가 주요 연구 분야로 인식되고 있다. Ad-Hoc 무선망은 무선 매체를 사용해서 통신을 하기 때문에 신호의 간섭, 잡음, 환경에대한 데이터 손실이 있는 통신상의 제약이 있다.

TCP 는 현재 유선망에서 가장 널리 사용되고 있는 신뢰성 있는 전송 프로토콜이다. End-to-End 전달 방식을 사용하는 TCP 를 이용한 여러 어플리케이션인 파일 전송, E-mail, 인터넷 검색등을 Ad-Hoc 무선망에서 사용할 경우 신뢰성 있는 서비스를 제공하지 못한다. Ad-Hoc 무선망은 hop 단위로 데이터를 전송하는 hop-by-hop 전달 방식을 사용하기 때문이다. 특히, Ad-Hoc 망에서의 패킷 손실을 TCP 는 혼잡에 의한 손

실로 인식을 한다. 그래서 혼잡 제어를 하기 때문에 전송을 저하시키며 노드의 자원을 불필요하게 소모시키게 된다.

따라서 본 논문에서는 Ad-Hoc 무선망에서 노드의 경로 재설정시 발생하는 패킷 손실을 네트워크 상황을 반영해서 최소화 함으로써 Ad-Hoc 무선망에서 TCP 의 성능을 향상시키는 방안을 제시하고자 한다.

본 논문은 1 장의 서론에 이어, 2 장에서는 Ad-Hoc 무선망에서의 TCP 성능 향상에 관한 연구를 살펴본다. 3 장에서는 재설정된 네트워크 상황을 반영해서 패킷의 손실을 최소화하고, TCP 성능을 개선하는 TCP-SF(TCP-Selective Feedback) 알고리즘을 제안하고자 한다. 4 장에서는 제안한 알고리즘을 시뮬레이션하고 TCP-F와 성능을 비교하고 5 장에서 결론으로 끝맺는다. ¹

2. Ad-Hoc 무선망에서의 TCP 성능 향상을 위한 연구

무선망에서 TCP 성능을 개선하기 위한 많은 방안이 제안되었다. 그러나 이런 방안들은 거의 셀룰러 환경에만 적합하고, Ad-Hoc 무선망에는 적합하지 않

¹ 본 연구는 대학 IT연구센터 육성 지원 사업의 연구 결과로써 HY-SDR 연구센터의 연구비 지원으로 수행되었음

다. 그래서 최근에는 Ad-Hoc 무선망에서 TCP 성능을 향상시키기 위한 연구들이 제시되고 있다. 여러 연구 방안들 중 Feedback 방안과 Heuristic 방안이 있다. [4]

2.1 Feedback 방안

TCP-F(TCP-Feedback)[2]는 Active 상태와 Snooze 상태로 구성된 전송제어 방식이다. Active 상태에서는 TCP 표준에 따라서 제어를 한다. 그리고 중간 노드가 링크 실패를 감지하면 RFN(Rout Failure Notification) 패킷을 소스 노드에게 전송 한다. RFN 을 받은 소스 노드는 더 이상 패킷을 보내지 않고, 재전송 타이머와 혼잡 윈도우 (Congestion Window) 사이즈와 같은 상태 값을 고정시킨 후 ‘SNOOZE’ 상태에 들어간다. 중간 노드로부터 RRN(Route Reconstruction Notification) 패킷을 통해 경로 재설정에 관한 공지를 받을 때까지 SNOOZE 에 있다가 RRN 을 받으면 다시 Active 상태로 들어간다.

TCP-ELEN(Explicit Link Failure Notification)는 혼잡이 발생하는 것처럼 보이는 경로 실패를 방지하는 Feedback 에 기반한 방식이다. TCP 소스에 경로 실패에 관한 정보를 제공하기 때문이다. 응답하는 것을 피할 수가 있다. TCP 소스가 ELFN 을 받으면, 재전송 타이머를 중단시키고, stand-by 모드에 들어간다. 주기적으로 패킷은 네트워크 길이 설정되었는지 알기 위해 보내어진다. 새로운 ACK(새로운 길이 설정된 것을 의미)을 받은 이후, 소스는 '스탠바이'에 모드에서 빠져나와서 전송 타이머를 복원하고, TCP 전송을 정상적으로 처리한다.[1]

TCP-BUS[3]는 TCP-F 의 매커니즘이 동일하고, 불필요한 혼잡 처리를 방지하기 위해 명확한 통지 매커니즘을 사용한다. 경로 실패가 감지되면, 노드는 ERDN (Explicit Route Disconnection Notification) 을 TCP 소스에 전송한다. 그리고 새로운 경로가 재설정되면 TCP 소스는 ERSN (Explicit Route Successful Notification)을 받게되고, TCP 전송을 재개한다. TCP-BUS 는 ERDN 과 ERSN 의 손실에 관해 대처할 수 있다.

2.2 Heuristic 방안

Fixed RTO 는 하위 계층으로부터 피드백이 필요 없다. 대신 경로 실패와 혼잡을 다음과 같이 식별한다. 타임 아웃이 연속적으로 발생할 때, 소스는 네트워크 혼잡이 발생 한 것이 아니라 경로 실패가 일어났다고 간주한다. 이때 UNACK (Unacknowledged) 패킷이 재전송 되지만 RTO (Retransmission Timeout) 는 두 배가 되지 않는다. 경로가 재설정되고 재전송된 패킷이 도달할 때까지 RTO 는 변하지 않는다.

DOOR 매커니즘은 각각의 TCP 송신자와 TCP 수신자의 데이터 패킷 순서와 ACK 의 시퀀스 번호 순서를 분석해서 경로의 실패/변화를 감지한다. 그리고, 일시적으로 혼잡 제어 또는 즉시 복구를 무능하게 하는 조치를 취한다.

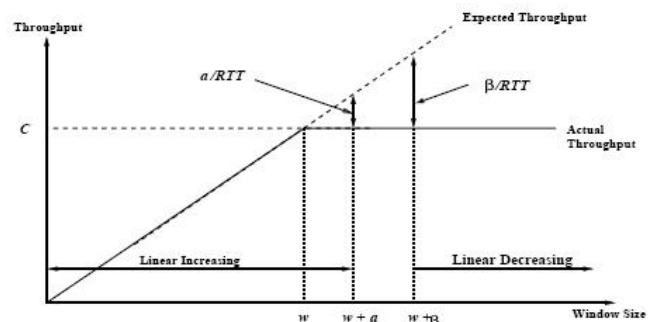
이 두 가지 방안들은 차이가 있지만, 경로 실패와 재설정을 감지한 후 취하는 조치가 유사하다. 즉, 혼잡 제어를 일시적으로 중단 시키고, 안전한 TCP 연결

을 유지하기위해 TCP 상태 조절 메커니즘 없이 TCP 전송을 재개할 수 있다. 본 논문에서는 재설정된 경로의 네트워크를 반영하지 못하고 RTO 와 Congestion Window 의 크기를 이전 경로의 것을 그대로 사용함으로써 Ad-Hoc 무선망의 특성을 반영하지 못하는 TCP-F 의 문제점을 개선하고자 한다.

2.3 Windows Control of TCP

본 논문에서 제안하는 알고리즘은 기본적인 TCP 혼잡 제어 알고리즘을 기반으로 하였다. [5]

$$CWND = \begin{cases} CWND + 1 & \text{if } Diff < \alpha \\ CWND - 1 & \text{if } Diff > \beta \\ CWND & \text{otherwise} \end{cases}$$



(그림 1) Windows Control of TCP

3. TCP-SF (TCP-Selective Feedback) 방안

본 절에서는 소스 노드가 패킷 손실의 원인을 구별하는 TCP-Vegas 의 기반해서 혼잡인 경우와 채널 에러에 의한 손실인 경우를 구분한다. TCP-F 가 선택적으로 경로의 혼잡 상황을 정확히 반영할 수 있도록 하는 TCP-SF(Selective Feedback)를 제안해서 Ad-Hoc 무선망에서 TCP 의 성능을 향상시킨다. 이전 알고리즘들은 혼잡이 발생하면 전송률을 1/2 로 감소시킨 후 재전송을 하였다. 이러한 전송률 방식은 제공하는 서비스에 알맞지 못한 방식이다. 제공되는 서비스에 따른 전송률 기법이 필요하다. 혼잡이 발생했을 때 이 혼잡이 어떤 혼잡인지 구별해주는 방식이 필요하다. TCP-Vegas 는 송신자가 패킷 손실의 원인을 구별하는 Vegas Loss Predictor 에 사용된 알고리즘이다. 이 TCP-Vegas 알고리즘을 사용함으로써 패킷 손실의 원인이 혼잡인지, 채널 에러에 의한 것인지 알 수가 있다. 혼잡일 경우는 네트워크의 상황을 반영해서 TCP-F 를 변형해서 TCP 의 성능을 향상시킨다. 즉, CWD (Congestion Window), SSTHRESH (Slow Start Threshold), RTO (Retransmission Timeout) 값들을 사용해서 변형된 값으로 통신을 한다. TCP-Vegas 를 기반으로 해서 다음과 같이 TCP-SF 를 제안하고자 한다.

$$CWND_{After} = (BaseRTT + \frac{CWND_{before}}{Actual} \times (1 - \frac{BaseRTT}{RTT})) \times Bandwidth \quad (1)$$

$$RTT = BaseRTT + \frac{N}{Actual} \quad (2)$$

Actual : 현재 전송률

CWND_{before} : 현재 Congestion Window Size

BaseRTT : 측정된 RTT

혼잡이 발생 할 경우 식(1), (2)을 사용해서 CWND 값을 비교해서 큰 CWND 값을 가지고 Threshold 값을 변형한다. 이 값을 새로 설정된 Ad-Hoc 무선망에 적용한다. 알고리즘을 살펴보면 다음과 같다. 여기서 α 값은 0.75 (새로운 경로의 CWND 값이 이전 경로의 75% 값인 경우)을 의미한다.

```

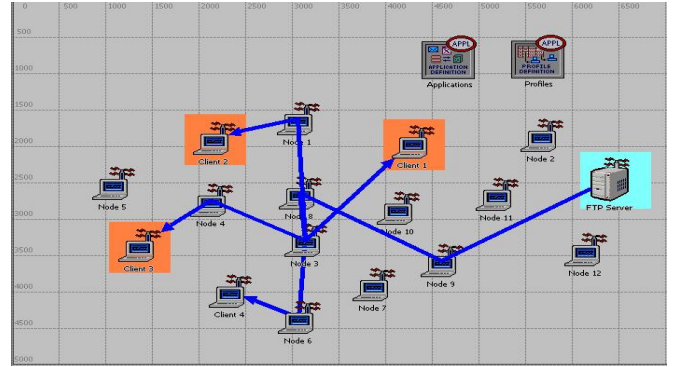
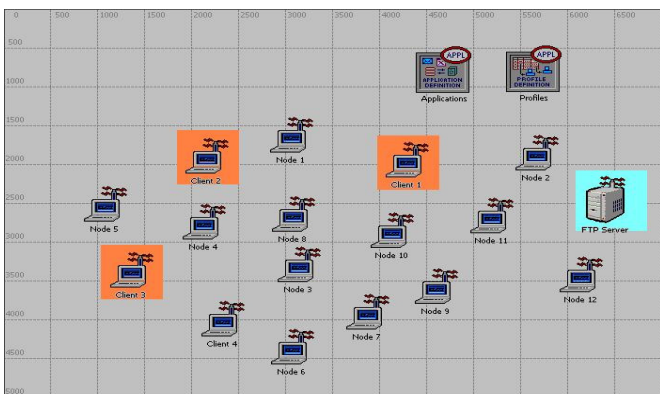
Re-Transmission: TCP-SF
State == SNOOZE
REQUEST: RRN
if (CWND_BEAF < CWND_AFT) {
    Re-Transmission Use: TCP-Reno
    CWND = CWND_AFT / 2
    if (CWND > SSTHRESH) {
        SSTHRESH = (CWND/2) *  $\alpha$ 
        CWND = SSTHRESH
    }
} else {
    SSTHRESH = (CWND/2) *  $\alpha$ 
} else {
    SSTHRESH = CWND_BEAF/2
    CWND = (CND_BEAF / 2) *  $\alpha$ 
}
    
```

(그림 2) TCP-SF Retransmission Algorithm

4. TCP-SF (TCP-Selective Feedback) 실험결과

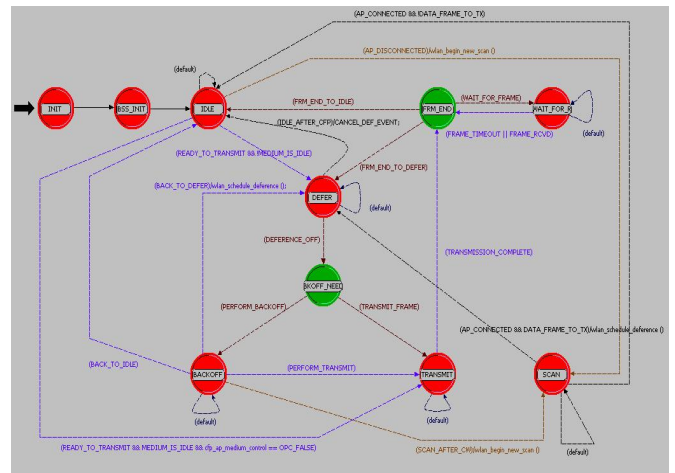
4.1 무선망 실험 환경

본 논문에서는 TCP-SF 알고리즘을 사용해서 Ad-Hoc 무선망에서 인터넷 트래픽의 처리율을 분석하기 위해 TCP/Reno 를 사용하였다. 또한 시뮬레이션 환경은 다음과 같다. 무선망의 구성은 임의의 이동 노드들과 경로를 설정하였다. 전체 대역폭은 512Kb/s 로 설정



(그림 3) DSR 라우팅 프로토콜이 적용된 Mobile Ad-Hoc Network 구조

하였다. 패킷의 크기는 100 bytes 로 하고, 노드간 지연은 5ms 이다. 또한 무선 Ad-Hoc 망에서 FTP Server 제공할때의 성능을 측정하였다. 각각의 Client (MN)은 4 개를 대상으로 측정하고, 다운로드 시간은 30 초, 35 초, 40 초, 45 초 각각 5 초 간격으로 설정하였다. 그리고 Routing Protocol 은 DSR 을 사용하였다.

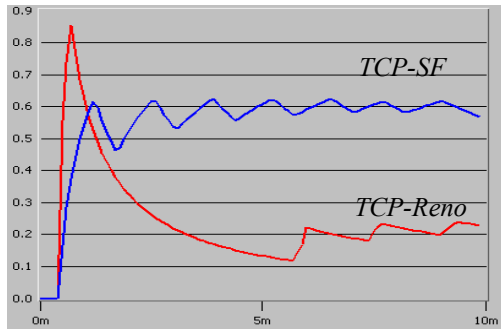


(그림 4) OPNET Modeler 11.0 에서의 Wireless Lan Mac 모델

다음은 (그림 4)에서 각 상태에 대한 구체적인 설명이다.

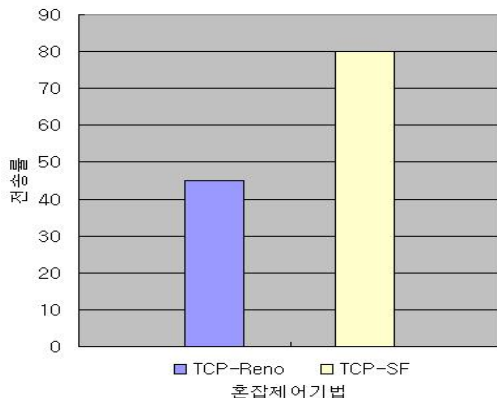
- INIT : 시뮬레이션 초기화 작업
- BSS_INIT : 모든 노드에 대한 BSS 설정
- IDLE : 대기 상태
- DFFER : 프레임 전송전의 ISF 대기상태
- BAKOFF_NEED : Bakeoff 과정의 수행여부
- BAKOFF : 랜덤으로 측정된 Bakeoff Slot 동안을 대기하는 상태
- TRANSMIT : 프레임을 전송하는 상태
- FRM_END : 프레임 전송 완료
- WAIT_FOR : 전송한 프레임에 대한 응답요청 대기 상태
- SCANN : Roaming 시에 주변 AP 검색상태

4.2 분석



(그림 5) MN의 초당 TCP Packet 재전송률 (TCP-SF vs. TCP-Reno)

(그림 5)와 같이 TCP-Reno는 재전송률이 처음의 1/2로 돌아가는데 10초 이상이 걸린다. 하지만, TCP-SF는 임계값 α 를 설정해두었기 때문에 재전송률이 TCP-Reno에 비해 더 빠른 것을 확인할 수가 있다. 이것은 TCP-SF가 손실의 원인에 따른 재전송률이 선택적으로 작동하기 때문이다.



(그림 6) TCP-Reno 대 TCP-SF의 전송률

Ad-Hoc 무선망에서 FTP 서비스를 제공받기위해 망이 구성된 상황에서 각 MN1, MN2, MN3, MN4를 선택하였다. (그림 6)에서 보듯이 전송률이 TCP-SF가 더 뛰어난 것을 볼 수 있다. 각각의 MN이 이동시 TCP에서 패킷의 손실이 발생하면 TCP/Reno는 전송률을 1/2로 감소시킨 후 다시 재전송을 한다. 그러면 처리율이 떨어진 후 다시 원상태로 가는데, 지수적인 형태의 시간 그래프가 형성된다. TCP-SF는 TCP-Reno에 비해 성능이 30% 우수한 것을 알 수 있다. 또한 Ad-Hoc 네트워크에서 노드들의 거리가 멀어질수록 처리율이 떨어진다. 반면에 TCP-SF를 적용한 알고리즘을 사용할 경우, 처리율의 변화가 급격하지 않은 것을 알 수 있다. 즉 TCP-SF가 패킷의 손실에 따른 원인을 선택적으로 구별해서 송신률을 변경하기 때문이다.

5. 결론

본 논문에서는 OPNET 시뮬레이터에 TCP-SF 모듈을 구현하고 이를 사용해서 Mobile Ad-Hoc 무선망

에서의 성능을 분석하였다. 또한 분석 시뮬레이션을 구성하기위해 DSR 라우팅 프로토콜을 사용하였다. DSR 라우팅 프로토콜은 라우팅 소스를 기반으로 한 on-demand 라우팅 프로토콜로 모바일 노드(MN)들은 소스 경로를 포함한 경로 캐쉬를 가지고 있다. 따라서 본 논문에서 FTP를 제공하는 TCP의 성능을 측정하기에 적당하다.

분석결과 Ad-Hoc 무선망에서 TCP의 패킷 손실의 원인을 혼잡에 의한 것으로 인식하기 때문에 성능이 저하되는 경향이 발생한다. 그래서 이러한 문제를 해결하기 위해 송신자에서 손실의 원인을 구별하는 기법들이 제안되었다. 하지만, 노드가 많아지고, 거리가 멀어지는 경우에 있어서의 상황에서는 성능이 떨어지는 단점이 있다.

본 논문에서는 RTT의 변화량과 Actual 값을 반영해서 노드의 거리와 환경변화에 따른 전송률을 선택적으로 조절함으로써 더 좋은 성능이 나타나는 것을 확인하였다. 앞으로의 과제는 충돌 상황과 노드의 배터리를 고려한 라우팅 알고리즘과 성능향상에 대한 연구와 UDP의 경우에서의 Ad-Hoc 무선망에서의 멀티미디어 서비스와 같은 서비스를 제공할 때의 성능 향상을 위한 방안과 측정 방식, Handoff를 고려한 처리방안에 대한 연구가 수행되어야 할 것이다. 제시되어야 할 것이다.

참고문헌

- [1]유시환, 최진희, 유혁, “무선 애드 혹 네트워크를 위한 TCP” 한국정보과학회 학술발표논문집 2003년도 가을 (2003년 10월)
- [2]Bing Zhang; Shirazi, M.N.; Komiyama, B, “An ELFN-based TCP-freeze scheme using the route information of sender node for ad hoc networks”, Communications, 2004 and the 5th International Symposium on Multi-Dimensional Mobile Communications Proceedings. Volume 1, 29 Aug.-1 Sept. 2004 Page(s):457 - 461 vol.1
- [3]Dongkyun Kim; Toh, C.-K.; Yanghee Choi, “TCP-BuS: improving TCP performance in wireless ad hoc networks”, Communications, 2000. ICC 2000. 2000 IEEE International Conference on Volume 3, 18-22 June 2000 Page(s):1707 - 1713 vol.3
- [4]Jain, A.; Pruthi, A.; Thakur, R.C.; Bhatia, M.P.S. “TCP analysis over wireless mobile ad hoc networks”, Personal Wireless Communications, 2002 IEEE International Conference on 15-17 Dec. 2002 Page(s):95 - 99
- [5]Jeonghoon Mo; Richard J.La, Venkat Anantharam, Jean Walrand. “Analysis and Comparison of TCP Reno and Vegas