# 유선과 무선 네트워크 환경에서 DCCP를 이용한 QoS 성능분석

서두옥° 이동호
광운대학교 컴퓨터과학과
{clickseo°, dhlee}@cs.kw.ac.kr

# QoS Performance Analysis in Wired and Wireless Networks Using DCCP

Doo-ok Seo° Dong-ho Lee
Dept.of Computer Science, Kwangwoon University

요 약

최근 UDP를 이용한 실시간 어플리케이션이 증가하고 있지만, 대부분 혼잡제어를 사용하지 않고 있다. 이러한 문제를 해결하기 위하여 IETF에서는 UDP를 대신하면서 실시간 트래픽 전송을 위하여 혼잡제어 메커니즘을 포함한 DCCP(Datagram Congestion Control Protocol)를 제안하였다. DCCP는 윈도우 기반 흐름제어를 사용하여 실시간 트래픽을 지원하기 위하여 제안된 차세대 전송 프로토콜이다.

본 논문에서는 DCCP의 기본 개념과 혼잡제어 기법을 소개하고, 유선과 무선에서 다양한 전송 프로토콜을 이용하여 DCCP의 성능평가를 한다. DCCP를 기반으로 유선과 무선 네트워크 환경에서 시뮬레이션 성능분석을 위해 Network Simulator 2(NS-2)를 이용한다.

## 1. 서 론

최근 인터넷상에서 새로운 형태의 다양한 실시간 어플리케이션들이 폭발적으로 증가하고 있다. 현재 대부분의 실시간 어플리케이션은 오디오, 비디오 그리고 온라인 게임과 같은 다양한 형태의 데이터들을 포함한 복잡한 멀티미디어 전송을 제공하고 있다. 이와 같은 트래픽들의 짧은 지연을 통한 전송기법의 중요성은 점점 더 증대되고 있다.

현재 실시간 어플리케이션 전송을 위해 User Datagram Protocol(UDP)이 사용되고 있다. UDP의 장점은 복잡한 혼잡제어 기법을 포함하지 않는 단순한 메커니즘을 사용하므로 구현이 쉽다는 것이다. 하지만 혼잡제어 기법을 제공하지 않으므로써 Transmission Control Protocol(TCP) 트래픽과의 불공평성을 야기한다. 또한 TCP 프로토콜은 혼잡제어와 흐름제어, 그리고 재전송 기법들과 같은 많은 기능들로 인해 복잡한 소프트웨어 프로토콜 스택으로 구성되어 있다. 또한 TCP의 지연유발 정책(재전송, Slow Start 등)들은 심각한 문제점으로 대두되고 있다. 그리고 UDP 프로토콜은 흐름제어 등의 기능이 없는 경량화 된 프로토콜인 반면에 신뢰성을 보장하지 못한다.

이와 같은 문제점을 해결하기 위해서 UDP에 약간의 신뢰성을 부여하기 위한 목적으로 TCP-like와 TCP-friendly와 같은 응용 계층의 혼잡제어 기법들이 제안되었으며, 현재 혼잡제어 기법을 제공하고 UDP에 기반 하여 실시간 트래픽을 수용할 수 있는 Real-Time Transport Protocol(RTP), RTP Control Protocol(RTCP) 그리고 TCP 트래픽과 대역폭을 공유할 수 있도록 Additive Increase and Multiplicative Decrease(AIMD) 알고리즘을 적용한 Rate Adaptation Protocol(RAP) 등이 이용되고 있다. 하지만 이러한 프로토콜은 어플리케이션 계층에서 구현되어 사용자에 의하여 수정될 수 있는 안전성의 문제점을 안고 있다. 또한 신뢰성과 순차적 전송 및 멀티호밍 전송 기능을 제공하는 Stream Control Transmission Protocol(SCTP)이 있지만 능률성이 떨어진다.

Internet Engineering Task Force(IETF)에서는 사용자에 의한 수정 없이 전송 계층에서 직접적으로 혼잡제어를 제공하기 위하여 오랜 논의 끝에 2006년 3월에 최종적으로 Datagram Congestion Control Protocol(DCCP)를 표준으로 제정하였다[1].

본 논문에서는 네트워크 시뮬레이터인 NS-2를 이용하여 유선과 무선 네트워크 환경에서 DCCP를 기반으로 성능평가를 하였다. 2장에서는 DCCP의 기본 개념과 혼잡제어 기법을 소개하고, 3장에서는 DCCP를 이용하여 유선과 무선 환경에서 DCCP의 성능을 평가하고 분석한다. 4장에서는 결론 및 향후 연구되어져야 할 내용들을 논한다.

## 2. DCCP의 기본 개념

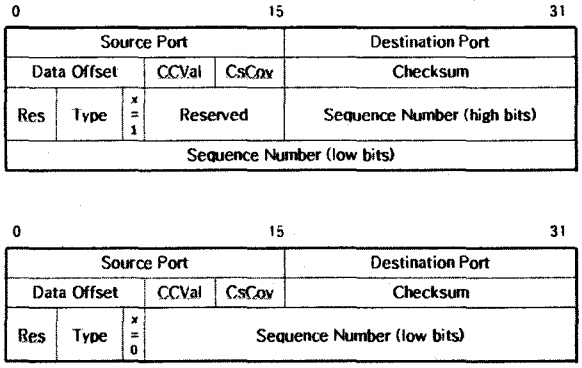
DCCP는 2001년 7월에 IETF에서 Datagram Control Protocol(DCP)로 처음 소개되었다. IETF는 2002년 3월에 DCCP로 이름을 바꾸고 새로운 DCCP working group을 만들었다[2]. DCCP는 TCP와 같은 신뢰전송과 순차전송의 특징을 제공하지 않지만, 실시간 어플리케이션을 위하여 혼잡제어 기능을 제공하는 전송 프로토콜이다. 현재 Linux와 FreeBSD 운영체제에서 구현 되었다.

실시간 어플리케이션은 적은 오버헤드와 빠른 전송 프로토콜을 필요로 하기 때문에 DCCP는 단순하고 low delay 그리고 혼잡 발생에 빠르게 반응하기 위하여 설계되었다. 또한 주목할 만한 특징으로 복합적인 혼잡제어 알고리즘 중에서 선택할 수 있으며, 현재 TCP-like 혼잡제어 및 TCP-friendly 혼잡제어 알고리즘의 두 가지 혼잡제어 방식이 표준화 되어 있다. 혼잡제어를 위해 통신단말간에 연결 상태 정보를 기억하며, 이를 확장하여 DCCP 기반 이동성 등의 다양한 응용기법이 논의되고 있다[3].

### 2.1 패킷 형식 및 연결

DCCP 패킷은 일반적인 헤더와 어플리케이션 데이터로 구성되는데 어플리케이션 데이터는 불필요하다. 헤더의 길이는 12 ~ 1020 bytes의 가변적인 길이를 갖는다. DCCP 헤더는 일반적인 헤더, 타입에 의존한 추가적인 필드들 그리고 옵션의 세 부분으로 구성된다. DCCP의 일반적인 헤더는 12 또는 16 bytes 이며, [표 1]과 같다[1].

[표 1] DCCP의 일반적인 헤더




[표 1]에서 "CCVal"(Congestion Control Indetifier Value)은 연결에 적용된 혼잡 알고리즘을 의미하고, "type"은 패킷의 유형이다. 그리고 "x"는 확장된 sequence number bit를 의미한다. "x"의 값에 의존한 일반적인 헤더의 타입은 두 가지가 있다. 만약 "x"가 0 이면 sequence number 필드는 24비트이고, "x"가 1 이면 나머지 24비트가 확장되어 총 48비트가 된다.

2.2 패킷 유형 및 연결

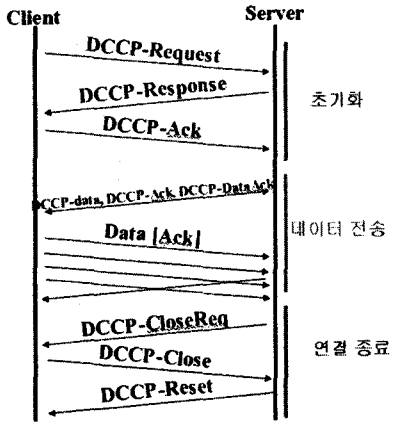
프로토콜의 유연성을 제공하기 위하여 10개의 다양한 패킷형태를 정의하고 있다. DCCP에 정의된 10가지 패킷 유형은 [표 2]와 같다[1].

[표 2] DCCP 패킷 유형

| | | | |
|---|---|---|---|
| 0 | DCCP-Request | 1 | DCCP-Response |
| 2 | DCCP-Data | 3 | DCCP-Ack |
| 4 | DCCP-DataAck | 5 | DCCP-CloseReq |
| 6 | DCCP-Close | 7 | DCCP-Reset |
| 8 | DCCP-Sync | 9 | DCCP-SyncAck |
| 10 - 15 | Reserved | | |

전형적인 DCCP의 연결 절차는 다음과 같다.

(1) 클라이언트는 서버에 클라이언트와 서버의 포트, 요구하는 서비스 등을 명시한 DCCP-Request 패킷을 보내고, 서버와 CCID 등을 협상한다.
(2) 서버는 클라이언트와 통신하기 위하여 클라이언트에게 DCCP-Response 패킷을 보낸다.
(3) 클라이언트는 DCCP-Response 패킷의 응답으로 DCCP-Ack 패킷을 서버에게 보내므로써 서버와 클라이언트간의 연결을 설정한다.
(4) 서버와 클라이언트는 DCCP-Data 패킷을 교환하고, 만약 클라이언트가 더 이상 보낼 데이터가 존재하지 않으면 서버는 DCCP-Data와 DCCP-DataAck 패킷을 보낼 것이다.
(5) 서버는 DCCP-CloseReq 패킷을 보냄으로써 연결 해제를 요구한다.
(6) 클라이언트는 DCCP-Close 패킷을 보냄으로써 연결 해제에 동의한다.
(7) 서버는 Reset Code 1 "Closed"와 함께 DCCP-Reset 패킷을 보냄으로써 연결 상태를 종료한다.




[그림 1] 전형적인 DCCP 연결 절차

2.3 혼잡제어 메커니즘

현재 TCP-like에 기반한 Congestion Control ID 2(CCID2)와 TCP-Freindly Rate Control(TFRC)에 기반한 Congestion Control ID 3(CCID3) 의 두 가지의 혼잡제어 메커니즘이 표준화 되어 있다[4,5]. CCID2은 윈도우 기반 흐름제어를 위한 AIMD 알고리즘에 기반한 DCCP의 혼잡제어 메커니즘의 하나이다. CCID3은 TCP-Friendly Rate Control(TFRC) 이다.

## 3. 실험 및 성능 평가

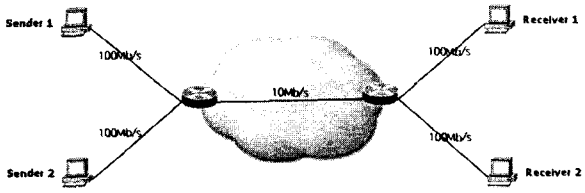
본 장에서는 IETF에서 TCP와 UDP의 장단점을 수용하여 표준화된 DCCP에 대해서 NS-2 시뮬레이터를[6,7] 통해서 실험 및 성능 평가를 수행하였다. 현재 NS-2 에서는 DCCP 에이전트를 지원하지 않아 DCCP 모듈을 추가적으로 패치 하여 성능 평가를 수행하였다[8].

유선 네트워크 환경에서 다양한 혼잡 상황을 고려하여 기존 UDP를 대체할 수 있는 차세대 전송 프로토콜로써 DCCP의 성능 평가를 통하여 가능성을 점검해본다. 또한 기존 TCP를 무선에 적용했을 때 혼잡에 의해 발생한 패킷 손실과 비트 에러에 의한 패킷 손실을 구분하지 못하는 문제점을 안고 있었다. DCCP를 무선 네트워크 환경에 적용하였을 때의 성능평가를 통하여 무선 네트워크 환경에 적용하였을 때 DCCP의 문제점을 점검해본다.

3.1 유선망에서의 실험 및 성능 평가

유선 네트워크 환경에서 시뮬레이션 환경은 네트워크 토폴로지에서 2개의 송신자와 2개의 수신자 노드를 가지며, 각각의 송신자의 데이터 흐름은 하나의 수신자와 대응한다. 병목 링크는 10Mb/s의 대역폭을 가지며, 다른 모든 링크는 100Mb/s의 대역폭을 갖는다. 또한 모든 데이터의 패킷 사이즈는 헤더를 포함하여 500bytes 이며, 모든 시뮬레이션 주기는 60s로 한다.

시뮬레이션을 위한 전송 프로토콜은 UDP, TCP 그리고 DCCP 전송 프로토콜을 선택했으며 TCP는 Sack 알고리즘을 사용하였다. UDP와 TCP 패킷의 성능분석을 위해 UDP 패킷을 먼저 시작한 후에 TCP 패킷은 5s후에 시작한 후 성능을 비교분석한다. 그리고 TCP와 DCCP 패킷, 그리고 두 가지의 DCCP 패킷도 동일한 시나리오를 이용한다. 시뮬레이션의 전체적인 네트워크 구성도는 [그림 2]와 같다.

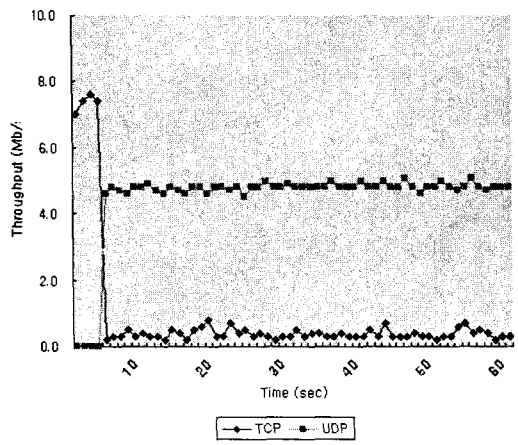



[그림 2] 유선망에서의 네트워크 토폴로지

[그림 3]은 UDP와 TCP의 혼잡 상황 발생시 시간에 따른 패킷 처리량을 나타낸 그래프이다. 0s ~ 5s 에서는 TCP가 병목 링크를 활용하다가 5s에 UDP 패킷이 시작되면서 급격하게 손실 되는 것을 볼 수 있다. 5s에 혼잡 상황이 발생하면서 TCP 패킷은 충분한 대역폭을 확보하지 못하고 UDP 패킷이 병목 링크의 10Mb/s를 거의 90%이상 점유하는 것을 볼 수 있다.

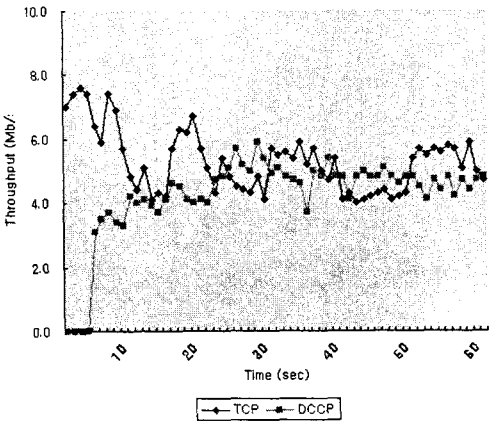
최근의 인터넷 환경은 UDP를 사용하는 실시간 어플리케이션의 사용이 증가되고 있지만, UDP는 혼잡 발생 시 TCP패킷을 보장하지 못하므로, 인터넷 환경에서 트래픽의 대부분을 차지하고 있는 TCP와 혼잡 발생 시 적절하지 못함을 알 수 있다.

UDP의 문제점을 보완하기 위하여, UDP에 혼잡제어 기능을 추가한 DCCP와 TCP는 혼잡 발생 시 어떠한 결과를 보일지를 분석하였다. [그림 2]의 전체적인 네트워크 구성도를 이용하여 TCP와 UDP의 성능분석 시 사용한 시뮬레이션 시나리오를 적용하였다. [그림 4]는 TCP와 DCCP의 혼잡 발생시 시간에 따른 패킷 처리량을 나타낸 그래프이다. DCCP는 UDP와 반대로 혼잡 상황이 발생하였을 때 기존 TCP 패킷을 주어진 대역폭 안에서 수용하여 대역폭을 조절하는 것을 확인할 수 있다.

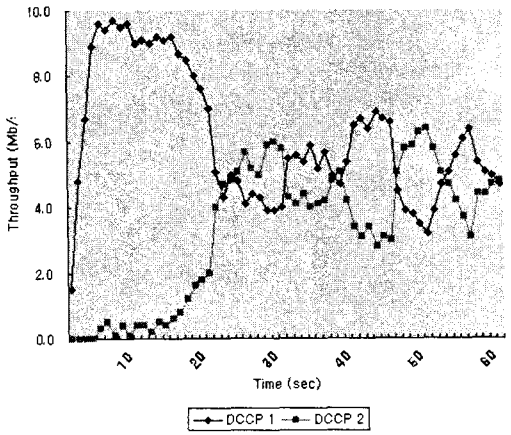



[그림 3] TCP와 UDP

[그림 5]는 두 가지 DCCP 패킷 사이에서 혼잡 상황 발생 시 시간에 따른 패킷 처리량을 나타낸 그래프이다. 대체적으로 두 가지 DCCP 패킷 사이에서 안정적으로 대역폭을 활용하고 있는 것을 볼 수 있다. 하지만 두 번째 DCCP 패킷이 발생된 5s 시점부터 15s사이에 DCCP2는 DCCP1과 비교하여 대역폭을 매주 적게 활용하고 있는 것을 볼 수가 있다. Slow-Start절에서 충분한 대역폭을 획득하지 못하기 때문이다.




[그림 4] TCP와 DCCP



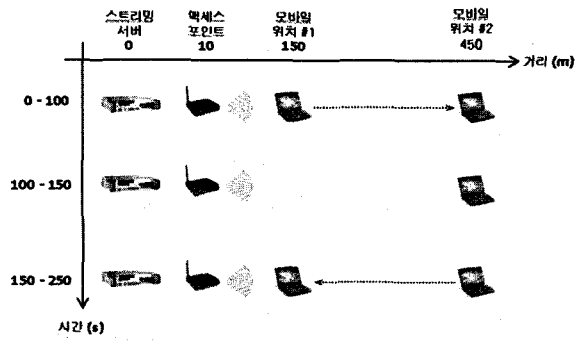

[그림 5] DCCP와 DCCP

### 3.2 무선망에서의 실험 및 성능 평가

기존 TCP는 유선 네트워크에서 혼잡의 주된 요인을 패킷 손실로 간주하였다. 하지만 무선 네트워크에서는 물리적 계층에서 패킷 손실이 패킷 손실이 주로 발생하기 때문에 버퍼 오버플로우와 비트에러에 의한 패킷 손실을 구분하지 못하는 문제점이 있었다. 지난 몇 년간 무선 네트워크에서 TCP 성능향상을 위해 다양한 연구가 시도 되었다[9].

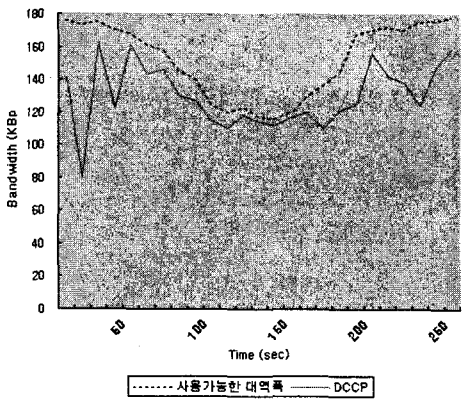
유선 네트워크 환경에서 혼잡 상황 발생 시 UDP보다 향상된 성능을 보여준 DCCP를 무선 네트워크 환경에 적용시 TCP와 동일한 문제점이 발생하는지를 살펴보았다. 무선 네트워크 환경에서 시뮬레이션 환경은 유선 스트리밍 서버, 액세스 포인트 그리고 모바일 노드로 구성한다. 모바일 노드의 이동에 관한 시뮬레이션 시나리오는 [그림 6]과 같다.

(1) 모바일 노드는 액세스 포인트로부터 직선으로 이동한다.
(2) 100s에 모바일 노드는 이동을 중지하고 50s동안 머무르게 된다.
(3) 250s에 모바일 노드는 시뮬레이션 시작 위치로 되돌아 온다.




[그림 6] 무선망에서의 시뮬레이션 시나리오

[그림 7]은 모바일 노드의 이동에 따른 3가지의 상태를 보여주고 있다. 처음 50s에서 100s 사이는 throughput이 사용 가능한 대역폭 보다 적게 나타나고 있다. 그 이후 100s과 150s 사이는 사용 가능한 대역에 적합한 것을 볼 수가 있다. 실험결과를 통하여 DCCP도 기존 TCP처럼 무선망에 바로 적용하였을 때 혼잡에 의해 발생한 패킷 손실과 비트에러에 의해 발생한 패킷 손실을 구분하지 못함으로써 주어진 대역폭을 전부 활용하지 못하고 있음을 알 수 있다.




[그림 7] Throughput 비교

## 4. 결론 및 향후과제

최근 인터넷에서 실시간 어플리케이션의 증가로 인하여 UDP와 TCP의 혼잡 발생 시 야기된 문제점을 전송계층에서 DCCP를 통하여 프로토콜로 제공함으로써, 혼잡 제어를 지원하기 위해 UDP와 비슷한 동작을 필요로 할 때 최소한의 메커니즘만 추가하면 되기 때문에 차세대 전송 프로토콜로써 UDP를 대체할 가능성이 있다. 또한 표준화의 문제점이 2006년 3월에 마무리 되어 DCCP는 TCP와 UDP의 장점을 통합하고 지난 20년간의 혼잡 제어의 성취를 이용할 수 있게 한다.

본 논문에서는 네트워크 시뮬레이터인 NS-2를 이용하여 유선과 무선 네트워크 환경에서 DCCP를 기반으로 성능평가를 하였다. 유선 네트워크 환경에 혼잡 상황 발생시 기존 TCP 패킷을 주어진 대역폭 안에서 수용하여 대역폭을 조절하는 것을 확인할 수 있었다. 하지만 무선 네트워크 환경에서는 기존 TCP와 마찬가지로 버퍼 오버플로우와 비트에러에 의한 패킷 손실을 구분하지 못하는 문제점이 있었다.

향후 연구 계획으로는 유무선 통합 망과 이기종 무선 망간에 DCCP를 적용하기 위한 연구를 진행할 것이다. 또한 혼잡제어를 위해 통신단말간에 연결 상태 정보를 기억하므로, 이를 확장하여 DCCP 기반 이동성 등의 다양한 연구를 진행할 것이다.

## 5. 참고문헌

[1] S. Floyd, M. Handley, E. Kohler, Datagram Congestion Control Protocol (DCCP), March. 2006. RFC 4340.

[2] IETF DCCP working group, http://www.ietf.org/html.charters/dccp-charter.html.

[3] E. Kohler, S. Floyd, Datagram Congestio Control Protocol(DCCP) Overview, http://www.icir.og/kohler/dcp, July 2003.

[4] E. Kohler, M. Handley, S. Floyd, Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control, March. 2006. RFC 4341.

[5] S. Floyd, E. Kohler, J. Padhye, Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC), March. 2006. RFC 4342.

[6] The VINT project : The network simulator - ns2, http://www.isi.edu/nsnam/ns/.

[7] The Network Simulator ns-2 : The NS nanual, http://www.isi.edu/nsnam/ns/ns-documentation.html.

[8] Wireless DCCP in NS2 version 2.28 and 2.29, http://lifc.univ-fcomte.fr/~dedu/ns2/

[9] Minghua Chen, Avideh Zakhor, Rate control for streaming video over wireless, IEEE Wireless Communication, pages 32-41, Aug. 2005.

[10] S. Takeuchi, H. Koga, K. Iida, Y. Kadobayashi, S. Yamaguchi, Performance evaluations of DCCP for bursty traffic in real-time applications, Applications and the Internet 2005 (SAINT'05), pages 142-149, Jan. 2005.

[11] S. Linck, E. Mory, J. Bourgeois, E. Dedu, F. Spies, Video quality estimation of DCCP streaming over wireless networks, In 14th Euromicro International Conference, Parallel Distributed and Network-Based Processing (PDP 2006), Feb. 2006.

[12] Changbin Xu, Ju Liu, Caihua Zhao, Performance analysis of transmitting H.263 over DCCP, VLSI Design and Video Technology, 2005 IEEE International Workshop, pages 328-331, May 2005.