

# 다중 송신자 환경에서 푸시 서비스 기반 파일 전송 기법

(File Transmission using Push Service in Multi-source Environment)

안동혁\*

(Dong Hyeok An)

## 요약

스마트 단말(스마트폰, 태블릿 PC 등)들의 수가 증가함에 따라, 사용자들 간의 멀티미디어 콘텐츠 공유도 증가하고 있다. 스마트 단말 사용자들은 사용자 간 멀티미디어 콘텐츠 전송을 위해서 개별 앱들을 사용하고 있다. 하지만, WiFi나 3G에 연결되어 있는 단말들은 NAT(Network Address Translator)를 사용해 인터넷에 연결되기 때문에 단말 간의 직접적인 연결이 불가능하다. 이와 같은 한계를 극복하기 위해서, 본 논문에서는 추가적으로 서버를 활용하지 않고 거리의 제약이 없도록 푸시 서비스 기반의 파일 전송 기법을 제안하고 있다. 두 번째로, 수신 단말의 링크 대역폭을 기반으로 송신 단말의 전송량을 조절하는 혼잡 제어 기법을 제안하고 있다. 세 번째로, 서로 다른 대역폭을 가진 송신자들을 위해 대역폭 인지 기반 전송 기법을 제안하고 있다. 마지막으로 본 논문에서는 앱을 구현하고 스마트 단말을 이용해 실험을 수행함으로써, 제안하는 기법의 성능을 검증하였다.

■ 중심어 : 혼잡제어 ; 푸시 서비스 ; 다중 송신 ; 스마트 단말 ; 대역폭 인지 기반 전송

## Abstract

The increased number of smart devices leads users to share multimedia contents with others. Smart device users transmit multimedia files using apps. For multimedia file transfer under NAT (Network Address Translator), a lot of apps exploit a server with public IP or short range communication technologies such as WiFi Direct or Bluetooth because smart devices with private IP address cannot communicate each other directly. We first proposed the push service based file transmission scheme without an additional server and distance limitation. Second, for the push service based transmission, we proposed congestion control scheme to adapt transmission rate to the receiver's link bandwidth. Third, we considered bandwidth aware transmission for senders with different bandwidth. We implemented apps to evaluate the proposed scheme, and conducted experiments with smart devices.

■ keywords : Congestion Control ; Push Service ; Multi-source ; Smart Device ; Bandwidth Aware Transmission

## I. 서론

최근, 스마트 단말의 증가로 인해 사용자들은 사진, 음악, 비디오 등의 멀티미디어 콘텐츠를 더 자주 접할 수 있게 되었다. 많은 스마트 단말 사용자들은 다른 사용자들과 멀티미디어 파일을 공유하기 위해서 파일을 전송하거나 수신받기를 원한다. 하지만, WiFi, 3G, LTE 네트워크 등에 연결된 스마트 단말들은 NAT(Network Address Translator)를 통해 인터넷에 연결되기 때문에 사설 IP(Private IP)를 할

당받는다. 따라서 단말들 간의 직접적인 연결을 통한 파일 전송은 불가능하다. 이를 해결하기 위해서, 스마트 단말 상의 많은 어플리케이션들은 다음 두 가지 방법을 사용해서 NAT하의 단말 간 멀티미디어 콘텐츠 전송을 지원하고 있다. 첫 번째 방법은 공인 IP 주소를 할당한 서버를 중간 경유자로 활용하는 것이다. 스마트 단말이 서버에 연결해 공유하고자 하는 콘텐츠를 전송하고, 수신 단말들이 서버에 접속해 파일을 다운받는 방법이다. 두 번째 방법은 WiFi Direct 또는 블루투스(Bluetooth) 기술과 같은 단거리 통신 기술을 활용해 스마트 단말 간에 직접적으로 통신하는 것이다.

\* 정회원, 계명대학교 컴퓨터공학과  
본 연구는 2015년도 계명대학교 연구기금으로 이루어졌음

접수일자 : 2015년 09월 04일

수정일자 : 2015년 09월 25일

게재확정일 : 2015년 09월 29일

교신저자 : 안동혁 e-mail : donghyeokan@kmu.ac.kr

위의 두 가지 방법은 스마트 단말 간 파일 전송을 가능하게 하지만, 각각의 단점이 존재한다. 첫 번째 방법은 NAT를 통과하기 위해서 공인 IP를 할당할 서버가 필요하며, 이는 서버의 설치 및 운영비용이 수반된다. WiFi Direct나 Bluetooth와 같은 단거리 통신 기술은 서버가 필요하지 않지만, WiFi Direct와 Bluetooth의 전송 거리는 각각 약 100미터와 10-20미터이기 때문에, 통신 범위가 제한적이다[1, 2].

본 논문에서는 추가적인 서버의 운영과 전송 범위의 제한 없이 스마트 단말 간의 파일 전송을 지원하는 것을 첫 번째 목표로 한다. 이를 위해서, 푸시 서비스 기반의 파일 전송 기법을 제안한다. 애플, 구글, 블랙베리 등의 회사들은 푸시 서비스를 제공하기 위해서 푸시 서버를 운영하고 있으며, 푸시 서비스를 사용하는 스마트 단말들은 항상 푸시 서버와의 연결을 유지한다[3-5]. 이를 통해, 스마트 단말들은 푸시 서버를 통해 다른 단말에게 패킷을 전송하거나 패킷을 수신할 수 있다.

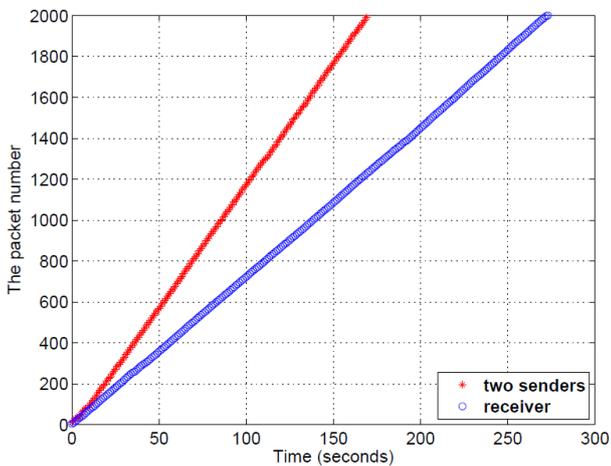


Fig. 1. The transmission using GCM  
 그림 1. GCM 상에서의 파일 전송

두 번째로 본 논문에서는 다중 송신자 환경에서 푸시 서비스 기반의 파일 전송을 위한 혼잡 제어(Congestion Control) 기법을 제안한다. 스마트 단말은 WiFi, 3G, LTE 네트워크 중 하나를 통해 푸시 서버와 연결한다. 이 때, 단말의 송신률과 수신율은 사용하는 각 네트워크에 따라 다르다. 여러 송신 단말들이 하나의 수신 단말에게 데이터를 전송할 때, 만약 단말들의 송신률의 합이 수신율보다 작다면, 혼잡(Congestion)은 발생하지 않는다. 하지만, 수신 단말의 네트워크의 성능이 낮을 경우, 네트워크에 혼잡이 발생한다. 본 논문에서는 네트워크의 혼잡을 확인하기 위해서, 안드로이드 단말과 구글 푸시 서비스인 GCM(Google Cloud Messaging)을 사용해 다음과 같은 실험을 수행하였다. WiFi에 연결된 두 안드로이드 단말이 3G에 연결된 안드로이드 단말에게 각각 1MB 데이터를 전송하였고, 그림 1은

실험 결과를 나타낸다. 시간이 경과됨에 따라 전송량에 비해 수신량의 차이가 증가하고 있으며, 이는 네트워크에서 전송중인 패킷의 수가 증가한다는 것을 의미한다. 그 결과 두 송신 단말들의 전송이 끝났을 때, 수신 단말은 전송된 전체 데이터의 62%만을 수신하였다. 이는 네트워크가 혼잡하다는 것을 뜻하며, 패킷 전송이 지연된다는 것을 의미한다. 따라서 본 논문에서는 수신 단말의 링크 대역폭에 맞추어 송신률을 조절하는 혼잡 제어 기법을 제안한다.

세 번째로, 여러 송신자들이 서로 대역폭이 다른 네트워크에 연결되어 있을 때, 각 네트워크 대역폭에 비해 전송량을 조절하는 전송 기법을 제안한다. 다중 송신자 환경에서는, 여러 송신자로부터 동시에 데이터를 수신 받는 것이 효율적이다. 송신 단말의 전송률은 해당 네트워크의 대역폭에 의해 결정되며, 대역폭이 서로 다른 경우 대역폭에 비해 전송량이 결정되어야 모든 송신자로부터의 전송이 거의 동시에 완료될 수 있다. 이를 위해 본 논문에서는 콘텐츠 중심 네트워크에서와 같이 수신자 중심의 전송 기법을 제안하였다[6].

논문의 구성은 다음과 같다. 2장에서는 TCP (Transmission Control Protocol)의 혼잡 제어 기법 및 관련 연구를 소개한다. 3장에서는 푸시 서비스 기반의 전송 기법 및 혼잡 제어 기법과 대역폭에 따른 전송량 조절 기법을 제안한다. 4장에서는 실험을 통해 성능을 평가하고, 5장에서는 결론을 기술한다.

## II. TCP 혼잡 제어 기법 및 관련 연구

본 장에서는 기존의 혼잡 제어 기법 중 대표적인 TCP 혼잡 제어 기법 및 관련 연구들을 소개한다. TCP에서는 혼잡으로 인해 대규모 패킷 손실이 발생하는 것을 방지하기 위해서, ACK 없이 전송할 수 있는 패킷의 수인 CW (Congestion Window) 값을 기반으로 송신률을 조절한다. 네트워크 대역폭을 최대한으로 사용하면서 네트워크의 혼잡을 회피하기 위해서 CW값은 AIMD (Additive Increase/Multiple Decrease) 방식을 사용해 계산되며, 네트워크 성능을 최대한으로 높이기 위해서 TCP는 ACK 패킷을 받을 때마다, CW값을 1/CW만큼 증가시킨다. 타임아웃이 발생하거나 세 번의 중복된 ACK를 수신할 경우, TCP는 네트워크가 혼잡하다고 판단하고 각 경우에 따라 다르게 동작한다. 송신자가 세 번의 중복된 ACK를 수신하면, TCP는 CW값을 절반으로 감소시키고 손실된 패킷들을 재전송한다. 타임아웃이 발생하면, TCP는 CW값을 1로 감소시키고 손실 패킷부터 재전송을 시작한다.

TCP와 관련된 연구는 예전부터 지속적으로 수행되고 있다. [7]에서는 위성 랜덤 액세스 채널 상에서의 혼잡 제어 기법을 제안하였으며, 트래픽 부하를 넘어서는 경우, 새로운 트래픽의 유입을 지연시켜 이미 네트워크에서 전송 중인 트

래픽의 처리를 우선시한다. [8]에서는 모바일 단말의 에너지 절감을 위한 TCP 프로토콜을 제안하고 있으며, 이를 위해 혼잡 제어 구간을 최대한 피해 데이터 전송 효율을 높이고자 한다. 본 연구에서는 푸시 서비스를 통신 채널로 사용하고 있으며, 이를 위한 혼잡 제어 기법 외에도 대역폭 기반 차등적인 전송량 할당 기법이 기존연구와 차별된다.

푸시 서비스에 관련된 기존 연구들은 크게 두 분야로 구분할 수 있다. 하나의 분야는 푸시 서비스들의 성능 및 사용성 분석이다[9]. [9]에서는 안드로이드를 위한 네 가지 종류의 푸시 서비스에 대해서 검증했으나, 혼잡 제어 또는 대역폭 기반 전송량 할당 기법과 같은 파일 전송과 관련된 연구는 다루지 않았다. 다른 분야는 푸시 서비스를 통신 채널로 활용한 연구들도 존재한다[10-13]. [10]에서는 게이트웨이를 위해 MQTT를 활용한 푸시 알림 프로토콜을 설계하였다. [11-13]에서는 MCM으로 명명된 미들웨어와 멀티미디어 프레임워크를 제안하였다. 기존 연구에서는 푸시 서비스를 스마트 단말과의 알림 통신용으로 사용하고 있을 뿐, 파일 전송용으로는 고려치 않고 있다는 점에서 본 연구와 차별된다.

### III. 제안한 전송 기법

본 논문에서 제안한 전송 기법은 다음과 같은 순서로 설명한다. 먼저, 스마트 단말 간 푸시 서비스 기반의 파일 전송 기법을 다루고, 수신 단말의 대역폭에 적응하는 혼잡 제어 기법을 제안한다. 마지막으로 서로 다른 대역폭을 가진 송신 단말간의 차별화된 전송량 결정 기법을 제안한다.

#### 1. 푸시 서비스 기반 파일 전송 기법

푸시 서비스는 하나의 앱 제공자(또는 스마트 단말)가 스마트 단말에게 알람을 전송할 수 있는 서비스이며, 본 연구에서는 파일 전송으로 그 사용을 확장하였다. 그림 2는 푸시 서비스 기반의 파일 전송을 나타내고 있다. 먼저, 서버에게 데이터 전송 또는 서버로부터 데이터 수신이 가능하도록 NAT 네트워크에 연결된 스마트 단말들이 푸시 서버와의 연결을 설정한다. 송신 단말은 수신 단말에게 데이터를 보내기 위해 푸시 서버에게 데이터를 전송하고, 푸시 서버는 수신 받은 데이터를 수신 단말에게 재전송한다. 스마트 단말들은 사실 IP를 할당받고 NAT를 통해 인터넷에 연결되기 때문에, 단말 간 직접 연결이 불가능하다. 하지만, 푸시 서비스 기반 전송 기법을 사용할 경우, 단말들은 서버의 설치와 거리의 제한이 없이 통신이 가능하다.

구글 푸시 서비스인 GCM의 경우, 안드로이드 단말이 푸시 서비스를 사용한다고 가정하면, 해당 단말은 GCM 서버와 연결을 설정하고 서버와의 연결을 유지해야 한다. 스마

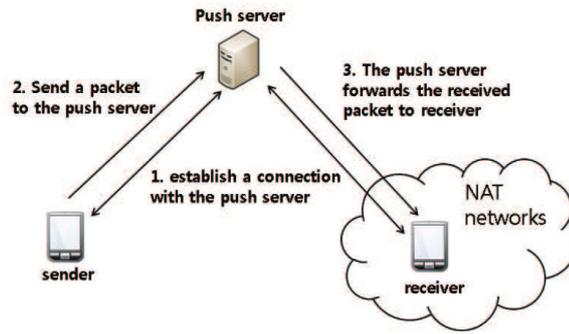


Fig. 2. The push service based transmission scheme

그림 2. 푸시 서비스 기반 전송 기법

트 단말들이 GCM을 사용해 패킷 전송을 하기 위해서는 수신 단말들의 등록 ID(Registration ID)를 알아야 한다[3]. 본 논문에서는 각 푸시 서비스를 사용하기 위해서 필요한 등록 ID와 같은 정보를 단말들이 이미 알고 있다고 가정한다.

#### 2. 혼잡 제어

다중 송신자 환경에서, 송신량이 수신량보다 클 경우 푸시 서비스 네트워크에 혼잡이 발생한다. 제한한 혼잡 제어 기법은 수신자의 대역폭을 기반으로 송신률을 조절하는 것이 주된 개념이다. 이를 위해서, 수신 단말은 인터벌 T동안 수신한 패킷의 수를 계산하고, 결과를 송신 단말에게 알린다. 송신 단말이 수신 단말로부터 (인터벌 T동안 수신한 패킷의 수를 포함하고 있는) 피드백을 수신하면, 수신 단말 네트워크의 대역폭에 맞춰 전송률을 조절한다. 송신 단말의 혼잡 제어 알고리즘은 다음과 같다.

```

 $T_i$  : i-th interval
 $NS_i$ : The number of packets sent during interval  $T_i$ 
 $NR_i$ : The number of packets received during interval  $T_i$ 
 $Threshold_{lower}$ : The lower threshold
 $Threshold_{upper}$ : The upper threshold

 $NS_1 \leftarrow \infty$  (infinite)
while file transmission finishes do
  if  $NS_i - NR_i < Threshold_{lower}$ 
     $NS_{i+1} < NS_i * \alpha$ 
  if  $Threshold_{lower} < NS_i - NR_i < Threshold_{upper}$ 
     $NS_{i+1} < NS_i$ 
  if  $NS_i - NR_i > Threshold_{upper}$ 
     $NS_{i+1} < NS_i * \beta$ 
end while

```

전송을 시작한 첫 번째 인터벌  $T_1$  동안, 송신 단말은 네트워크의 대역폭을 최대한으로 사용하기 위해서 패킷을 연속적으로 전송한다. 송신 단말이 수신 단말로부터 인터벌  $T_1$ 동안 수신한 패킷의 수를 전달받으면, 전송한 패킷의 수

와 수신 받은 패킷의 수의 차이를 계산한다. 만약 차이가 0에 근접한다는 것은 전송율과 수신율이 거의 동일하다는 것을 의미한다. 따라서 송신 단말은 현재의 전송률을 유지한다. 만약 차이가  $Threshold_{lower}$ 보다 낮다는 것은 송신률이 수신율보다 낮다는 것을 의미하므로 송신 단말은 송신률을  $\alpha$ 의 비율만큼 증가시킨다. 차이가  $Threshold_{upper}$ 보다 큰 경우는, 송신 단말이 전송하는 패킷들을 수신 단말이 전부 수신하지 못한다는 것을 의미한다. 따라서 송신 단말은 다음 인터벌 동안  $\beta$ 의 비율만큼 전송량을 줄임으로써 송신률을 낮추게 된다.

### 3. 대역폭 기반 차별화된 전송량 결정 기법

다중 송신자 환경에서 송신 단말들이 서로 다른 대역폭을 가진 네트워크에 연결되어 있을 경우, 각 송신 단말들의 총 전송량은 대역폭에 비해 차등적으로 할당되어야 한다. 만약 송신 단말이 동일한 전송량을 할당받을 경우, 가장 낮은 대역폭을 가진 송신 단말의 전송이 끝나야 전체 데이터 전송이 완료된다. 이 경우 전송의 효율성이 저하되는 단점이 발생한다. 따라서 낮은 대역폭에 연결된 송신 단말은 적은 전송량을 전송하고 높은 대역폭에 연결된 송신 단말에게는 많은 전송량을 할당함으로써 전송 효율성을 증가시킨다.

각 송신 단말들에게 차별화된 전송량을 할당하기 위해서, 데이터는 다수의 패킷으로 구성되어 있는 블락 단위로 전송된다. 각 블락들은 송신자가 아닌 수신자의 요청에 맞추어 전송된다. 즉, 하나의 블락 전송이 완료되면, 수신 단말은 송신 단말에게 특정 블락 전송을 요청한다. 송신 단말은 요청받은 블락의 전송을 시작한다. 수신 단말로부터 더 이상 블락 전송을 요청 받지 못한 송신 단말은 전송을 종료한다. 만약 대역폭이 큰 송신 단말의 경우, 데이터 블락 전송이 더 빨리 끝나기 때문에, 송신 단말은 수신 단말로부터 더 많은 블락 전송을 요청받는다. 그렇지 못할 경우, 송신 단말은 더 적은 블락 전송 요청을 받아, 상대적으로 적은 전송량이 할당된다. 예를 들면, 두 개의 송신 단말이 존재할 때, 하나는 빠른 송신자이고 다른 하나는 상대적으로 느린 전송 속도를 가지고 있다고 가정하자. 빠른 송신 단말의 전송 시간이 느린 송신 단말보다 짧기 때문에, 일정 시간동안 수신 단말은 빠른 단말에게 더 많은 블락 전송을 요청하고 더 많은 패킷들을 수신한다.

## IV. 성능 평가

본 논문에서는 실험을 통해 제안한 전송 기법의 성능을 평가하였다. 2015년 6월 기준으로, 안드로이드 운영체제의 점유율은 전세계에서 63.75%로 가장 많이 사용되고 있다 [14]. 따라서 본 연구에서는 안드로이드 단말과 안드로이드의 푸시 서비스인 GCM에 초점을 맞추어 실험을 수행하였다. 이를 위해, GCM을 사용하는 안드로이드 앱을 구현하였

다, 실험은 세 개의 안드로이드 단말을 사용해 진행되었고, 각각의 안드로이드 단말은 Galaxy Note (Android 4.1.2), Galaxy Nexus (Android 4.0.4), Galaxy S2 (Android 4.1.2)이다. 3G 네트워크는 KT 통신망을 사용하였다. 실험에서 전송한 패킷의 크기는 1KB이며, 전송 인터벌  $T_i$ 는 3초로 설정하였다.  $Threshold_{lower}$ 와  $Threshold_{upper}$ 는 각각 -10과 10으로 설정하였고,  $\alpha$ 와  $\beta$ 는 1.1 0.9로 설정 하였다. 먼저, WiFi에 연결된 두 개의 송신 단말이 3G에 연결된 수신 단말에게 데이터를 전송할 때 적용된 혼잡 제어 기법의 성능을 검증하였다. 이후, 송신 단말의 대역폭에 따른 차별화된 전송량 결정 기법의 성능을 검증하였다.

### 1. 혼잡 제어 기법의 성능 검증

제한한 혼잡 제어 기법만의 성능을 검증하기 위해서 두 개의 송신 단말은 동일한 대역폭을 가지도록 설정하였다. 이를 위해서, 두 송신 단말은 동일한 WiFi 네트워크를 사용하여 GCM 서버에 연결하였다. 수신 단말은 3G 네트워크를 통해 GCM 서버와 연결을 설정하였다. 두 송신 단말들은 총 2MB의 데이터를 수신 단말에게 전송하고, 결과는 그림 3에 나타내었다.

처음에는 두 송신 단말이 수신 단말의 대역폭을 최대한으로 사용하기 위해서 패킷을 최대한의 전송 속도로 전달한다. 따라서 수신 패킷의 수보다 송신 패킷의 수가 많아서, (송신 패킷 수 - 수신 패킷 수)의 값이 증가한다. 하지만 이 동안에서 송신 단말들은 전송 속도를 낮추려고 시도하고 있으며, 45초 이후부터 송신률과 수신율의 차이가 줄어든다. 이는 전송 속도가 수신 속도보다 낮다는 것을 의미한다. 95초부터 140초까지 송신률과 수신율이 비슷하게 유지되어 차이가 거의 없는 것을 확인할 수 있다. 그림 1에서는 송신률과 수신률이 일정해, 송신량과 수신량의 차이가 지속적으로 벌어졌으나, 본 연구에서 제안한 혼잡 제어 기법은 네트

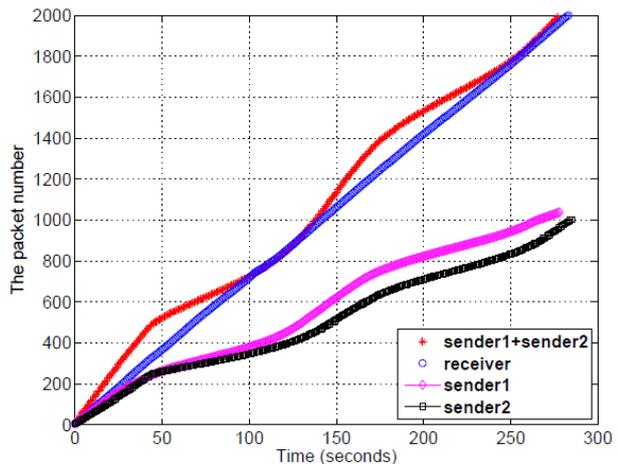


Fig. 3. The transmission with proposed congestion control  
 그림 3. 제안한 혼잡 제어 기법을 사용한 전송

워크 혼잡을 피하기 위해서 전송율을 조절한다는 것을 의미한다.

서로 다른 대역폭에 연결된 두 개 이상의 송신 단말에게 차별화된 전송량을 할당하기 위해서, 두 개의 송신 단말을 서로 다른 대역폭을 가지도록 실험 환경을 구축하였다. 하나의 송신 단말은 WiFi에 연결되었고, 다른 하나는 3G 네트워크에 연결되었다. 수신 단말은 이전 실험과 동일하게 3G 네트워크를 통해 GCM 서버에 연결되었다. 두 송신 단말은 총 10MB를 수신 단말에게 전송하였고, 그림 4와 그림 5는 각각 대역폭을 고려하지 않은 결과와 고려한 결과를 나타내고 있다.

그림 4는 대역폭의 고려 없이 두 송신 단말이 동일한 전송량을 전송했을 때의 결과이다. 즉, 실험에서 각 송신 단말은 5MB를 수신 단말에게 전송하였다. 3G 네트워크의 대역폭이 WiFi보다 낮기 때문에, 3G 네트워크를 사용한 송신 단말은 WiFi에 연결된 송신 단말보다 전송 속도가 느리다는 것을 확인할 수 있다.

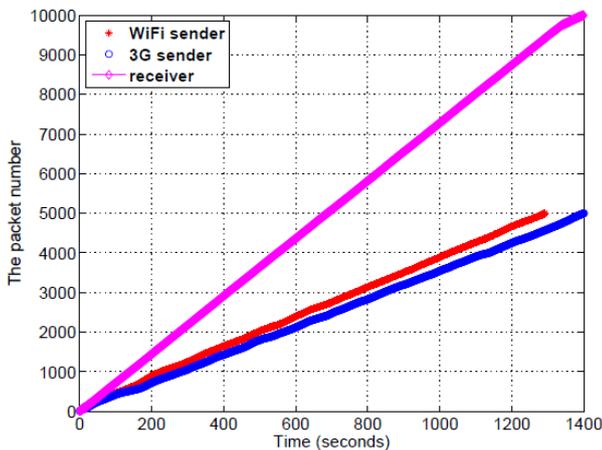


Fig. 4. The transmission without the bandwidth aware scheme

그림 4. 대역폭 고려 없는 전송

그림 5는 대역폭 기반 차별화된 전송량 결정 기법을 사용한 전송 결과이다. WiFi 네트워크의 전송 속도가 3G보다 빠르기 때문에, 수신 단말은 WiFi 네트워크를 사용하는 송신 단말에게 더 많은 데이터 블록을 요구한다. 따라서 WiFi 네트워크를 사용하는 송신 단말이 3G에 연결된 송신 단말보다 더 많은 패킷을 전송한다.

그림 4와 같이, 대역폭의 고려 없이 전송을 할 경우, 1290 초 이후 WiFi에 연결된 송신 단말은 전송을 완료한다. 이후 3G에 연결된 송신 단말만이 수신 단말에게 패킷을 전송하게 되며, 이로 인해 수신 단말의 네트워크 대역폭을 최대한으로 활용하지 못한다. 하지만, 대역폭을 고려해 송신 단말들에게 서로 다른 전송량을 할당하였을 경우, WiFi와 3G에

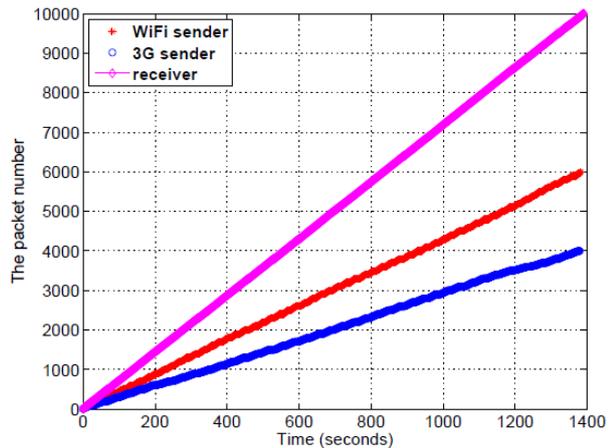


Fig. 5. The transmission with the bandwidth aware scheme  
그림 5. 대역폭 기반 차별화된 전송량 전송

연결된 두 송신 단말의 전송이 거의 동시에 완료되기 때문에, 수신 단말 대역폭의 효율적인 사용이 가능하다. 그 결과, 차별화된 전송량을 할당 기법을 사용한 전송이 그렇지 못한 전송보다 조금 더 빠르게 전송을 완료하였으며, 이는 송신 단말들 간의 네트워크 성능 차이가 더 클 경우, 두 기법 사이의 성능 차이가 더 커질 수 있다는 것을 예상할 수 있다.

## V. 결론

본 연구에서는 다중 송신자 환경에서 푸시 서비스 기반의 파일 전송을 위한 혼잡 제어 기법을 제안하였다. 본 연구에서 제안한 기법의 주된 장점은 다음과 같다. 첫째, 푸시 서비스의 사용을 알람용에서 파일 전송으로 기능을 확장하였고 이를 위한 전송 기법을 제안하였다. 이를 통해 사설 IP가 할당된 스마트 단말들 간의 통신을 위한 서버가 필요치 않으며, 단말 간 통신 범위의 제약도 존재하지 않는다. 둘째, 다중 송신자들의 전송률의 합이 수신 단말의 대역폭을 초과하지 않도록 혼잡 제어 기법을 제안하였다. 셋째, 서로 다른 네트워크를 사용하는 여러 송신 단말들에게 각 네트워크의 대역폭을 기반으로 차별화된 전송량을 할당하였고, 이를 통해 전송 속도를 향상시킬 수 있었다. 마지막으로, 본 연구에서는 안드로이드 단말 기반의 실험을 통해 성능을 평가함으로써, 모의실험보다 현실적인 성능 검증을 수행하였다.

## Acknowledgement

본 연구 2015년도 계명대학교 연구기금으로 이루어졌음.

## REFERENCES

- [1] Rossana Motta and Joseph Pasquale, "Wireless

- P2P: Problem or Opportunity?," in Proceedings of the Second International Conference on Advances in P2P Systems, pp. 32-37, Florence, Italy, Oct. 2010.
- [2] Jing Su, Kelvin K. W. Chan, Andrew G. Miklas, Kenneth Po, Ali Akhavan, Stefan Saroiu, Eyal de Lara, and Ashvin Goel, "A preliminary investigation of worm infections in a bluetooth environment," in Proceedings of the 4th ACM workshop on Recurring malware, pp. 9-16, Alexandria, USA, Oct. 2006.
- [3] Google Cloud Messaging for Android. <http://developer.android.com/guide/google/gcm/index.html>
- [4] Local and Push Notification Programming Guide. <http://developer.apple.com/library>
- [5] Push Service. [https://developer.blackberry.com/develop/platform\\_services/push\\_overview.html](https://developer.blackberry.com/develop/platform_services/push_overview.html)
- [6] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard, "Networking Named Content," in Proceedings of the 5th international conference on Emerging networking experiments and technologies, pp. 1-12, Rome, Italy, Dec. 2009.
- [7] H. Noh, Y. Lee, J. Lim, H. Park, and H. Lee, "Fast Congestion Control to Transmit Bursty Traffic Rapidly in Satellite Random Access Channel," J.KICS, vol. 39C, no. 11, pp. 1031-1041, Nov. 2014.
- [8] W. Choi, Ramneek, and W. Seok, "Yellow-Light TCP: Energy-Saving Protocol for Mobile Data Transmission," J.KICS, vol. 40, no. 03, pp.478-490, Mar. 2015.
- [9] Hansen, Jarle, Tor-Morten Grnli, and Gheorghita Ghinea, "Towards Cloud to Device Push Messaging on Android: Technologies, Possibilities and Challenges," in International Journal of Communications, Network and System Sciences, Vol. 5, no. 12, pp. 839-849, Dec. 2012.
- [10] Shinho Lee, Hyeonwoo Kim, and Hongtaek Ju, "Design of The High-Level Architecture of Mobile Integration SNS Gateway and The MQTT Based Push Notification Protocol," J. KICS, Vol. 38B, no. 05, May 2013.
- [11] C. Paniagua, S. N. Srirama and H. Flores, "Bakabs: Managing Load of Cloud-Based Web Applications from Mobiles," in Proceedings of the 13th International Conference on Information Integration and Web-Based Applications and Services, pp. 485-490, New York, USA, Dec. 2011.
- [12] Huber Flores, Satish Narayana Srirama, and Carlos Paniagua, "Towards mobile cloud applications: Offloading resource-intensive tasks to hybrid clouds," in International Journal of Pervasive Computing and Communications, Vol. 8, no. 4, pp. 344-367, 2012
- [13] Lee Dongcheul, "Designing the Multimedia Push Framework for Mobile Applications," in international Journal of Advanced Science and Technology, Vol. 32, pp. 117-124, July 2011.
- [14] StatCounter. <http://gs.statcounter.com>

---

 저 자 소 개
 

---



## 안동혁(정희원)

2006년 한동대학교 전산전자공학부 학사

2013년 한국과학기술원 전산학과 박사

현재 계명대학교 컴퓨터공학부 컴퓨터공학전공 조교수

&lt;주관심분야 : 무선 네트워크,

IoT, 미래 인터넷&gt;