

무선 P2P 네트워크에서 Peer 사이의 효율적 자료 분배 방법에 대한 연구

강승석¹⁾

Efficient Data Distribution Method in a Wireless P2P Network

Seung-Seok Kang

ABSTRACT

Popular content stored within the Internet may be convenient to download via modern 3G connections, but costly due to a fee charged by telecommunication providers on the basis of the amount of data transferred. This paper presents a network in which several mobile devices, called the peers, cooperate to download their assigned portion of the content using a 3G connection in order to reduce the overall cost. Then, the peers exchange their assigned portion of the content with other peers via a wireless ad hoc network and the peers reconstruct the whole content using the exchanged portion. This paper focuses on the per-peer based distribution method, and presents the performance comparison with a similar method, called the per-packet distribution method. Both methods are designed to reduce packet collisions while content is exchanged. The simulation result indicates that the per-peer based method outperforms the per-packet based method, and approximately 90% of the telecommunication cost is saved with as few as 10 peers.

Key words : Ad hoc network, Peer-to-peer network, Cost reduction, Parallel download, Content distribution

요약

인터넷 상에 저장된 콘텐츠를 3세대 통신 서비스를 이용해 내려 받으면 편리하지만, 자료의 양에 비례하여 통신료를 부과하는 경우 비용 면에서 부담스러울 수 있다. 본 논문은 peer라고 하는 이동단말기들이 협력하여 콘텐츠를 내려 받는데 지불하는 3G 통신비용을 줄이고자 콘텐츠의 지정된 일부분만을 유료인 3G 채널을 통해 내려 받고, 각 peer들은 자신이 내려 받은 부분을 다른 모든 peer들이 내려 받은 부분과 무료인 애드 hoc 네트워크를 통해 교환 후 각 부분을 재구성하여 전체 콘텐츠를 생성한다. 본 논문은 per-peer 방식의 배분 방법을 주로 소개하며, 본 방법을 이와 유사한 방법인 per-packet 배분 방법과 성능을 비교한다. 시뮬레이션 결과로 볼 때, per-peer 방법이 per-packet 방법보다 우수하며, 두 방법 모두 10개의 peer가 참여하였을 때 약 90% 정도의 통신비용을 절감할 수 있다.

주요어 : 애드 hoc 네트워크, P2P 네트워크, 저비용 내려 받기, 병행 내려 받기, 콘텐츠 분배

1. 서론

3세대 통신 서비스는 PDA나 무선기기 사용자들이 인

* 본 연구는 2006학년도 서울여자대학교 바롬학술연구비 지원에 의하여 수행되었음.

2006년 11월 28일 접수, 2006년 12월 9일 채택

¹⁾ 서울여자대학교 컴퓨터학부

주 저자 : 강승석

교신저자 : 강승석

E-mail; msukang@swu.ac.kr

터넷을 통해 멀티미디어 자료나 게임 등의 콘텐츠를 내려 받을 수 있는 서비스를 제공하고 있다. 그림 1은 가까이 있는 여러 무선기기 사용자들이 각자의 무선통신 서비스를 이용하여 인터넷에 접속하여 관심 있는 콘텐츠를 조회하는 상황을 나타낸다. 이런 상황은 운동경기장이나 미술관 등의 전시관, 학생들의 야외수업 등에서 발생할 수 있다. 그림에 있는 무선기기를 운동선수나 작품의 정보, 야외수업자료, 게임, MP3 파일 등 콘텐츠 제공업자가 서비스하는 똑같은 파일 중 하나의 파일을 모든 사용자가 자신의 단말기로 내려 받는 경우를 가정한다. 인터넷에

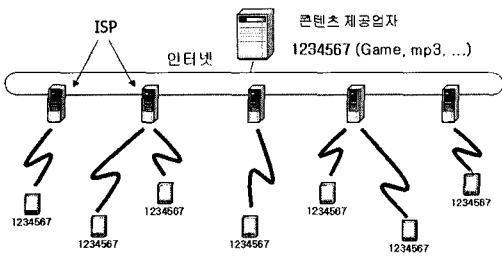


그림 1. 무선기기가 콘텐츠를 모두를 내려 받음

서 자료를 내려 받는 경우 무선기기의 통신비용은 내려 받는 자료의 양과 비례하기 때문에, 통신비용을 줄이기 위해서는 같은 콘텐츠를 내려 받는 무선기기는 각자 전체 콘텐츠의 일부만을 통신료가 부과되는 3G 채널을 통해 내려 받고, 비용이 부과되지 않는 애드 혹 채널을 통해 콘텐츠의 나머지 부분을 다른 무선기기들과 교환하면 전체 자료를 재구성할 수 있다. 본 논문은 무선기기들이 협동하여 인터넷에 저장된 하나의 콘텐츠에 대해 각 무선기기가 자신에게 할당된 콘텐츠의 일부분을 유료인 3G 통신채널을 통해 내려 받고, 나머지 부분은 참여한 다른 기기들과 애드 혹 네트워크를 구성하여 무료인 애드 혹 채널을 통해 서로 교환하는 방법을 소개한다. 이런 방법은 무선기기가 무선 WAN과, 802.11^[1]이나 Bluetooth^[2]와 같이 애드 혹 네트워크를 구성할 수 있는 무선 LAN 인터페이스를 모두 갖추고 있는 경우에 가능하다.^[3]

그림 2는 각 무선기기들이 내려 받고자 하는 콘텐츠 중에서 일부분만 내려 받는 상황을 나타낸다. 이런 상황은 여러 친구들이 하나의 모바일 게임을 서로 공유하여 같이 게임을 즐긴다거나 선생님과 학생들이 수업과 관련된 자료를 내려 받는 상황일 수 있다. 각 무선기기는 내려 받고자 하는 파일 중 내려 받을 부분을 지정받고 자신의 3G 채널을 이용해 인터넷을 접속한 후 지정된 부분을 내려 받는다. 그림 1과 그림 2의 차이는 각 무선기기가 3G 채널을 통해 내려 받는 자료의 양에 있다. 그림 1은 각 무선기기가 콘텐츠 제공업자(CP)로부터 전체 콘텐츠를 내려 받는 경우이다. 그러나 그림 2는 전체 콘텐츠 중 할당받은 부분만 내려 받는다. 이렇게 할당받은 전체 콘텐츠의 일부분만 내려 받게 되면 나머지 부분을 얻기 위해 참여한 다른 멤버 무선기기들과 애드 혹 네트워크를 구성하여 나머지 부분을 교환한다. 그림 3은 모든 무선기기들이 자신이 내려 받은 일부분의 내용을 참여한 다른 기기들과 교환하여 전체 파일을 재구성하는 상황을 보여준다.

본 논문에서는 각 무선기기가 3G 채널과 애드 혹 채널을 이용하여 다른 무선기기와 협력하여 저비용으로 인터

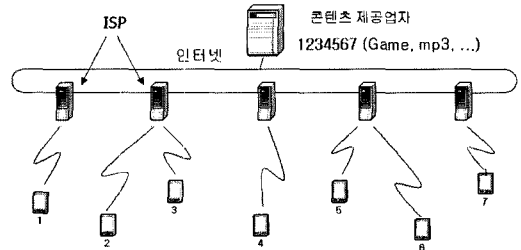


그림 2. 무선기기가 콘텐츠 일부만 내려 받음

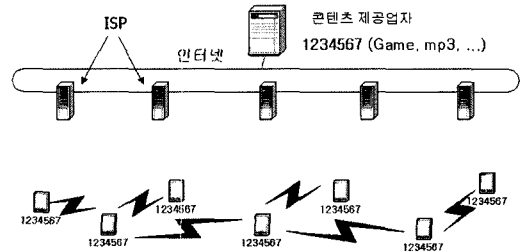


그림 3. 무선기기들이 내려 받은 일부의 콘텐츠를 애드 혹 네트워크를 통해 전체 콘텐츠 재구성

넷에 저장된 콘텐츠를 내려 받는 환경을 제안한다. 그중에서 하나의 무선기기가 자신이 내려 받은 일부분의 자료를 애드 혹 네트워크를 통해 다른 무선기기와 교환하는 효율적인 분배 방법을 제안한다. 이 경우 모든 무선기기가 송신자이면서 동시에 수신자가 되기 때문에 잘 정의된 분배방법이 필요하다. 이런 통제된 분배방법을 이용하지 않을 경우 많은 수의 패킷이 충돌하는 broadcast storm problem^[4]이 발생할 수 있다. 모든 무선기기는 애드 혹 라우팅 알고리즘을 사용하지 않으며, 대신 브로드캐스팅 방식을 이용하여 콘텐츠를 전송한다. 브로드캐스팅 방식을 이용하는 경우 복수개의 무선기기가 패킷을 전송할 때 발생할 수 있는 충돌을 방지하기 위해 한 시점에 하나의 무선기기가 복수개의 패킷을 이웃한 무선기기들에게 전달하며, 다음 순서의 무선기기가 이를 반복하면서 각자 내려 받은 자신의 일부 콘텐츠를 모든 이웃에게 전달한다.

본 논문은 peer-to-peer^[5] 개념을 활용한 all-sender-all-receiver 환경에서의 효율적인 콘텐츠 분배방식을 제안한다. 이와 비슷한 환경에서 하나의 무선기기가 전송한 패킷을 애드 혹 네트워크 전체에 전달하는 방식으로 전체 콘텐츠를 구성하는 분배방식을^[6]에서 제안하였다. 여러 송신자가 단일 수신자에게 전송 시 발생 가능한 공정성에 대한 문제를^[7]에서 다루었다. SplitStream^[8]은 송신 peer가 하나의 파일을 k 개로 나누어 k 개의 multicast tree를 이용하여 동시에 multicasting을 수행하는 방법을 제안하였다. BitTorrent^[9]는 여러 downloader가 URL로 지정된 송

신자로부터 자료를 내려 받는 경우, 각 downloader는 다른 downloader의 내려 받기 상태를 확인하여 자신이 내려 받은 일부 자료를 특정 downloader에게 upload하여 송신자의 부담을 덜어준다. BitTorrent는 하나의 파일을 고정크기(256 KByte)로 나누어 어떤 downloader가 어떤 파일 조각을 가지고 있는지 기록해둔다. 논문^[10]의 저자들은 혼잡한 경로를 피해 네트워크 자원의 사용률을 극대화하는 방식으로 자료를 수집하는 스케줄링 작성방법을 제안하였다.

본 논문의 구성은 다음과 같다. 2장에서는 각 무선기기들이 내려 받은 일부분의 콘텐츠를 참여한 다른 무선기기들과 교환하여 전체 콘텐츠를 구성하는 과정을 서술한다. 3장에서는 제안한 분배방법을 시뮬레이션을 통해 기존의 방식^[6]과 성능을 비교한다. 4장에서는 제안한 방법에 대한 결론을 맺는다.

2. 네트워크의 구성과 콘텐츠의 분배

2.1 네트워크의 구성

본 논문은 각 무선기기(peer 라 칭함)가 각자의 유료의 3G 채널을 이용해 전체 콘텐츠의 일부분을 내려 받아 무료인 애드 혹 채널을 이용하여 서로 내려 받은 콘텐츠를 교환하여 전체 콘텐츠를 재구성하는 방법을 서술한다. 이를 위해서는 먼저 애드 혹 네트워크를 구성해야 하는데 네트워크 구성에 대한 세부 사항은 논문^[6]에 기술되어 있다. 네트워크의 구성에 대해 간략히 요약하면 다음과 같다.

각 peer는 인터넷에 연결된 관리자 서버의 도움을 받는다. 관리자 서버는 peer의 ISP, 콘텐츠 서버, 혹은 인터넷의 어느 곳에 위치해도 상관없다. 각 peer는 반드시 하나의 관리자 서버와 관계를 맺어야 하지만, 관리자 서버는 여러 peer를 관리할 수 있으며 라이선스 문제나 콘텐츠의 공동구매에 따른 할인 등의 기능을 수행할 수 있다. 각 peer는 이웃한 peer에 대한 정보를 3G 채널을 통해 관리자 서버에게 전달한다. 여러 관리자 서버 중에서 가장 먼저 peer와 관계를 맺은 관리자 서버는 애드 혹 네트워크 관리를 위한 마스터 서버의 기능도 수행한다. 각 관리자 서버는 관계를 맺은 peer가 전달한 이웃 peer의 정보를 마스터 서버에게 전달하며, 마스터 서버는 이런 이웃정보들을 이용해 참여한 모든 peer에 대한 전체 애드 혹 네트워크 토폴로지를 구성한다. 또한 마스터 서버는 참여한 무선기기가 내려 받고자 하는 전체 콘텐츠의 어느 부분을 내려 받을지도 결정한다. 마스터 서버의 기능은 여러 관리자 서버들끼리 돌아가며 수행할 수도 있고, 하나의 서

버가 모든 peer와 관계를 맺으며 마스터 서버의 기능을 수행할 수도 있다.

2.2 콘텐츠의 분배

각 peer는 전체 콘텐츠 중 일부만 내려 받지만, 전체 콘텐츠가 필요하다. 전체 콘텐츠를 구성하기 위해 각 peer는 송신자이면서 동시에 수신자가 된다. 그러나 모든 송신자가 동시에 혹은 통제 없이 전송을 하게 되면 broadcast storm problem^[4] 문제가 발생하기 때문에 통제된 방식에 의해 효율적인 방법으로 각 peer가 내려 받은 일부분의 콘텐츠를 다른 peer들에게 전달하여야 한다.

기존에 제안한 per-packet^[6] 방식은, 참여한 peer들 중에서 재전송 집합을 선정하여, 한 peer가 패킷을 전송하면 그 송신자가 정의한 재전송 집합에 있는 peer들만 수신된 패킷을 forwarding 하는 방식으로 하나의 패킷을 전체 애드 혹 네트워크로 전달한다. 그림 4는 10개의 peer로 이루어진 애드 혹 네트워크를 보여준다. peer A가 하나의 패킷을 전송하면 peer B가 다음번에 전송을 시작한다. peer A의 재전송 집합은 peer B, C, D 이다. 이들이 A가 전송한 패킷을 받으면 그 패킷을 이웃 peer들에게 재전송하게 된다. 재전송 집합의 끝에 위치한 종단 peer는 관리자 서버가 선정하는데 종단 peer는 패킷 수신 후 하나의 패킷이 전체 네트워크에 전송이 완료되었다는 DONE packet을 전송한다. 지정된 다음 peer는 DONE packet을 받은 후 자신이 내려 받은 콘텐츠를 담은 하나의 패킷을 자신의 재전송 집합을 이용해 전체 애드 혹 네트워크의 모든 peer에게 전송한다. 그림 5는 per-packet

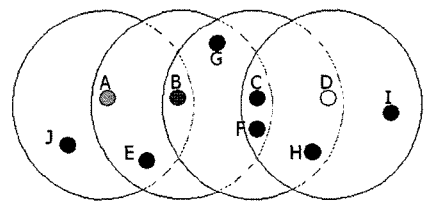


그림 4. 애드 혹 네트워크 구성의 한 가지 경우

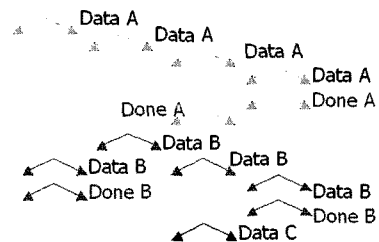


그림 5. per-packet 방식에서 Data와 DONE 패킷전달

방식으로 패킷을 전달하는 일련의 과정을 보여준다.

per-packet 방식처럼 하나의 데이터 패킷을 전체 애드 혹 네트워크에 전송 후 하나의 DONE packet을 전송하게 되면 한 번에 여러 데이터 패킷을 보내는 방식에 비해 전송률이 감소한다. 또한 애드 혹 네트워크의 중앙에 위치한 peer들은 더 많은 패킷 충돌을 경험하게 되어 재전송 횟수도 증가한다. 또한 빈번한 DONE packet의 전달은 전송완료시간이 늦어지는 원인이 된다.

본 논문에서 제안하는 per-peer 분배방식은 패킷 충돌 횟수를 줄이고 더 많은 데이터 패킷이 전달될 수 있도록 하기 때문에 per-packet 방식보다 전송완료시간이 빠르며, 3G 통신비용도 줄일 수 있으며 전일사용량도 감소한다. per-peer 방식에서는 어느 한 시점에 하나의 peer가 여러 개의 고정크기인 데이터 패킷을 자신의 이웃 peer들에게 전달한다. 전송한 peer는 데이터 패킷에 대한 재전송 peer 집합이나 종단 peer 등을 정의하지 않아도 된다. 즉, 이웃 peer들은 받은 패킷을 바로 재전송하지 않으므로 패킷의 충돌확률이 적어진다. 각 peer들은 독립적으로 어떤 패킷들을 전달할지 결정하며 일정수의 패킷을 전달하고 나면 DONE 패킷을 전송한다. DONE 패킷을 수신한 경우 이웃한 peer는 바로 재전송을 해야 하며 다음 전송예정 peer가 DONE 패킷을 받게 되면 자신의 데이터 패킷 전송을 시작한다. 그림 6은 그림 4의 상황을 이용하여 per-peer 방식의 패킷 전달 과정을 나타낸다.

마스터 서버는 각 관리자 서버로부터 얻은 peer들의 이웃 정보를 이용해 전체 애드 혹 네트워크의 토폴로지를 구성할 수 있다. 마스터 서버는 이런 애드 혹 네트워크 정보를 이용해 전송순서를 결정하여 환형리스트에 저장한다. 이 전송순서에 의해 각 peer는 몇 개의 데이터 패킷을 전송하고 다음 peer에게 전송기회를 넘겨준다. 환형리스트를 만들기 위한 규칙은 다음과 같다. 마스터 서버는 이웃의 수가 가장 많은 peer를 선정한다. 또한 이전에 선택한 peer와 직접 연결되어 있는 peer를 다음 전송 peer로

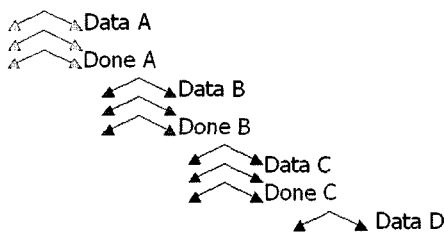


그림 6. per-peer 방식에서 Data와 DONE 패킷전달

선정한다. 선택한 peer의 모든 이웃 peer들이 리스트에 선택된 경우 마스터 서버는 현재 선택한 peer에서 가장 가까이 있는 peer를 선택한다. 그림 7은 간단한 애드 혹 네트워크 토폴로지를 나타낸다. 이 그림에서 마스터 서버는 먼저 가장 많은 이웃을 가진 peer B를 선택한다. 이후 E, A, D 순서로 선택이 가능하다. peer D는 선택되지 않은 이웃 peer가 없으므로 F를 다음 peer로 선택하고 마지막으로 C를 선택하여 환형리스트에 저장한다. 각 peer는 전송 순서상 자신의 앞에 있는 peer와 뒤에 있는 peer를 저장하여 자신의 순서를 판단한다.

2.3 neighbor 변수

내려 받고자 하는 파일의 크기를 알고 있고, 송신하는 데이터 패킷의 크기가 정해져 있다면(예를 들어 500byte) 몇 개의 패킷을 처리해야 하는지 결정할 수 있다. 만약 파일의 크기가 1MB 인 경우 2048개의 고정크기 패킷이 필요하다. 각 peer는 두 개의 변수를 둔다. 첫 번째 변수는 neighbor라는 정수 배열로 그 크기는 전체 데이터 패킷의 수와 같다. 두 번째는 같은 크기의 bitmap이 필요하다. 배열 neighbor의 한 값은 그 첨자에 해당하는 데이터 패킷을 이웃한 peer들이 가지고 있는지에 대한 정보를 나타낸다. 예를 들어 3G 채널을 통해 콘텐츠 일부를 내려 받는 경우 그 콘텐츠에 해당하는 데이터 패킷을 이웃한 peer들이 가지고 있지 않으므로 그 데이터 인덱스에 위치한 neighbor 값은 그 peer의 이웃한 peer들의 수와 같다. 만약 그 peer가 그 패킷을 전송하면 이웃한 모든 peer가 해당 패킷을 가지므로 그 패킷에 해당하는 neighbor 값은 0이 된다. 두 번째 변수인 bitmap은 각 peer가 특정 bit에 해당하는 데이터 패킷을 받은 경우 1로 설정하여 현재 어느 패킷을 소유하고 있는지 알 수 있다. 그림 6에서 B가 neighbor 값이 4인 패킷을 전송하면 A, C, E, F가 그 패킷을 수신하게 되고, 그 패킷에 해당하는 B의 neighbor 값은 0이 된다. peer A와 E가 유지하는 그 패킷의 neighbor 값은 1이 된다. 그 이유는 모든 peer가 2-hop에 해당하는 이웃의 정보를 가지고 있어서^[11-14], 이 정보를 이용하면 한 peer가 하나의 패킷을 받은 경우 같은 패킷을 또 어느

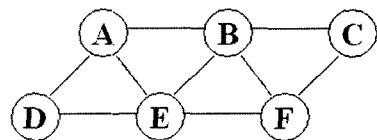


그림 7. 6개 peer로 이루어진 네트워크 토폴로지

peer가 받았는지 알 수 있기 때문이다. 즉 A와 E는 2-hop 이웃 정보를 이용해 D만 그 패킷을 가지고 있지 않다는 것을 계산할 수 있다. 만약 E가 그 패킷을 전송하게 되면 A의 해당 neighbor 값은 0이 되어 나중에라도 이 패킷을 전송할 필요가 없어진다.

per-peer 방식에서 전송 성능을 증가시키는 한 가지 방법으로 하나의 peer가 다수의 양수 neighbor 값을 가지는 경우 더 많은 데이터 전송 기회를 줄 수 있다. 데이터 분배 과정이 진행되면서 중앙에 위치한 peer는 가장자리에 위치한 peer보다 더 많은 데이터를 받게 된다. 만약 한 번에 하나의 패킷만 전송 가능하다면 DONE 패킷이 다수 발생되어 분배 완료시간이 길어질 수 있다. 각 peer가 한 번에 여러 패킷을 전송하는 경우 DONE 패킷 발생 빈도가 줄어들고 분배 완료시간도 단축된다. 그림 8과 그림 9는 각 peer가 한 번에 전송하는 패킷의 수에 따라 전송을 완료하는 시간의 차이를 나타낸다. 분배될 파일의 크기는

1MB이고 500byte 크기의 패킷을 사용한다. 자세한 모의 실험 환경은 3장에 기술되어있다. 그림 8은 모든 peer가 서로의 전송영역 안에 있는 경우이다. 한 peer가 한 번에 보낼 수 있는 패킷의 수가 증가하면 분배완료시간이 줄어든다. 그러나 한 번에 10개 이상의 패킷을 보내는 경우 완료 시간은 크게 변화되지 않는다. 한 가지 특이한 사항은 참여한 peer의 수와 상관없이 비슷한 완료시간이 소요된다. 그 이유는 전체 전송과정에서 볼 때 1-hop 범위 안에 모든 peer가 위치할 경우 전송될 데이터 패킷의 수는 참여한 peer의 수와 관계없이 동일하기 때문이다. 그러나 많은 peer가 참여할 경우 내려 받는 자료의 양이 적어지기 때문에 각 peer당 통신비용은 절감된다. 그림 9는 10개의 peer가 multi-hop에서 전송이 완료되는 시간을 나타낸다. hop 수가 증가할수록 완료시간은 증가한다. 모의 실험 결과에 따르면 각 peer가 한 번에 보내는 패킷의 수가 증가할수록 per-peer 방식의 전송성능은 증가한다.

2.3 콘텐츠 복구과정

per-peer 방식의 자료 분배 과정은 per-packet 방식과 달리, 데이터 패킷을 수신 후 즉각적인 재전송을 허용하지 않는다. 그러나 한 가지 예외는 DONE 패킷을 받은 경우이다. DONE 패킷은 다음 peer가 전송을 시작하도록 알려주는 시그널링 방법이다. DONE 패킷을 송신한 후 송신 peer는 자신이 전송한 패킷이 이웃 peer들에게 전송되었는지 확인하여야 한다. 송신 peer는 DONE 패킷 송신 후 이웃 peer가 데이터 패킷이나 DONE 패킷을 송신할 때 까지 기다린다. 만약 지정된 시간(예를 들어 100 msec) 동안 이웃 peer에서 응답이 없는 경우 송신 peer는 DONE 패킷을 다시 송신한다. 3회의 수신 후 아무런 응답이 없는 경우 해당 peer는 DONE 패킷을 재전송할 peer를 관리자 서버에게 요청한다.

모든 peer의 neighbor 값이 0이 되어 교환이 끝난 뒤에 일부 peer가 특정 위치의 콘텐츠가 없는 경우 복구 절차를 수행한다. per-peer 방식도 per-packet 방식과 마찬가지로 bitmap을 사용하지만, per-packet 방식은 bitmap 정보를 3G 채널을 이용해 관리자 서버에게 올려 보내지만 per-peer 방식은 애드 혹 채널을 이용해 이웃한 peer들에게 전송한다. 한 peer가 이웃 peer로부터 bitmap 정보를 받으면, 특정 bit가 0으로 되어있는지 확인한다. 만약 그 peer가 그 bit에 해당하는 데이터가 없는 경우에는 그 bit를 무시하며, 해당 데이터가 있는 경우에는 해당 위치의 neighbor 값을 수정하여 차후에 전송 순서가 되었을 때 해당 패킷을 전송할 수 있도록 한다.

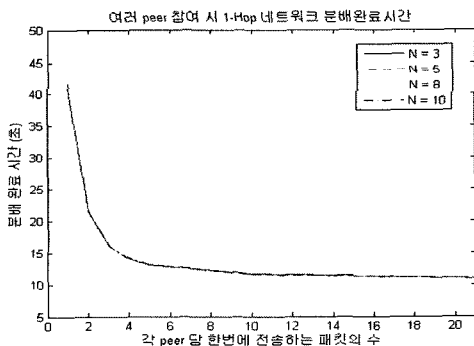


그림 8. 1-hop 네트워크에서 한 번에 전송 가능한 패킷의 수에 따른 전체 패킷 분배 완료시간

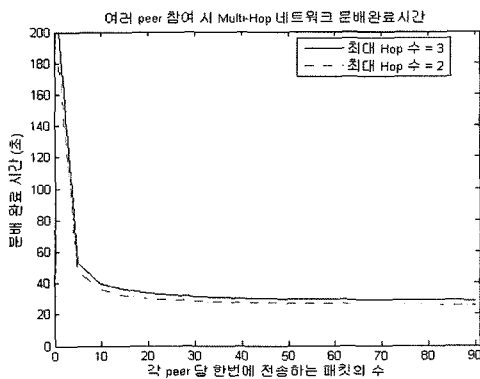


그림 9. Multi-hop 네트워크에서 한 번에 전송 가능한 패킷의 수에 따른 전체 패킷 분배 완료시간

만약 한 peer가 모든 콘텐츠를 저장하였고 bitmap의 모든 bit가 1로 설정되었다면, 그 peer의 순서가 되었을 때 BITMAP이란 패킷을 송신한다. BITMAP 패킷은 bitmap값 이외에 본 패킷을 생성한 peer의 네트워크 ID (주소)와 true인 boolean 값을 가진다. BITMAP 패킷은 DONE 패킷과 마찬가지로 지정된 다음 peer가 전송을 시작하도록 하는 기회를 준다. BITMAP 패킷을 받은 peer는 세 가지 중 하나의 선택이 가능하다. 첫 번째로, 그 peer의 neighbor 값 중 0이 아닌 양수 값이 있으면 BITMAP 패킷을 무시하고 해당하는 데이터 패킷을 전송한다. 두 번째 선택으로, neighbor 값은 모두 0이나 자신의 모든 bitmap이 0이 아닌 경우, 수신한 BITMAP 패킷에 있는 ID 값을 자신의 ID로 수정하고 boolean 값을 false로 바꾸며 자신의 bitmap을 기입한다. BITMAP의 boolean 값이 false인 경우 어떤 peer가 아직 모든 콘텐츠를 저장하지 못했다는 의미를 포함한다. 해당 peer는 수정한 BITMAP 패킷을 전송하여 이웃한 peer가 비어있는 콘텐츠를 채워주기를 기대한다. 세 번째 선택으로, 모든 neighbor 값이 0이고 bitmap이 모두 1인 경우, BITMAP 패킷을 수정 없이 재전송한다. 만약 처음 BITMAP 송신 peer가 자신의 ID와 boolean 값이 바뀌지 않은 BITMAP 패킷을 수신했다면, 애드 혹 네트워크의 모든 peer들은 전체 콘텐츠를 받았다고 판단할 수 있다.

3. 시뮬레이션 결과

per-packet 방식과 per-peer 방식의 성능을 비교하기 위해 ns2^[15] 시뮬레이터를 이용해 모의실험을 하였다. 본 시뮬레이션에서는 마스터 서버가 설정한 콘텐츠의 일부 분을 각 peer가 3G 채널을 통해 내려 받고, 애드 혹 네트워크를 이용해 각자 내려 받은 부분을 다른 peer들과 교환하는 방식으로 모의실험을 진행하였다. 각 peer는 다른 peer에게 패킷 전송 시 버퍼에 데이터가 없어서 전송하지 못하는 상황은 발생하지 않도록 하였다. 3G 채널의 패킷 크기는 500byte로 설정하였고, 애드 혹 네트워크에서도 같은 크기의 데이터 패킷을 사용하였다. 내려 받고자 하는 파일의 크기를 1M Byte로 설정하여 교환될 전체 패킷의 수는 2048개 이다. 각 peer는 802.11 MAC을 이용하고 전송 범위는 250미터이다. 참여한 peer의 수는 2에서 10이다. 이전의 논문^[6]에 의하면 전송 완료시간이 비교적 짧기 때문에 peer의 이동성은 고려하지 않았다. 모의실험은 정방형 크기의 400미터 영역에서 진행되었다. 각 peer의 패킷 버퍼 크기를 100으로 설정하였기 때문에, per-peer

방식의 경우 각 peer가 동시에 최대 90개의 패킷을 전송할 수 있다. 각 peer는 새로운 이웃에 대한 정보가 있으면 1초 이내에 upload한다. 애드 혹 네트워크에서 모든 peer는 항상 연결되어있다고 가정한다.

그림 10과 그림 11은 애드 혹 네트워크에서 hop의 수와 참여한 peer의 수에 따라 두 가지 분배 방법의 완료시간이 어떻게 달라지는지를 보여준다. 그림 10은 애드 혹 네트워크의 peer들이 일직선상에 200미터 간격을 두고 위치해있다. N개의 peer가 참여한 경우 한쪽 끝의 peer가 다른 쪽 끝의 peer에게 패킷을 전달할 경우 N-1 hop이 필요하다. 예를 들어 hop수가 4라고 하면 5개의 peer가 일직선상에 놓여있는 경우가 된다. 그림에서 보여주는 것처럼 per-peer 분배방식은 네트워크 내에서 hop수의 증가에 크게 영향을 받지 않음을 보여준다. per-packet 방식의 경우 hop수가 4 이상인 경우 전송완료 시간이 급하게 증가한다. 그 이유는 패킷교환 과정이 진행될수록 데이터 패킷과 DONE 패킷이 서로 충돌하기 때문이다. 이 충돌로 인해 DONE 패킷을 받지 못한 peer는 전송할 기회를 잃어버린다. per-packet 방식의 경우 패킷의 충돌이나 peer

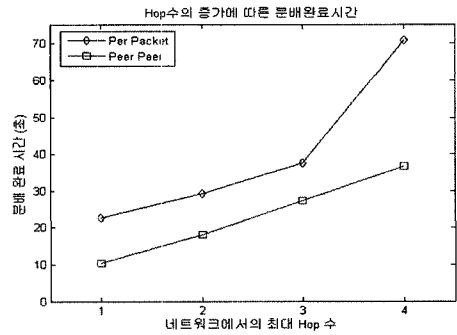


그림 10. Hop 수의 증가에 따른 분배완료시간

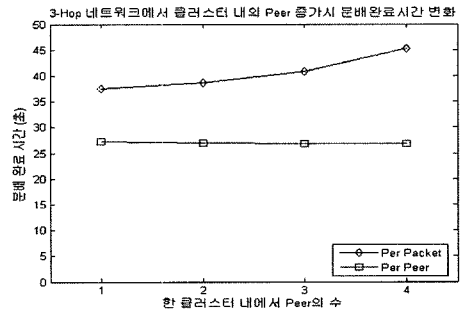


그림 11. 3-Hop 네트워크에서 한 클러스터 안의 Peer 수 증가에 따른 분배완료시간

의 이동에 의해 패킷을 잃어버릴 수 있다. 만약 일부 재전송 peer들이 다른 곳으로 이동한 경우, 패킷교환은 잠시 정지된다. 전송하는 peer는 패킷을 잃어버린 이유가 패킷의 충돌 때문인지 peer의 이동 때문인지 구별할 수 없다. 패킷 분실을 확인하게 되면 관리자 서버로부터 새로운 재전송 peer 정보를 받아 상태를 복구하여야 한다. per-peer 방식은 패킷의 충돌이 거의 발생하지 않는다.

그림 11은 4개의 클러스터를 가진 3-hop 네트워크에서의 분배완료 시간을 나타낸다. 각 클러스터에는 N개의 peer가 위치하며 각 클러스터에 있는 peer들은 충분히 가까이 위치한다. 이 경우 전체 애드 혹 네트워크에는 4*N개의 peer가 존재한다. 그림 10에서와 같이 각 클러스터는 일직선상에 200미터 간격을 두고 위치해있다. 그림 11은 각 클러스터 내에 한 개에서 네 개의 peer가 하나의 클러스터를 이루며, 네 개의 peer가 클러스터를 구성하는 경우 전체 16개의 peer가 애드 혹 네트워크를 이루게 된다. 그림 11에서 peer의 수가 1인 결과와 그림 10에서 hop수가 3인 것은 같은 결과를 나타낸다. per-peer 분배 방식의 경우 전체 네트워크의 peer수와 무관하게 일정한 시간에 콘텐츠의 분배가 완료됨을 알 수 있다. 반면 per-packet 방식에서는 참여한 peer의 수가 증가할수록 완료시간이 증가한다. 즉, per-packet 방식은 hop수에 영향을 받을 뿐만 아니라 클러스터 안에서의 peer수에도 영향을 받는다. 본 그림에서 per-peer 방식이 per-packet 방식에 비해 분배완료 시간이 우수하다.

그림 12는 참여한 peer의 수에 따라 통신회사에 지불해야 하는 유료 패킷 개수의 변화를 나타낸다. 그림에서 사용된 각 값은 30회 모의 실험을 통한 평균값이다. 참여한 peer의 수가 증가할수록, 콘텐츠를 내려 받는 데이터 패킷의 수가 감소하고 각 peer가 사용하는 전체 유료 패

킷의 수도 급격히 감소한다. 만약 여러 peer가 이미 애드 혹 네트워크에 참여하고 있는 경우, 새로운 peer의 참여로 인한 비용감소는 미비하지만, 새로운 peer는 상당한 비용의 감소혜택을 받는다. 10개의 peer가 참여하는 경우 대략 90% 정도의 비용감소 효과를 얻을 수 있다. per-packet 방식의 경우 per-peer 방식보다 좀 더 많은 유료 패킷을 사용한다. 그 이유는 per-packet 방식의 경우 수신되지 못한 패킷의 복구를 위해 각 peer는 bitmap 정보를 자신의 관리자 서버에게 올려주어야 한다. 그러나 per-peer 방식은 bitmap 정보를 무료인 애드 혹 채널을 통해 교환하여 수신하지 못한 패킷을 복구하게 됨으로 per-packet 방식보다 비용이 감소한다.

그림 13은 참여한 peer의 수에 따라 분배완료 시간의 변화를 나타낸다. 각 peer는 임의의 위치에 존재하며 모든 peer는 최소한 다른 하나의 peer와 연결되어 있다. 오른쪽 위로 증가하는 선은 30회의 모의실험을 통한 평균값을 나타내고 수직선은 표준편차를 나타낸다. 참여한 peer의 수가 증가할수록 분배완료 시간은 증가한다. 그 이유는 애드 혹 네트워크에서 peer간 hop수가 증가하기 때문이다. 참여한 peer의 수가 적은 경우 peer들은 가까운 곳에 모여 있고(모든 peer가 연결되어야 하므로) peer의 수가 증가할수록 넓은 공간으로 퍼지게 되며, 결과적으로 peer간 hop 수가 증가한다. 같은 이유로 peer의 수가 증가할수록 표준편차도 증가한다. 그림에 따르면 per-peer 분배 방식은 per-packet 방식보다 콘텐츠의 분배시간이 더 빠르다. 한 가지 특이한 점은 peer의 수가 증가할수록 두 방식의 분배완료 시간차이는 감소한다. 이 결과는 네트워크의 hop수라는 요소가 peer가 밀집해 있는 정도라는 요소보다 더 중요하다는 것을 의미한다. 그 이유는 per-peer 방식의 경우 패킷 충돌이 거의 발생하지 않기 때문이다.

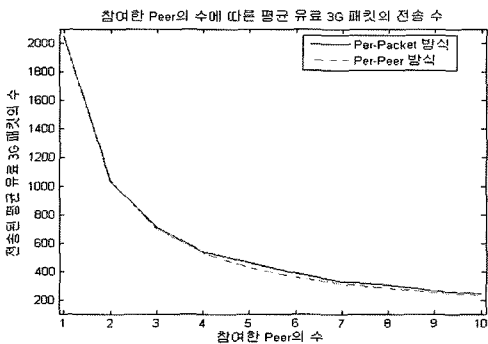


그림 12. 참여한 Peer에 대한 유료 3G 패킷 개수

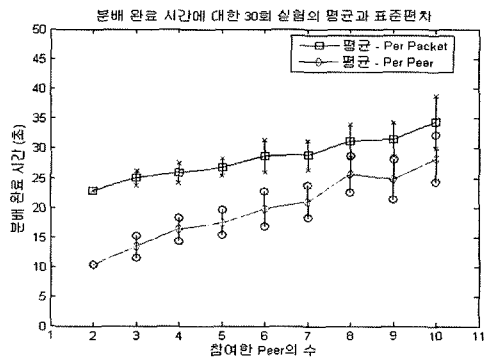


그림 13. 참여한 Peer의 수에 따른 분배완료시간

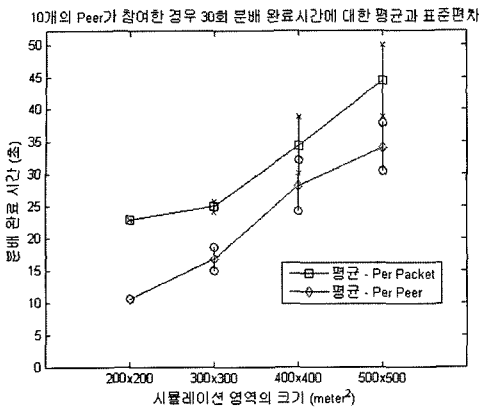


그림 14. 모의실험 영역에 따른 분배완료시간

평방 400미터 환경에서 작성된 270개의 네트워크 토폴로지 중에서 최대 hop 수는 3이다. 만약 모의실험 영역이 더 커서 최대 hop 수가 4 혹은 그 이상이면 그림 13의 경우보다 그 차이가 더 커지게 된다.

그림 14는 모의실험 영역에 따라 분배완료 시간이 변화되는 모습을 보여준다. 네 개의 모의실험 영역을 설정하였고 각 영역별로 30개의 네트워크 토폴로지를 생성하였다. 네트워크에는 10개의 peer가 참여하였다. 120개의 토폴로지 중에서 최대 hop수는 3이다. 각 값은 30회의 모의실험을 통한 평균값이고 수직선은 각 값의 평균에 대한 표준편차를 나타낸다. 모의실험 영역의 크기가 증가할수록 hop수도 증가하며, 더불어 분배 완료시간도 증가한다. 모든 모의실험 결과를 종합해볼 때 per-peer 분배방식은 per-packet 방식에 비해 분배완료 시간이 짧으며 적은 수의 유료 3G 패킷을 사용한다.

4. 결 론

본 논문은 애드 혹 네트워크에서 각 peer가 지정된 파일의 일부분을 3G 채널을 통해 내려 받고, 서로 다른 부분을 내려 받은 peer들이 애드 혹 채널을 이용하여 서로 교환함으로써 통신비용을 줄이는 방법을 제안하였다. 모든 peer는 인터넷에 위치한 관심 있는 콘텐츠의 일부를 자신의 3G 채널을 통해 내려 받기로 동의해야 한다. 각 peer는 하나의 관리자 서버와 관계를 맺으며 여러 관리자 서버 중에서 하나를 마스터 서버로 지정한다. 마스터 서버는 각 peer가 내려 받을 부분을 지정하고 자료 분배 순서 등을 정한다. 각 peer는 전체 콘텐츠 중에서 지정된 일부분을 유료인 3G 채널을 통해 내려 받고 그 부분을 다른

peer들과 무료인 애드 혹 채널을 이용해 다른 peer들과 공유하여 전체 콘텐츠를 재구성한다. 본 논문은 기존에 발표된 per-packet 방식을 소개하고 개선된 방식인 per-peer 방식을 소개하였다. 두 방법 모두 애드 혹 라우팅 알고리즘이나 멀티캐스팅 알고리즘을 사용하지 않으며 무선의 고유한 속성인 브로드캐스팅 기능만을 이용하였다. per-peer 방식은 2-hop 정보를 이용하여 neighbor 값에 따라 다수의 패킷을 이웃 peer에게 전달한다. 전체 콘텐츠를 구성 하는 과정에서 수신하지 못한 부분은 복구 과정을 통해 모든 peer가 전체 패킷을 받도록 한다. 여러 시뮬레이션 결과에 따르면 per-peer 방식이 per-packet 방식에 비해 분배 완료시간이 짧으며 적은 수의 유료 패킷을 사용하는 등 우수한 성능을 나타낸다. 또한 10개의 peer가 참여하는 경우 두 방법 모두 약 90% 정도의 통신 비용을 절감하였다.

참 고 문 헌

1. IEEE (1999), "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Standard 802.11.
2. Bhagwat, P. (2001), "Bluetooth: Technology for Short-Range Wireless Apps", IEEE Internet Computing, Vol. 5, No. 3, pp. 96-103.
3. Xiao, Y., Leung, K., Pan, Y., and Du, X. (2005), "Architecture, mobility management, and quality of service for integrated 3G and WLAN networks", Wireless Communications and Mobile Computing, John Wiley & Sons, Vol 5, No. 7, pp. 805-823.
4. Tseng, Y., Ni, S., Chen, Y., and Sheu, J., (2002), "The Broadcast Storm Problem in a Mobile Ad Hoc Network", Wireless Networks, Vol. 8, No. 2, pp. 153-167.
5. Oram, A., (2001), "Peer-to-Peer: Harnessing the Power of Disruptive Technologies", 1st Edition, O'Reilly and Associates.
6. Kang, S., and Mutka, M., (2004), "Efficient Mobile Access to Internet Data via a Wireless Peer-to-Peer Network", Proc. of the IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom), Orlando, Florida, pp. 197-205.
7. Karbhari, P., Zegura, E., and Ammar, M., (2003), "Multipoint-to-Point Session Fairness in the Internet", Proc. of InfoCom, San Francisco, California, pp. 207-217.
8. Castro, M., Druschel, P., Kermarrec, A., Nandi, A., Rowstrom, A., and Singh, A., (2003), "Split Stream: High-Bandwidth Multicast in Cooperative Environments",

- ACM Symposium on Operating Systems Principles (SOSP'03), pp. 298-313.
9. BitTorrent, The BitTorrent file distribution system, "<http://bittorrent.org>".
 10. Cheng, W., Chou, C., Golubchik, C., Khuller, S., and Wan, Y., (2003), "Large-scale Data Collection: a Coordinated Approach", Proc. of InfoCom, San Francisco, California, pp. 218-228.
 11. Lim, H., and Kim, C., (2000), "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks", ACM MSWiM 2000, pp. 61-68.
 12. Qayyum, A., Viennot, L., and Laouti, A., (2002), "Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks", Proc. of Hawaii International System Sciences (HICSS), pp. 3866-3875.
 13. Peng, W., and Lu, X., (2001), "AHBP: An Efficient Broadcast Protocol for Mobile ad hoc Networks", Journal of Science and Technology (JCST) - Beijing, China, Vol 16, No. 3, pp. 114-125.
 14. Calinescu, G., Mandoiu, I., Wan, P., and Zelikovsky, A., (2001), "Selecting Forwarding Neighbors in Wireless Ad Hoc Networks", ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications pp. 34-43.
 15. ns2, The network simulator, "<http://www.isi.edu/nsnam/ns>".



강 승 석 (msukang@swu.ac.kr)

1992년 고려대학교 이과대학 전산학과 학사
 1998년 Michigan State University 전산학 석사
 2004년 Michigan State University 전산학 박사
 2005년 수원대학교 컴퓨터학과 전임강사
 2006년 ~ 현재 서울여자대학교 컴퓨터학부 전임강사

관심분야 : ad hoc network, mobile computing, wireless communication, sensor network, QoS, anonymous communication, multimedia communication