韓國 컴퓨터情報學會 論文誌
第12卷 第2號, 2007. 5.

2007-12-2-2-6

애드 혹 네트워크에서의 효율적인 콘텐츠 공유 방법

강 승 석*

Efficient Content Sharing in Ad Hoc Networks

Seung-Seok Kang *

요 약

최근의 이동기기는 이동하면서 전화를 걸 수 있고, 무선으로 인터넷을 접속하고, 멀티미디어 영상과 음성을 출력하며, 주변의 유사한 이동기기들과 무선으로 통신을 하는 등 다양한 복합기능 때문에 실생활에서 자주 사용되고 있다. 3세대 이동통신 서비스는 이동기기에게 편리하게 인터넷을 접속하는 서비스를 제공한다. 이 경우 이동기기 사용자는 서비스 이용시 사용한 데이터의 양에 비례하여 통신요금을 지불한다. 본 논문은 peer라고 부르는 여러 이동기기들이 관심 있는 하나의 콘텐츠를 내려 받을 때, 비용절감을 위해 서로 협력하는 특별한 애드 혹 네트워크를 소개한다. 애드 혹 네트워크에서 각 peer들은 내려 받고자 하는 파일의 전체가 아닌 각자에게 할당된 부분만을 비용이 부과되는 자신의 3G 채널을 통해 내려 받고, 자신이 내려 받은 부분을 다른 peer 들과 비용이 지불되지 않는 애드 혹 채널을 이용하여 서로 교환하여 교환된 각 부분을 통합하여 전체 파일을 재구성한다. 모의실험 결과에 따르면, 참여한 peer의 수가 많을수록 비용 절감 효과는 커지며 10개의 peer가 참여한 경우 90% 정도의 통신비용 감소 효과가 있으며, peer의 참여가 증가하여도 전체 파일 재구성 완료시간은 완만하게 증가한다.

Abstract

Mobile devices become a part of our daily life due to their versatility, such as the wireless phone calls, the wireless accessibility to Internet, the display of multimedia content, and the communication with nearby mobile devices. Third generation telecommunication service provides an easy access to the Internet for mobile devices. Mobile users pay a fee charged by the telecommunication provider based on the amount of data transferred. This paper introduces a special ad hoc network in which mobile devices cooperate each other to download an interesting content from the Internet in order to reduce the telecommunication cost. The mobile devices, called the peers, in the ad hoc network are assigned a portion of the target file, and are responsible for downloading the portion using their 3G connection. Then, the peers exchange their downloaded portion with other participating peers using their cost-free ad hoc connection in order to reconstruct the whole content. According to the simulation results, large number of participating peers saves the telecommunication cost up to 90% with as few as 10 peers, although it slightly increase the overall content reconstruction time.

▸ Keyword : ad hoc network, content distribution, dual channel, parallel download

---

• 제1저자 : 강승석
• 접수일 : 2007.3.29, 심사일 : 2007.4.11, 심사완료일 : 2007. 5.11.
* 서울여자대학교 컴퓨터학부 전임강사
※ 이 논문은 2007학년도 서울여자대학교 바롬학술연구비의 지원을 받았음

Ⅰ. Introduction

Telecommunication services, such as 3G and HSDPA, are widely available for mobile devices to access the Internet anytime anyplace at the actual transmission rate of 2.05 Mbps to 4.8 Mbps [1].

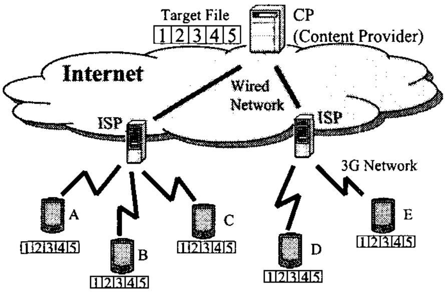


그림 1. CP로부터 전체 콘텐츠 내려 받기

Fig 1. Download entire content from CP

Figure 1 illustrates a common situation in which mobile devices contact their ISPs in order to download a specific target file from a content provider (CP) located in the Internet. As the figure 1 shows, some mobile devices, called the peers, may download the same content from the same CP. This is possible when a teacher wants to share the same class material stored in the Internet with his/her students in an outdoor class. In a stadium, several spectators may want to download current records of their favorite teams or players.

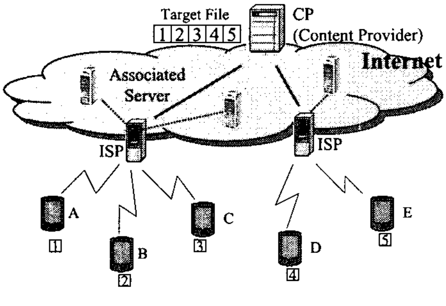


그림 2. CP로부터 일부 콘텐츠만 내려 받기

Fig 2. Download partial content from CP

Several researchers[2] focus on the devices with two network interfaces, one for telecommunication networks, such as 3G and HSDPA, and the other for WLAN, such as IEEE 802.11 or Bluetooth to form an *ad hoc* network. Integration of the two different networks offers several benefits both to users and service providers. Figure 2 displays how to reduce the telecommunication cost by the cooperation of several peers to share a favorite content stored by a CP in the Internet. Suppose several close friends wish to download a game program in order to play together interactively. Each device user is given a portion of the target file to download from the Internet with its 3G channel. Each device contacts to its ISP and access the assigned portion of the program provided by the CP. The difference between figure 1 and figure 2 is the amount of data downloaded by each peer. In figure 1, each peer downloads the *whole* content from the CP. In contrast, peers download an assigned *part* of the entire content in figure 2, which reduces the overall download cost of the content.

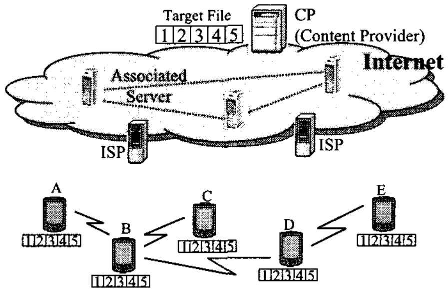


그림 3. 애드 혹 네트워크를 통해 자료 교환

Fig 3. Exchange partial content over an ad hoc network

Figure 3 shows that all peers cooperate to form an *ad hoc* network and exchange their partial content with other peers in order to reconstruct the whole content. Each peer associates with a server located in the Internet. An associated server may manage several peers, and occasionally all peers may associate with one server. The associated server may be located at the ISPs, the CPs, or some other places in the Internet. The associated servers communicate each other to share information about peers. Communication between servers takes place within the wired Internet. One of the associated server is selected as a master server that manages the *ad hoc* network. Each peer is responsible for informing any change of neighbor peers to its associated server. The associated server forwards the changed neighbor information to the master server. The master server builds a global topology of the *ad hoc* network using the neighborhood information from other associated peers, and decides which peer downloads which portion of the target file.

There are some assumptions of the mobile devices in this paper. First, the peers have WLAN interface that enable the formation of an *ad hoc* network. No specific *ad hoc* routing algorithm nor unique IP address is needed, but each peer has an unique ID within the *ad hoc* network. Second, the devices are equipped with an interface that enables a connection to an ISP. Third, all mobile devices share the same wireless communication channel. They have omni-directional antennas and the same transmission range.

The special *ad hoc* network concept exploits ideas that are often described as peer-to-peer computing [3]. The authors in [4] propose an approach to share a WWAN connection within an *ad hoc* network with participating hosts. The authors in [5] investigated the performance improvements of the multiple description (MD) Content Delivery Network (CDN) over the single description (SD) CDN. Fairness issues of the multiple sources to single receiver (multipoint-to-point) is studied in [6]. The authors in [7] proposed an approach for computing a schedule for coordinated data collection with avoiding congested links and maximizing the network resource utilization. SplitStream [14] allows a source peer to split a file into k stripes and to multicast each stripe using *k* multicast trees. SplitStream builds *k* multicast forests in order to spread the forwarding load across all participating peers. In BitTorrent [15], when many downloaders try to download a file from a URL-specified location, the downloaders upload to each other concurrently to help reducing the load of the source. BitTorrent partitions files into pieces of fixed size, typically 256 KBytes, and keeps track of which downloaders have what pieces. BlueTorrent [16] inherits the idea from BitTorrent. In BlueTorrent, the data source, for example an access point or peers, divides a target file into fixed-size blocks. Whenever Bluetooth-enabled devices have available connections, they exchange their bitmaps in order to specify which blocks of the file is missing. Cellular phones become active participants in content sharing networks. In [17], a cell phone plays a role of a content storage for music, photos and videos, and connects to federating surround infrastructure, via access hosts using WiFi or Bluetooth, in order to share the content with larger screen or speakers. The authors in [18] implement mobile content sharing service in GPRS/UMTS network using C++. It consists of a peer-to-peer client application in the mobile phone and an application server in the wired network.

Many research work for content sharing focuses on several issues such as the performance improvement of content sharing by using multiple sources as well as content discovery and maintenance. This paper mainly focuses on the benefits of using two wireless channels, such as the cost reduction of payable 3G packets as well as fast distribution of the downloaded content. Several papers do not deal with binding the two channels in order to reduce telecommunication cost. They focuses on the service continuation of the connection when users are indoors or outdoors. This unique scheme provides for each peer to receive large telecommunication cost reduction as more peers participate at the cost of slight increment of the distribution time.

The rest of the paper is organized as follows. Section 2 deals with the operations of the *ad hoc* network. Content distribution scheduling is explained in section 3. Simulation results are described in section 4. Section 5 draws conclusions.

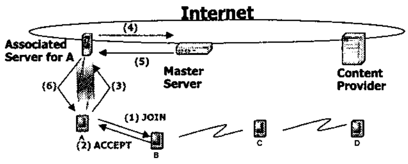
II. Network management

2.1 Construction of an ad hoc network

One of the mobile device users initiates the process of the special network formation. The device user makes a connection to its ISP to access the Internet and contacts one of the servers on the Internet. The device gives information such as the minimum number of participating mobile devices and the file specification to be downloaded. The server becomes the associated server of the device. The server generates a group ID (GID) of the *ad hoc* network and group key so that only the network members can exchange packets. The server also creates a network ID (NID) used in the *ad hoc* network to distinguish between several other *ad hoc* networks by the server, and a unique peer ID (PID) for its associated device. The server passes the group information, the NID, and the PID to its associated device through the 3G connection of the device. The device stores the information and periodically transmits a beacon. The beacon contains the description of the *ad hoc* network including the transport level information of the server, the NID, and the PID of the beacon sender.

The server of the initiating mobile device plays two roles in the *ad hoc* network: the master server and the associated server. The master server deals with several management tasks for the *ad hoc* network. First, it creates and maintains group information and assigns new PID when new device joins the network. Second, it manages the membership and the logical topology of the *ad hoc* network. Third, it schedules the downloading process which device to download which portion of the target file. Finally, it detects and recovers from several failures. The associated server contacts its associated mobile peer and collects local neighborhood information from its peer. In addition, the associated server selects a set of rebroadcasting peers for its associated peer when the peer broadcasts downloaded data to other peers.

The remaining cooperative devices hear the beacon and send JOIN packets to the sender of the beacon. The sequence of joining an *ad hoc* network is displayed in figure 4. The JOIN packet includes the PID of the beacon sender and a random number that indicates the sender of the JOIN packet. Any member peer of the *ad hoc* network receiving the JOIN packet compares the PID of the beacon sender in the packet with its own, and drops if unmatched. If both PIDs are identical, the peer replies with an ACCEPT packet. The ACCEPT packet contains the transport information of the master server, the NID, and the received random number from the JOIN packet.



(1) New device sends JOIN packet to existing ad hoc network
(2) ACCEPT packet contains info about the master server
(3) Peer delivers master server info to its associated server
(4) Associated server registers its peer to the master server
(5) Master server passes IDs to associated server
(6) IDs are passed to associated peer to join network

그림 4 새로운 peer의 합류
Fig 4. New peer's joining

When the device transmitting the JOIN packet receives an ACCEPT packet with the same random number, the device becomes a peer of the network. The peer informs its associated server about the NID and transport level information of the master server. The associated server of the peer registers the presence of the new peer to the master server by showing the NID. As a response, the master server provides information about the *ad hoc* network specified by the NID to the associated server including the GID and the PID of the joining peer that is unique within the *ad hoc* network. The PID will be used as a network address for the newly joined peer in the *ad hoc* network. The associated server delivers the *ad hoc* network information to its associated peer. The peer configures the PID as a network address in the *ad hoc* network, and stores the group information.

2.2 Membership management

When a peer joins the *ad hoc* network, the master server updates the membership information. In addition, both the master server and the associated server require neighborhood information to perform their tasks. Each peer detects its one-hop neighbors by exchanging periodic HELLO packets using the *ad hoc* network or by operating in promiscuous mode. If there is a change, the peer uploads the newly configured neighbor list. The associated server delivers the change to the master server. The master server updates the change of the membership and the neighborhood information.

The master server is able to create a global topology of the *ad hoc* network from the partial neighborhood information. When the master server receives the information, it updates the global topology of the network. The global topology is essential for both the master server and associated servers to compute the download schedule and the minimum rebroadcasting set. The master server maintains a *boolean connectivity matrix* in which an element of *(i, j)* is set to true if the i^{th} peer is a neighbor of the j^{th} peer. The matrix represents the global topology of the *ad hoc* network. The master server computes the download schedule with the membership information. The master server also distributes the topology information to all associated servers so that they can compute the rebroadcasting set of peers.

2.3. Download scheduling

The master server schedules each peer to download the same or similar amounts of the target file. Suppose N is the size of the target file to download, and D is the size of assigned portion downloaded by each peer in each round. The value D is equal to $k * u$, where u is a payload of each packet downloaded at a time from the CP and k is an integer. The server assumes $N = D * t$, where t is an integer. If the given file size N is not a multiple of D, the server adjusts the value N by adding an amount by the following equation.

$$N' = N + (\lceil N/D \rceil * D - N)$$

The value N' becomes a multiple of D and is set to be a new size of the target file. In each round, the total of $D * p$ bytes are downloaded by all peers, where p is the current number of peers. The master server executes the following algorithm to assign the beginning location of the target file for each peer.

```
for (i = 0; i < p; i++)
  { download-peer (i, base+i * D); }
base = base + D * p;
```

Each peer is indexed from zero to *p-1* by the master server and the peer with index *i* should download a portion of the target file starting at the second argument of the function *download-peer()* with the size of D bytes. The master server notifies all associated servers about the location where to download for their peers. The master server keeps a mapping table that identifies which peer has downloaded which portion of the file. The value *base* is initialized to zero and increased at the end of each round.

In the middle of the downloading process, some mobile devices may want to join the existing network. In this case, the master server needs to decide whether to accept the new devices. The master server computes the minimum value of n that satisfies the following inequality

$$D * p_r + D * n * (r + 1) \leq N'' \ldots\ldots\ldots\ldots\ldots (1)$$

where r is the current round of download completed, p_r is the number of participating peers in round r, $n = p_{(r+1)} - p_r$, a number of newly joining peers, and $N'' = N' - base$, the remaining bytes to download. That is, all existing peers will download D bytes in the $(r+1)^{th}$ round, and the newly joined n peers will download $(r+1) * D$ bytes. To be fair, all participating peers (including new peers) download the same or similar amounts of data at the end of downloading process. If the minimum value n is equal to zero, no more new devices will be accepted. Otherwise, the first n peers registered are accepted as peers to participate. At the end of each round, the master server checks whether there exists a positive integer n that satisfies equation (1). If there is no such value n, there is no need to add more mobile devices. The master server directs all associated servers that their peers should stop transmitting beacons.

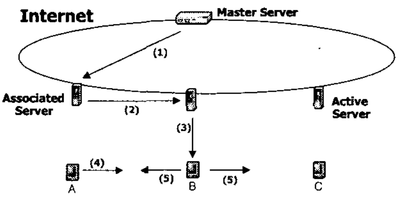
At the final round, the remaining bytes may be smaller than $D * p$. The master server finds the smallest index *rem* such that $D * rem$ becomes larger than the remaining bytes to download. Then, the master server assigns the remaining portions of the file to the first *rem* peers, or randomly selects *rem* number of peers. After the completion of the final round, the complete target file will be stored at the peers in the *ad hoc* network.

III. Content Distribution Scheduling

In order to reconstruct the target file, every peer in the network becomes a sender as well as a receiver. If all peers, as senders, transmit their data in an uncontrolled manner, the broadcast storm problem [8] may arise. Each peer needs a controlled way of broadcasting its received data to other peers, and selected peers need to rebroadcast the packets if some peers are out of transmission range of another.

3.1. Selection of rebroadcasting peers

The problem of selecting the minimum rebroadcasting set is very similar to the MCDS (Minimum Connected Dominating Set) problem. The authors in [9] mentioned that the minimum rebroadcasting set problem is harder than the MCDS problem, which has been proved to be NP-complete. Several wireless broadcasting algorithms [9][10][11][12] have been proposed for the approximation of MCDS to compute the backbone wireless peers or the rebroadcasting peers using local neighborhood information. Starting from the source peer, two-hop knowledge helps to decide the next rebroadcasting peers in a completely distributed manner.



(1) Master server passes global topology of *ad hoc* network
(2) Associated server passes rebroadcasting set for A
(3) Involved associated servers notify to its peer to rebroadcast
(4) Source peer transmits its downloaded data to its neighbors
(5) Peer(s) in the set rebroadcasts packets originated from A

그림 5. rebroadcasting peer의 선정
Fig 5. Selection of rebroadcasting peers

The local neighborhood information from each peer provides a hint to generate a mesh of connected peers. The master server merges the connected peers into one connected set of peers. The master server passes the information to all associated servers. Each associated server produces the minimum set of rebroadcasting peers for its associated peer working as a source peer. As the source peer changes, so does the rebroadcasting set. Each associated server notifies the involvement of the rebroadcasting to other associated servers of the peers in the set. Each responsible associated server directs its peer(s) to rebroadcast packets when it receives data packets originated from the specified peer. Figure 5 shows the computation and the notification sequence of the rebroadcasting set. In figure 5, the associated server for peer A selects peer B as its rebroadcasting set. When peer B receives packets that originated from peer A, it rebroadcasts the packets for peers, such as C, that are located far from peer A.

3.2. Data distribution

Figure 6 shows a typical example of an *ad hoc* network topology. Seven peers, labeled A to G, form an *ad hoc* network. They have downloaded some portion of data from the Internet. The master server defines the transmission sequence of peers to distribute their downloaded data. Figure 6 assumes the transmission sequence as an alphabetical order of peers from A to G. Every peer knows which peer is its predecessor that triggers the next transmission. As an example, when peer A finishes its data delivery, peer B resumes its transmission. Every peer in the set has a list of rebroadcasting information that describes when to rebroadcast if received packets have originated from a specific peer. For example, the associated server of peer A selects peer B, C, and D as A′s rebroadcasting set in figure 6.

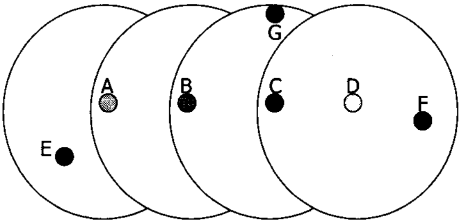


그림 6. 애드 혹 네트워크의 예
Fig 6. Example of an ad hoc network

In a similar way, peer B′s rebroadcasting set will be A, C and D. Peer C has rebroadcasting information that directs to rebroadcast if it receives packets originated from peer A or B. Peer D also has additional knowledge that peer D is the *terminal peer* of the rebroadcasting set. That is, when peer D rebroadcasts, there are no more peers to rebroadcast the same packet from peer D. The terminal peers play an important role to trigger the next sequence from the source peer to transmit its downloaded portion of data.

The *ad hoc* network adopts an asynchronous transmission mechanism in which a packet from the predecessor triggers the transmission of the successor′s packets. In figure 6, peer A is the first peer to transmit its data in the sequence. When peer A broadcasts its data, peer B, C, and D, in that order rebroadcast the packet originated from peer A, which enables all peers to have the content downloaded by peer A. Peer B needs to know when to broadcast its downloaded data. Suppose peer B rebroadcasts its data right after receiving data from its predecessor peer (peer A in this case). This may cause collisions in some peers and data lost because peer B may transmit its downloaded data while peer C may rebroadcast A′s packet, which causes peer G to experience a collision. In order to reduce the possibility of collisions, peer B should delay its transmission until peer A′s packet is delivered to all peers.

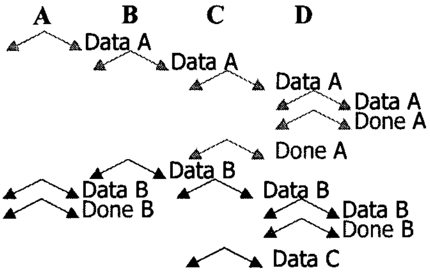


그림 7. 패킷의 전송 순서
Fig 7. Packet transmission sequence

The terminal peers are useful to decrease the probability of collisions. When peer A is a source, peer D becomes a terminal peer. In contrast, if peer B is a source, then both peer A and D become terminal peers. When a terminal peer rebroadcasts a data packet, it immediately broadcasts a DONE packet that notices the completion of rebroadcasting in one end of the data flow. The DONE packet propagates backward to the source peer until the next peer in the transmission sequence receives the DONE packet. The packet triggers the broadcast of B′s downloaded data. Figure 7 shows the snapshot of data propagation from peer A, B, C and D in the network displayed in figure 6.

3.3. Recovery from missing data

Every peer maintains a receiving status bitmap of size t such as $N' = t * u$ where N' and u are defined in section 2.3. As the size of the target file is known, so is the size of the bitmap. Each bit indicates whether the mapped content has been received either from the 3G connection or from the *ad hoc* network. A bit is set if the corresponding unit-sized payload is received.

When the final round of downloading is completed and all packet exchanges within the *ad hoc* connection have finished, each peer uploads its bitmap to its associated server. When one or more peers have failed or left the network, some part of the target file will be missing. This requires additional downloading rounds for the missing portion of the file.

The master server collects the bitmaps from all peers and applies bitwise-OR operation on all bitmaps. If the resulting bitmap contains at least one bit that remains reset, it indicates that no peer downloaded that portion of the file. It is because one or more peers left or failed during the download phase. The master server counts the number of reset bits, and records the position of the missing portion of the file. Then, the server decides which peer(s) to download the missing portion. The extra downloading schedule is delivered to the associated server of the peers involved. The responsible peers resume downloading the assigned portion of data in order to fill the missing gap.

When the complete content of the target file is in the *ad hoc* network, the remaining task is for all peers to share the complete content. The master server maintains the bitmap matrix of the size p by t, where p is the number of peers and t is the total number of payload units. If the element (i, j) in the bitmap matrix has a value of zero, it implies that the i^{th} peer does not have the j^{th} unit segment of the file. The master server scans each column of the matrix whether the column includes any bit reset. If it exists in the j^{th} column, the master server selects the source peer and a rebroadcasting peer set for the j^{th} unit segment. When peers receive the missing segment of data, they update their bitmaps. After completing the scan of all columns of the bitmap matrix and processing any existing gaps, all peers upload their bitmaps to the master server. The master server finishes the missing data recovery process when it finds that all bitmaps contain all ones, which indicates all peers have stored the target file.

3.4. Network Partition Recovery

It is possible for an *ad hoc* network to be partitioned into two or more subnetworks because some intermediate peer(s) may move to another place(s). The boolean connectivity matrix, discussed in section 2.2, is useful when the master server detects any partition of the *ad hoc* network. The server applies a transitive closure operation, e.g. Warshall's algorithm, on the matrix and discovers any partition, if exists. The resultant matrix has all one's if the *ad hoc* network is connected. If there is any partition in the network, each row of the matrix indicates which group of peers is connected and disconnected. The master server may notify the disconnectivity to the associated servers of the peers in the partitioned network in order to construct another *ad hoc* network on their own.

The associated servers in the partitioned network elect a new master server among the associated servers. The newly elected master server generates new NID and GID, then distributes them to other associated servers whose peers are in the partitioned network. The peers in the partitioned network may resume their download operations with the newly computed download schedule by the elected master server. The master server of the original network deletes the peer information of the partitioned network from both the membership and the connectivity information. In addition, the original master server also recomputes the download schedules with the remaining member peers. The associated servers recompute the rebroadcasting sets for their peers with the updated peer information from the master server.

IV. Simulation results

The *ns2* network simulator [13] is used in this simulation. The model assumes each peer downloads portions using its 3G connection that the master server schedules to download, while it exchanges the portions with other peers using its *ad hoc* network. The peers do not experience buffer underrun when they transmit their portions to other peers.

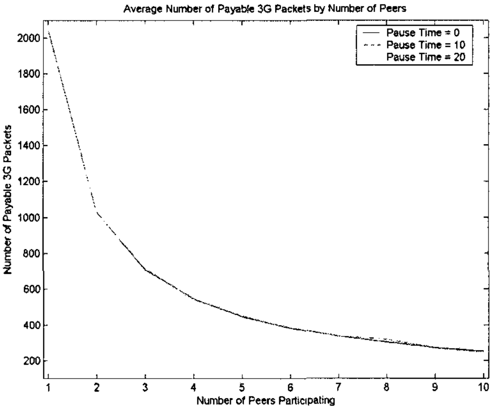


그림 8. 참여한 peer에 따른 비용 절감효과
Fig 8. Cost reduction by the number of participating peers

At the beginning of simulation, each peer is assigned a portion to download, and computes the number of 3G packets used. The unit packet size of the 3G connection is set to 500 byte. Each peer transmits its downloaded portion of packets over the ad hoc network. The *ad hoc* network uses the same size of data packet of 500 byte. The peers download the file size of 1 MBytes that consists of 2048 unit packets. Each peer has the 802.11 MAC with the transmission range of 250 meters. The number of peers varies from 2 to 10. The peers move at 2 m/s according to the way-point mobility model in a 400 meter by 400 meter grid. Each peer uploads its new neighborhood information in a second using 3G channel.

Figure 8 displays the average number of packets used by the peers that may have a fee charged a telecommunication provider. Each value in figure 8 is the average of ten runs. The average number of payable 3G packets is collected by the addition of the number of content packets downloaded from the Internet and the number of bitmap packets uploaded to the associated servers during the exchange period over the ad hoc network. As the number of peers increases, the average number of fee-based packets decreases. It is mainly because the amount of downloaded data becomes smaller as the number of participating peers increases. As a result, an additional peer only reduces a marginal cost when there are already enough peers, whereas the new peer still receives the same benefit of the cost reduction. The pause time does not influence the results due to the short download completion time.

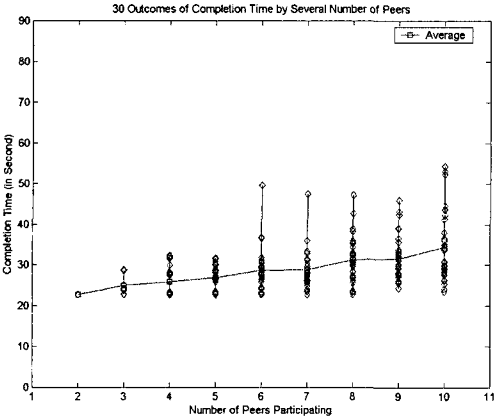


그림 9. Peer 수에 따른 전송완료 시간
Fig 9. Number of peers vs. completion time

The completion time of each simulation is shown in figure 9. This simulation uses the pause time of zero seconds. The average completion time increases slowly as more peers participate. It is because when the area becomes crowded, the chance of packet collision increases. The *ad hoc* network operates with the packet triggering mechanism in which the current packet triggers the next packet transmission. The rebroadcasting peer or the scheduled next source peer may lose its chance to transmit and waits for receiving the lost packet from its previous transmitting peer.

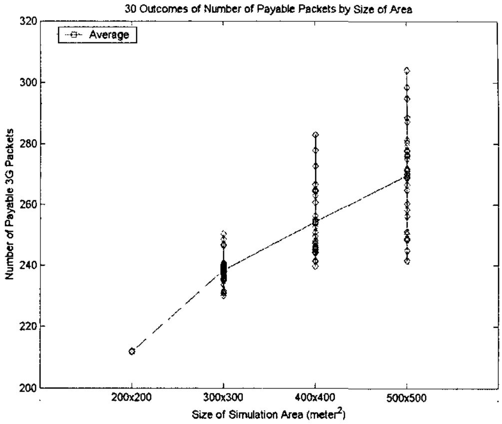


그림 10. 모의실험 영역 크기와 패킷의 수
Fig 10. Simulation area vs. number of payable 3G packets

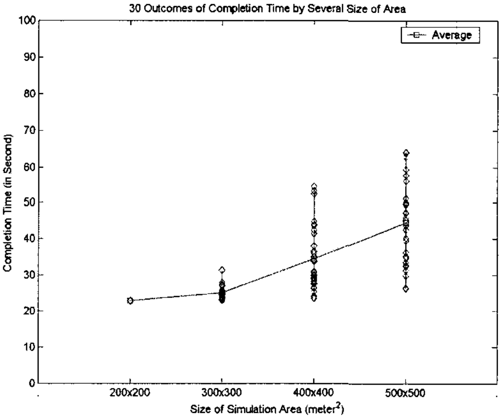


그림 11. 모의실험 영역 크기와 전송완료시간
Fig 11. Simulation area vs. completion time

The *ad hoc* network is greatly influenced by the area size. When all peers are located within the transmission range of each other, no collisions are expected because only one peer transmits at a time. Figure 10 and 11 illustrates how the performance changes due to a change in the grid size. The simulation runs with 10 peers in constant movement. As the grid size increases, so does the completion time and the number of fee-based packets due to the fact that when peers are located in wider area, there needs to be more rebroadcasting peers. Some packets may collide and some peer lose the packets, which deprives the chance to rebroadcast the packet or to transmit the next sequenced data packet.

V. Conclusion

This paper introduces a special *ad hoc* network in which mobile peers have two wireless channels which helps to save telecommunication cost by sharing their partially downloaded data with other peers. All peers consent to download a predefined portion of a target file located in the Internet using their fee-based WWAN connection. The peers are associated with their servers, which perform peer membership management, download scheduling, data distribution scheduling, and failure recovery. Each participating peer downloads only a portion of the file that is assigned by the master server. The peers then distribute their downloaded portions to all other member peers over the cost-free *ad hoc* channel so that all participating peers can construct the target file. The distribution process does not use any specific routing or multicasting protocol, but the inherent wireless broadcast mechanism with the rebroadcasting set information enables all peers to share their partially downloaded data with other peers. As the telecommunication cost is proportional to the amount of data downloaded, when larger number of peers participates, peers download smaller amount of data which leads to the reduction of the telecommunication cost. The benefit of cost reduction is distributed to all participating peers. The simulation results show that approximately 90% of the telecommunication cost is saved with as few as 10 peers, although it slightly increase the overall content reconstruction time.

참고문헌

〔1〕 I. Kovacs, K. Pedersen, J. Wigard, F. Frederiksen, T. Kolding, "HSDPA Performance in Mixed Outdoor- Indoor Micro Cell Scenario", IEEE International Symposium on Personal, Indoor and Mobile Radio Communications(PIMRC), pp. 1 - 5, 2006

〔2〕 Y. Xiao, K. Leung, Y Pan, X. Du, "Architecture, mobility management, and quality of service for integrated 3G and WLAN networks", Wireless Communications and Mobile Computing vol 5 pp.805-823, 2005

〔3〕 A. Oram, "Peer-to-Peer : Harnessing the Power of Disruptive Technologies", O'Reilly & Associates, March, 2001.

〔4〕 M. Papadopouli and H. Schulzrinne, "Connection Sharing in an Ad Hoc Wireless Network among Collaborating Hosts," NOSSDAV 1999, June 1999.

〔5〕 J. Apostolopoulos, T. Wong,W. Tan, and S. Wee, "On Multiple Description Streaming with Content Delivery Networks," in Proceedings of INFOCOM 2002, pp. 1736 - 1745, June 2002.

〔6〕 P. Karbhari, E. Zegura, and M. Ammar, "Multipoint-to- Point Session Fairness in the Internet," in Proceedings of INFOCOM 2003, pp. 207 - 217, March 2003.

〔7〕 W. Cheng, C. Chou, L. Golubchik, S. Khuller, and Y. Wan, "Large-scale Data Collection: a Coordinated Approach," in Proceedings of INFOCOM 2003, pp. 218 - 228, March 2003.

〔8〕 Y. Tseng, S. Ni, Y. Chen, and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," Wireless Networks, vol. 8, pp. 153 - 167, March-May 2002.

〔9〕 H. Lim and C. Kim, "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks," ACM MSWiM 2000, pp. 61 - 68, August 2000.

〔10〕 A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks," Proceedings of Hawaii International System Sciences (HICSS) 2002, pp. 3866 - 3875, Jan 2002.

[11] W. Peng and X. Lu, "AHBP: An Efficient Broadcast Protocol for Mobile Ad Hoc Networks," Journal of Science and Technology (JCST) - Beijing, China, vol. 16, pp. 114 - 125, March 2001.

[12] G. Calinescu, I. Mandoiu, P. Wan, and A. Zelikovsky, "Selecting Forwarding Neighbors in Wireless Ad Hoc Networks," ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 34 - 43, July 2001.

[13] ns2, The network simulator. http://www.isi.edu/nsnam/ns.

[14] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Split-Stream: High-Bandwidth Multicast in Cooperative Environments," ACM Symposium on Operating Systems Principles, pp. 298-313, October 2003.

[15] BitTorrent, The BitTorrent file distribution system. http://bitconjurer.org/BitTorrent/

[16] S. Jung, U. Lee, A. Chang, D. Cho, M. Gerla, "BuleTorrent: Cooperative Content Sharing for Bluetooth Users", IEEE Int'l Conference on Pervasive Computing and Communications, pp. 47-56, March, 2007

[17] T. Pering, D. Nguyen, J. Light, R. Want, "Face-to- face media sharing using Wireless Mobile Devices", IEEE Int'l Symposium on Multimedia, Dec, 2005

[18] M. Matuszewski, N, Beijar, J. Lehtinen, "Mobile Peer- to-Peer Content Sharing Application", IEEE Consumer Communications and Networking Conference, pp 1324-1325, Jan 2006

저 자 소 개

강 승 석

2004년 8월 Michigan State University
Computer Science. Ph.D.
2006 ~ 현재 :
서울여자대학교
정보미디어대학 컴퓨터학부
전임강사