Anonymous Content Sharing in Ad Hoc Networks *

Seung-Seok Kang, Matt W. Mutka and Li Xiao

Department of Computer Science and Engineering

Michigan State University

East Lansing, MI 48824, USA

{sskang, mutka, lxiao}@cse.msu.edu

Abstract

It may be costly for mobile pervasive computing device users to download content from the Internet using their 3G connections if the 3G connection cost is a function of the amount of data downloaded. This paper introduces an approach in which mobile pervasive computing devices form an ad hoc network and share downloaded content with each other. In order to improve privacy when sharing content, this paper describes an anonymous connection between the sending peer and the receiving peer. Simulation results show that the transmission overhead of the anonymous connection may increase 50% or less as the number of peers increase or the peers are scattered over the larger area.

1 Introduction

The number of Internet users with mobile pervasive computing devices has increased. Wireless interfaces on mobile pervasive computing devices may be in the form of third generation (3G) telecommunication interfaces or wireless local area network (WLAN) interfaces, such as 802.11. Third generation networks plan to provide service for both voice and data with the transmission speed of up to 2.05 Mbps [1] with wide area coverage. However, it is expected that the 3G connection cost to download data from the Internet is a function of the amount of data downloaded. Other research work has proposed to reduce the 3G telecommunication cost by sharing the downloaded content with others [2, 3] or by partially downloading and combining the content [4]. The research activity assumes that each mobile device has both a 3G interface for WWAN access and a WLAN interface such as 802.11 or Bluetooth, to form an ad hoc network. In addition, it is expected that all participating mobile devices cooperate with each other to share content

or to reconstruct the target content by exchanging partially downloaded segments.

This paper focuses on sharing content within an ad hoc network in order to reduce downloading cost by using an anonymous connection. The cooperating mobile devices, called the peers, form an ad hoc network in order to share the content. We call the network a Cooperating ad Hoc network to sUpport Messaging (CHUM) [2, 3, 4, 5, 6]. Each peer associates with a CHUM server, which may be located at the ISPs, CPs, or some other place in the Internet. A server may be associated with several peers. For some cases, all peers may contact a single server that controls the operation of the CHUM network. Each associated server collects neighborhood information and sharable file information from its associated peer. One of the associated servers plays a role of the master server. The master server gathers all partial neighborhood information and creates a logical network that mirrors the global view of the ad hoc network. In addition, the master server maintains file status information that represents files stored by the participating peers. A peer is able to discover its needed content stored in the CHUM network with the aid of its associated server.

As both the content source and receiving peers are selected, a path between the two peers can be constructed with or without anonymity. An anonymous connection makes it difficult for other peers to determine which peers are communicating, such as done within Anonymizer [7], Onion (Tor) [8], Crowds [9], mCrowds [10], Tarzan [11], P^5 [12], APFS [13], and Shortcut-Responding [14]. When users do not want to expose their identities when sending/receiving content among members of the ad hoc network, the network should support mutual anonymity for the participating peers. The associated server of the requesting peer is responsible to construct the anonymous path. The master server needs to select source peers when multiple peers hold the requested content.

This paper describes the functionality of the CHUM network and anonymous communications. The network operation and management are in section 2. Simulation results

*This research is supported in part by NSF Grant No. 0334035 and NIH Grant No. EB002238-01.

- (1) Peer Req wants a file (segment) F
- (2) Associated server of Req, AR, requests F to master server
- (3) Master server confirms that peer Src has the file F
- (4) AR requests a neighbor of Src to AS, associated server of Src
- (5) AS selects NS peer and sends PID of NS to AR
- (6) AR selects NR peer, creates CIDS, and sends CIDS to AS
- (7) AS passes CIDS to construct an anonymous connection

Figure 1: Overall sequence of anonymous connection setup

are explained in section 3. Section 4 draws conclusion.

2 CHUM network operation

The CHUM network in this paper provides privacy between the source and requesting peers. The requesting peer does not know who is the actual source peer, and vice versa. Moreover, it is difficult for any neighboring peers in the CHUM network to perform traffic analysis. The purpose of traffic analysis is to help deduce which peer is talking to whom by analyzing traffic patterns instead of the content that is transmitted. Although data is encrypted, traffic analysis may detect the two peers communicating. Privacy requires an anonymous connection between the source and the requesting peer. The master server or a third party may become a trusted entity to distribute the secret key set to peers for data encryption/decryption. The associated server of the requesting peer performs the detailed anonymous connection setup. Only the associated server of the requesting peer knows about the actual source and the requesting peer. No other associated servers nor the master server know both peers. Furthermore, no peers can guess correctly who is talking to whom, including the source and the requesting peer themselves. Two intermediate peers are involved in the anonymous connection between the source and the requesting peer.

2.1 Anonymous connection setup

The anonymous connection between the source and requesting peers requires participation of intermediate peers. Fig. 1 shows the overall sequence of the anonymous connection setup. The associated server of the requesting peer,

Dest. Address	Crypt for Padding	Crypt for Forward Data	Crypt for Backward Data
NS	Sym Key P1	Sym Key E1	Null
NR	Sym Key P2	Sym Key E2	Sym Key E1
Req	Sym Key P3	Sym Key E3	Sym Key E2
Null	Sym Key P4	Null	Sym Key E3
Padding			

Figure 2: CIDS example of Fig 1

called the AR server, selects one peer from the neighbor list of the requesting peer, called the NR peer. The AR server, then, requests the associated server of the source peer, called the AS server, to pick one peer from the neighbor list of the source peer, called the NS peer. Communication between the AR server and the AS server is encrypted to hide the connection information. The anonymous data path starts from the source peer to the NS peer, to the NR peer, and finally to the requesting peer. The route from the source peer to the requesting peer is known except for the path between the NS and the NR peer. An ad hoc routing protocol, such as AODV, is used to route messages between the two intermediate peers.

The AR server knows the source and the requesting peers. It has the responsibility to construct the anonymous connection. No other servers, including the master server or any peers, know completely the source and requesting peers. The AR server creates a connection information data structure (CIDS), which is similar to a collection of onions in onion routing [8]. CIDS is a four-layered data structure. Each layer has the same format and the same size. One layer contains a destination peer address and three keys for its bi-directional virtual connection depicted in Fig. 2. Each layer, except the innermost or the fourth layer from the top, represents one of the three virtual connections made by the four peers: the source, the NS, the NR, and the requesting peer. From the outermost to the innermost, each layer is used by each of the four peers in that order.

The CIDS is delivered from the AR server to the source peer via the AS server. Since the layer is encrypted with the peer's private key, when one of the four peers receives the CIDS, it applies its private key to the fixed-sized top layer to read the virtual connection information, such as a destination address and keys. Since the peer knows the next peer and the keys, it is able to deliver the CIDS to its next peer. The anonymous connection information for the next peer is hidden inside the padding of the CIDS. Before sending the CIDS, the peer removes the top layer of the CIDS and applies the first key on the remaining layers (padding) of the

CIDS. The new CIDS will be transmitted to the next destination peer in order to build a bi-directional virtual connection. For example, when the source peer receives a CIDS, it applies its private key on the fixed-sized top layer to acquire the address of the next virtual hop peer (the NS peer) and three keys. The second key will be used to encrypt data transmitted to the NS peer. The third key is used for backward transmission, but it is useless for the source peer. The source peer removes the top layer and applies the first key on the remaining part of the CIDS. The source peer could not recognize the PID of the NR peer after decrypting the CIDS because the decrypted part is also encrypted with the public key of the NS peer. The source peer sends the decrypted CIDS to the NS peer so that they construct a virtual socket connection between the source and the NS peers. The NS peer performs a similar task and creates a virtual connection with the NR peer. The backward key of the NS peer is the same as the forward key of the source peer. When the requesting peer receives the CIDS from the NR peer, it finds that the peer itself is the final destination because the destination address is set to a null value. As a result, a series of actions generates a bidirectional anonymous virtual connection between the source and the requesting peer with the aid of both the NS and the NR peer.

The AR server, initially, should create the innermost layer of the CIDS first. Then, it decides a size of padding right after the innermost layer. The innermost layer will be encrypted using the public key of the requesting peer and the padding will be encrypted with the key P4 in Fig. 2. The AR server, then, creates the second innermost layer for the NR peer. This layer will be encrypted using the public key of the NR peer and the remaining part (the layer for the requesting peer plus the padding) will be encrypted with the key P3. This process repeats until the outermost layer for the source peer is encrypted by the public key of the source peer and the remaining part (the three inner layers plus the padding) is done with the symmetric key P1.

2.2 File Distribution

As the anonymous connection is established, the source peer receives an acknowledgement from the requesting peer over the backward connection. The source peer, then, starts transmitting the file (segment) to its next virtual hop peer, the NS peer. The route discovery may not be required between the source and the NS peer because they are within one hop. If one of the two peers move away and the connection is broken, the source peer should recover the path by using the path recovery procedure of the ad hoc routing protocol. When one of the two intermediate peers could not be discovered, the missing event is reported back to the AR server. The server repeats the construction of the new CIDS with a newly selected intermediate peer.

The route between the NS and the NR peer needs to be

discovered prior to data transmission. When the NS peer receives a CIDS, it should know the next virtual hop peer, the NR peer, and execute an ad hoc routing protocol. The NS peer relays the received data from the source peer to the NR peer over the found route. The NS peer becomes the packet source of the relayed data in view of the NR peer. The NS peer reconstructs a network-level header without changing the payload that is received from the source peer, and transmits the packet to the NR peer. Similarly, the NR peer unicasts the received data to its destination peer, the requesting peer. The backward route from the requesting peer to the source peer is used when the requesting peer needs to send acknowledgements back to the source peer. If the ad hoc routing protocol does not provide a backward route while constructing a forward route, the NR peer needs to discover an alternate route to the NS peer.

3 Simulation results

We use the *ns2* network simulator to evaluate the performance overhead of the anonymous connection. Overhead is incurred because the anonymous connection needs a larger number of hops between the source and requesting peers than for communication that has no anonymity. The simulation model assumes that the two intermediate peers are selected and the anonymous connection is established between the source and requesting peers in advance. The source peer has a file segment of 500 Kbytes. It transmits the file using TCP. Each peer has the 802.11 MAC with the transmission range of 250 meters. The transmission speed is 2 Mbps. The number of participating peers varies from 4 to 20. All peers are located within a 600 square meter grid unless specified otherwise. Due to the short completion time, peer mobility is not considered. In all cases, the ad hoc network is connected. One of the three TCP packet sizes is used (256, 512, and 1024 bytes), and the simulation uses 1024 bytes packets unless specified otherwise. The simulation adopts three methods of data transmission. The first transmission method, labeled "Direct" in the following figures, does not use an anonymous connection. That is, the packets are transmitted from the source peer to the requesting peer over the shortest path that is discovered by the ad hoc routing algorithm (AODV). The second method, labeled "1-Hop", selects two intermediate peers for the anonymous connection. One peer comes from the one-hop neighbors of the source peer and the other from the one-hop neighbors of the requesting peer. The third method, labeled "Random", randomly chooses two intermediate peers, excluding the source and requesting peers. This scheme produces a higher degree of anonymity and transmission delay than the 1-hop method. The overhead is measured by the completion time of transmitting 500 Kbytes file at which the source peer receives the last TCP ACK packet

Figure 3: Completion time of three different transmission methods

from the requesting peer.

The left side of Fig. 3 illustrates the completion time for receiving the 500th ACK packet when different numbers of peers participate in the ad hoc network. This figure and the following figures display the average of ten simulation runs from independently generated random topologies. According to the figure, both the 1-hop and the direct method show similar completion times, except for the random method. As the number of peers increases, the completion time of the random method increases. However, in the 12-peer case, some topologies generate relatively larger completion times, which takes more time than that of the 16-peer case. In the 8-peer case, some topologies have longer paths from the source to the destination than cases of other numbers of peers, which causes relatively longer completion times for the 1-hop and the direct method. In the 20-peer case, the 1-hop method takes 50% more time to complete the transmission than that of the direct method. The right side of Fig. 3 shows the completion time differences when the size of each TCP packet varies from 256 bytes to 1024 bytes in the 12-peer case. If the packet size is set to 256 bytes, the sender needs to transmit 2000 packets. When the packet size is 256 bytes, the 1-hop method needs 50% more time to complete than that of the direct method. In the case of 1024 bytes, an average of 78% more time is required.

The left side of Fig. 4 shows the completion time of transmitting 1024 bytes TCP packets with different sizes of simulation area. Each method has two different numbers of participating peers (N=12 and N=16). In both the 1-hop and the direct methods, the lower line indicates the result when 12 peers participate, and the upper line for the 16 peer case. The random method shows the crossed lines between the 400 meter case and the 600 meter case. When all peers are placed within one-hop transmission range (200 meter case),

the 1-hop method produces the worst performance. It is because only one peer can use the wireless medium at a time in this simulation. In general, as the simulation area enlarges, the completion time also increases because the number of hops between the source and the destination tends to increase. The right side of Fig. 4 displays the completion time in different size of simulation areas when 12 peers participate. In all methods, the use of small-sized packet takes more time. The 1-hop method takes the longest time in this simulations when all peers are within the transmission range of each other. The 1-hop method shows relatively smaller overhead when using 1024 bytes packet than using 512 bytes packet. In addition, the larger simulation area decreases the overhead of the 1-hop anonymous connection method to 44% (in the 800x800 meter case).

4 Conclusion

This paper introduces a CHUM network for a mobile pervasive computing system to reduce telecommunication costs by receiving content (or partial segments) via ad hoc connections from peers that have downloaded the content before. The receiving peer may use a direct ad hoc communication path to the sending peer, or may wish to use an anonymous channel to the sender in order to encourage privacy and to discourage traffic analysis. Two intermediate peers help establish the anonymous connection. Only the associated server of the receiving peer knows both the sender and the receiver of the anonymous connection. No other servers or any peers know both the sender and the receiver. The overhead for the anonymous connection is measured and compared with the direct communication between the sender and the receiver. The simulation results indicate that an 1-hop anonymous connection behaves similarly to the direct communication, and the overhead be-

Figure 4: Completion time of four different simulation areas

comes smaller when more peers participate or the peers are scattered over a larger area.

References

- [1] L. Garber, "Will 3G Really Be the Next Big Wireless Technology?," *IEEE Computer*, pp. 26–32, Jan 2002.
- [2] S. Kang and M. Mutka, "Mobile Peer Membership Management to Support Multimedia Streaming," *ICDCS Workshop on Mobile and Wireless Networks*, pp. 770–775, May 2003.
- [3] S. Kang and M. Mutka, "Chumcast in Two-Tier Networks," *Lecture Notes in Computer Science (Springer)*, vol. 3090, pp. 124–133, May 2004.
- [4] S. Kang and M. Mutka, "Efficient Mobile Access to Internet Data via a Wireless Peer-to-Peer Network," *IEEE Int'l Conference on Pervasive Computing and Communications*, pp. 197–205, March 2004.
- [5] D. Zhu and M. Mutka, "Fair Sharing of Proxy Responsibilities in an Ad Hoc Network," *IEEE Int'l Conference on Pervasive Computing and Communications*, March 2004.
- [6] D. Zhu and M. Mutka, "Sharing Presence Information and Message Notification in an Ad Hoc Network," *IEEE Int'l Conference on Pervasive Computing and Communications*, pp. 351–358, March 2003.
- [7] E. Gabber, P. B. Gibbons, Y. Matias, and A. Mayer, "How to Make Personalized Web Browsing Simple, Secure, and Anonymous," in *Proceedings of Conf. on Financial Cryptography*, pp. 17–31, February 1997.
- [8] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," in *Proceedings of the 13th USENIX Security Symposium*, pp. 303–320, August 2004.
- [9] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for Web Transactions," *ACM Transactions on Information and System Security*, vol. 1, pp. 66–92, November 1998.
- [10] C. Andersson, R. Lundin, and S. Fischer-Hubner, "mCrowds: Anonymity on the Mobile Internet," in *Proceedings of the 2nd IFIP Summer School*, August 2003.
- [11] M. J. Freedman and R. Morris, "Tarzan: A Peer-to-Peer Anonymizing Network Layer," in *ACM Conference on Computer and Communications Security*, pp. 193–206, November 2002.
- [12] R. Sharwood, B. Bhattacharjee, and A. Srinivasan, "P5: A Protocol for Scalable Anonymous Communications," in *IEEE Symposium on Security and Privacy*, pp. 53–65, May 2002.
- [13] V. Scarlata, B. N. Levine, and C. Shields, "Responder Anonymity and Anonymous Peer-to-Peer File Sharing," in *Proceedings of ICNP 2001*, pp. 272–280, November 2001.
- [14] L. Xiao, Z. Xu, and X. Zhang, "Low-cost and Reliable Mutual Anonymity Protocols in Peer-to-Peer Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, pp. 829–840, September 2003.