Ad Hoc Network에서 익명성 제공에 관한 연구☆

On Providing Anonymity in Ad Hoc Networks

강 승 석*
Seung-Seok Kang

요 약

네트워크 환경은 외부의 침입으로부터 항상 노출되어 있으며 프라이버시 보장에도 위험이 따른다. 전파 송신을 통한 브로드캐스트의 속성 때문에 무선기기들은 유선 네트워크에 연결된 통신기기보다 더 많은 위험 상황을 경험하게 된다. 본 논문은 무선통신 기기가 두 개의 통신채널을 장착하고 있다고 가정한다. 하나는 3G 서비스를 통해 인터넷을 접속하는데 사용되며 다른 하나는 애드 혹 네트워크를 구성하여 익명의 통신 서비스를 제공하는데 사용된다. 프라이버시 위험을 극복하기 위해 본 논문에서는 무선기기들이 애드 혹 네트워크를 구성해 익명 통신채널을 구축하여 이를 이용해 자료를 교환하는 방법을 소개한다. 익명 통신채널을 구축하기 위해서는 하나 이상의 무선기기가 익명 채널의 경로에 참여해야 한다. 제안한 익명 통신방법은 트래픽 분석을 어렵게 하며 사용자의 프라이버시를 향상시킨다. 모의실험 결과에 따르면, 애드 혹 네트워크에서 익명성을 제공하는 통신은 송신자나 수신자 근처에 있는 중간 무선기기를 선택하여 익명 채널의 경로로 선택하는 것이 무작위로 중간 무선기기를 선택하는 방식보다 좋은 성능을 보였다.

Abstract

Networking environments are exposed to outside attacks and privacy threats. Due to broadcast nature of radio transmissions, wireless devices experience more vulnerable situations than those of wired network devices. This paper assumes that a wireless device has two network interfaces, one for accessing Internet using 3G services, and the other for constructing an ad hoc network. To deal with privacy threats, this paper introduces an approach in which wireless devices form a special ad hoc network in order to exchange data using anonymous communications. One or more intermediate peers should be involved in the construction of an anonymous path. The proposed anonymous communication mechanism discourages traffic analysis and improves user privacy. According to simulation results, the anonymous connection in an ad hoc network prefers the intermediate peer(s) which is located near the source and/or the destination peer, rather than randomly-selected peers.

☞ Keyword : anonymous communication, ad hoc network, dual channel, traffic analysis

1. Introduction

There has been considerable interest in the research community to provide privacy and anonymity between two or more communicating devices. In case of transmitting private data over the networks, users might want the data to be securely encrypted. However, outside attackers want the content of the exchanged data as well as other information such as who sends to whom. The purpose of traffic analysis is to help which network device is talking to which device by analyzing traffic patterns instead of the content that is transmitted. Although data is encrypted, traffic analysis may detect the two peers communicating. The environment of the wireless networks are more susceptible from the traffic analysis due partially to more resource constraints and less research work performed compared with that of the wired

* 정 회 원 : 서울여자대학교 정보미디어대학
컴퓨터학부 전임강사
msukang@swu.ac.kr
[2007/03/12 투고 - 2007/03/22 심사 - 2007/06/20 심사완료]
☆ This work was supported by a special research grant from Seoul Women's University (2007)

networks. Moreover, the data transmitted from each wireless device is always open to its neighbor devices. One of its neighbors may be a network attacker and it is able to analyze the in-transit packets and to identify who is a sender and who is a receiver. Any *ad hoc* network that provides anonymity should hide or disguise these information. This paper focuses on providing anonymity in *ad hoc* networks in order to discourage traffic analysis.

It becomes a trend [1, 2, 3] that each wireless mobile device has two network interfaces, one for 3G networks, such as UMTS or CDMA 2000, 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. Device users may use WLAN interface in public hot-spots including airports, hotels, and libraries. Whereas, the 3G network covers wider area with providing reasonable mobility management support. In this paper, the participating mobile devices, called the peers, have two wireless channels, one for accessing the Internet using the 3G connection, and the other for constructing an *ad hoc* network.

An anonymous connection makes it difficult for others to determine which peers are communicating. In order for two peers to communicate anonymously, at least one additional peer is involved in the connection. When a sending peer transmits data to a receiving peer, the sending peer builds an anonymous path between them with the aid of its associated server. The anonymous path includes some additional peers placed in between the sending peer and the receiving peer. After constructing an anonymous path, the sending peer transmits data to one of the in-between peer that delivers data to the specified next intermediate peer, and finally to the receiving peer. When an outside attacker analyzes the data traffic, it will collect a false information about the sending /receiving peer. Onion routing [4] indicates that a single intermediate node is sufficient to complicate traffic analysis. Even if an attacker participates in the network and plays a role of an in-between peer, this case will fail when two or more additional peers are involved in the anonymous connection.

There have been several types of anonymity systems studied and developed. Proxy based anonymity systems, such as Anonymizer [5], and Lucent Personalized Web Assistant (LPWA) [6], work between a requesting (initiating) host and a source (responding) host. Some systems provide receiver anonymity [7, 8] and sender anonymity including Mix[9], Onion (Tor) [4], Crowds [10], Hordes [11], and Tarzan[12]. Mutual anonymity is provided in P^5 [13], APFS [14], and Shortcut Responding [15]. Location anonymity in mobile networks are studied in mCrowds [16] and [17]. These focus on concealing the location information of mobile device users, whereas this work deals mainly with how to transmit data between two mobile users anonymously in the ubiquitous environment.

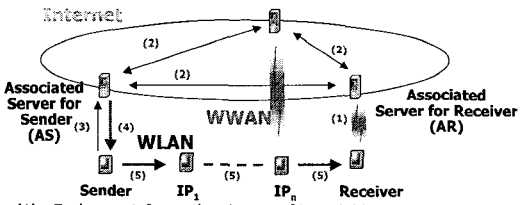
This paper describes the functionality of the anonymous communications in *ad hoc* networks. The network operation and management are discussed in section 2. Simulation results are explained in section 3. Section 4 draws conclusion.

2. Anonymous Connection Construction

This section briefly explains how to construct a special *ad hoc* network, and to setup an anonymous connection between two peers in the network.

2.1. Ad hoc network formation

The detailed formation procedure of the special *ad hoc* network is described in [2, 3]. An initiating peer contacts a server, called the associated server, in the Internet in order to acquire a peer ID (PID) and a network ID (NID) of the *ad hoc* network. Subsequent peers joining the *ad hoc* network also associate with their servers. The associated server may locate at an ISP or some other places in the Internet. The server may associate with several peers. For some cases, all peers may associate a single server that manages all participating peers in an ad hoc network. Each associated server exchanges information of its associated peers with other associated servers of the participating peers. All participating associated servers cooperate each other to provide several services including the provisioning anonymous connections. The overall formation sequence is displayed in figure 1.



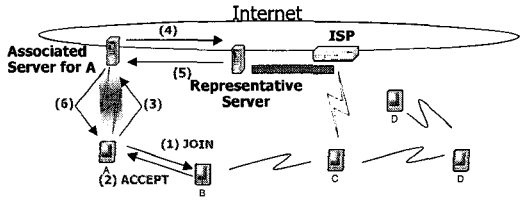
(1) Each peer informs the change of its neighbors discovered by Hello Packets or in promiscuous mode
(2) Associated servers exchange neighbor information each other
(3) Sender requests an anonymous connection with Receiver
(4) AS creates ACPI and passes it to Sender
(5) Construction of anonymous communication path proceeds

(Figure 1) Joining sequence of a mobile device

When an associated server accepts a new peer, it reports the event to other associated servers. Each peer periodically senses its neighbor peers by periodic HELLO packets or in promiscuous mode. In case that a peer detects a change of its neighborhood peers, it also reports the change to its associated server. The associated server, then, delivers new neighborhood information about the peer to other associated servers, so that each associated server is able to keep up-to-date global topology information of the *ad hoc* network. Any data exchanged between servers are encrypted to hide the connection information in the wired networks.

2.2. Anonymous connection setup

In this paper, the participating mobile devices, called the peers, have two wireless channels, one for accessing the Internet using the 3G connection, and the other for constructing an *ad hoc* network. The anonymous connection setup requires the participation of some intermediate peers, which makes traffic analysis difficult. Figure 2 displays the overall sequence of the anonymous connection setup.



(1) New device sends a JOIN packet to existing network
(2) ACCEPT packet contains info about the representative server
(3) Peer delivers representative server info to its associated server
(4) Associated server informs its peer to representative server
(5) Representative server passes NID, PID, and info about other participating associated servers
(6) Both NID and PID are passed to its associated peer

(Figure 2) Anonymous connection setup sequence

When a peer, called Sender, wants to transmit data anonymously to another peer, called Receiver, Sender asks an anonymous connection to its associated server, called AS. Sender has an option to select the degree of anonymity by employing different number of intermediate peers. AS selects some intermediate peers between Sender and Receiver depending on the anonymity degree. It,

next, creates an anonymous connection path information (ACPI) that is described in the following subsection, and passes it to Sender. The anonymous data path starts from Sender to the first intermediate peer IP_1 up to the N^{th} intermediate peer IP_n, and finally reaches to Receiver. One anonymous connection with N intermediate peers consists of $N+1$ virtual hops. The sender of each virtual hop transmits data to the receiver of the virtual hop, and the receiver becomes the sender of the next virtual hop. This chaining process finishes when the data reaches at Receiver. The route between the sender and the receiver of each virtual hop can be found using well-known *ad hoc* routing protocols such as AODV [18].

There exists a tradeoff between the degree of anonymity and the transmission performance. Additional intermediate peers increase the degree of anonymity, but decrease the transmission performance because it requires to visit additional peers to deliver packets. Sender may choose the degree of anonymity, and its associated server may satisfy the degree by selecting the corresponding number of intermediate peers. It is typical to select one or two intermediate peers, and three or more are possible in order to increase the degree of anonymity at the cost of degrading the transmission throughput. In addition, the selection of which intermediate peers also greatly affects the overall performance. The participation of the random number of intermediate peers, used in Crowds [10], increases the anonymity degree. In the simulation in section 3, it compares several transmission performance results depending on the selection of how many intermediate peers, and on the method of how to select the intermediate peers.

2.3. Anonymous Connection Path Information

The anonymous connection path information (ACPI) is a collection of virtual hop information between Sender and Receiver. The ACPI is a multi-layered data structure depending on the degree of anonymity. Each layer has the same size and the same format. One layer contains a destination peer address and three keys for its bi-directional virtual connection showed in figure 3.

	Next Address	Crypt for Padding	Crypt for Forward Data	Crypt for Backward Data
For Sender	IP_1	Sym Key P_1	Sym Key E_1	Null
For IP_1	IP_2	Sym Key P_2	Sym Key E_2	Sym Key E_1
	:	:	:	:
For IP_{n-1}	IP_n	Sym Key P_n	Sym Key E_n	Sym Key E_{n-1}
For IP_n	Receiver	Sym Key P_{n+1}	Sym Key E_{n+1}	Sym Key E_n
For Receiver	Null	Null	Null	Sym Key E_{n+1}

(Figure 3) Format of ACPI

The outermost layer is for Sender, and the second outermost layer is for the first intermediate peer, and so on. The innermost layer of the ACPI is used by Receiver. Each layer consists of four elements, next address, symmetric key for padding, symmetric key for forward data, and symmetric key for backward data. When a peer in the anonymous path receives the ACPI, it applies its private key on the fixed-sized top layer to read the virtual connection information such as a destination address and keys because the layer is encrypted using the public key of the peer. As the peer knows information about its next destination peer and the keys, it is able to deliver the ACPI to its next peer. Before sending the ACPI, the peer removes the top layer of the ACPI and applies the first key on the remaining

layers of the ACPI. The new ACPI will become smaller in size and be transmitted to the next destination peer in order to build a bi-directional virtual connection. As an example, when Source receives an ACPI, it applies its private key on the fixed-sized top layer to acquire the address of the next virtual hop peer IP_1 and three keys. The second key, E_1, will be used to encrypt data transmitted to IP_1. The third key, E_2, is used for backward transmission, but it is useless for Sender. Sender removes the top layer and applies the first key, P_1, on the remaining part of the ACPI which starts from "for IP_1" to the end of ACPI. Sender could not recognize the address of IP_2 after decrypting the ACPI because the decrypted part is also encrypted with the public key of the IP_1. Sender transmits the decrypted ACPI to IP_1 so that it is able to construct a virtual connection between Sender and IP_1. IP_1 performs a similar task and creates a virtual connection with the next intermediate peer such as IP_2. The backward key for IP_1 is the same as the forward key for Sender. When Receiver gets the ACPI from IP_n, it finds that Receiver itself is the final destination of the virtual path because the destination address is set to a null value. As a result, a series of actions generates a bidirectional anonymous virtual connection between Sender and Receiver with the aid of N intermediate peers.

The AS server needs to create the innermost layer of the ACPI first. The innermost layer will be encrypted using the public key of Receiver. The server, then, creates the second innermost layer for IP_n peer. This layer will be encrypted using the public key of the IP_n peer, and the remaining part (the layer for Receiver) will be encrypted with the key $P(n+1)$. This process repeats until the outermost layer for Sender is encrypted by the public key of Sender and the remaining part (the $(n+1)$ inner layers) is done with the symmetric key $P1$.

As an anonymous connection is established, Source receives an acknowledgement from Receiver over the backward connection. Source, then, starts transmitting data to its next virtual hop peer, IP_1. If one of the two peers in a virtual hop moves away and the connection is broken, the source peer of the virtual hop should recover the path by using the path recovery procedure of the *ad hoc* routing protocol. When the next virtual hop peer could not be discovered, the missing event is reported back to the associated server. The server repeats the construction of the new ACPI with a newly selected intermediate peer.

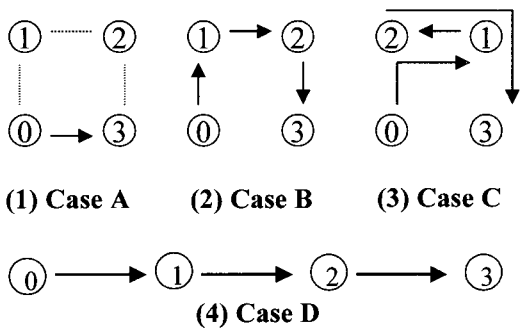
The route between intermediate peers needs to be discovered prior to data transmission. When IP_1 receives an ACPI packet, it should know the next virtual hop peer, and executes an *ad hoc* routing protocol. IP_1 relays the received ACPI from Sender to the next intermediate peer over the found route. IP_1 becomes the packet source of the relayed data in view of the next intermediate peer. IP_1 reconstructs a network-level header without changing the payload that is received from Sender, and transmits the packet to the next intermediate peer, such as IP_2. Similarly, IP_2 unicasts the received data to its next virtual destination peer. The backward route from Receiver to Sender is used when Receiver needs to send acknowledgements back to Sender. If the *ad hoc* routing protocol does not provide a backward route while constructing a forward route, IP_n peer needs to discover an alternate route to IP_{n-1} peer.

3. Simulation results

This section describes the performance overhead of the anonymous connection using the *ns2* network simulator [19]. The anonymous connection incurs an overhead because it needs an additional number of hops between Sender and Receiver than for the communication that has no anonymity feature. The simulation model in this paper assumes that the small number of intermediate peers are selected and the anonymous connection is established between Sender and Receiver in advance depicted in figure 2. Assume that Sender has a file of 500 Kbytes, and it transmits the file using TCP. Each peer has the 802.11 MAC with the transmission range of 250 meters and the transmission speed of 2 Mbps. The overhead is measured by the completion time of transmitting 500 Kbytes file at which Sender receives the last TCP ACK packet from Receiver.

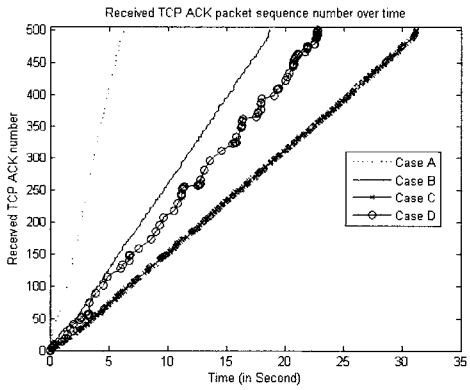
3.1. Four peer case simulation

Figure 4 shows five simple topologies consisting of four peers labeled from 0 to 3. The topologies assume that peer 0 is Sender and peer 3 is Receiver. Further assume that two intermediate peers are employed to build an anonymous connection.



(Figure 4) Four-peer network topoloties

For example, both in Case B and C, peer 1 is the next anonymous connection peer from peer 0, and peer 2 is the next from peer 1. Case A uses no anonymous feature and Sender communicates with Receiver directly. Case B uses a predefined anonymous connection and it needs three hops to transmit packets from the source to the destination. In Case C, two intermediate peers are selected with five hops to deliver packets from Source to Receiver. In Cases A, B, and C, each peer is directly connected with only two other peers. All peers are placed in a line in Case D. This case expects no performance overhead of the anonymous connection because the packets travel the same path with and without the anonymous feature.

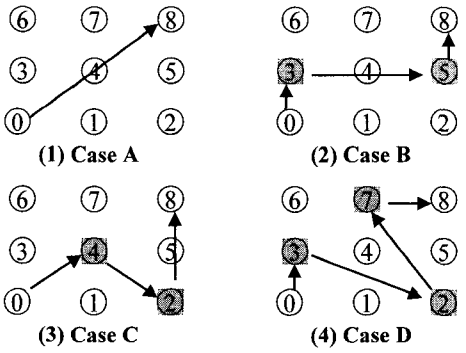


(Figure 5) Completion time with four peers

Figure 5 displays the sequence of times receiving 500 ACK packets by peer 0 when peer 0 transmits 500 1024-byte TCP packets to peer 3. Case A completes its transmission in less than 7 seconds. Cases B takes somewhat less than 20 seconds, whereas, case D finishes slightly later than 20 seconds. Case C finishes its transmission taking more than 30 seconds because it follows the longest anonymous path.

3.2. Nine-peer-case simulation

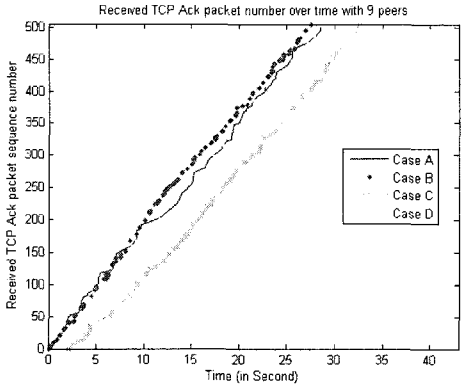
Figure 6 displays several types of anonymous paths from Sender to Receiver. Nine peers are participated and peer 0 is Sender and peer 8 is Receiver. The simulation area in figure 6 is doubled in length and width compared to that in figure 4. The enlargement of the simulation area allows to have diverse choice to select intermediate peers. The intermediate peers are marked as shaded nodes in figure 6. The distance between each adjacent peer is 200 meters. This means each peer is able to directly reach peers located only at its left, right, up, and down side peers. No peer transmits data directly to its diagonally-located peers. For example, when peer 0 transmits data to peer 4, it requires two hops to deliver data, so peer 1 or peer 3 should be in the transmission path.



(Figure 6) 9-peer network topologies

In Case A, peer 0 needs at least four hops to send data to peer 8. AODV routing protocol will decide the path from Sender to some predefined intermediate peers, and to Receiver. This case delivers data without anonymity feature. Case B selects peer 3 and peer 5 as an intermediate peer in order to construct an anonymous path between Sender and Receiver. Case C has two intermediate peers for the anonymous connection. The difference between the case B and the case C is that both intermediate peers are the direct neighbors of Sender and Receiver in case B. The case C improves the degree of anonymity, but may expect longer time to complete the transmission. Case D adopts three intermediate peers to complicate the traffic analysis at the cost of degrading the transmission performance.

Figure 7 shows the performance overhead of the anonymous transmission for another four cases with four peers. Each line indicates the sequence of times receiving 500 ACK packets by peer 0 when peer 0 transmits 500 1Kbyte TCP packets to peer 8. The lines for Case A and B are very close, and Case D takes relatively long time to complete. Case C is placed in the middle.



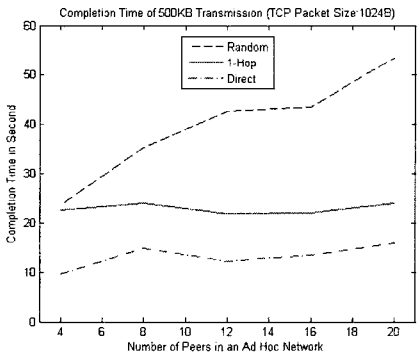
(Figure 7) Completion time with nine peers

One of the implications in this figure is that Case B offers some degree of anonymity, and it performs similarly as Case A which has no anonymous feature. That is, when the associated server of the sender selects some intermediate peers, it is better to choose the peers located near Sender and/or Receiver. One heuristic rule of the peer

selection is to pick 1-hop neighbor(s) of Sender and/or Receiver depending on the degree of anonymity. Random selection, displayed in Case C and D may incur longer delay. The selection of three or more intermediate peers are possible, illustrated in Case D, which may increase the anonymity degree at the cost of large transmission overhead.

Second implication is that the transmission overhead becomes smaller as the number of hops between Sender and Receiver is large. In figure 5, the overhead of Case C is more than four times larger than that of Case A. However, in figure 7, the overhead becomes relatively small compared with the same degree of anonymity in figure 5.

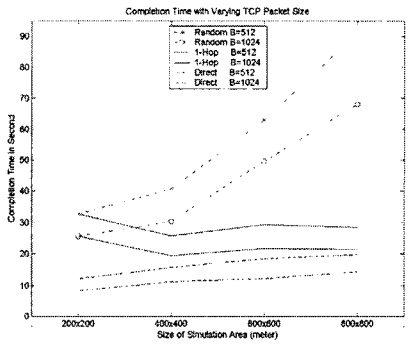
3.3. Random topology case simulation



(Figure 8) Completion time with multiple peers

Figure 8 uses two intermediate peers for constructing an anonymous connection. It illustrates the completion time for sending 500 1 Kbyte TCP packets and receiving the 500th ACK packet with the number of participating peers varying from 4 to 20. All peers are located in a 600 square meter grid. Because of the short completion time, the mobility of peers is not considered. In all cases, the *ad hoc* networks are connected. That is, each peer is connected to at least one other peer. This figure displays the average of ten simulation runs from independently generated random topologies.

This figure adopts three methods to select the intermediate peers. The first method, labeled "Direct, does not use any intermediate peers. Packets will follow the route found by the AODV protocol. The second method, labeled "1-Hop", chooses two intermediate peers, one from the neighbors of Sender and the other from the neighbors of Receiver. The third method, labeled "Random", uses two arbitrary peers excluding Sender and Receiver. According to the figure, both the 1-hop and the direct method follows similar shape of completion time. As the number of peers increases, the completion time of the random method increases. In the 20-peer case, the 1-hop method takes 50% more time to complete the transmission than that of the direct method.

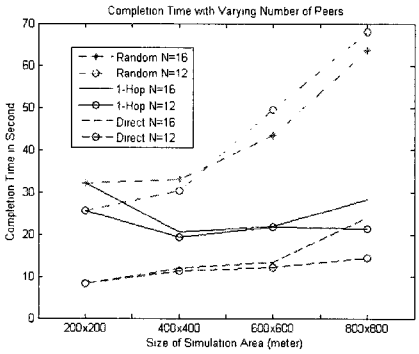


(Figure 9) Completion time with varying packet size

Figure 9 displays the completion time in different size of simulation areas when 12 peers participate. As the packet size changes from 1 Kbyte to 512 byte, Sender needs two times more packets in order to transmit 500 Kbyte of data. 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 (in 200x200 meter case). 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 800x800 meter case).

Figure 10 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 simulations. 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.



(Figure 10) Completion time with varying # peers

4. Conclusion

This paper introduces a special *ad hoc* network in which any participating mobile device is able to communicate each other anonymously. The construction of an anonymous path includes some intermediate peers in between a sending peer and a receiving peer. More intermediate peers increase the degree of anonymity and discourage traffic analysis. However, one intermediate peer is enough to provide anonymous communication. The wise selection of intermediate peers affects the overhead of the anonymous communication. In order to reduce the overhead, it is prefer for intermediate peers to be located on the shortest (routing) path between the sender and the receiver. The overhead for the anonymous connection is measured and compared with the direct connection. It is desirable to select two 1-hop intermediate peers from the original sender and the receiver because it gives reasonable degree of anonymity and similar communication overhead than that of choosing a single intermediate peer or the direct connection. The wise selection is inevitable when three or more intermediate peers are involved.

Reference

[1] 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

[2] S. Kang, 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, 2004

[3] S. Kang, M. Mutka, "A mobile peer-to-peer approach for multimedia content sharing using 3G/WLAN dual mode channel", Wireless Communications and Mobile Computing vol 5 pp.633-645, 2005

[4] R. Dingledine, N. Mathewson, P. Syverson, "Tor: The Second-Generation Onion Router", In: Proceedings of the 13th USENIX Security Symposium, pp.303-320, 2004

[5] E. Gabber, P. Gibbons, Y. Matias, A. Mayer, "How to Make Personalized Web Browsing Simple, Secure, and Anonymous", In Proceedings of Financial Cryptography", pp.17-31, 1997

[6] E. Gabber, P. Gibbons, D. Kristol, Y. Matias, A. Mayer, "Consistent, yet anonymous, Web access with LPWA", Communications of the ACM vol. 42, pp.42-47, 1999

[7] D. Chaum, "The Dining Cryptographers Problem: Unconditional sender and recipient untraceability" , Journal of Cryptology, vol. 1 pp.65-75, 1988

[8] S. Dolev, R. Ostrovsky, "Xor-Trees for Efficient Anonymous Multicast and Reception", ACM Transactions on Information and System Security (TISSEC) vol. 3, pp.63-84, 2000

[9] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", Communications of the ACM", vol. 24, pp.84-88, 1988

[10] M. Reiter, A. Rubin, "Crowds: Anonymity for Web Transactions", ACM Transactions on Information and System Security vol. 1, pp. 66-92, 1998

[11] C. Shields, B. Levine, "A Protocol for Anonymous Communication Over the Internet", ACM Conference on Computer and Communications Security, pp.33-42, 2000

[12] M. Freedman, R. Morris, "Tarzan: A Peer-to-Peer Anonymizing Network Layer", ACM Conf. on Computer and Communications Security, pp. 193-206, 2002

[13] R. Sharwood, B. Bhattacharjee, A. Srinivasan, "P5: A Protocol for Scalable Anonymous Communications", In: IEEE Symposium on Security and Privacy, pp.53-65, 2002

[14] V. Scarlata, B. Levine, C. Shields, "Responder Anonymity and Anonymous Peer-to-Peer File Sharing". In Proceedings of ICNP, pp.272-280, 2001

[15] L. Xiao, Z. Xu, 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, 2003

[16] C. Andersson, R. Lundin, S. Fischer-Hubner, "mCrowds: Anonymity on the Mobile Internet", In Proceedings of the 2nd IFIP Summer School, pp.4-8, 2003

[17] A. Pascual, "Kista - IT University Wireless Network. Privacy in mobile internetworking?", In Proceedings of Internet Society Conference (INET2001), pp.11-12, 2001

[18] C. Perkins, E. Royer, "Ad-hoc On-Demand Distance Vector Routing", Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp.90-100, 1999

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

◐ 저 자 소 개 ◑

강 승 석(Kang, Seung-Seok)
1992년 고려대학교 전산과학과 졸업 (이학사)
1998년 Michigan State University Dept. of Computer Science (공학석사)
2004년 Michigan State University Dept. of Computer Science & Engineering (공학박사)
2006~현재 서울여자대학교 정보미디어대학 컴퓨터학부 전임강사
관심분야 : 애드 혹 네트워크, QoS, Anonymity, 센서 네트워크, Wireless Communication
E-mail : msukang@swu.ac.kr