Low Latency High Bandwidth Anonymous Overlay Network with Anonymous Routing

Roman Schlegel and Duncan S. Wong

Department of Computer Science
City University of Hong Kong

Abstract. Most existing anonymous networks focus on providing strong anonymity for the price of having lower bandwidth, higher latency and degraded usability when compared with the conventional use of the Internet. They also often anonymize only a few specific applications. In this paper, we propose a new approach of constructing an anonymous network. The network consists of an overlay network, which provides anonymity to all applications running on top of it, and a routing protocol, which can be considered as an anonymized version of path vector routing. The protocol preserves the high performance characteristics of the path vector routing and also has the added advantage of hiding the overlay network topology. Our simulation results show that the expected latency of our approach is 50% better than that of existing systems. Besides the new anonymous routing protocol, this paper aims to provide the general overview of this new anonymous overlay network which may serve as the input for further research.

1 Introduction

Anonymity on the Internet is a topic which has been the subject of research of many papers and numerous anonymous networks have been proposed [3,4,10,7,6,16,14,8,11]. Most of these networks are based on one of two methods: onion routing and DC-nets [3,4,10]. Anonymous networks have many useful applications. For example, information which might be censored for political reasons in some countries may be published and accessed anonymously using anonymous networks over the Internet; similarly, a whistleblower may use an anonymous network to publish relevant information. In this case it is mostly the whistleblower himself who needs to remain anonymous, not the website to which he publishes the information; and organizations providing medical information can provide web services within an anonymous network so that people can access information anonymously. This can prevent them from being discriminated against because of a medical condition.

Many of the proposed anonymous networks [7,16,8,11,6,14] provide good anonymity. Nevertheless, most of them have drawbacks which limit their usefulness or hinder adoption. These disadvantages include high latency, limited bandwidth, narrow suitability (e.g. only individual protocols can be anonymized and applications have to be re-configured). Research into the performance of Tor [6] and JAP [14] shows a high latency (e.g. around 4 seconds for some Cascades in JAP, around 4.5 seconds for Tor) and either a low (e.g. around 15 KB/s for JAP) or inconsistent bandwidth (e.g. 0 to 90 KB/s for Tor with a mean of 40 - 45 KB/s) offered by existing anonymous networks [15], when compared to the 0.4 second average loading time of the headers of a website on the conventional non-anonymous Internet.

Our Contributions. In this paper, we propose a new approach for constructing an anonymous network. The idea behind is twofold. First, we build an anonymous overlay network, then we propose an anonymous routing protocol which not only protects the anonymity of routing nodes but also helps route traffic for lower latency and higher bandwidth. Every node on our anonymous overlay network obtains a set of randomized identifiers called *overlay addresses*. The overlay addresses are dynamic and can be changed from time to time, multiple overlay addresses can co-exist on a single node at any given time and more importantly, we introduce mechanisms which ensure that no adversary would be able to correlate the overlay address(es) of a particular node with its real IP address under the condition that the adversary does not have full control of the entire network.

Unlike other anonymous networks, the network is not circuit-based but packet-based. In other words, the two-way communications between any two nodes could go through two different paths. Furthermore, the anonymous routing protocol, which can be considered to be an anonymized version of path vector routing, preserves the performance characteristics of the path vector routing but has the added advantage of hiding the overlay network topology. The anonymity of the overlay network relies heavily on the number of participating nodes, the higher the number of nodes joining the network, the higher the level of anonymity of the nodes in the network would be.

Our simulation results show that the expected latency of our proposed anonymous network is 50% better than that of existing systems. Additionally, our approach provides a generic network layer which can run almost any application on top of it anonymously.

Paper Organization. In the next section, we review some major anonymous networks previously proposed. This is followed by the description of our anonymous overlay network in Sec. 3. In Sec. 4, we describe the anonymous routing protocol and in Sec. 5, we conclude the paper and indicate some of our future work.

2 Related Work

As mentioned in Sec. 1, among the existing solutions to anonymity on the Internet, most of them go back to the idea of mixes [3], a system where messages are randomly passed between a number of hosts and each host performs specific cryptographic operations on a message to obfuscate who is communicating with whom. A popular technology building on these mixes is circuit-based onion routing [10]. In onion routing, a message is encrypted repeatedly (“onion layers”) and then sent along a path of different hosts. Each host in the path removes the outermost layer of the encryption which reveals the next hop in the path. This is repeated until the packet reaches the last hop which can unwrap the last layer and access the message itself. Using onion routing, each host only knows the preceding host and the next host in the whole path, making the communication between the source and the destination anonymous. Systems based on the idea of onion routing include for example Tor [6], Tarzan [7] and Cashmere [16].

One of the disadvantages of onion routing is that to actually provide anonymity the path length between the source and the destination has to be artificially inflated. Each communication passes through several intermediate hops which is less efficient in terms of latency and bandwidth than a direct connection. Measurements by Panchenko et al. show that performance is considerably worse compared to normal, direct network traffic, with a mean-time of 4 seconds just to load the headers of a website compared to 0.4 second for the non-anonymous Internet access [12]. Our approach for an anonymous network tries to improve the performance by actively looking for paths with low latency and high bandwidth without leaking the network topology or any information about the actual locations of the source and destination.

Another class of anonymous networks is based on the problem of the dining cryptographers introduced by Chaum [4]. This class of networks uses XORing of bit vectors to transmit information with provable untraceability. One issue about DC-nets is that the available bandwidth drops off with the square of the number of users and therefore does not scale well. One system building on DC-nets is Herbivore, which tries to solve the scalability problem through partitioning, with the trade-off of reducing untraceability to individual partitions [8].

3 Anonymous Overlay Network

The anonymous network we propose here is in the form of an *overlay network*. Besides anonymity, the other objective of proposing such a new anonymous network is to achieve a network performance in terms of low latency and high bandwidth. We start our description by specifying the underlying adversarial model.

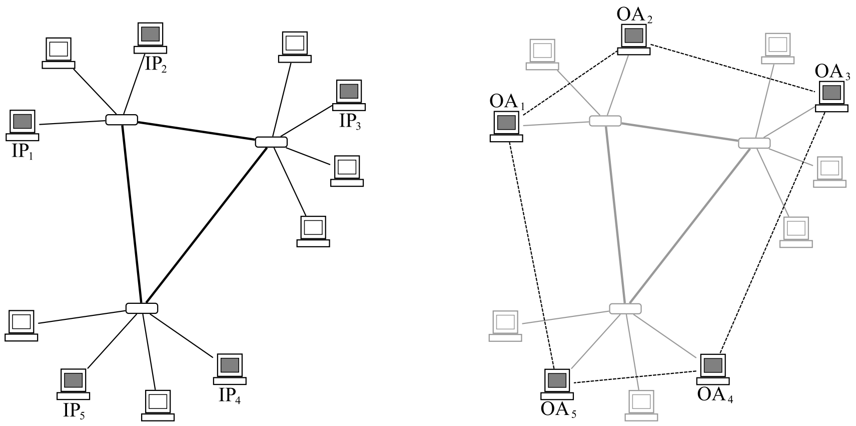


Fig. 1. Schematic drawing of an overlay network. The left drawing shows the physical connections, the right drawing shows the virtual/logical connections among the nodes which are part of the overlay (shaded nodes). Also, addresses in the physical network are different from the addresses in the overlay network.

3.1 Attack Model

We consider an adversary which may have full control of several parts of the network but do not have the control of the entire network. Full control means that besides eavesdropping, the adversary may even have corrupted and taken full control of several nodes in the network. However, we do not assume that all the nodes in the network have been corrupted (or are being malicious).

Although the attacker can launch traffic analysis, timing attacks and many different types of statistical attacks, by assuming a "non-global" attacker, we can see that we are considering the scenario that the attacker is less capable of launching an effective attack. This is different from some previous attack model (e.g. Tarzan [7]), in which a global eavesdropper is considered, with the trade-off of having less desirable performance.

In the following, we start the description of our anonymous network by giving an overview of the overlay network we propose. This is followed by the detailed description of each component in the network.

3.2 Overview

The overlay network consists of nodes which form a virtual network on top of the Internet by making logical links among them. Fig. 1 shows an example. The left part of the figure illustrates the physical connections among the nodes but only the shaded nodes (with IP addresses) are part of the overlay network. The right part shows the topology of the overlay network. As usual, the virtual links that delineate the topology of the overlay network also illustrate the relation of *virtual neighbors* on the overlay network. For example, the node with overlay address OA_5 has two virtual neighbors which have overlay addresses OA_1 and OA_4. However, due to anonymity, the topology of the overlay network is hidden to all the nodes on the network (we will see shortly on how this is done). As a result, the node OA_5 does not know the overlay addresses of its two virtual neighbors. Instead, it only knows that it can connect to the overlay network via two other nodes with real IP addresses IP_1 and IP_4. The IP addresses of these two virtual neighbors of the node OA_5 will be used when forwarding data to their destinations which are specified using overlay addresses. We now provide the details in the following.

A new node which is about to connect to the anonymous overlay network will take the following steps. For illustration, suppose the new node has IP address IP_5 (Fig. 2 - Fig. 5).

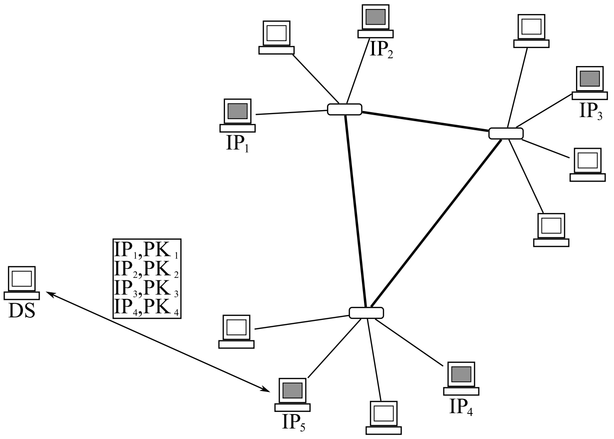


Fig. 2. The new node (IP_5) connects to the directory server DS and obtains a list of nodes which are already part of the overlay network. Each entry on the list also has an associated public key.

1. **Directory Server**: The new node first contacts a public directory server DS which returns a list of IP addresses of nodes which are already part of the overlay network (see Fig. 2). Furthermore, each node in the list also has a public key associated.
2. **Connection Establishment**: The node then arbitrarily selects a certain number of nodes from the list and connects to them. For each of these nodes, the new node also runs an authenticated key establishment protocol (e.g. authenticated Diffie-Hellman [5,2]) to establish an encrypted and authenticated connection (using the public keys associated with each node). This is exemplified in Fig. 3.
3. **Overlay Address**: Carried out independently from the previous step, the new node also randomly chooses one or several overlay addresses and anonymously connects to a central authority for reporting and checking if these overlay addresses are already in use. If so, the

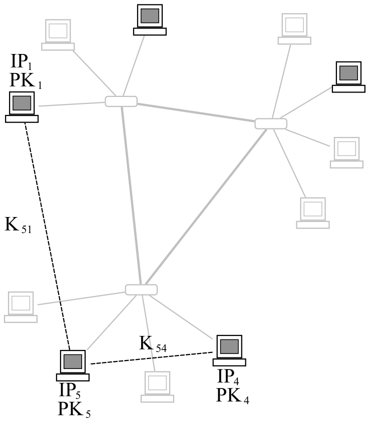


Fig. 3. The new node (IP_5) chooses two nodes from the list, connects to them, and establishes session keys K_{51} and K_{54} with these nodes. The session keys are used for forming encrypted and authenticated channels. Public keys (PK_5, PK_1, PK_4) are used during the authenticated key establishment process. The nodes with IP address IP_1 and IP_4 will be the *virtual neighbors* of the new node on the overlay network.

node will randomly pick another set of addresses and try again until free overlay addresses are found.

The purpose of choosing several overlay addresses on one single physical node is to eliminate any correlation between the number of virtual nodes (identified by their overlay addresses) on the overlay network and the actual number of physical nodes on the Internet that have joined the overlay network. The central authority can be a directory server or simply a bulletin board which has overlay addresses posted. We will give more details about the overlay address in Sec. 3.3.

The right drawing of Fig. 1 shows an example of having five virtual nodes over the overlay network, each having a unique but randomly generated overlay address. For simplicity, the drawing shows the scenario where each physical node has only one overlay address. However, we stress that in general, a physical node can have multiple overlay addresses to achieve better anonymity.

4. **Routing**: Once overlay addresses are set, the node (precisely, a set of new virtual nodes with overlay addresses) can start running an anonymous routing protocol (Sec. 4) for learning how to reach other virtual nodes on the overlay network. The anonymous routing protocol will ensure that

   (a) the relationship between the real IP address of a physical node and the set of virtual overlay addresses of the node cannot be determined by any other nodes;

   (b) the overlay network topology will remain hidden to all nodes; and

   (c) the path of any individual route will not be disclosed.

   Fig. 4 illustrates the path of sending data from a sender with overlay address OA_5 to a receiver with overlay address OA_3. As we can see, there are two paths. By running the anonymous routing protocol, the sender finds that the "best" way for delivering data to the receiver OA_3 is by forwarding the data to the node with direct virtual link which has IP address IP_4 rather than to the other directly-linked (virtually) node with IP address IP_1.

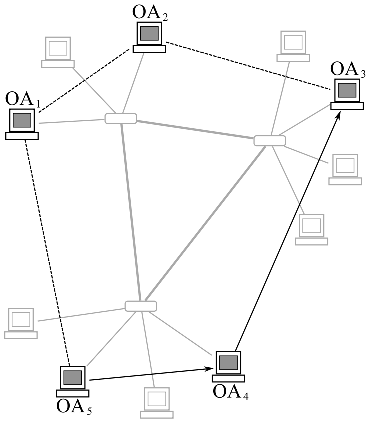


Fig. 4. The routing table in the node OA$_5$ indicates that routing data via the node with IP address IP$_4$ is better then going through another virtual neighbor which has IP address IP$_1$.

The "best" way here refers to the routing objectives of the anonymous routing protocol, for example, achieving low latency and high bandwidth.

Unlike the existing circuit-based anonymous networks, this new network is packet-based and the reverse path from the node with overlay address OA$_3$ back to the node with overlay address OA$_5$ may go through a different path, which is determined by the routing table of the node OA$_3$ and that of other intermediate nodes along the reverse path.

Also specified above, the anonymous routing protocol ensures that no information about the relationship between the real IP address and the set of virtual overlay addresses of a node is leaked. In the example illustrated in Fig. 4, this indicates that the node OA$_5$ only knows that the real IP addresses of its two virtual neighbors are IP$_1$ and IP$_4$. It has no idea on their virtual overlay addresses, namely OA$_1$ and OA$_4$. The distributed nature of the anonymous routing protocol helps to ensure that no one can discover the topology of the overlay network and each node only knows its virtual neighbors' real IP addresses but not their overlay addresses.

5. **Services**: After a routing table has created, a new virtual node can then communicate with all other virtual nodes on the overlay network. In their communication, they specify each other using their virtual overlay addresses.

   Fig. 5 illustrates how data are transferred from a sender with overlay address OA$_5$ to a receiver with overlay address OA$_3$. The intermediate node has IP address IP$_4$ and overlay address OA$_4$. As mentioned, the anonymous routing protocol ensures that the sender (OA$_5$) only knows that the "best" way to reach the receiver (OA$_3$) is to forward data to its virtual neighbor with IP address IP$_4$ rather than to its other virtual neighbor with IP address IP$_1$ (the virtual neighbors' IP addresses are known to the sender as it chooses them in Step 2 **Connection Establishment**). However, the sender has no idea that the virtual neighbor with IP address IP$_4$ corresponds to the overlay address OA$_4$. Similarly, the intermediate node with IP address IP$_4$ knows (thanks to the anonymous routing protocol) that the "best" way to reach the receiver (OA$_3$) is to forward the data to its virtual neighbor with IP address IP$_3$. It has no idea that this virtual neighbor is the actual receiver with the overlay address OA$_3$. To elaborate further, the intermediate node also has no idea that the node with IP address IP$_5$ which forwards the data to it is actually the sender with overlay address OA$_5$.

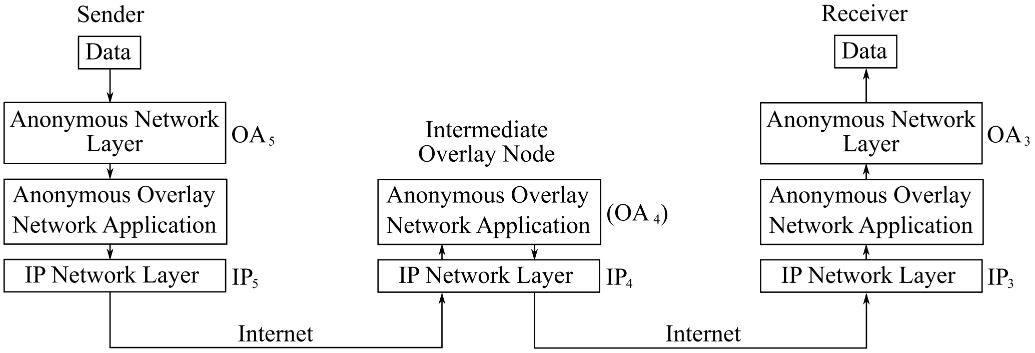


Fig. 5. Interaction between the Anonymous Network Layer and IP Network Layer when data is sent from a sender to a receiver via an intermediate node.

In a nutshell, the anonymous routing protocol (Sec. 4) can ensure that the real IP address of a physical node is unlinkable to any of its virtual overlay addresses.

Apart from the encryption of the connection between individual hops (K_{51} and K_{54} in Fig. 3), communication between any two nodes on the overlay can also be encrypted. This end-to-end encryption prevents intermediate nodes that are routing traffic within the overlay network from learning the content of a communication. In this paper, we would not discuss any concrete implementation of such an end-to-end encryption.

This completes the overview of our anonymous network. In the following, we discuss certain aspects of this new anonymous network in more detail. This is followed by a detailed description of our anonymous routing protocol in Sec. 4.

3.3 Overlay Network Namespace and Dissemination

As the overlay addresses for this new anonymous network, we propose to use IPv6 for a number of reasons. Besides the fact that many Internet applications already support IPv6 and most operating systems (Windows, Mac OS X, Linux, UNIX, *BSD) already have a dual protocol stack which makes access to the anonymous network transparent to most applications. Another major reason is that each IPv6 address is 128 bits long, which makes the chance of having a collision when the addresses are randomly generated negligible (in practice a little bit less than 128 bits would be used as to prevent collisions with the already assigned IPv6 address space). According to the birthday paradox, the expected number of overlay network nodes has to reach about 2^{64} for having a collision on the randomly generated addresses. In other words, Step 3 **Overlay Address** above is expected to be efficient in practice as it is expected that each node can generate overlay addresses which are unused on the first try.

3.4 Anycast Proxies

The ultimate goal of setting up such an anonymous network is to have the network be self-sustainable in the sense that all services are being offered within the network. However, it is also very important to allow users to use this anonymous network for accessing services that are on the Internet at large anonymously.

For this purpose, we propose to use *anycast proxies* [1] to enable anonymous access to normal Internet services from within the overlay network. To do so, a node inside the overlay network will send traffic destined for a node (let us consider it to be a web server) on the normal Internet to a pre-defined anycast overlay address within the overlay network. Taking the conventional anycast mechanism, the traffic will be routed to the “nearest” (due to the effect of running the anonymous routing protocol described in Sec. 4) proxy available.

About the downside of using anycast proxies, we notice that it needs reconfiguration of web browsers for example.

3.5 Simulation Results

In this section, we describe the simulation results we obtained by simulating three regions of a total of 1,280 nodes on the overlay network, in which 512 nodes are in America, 256 in Asia and 512 in Europe. Due to the page limitation, we refer readers to Appendix A for detailed description of our simulation settings and the experiments we have carried out. Below is the summary of our simulation results.

For a connection made between any two nodes, the Round-Trip-Time (RTT) we simulated is mostly below one second. This is the most significant and indicative improvement when compared with the latencies of existing solutions (Sec. 1). In addition, connections between nodes in the same region range from 200 ms to 300 ms RTT while inter-regional connections are in the order of 600 ms to 800 ms. All these results are several times better than existing solutions.

The end-to-end latency when using anycast proxies goes from 200 ms to 400 ms RTT for intra-regional connections. It goes up to 600 ms to 800 ms RTT when two regions are involved and to between 800 ms and $1,200$ ms RTT when three regions are involved. For the latter case where three regions are involved, this is the case where the sender is in one region, the anycast proxy is in another region, and the destination is in the third region.

A direct comparison with currently used systems for anonymity is not straightforward, but according to the numbers determined by Wendolsky et al. [15], these systems have an initial delay when accessing a webpage of between at least $1,000$ ms and $7,000$ ms with a mean of anywhere between $1,200$ ms to $4,700$ ms depending on the system. Some systems also incur occasional additional delays (i.e. circuit establishment in Tor).

Remark: These numbers do not include any queueing delay, which can be a problem for bandwidth-constrained nodes. But compared to other systems with a list of core routers used by all clients, in our network every client provides part of the aggregate bandwidth. Still, more simulations or experiments are needed to determine delays under constrained bandwidth. We consider this as one of our future work.

4 Anonymous Routing

In this section, we propose an anonymous routing protocol which is suitable for use in Step 4 **Routing** as described in Sec. 3.2. As mentioned, the anonymous routing protocol should ensure that

1. the relationship between the real IP address of a physical node and the set of virtual overlay addresses of the node cannot be discovered by any other node (for simplicity, consider that each physical node corresponds to one virtual overlay node);

2. the overlay network topology will remain hidden from all nodes; and
3. the path of any individual route will not be disclosed.

Before describing our new routing protocol, it is worth mentioning that traditional routing protocols cannot actively ensure the three objectives above. On the Internet, the inter-domain topology is freely advertised as part of the normal routing process and can easily be obtained from public sources. This enables an attacker to join the network and discover large parts of the network topology by participating in the routing protocol. Therefore, existing routing protocols cannot be applied directly to the virtual overlay network.

4.1 Anonymous Pseudo Path Vector Routing

The anonymous routing protocol we propose here can be considered as an anonymous version of path vector routing. Path vector routing is a variant of distance vector routing which solves the problem of routing loops and counting to infinity by remembering the path-so-far of every route (distance vector routing only keeps the aggregate distance, but not the exact path). Whenever a node receives a route announcement where it is already in the path, the route forms a loop and can be discarded. The best known path vector routing protocol is BGP-4 [13] which is the routing protocol used for inter-AS routing on the Internet.

In a network where normal path vector routing is used, every node participating in the routing can build a fairly detailed map of the network just by using the routing information received from neighbors. The routing protocol we propose here hides the path information so that no one can learn the topology of the network through participating the routing process. Figure 6 shows the structure of a route announcement in the proposed protocol.

Destination	**Link Quality**	**Pseudo Path (PSP)**

Fig. 6. The structure of a routing entry in the anonymous pseudo path vector routing protocol.

As shown in the figure, there are three component in a route announcement. **Destination** indicates the overlay address of the destination of the route. **Link Quality** contains one or several types of measures indicating the quality of the route. Possible measures include latency and bandwidth. The link quality is *aggregated* along the route. When a node receives a route announcement, the link quality indicates the quality along the whole path from the current node to the destination. **Pseudo Path** (PSP) replaces the normal path in conventional path vector routing. This pseudo path will still allow nodes along a routing path to detect routing loops. But unlike the original path vector routing protocol, this pseudo path version can prevent any node from discovering the actual path. Below are the details of the pseudo path.

Pseudo Path. The Pseudo Path (PSP) component in a route announcement is a fixed-length random looking binary string which consists of N segments, each to be L bits long. For example, N could be 20 and L 128 bits. We will see shortly on how these values are set in practice.

The purpose of the PSP is to detect routing loops. It does not leak any information about the routing path such as the node's addresses (both overlay addresses and real IP addresses) or the number of hops along the path. As we will see shortly, each random-looking segment in the PSP component contains an L-bit pseudo-random binary string which is constructed so that no one can correlate it to any useful information about the path. In the following, we describe the PSP component by describing how it is created and updated while propagating from one node to another.

In the anonymous network, each node i randomly generates a k-bit secret symmetric key K_i (e.g. $k = 128$), a random string R_i of l_R bits long (e.g. $l_R = 80$) and a counter C_i of size l_C bits (e.g. $l_C = 48$) with an arbitrary initial value (*Note*: $L = l_R + l_C$). (K_i, R_i, C_i) are all kept secret and will not be shared with any other node.

When a new route announcement is created in the form shown in Fig. 6, the PSP component is initialized with random bits ($L \times N$ bits long in total). This is followed by replacing the last segment (i.e. the rightmost segment of the PSP component) with x_i where

$$x_i \leftarrow E(K_i, R_i \| C_i). \tag{1}$$

$E : \{0,1\}^k \times \{0,1\}^L \rightarrow \{0,1\}^L$ is a symmetric encryption algorithm (e.g. AES) and x_i is generated by encrypting "message" $(R_i \| C_i)$ under key K_i, in which the symbol '$\|$' denotes string concatenation.

After filling the last segment with x_i, the PSP component is rotated to the right so that the segment with x_i becomes the most significant segment (i.e. the leftmost segment of the PSP component). The two operations described above are illustrated in Fig. 7.

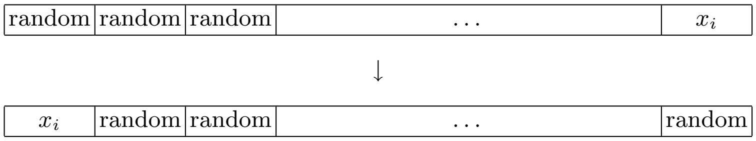


Fig. 7. Change on the PSP component after replacing the last segment (upper diagram) and after rotation (lower diagram).

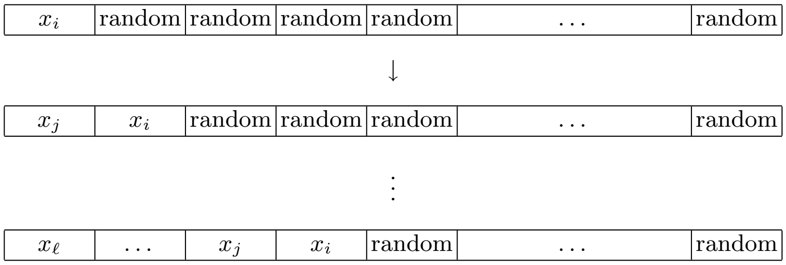
Note that under the assumption that the symmetric encryption algorithm is a pseudo-random function family [9], it is infeasible for any other node to distinguish x_i from any other randomly generated segment on the PSP component.

After these two operations above, the route announcement is sent out to all the neighbors of node i. Note that the neighbors of node i are defined in Step 2 **Connection Establishment** described in Sec. 3.2.

After sending the announcement, node i updates the counter C_i by increasing its value by one. The value of R_i remains static.

When one of the neighbors, say node j, receives the route announcement, it checks for routing loops using the method described below under the part called **Loop Detection**. If there is no routing loop, the node then updates its routing table for the entry of the overlay address specified in the **Destination** component if the measures specified in the (accumulated) **Link Quality** component of the route announcement are better than the current ones in its routing table. Better measures may refer to lower accumulated latency and higher minimum bandwidth along the route.

If the entry in its routing table is updated, that is, a more efficient route to the destination specified in the route announcement is found, node j needs to update the route announcement and forward it to all its neighbors except node i. To update the route announcement, node j first computes the updated measures for the **Link Quality** component. Then, equivalently to Eq. (1) and the two operations described above, it computes $x_j \leftarrow E(K_j, R_j \| C_j)$, replaces the last segment of the PSP component and then rotates the PSP component to the right. The propagation then continues as illustrated in Fig. 8.



x_i | random | random | random | random | ... | random
↓
x_j | x_i | random | random | random | ... | random
⋮
x_ℓ | ... | x_j | x_i | random | ... | random

Fig. 8. The PSP component is updated by each node along the route.

Loop Detection. Suppose node i receives a route announcement from one of its neighbors. To detect if there is a routing loop, node i uses its symmetric key K_i to decrypt every segment in the PSP component. For each decrypted binary string, set the first l_R bits as R and the remaining l_C bits as C. If $\mathsf{R} \equiv R_i$ and C is strictly smaller than the current value of C_i, then the node i will conclude that a routing loop occurs and the announcement will be discarded.

The motivation for using the structure with K_i, R_i and C_i is that the node i can use them to generate random-looking strings $x_i \leftarrow E(K, R_i \| C_i)$ and only needs to remember these three values for being able to recognize any segment on a PSP component created by itself.

Remark: A node after receiving a route announcement which has traveled in a loop can determine the number of hops in the loop by looking at the position of its own identifier (i.e. the segment) on the PSP component. This does not undermine the anonymity of the path towards the specified destination because no other information other than the number of hops in the loop can be obtained by the node.

Pseudo Path Size. From Fig. 8, it is obvious that N is the maximum limit for the number of nodes on each route. The value of N has to be pre-determined before running the protocol and it depends on the structure of the overlay network. At the same time, as the routing optimizes performance, is is likely that the longer a route gets, the more probable that there is a shorter route available, so the value for N could be chosen reasonably small. With reference to BGP-4 [13] used for inter-AS routing on the Internet, setting N to 20 should be a good starting point.

The size l_R of the random component R_i and the size l_C of the counter C_i add up to L bits. If L is of size 128 bits, a possible division is 80 bits for l_R and 48 bits for l_C. This allows up to 2^{48} route announcements for each node before the counter wraps around. When the counter wraps around, the node will generate a new random string R'_i and a new counter C'_i and then starts using these new values while still remembering the old values for a period of time.

4.2 Security Analysis and Open Problems

Anonymity. We can see that the anonymous routing protocol above can hide the network topology while still detecting routing loops. In addition, individual nodes do not even know the overlay addresses of their virtual neighbors. In the following, we discuss two specific scenarios where attacks against the anonymity of an individual node can be mitigated.

Initial route announcement: A node may be able to infer the overlay address(es) of one of its virtual neighbors by monitoring the routing updates being sent from a newly connected node and then compare these updates to the current routing table. If the route did not exist before, it is possible that it points to the newly connected node. This problem is common to many existing anonymous networks and generally be mitigated through frequent joining of new nodes, randomizing the route announcement time and slightly modifying the link quality of certain routes when generating new announcement from time to time. Furthermore, an existing node can also do frequent updates of their overlay address(es).

Link Quality: A similar problem is related to the link quality. If the exact latency is included in the link quality, as an example, a node may make use of it to determine how many hops away the destination is from the node. For defending against this type of attacks, we can introduce a certain amount of randomization to distort the real values in the Link Quality component of each route announcement. This conventional method can be applied directly to our anonymous routing protocol.

Route Authentication. The protocol as presented here does not include any provisions for authenticating or verifying route announcements. A malicious node can arbitrarily create false route announcements or alter received route announcements maliciously. For example, a malicious node may tamper with the PSP component of a route announcement so that routing loops cannot be detected. This problem is similar to the current BGP used on the Internet. We leave this and similar type of authentication problems as our future work.

5 Concluding Remarks

In this paper we proposed a new anonymous network in the form of an overlay and an anonymous routing protocol. The overlay network with a new and independent namespace (i.e. overlay addresses) which separates the overlay network from the non-anonymous Internet. The anonymous routing protocol helps improve the performance of the overlay network by facilitating each node to find an anonymous path to reach any destination with low latency and high bandwidth. In addition, it hides the overlay network topology, delinks any relationship between the real IP address of a physical node and the set of virtual overlay addresses of the node and conceals the path of individual routes.

Our simulation results also showed that the expected latency of the anonymous network is cut short by 50% from that of existing systems. Furthermore, we also discussed and proposed solutions for several important implementation issues, which include the overlay namespaceand anycast servers for anonymous access to normal Internet services.

We consider this paper to be the first, introductory and conceptual one for proposing the notion of a new anonymous network. Future work includes evaluating the performance of this new network in terms of bandwidth, looking into the performance of the anonymous routing protocol under highly dynamic situations (e.g. lots of nodes continuously joining and leaving the network) and solving the problems mentioned in Sec. 4.2.

References

1. J. Abley and K. Lindqvist. Operation of anycast services. RFC 4786 (Best Current Practice), Dec. 2006. (Cited on page 8.)
2. S. Blake-Wilson and A. Menezes. Authenticated Diffie-Hellman key agreement protocols. *Lecture Notes in Computer Science*, 1556:339–361, 1999. (Cited on page 4.)
3. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications ACM*, 24(2):84–90, 1981. (Cited on pages 1 and 2.)
4. D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988. (Cited on pages 1 and 2.)
5. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. (Cited on page 4.)
6. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–320, 2004. (Cited on pages 1 and 2.)
7. M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 193–206, 2002. (Cited on pages 1, 2, and 3.)
8. S. Goel, M. Robson, M. Polte, and E. Sirer. Herbivore: A scalable and efficient protocol for anonymous communication. *Technical Report TR2003-1890, Cornell University Computing and Information Science*, Feb. 2003. (Cited on pages 1 and 3.)
9. O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001. (Cited on page 10.)
10. D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. *Communications ACM*, 42:39–41, 1999. (Cited on pages 1 and 2.)
11. I2P anonymous network. Online. http://www.i2p2.de/. (Cited on page 1.)
12. A. Panchenko, L. Pimenidis, and J. Renner. Performance analysis of anonymous communication channels provided by Tor. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, pages 221–228, 2008. (Cited on page 2.)
13. Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), Jan. 2006. (Cited on pages 9 and 11.)
14. University of Regensburg. JAP anonymity & privacy. Online. http://anon.inf.tu-dresden.de/. (Cited on page 1.)
15. R. Wendolsky, D. Herrmann, and H. Federrath. Performance comparison of low-latency anonymisation services from a user perspective. *Lecture Notes in Computer Science*, 4776:233–253, 2007. (Cited on pages 1 and 8.)
16. L. Zhuang, Z. Feng, B. Y. Zhao, and A. Rowstron. Cashmere: Resilient anonymous routing. In *NSDI 05: 2nd Symposium on Networked Systems Design & Implementation*, pages 301–314, 2005. (Cited on pages 1 and 2.)

A Detailed Simulation Results

In our simulation, we model the Internet as three regions (one can imagine them as three continents, namely America, Asia, Europe). In each region, there is a core router, to which regional nodes are connected in a star pattern. Then, inter-regional links connect the regional core-routers between the different regions. The topology is shown in Fig. 9.

The latencies on different links are modeled after latencies found in the real Internet, although simplified by assuming them to be fixed and the same for each node in a region. Regional latency for the link between a node and a core-router is 25 ms for America and Asia and 15 ms for Europe. The round-trip time (RTT) between nodes within a region is thus 100 ms for America and Asia and 60 ms for Europe, respectively. Inter-regional links between the core routers have a latency of 90 ms for America - Asia, 40 ms for America - Europe and 150 ms for Europe - Asia.

The number of nodes we simulated is $1,280$ in total, with the nodes distributed as follows: 512 in America, 256 in Asia and 512 in Europe.

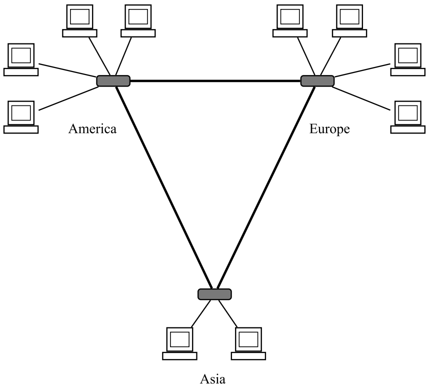


Fig. 9. A simplified model of Internet used in the simulator.

To create our proposed overlay network over the modeled Internet described above, we let each node connect to a number of randomly chosen nodes. By varying the number of outgoing connections of a node, we could observe the latency characteristics of the overlay network as a whole. As expected, the latency increased with more nodes if the number of connections was constant and decreased with more connections if the number of nodes was constant. This is illustrated in Fig. 10 which shows the one-way latency between nodes on the overlay network for the simulated case of Europe. From the figure, we can see that up to 4 outgoing connections for each node, there is a significant improvement of the latency; above 4 connections, the improvement is smaller with each additional connection.

Results on RTT of intra-regional communications are shown in Table 1.

Table 1. RTT for intra-regional communications

America	323 ms
Asia	288 ms
Europe	192 ms

Table 2 shows the RTT for inter-regional communications.

Table 2. RTT for inter-regional communications

America - Asia	795 ms
America - Europe	647 ms
Asia - Europe	751 ms

Simulating the latency of accessing normal Internet services through anycast proxies leads to the results shown in Table 3 where the node, the anycast proxy and the Internet service are located in the same region. For the case that the node is located in one region and the anycast proxy together with the Internet service are in another region, the results in Table 4 apply.

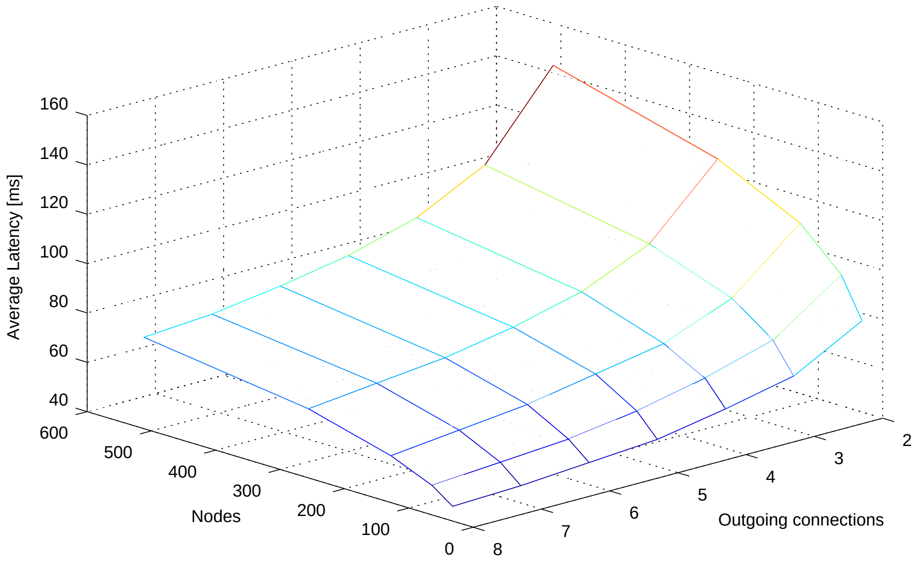


Fig. 10. One-way latency between nodes within the same region (in this case Europe).

Table 3. RTT for accessing an Internet service through an anycast proxy in the same region

America	373 ms
Asia	338 ms
Europe	222 ms

Table 4. RTT for accessing an Internet service through an anycast proxy where the node is in one region, and the anycast proxy and the Internet service are in another region

America - Asia	845 ms
America - Europe	677 - 697 ms
Asia - Europe	781 - 801 ms

For the case when the node, the anycast proxy and the Internet service are all in different regions, RTT values vary from around 800 ms up to $1,200$ ms.