Analysis of Routing Algorithms for Secure Overlay Service

Debra Cook
Columbia University
dcook@cs.columbia.edu

Abstract

The routing of packets through an overlay network designed to limit DDoS attacks is analyzed. The overlay network structure includes special purpose nodes which affect the routes taken through the overlay. Two main factors are considered: the routing algorithm utilized for the overlay and the method for selecting the special purpose nodes. The routing algorithms considered are based on methods originally defined for peer-to-peer services. A model was developed for the overlay network which allowed altering the routing algorithm, method for selection of special purpose nodes and the underlying ISP structure. The model was used to assess the impact of specific routing algorithms and selection methods on latency and path length. The implications of utilizing a specific method for node selection on the probability of a successful DDoS attack is briefly discussed.

1. Introduction

A method of utilizing an overlay network to reduce the likelihood of a denial of service attack is described in *Secure Overlay Service* (SOS) [1]. The method utilizes special purpose nodes and filtering to reduce the probability of an attack successfully blocking traffic to a specific destination. SOS alters the route to the destination by requiring packets pass through specific nodes, which may be dynamically selected, in order be permitted access to the destination. As a result, the route a packet traverses to the destination may be significantly different than the route normally taken through ISP(s). This paper summarizes the impacts of two algorithms for routing packets through the overlay network in accordance with the requirements imposed by SOS. These are combined with rules for selecting the special purpose nodes. Specifically, the change in expected latency and number of cities through which the route passes as compared to the normal route are presented.

2. Background

2.1 Terminology

The following terminology will be used:

- Access Point: A node which provides access into the overlay network.
- Beacon: A node in the overlay which receives traffic destined for a target and forwards the traffic to a servlet for the target.
- Client: A customer of the ISP. The term is used to refer to LANs connected to the ISP.
- Normal Route: A route which is based only on the destination address and thus can follow a route used by regular traffic through an ISP. Shortest path based on hops is used in the model.
- Overlay network: In the context of this paper, the term will refer to an overlay which contains the beacons, servlets and access points as required by SOS. The nodes may be within ISP(s) or client nodes.
- Overlay Route: A route that requires sending to a packet to one of more overlay nodes in order to reach a node which has information about the destination and can then use a normal route to the intermediate or final destination.
- POP: Point of Presence. The term is used to refer to the main routing hubs in an ISP network. Two nodes will be referred to as being served by or on the same POP if they are clients that connect into routers within the POP.
- Servlet: A node in the overlay which provides access to the target.
- SOS: Secure Overlay Service. SOS route will be used to refer to the route a packet travels when SOS is applied.
- Source: A node from which legitimate traffic to a target originates.
- Special purpose nodes: This will refer collectively to access points, servlets and beacons.
- Target: A destination node that may be subject to an attack.

2.2 SOS

The goal of SOS is to prevent denial of service attacks via a combination of traditional packet filtering for security and the use of an overlay network to provide a level of concealment. Two types of special purpose nodes, servlets and beacons, are utilized by SOS for routing packets to a destination. Given a set of overlay nodes, the target for a particular service selects one or more overlay nodes to act as servlets. Each servlet then selects one or more overlay nodes to act as beacons. All packets destined for the target must pass through a servlet. Only the beacons know which node(s) act as servlets for a given target, thus all packets destined for the target must pass through the appropriate beacon to find the servlet. Packets from a source destined for the target are required to route from source to an access point to enter the overlay. The packet routes through the overlay until a beacon is reached which can direct it to the servlet for the target. Upon reaching the servlet, the packet will be forwarded to the target. If a node leaves the overlay, another overlay node will assume its responsibilities. The target node may at any point in time change the nodes acting as its servlets and a servlet may change the nodes acting as its beacons.

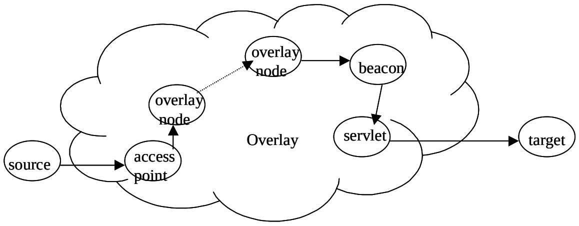


Figure 1: SOS Architecture

2.3 Routing Algorithms

Before describing the model, overviews of two peer to peer routing algorithms are provided as background material for understanding the routing algorithms utilized in the overlay network. The first is CHORD [3], which [1] suggests using as the mechanism for overlay routing. The 2^{nd} is Content Addressable Networks (CAN) [2], of which a variation is used in the model as an alternative to CHORD. The following describes the general routing algorithms for CHORD and CAN. Refer to [3] and [2] for details regarding how the overlay is maintained when nodes leave or join it.

2.3.1 CHORD

In CHORD, each node is assigned an id via a hash function. The nodes in the overlay are ordered by their ids. Conceptually this can be viewed as a circle.

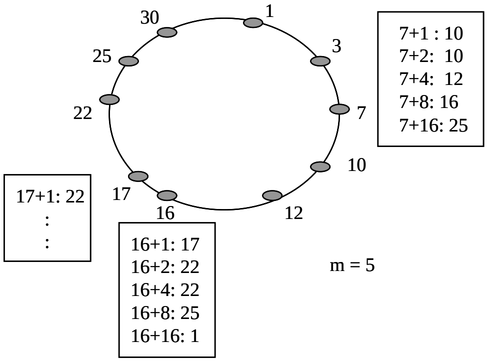


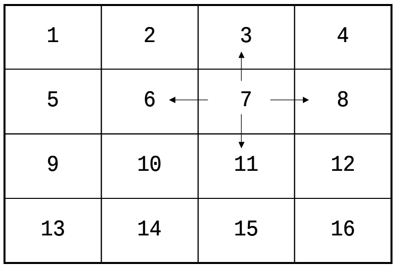
Figure 2: CHORD Overlay Routing

Let 2^m be the number of possible ids. Let x denote the id of an overlay node. Each overlay node contains a table of m entries , the i^{th} entry is the 1^{st} overlay node whose id is $\geq x + 2^{i-1}$ (mod 2^m), as shown in the diagram. For a node not in the overlay whose is assigned id y by the hash function, the overlay node whose id is closest to but ≥ y will store information about y, such as z's IP address. If overlay node x receives a packet destined for node y, it forwards the packet to the overlay node in its table whose id is closest to, but

≤ y’s id. In the example, if node 7 receives a packet for node 20, the packet will route from 7 to 16 to 17, When the packet reaches node 17, the next node in the overlay is 22. Node 17 checks that it does not have the information for node 20 and knows that 22 is the next node in the overlay, thus 22 must contain node 20’s information. The CHORD algorithm results in packets being routed around the overlay “circle”, while progressively getting closer to the desired overlay node. O(log n) overlay nodes may be visited, where n is the size of the overlay. Notice that there is no rule for determining the relationship between overlay node ids. Two nodes with consecutive ids could be any geographical distance from each other.

2.3.2 CAN

CAN maps objects using a hash function to a point in the coordinate space. Each overlay node is responsible for a section of the space. After determining the point to which an object is mapped, the overlay node responsible for the region containing the point is used to store the object’s information. For the purpose of this paper, consider the object to be a node and the information to be its IP address. Each overlay node contains a table of overlay nodes responsible for neighboring areas in the coordinate space. As shown in Figure 3, overlay node 7 would contain pointers to nodes 3,6,8 and 11. In its basic form CAN does not assume any relationship between node positions of the coordinate space and their geographical positions in the real world. A variation suggested in [2] regarding the coordinate space being representative of the geography provided the basis for the heuristic used in the model.



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Figure 3: Overlay Nodes Serving Regions of a Coordinate Space.

3. Model

3.1 Purpose

In order to understand the impacts of the overlay network on the routing of packets between the source and target nodes, the SOS algorithm was applied to two models of ISP networks. One model was for the United States based on AT&T’s network and one was for Europe based on Worldcom’s network. These are shown in Figures 5 and 6. Refer to [4] for additional maps of ISP networks. While the US model was based on AT&T’s network, remote access points were excluded from the model. Connections from Worldcom’s European POPs to points outside the area shown in 6 were excluded from the Europe model. For each model, two algorithms for routing traffic through the overlay were tested, one based on CHORD, which uses a random ordering of the overlay nodes, and a heuristic based on the variation of CAN using a geographical ordering of the overlay nodes. In all cases, variations on how the beacons and servlets were chosen in relation to each other, to the target and to the source were tested; for example, requiring some minimum distance between the servlet and target.

The purpose for constructing the model was to determine the affects of the algorithm on the number of hops, distance and latency. The effects of increased traffic generated by an attacker on the route were not included because it was assumed that the links between ISPs pops are utilized at less than capacity such that the choke points in any route using an overlay are points outside of the ISPs main backbone. For example, if the ISP has average utilization of 60% to 80% on OC192 links, the attacker would not be able to consume the bandwidth between two of the ISP's POPs. The model was also not required to determine the probability of a successful attack. This is the probability that the attacker determined either the target's location and/or all access point locations and/or all overlay nodes and was able to take them out of service. In [1], it is assumed filtering around the target in high bandwidth areas prevents unauthorized traffic from reaching the target and consuming the entire bandwidth on the links to the target. Therefore, a successful attack must disable specific overlay nodes, such as the access points, servlets and/or beacons. Consider a case where there are 100K clients eligible to be overlay nodes and 1000 are used in the overlay. Also assume the target selects two servlets and there are more than two access points. The attacker would have to take at least two nodes, the servlets in this case, out of service and be able to do so repeatedly since the target will automatically select new servlets. If the attacker knows which nodes participate in the overlay, the probability of correctly guessing the servlets once for a specific target is < 0.001%. If the overlay nodes are not known, this becomes < 10^{-9} %. Thus the probability of locating the appropriate attack points is negligible.

3.2 Network Layout

A representation of the ISP at the POP level was created. Each POP was assumed to consist of a hierarchy of routers as shown in Figure 4. At the top level are routers with links to other POPs. At the lowest level are links to client LANs.

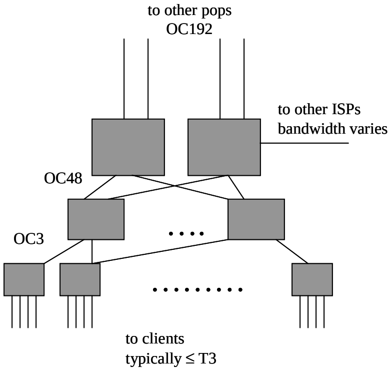


Figure 4: ISP POP Structure in Model

Latencies between POPs were estimated from a subset of known latencies, assuming adequate bandwidth is available. Distance between POPs were based on airline miles due to facility miles not being available. Assuming the facility miles between POPs are proportional to the airline miles, estimating the distances and latencies do not affect the final results, which reflect the ratio of the latency when using SOS to that of the normal route. The estimates did not impact the number of hops in a route. Three routers were included at the second level and twelve at the lowest level of each POP; however, for the statistics computed, the exact number of routers within a POP was not relevant, only the latency from the time a packet entered a router within a POP to the time it left the POP was needed.

While the model assumed adequate bandwidth between POPs and that the choke points would be the links to clients, for future use, the model included the bandwidth between all routers. The bandwidth between POPs was OC192's for the United States model and varied in the Europe model from OC3s to OC192s. T1s and T3s were used for links between the ISPs and clients. For the purposes of the model, the average distance and latency to a client from an access point in a POP was needed as opposed to individual latencies for very client; therefore, all latencies and distances to clients were assumed to be the same.

There were 19 POPs in the US model and 18 in the Europe model. The same structure within a POP was utilized in both models. The overlay nodes were client nodes in all cases. Refer to Section 4 *Other Considerations* for a discussion on the impact of moving these into the ISP POPs.

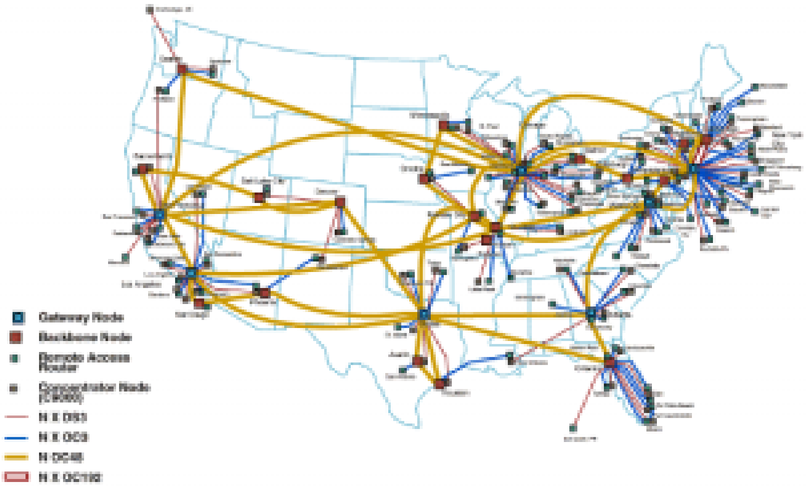


Figure 5: AT&T US Network [1]

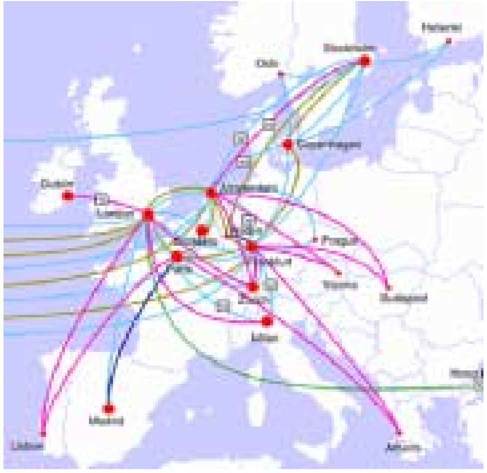


Figure 6: Worldcom's European Network[2]

The eligible overlay nodes were evenly distributed across POPs, meaning each POP served the same number of client nodes eligible to be overlay nodes. In the cases where servlets and beacons were randomly chosen, this allowed each POP to be equally likely to have a client site that was a servlet or beacon. In the cases where the servlet and beacon nodes were not randomly chosen, there were more eligible nodes per POP than utilized and the even distribution did not impact selection. A node was not allowed to serve more than one purpose for a specific source-target pair, for example, a node could not be both a beacon and a servlet for the same target. Removing the restriction would result in shorter routes on average because some scenarios tested would pick the same node for both the servlet and beacon.

In each case, two client nodes served by each POP were included in the overlay. Since each source – target pair was tested individually, at most two nodes per POP would be selected to serve the functions of beacon and servlet. When ordering the overlay nodes according to the geographic heuristic described below, designating more than two nodes per POP could only change a route between a source and target by possibly passing through a different client on a given POP. When ordering the overlay nodes randomly and using CHORD as the routing algorithm for the overlay, the probability that a client on a specific POP was

[1] Map is cropped image from http://www.cybergeography.org/atlas/more_isp_maps.html
[2] Map is cropped image from http://www1.worldcom.com/global/about/network/maps/europe/

picked as a beacon or servlet, or was at a certain position in the overlay impacted the route. Since it was assumed overlay eligible nodes were evenly distributed across all POPs, having 2 versus 100 overlay nodes per POP would not impact the probabilities and thus would not affect the results. The access point for the source was chosen to be a client on the same POP as the source. The impact due to it being served by a different POP than the source would be to add the cost of the normal route between the source and access point to the cost of the route between the access point and target.

3.3 Routing Algorithms

In SOS, traffic from a source to a target utilizes a route which contains the following sequence of nodes in order: source, access point, beacon, servlet and target. Refer to Figure 1. Normal routing is used to reach the access point. Also, since the beacon knows the specific servlet for the target, and the servlet knows the location of the target, normal routing is used between the beacon and servlet, and between the servlet and target. An overlay route is used between the access point and beacon. The increase in the route length over that of the normal route between the source and target is due not only to the requirement that the route pass through specific nodes, but also due to the need to route through an overlay network between the access point and beacon as opposed to using the normal route between the two nodes. For normal routing, each node in the model contained a routing table populated via Dijsktra's algorithm, using minimum hops as the criteria for shortest path. Each node in the overlay network also contained a table with the destination address and overlay node id of a subset of overlay nodes. The table was populated based on the routing algorithms described below.

A routing algorithm for use in overlays is required to send traffic between the access point and beacon. The CHORD algorithm was utilized in the first set of experiments. The overlay nodes where randomly ordered. The tables within each overlay node were populated using the method described above involving powers of 2. The size of a node's table is O(log n) where n is the size of the overlay.

The second set of experiments used a heuristic which divided the POPs into geographical areas. This method is based on modifications suggested to the basic algorithm for CAN. [3] For a specific area, A, a node n_A was chosen as the area's representative. Each n_A was an entry in each overlay node's table. In addition, if n_i is an overlay node in area A, n_i's table would include entries for each n_j in A, $i \neq j$. Thus an overlay node maintained pointers to every other overlay node in the same geographical area and to one overlay node in each other geographical area. For an overlay of size n, the size of a node's table is O(n/5) + # of areas, which is O(n/5) when n is large compared to the number of areas. The US model involved 6 areas, one contained 2 POPs and the other contained 3 or 4 POPs each. The Europe model contained 4 areas with 4 to 5 POPs each.

3.4 Scenarios

Seven source-target pairs were chosen in each of the two models. They were selected to represent a variation in source-target relations. Factors considered when selecting the pairs included the distance between cities, whether they were served by neighboring POPs and the level of connectivity for the POP. In all cases a servlet and beacon for a specific target were not permitted to be the same node and neither could serve as an access point.

For each model and each routing algorithm, the normal route between each source-target pair was computed then the following eight scenarios were tested on each pair. In the scenarios, minimizing the number of hops refers to the number of hops as calculated by normal routing.

1. Randomly select the servlet and beacon. (100 trials per source-target pair were run)
2. Select the servlet to minimize the number of hops between the servlet and target, then select the beacon to minimize the number of hops between the beacon and servlet, with the restriction that the servlet and beacon not be served by the same POP.

[3] The heuristic used in the model was derived from two variations suggested for CAN in [2]. The use of geographic areas is based on a variation to incorporate geographic relations between nodes when partitioning the coordinate space. Due to the small number of areas in the model, the nodes are manually assigned to an area as opposed to building an algorithm to determine nodes in the same area as suggested in [2]. Nodes maintaining pointers to all other nodes in an area is based on the overloading of zones variation except that the zones will never be split. Also in [2], a node will only have a pointer to a node in another area if the areas of the two nodes border each other; whereas, in the heuristic, a node will have a pointer to one node in each other area. This is feasible due to the small number of areas in the model.

3. Select the servlet to minimize the number of hops between the servlet and target, then select the beacon to minimize the number of hops between the beacon and source.
4. Select a servlet randomly from those approximately X miles from the target then select a beacon randomly from those approximately X miles from the servlet, where X was 1000 in the US model and 500 in the Europe model. In the case of the Europe model, a few POPs did not have neighbors within this distance, in which case the next closest available overlay node was used.

The first scenario was used to obtain an understanding of the impact when no selection criteria was utilized for the servlet and beacon. This would be the simplest version to implement. The second and third scenarios were aimed at keeping the intermediate nodes in the route near the end points to determine if the route between the source and target would then be similar to the normal route. These two scenarios using minimum distance instead if hops were tested on the US version, but the results were not noticeably different from the scenarios using hops. The fourth scenario was used to understand the impact of selecting the servlet and beacon so they would be served by different POPs than the target, which may be desired for diversity, but at the same time guaranteeing they would be relatively close in an attempt to avoid an unnecessarily long route.

3.5 Results

Table 1 shows the ratio of the latency using SOS to the latency expected when using normal routing. The scenario number corresponds to the above list. These were averaged over all source-target pairs. The worst case from all source-target pairs is shown in Table 2. Table 3 indicates the increase in the number of ISP POPs involved in a route compared to that of the normal route.

model and overlay routing method	US		Europe	
	CHORD	Geographical	CHORD	Geographical
scenario				
1 random selection	4.51	4.16	5.69	4.11
2 min hops	3.45	2.4	3.25	2.54
3 min hops	7.19	1.75	6.77	1.74
4 diversity	5.18	4.08	5.6	2.88

Table 1
Average Ratio: Latency with SOS vs Normal Route

model and overlay routing method	US		Europe	
	CHORD	Geographical	CHORD	Geographical
scenario				
1 random selection – worst individual source-target average over 100 trials	8.76	6.05	8.05	5.81
2 min hops	7.57	3.76	4.74	3.26
3 min hops	10.9	2.14	11.29	2.14
4 diversity	10.57	6.24	8.1	3.57

Table 2
Worst Case Ratio: Latency with SOS vs Normal Route

model and overlay routing method	US		Europe	
	CHORD	Geographical	CHORD	Geographical
scenario				
1 random selection – worst individual source-target average over 100 trials	4	3	4	2.5
2 min hops	2	1.5	2	1.5
3 min hops	5	1	4.2	1
4 diversity	3.5	2.5	4.2	2

The normal route involved 3 POPs on average in both models.

Table 3
Average Ratio: # of POPs in SOS Route vs # POPs in Normal Route

When using scenario 3 with the geographic heuristic, the servlet was always selected from a node on the same POP as the target and the beacon was selected from a node on the same POP as the source and access point because there were eligible nodes at every POP. This resulted in the SOS route being identical to the normal route with the addition of a few detours to clients within the first and last POPs in the route, thus it was expected to produce the best results in terms of latency.

The results reported for random selection are averaged over 100 trials run per source-target pair. The actual increase in latency may be much higher depending on the specific servlet and beacon chosen. The greatest increase occurs when the source and target are close together. The overlay route may involve points geographically far from the source and target, turning a normally short route into one that may traverse every POP in the ISP at least once. Among all trials involving random selection, the worst case in the Europe model was an increase in latency 15 times that of the normal route between London and Paris when using CHORD and 9.5 times when using the geographical heuristic. In the US model, the worst case also involved a latency 15 times normal between NY and Philadelphia when using CHORD and 8.86 times when using the geographical heuristic. For NY to Philadelphia, the worst case increase using the geographical heuristic is approximately the same as the average (8.76) when using CHORD.

Results have been reported in % increases due to the latencies being estimated. Assuming the estimates are correct, the worst cases from all trials involved latencies of 378ms using CHORD and 230ms using the geographical heuristic. These are for one direction only, source to target, so the return trip time via a normal route must be added. In circumstances requiring the use of SOS, it may be expected that internet traffic is higher than normal and thus the % increase is more relevant as a performance measure.

The number of POPs serves as a measure of the complexity of the route but does not necessarily imply a physically long route because several POPs may be geographically close. In scenario 3, the beacon would be selected on the same POP as the access point. The ratio for scenario 3 using CHORD is high due to a couple of source-target pairs in which the beacon's overlay id was just prior to that of the access point's id, resulting in routing through several overlay nodes in the path between the access point and beacon.

When using CHORD, other variations for populating the overlay node's tables using powers of 3 and $i + x_j$ where x_j is the j^{th} number in the Fibonacci series, for $j = 3,4,5….$, were tested on a subset of source-target pairs but had no noticeable impact on the length of the route between the access point and beacon. A geographic ordering of the overlay nodes was also tested while maintaining the CHORD routing. Nodes that were geographically close were assigned ids placing them close together on the overlay network. While this shortened the route in cases where nodes X and Y were physically close, a packet was being routed from X to Y using the overlay and X was assigned a lower overlay id than Y; it resulted in a worst case scenario when Y was assigned the overlay id just prior to X's because the packet would route to O(log n) overlay nodes before reaching the one that knew about X.

4. Other Considerations

If the overlay nodes were placed within POPs as shown below opposed to being located in clients' LANs, this eliminates the latency due to the connection between the POP and client, and it could be more difficult to attack. In contrast to a client's LAN which may receive traffic for multiple reasons and has a relatively low bandwidth connection to the POP, a server dedicated to SOS and attached to a router within a POP allows most invalid traffic to be filtered out in a high capacity area. However, the use of special purpose servers would result in fewer potential overlay nodes. Such servers would also not remove the delay due to cross country routes through the overlay.

Having the overlay network span multiple ISPs will increase the latency of the SOS route. There will be a larger number of POPs serving potential overlay nodes. Even if the overlay nodes are geographically distributed in the same manner as with one ISP, the route between any pair of overlay nodes will increase on average due to having to route between ISPs. When the overlay nodes are in the same city but are served by different ISPs, having to route from one ISP POP to another ISP's POP as opposed to routing between nodes within the same POP will increase latency. Furthermore, if there is no peering point between the ISPs for that city, the route will require a path to a different city to reach a peering point.

5. Conclusion

Two factors affecting path length to consider when designing SOS are the method by which special purpose nodes are selected and the algorithm utilized for routing through the overlay. In order to decrease the probability of a successful attack, it is desirable to have no discernable relationship between the nodes serving special purposes in the overlay. Randomly selecting servlets and beacons accomplishes this goal. In contrast, setting restrictions on how these nodes are selected, such as selecting servlets within some distance of the target, reduces the number of overlay nodes that will be the focus of an attack if the attacker has some knowledge of the selection method or can make an educated guess. For example, if the attacker wishes to disable all potential servlets and assumes packets will not be routed across the US when the source and target are in neighboring cities, the attacker would conclude the servlets are in the same portion of the country as the target. If the actual policy is to select servlets within a specified distance of the target, the attacker can eliminate a significant portion of the nodes it must consider. Thus, while imposing a policy on how servlets and beacons are selected can reduce the path taken between the source and target, especially when a geographical relationship for routing has been established between overlay nodes, the probability of a successful attack may be lower using random selection.

The logic for how packets are routed through an overlay can have a significant impact on the path taken. In the case of CHORD, even if two overlay nodes are served by the same POP, the path through the overlay between the two nodes may involve multiple POPs over a large geographical area. As a result, the overlay routing algorithm may have a greater impact on the length of the route than the method by which the special purpose nodes are selected. Utilizing a geographical relationship when establishing routes between overlay nodes reduces the route length without increasing the probability of a successful attack because the overlay routing algorithm is not related to how special purpose nodes are selected. If selection of special purpose nodes is random, packets may still route over a large geographical area even when the overlay route is optimized merely because of the special purpose nodes being in geographically diverse locations in relation to themselves, the source and the target. However, direct routes are used between all nodes except from the access point to the beacon. If a geographical relationship, such as the heuristic, is used in overlay routing, the path length between the access point and beacon will be relatively close to that of the normal route.

If the number of overlay nodes is large enough such that utilizing a specific policy for selecting special purpose nodes does not result in a non-negligible probability of a successful attack, a policy can be combined with geographical relationship amongst overlay nodes to set bounds on the path length and expected latency resulting from SOS.

References

[1] A. Keromytis, V.Misra, D.Rubenstein, *SOS: Secure Overlay Services*, draft 2/2002.

[2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, *A Scalable Content-Addressable Network*, Proc. ACM SIGCOMM, San Diego, CA, August 2001.

[3] I. Stoica, R. Morris, D. Liben-Nowell, *Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications*, Proc. ACM SIGCOMM, San Diego, CA, August 2001.

[4] http://www.cybergeography.org/atlas/more_isp_maps.html