SHAPING NETWORK TOPOLOGY FOR PRIVACY AND PERFORMANCE

by

MD. MONJURUL HASAN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2012

To my parents for their prayers, blessings and for their consistent support

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor, Dr. Matthew Wright, whose expertise, understanding and patience, added considerably to my graduate experience. His guidance helped me throughout the time of research and writing of this thesis. I would also like to thank Dr. Vassilis Athitsos and Dr. Gergely Zaruba for taking time to serve on my thesis committee.

My thanks goes to the members of iSec lab, who always helped me through constructive discussion and inspired me by their works. I have been very fortunate to find new friends here, who helped me in various occasions and I must thank them all. I would also like to express my gratitude to National Science Foundation for sponsoring this research project.

Finally, I thank my family for their unconditional love and support all this time.

November 21, 2012

ABSTRACT

SHAPING NETWORK TOPOLOGY FOR PRIVACY AND PERFORMANCE

MD. MONJURUL HASAN, M.S.

The University of Texas at Arlington, 2012

Supervising Professor: Matthew Wright

Anonymity systems provide privacy for Internet communication and are becoming popular. Unfortunately, users experience slow Internet connection through existing systems because paths used to route traffic are selected randomly without considering latency between nodes. In our work, we aim to design a better approach for selecting paths in anonymity systems with improved performance and without sacrificing privacy. We consider stratified topologies for our design where nodes are divided into three hops. We propose a scheme to build restricted network topologies on top of a stratified topology that maximizes throughput. First, we use Tabu Search to build latency-aware stratified restricted topologies that select paths with low latency. Then we extend this approach for heterogeneous bandwidth and propose a bandwidth scheme to build multi-link stratified restricted topologies, with bandwidth capacity equally shared by each link. Using the reduced-overhead DLP scheme for padding, we measure the anonymity of our topologies by calculating entropy on the sender probability distribution. We evaluate our system in simulator by running traces of real Tor traffic through each topology. We compare our results with several restricted topologies based on brute force and greedy approaches. We show that our proposed topologies provide 22% gain in performance with no increase in dummy traffic overhead while maintaining reasonable levels of anonymity.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

Anonymity systems protect privacy during Internet communication. The motivation is to conceal the identity of the users from surveillance when they are communicating with each other. It was developed primarily for U.S. Navy, to protect government communications. But over a short span of time, it has gained a large user base from all around the world, who are more concerned about their online privacy. It has become a hope for the people from difficult environment, such as, politically oppressed people and journalists, who want to express their freedom of speech.

The system idea was first introduced by Chaum [1] in the form of *mix* networks. A mix can provide anonymity in mailing systems by hiding the correspondence between the input messages and the output messages in a cryptographic way. This concept leads to the design of various anonymity systems. Low-latecy anonymity systems try to anonymize interactive Internet traffic. They can provide real-time traffic response and suitable for Web browsing and instant messaging.

Tor, a low-latency anonymity system, is one of the largest deployed anonymity systems that has currently 3000 relays and $500,000$ users [2]. 95% of Tor users use it for interactive Web traffic and most of the rest use it for peer-to-peer file sharing. For interactive Web users, quick traffic response is a necessity but unfortunately they experience very slow internet over Tor. The privacy comes at the cost of performance, but this situation demands improvement.

Tor has several performace bottlenecks and one of the main issues lies in its path selection mechanism. Current path selection approach does not take network latency into consideration and selects relays for path considering only bandwidth, which has also several issues [3]. Although Tor path selection mechanism works in routing level and requires improvement, we believe that the improvement should begin from the topology level and so, we focus on building network topologies using scalable network paths. Network topology provides the base structure to ensure privacy and performance in low-latency anonymity systems [4]. The selection of a network topology requires care and consideration of many factors, otherwise there are good chances that the system will downgrade performance and disclose privacy to a passive adversary.

Restricted-route topologies offer better scalability and security when it comes to the choice of topology selection. It provides security against intersection attack [5] and requires low dummy packet overhead [5]. Of all the restricted-route topologies, Stratified Restricted topology [4] provides superior performance and anonymity. We also build several stratified restricted topologies in our work.

A balanced path selection in a network depends on several constraints. In Cisco IGRP protocol [6], the path selection metric is a composite metric of bandwidth and latency, and it selects a path that has the smallest value for that metric using the formula, $b + d$, where b is the inverse bandwidth of the flow and d is the topological delay (network latency). In low-latency anonymity systems, bandwidth variation is very high and we can't select a path using this composite metric formula. However, we intend to use network latency and node bandwidth as our main constraints for topological path construction.

1.1 Contribution

We propose a scheme that builds a latency-aware and bandwidth-aware stratified restricted topology. To select paths with minimum latency, we use Tabu Search and then we assign bandwidths to those paths using a bandwidth scheme. The final outcome of our scheme is a multi-link stratified restricted topology that improves performance significanly, while maintaining the privacy. We also build several other stratified restricted topologies using brute force and greedy approaches for our study. We analyze our topologies using RO-DLP padding scheme [4] for padding overhead and measure the anonymity of our system using entropy [21]. We have developed the necessary simulators to implement our schemes and integrated our results with RO-DLP simulator [4].

In chapter 2, we give the overview of low-latency anonymity systems, recent works to improve Tor's performance, different network topologies and basic knowledge of our proposed techniques. Chapter 3 introduces our system model and attack model. We also present our system design and describe the algorithms in this section. Chapter 4 describes our experimental setup and the purpose of the simulators, also the challenges we faced during experiment. We discuss the performance and anonymity results in chapter 5. Finally, we conclude in chapter 6 with a brief future work plan.

CHAPTER 2

BACKGROUND

2.1 Low-Latency Anonymity Systems

Chaum [1] first proposed the idea of *mix* based electronic mail system that hides the correspondence between sender and receiver without using any universal trusted authority. This idea, based on public key cryptography, forms the basis of anonymity systems. Low-latency anonymity systems is a class of anonymous systems that is intended to allow very limited packet delay and needs quick time response. This property is critical for interactive Web applications like Web browsing, instant messaging and SSH (secure shell). Over the years a number of low-latency systems have gained attention. Tor [8] is now the most popular and widely used anonymity system, while other mentionable names are Tarzan [9], Morphmix [10], Crowds [11] and Anonymizer [12]. All these systems are built on different types of network topologies.

2.1.1 Tor

Tor is a distributed overlay network for anonymous Internet communication operated by the volunteers. It routes the traffic through a series of onion routers making it hard for any third party to identify the communicating parties. For any communication, the tor client at the sender end creates a secure tunnel between itself and the receiver by randomly selecting three relays. The encryption mechanism makes sure that each relay only knows the relays located before and after itself, so no individual relay has the knowledge of full path. This is an important property to preserve anonymity. Tor nodes are cotegorized as onion proxies, onion routers and directory servers. Onion proxies run on the user's machine, onion routers relay the traffic and, directory servers keep all the information regarding onion routers [8] and distribute to the proxies as requested.

2.2 Enhancing Performance in Tor

Most of the Tor traffic is interative and demands quick response. But due to its design limitation and the constraint to preserve privacy, users experience high delay in Tor. To overcome the performance bottlenecks and ensure proper load balancing, a good amount of research has been conducted in the areas of path selection, congestion control and incentives.

2.2.1 Path Selection

In Tor, clients select the relays randomly from all active relays, with the selections weighted by router bandwidth. Clients receive these bandwidth information from the directory servers which are self advertised by the tor routers. Since routers are constantly engaged for relaying traffic, this may not reflect the actual capacity of the routers. Malicious routers can also falsify the information and capture a large amount of traffic by reporting high bandwidth. Snader *et al.* [3] proposed an opportunistic bandwidth measurement mechanism where each router keeps track of the peak bandwidth of its peers and directory server aggregates multiple router statistics to obtain a single router bandwidth. This active probing continues at a regular interval to get the up-to-date router bandwidth information. Clients have the freedom to select between performance-enhanced and privacy-enhanced path using a router selection function,

$$f_s(x) = \frac{1 - 2^{sx}}{1 - 2^s} \tag{2.1}$$

where s is the router selection parameter. High value of s guarantees high performance relays, where low value of s increases the anonymity level.

2.2.1.1 Path Selection Metrics

Rather selecting the path based on just router bandwidth, Panchenko *et al* [13] used link-latency and link-capacity as metrics for QoS-improved path selection. Link-latency is used as the primary metric and measured by calculating round trip time (RTT) of the established circuits in regular time intervals. Link-capacity is mainly used for throughput-oriented applications and estimated by the router itself by observing available bandwidth on a particular link. By aggregating the statistics for all connected links, the active bandwidth for that router is estimated and can be used as a metric for router selection. Wang et al. [14] showed node latency, which indicates node congestion delay, is sometimes larger than the link latency and needs to be considered as a path selection metrics to avoid congested nodes.

2.2.1.2 AS-Aware Tor Client

The random selection of tor relays can lead to a situation where relays from opposite end of the globe are selected. Since the distance of geolocation directly affects the network latencies, this random selection of path incurs high latency. On the other hand, location diversity is important for protecting anonymity, an adversary who controls a single AS can easily break the anonymity if a circuit has both end points in the same AS. Akhoondi *et al.* [15] proposed LASTor, an Autonomous System(AS) aware tor client that has the knowledge of the geolocation of tor relays and AS-level representation of internet topology. Geographically nearby tor relays are clustered together and each path is selected probabilistically from these clusters using a Weighted Shortest Path algorithm running at the client end. The path selection in LASTor is tunable and user can set preference for low latency or high entropy.

2.2.2 Congestion Control

Tor is mainly designed for interactive traffics but some users also use it for bulk downloads, using major share of the bandwidth. Sometimes due to improper path selection, the bandwidth difference between tor relays are very high. As a result, outgoing packets need to wait in a long queue for delivery and clients suffer from unexpected delay. One reason for the uncontrolled congestion is the wrong circuit-level and stream-level window size. These are fixed and relatively large, 512KB and 256KB respectively. AlSabah *et al.* proposed to make circuit window size dynamic, by initially starting with small window and then increasing the size dynamically by estimating congestion. Congestion is inferred using circuit round trip time and requires to modify the tor exit routers. Furthermore, tor flow control mechanism can be improved much by replacing this window based approach with the N23 scheme, used for ATM networks, that guarantees no packets are dropped.

2.2.2.1 Torchestra

Deepika et al. [16] showed that high improvement in delay reduction can be made by separating heavy traffic from the light traffic. For this, Tor node pairs have two separate connections for heavy traffic and light traffic, and the connections enjoy equal share of the bandwidth. Tor traffics are identified as light or heavy by estimating the weighted moving average of the number of cells in a circuit. Since the connections are separate, TCP congestion control for bulk taffic does not affect the interactive traffic.

2.2.3 Incentives

Tor depends on the volunteers to run the relays and donate their resources for its operations. Users experience quick time response if the selected relays have enough bandwidth to carry the traffic without congestion. So its important to maintain relays with good capacity in the tor network. Currently there are 1500 relays pushing 1Gbit/s of aggregate traffic [17] but it is still low considering the demand. To encourage the contributors to provide more resource for relaying traffic, incentives can be given, possibly in the form of good service. Ngan *et al.* [17] proposed an incentive scheme by appreciating the good service provider relays with gold star status and prioritizing their traffic. Tor directory authorities give this gold star status to the relays by measuring their performance. A gold star relay's traffic always receives high priority on the next hop.

2.3 Dependent Link Padding

Timing analysis attack has so far been proven the most successful attack to break the anonymity of the Low Latency Anonymity Systems. The attacker can correlate the communicating parties by analyzing the timing of messages. To resist this attack, one way is to make the incoming and outgoing flow pattern independent by sending outgoing packets at a constant rate [18], as it makes harder for the attacker to correlate between the timing of the flows. This scheme, known as Independent Link Padding, introduces dummy packets in the flow at a pre-scheduled time, even there is no incoming flow. This scheme largely depends on dummy packets and the sending schedule and, there are chances that it will have negative effect on the performance of the network when the number of dummies are too large. Dependent Link Padding (DLP) scheme [19] proposed alternate to this by sending packets based on the incoming flow so that number of dummy packets and packet drop rate can be significantly reduced. Also there is a strict delay bound (Δ) for all packets to send, to save dummy packets. DLP is considered as the most effective padding scheme to prevent timing analysis attack.

2.3.1 Reduced Overhead Dependent Link Padding

Diaz et al [4] proposed Reduced Overhead Dependent Link Padding (RO-DLP) algorithm for the links that multiplex several circuits. Link encryption is a common feature of most anonymous networks that conceals the correspondence of cells to the circuits within a link. A single link can contain single circuit or multiple circuit at the same time. However, in DLP scheme, this is not considered and every circuit is treated equally. Dummy cells are added to every outgoing circuit, so more padding overhead occurs to the network. RO-DLP removes dummy cells from a link if the number of circuits is greater than the number of packets to be forwarded. If each link has only one circuit, RO-DLP acts same as DLP. RO-DLP outperforms DLP when overhead of dummy traffic is a major issue, while providing the same security level when only external adversaries are considered.

2.4 Impact of Network Topology

In multi-hop anonymous communication systems, network topology plays the fundamental role to ensure that the network is capable to provide services with good user experience and sufficient privacy. Earlier works [4] [20] [5] shows that choice of network topology is likely to effect the padding overhead and more importantly the anonymity of the mix and users. In the following subsection, several topologies are discussed with their advantages and disadvantages.

2.4.1 Free Routes

Fully connected mix network is called *Free Routes*, where the clients randomly select several nodes from all the mix nodes with equal probability to build the message path. Free route topology offers best anonymity but more dummy packet overhead since every node can act as a sender in this topology. It is susceptible to the intersection attack [20] where an adversary observes the sender message patterns over time and takes the intersection of those messages to disclose the original receiver. Diaz *et al.* showed that with the implementation of DLP scheme, Free route topologies suffer from the feedback effect [4], which makes the padding overhead worst and leaves the topology out of consideration.

2.4.2 Mix Cascades

In mix cascades, clients use a predefined sequence of mixes for routing. It solves the intersection attack and requires less padding [4] overhead but in general, it provides less anonymity and less scalability [20]. It is also easy to perform *denial-of-service* attack against such cascades since disabling any mix of the cascade makes that cascade unusable.

2.4.3 Stratified Topology

A stratified network of N nodes with three hops would have the nodes divided into equal number of $N/3$ entry, middle, and exit nodes. Every entry node is connected to every middle node, and every middle node is connected to every exit node. With DLP, stratified topologies offer best tradeoff with slightly less anonymity than the free routes and more padding overhead than the cascades [4].

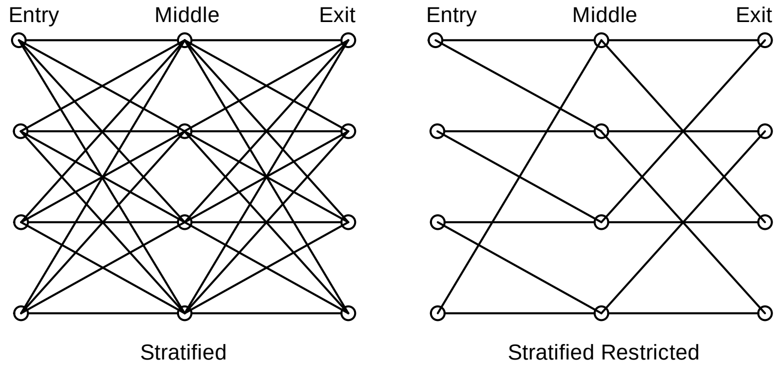

Figure 2.1. Stratified and Stratified Restricted topology for N=12 (figure from Diaz *et al.* [4]).

2.4.4 Stratified Restricted Topology

This is a restricted version of the stratified topology, in which a circuit, constructed through any entry node, can reach any exit node. The restriction limits the number of links in the network so that a node is connected only to a subset of nodes at each step, i.e., not all pairs have connection between them. Similar to the stratified topology, nodes are equally divided into entry, middle, and exit nodes. This topology improves the result further with less padding and more anonymity, with the implementation of RO-DLP.

The value of N is chosen of the form $N = 3K^2$, where K is an integer. Each entry node is connected to $K = N/3$ middle nodes, and each middle node to K exit nodes. An entry node i ($0 \leq i \leq N/3-1$) is connected to middle nodes $N/3 + [(i+j) \mod N/3]$, with $j = 0 \ldots (K-1)$; and a middle node i ($N/3 \leq i \leq 2N/3-1$) is connected to exit nodes $2N/3 + [(i + j \cdot K) \mod N/3]$, with $j = 0 \ldots (K-1)$ [4].

2.4.5 Restricted-routes

In restricted-route topologies, nodes are connected to a few nodes, rather than all the nodes present in the system. It provides better performance as the dummy packet overhead is greatly reduced, because the system has less number of links and requires to carry less dummy packets. Danezis [5] shows that restricted topologies are resistant to intersection attack and proposes expander mix networks using sparse, constant degree graph, where each mix is connnected to a few neighbors. This topology provides more scalability than mix cascades. But the topology suffers from feedback effect just as free routes with the implementation of DLP.

2.5 Anonymity Metrics

Degree of anonymity can be measured by calculating entropy over the probability associated with each user [21]. In a mix network, a global adversary can correlate the send and receive events of a particular message and can calculate the probability distribution of the users with each message sent and received. Then the adversary applies *entropy* formula to estimate the effective entropy size using the following formula,

$$S = -\sum_{u \in \psi} p_u \log_2 p_u \tag{2.2}$$

where ψ is the set of all users and p_u is the probability of a single user in the probability distribution.

The value of S denotes the number of bits that the adversary requires to know the actual identity of the user. Maximum entropy occurs when each user has uniform probability for the messages that transmit through the system. For mix networks, this value is $\log_2 \psi$. If the entropy size is 0, then the adversary does not require any additional information to identify the users and the mix network is totally compromised.

2.6 Tabu Search

Tabu Search [22] is an aritificial intelligence technique to solve NP-complete combinatorial optimization problems. When the search space is big, Tabu Search tries to find the global optimum value by iteratively selecting the best solution within its smaller neighbor search space. Local search procedure, such as hill climbing, is often stuck with the local optima, Tabu Search avoids this issue by using a memory structure called tabu list. The tabu list contains already visited solutions for the next few moves, thus reducing the solution space and preventing cycling. The success of Tabu Search largely depends on the neighborhood structure, the selection of objective function and the stopping criterion. It does not guarantee any global optimum solution but historically it has performed well for constraint satisfaction problems.

CHAPTER 3

SYSTEM

In this chapter, we describe our models and design. We start with system model and adversary model and then, we describe the design of our latency-aware and bandwidth-aware topologies.

3.1 System Model

We build a low-latency anonymity system based on stratified topology. This system uses onion routing to preserve anonymity. We assume number of hops in this system is always three. That means every circuit has path length of three, first being the connection between the sender and the entry mix, while last path connects the exit mix and the receiver. First hop nodes are entry guard nodes, second hop nodes are middle nodes and final hop nodes are exit nodes.

In our system, we evaluate directed one-way traffic from the sender to the receiver. However, it can be extended for duplex communication. To resist timing analysis attack, the system implements dynamic link padding scheme.

3.2 Attack Model

In our model, we assume adversary is global and active. Adversary can observe all the traffic between mixes but does not have internal knowledge of any mix. We assume all mixes in our system are benign. Since we implement DLP, attacker can not correlate sender and receiver by timing analysis. Also our system is resistant to intersection attack as we adopt restricted-route topologies [20].

3.3 Design

We consider network latency and bandwidth as the main constraints along with padding and anonymity metrics to design our system. First we design a latency-aware stratified restricted topology and later apply a bandwidth scheme on top of that for performance enhancement. The fundamental constraints considered here are almost same as with Stratified Restricted:

- Every exit node can be reached from every entry node.
- All middle nodes are used.
- Every node is connected to at least K number of nodes. This means each entry node has outdegree of atleast K, each middle node has atleast indegree K and outdegree K, and each exit node has atleast indegree K.

3.3.1 Latency-aware Stratified Restricted Topologies

In this section, we consider network latency as the only constraint to design latency-aware stratified restricted topologies. Our target is to achieve minimum latency with minimum number of links using the stratified topologies. This requires to compute all possible combinations of links between the node sets that meet the constraints, which clearly belongs to the class of NP-complete problems. We consider several algorithmic approaches to solve this NP-complete problem. The Tabu Search algorithm gives us the better tradeoff solution between performance and privacy, while we also consider greedy and bruteforce approaches for our study.

3.3.1.1 Tabu Search (TS) Approach

We propose our version of the TS algorithm to design scalable stratified restricted topologies. We start with a randomly generated K-restricted Stratified topology described in section 2.4.4 and apply this search technique to improve the latencies hop by hop. Tabus are mainly used to prevent cycling by storing previously visited solutions. First, we apply TS to the nodes in the middle hop to reorder them in a more latency effient way and then we do the same to the exit hop. The entry hop nodes are kept in fixed positions. As the TS proceeds, the network latency decreases. When the search completes, the node pairs between the hops have minimum latency links between them.

Algorithm 1 Tabu Search algorithm

Input: A Stratified Restricted Topology $G(V, E)$

Output: A Latency-aware Stratified Restricted Topology $G(V', E')$

```
1: Initial Solution $S \leftarrow$ randomized middle hop nodes $M < V1, V2, ...Vn >$
2: Best Solution $S^* \leftarrow S$
3: Tabu List $T \leftarrow S$
4: while Termination conditions not satisfied do
5:     Generate neighbor solutions $N(S)$
6:     $S' \leftarrow$ best neighbor solution of $N(S)$
7:     if $Latency(S') < Latency(S^*)$ then
8:         Set $S^* \leftarrow S'$
9:     end if
10:    $S \leftarrow S'$
11:    $T \leftarrow T \cup S$
12: end while
13: $M \leftarrow S^*$
14: For exit hop nodes, repeat the steps from Step 1
15: Output Latency-aware Stratified Restricted Topology $G(V', E')$
```

Now we discuss several design strategies for our TS algorithm.

Initial solution: Initially, we generate a random solution for each of the middle set and exit set.

Search space: This is the permutation set of the nodes for each of the sets. Generally, it includes all possible solution that can be traversed for the given problem.

Neighborhood structure and Candidate list: A neighbor of a solution is obtained by applying a single transformation to the current solution. At each iteration of our TS, we obtain a neighbor solution by swapping a single pair of nodes. Our candidate solutions consist of all the neighboring solutions that are non-tabu. In our topology, we have total 3 set of nodes for three hops and each set has $N/3 = K^2$ nodes, so we obtain $\binom{K^2}{2}$ candidate solutions at each step. In Figure 3.1, $S_1 < V_2, V_1, V_3, ...V_n >$ represents one of the six possible neighbor solutions from initial solution, $S < V_1, V_2, V_3, ...V_n >$.

Move mechanism: We move to a neighbor solution if the neighbor solution has less cost than other neighbor solutions. We measure the cost by the change of latency incurred for the node swap. Tabu Search also allows moves that do not improve the current best solution, this helps to prevent getting stuck at a local optimum solution.

Tabu stucture: We define the tabus as the visited solutions in our search. We store all the tabus in a list until we reach to the end of the search. This creates a potential tradeoff between memory usage and number of search steps. We assume that we have enough memory to hold all the tabus to avoid loops and to achieve higher performance.

Termination condition: We terminate our search when there is no candidate solution left to examine or when the search does not improve the current best solution for a fixed number of iterations.

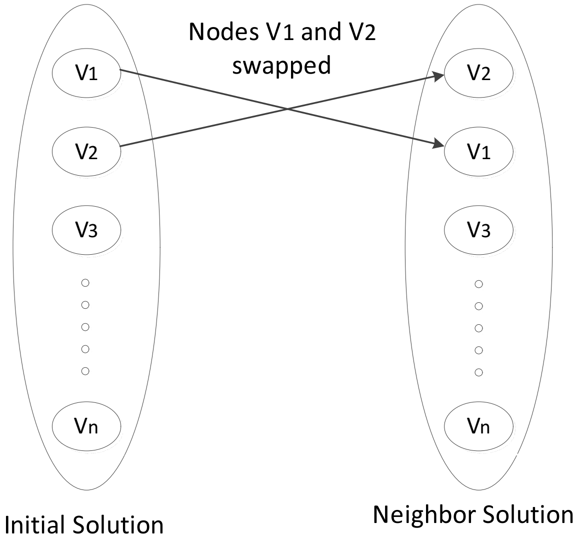

Figure 3.1. Generating a candidate solution for Tabu Search.

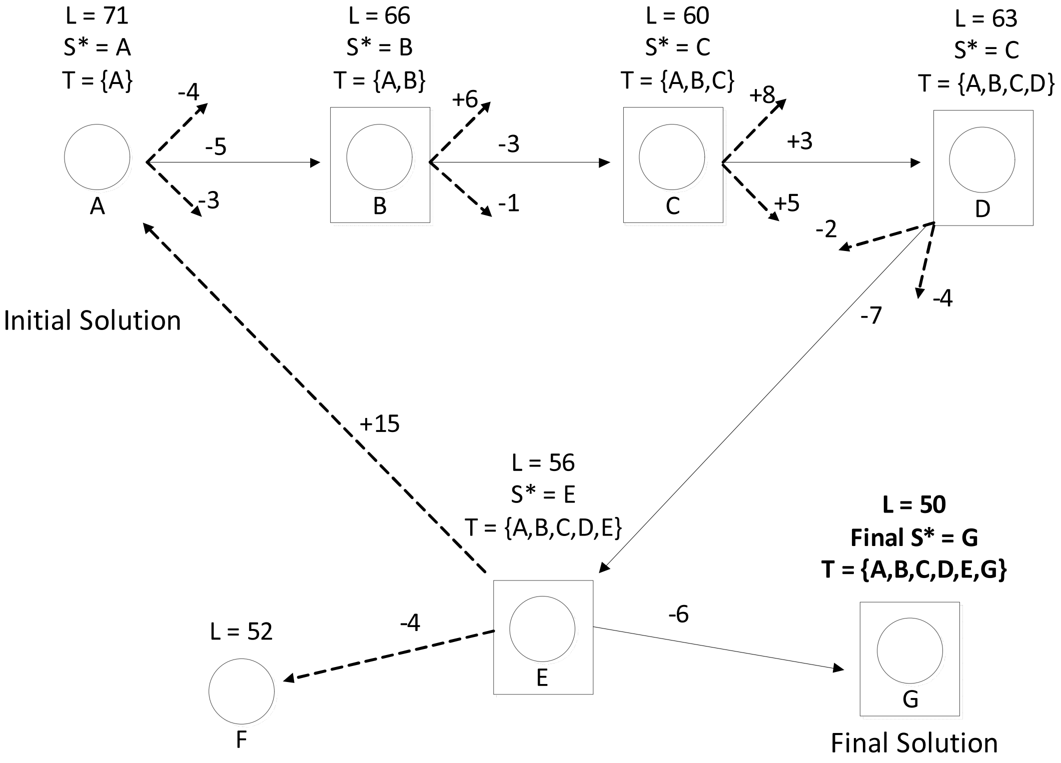

Figure 3.2. Building latency-aware restricted topologies by Tabu Search.

We illustrate the Algorithm 1 by Figure 3.2. Starting from a random initial solution A, Tabu search continues to run until it reaches the final solution G that has the lowest latency, L. At each step, it generates candidates for the next step and estimates the latency difference for all possible candidates. It selects the candidate with lowest latency and moves to that solution. Dotted links show the link to the potential other candidates. Initially, the tabu list, T is empty and its updated at every move with the best candidate solution. $S*$ stores the current best solution, its final value is G, the lowest solution achieved using TS.

We also study two variants of TS to make it usable when computation resources are limited: probabilistic and limited memory.

Probabilistic Tabu Search: The neighbor space of our candidate solution becomes very large when the network size grows. So it becomes computationally expensive to evaluate all the neighbors to find the best candidate. To make this more efficient, we take a random sample of the neighbor space and only evaluate that portion. This gives us the advantage of keeping the Tabu list short, while there is possibility to miss attractive solutions. We show that our probabilistic sampling gives almost the same result as the regular Tabu search for larger network. In our analysis, we evaluate 40% random neighbors from all possible neighbors of a solution.

Tabu Search with Limited Memory: For an environment where memory is limited, long-term tabu memory may be inappropriate if the network size is large. We examine short-term tabu memory that can store only N^2 recent best solution. We show that it also gives a similar result to long-term tabu but requires more iterations.

3.3.1.2 Greedy Approaches

We consider several greedy methods for our study. Although greedy approaches may not provide the optimal latency pairings in many or even most cases, we show that it still provides substantial performance improvements over randomly-selected Stratified Restricted topologies. We applied two greedy stratigies: high cost links removed and low cost links added.

Stratified Restricted with High Cost Links Removed: Starting from a Stratified topology with edges between all pairs of nodes at each hop, we remove the edges with high link costs. We continue this deletion process as long as the constraints described in Section 2.4.4 remain satisfied.

Stratified Restricted with Low Cost Links Added: In our second greedy approach, we start with an empty graph and continue to add the edges with lowest latencies until we achieve the constraints of a stratified restricted topology described in Section 2.4.4.

3.3.1.3 Brute force Approach

We try all possible stratified restricted topologies that strictly maintain all of the constraints described in Section 2.4.4 and output the topology that provides lowest network latency. This requires us to compute topologies for all K combination of links from each entry to each middle node and same for each middle to each exit node that satisfy the constraints. We achieve this by the permutation of the nodes in the middle set and the exit set.

Although the brute force approach gives us the best solution within the constraints, it requires a lot of computation to achieve that result. For N nodes with K^2 links between each hops, we need to compute $K^2! * K^2!$ possible stratified restricted topolgies. We design a parallel algorithm for brute force that takes the advantage of having multi-core processor and effectively reduces the runtime.

The chunk size at Line 4 of Alorithm 2 is the portion of the middle set that is submitted to each of the processors. In a multi-core environment, this size depends

Algorithm 2 Parallel Brute force algorithm

```
Input: A Stratified Restricted Topology $G(V, E)$
Output: A Latency-aware Stratified Restricted Topology $G(V', E')$
$P \leftarrow$ number of processors
$ML \leftarrow$ list of all possible ordering of the middle set nodes $\in$ V
$EL \leftarrow$ list of all possible ordering of the exit set nodes $\in$ V
Calculate $Chunk \leftarrow \frac{length(ML)}{P}$
for $p = 0 \rightarrow P$ do
    Create a new process $p$
    Assign $p$ with entry set, $ML[Chunk * i : Chunk * (i + 1)]$, $EL$
    Run $p$ to create Stratified Restricted Topologies
    Store the best solution from $p$
end for
Output the lowest latency topology $G(V', E')$ from all $p$
```

on the number of cores when the network size is constant. Smaller chunk size means more efficiency and can be achieved by using more cores.

3.3.2 Bandwidth Scheme

In our second phase, we propose a bandwidth scheme to build a multi-link bandwidth-aware stratified restricted topology on top of latency-aware stratified restricted topology. In a heterogeneous bandwidth environment, the bandwidth difference between node pairs are very high, as a result the high bandwidth nodes remain under-utilized. This scheme aims to increase the througput of the high bandwidth nodes by adding dedicated bandwidth links with the neighbor nodes. The multi-link approach provides alternate route for the traffic of different speeds, unlike Tor like systems that suffer from the congestion of nodes due to variety of traffics [16].

In our bandwidth scheme, nodes distribute their bandwidths by slots. At first, the nodes calculate their total bandwidth slots and then divide them equally for incoming and outgoing links. We add a new link between each node pair if there is enough bandwidth to share with the paired predecessor or the successor node. Each link is assigned a single slot which is the unit of shared bandwidth in our topology. A bandwidth slot is the ratio of half of the minimum bandwidth and K degree of a node.

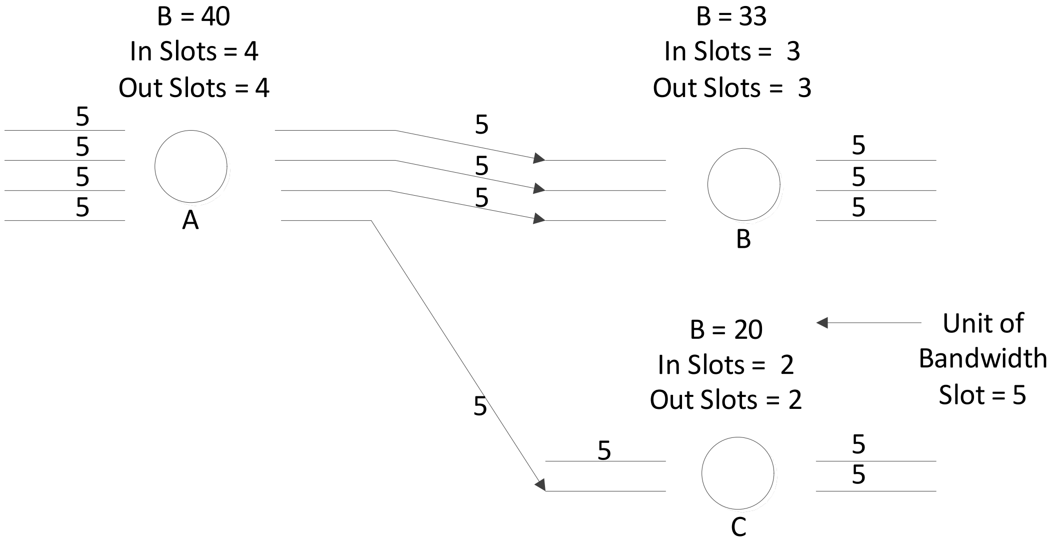

Figure 3.3. Bandwidth distribution for bandwidth-aware topologies.

In Figure 3.3, node A has two neighbors, node B and C. The unit bandwidth slot is calculated from the minimum of the assigned bandwidths to the nodes, using Tor bandwidth list, as described in Algorithm 3. Here, node C has the lowest bandwidth of 20 kilobytes per second, so the unit bandwidth slot has a bandwidth of $20/4 = 5$ kilobytes per second, where 4 is the total number of incoming and outgoing links of node C. After that, the total bandwidths of node A, B and C are distributed equally at the incoming and outgoing side, and bandwidth slots are calculated for each side of each node. Then one slot from each node is assigned to each existing connections. Node A has link with B and C. So 2 slots from A, 1 slot from both B and C are used for these links. Now node A has 2 bandwidth slots left at the outgoing side and Node B has 2 bandwidth slots at the incoming side, so they can share up to 2 bandwidth slots, each results in a new link between the node pair. Thus the scheme creates links between node pairs and at the end, high bandwidth nodes have more number of links in the system than the other nodes. ensuring their bandwidth utilization.

Algorithm 3 Bandwidth Scheme

Input: A Latency-aware Stratified Restricted Topology $G(V, E)$, Tor bandwidth list BL

Output: A Multi-link Bandwidth-aware Stratified Restricted Topology $G(V, E')$

```
for each node ∈ V do
    assign random bandwidth from BL
end for
unit bandwidth slot ← $\frac{minimum\, bandwidth/2}{K}$
for each node ∈ V do
    divide bandwidth equally for incoming and outgoing side
    calculate total slots for incoming and outgoing bandwidths
    assign one bandwidth slot for each of the existing incoming and outgoing link
end for
for each middle node ∈ V do
    while the paired nodes have free bandwidth slot do
        add a link $e$ to a paired node if the middle node has free slot
        assign a unit bandwidth slot to $e$, remove one slot from both the nodes
        $E' = E + e$
    end while
end for
Output the bandwidth-aware topology $G(V, E')$
```

CHAPTER 4

EXPERIMENTS

We have implemented an experimental evaluation platform to conduct our analysis using real Tor data. Our platform consists of three simulators, which are executed sequentially and, the output of previous simulator is used as the input to the next simulator. These simulators are Link-Cost, Bandwidth and RO-DLP simulator.

Our platform is written entirely in Python. We use Networkx [23], a Python package to build and analyze our networks.

4.1 Link-Cost Simulator

Link-Cost simulator takes the size of the network as input and then generates the latency-aware Stratified Restricted Topologies. We use median RTT value from the KING dataset [24] to estimate link cost between the nodes of our networks. To avoid complexity, we do not implement any encryption features in the system.

We use a quad-core system to evaluate our parallel bruteforce algorithm, which gives us almost quadruple performance over single core system.

4.2 Bandwidth Simulator

The second simulator is the Bandwidth simulator that distributes bandwidth to the multiple links between the pair of nodes of the latency-aware Stratified Restricted topologies. For each node, we assign bandwidth randomly from a bandwidth list, which is collected from Tor routers. Bandwidth is distributed on slot basis until the paired nodes have enough bandwidth slots to share with each other.

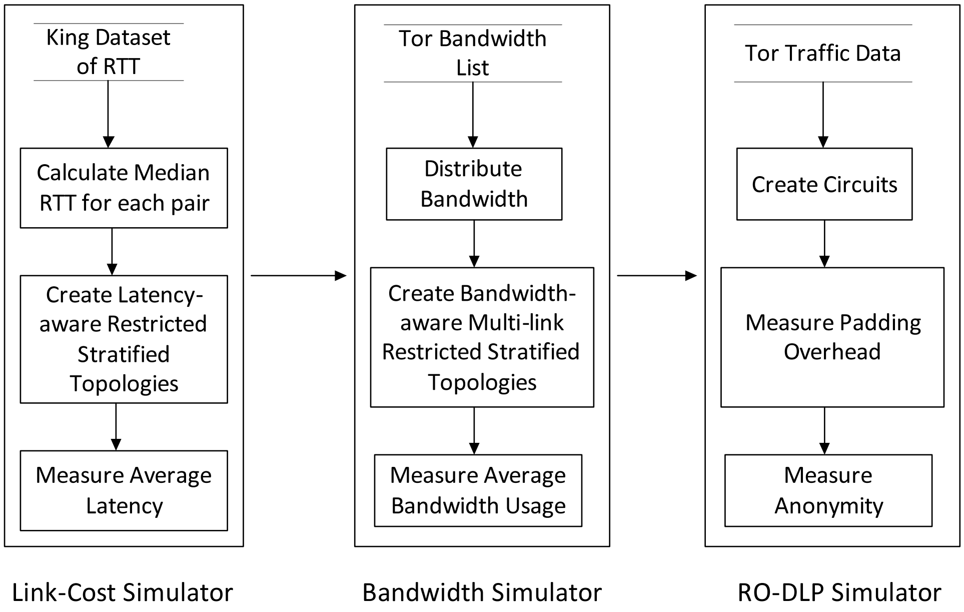

Figure 4.1. Simulators for our experiments.

4.3 RO-DLP Simulator

We evaluate the latency-aware and bandwidth-aware topologies for padding overhead and anonymity using the simulator of RO-DLP paper [4]. The simulator uses the traces of real traffic data of TOR for the timespan of 24 hours. The simulator itself has two parts, first part creates the circuits and run the TOR traffic with dummy packets as defined in RO-DLP scheme. The second part analyzes the logs of the circuits for hidden states and measures the anonymity of the system using Bayesian Inference technique.

4.4 Experimental Challenges

4.4.1 Brute force Computation

Our parallel brute force algorithm can compute all possible stratified restricted topologies for small size networks, but for large networks, it becomes infeasibe to compute, considering the resources we have. The main reason is, the number of computation increases exponentially with the number of nodes in the system. For example, we need to compute $9! * 9! = 1.3e+11$ possible stratified restricted topologies for 27 nodes and for 48 nodes, the number becomes $!16 * !16 = 4.3e + 26$, which is $3.3e + 15$ times larger than the previous computation. Our experimental setup for brute force compute all the stratified restricted topologies for 27 nodes in 13 days 21.5 hours. At this rate, it is quite clear that the brute force algorithm is unable to compute large networks in reasonable time, even we tripple the number of processors.

CHAPTER 5

RESULT AND DISCUSSION

In this chapter, we present the results of our experiments towards building a latency-aware and bandwidth-aware stratified restricted topology. We have performed all of our experiements on networks with 12, 27, 48, 75, 108, 147 and 192 nodes. This gave us a clearer picture of how the latency, anonymity, and overhead varies with the size of the network and is affected by the topology.

5.1 Edges for stratified restricted topologies

First, we compare the total number of edges required to build the different stratified restricted topologies. As we can see from Figure 5.1, among the restricted topologies, Stratified Restricted results in the lowest number of edges for K or at least K degrees. Our latency-aware restricted topology with Tabu Search also gives the same result, since it is built using the same restricted formula. Edges for High Cost Links Removed topology remains same for the small size networks and the difference increases for large size networks. Low Cost Links Added topology shows the worst case with a very large number of links, which means, it would be a bad choice in terms of restrictedness.

5.2 Performance analysis of stratified restricted topologies

Figure 5.2 shows the latency performance of the stratified topologies based the on median RTT between all pair of nodes. To determine the link cost, we measured

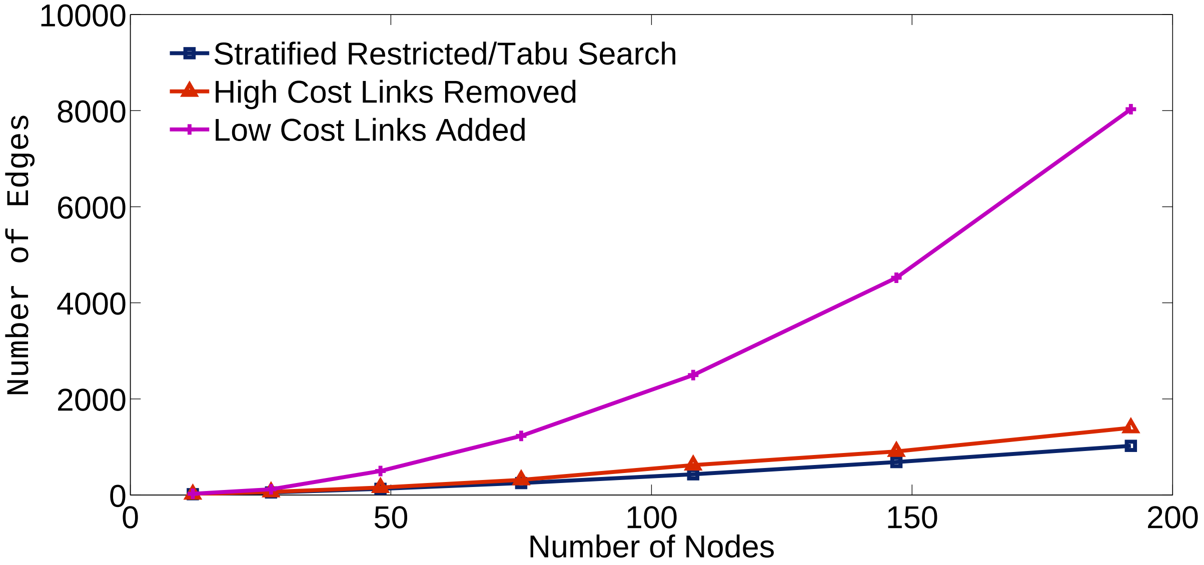

Figure 5.1. Total edges for stratified restricted topologies.

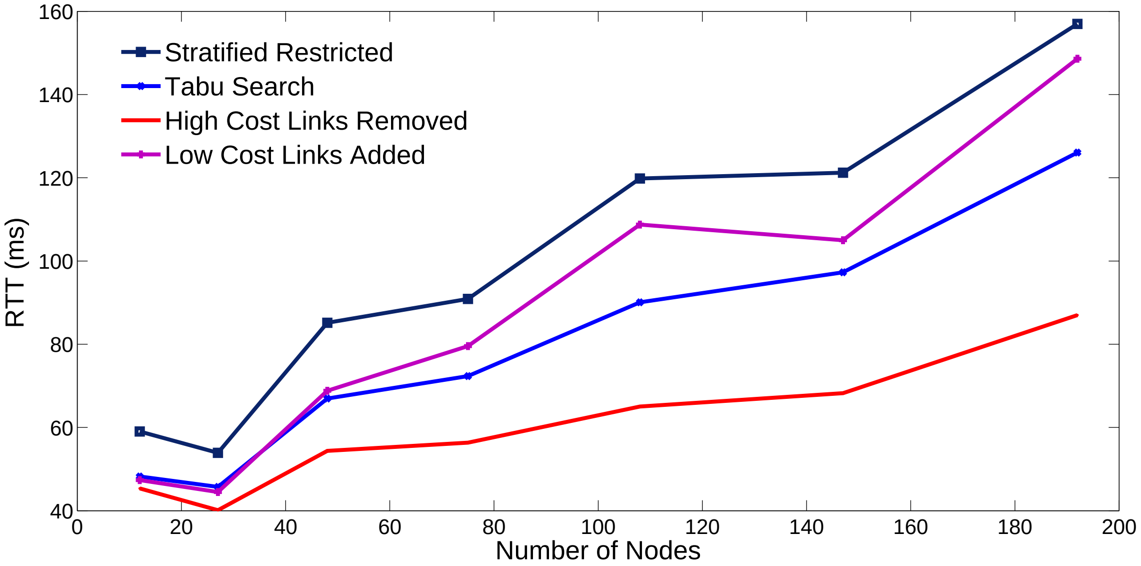

Figure 5.2. Average latency for stratified restricted topologies.

average latency which is the ratio of total latencty for the topology and total number of edges.

We see that our restricted topology with Tabu Search achieves significant performance improvement over Stratified Restricted. Our greedy topologies show improved latencies and the best result comes from High Cost Links Removed, which shows 50% improvement. This topology, however, has more links and thus has less of the anonymity benefit of restricted topologies. For larger networks with more than 147 nodes, the margin of improvement increases (by almost 22%) for the restricted topolgy with Tabu Search. Thus, our proposed network topologies perform better for larger networks, which are more realistic.

We compared our results with the optimal results found by the brute force apporoach. We see that the average latencies for networks with 12 and 27 nodes are 47.8275 ms and 44.3 ms respectively, which are same as the best case result of our Tabu Search topology and close to the average result. Due to its extremely high computation requirement, our parallel brute force approach is infeasible for larger networks and we could not compare our results any further.

5.3 Variation of Tabu Search

Figure 5.3 shows the latency performance of different stratified restricted topologies based on Tabu Search, using same test as described above. The variations are done with neighborhood size and tabu memory, as described in Section 3.3.1.1. We see that the regular Tabu Search and the variations give us nrealy the same results, with the variations adding the cost of running for more number of iterations. This indicates that our basic approach works well, which is good for keeping overhead low. However, these variations have their own advantages and disadvantages that need to be considered, as pointed out in Section 3.3.1.1.

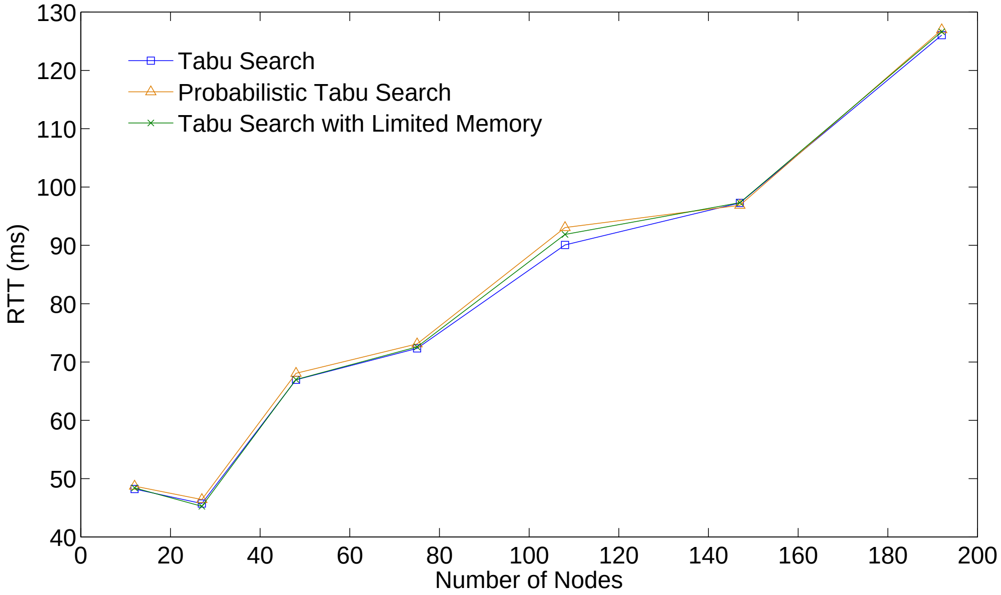

Figure 5.3. Average latency for different Tabu Search approaches.

5.4 Bandwidth utilization of the bandwidth-aware topology

We now estimate the bandwidth utilization of our multi-link bandwidth-aware restricted topology and use the remaining unused percentage of bandwidth as our metric, The usused bandwidth percentage of a node is measured by the ratio per hundred of total unused bandwidth and the total bandwidth. For a network, this metric is calculated by taking the average of the unused bandwidth percentage of the nodes, which is the ratio of total unused bandwidth percentage of the nodes and total number of nodes.

Figure 5.4 shows the unused bandwidth percentage for three variable size networks, the smallest one, one middle size and the largest network, of our experiment. For each network, the results were obtained by running the bandwidth scheme for a dozen times. We observe that the un-utilization rate decreases when the network size

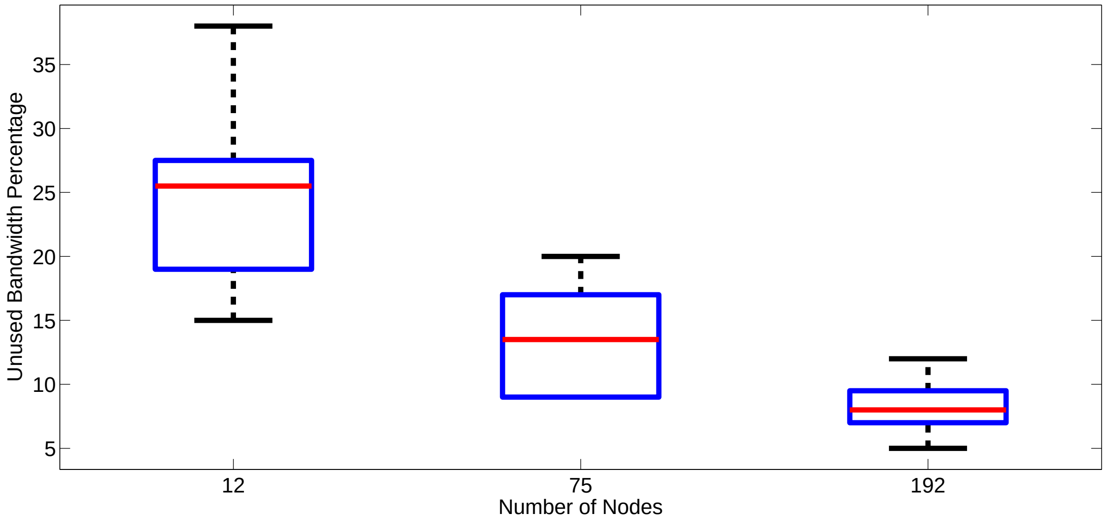

Figure 5.4. Unused bandwidth percentage for bandwidth-aware restricted topology.

grows and the decrement rate of the median value is almost 50% from the small to the middle network and middle to the large network. This gives us a clear indication that nodes are likely to have a very high bandwidth utilization rate in our system for larger network.

5.5 Dummy Packet Overhead of the Topologies

We measure the padding overhead factor which is the ratio of total dummy packets sent and total real packets sent over the same period. The overhead factor in Figure 5.5 is measured for a middle size network of 75 nodes. We see that the overhead factor is lowest for Tabu Search and highest for Low Cost Links Added, the difference is almost 90%. That means for the same size network, almost 90% more dummy packet is required to prevent timing analysis attack in Low Cost Links Added topology than the latency-aware Tabu Search topology. The factor increases when more number of links are present in the network.

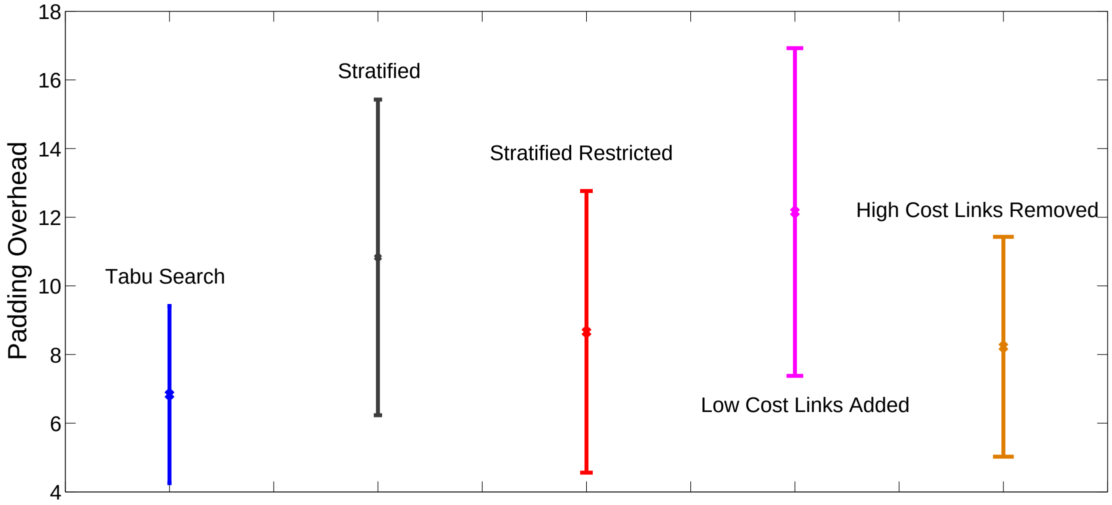

Figure 5.5. Padding overhead for stratified restricted topologies.

5.6 Anonymity Measurement

In Figure 5.6, we measure the sender anonymity of the topologies by calculating entropy on the sender probability distribution. We consider entry nodes as the sender and exit nodes as the receiver, and calculate the probability for each sender message that is received at the exit nodes. Stratified Restricted topology and Tabu Search show the highest anonymity for all networks and the attacker requires maximum number of bits to guess the actual sender. For bandwidth-aware topology, the anonymity level is the lowest and for larger networks, the chances for the attacker to compromise the system gets almost double. Low Cost Links Added shows almost same level of anonymity as the Stratified Restricted and High Cost Links Removed remains closer with a reduction of 0.25 entropy. Although High Cost Links Removed topology shows improved latency performance, it still discloses some information about the anonymity set to the attacker. The amount of disclosed information may be large enough and significant to the attacker when the network size is big.

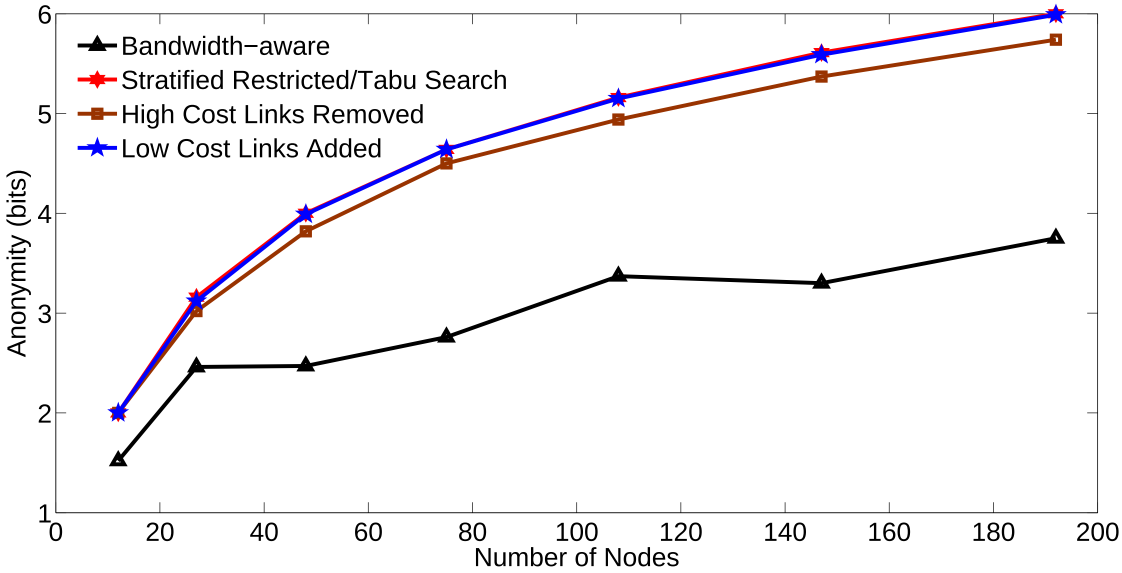

Figure 5.6. Anonymity for stratified restricted topologies.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, we proposed a scheme to build latency-aware and bandwidth-aware stratified restricted topologies for low-latency anonymity systems. We simulated our scheme and show latencies, padding overhead and anonymity for networks up to 192 nodes. We showed that our restricted topology provides improved performance with no loss in anonymity. In a network with homogenous bandwidth, we build topologies by using Tabu Search. When heterogenous bandwidth is considered, we created bandwidth-aware topologies that have multiple links between node pairs, with each link having dedicated bandwidth provided by bandwidth slot. This scheme increases the bandwidth utilization of the nodes and provides alternate ways for traffic routing for different types of streams. We also studied several other latency-aware stratified restricted topologies using brute force and greedy approaches, and we conclude that our topology offers better trade-off between privacy and performance.

In the future, we plan to use our scheme to build topologies with larger network size and analyze the results. We want to run real traffic inside our multi-link bandwidth-aware topology and measure the actual bandwidth used by the nodes at routing level, to show the real-time efficiency of our topology. Our multi-link topology does not have any upper bound for the links between node pairs. It may be hard to maintain if the number of links get too high. We plan to address this issue by setting an upper bound for the number of links between pairs or by combining links together. When nodes are congested, they effect the overall network performance. So we also plan to integrate congestion control mechanism in our topologies.

During the construction of our topology, we assume that each hop has a equal size, fixed collection of nodes. We did not explore the problem of building the set of nodes for each hop, i.e., given a collection of nodes, which nodes belong to the entry hop and which nodes belong to the exit hop, the rest of the nodes are then considered for the middle hop. It is also unclear how the networks will behave when the number of nodes vary in each hop of our system. These optimization issues are likely to affect our topology and we plan to look into these in the future.

REFERENCES

[1] D. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Communications of the ACM*, 1981.

[2] “Tor metrics portal,” `https://metrics.torproject.org/`.

[3] R. Snader and N. Borisov, “A tune-up for Tor: Improving security and performance in the Tor network,” in *Proceedings of the Network and Distributed Security Symposium - NDSS ’08*, 2008.

[4] C. Diaz, S. J. Murdoch, and C. Troncoso, “Impact of network topology on anonymity and overhead in low-latency anonymity networks,” in *Proceedings of the 10th Privacy Enhancing Technologies Symposium (PETS 2010)*, 2010.

[5] G. Danezis, “Mix-networks with restricted routes,” in *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*, 2003.

[6] D. Farinacci, “Introduction to enhanced igrp(eigrp),” `http://www.cisco.com`, 1993.

[7] C. Troncoso and G. Danezis, “The bayesian traffic analysis of mix networks,” in *Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009*, 2009.

[8] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The second-generation onion router,” in *Proceedings of the 13th USENIX Security Symposium*, 2004.

[9] 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 (CCS 2002)*, 2002.

[10] M. Rennhard and B. Plattner, “Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection,” in *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002)*, 2002.

[11] M. Reiter and A. Rubin, “Crowds: Anonymity for web transactions,” *ACM Transactions on Information and System Security*, 1998.

[12] J. Boyan, “The anonymizer - protecting user privacy on the web,” 1997.

[13] A. Panchenko and J. Renner, “Path selection metrics for performance-improved onion routing,” in *Proceedings of the 2009 Ninth Annual International Symposium on Applications and the Internet*, 2009.

[14] T. Wang, K. Bauer, C. Forero, and I. Goldberg, “Congestion-aware Path Selection for Tor,” in *Proceedings of Financial Cryptography and Data Security (FC’12)*, 2012.

[15] M. Akhoondi, C. Yu, and H. V. Madhyastha, “LASTor: A Low-Latency AS-Aware Tor Client,” in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, 2012.

[16] D. Gopal and N. Heninger, “Torchestra: Reducing interactive traffic delays over tor,” in *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2012)*, 2012.

[17] T.-W. J. Ngan, R. Dingledine, and D. S. Wallach, “Building Incentives into Tor,” in *Proceedings of Financial Cryptography (FC ’10)*, 2010.

[18] A. Pfitzmann, B. Pfitzmann, and M. Waidner, “ISDN-mixes: Untraceable communication with very small bandwidth overhead,” in *Proceedings of the GI/ITG Conference on Communication in Distributed Systems*, 1991.

[19] W. Wang, M. Motani, and V. Srinivasan, “Dependent link padding algorithms for low latency anonymity systems,” in *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008)*, 2008.

[20] O. Berthold, A. Pfitzmann, and R. Standtke, "The disadvantages of free MIX routes and how to overcome them," in *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[21] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*, 2002.

[22] F. Glover and M. Laguna, "Modern heuristic techniques for combinatorial problems," 1993, ch. Tabu search.

[23] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, 2008.

[24] "King dataset," `http://pdos.csail.mit.edu/p2psim/kingdata/`.

BIOGRAPHICAL STATEMENT

Md. Monjurul Hasan is a Master's student in Computer Science at the University of Texas at Arlington. He was born and raised in Dhaka, the capital of Bangladesh, the beautiful land of six seasons. He received his B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), in 2008. He joined the Information Security (iSec) Lab, University of Texas at Arlington, in Fall 2010, as a graduate research assistant. His current research interest lies in anonymous communication systems and software security.