A Dynamic Selection Algorithm of Tor Relay Based on Client Bias

Yun Zhang
School of Computer Science (National Pilot Software Engineering School)
Beijing University of Posts and Telecommunications
Beijing, China
zhangyunu@bupt.edu.cn

YaMei Xia
School of Computer Science (National Pilot Software Engineering School)
Beijing University of Posts and Telecommunications
Beijing, China
ymxia@bupt.edu.cn

***Abstract*—Tor network is one of the most widely used low latency anonymous communication systems. To balance the network load, the Tor network uses an adjusted bandwidth weighted random selection algorithm to uniformly select relays of a circuit. A client can not adjust the bias of the relays during the circuit establishment process. However, the client's different requirements for network anonymity and performance may affect the further extension of Tor. This paper proposes a relay dynamic selection algorithm that allows the client to set the relay preference when establishing a circuit. The algorithm defines a dynamic parameter that can be adjusted by the client. Defining different dynamic parameters can realize the degree of bias for high-bandwidth relay or low-bandwidth relay during circuit establishment. The proposed algorithm is implemented in the Tor source code and the homogeneous network and heterogeneous network are respectively deployed on the Shadow simulation platform for experiments. Based on the simulation results, we have observed that setting different dynamic parameters can achieve partial improvement of network performance or network anonymity.**

Keywords—Tor Network, Relay Selection, Client Bias.

I. INTRODUCTION

Tor network is a low-latency anonymous communication system[1], which provides anonymity for applications based on TCP connection by establishing a reroute circuit encrypted by three relays between the sender and the receiver to transmit messages. The first, middle, and last node in the circuit is called the entry guard relay, middle relay, and exit relay, respectively. As a distributed network of routers provided by volunteers, the Tor network has grown from more than 1,000 routers to over 7,000 now.

In the Tor network, the adjusted bandwidth weighted random selection algorithm is adopted to balance the network load to prevent the abuse of scarce resources when the resources of the Tor network are in short supply. For example, if the system lacks exit relays, a router with the Exit flag is more likely to be selected as the exit relay of the circuit instead of the middle relay. At the same time, it tends to choose routers with larger bandwidth to be the circuit relay when the client establishes a circuit. However, a previous study has shown that the load balancing process of the Tor network may lead to the possibility of low-resource attacks[2]. Therefore, there is a mutual restriction between performance and anonymity in the Tor network.

A Tor network circuit is composed of three encrypted relays to forward client traffic. This special structure causes it to be slower than the average network speed. To solve the problem of poor performance, researchers have made improvements from different perspectives. Tang et al.[3] proposed a new scheduling mechanism for the circuit selection and deployed it in Tor, which ensures that the burst interactive circuit has a higher priority than the bulk transmission circuit. The congestion-aware routing algorithm proposed by Wang et al.[4] introduces the relay waiting time as the parameter of congestion. The client uses the opportunistic measurement method to avoid nodes with too long a waiting time as the relay. The dynamic traffic segmentation technology proposed by Alsabah et al.[5] improves the performance of clients using low-bandwidth bridge routers. References [6][7][8] encourage more volunteers to join the network to provide more usable bandwidth to satisfy the interactive users and bulk clients. However, Lei et al.[9] believe that the simple encouragement mechanism may lead to the decrease of network performance. Snader and Borisov[10] proposed a tunable node selection strategy. It allows the client to adjust the circuit by adjusting the degree to which the parameter selection is biased towards different relays. Akhoondi et al.[11] proposed a Tor client called LASTor, which uses geographic distance to estimate the delay to reduce circuit delay. Wacek et al.[12] conducted a comprehensive test on some improved routing algorithms. The research results showed that the relay selection algorithm without considering router bandwidth has poor performance. Considering Wacek's suggestion, Lei et al.[9] proposed mTor, which constructs multiple circuit groups composed of low-bandwidth routers and transmits bulk data through the circuit to avoid overloading of high-bandwidth relays. Reference [13] explored a weighted function that balances bandwidth and nodes' geographic distance by comparing network congestion, circuit delay, geographic length, and their combinations.

In this paper, we propose a relay dynamic selection algorithm to improve the network performance or anonymity within limits by adjusting the options. First, considering the constraints of performance and anonymity in the Tor network and the importance of bandwidth, a relay selection algorithm is proposed. When a circuit is establishing, the candidate routers at each position are sorted according to the weighted bandwidth. The nodes are biased when the relay is selected according to the dynamic parameter. Second, we modify the Tor source code and conduct a Shadow simulation experiment. By analyzing the results, it is found that the algorithm can partially improve the network performance, while the network anonymity decreases, and the converse is also true.

The organization of the rest of this paper is as follows. The section II explains the formula definition and the relay selection process of each position. The section III and section IV conducts experiments and analyzes the experimental results. The section V concludes this paper.

II. MODEL

According to status flags, routers are divided into four categories: Guard-flagged, Exit-flagged, Guard+Exit-flagged, and non-flagged. The general selection of relays is selecting an entry guard relay from routers with Guard flags (Guard-flagged and Guard+Exit-flagged), selecting an exit relay from routers with Exit flags (Exit-flagged and Guard+Exit-flagged), and selecting a middle relay from all routers. The three selected relays cannot be duplicated, in the same family, or in the same /16 subnet. The entry guard relay does not change easily after selection, and a client uses this relay as the first hop of all circuits unless the currently selected relay is unreachable or has existed for 60 days to 9 months before selecting a new entry guard relay[14]. The algorithm in this paper obeys these rules.

When selecting a relay for each position of the circuit, it is necessary to multiply the node bandwidth given in the network consensus document by the position weight to get the new value of each node at this position. For the convenience of description, this value is called the weighted bandwidth of the relay. Then, each relay is assigned a probability according to the weighted bandwidth, and the relay is selected into the circuit according to the assigned probability.

A. Formula Definition

We introduce a random variable $x \in (0,1)$ and a dynamic function $f_k(x)$. x can be reasonably enlarged or reduced to a new (0,1) interval through calculation, and different dynamic levels can be achieved by adjusting the dynamic parameter k. $f_k(x)$ is defined by Equation (1).

$$f_k(x) = 2^{x^k} - 1, \qquad 0 < k \le 0.7 \cup k \ge 0.9 \qquad (1)$$

The available data range of the dynamic parameter k is a real number in $0 < k \le 0.7$ and $k \ge 0.9$. For example, the result of the function $f_k(x)$ on $x \in (0,1)$ as $k = \{0.25,0.5,1,5,10\}$ is shown in Fig. 1. Fig. 1 shows that the value of $f_k(x)$ becomes larger as k decreases when $0 < k \le 0.7$, and the value of $f_k(x)$ becomes smaller as k increases when $k \ge 0.9$. We define that the client tends to choose more high-weighted bandwidth relays when $0 < k \le 0.7$, and the client tends to choose more low-weighted bandwidth relays when $k \ge 0.9$.

We do not recommend $0.7 < k < 0.9$. For example, Fig. 2 shows the value of $f_k(x)$ when $k = \{0.75,0.8,0.85\}$. The function $f_k(x)$ is close to the function $f(x) = x$ when the dynamic parameter $k \in (0.7,0.9)$, so the value deformed by the function $f_k(x)$ is close to the original value of the random variable x.

B. Relay Selection

We set up the same selection process for the entry relay, middle relay, and exit relay of the circuit. We assume that the client has information about a sufficient number of nodes, and nodes at different positions have calculated the corresponding weighted bandwidth according to the weights. Then, the process of selecting relays on the three positions of the Tor circuit is as follows:

Step 1: We suppose a total of m candidate routers at the entry position and the weighted bandwidth of router $j \in \{1,2,\dots j,\dots,m\}$ is B_j. Firstly, we sort the router according to the weighted bandwidth B_j from small to large, and obtain the weighted bandwidth sum $B = SUM(B_1, B_2, \dots, B_j, \dots, B_m)$. Secondly, the random number x is generated in $(0, B)$ and normalized to $(0,1)$. Thirdly, x is transformed according to formula (1) to obtain a new random number $x = f_k(x)$. Finally, the weighted bandwidth of each router is accumulated, and we will find that the ith router satisfies the calculation result $SUM(0, B_1, B_2, \dots, B_{i-1}) < B * x \le SUM(0, B_1, B_2, \dots, B_{i-1}, B_i)$, so the ith router is the relay selected at the entry position.

Step 2: We follow step 1 to select a middle relay from the alternative routers in the middle position.

Step 3: We follow step 1 to select an exit relay from the alternative routers in the exit position.

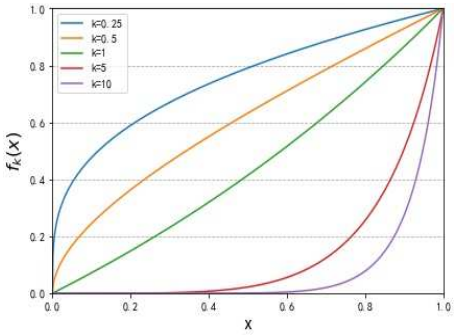


Fig. 1. Values of $f_k(x)$ in $x \in (0,1)$ with different k

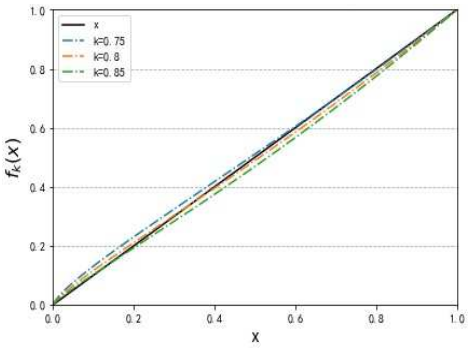


Fig. 2. Values of $f_k(x)$ in $x \in (0,1)$ when $k = \{0.75,0.8,0.85\}$

III. EXPERIMENT SETTINGS

Shadow simulation platform can run Tor source code. It can run any scale Tor network simulation by calling the historical data. The proposed algorithm is implemented in Tor source code and simulated on Shadow. During the simulation, each web browsing client will intermittently request a file with the size of 320KB, and each bulk client will continuously request a file with the size of 5MB.

The experiment sets up two networks to simulate client behaviors. One of the networks is homogeneous, and the other is heterogeneous. In the two networks, 3 directory server nodes and 145 relay nodes are set up, and 100 of the 1000 servers announced on the Alexa website are selected as server nodes. The bandwidth of each router is from the real node bandwidth published in the Tor network. The main difference between the two networks is the client nodes. The homogeneous network sets up 600 clients, among which the web browsing clients and the bulk clients are set according to Webclient:Bulkclient=9:1. The ratio of the network relays to the clients is about 1:4. In the heterogeneous network, 800 web browsing clients are set up, of which 500 clients use the

original algorithm of the Tor network, and the rest 300 clients use the improved algorithm. A total of 6 dynamic levels is set in the experiment, respectively $k = \{0.4,0.5,0.6,3,4,5\}$, and each dynamic level has 50 clients. The number of different types of nodes in the networks is shown in Table I.

TABLE I. NODES USED DURING EXPERIMENT

Nodes Type	Networks	
	Homogeneous Network	***Heterogeneous Network***
Guard-flagged Relays	53	53
Exit-flagged Relays	6	6
Guard+Exit-flagged Relays	24	24
Non-flagged Relays	62	62
Clients (320KB)	540	800
Clients (5MB)	60	/

The comparison of performance in the two simulation networks is achieved by counting 1) the cumulative distribution of the first byte download time of a file, 2) the cumulative distribution of download completion time of a file, and 3) the cumulative distribution of file download completion amount of each client.

The Gini coefficient proposed by Snader and Borisov[10] is used to compare the equality of selected relays in the circuits to compare the anonymity. A Gini coefficient of 0 means that the relay selection is completely equal (*i.e* the selection frequency of all routers is the same), and the network has the best anonymity. A Gini coefficient of 1 means that the relay selection is completely unequal (*i.e* only one router is selected), and the network has the worst anonymity.

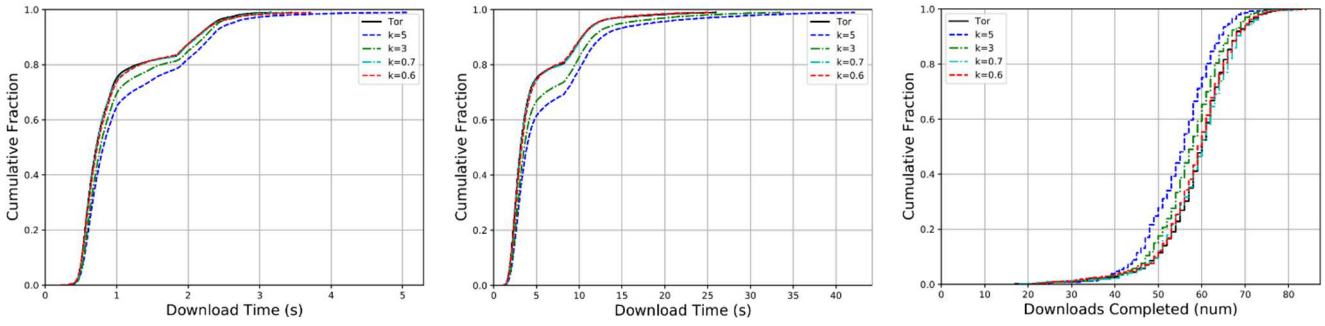
IV. EXPERIMENT RESULTS

This section describes the results obtained in the simulation. We compare performance from Shadow results and calculate the Gini coefficient to observe anonymity.

A. Performance Results

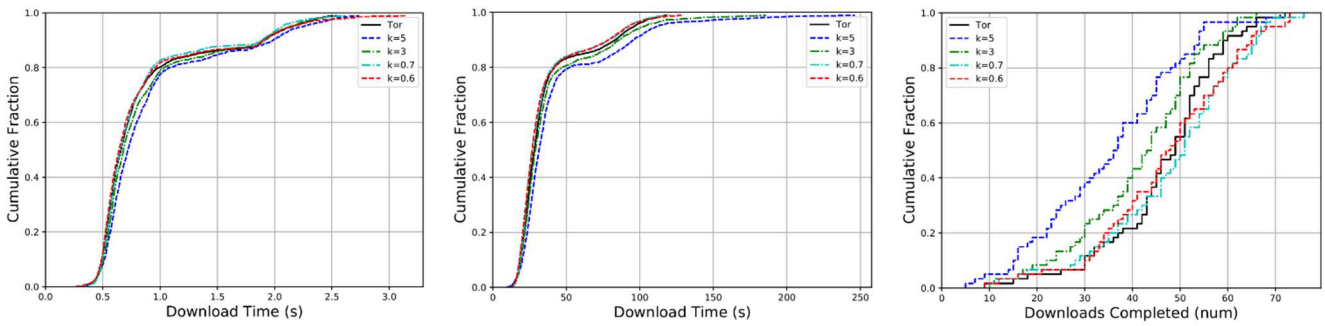
In the homogeneous network, we set $k = \{0.6, 0.7\}$ represents different high-bandwidth demands, and $k = \{3,5\}$ represents different low-bandwidth demands. The results of performance for the web browsing client are shown in Fig. 3 and for the bulk client in Fig. 4. In terms of download time of the first byte of a file in Fig. 3a and Fig. 4a, the download time with a smaller value of k is close to or slightly better than the result of the Tor network. As k increases, the download time is slightly longer. From the perspective of completion time of a file download in Fig. 3b and Fig. 4b, it has similar results with the first byte download time. From the point of view of the number of file downloads in Fig. 3c and Fig. 4c, Fig. 4c shows that the client can download more files when k is small, and the download amount is less when k is large.

In the heterogeneous network, we set $k = \{0.4,0.5,0.6\}$ represents different high-bandwidth requirements, and $k = \{3,4,5\}$ represents different low-bandwidth requirements. The performance results of the experiment are shown in Fig. 5. From the view of download time of the first byte of a file in Fig. 5a and completion time of a file in Fig. 5b, the download time is shorter than the Tor network when k is small, and the download time becomes longer as k increases. From the perspective of the file download completion amount in Fig. 5c, the download completion amount is more when k is small, and the download completion amount is less when k is large.



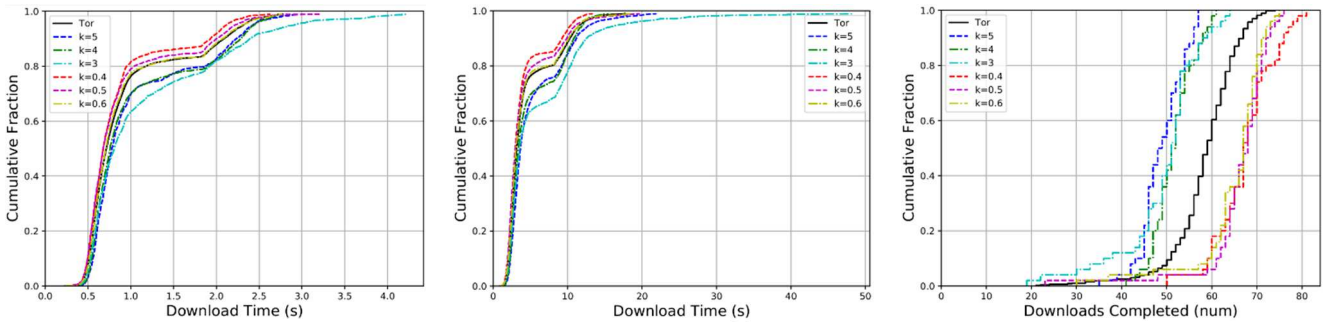
(a) Time to download the first byte of a file (b) Time to download 32KB of a file (c) Number of downloads completed for each client

Fig. 3. Performance of web browsing clients in the homogeneous network



(a) Time to download the first byte of a file (b) Time to download 5MB of a file (c) Number of downloads completed for each client

Fig. 4. Performance of bulk clients in the homogeneous network



(a) Time to download the first byte of a file (b) Time to download 32KB of a file (c) Number of downloads completed for each client

Fig. 5. Performance of web browsing clients in the heterogeneous network

B. *Anonymity Results*

In the homogeneous network, the Gini coefficient results comparing $k = \{0.6, 0.7, 3, 5\}$ and the Tor algorithm are shown in Table II. In the heterogeneous network, the Gini coefficient results comparing $k = \{0.4, 0.5, 0.6, 3, 4, 5\}$ and the Tor algorithm are shown in Table III. Our experimental results show that the Gini coefficient is greater than that of the Tor network when k is small, and the Gini coefficient is smaller than that of the Tor network when k is large. Therefore, the anonymity of the network will decrease when the clients prefer to choose more high-bandwidth relays, while the anonymity can be enhanced when the clients prefer to choose more low-bandwidth relays.

We found that smaller k results in lower network anonymity. So the dynamic parameter should not be set too small. At the same time, it is found that the Gini coefficient increases with the increase of k. The reason for this phenomenon is that the network anonymity decreases when the circuit selection tends to have more low-bandwidth relays, so the dynamic parameter should not be set too large.

TABLE II. GINI COEFFICIENT OF HOMOGENEOUS NETWORK

	Tor	$k = 0.6$	$k = 0.7$	$k = 3$	$k = 5$
Gini Coefficient	0.674	0.702	0.698	0.577	0.626

TABLE III. GINI COEFFICIENT OF HETEROGENEOUS NETWORK

	Tor	$k = 0.4$	$k = 0.5$	$k = 0.6$	$k = 3$	$k = 4$	$k = 5$
Gini Coefficient	0.7	0.846	0.815	0.769	0.616	0.667	0.695

V. CONCLUSION

This research proposes a Tor relay dynamic selection algorithm based on client bias. Depending on the dynamic parameter sets by the client, the circuit tend to choose more high-bandwidth relays or low-bandwidth relays to achieve partial improvements in performance or anonymity. Performance can be improved when the dynamic parameter is small. However, anonymity is reduced. Anonymity can be improved when the dynamic parameter is large. However, performance is reduced. The experimental results also show that the dynamic parameter should not be set too large or too small, because it will reduce the anonymity of the client.

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: the Second-Generation Onion Router. *In Proc. of the 13th conference on USENIX Security Symposium*, 2004.

[2] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker. Low-Resource Routing Attacks against Tor. *In Proc. of the 2007 ACM workshop on Privacy in electronic society*, 2007.

[3] C. Tang and I. Goldberg. An Improved Algorithm for Tor Circuit Scheduling. *In Proc. of the 17th ACM conference on Computer and communications security,* 2010.

[4] T. Wang, K. Bauer, C. Forero, and I. Goldberg. Congestion-Aware Path Selection for Tor. *In International Conference on Financial Cryptography and Data Security,* 2012.

[5] M. Alsabah, K. Bauer, T. Elahi, and I. Goldberg. The Path Less Travelled: Overcoming Tor's Bottlenecks with Traffic Splitting. *In International Symposium on Privacy Enhancing Technologies Symposium,* 2013.

[6] R. Jansen, N. Hopper, and Y. Kim. Recruiting new Tor relays with BRAIDS. *In Proc. of the 2010 ACM Conference on Computer and Communications Security,* 2010.

[7] R. Dingledine and D. S. Wallach. Building incentives into Tor. *In International Conference on Financial Cryptography and Data Security,* 2010.

[8] R. Jansen, A. Johnson, and P. Syverson. LIRA: Lightweight Incentivized Routing for Anonymity. *In Proc. of the Network and Distributed System Security Symposium,* 2013.

[9] Y. Lei and F. Li. mTor: A multipath Tor routing beyond bandwidth throttling. *In 2015 IEEE Conference on Communications and Network Security,* 2015.

[10] R. Snader and N. Borisov. A Tune-up for Tor: Improving Security and Performance in the Tor Network. *In Proc. of the Network and Distributed System Security Symposium,* 2008.

[11] M. Akhoondi, C. Yu, and H. V. Madhyastha. LASTor: A low-latency AS-aware Tor client. *In 2012 IEEE Symposium on Security and Privacy,* 2012.

[12] C. Wacek, H. Tan, K. S. Bauer, and M. Sherr. An Empirical Evaluation of Relay Selection in Tor. *In Proc. of the Network and Distributed System Security Symposium,* 2013.

[13] M. Imani, M. Amirabadi, and M. Wright. Modified relay selection and circuit selection for faster Tor. *IET Communications,* 13(17): 2723-2734, 2019.

[14] R. Dingledine, N. Hopper, G. Kadianakis, and N. Mathewson. One Fast Guard For Life (or 9 Months). *In Proc. of 7th Workshop on Hot Topics in Privacy Enhancing Technologies,* 2014.