entropy

MDPI

Article

A Traffic Splitting Algorithm for Load Balancing in Tor

Xiance Meng [1,2] and Mangui Liang [1,2,*]

[1] Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China; mengxiance@bjtu.edu.cn
[2] Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing Jiaotong University, Beijing 100044, China
* Correspondence: mgliang@bjtu.edu.cn

Abstract: As the most popular anonymous communication system, Tor provides anonymous protection for users by sending their messages through a series of relays. Due to the use of the bandwidth-weighted path selection algorithm, many more users choose routers with high bandwidth as relays. This will cause the utilization of high bandwidth routers to be much higher than that of low bandwidth routers, which will bring congestion risk. The Quality of Service (QoS) is difficult to guarantee for users who need delay-sensitive services such as web browsing and instant messaging. To reduce the average load of routers and improve the network throughput, we propose a circuit construction method with multiple parallel middle relays and conduct a dynamic load allocation method. The experiment demonstrates that our proposed method can provide better load balancing. Compared with other multipath anonymous communication networks, our proposed method can provide better anonymity.

Keywords: anonymous network; Tor; anonymity; load balancing

Citation: Meng, X.; Liang, M. A Traffic Splitting Algorithm for Load Balancing in Tor. *Entropy* **2022**, *24*, 807. https://doi.org/10.3390/e24060807

Received: 20 April 2022
Accepted: 7 June 2022
Published: 9 June 2022

1. Introduction

With the rapid development of Internet technology, more and more Internet users pay attention to user privacy. Various anonymous communication systems have been proposed in recent years, such as the Anonymizer [1], DC-Net [2], Crowds [3], Tarzan [4], LAP [5], HORNET [6], TARANET [7], and Tor [8]. The Second-Generation Onion Router (Tor) is the most widely used anonymous communication system with low latency, which is a distributed overlay network developed based on the existing general Internet. It encrypts the transmitted data in layers through a series of relays and transmits data to the receiver. Each node in the anonymous path only knows its predecessor node and successor node but does not know the information of other nodes in the path, thus protecting the anonymity of the connection. However, for delay-sensitive applications, such as web browsing and instant messaging, the delay of Tor leads to poor user experience [9]. Furthermore, this may cause some users to exit, which will reduce the size of the anonymity set, thus affecting the anonymity for all users [10,11].

Recent studies have found that the following factors mainly cause Tor's delay. First, users using video streaming applications, P2P applications, and other applications occupy a lot of bandwidth resources [12]. Second, Tor has a longer path and requires multiple encryptions and decryptions compared with the typical network. The last is that the performance of nodes in Tor is unreliable [13].

We find that the layered encryption and decryption scheme can guarantee Tor's security based on the above aspects. Furthermore, Tor is the most popular system for providing anonymity on the Internet. It is normal for many users to use high-bandwidth applications. In order to improve the performance of Tor, Mohsen et al. [14] proposed a path selection algorithm considering the geographical location information and bandwidth of nodes to reduce the delay. Armon et al. [15] proposed PredicTor, which can dynamically avoid selecting congested nodes and long-distance paths. However, these methods will

bring a small amount of anonymity loss. Recently, Tor has adopted a new method to deal with congestion [16,17]. This method will track the RTT measurement value of each circuit and compare it with the threshold to determine whether it is congested and update the path. This method can improve Tor's load balancing and congestion problems. However, this still cannot change that low-bandwidth nodes have poor load capacity. Therefore, we consider using a multipath method to solve this problem. In Tor, multipath routing has the following advantages:

- Improve load balancing [18]. Using multipath simultaneously can reduce the load assigned to each OR.
- Increase throughput [19]. The throughput of multipath users can achieve up to the sum throughput of all circuits, which is greater than the throughput of a single path.

We present TSMMR, a traffic splitting mechanism with multiple middle relays in parallel. The traffic sent by the sender splits into several streams by entry relay and will forward in parallel through multiple middle relays. Finally, the traffic will combine in the exit relay. We only have one entry relay and one exit relay at the two ends. The minimum bandwidth of each node on each path is the path's capacity, and the load of each path is allocated according to its capacity. We also propose a new performance evaluation metric, and, according to the evaluation, our method can reduce network utilization compared with Tor and thus reduce the load on nodes. Conflux [20], mTor [21], and TrafficSliver [22] have used multipath in Tor in recent years. Among them, Conflux and mTor can improve the performance in Tor. TrafficSliver involves a multipath method to resist website fingerprinting attacks, but it sacrificed some bandwidth and latency overheads. Therefore, we only compare the performance and anonymity with those of Conflux and mTor in this paper.

This work's significant contributions may be summarized as follows:

(1) Better balance of the utilization of various nodes to effectively improve the congestion problem of high-bandwidth nodes, as well as the problem of bandwidth scarcity and reducing the load of high-bandwidth nodes.
(2) A traffic splitting algorithm is proposed.
(3) A new performance evaluation metric is proposed for performance analysis.
(4) Anonymity can be guaranteed.

The rest of this paper is organized as follows. In Section 2, we present the related work. Then, in Section 3, we present our proposed approach. Next, we show our performance evaluation in Section 4. We analyze the anonymity in Section 5. Finally, we conclude our work in Section 6.

2. Related Work

2.1. Tor

As the most popular anonymous communication network with low latency, Tor usually chooses three nodes as relays to construct a path from the client to the destination, commonly called a circuit. These three kinds of nodes are Onion Proxies (OPs), Onion Routers (ORs), and Directory Servers (DSs). As shown in Figure 1, OPs run on a user's machine to fetch directories and construct circuits across the network. ORs of the circuit are responsible for relaying traffic to destinations or other relays. As a group of trusted and reliable servers, Directory Servers are deployed in the Tor network as centralized, and are responsible for collecting each OR's IP address, public key, policies, and bandwidth value of the OR. Generally, as the first router (entry guard), they can protect Tor from traffic analysis attacks, including predecessor attacks, statistical analysis attacks, and passive AS level association attacks. In order to enhance these resistances, Tor increases the cycle of rotating entry guards to 9 months and changes from using three entry guards to a single, fast entry guard [23]. The original definition of fast entry guard is higher than the median bandwidth or 250 kB/s. Now, the threshold is increased to 2 MB/s. The strategy for Tor to select other nodes, such as middle relay and exit nodes, is through the bandwidth-weighted

path selection algorithm. For example, the bandwidth of node i is B_i. Then, the probability of the node being selected is:

$$Q_i = \frac{B_i}{\sum_{i=1}^{n} B_i}. \quad (1)$$

The number of nodes is n. The strategy for Tor to select other nodes is through the bandwidth-weighted path selection algorithm.

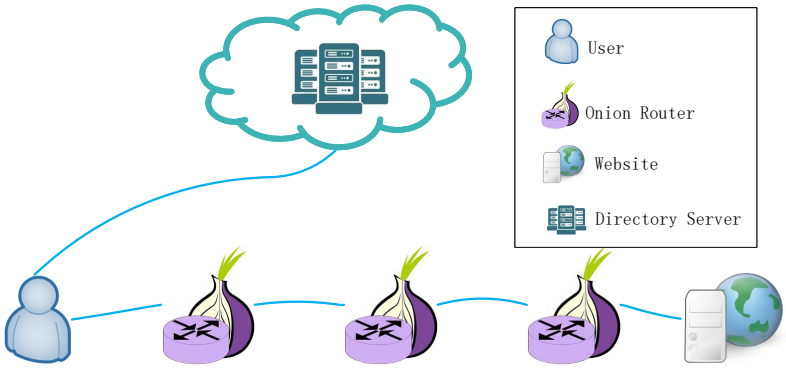

Figure 1. Tor architecture.

2.2. Conflux

Conflux [20] and mTor [21] are extensions to vanilla Tor that utilize multiple paths to improve the user experience. In Conflux, the OP creates two or more circuits with the same exit OR. The client will split traffic and transmit it across multiple circuits, dynamically measuring the throughput and latency of each circuit. If the throughput or the latency on a path is high, the client will reduce the load on that path. Since the split traffic will arrive out of order, the client adds a 4-byte sequence number to each cell. Finally, the traffic is reordered and combined by the exit OR. This method requires incremental deployment to upgrade the client and exit OR. It improves performance for low-bandwidth clients such as bridged users. However, since Conflux uses more nodes than Tor, it will have a higher path compromise rate.

2.3. mTor

In addition, mTor also involves a multipath anonymous communication network similar to Conflux. However, the number of circuit configurations is m as the parameter. In mTor, it selects a group of new low-bandwidth routers as relays, constructs multiple circuits to form anonymous tunnels for bulk data transmission, and uses an active congestion detection mechanism to prevent slow circuits from becoming the bottleneck of the whole tunnel to improve the performance of bulk data transmission.

3. Traffic Splitting Mechanism with Multiple Middle Relays in Parallel

Low-resource routing attack is an irresistible attack for Tor [24]. When both the entry and exit nodes on the anonymous path are malicious nodes, an adversary can collude with them to compromise Tor's anonymity. We find the nodes with high bandwidth are much more utilized than nodes with low bandwidth as most users tend to choose stable and fast nodes as Tor routers [25]. This causes two problems. The first is that more users select high-bandwidth nodes, so their bandwidth utilization and load will be high, resulting in risks such as congestion and increased delays. Second, nodes deployed or claimed by the adversary to have high bandwidth will be more likely to be selected, making the probability of both entry and exit nodes being malicious nodes higher. Therefore, we consider improving the utilization of low-bandwidth nodes to solve the above two problems. However, due to the poor capacity of low-bandwidth nodes, the multipath method should be considered to split the load. Next, we will describe the details of our proposed TSMMR.

3.1. Circuit Construction

In Figure 2, our proposed anonymous communication network has m paths with different middle relays. Common circuit construction strategies include random selection of nodes [26], geographical selection of nodes [27], and the bandwidth-weighted path selection algorithm [28]. The random selection of nodes strategy has high anonymity but cannot ensure bandwidth, so it is not easy to provide QoS to users. The geographical selection of nodes strategy makes it easy to expose the geographical location of users, thus reducing the anonymity of the system. In TSMMR, traffic is transmitted through multiple paths. Where the entry node and exit node are aggregation nodes for the traffic, we prefer to select high-bandwidth entry nodes and exit nodes to construct circuits. We propose a hybrid node selection strategy to meet the needs of different relay nodes. Next, we will first present our proposed node selection strategy.

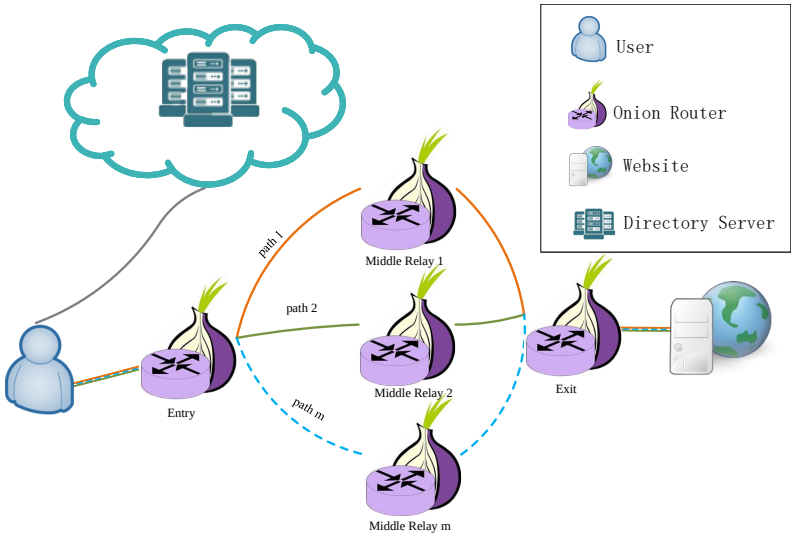

Figure 2. TSMMR architecture. The user splits the traffic and transmits it through m different paths to the website.

Selection of entry guard. The entry node is the first hop on the anonymous communication path, and this node knows the identity information of the sender. If an adversary controls the entry node, it will directly reveal the sender's identity. Research shows that rotating the entry node too frequently will make it easier to select the node controlled by the adversary as the entry node [29]. Therefore, nodes are generally rotated infrequently to defend against end-to-end correlation attacks and can be retained for 60 days to 9 months. Thus, the uptime of the entry node should be higher than that of most routers. In addition, the node will have a high probability of being used multiple times during long-term running. Therefore, we choose nodes above the median bandwidth (currently about 2MB/s). The user first downloads the information of the alternative nodes from the DS and excludes from it the nodes that do not satisfy the above conditions. Subsequently, the node is selected as the entry node using the bandwidth-weighted path selection algorithm. The specific description of the bandwidth-weighted path selection algorithm can be seen in Algorithm 1. Suppose there are n ORs and the bandwidth of a node is B_i. According to Formula (1), the probability of this node being selected as an entry node is Q_i.

Selection of exit node. The exit node is the node on the anonymous communication path that knows the receiver's identity information and is closely related to the receiver's anonymity guarantee. In addition, this node is also the aggregation node in TSMMR. In order not to decrease the throughput on the communication path, the bandwidth of the exit node should be higher than the median bandwidth. After excluding the nodes that do not meet the requirements, the user selects the exit node using the bandwidth-weighted path selection algorithm.

Selection of middle nodes. The entry and exit nodes of Tor are quite important. However, the role of middle relays on the anonymous communication path is also crucial. When a middle relay fails due to insufficient bandwidth, the sender will reselect a middle relay to construct a new communication path. However, it will increase the communication cost and increase the communication delay. Therefore, we consider choosing multiple middle relays, where each middle relay and the entry and exit nodes can construct a circuit and multiple circuits multiplex the entry and exit nodes. The multiple circuit construction method improves the network throughput on both sides of the communication and allows for fast traffic redistribution in case of failure of the middle relay node. The bandwidth of this node need not be higher than the median bandwidth. Using a bandwidth-weighted path selection algorithm in Algorithm 1, the user can select the middle relay directly among the running nodes.

Algorithm 1 Bandwidth-Weighted Path Selection Algorithm.

Require: A list of nodes *node_list* fetched from Directory Servers
Ensure: A chosen *router*
1: **for** $i \leftarrow 0$ to *node_list.size* **do**
2: $B_i \leftarrow node\_list[i].bw$
3: $bw\_list \leftarrow bw\_list \cup B_i$
4: $BW \leftarrow BW + B_i$
5: **end for**
6: $Rnd \leftarrow rand()\%BW$
7: **while** $T < Rnd$ **do**
8: $T \leftarrow T + bw\_list_i$
9: $i \leftarrow i + 1$
10: **end while**
11: **return** $router \leftarrow node\_list[i]$

3.2. Traffic Splitting

Since we construct circuits using multiple middle relays with different bandwidths, the average traffic splitting may lead to bandwidth redundancy in high-bandwidth relays and congestion in low-bandwidth relays. We propose an adaptive traffic splitting method in a round-robin fashion to solve this problem. Assume that the size of each data stream sent by the sender is D. When the sender first initiates a communication request, it asks for the used and advertised bandwidths of the middle relays in turn. As a result, the used bandwidths of the middle relays are $\{Bu_1, Bu_2, \ldots, Bu_m\}$. The advertised bandwidths of the middle relays are $\{B_1, B_2, \ldots, B_m\}$, where m is the number of paths, and B is the sum of the bandwidths of all middle relays. We can see the details in Algorithm 2.

$$B = \sum_{i=1}^{m} B_i. \tag{2}$$

For each circuit $C_i (i \in [1, m])$, the allocated traffic is:

$$D_i = D * \frac{B_i}{B}. \tag{3}$$

As the number of users grows, the probability of a circuit being multiplexed increases. Thus, it increases the risk of node congestion. We assume that the used bandwidth of a middle relay R_t is Bu_t, the advertised bandwidth is B_t, and the traffic allocated to this node is D_t. If there exists

$$Bu_t + D_t > B_t, \tag{4}$$

we consider that the middle relay R_t is no longer suitable for constructing circuits for an anonymous communication network. Therefore, the traffic needs to be reallocated, where the node R_t will be allocated traffic of $D_t = 0$. For other circuits $C_i (i \in [1, m], i \neq t)$, the

traffic is still allocated in proportion to the bandwidth. The number of paths is $m-1$ until the user finds a new middle relay. Our way of coping with node failures enables timely data transmission using other paths in case of node failure or congestion. Compared with a single path, it saves the time spent re-finding nodes to construct circuits. Our traffic splitting approach can provide better load balancing by allocating corresponding loads to nodes with different bandwidths in the network.

Due to the multipath routing scheme, the traffic will split into different streams along different paths. The sender adds the sequence number to each stream, and the receiver can reorder the traffic according to the sequence number to obtain the complete data. In addition, the exit OR is responsible for buffering the out-of-order data.

Algorithm 2 Traffic Splitting Algorithm.

Require: A list of middle relays *middle_list* and the traffic *LD* sent by the sender
Ensure: Traffic *ld_list* which is allocated at each relay
1: **for** $i \leftarrow 0$ to m **do**
2: $B_i \leftarrow middle\_list[i].bw$
3: $B \leftarrow B + B_i$
4: **end for**
5: **for** $i \leftarrow 0$ to m **do**
6: $ld\_list[i] \leftarrow LD * B_i / B$
7: **end for**
8: **return** *ld_list*

4. Performance Evaluation

We conduct a simulation to evaluate the performance improvements of our proposed method. The results show that our method can improve reducing the network utilization of most nodes, and it will provide benefits for load balancing.

4.1. Performance Metrics

In general, high-bandwidth routers are capable of carrying more traffic. As a router carries high traffic, its capacity to process data will decrease, which increases latency and thus increases the risk of congestion on the path. Therefore, we use the network utilization of nodes as a metric to evaluate network scalability. We conducted a simulation and compared it with other anonymous networks. The results show that our proposed TSMMR can carry more traffic load and has better scalability.

4.2. Simulation Design

Our simulation is implemented on OMNeT++ [30]. We design three types of nodes: user, website, and Onion Router. The Onion Router contains the entry node, the middle relay, and the exit node. The Onion Proxy runs on the user node and is responsible for selecting nodes and constructing circuits in the anonymous communication network. The user sends traffic to the website through the OR. We simulate Conflux, mTor, Tor, and TSMMR using different circuit construction methods.

To make the simulation environment close to the actual network environment, we download Tor's consensus network status document as the original data for our simulation [31]. This document records the bandwidth and online time of the nodes in Tor.

We use different path selection algorithms to select nodes from these nodes for constructing circuits. Different types of applications have different bandwidth requirements. To make the experiments more convincing, we set different sending rates for the user nodes to simulate different kinds of user requirements. For example, the lower sending rate can simulate applications such as real-time streaming. The higher sending rate can simulate applications such as video browsing and large file downloads. In Tor, when communication initiates, we use the send rate as the load for each Onion Router since there is only one communication path. In other multipath anonymous communication networks, the load

on each node is calculated based on different traffic splitting algorithms. Then, based on the load and bandwidth of each node, we can obtain the bandwidth utilization of each node, which is the ratio of the total load to the bandwidth on that node.

As the scale of users increases, there will be many users running simultaneously, so there will be some ORs being multiplexed. When the bandwidth utilization of an OR is higher, it means that the performance of this node is more severely affected. Under the same conditions, we believe that a network has better scalability when the percentage of nodes with higher bandwidth utilization is low. We simulate the bandwidth utilization of the whole network when 10,000 users use different anonymous networks separately.

4.3. Results

Because the anonymous network has a congestion control mechanism, in the actual simulation process, we found that only a few nodes with very low bandwidth are prone to the situation that the bandwidth occupation cannot meet the demand. Therefore, we compared the cumulative fraction for different bandwidth utilization. Where bandwidth utilization is the ratio of an OR's used bandwidth to its total bandwidth, the cumulative fraction is the number of ORs less than the specified bandwidth utilization as a percentage of the total number of ORs. Generally speaking, anonymous networks with a higher cumulative fraction with lower bandwidth utilizations have higher scalability.

As shown in Figure 3, we compare Tor with mTor, Conflux, and TSMMR. Our method has a higher cumulative fraction at low bandwidth utilization, which means fewer nodes with high bandwidth utilization in TSMMR. It also demonstrated that our method has better scalability and can accommodate more users with better congestion control.

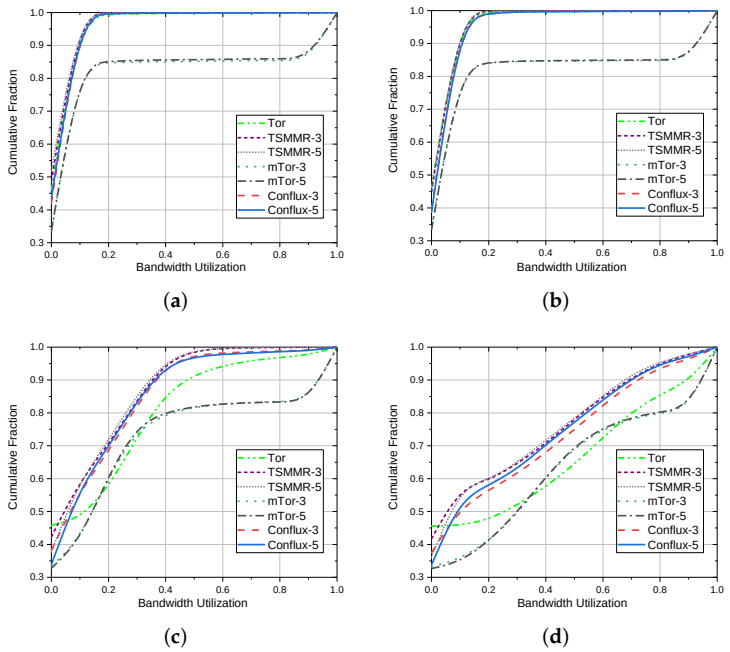

Figure 3. Cumulative fraction of node bandwidth utilization for different anonymous communication networks with different bandwidth requirements. (**a**) 50 kB/s sending rate. (**b**) 100 kB/s sending rate. (**c**) 500 kB/s sending rate. (**d**) 1000 kB/s sending rate.

We define nodes with bandwidth utilization below 30% as light-load nodes and nodes with bandwidth utilization above 80% as high-load nodes. High-load nodes are more likely to cause link congestion due to high latency. In Tables 1–4, we can see that the percentage of light-load nodes is higher in TSMMR than in other anonymous communication networks. Moreover, the percentage of high-load nodes in TSMMR is lower than that of other anony-

mous communication networks. Therefore, we can prove that TSMMR has better load balancing capability and can provide better congestion control for users.

Table 1. The percentage of nodes in different anonymous communication networks in different bandwidth utilization ranges when the sending rate is 50 kB/s.

Bandwidth Utilization	Tor	TSMMR-3	TSMMR-5	mTor-3	mTor-5	Conflux-3	Conflux-5
0∼30%	99.7%	100%	100%	84.9%	85.5%	99.8%	99.8%
30%∼80%	0.2%	0	0	0.5%	0.5%	0.1%	0.1%
80%∼100%	0.1%	0	0	14.6%	14%	0.1%	0.1%

Table 2. The percentage of nodes in different anonymous communication networks in different bandwidth utilization ranges when the sending rate is 100 kB/s.

Bandwidth Utilization	Tor	TSMMR-3	TSMMR-5	mTor-3	mTor-5	Conflux-3	Conflux-5
0∼30%	99.7%	100%	100%	84.6%	84.6%	99.6%	99.6%
30%∼80%	0.2%	0	0	0.3%	0.4%	0.2%	0.2%
80%∼100%	0.1%	0	0	15.1%	15%	0.2%	0.2%

Table 3. The percentage of nodes in different anonymous communication networks in different bandwidth utilization ranges when the sending rate is 500 kB/s.

Bandwidth Utilization	Tor	TSMMR-3	TSMMR-5	mTor-3	mTor-5	Conflux-3	Conflux-5
0∼30%	72.7%	84%	85.7%	74.9%	76.2%	82.4%	83.7%
30%∼80%	24.1%	15.9%	14.3%	8.6%	7.2%	16.4%	14.9%
80%∼100%	3.2%	0.1%	0	16.5%	16.6%	1.2%	1.4%

Table 4. The percentage of nodes in different anonymous communication networks in different bandwidth utilization ranges when the sending rate is 1000 kB/s.

Bandwidth Utilization	Tor	TSMMR-3	TSMMR-5	mTor-3	mTor-5	Conflux-3	Conflux-5
0∼30%	52%	64.5%	64.9%	49.3%	49.8%	61.4%	62.9%
30%∼80%	33.7%	30.7%	30.6%	30.7%	30.5%	32.5%	32%
80%∼100%	14.3%	4.8%	4.5%	20%	19.7%	6.1%	5.1%

5. Anonymity Analysis

The anonymous communication networks mentioned in this paper are all based on Tor. When the adversary controls both ends of the circuit, the anonymity of the circuit may be compromised by time analysis. The adversary we have assumed depends upon the threat model proposed by Syverson et al. [32]. The entry node knows the client's IP address in anonymous communication networks, while the exit node knows the server's IP address. When the adversary controls both nodes, the adversary can use traffic analysis to confirm the communication relationship of the communication, thereby destroying the anonymity of the link [24,33]. In this section, we first introduce the threat model and compare the potential for path compromise under a given adversary with other anonymous communication networks. Finally, we compare the anonymity degree of these anonymous communication networks.

5.1. Threat Model

Tor is deployed on a real network and is the most popular anonymous communication network in the world. As a result, it is difficult to avoid some malicious adversaries to evaluate the anonymity of anonymous communication systems and resist some adversaries. First, we need to define the capabilities of the adversary. We assume that some of the Tor routers are controlled by the adversary. As shown in Figure 4, the OR compromised by the

adversary can observe the network traffic, so the adversary can apply timing analysis [34,35] to destroy the network's anonymity. However, traffic will not be modified, deleted, or delayed. In addition, the percentage of routers or bandwidth controlled by these malicious adversaries cannot exceed 20% [20].

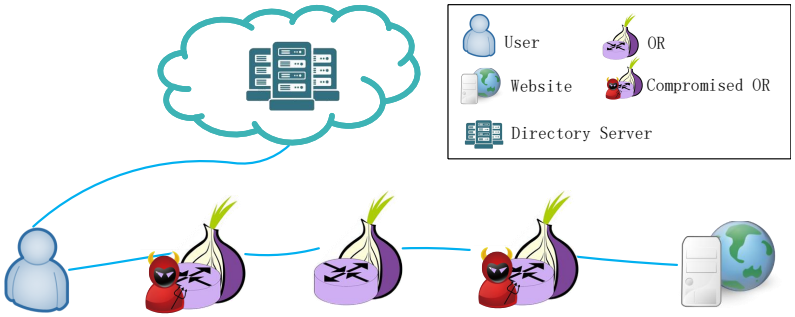

Figure 4. Threat model: The adversary controls the entry node and exit node on the path.

5.2. Path Compromise

In Tor, every relay can only obtain the node information before and after. Therefore, only the entry node can identify the sender, and only the exit node can identify the receiver. When both the entry node and the exit node are under the adversary's control, the anonymous communication network is considered compromised. We define $P(Compromised)$ as the probability that an entry node and an exit node on an anonymous communication path are controlled by an adversary simultaneously.

In Tor, we can calculate $P(Compromised)$ as follows:

$$P(Compromised) = f_{xbw} \cdot f_{gbw}. \tag{5}$$

Here, f_{gbw} is the proportion of the bandwidth of the entry nodes controlled by the adversary to the total bandwidth of Tor, f_{xbw} is the proportion of the bandwidth of the exit nodes controlled by the adversary to the total bandwidth of Tor. As Conflux and mTor have multiple entry nodes, the compromised probability is different from Tor.

In Conflux and mTor, we can calculate $P(Compromised)$ as follows:

$$P(Compromised) = f_{xbw} \cdot (1 - (1 - f_{gbw})^m). \tag{6}$$

Here, m is the number of entry nodes used in the multipath, f_{xbw} and f_{gbw} have the same definition as above. Like Tor, TSMMR has only one entry node and one exit node. As a result, as Formula (5) shows, the $P(compromised)$ of TSMMR is the same as Tor. In addition, mTor and Conflux have a similar $P(compromised)$.

The comparison of $P(compromised)$ between the several anonymous communication networks mentioned here is shown in Figure 5. The probability of being compromised increases with the adversary controlling more nodes. Whether the multipath number is 3 or 5, both Conflux and mTor have higher $P(Compromised)$ than Tor and TSMMR.

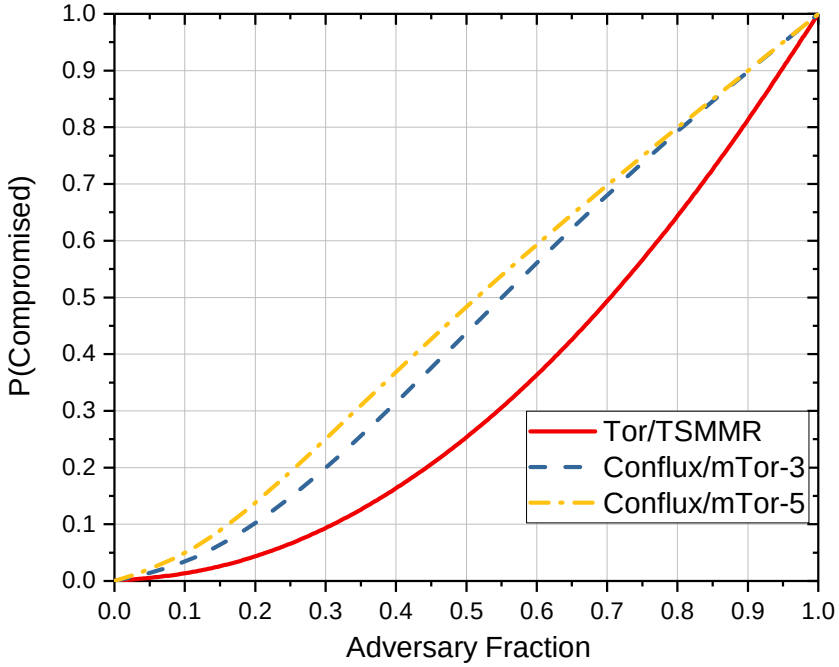

Figure 5. Compromise probability for different anonymous communication networks.

5.3. Anonymity Degree

As mentioned above, we analyze the anonymity of anonymous networks primarily for adversaries who can control some of the network nodes. To be able to adopt a more general manner of describing anonymity in anonymous networks, we define the anonymity degree based on entropy [36].

In Tor and TSMMR: We assume that there are N nodes in the anonymity network. Then, the maximum value of the anonymity set is N for the adversary. Suppose the adversary cannot exclude any node in the anonymity set. Then, the maximum entropy of N users is:

$$H_M = -\sum_{i=1}^{N} p_i \cdot \log_2 p_i. \tag{7}$$

Here, M is the maximum value of the anonymity set, and p_i is the probability that an adversary can identify a node as a sender. If p_i obeys a uniform distribution, that is,

$$p_i = \frac{1}{N}, \tag{8}$$

then we can obtain:

$$H_M = \log_2 N. \tag{9}$$

Subsequently, the adversary may exclude some improbable nodes through traffic analysis, timing attacks, and other attack methods. Then, the size of the new anonymity set is S.

At this time, $H_S = \log_2 S$. The anonymity degree is:

$$D_1 = 1 - \frac{H_M - H_S}{H_M} = \frac{H_S}{H_M} = \frac{\log_2 S}{\log_2 N}. \tag{10}$$

Suppose the adversary fails to exclude any node, i.e., $N = S$. Then, the degree of anonymity is 1, indicating that the anonymous communication system has the most significant anonymity. On the other hand, suppose the adversary can determine the sender's identity, i.e., $S = 1$. Then, we can obtain that the degree of anonymity is 0, indicating that the anonymous communication system is compromised.

In mTor and Conflux: For multipath anonymous communication systems such as Conflux and mTor, we let N be the size of the anonymity set. Then, the maximum entropy of N users is:

$$H_M = -\sum_{i=1}^{N} p_i \cdot \log_2 p_i. \tag{11}$$

For Conflux or mTor with m paths, it contains m entry nodes and 1 exit node. Then, the probability that the adversary determines that the user is the sender is:

$$p_i = \frac{m}{N}. \tag{12}$$

Then, the maximum entropy is:

$$H_M = -\sum_{i=1}^{N} p_i \cdot \log_2 p_i = m \cdot \log_2 \frac{N}{m}. \tag{13}$$

The adversary can reduce the anonymity set by excluding some unlikely nodes through traffic analysis, timing attacks, and other attacks. The size of the new anonymity set is S. At this time,

$$H_S = m \log_2 \frac{S}{m}. \tag{14}$$

We can obtain the anonymity degree:

$$D_2 = \frac{H_S}{H_M} = \frac{\log_2 S - \log_2 m}{\log_2 N - \log_2 m} \tag{15}$$

We assume that there are 10,000 nodes in each anonymous communication network, and the maximum anonymity set is 10,000. As Figure 6 shows, we compare the anonymity degree of different anonymous communication networks when the anonymity set increases. When the multipath number $m > 1$, it is evident that Tor and TSMMR have a more considerable anonymity degree than Conflux and mTor.

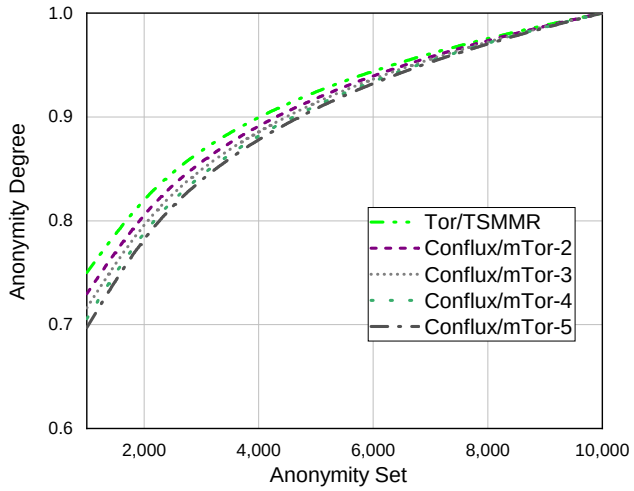

Figure 6. Anonymity degree of different anonymous communication networks.

6. Conclusions

This paper aims to reduce the potential for congestion of anonymous communication networks. We use multiple middle relays to split the traffic and transmit it in parallel through multiple paths. Furthermore, we use low-bandwidth nodes as middle relays to transmit traffic. We introduce TSMMR's design and compare its performance with other anonymous communication networks such as Tor, mTor, and Conflux. For the entry nodes, middle relays, and exit nodes on the anonymous communication path, we use different

node selection strategies to select them respectively. Furthermore, we use different sending rates to simulate different types of users. The results show that the percentage of nodes with lower bandwidth utilization is larger in TSMMR than in other anonymous communication networks. Conversely, the portion of nodes with higher bandwidth utilization is smaller in TSMMR than in other anonymous communication networks. The results are the same when the multipath number is varied, indicating that TSMMR can provide better load balancing than other anonymous communication networks. In addition, we also compare the probability of anonymous communication paths being compromised (the adversary controls the entry and exit nodes at the same time). Assuming that the adversary cannot control all nodes, TSMMR and Tor have the same compromise probability and are better than Conflux and mTor. Finally, we evaluated the anonymity of different anonymous communication networks more generally. We calculate the anonymity degree of these anonymous communication networks. The results show that TSMMR can provide the same anonymity degree as Tor. Moreover, when the set of users is constant, the smaller the anonymity set is, and the better anonymity TSMMR an provide than other multipath anonymous communication networks.

Author Contributions: Conceptualization, X.M. and M.L.; methodology, X.M. and M.L.; software, X.M.; writing and editing, X.M.; analysis, X.M. and M.L.; investigation, X.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National High Technology Research and Development Program of China (863 Program) under Grant No. 2007AA01Z203 and the Joint Project of the National Natural Science Foundation of China under Grant No. U1636109.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Boyan, J. The Anonymizer. Computer-Mediated Communication Magazine. 1997. Available online: http://www.cs.cmu.edu/afs/cs/user/jab/web/cv/pubs/boyan.anonymizer.pdf (accessed on 25 May 2022).
2. Chaum, D. The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptol.* **1988**, *1*, 65–75. [CrossRef]
3. Reiter, M.K.; Rubin, A.D. Crowds: Anonymity for Web transactions. *ACM Trans. Inf. Syst. Secur.* **1998**, *1*, 66–92. [CrossRef]
4. Freedman, M.J.; Sit, E.; Cates, J.; Morris, R. Tarzan: A peer-to-peer anonymizing network layer. In Proceedings of the ACM Conference on Computer and Communications Security, Washington, DC, USA, 18–22 November 2002; Volume 2429, pp. 193–206. [CrossRef]
5. Hsiao, H.C.; Kim, T.H.J.; Perrig, A.; Yamada, A.; Nelson, S.C.; Gruteser, M.; Meng, W. LAP: Lightweight anonymity and privacy. In Proceedings of the 2012 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–23 May 2012; pp. 506–520. [CrossRef]
6. Chen, C.; Asoni, D.E.; Barrera, D.; Danezis, G.; Perrig, A. HORNET: High-speed onion routing at the network layer. In Proceedings of the ACM Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; Volume 2015, pp. 1441–1454. [CrossRef]
7. Chen, C.; Asoni, D.E.; Perrig, A.; Barrera, D.; Troncoso, C. TARANET: Traffic-Analysis Resistant Anonymity at the Network Layer. In Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 24–26 April 2018; pp. 137–152. [CrossRef]
8. Dingledine, R.; Mathewson, N.; Syverson, P. Tor: The second-generation onion router. In Proceedings of the 13th USENIX Security Symposium, San Diego, CA, USA, 9–13 August 2004; Volume 13, p. 21. [CrossRef]
9. Dingledine, R.; Murdoch, S.J. Performance Improvements on Tor or, Why Tor Is Slow and What We' Re Going to do about It. 2009. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.557.1575&rep=rep1&type=pdf (accessed on 25 May 2022).
10. Dingledine, R.; Mathewson, N. Anonymity Loves Company: Usability and the Network Effect. *Econ. Inf. Secur.* **2006**, *314*, 547–559.
11. Reardon, I.G.J. Improving Tor using a TCP-over-DTLS Tunnel. In Proceedings of the Symposium a Quarterly Journal in Modern Foreign Literatures, Waterville, ME, USA, 2–7 August 2009; pp. 119–134.

12. McCoy, D.; Bauer, K.; Grunwald, D.; Kohno, T.; Sicker, D. Shining light in dark places: Understanding the tor network. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5134 LNCS, pp. 63–76. [CrossRef]
13. Snader, R.; Borisov, N. A Tune-up for Tor: Improving Security and Performance in the Tor Network. In Proceedings of the Network & Distributed System Security Symposium, San Diego, CA, USA, 1–13 February 2008; Volume 8, p.127.
14. Imani, M.; Amirabadi, M.; Wright, M. Modified relay selection and circuit selection for faster Tor. *IET Commun.* **2019**, *13*, 2723–2734. [CrossRef]
15. Barton, A.; Wright, M.; Ming, J.; Imani, M. Towards predicting efficient and anonymous Tor circuits. In Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), Baltimore, MD, USA, 15–17 August 2018; pp. 429–444.
16. RTT-Based Congestion Control for Tor. Available online: https://gitlab.torproject.org/tpo/core/torspec/-/blob/main/proposals/324-rtt-congestion-control.txt (accessed on 25 May 2022).
17. Congestion Control Arrives in Tor 0.4.7-Stable. Available online: https://blog.torproject.org/congestion-contrl-047/ (accessed on 25 May 2022).
18. Le Blond, S.; Choffnes, D.; Zhou, W.; Druschel, P.; Ballani, H.; Francis, P. Towards efficient traffic-analysis resistant anonymity networks. *ACM Sigcomm Comput. Commun.* **2013**, *43*, 303–314. [CrossRef]
19. Snader, R. *Path Selection for Performance- and Security-Improved Onion Routing*; University of Illinois at Urbana-Champaign: Champaign, IL, USA, 2009.
20. AlSabah, M.; Bauer, K.; Elahi, T.; Goldberg, I. The path less travelled: Overcoming Tor's bottlenecks with traffic splitting. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7981 LNCS, pp. 143–163. [CrossRef]
21. Yang, L.; Li, F. mTor: A multipath Tor routing beyond bandwidth throttling. In Proceedings of the 2015 IEEE Conference on Communications and NetworkSecurity (CNS 2015), Florence, Italy, 28–30 September 2015; pp. 479–487. [CrossRef]
22. De la Cadena, W.; Mitseva, A.; Hiller, J.; Pennekamp, J.; Reuter, S.; Filter, J.; Engel, T.; Wehrle, K.; Panchenko, A. Trafficsliver: Fighting website fingerprinting attacks with traffic splitting. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 9–13 November 2020; pp. 1971–1985.
23. Dingledine, R. One fast guard for life (or 9 months). In Proceedings of the 7th Workshop on Hot Topics in Privacy Enhancing Technologies, Amsterdam, The Netherlands, 16–18 July 2014.
24. Bauer, K.; McCoy, D.; Grunwald, D.; Kohno, T.; Sicker, D. Low-resource routing attacks against tor. In Proceedings of the 2007 ACM Workshop on Privacy in Electronic Society—WPES'07, Alexandria, VA, USA, 29 October 2007; p. 11. [CrossRef]
25. De la Cadena, W.; Kaiser, D.; Mitseva, A.; Panchenko, A.; Engel, T. Analysis of multi-path onion routing-based anonymization networks. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11559 LNCS, pp. 240–258. [CrossRef]
26. Castillo-Pérez, S.; Garcia-Alfaro, J. Onion routing circuit construction via latency graphs. *Comput. Secur.* **2013**, *37*, 197–214. [CrossRef]
27. Akhoondi, M.; Yu, C.; Madhyastha, H.V. LASTor: A low-latency AS-aware tor client. In Proceedings of the IEEE/ACM Transactions on Networking, San Francisco, CA, USA, 20–23 May 2014; Volume 22, pp. 1742–1755. [CrossRef]
28. Pries, R.; Yu, W.; Graham, S.; Fu, X. On performance bottleneck of anonymous communication networks. In Proceedings of the IPDPS Miami 2008—22nd IEEE International Parallel and Distributed Processing Symposium, Program and CD-ROM, Miami, FL, USA, 14–18 April 2008. [CrossRef]
29. Johnson, A.; Wacek, C.; Jansen, R.; Sherr, M.; Syverson, P. Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Hangzhou, China, 8–10 May 2013; pp. 337–348. [CrossRef]
30. Varga, A. The OMNET++ discrete event simulation system. In Proceedings of the European Simulation Multiconference, Prague, Czech Republic, 6–9 June 2001; pp. 319–324.
31. Tor Metrics. Available online: https://metrics.torproject.org/collector/archive/relay-descriptors/consensuses (accessed on 25 March 2022).
32. Syverson, P.; Tsudik, G.; Reed, M.; Landwehr, C. Towards an analysis of onion routing security. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2009, pp. 96–114. [CrossRef]
33. Hopper, N.; Vasserman, E.Y.; Chan-Tin, E. How much anonymity does network latency leak? *ACM Trans. Inf. Syst. Secur.* **2010**, *13*, 13. [CrossRef]
34. Shmatikov, V.; Wang, M.H. Timing analysis in low-latency mix networks: Attacks and defenses. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4189 LNCS, pp. 18–33. [CrossRef]

35. Back, A.; Möller, U.; Stiglic, A. Traffic analysis attacks and trade-offs in anonymity providing systems. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2137, pp. 245–257.
36. Díaz, C.; Seys, S.; Claessens, J.; Preneel, B. Towards measuring anonymity. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2482, pp. 54–68.