Electronic Communications of the EASST
Volume 080 (2021)

Conference on Networked Systems 2021
(NetSys 2021)

Improvements to the Secure Construction and Utilization of Greedy Embeddings in Friend-to-Friend Overlays

Martin Byrenheid, Stefanie Roos, and Thorsten Strufe

4 pages

ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Improvements to the Secure Construction and Utilization of Greedy Embeddings in Friend-to-Friend Overlays

Martin Byrenheid[1], **Stefanie Roos**[2], **and Thorsten Strufe**[3]

[1]TU Dresden [2]Delft University of Technology [3]Karlsruhe Institute of Technology

Abstract: Routing based on greedy network embeddings enables efficient and privacy-preserving routing in overlays where connectivity is restricted to mutually trusted nodes. In previous works, we proposed security enhancements to the embedding and routing procedures to protect against denial-of-service attacks by malicious overlay participants. In this work, we propose an improved timeout scheme to reduce the stabilization overhead of secure tree maintenance in response to node failures and malicious behavior. Furthermore, we present an attack-resistant packet replication scheme that leverages alternative paths discovered during routing.

Keywords: Friend-to-Friend Overlay, Network Embedding, Secure Routing

1 Introduction

Friend-to-Friend (F2F) overlay networks, such as Freenet and GNUnet, provide a substrate for privacy-preserving and censorship-resistant communication services. To reduce the exposure of identifying information, such as the IP address, of participating nodes as well as to limit the effectiveness of Sybil attacks, F2F overlays restrict connectivity to mutually trusted nodes.

To enable communication beyond directly connected nodes, F2F overlays require a routing algorithm for restricted topologies that is 1) highly efficient, 2) privacy-preserving, 3) robust against network dynamics, and 4) resistant to attacks. The solution proposed by Roos et al. [RBS16] achieves (1) to (3), making it the most promising candidate. Their approach leverages greedy network embeddings, which assign a unique logical coordinate to every node based on a rooted spanning tree of the network.

State-of-the-art embedding algorithms [RBS16, $\text{H}^{+}15$, HWT11] however provide limited protection against denial of service attacks by malicious nodes. Consequently, a malicious node with just two links to the honest part of the overlay can reduce routing success ratio by up to 40% [BSR20b]. To protect against malicious node behavior, we thus proposed attack-resistant algorithms for root election [BSR20a], breadth-first-search (BFS) tree setup [BRS19], and coordinate assignment [BSR20b].

In this work, we present an improvement to our secure tree construction that reduces stabilization overhead in response to non-root node failures. Furthermore, we propose an adaptive packet replication mechanism to further increase the chance of succesful routing in the presence of malicious nodes.

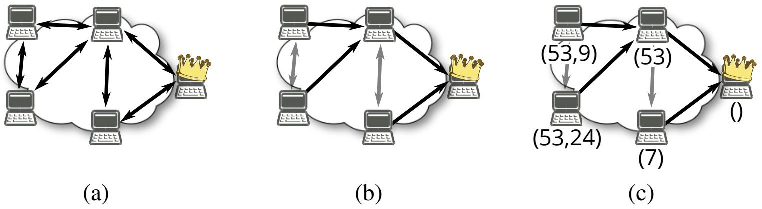

Figure 1: Primary steps of state of the art embedding algorithms: (a) root election, (b) tree construction, (c) coordinate assignment.

2 System Model and Adversary Model

We model a F2F overlay as a network of nodes with bidirectional, logical communication links. The network is dynamic, such that nodes may join or leave the network and links are set up and torn down over time.

To enable routing, the overlay utilizes a distributed *embedding algorithm*, which assigns a unique logical coordinate to each overlay node. In our work, we focus on embeddings that use vectors of integers as coordinates, as these enable efficient addressing without the need to obtain global topology information, such as the number of participating nodes.

Current embedding algorithms assign coordinates by first constructing a distributed BFS spanning tree of the overlay network. Afterwards, the root node chooses the empty vector as its logical coordinate. Each node subsequently obtains a logical coordinate by appending a random integer to the coordinate of its parent in the spanning tree, as illustrated in Figure 1.

To route a message to a given destination coordinate d, nodes use a *distance metric* to measure the logical distance of each neighbor's coordinate to d. The message is then forwarded greedily to one of those neighbors whose coordinate has the lowest logical distance to d.

In our work, we consider an *internal* adversary that controls a subset of the nodes participating in the overlay, which we call *malicious nodes*. For readability, we call a non-malicious node an *honest node* in the following.

Each malicious node may collude with other malicious nodes and deviate arbitrarily from the correct behavior, e.g. by sending messages with incorrect data and dropping messages. Due to the absence of a central admission control, the adversary may furthermore introduce an arbitrary number of additional malicious nodes that are not connected to any honest node.

While the adversary can arbitrarily set up links between malicious nodes, we assume that the total number of links between malicious and honest nodes is bounded. We consider this assumption to be realistic, since the setup of connections to honest nodes requires social engineering, which we consider to be too costly to conduct on a large number of individuals.

With regards to computational resources, we assume that the adversary is limited to polynomial time and hence unable to break computationally secure cryptographic primitives. However, the total computational power of the adversary may still be much higher than the computational power of the honest nodes combined.

3 Timeout thresholds for efficient and secure tree maintenance

To securely detect the failure of a root node, we proposed the periodic propagation of timestamps along with cryptographic signatures generated by the root node [BSR20a]. If the root node fails or becomes unreachable due to overlay partitioning, the existing timestamps will eventually expire and nodes will start to elect a new root node. However, if a non-root node u fails, its children may not be able to tell whether the root node is still reachable via another path.

To reduce stabilization overhead if the root node is still reachable, it is desirable to ensure that nodes in the subtree formerly rooted at u will first try to locate an alternative path before electing a new root. However, state-of-the-art algorithms for secure BFS tree maintenance [DMT15, BRS19] may temporarily create cycles, which require the affected nodes to change their state multiple times before a stable state is reached. To avoid the formation of cycles and thus reduce convergence time, we extend our algorithm by distinguishing between two thresholds $T_{search} \in \mathbb{N}$ and $T_{reset} \in \mathbb{N}$ with $T_{search} < T_{reset}$. When the difference between u's system time and the most recent timestamp reported by its current parent exceeds T_{search}, u checks if there are neighbors whose reported timestamp does not exceed threshold T_{tree}. If this is the case, then u will select one of those neighbors as its new parent. Otherwise, u will not change its parent. If all timestamps reported by u's neighbors exceed threshold T_{reset}, then u will consider its current root node failed.

Let $f_{heartbeat}$ be the frequency at which the root node generates new timestamps and let d_u be the hop distance from u to the root node. Given upper bounds on the maximum clock difference Δ_C between any two nodes, the maximum processing and transmission delay Δ_P between any two nodes, the number of lost timestamp messages n_{lost}, we set $T_{search} = \Delta_C + \Delta_P d_u + n_{lost} \cdot f_{heatbeat}$. Given an upper bound D on the network diameter, we set $T_{reset} = T_{search} + \Delta_P D$. We consider knowledge of an upper bound on the diameter to be realistic, as studies on social graphs indicate that even networks with millions of users have a diameter below 35 [Les].

In the case the root node actually became unreachable, the comparatively high value for T_{reset} causes the overlay to operate without a valid root node for a longer time. However, the study by Roos et al. [RBS16] shows that if multiple embeddings are constructed, routing is still highly likely to succeed despite a temporary root failure. Thus, it is possible to tolerate a delayed reaction to root node failures in favor of reduced overhead for adaptation to non-root faults.

4 Adaptive packet replication for robust and secure routing

When a node u routes a message m towards a particular coordinate C, it is possible that multiple neighbors of u have a lower distance to C. To enable discovery of alternative paths in the presence of faults, Roos et al. [RBS16] propose backtracking: if the neighbor v chosen by u as next hop is unable to forward m further towards C, it will notify u accordingly and u will send m to a different neighbor with a lower distance. If u is not the originator of m and received negative feedback from all closer neighbors, it will notify its predecessor about its failure to forward m.

While backtracking increases robustness against node failures, it relies on nodes to correctly report failures to route. Thus, if a malicious node silently drops a message received from an honest node u, then u will not detect the loss and initiate backtracking. As an alternative, we propose an adaptive replication of packets. To do so, we let the originator include a *replication counter* r in each message. If a node u needs to forward a message with $r = k$ and has at least k closer neighbors, it will send a copy of m with $r = 1$ to the k closest neighbors. If u has less than k closer neighbors, it may send a copy of m to each closer neighbor such that the sum of the r values yields k. The initial value of r can be changed to trade off overhead and attack-resistance.

Compared to having the originator sending the same message multiple times, this approach avoids duplicate transmissions of the same message over the same link.

5 Conclusion and future work

In this work, we proposed a timeout scheme that enables nodes to check for alternative paths to the root node before starting a new election. Furthermore, we presented a proactive message replication scheme to improve resilience against packet dropping by malicious nodes.

To evaluate the effectiveness of the proposed measures, we will conduct a simulation study using different real world social graphs. For the timeout thresholds proposed in this paper, we expect a significant reduction in stabilization overhead over a naive root timeout in response to failure of nodes with degree, as these are likely to have a large number of children. For the adaptive packet replication, we expect the increase in routing success ratio over routing with backtracking in the presence of malicious nodes to grow with the network density, as more alternative paths become available.

Bibliography

[BRS19] M. Byrenheid, S. Roos, T. Strufe. Attack-Resistant Spanning Tree Construction in Route-Restricted Overlay Networks. In *SRDS*. 2019.

[BSR20a] M. Byrenheid, T. Strufe, S. Roos. Attack resistant Leader Election in Social Overlay Networks by Leveraging Local Voting. In *ICDCN*. 2020.

[BSR20b] M. Byrenheid, T. Strufe, S. Roos. Secure Embedding of Rooted Spanning Trees for Scalable Routing in Topology-Restricted Networks. In *SRDS*. 2020.

[DMT15] S. Dubois, T. Masuzawa, S. Tixeuil. Maximum Metric Spanning Tree Made Byzantine Tolerant. *Algorithmica*, 2015.

[H$^+$15] R. Houthooft et al. Robust geometric forest routing with tunable load balancing. In *INFOCOM*. 2015.

[HWT11] J. Herzen, C. Westphal, P. Thiran. Scalable routing easy as PIE: A practical isometric embedding protocol. In *IEEE ICNP*. 2011.

[Les] J. Leskovec. Friendster social network and ground-truth communities. http://snap.stanford.edu/data/com-Friendster.html. February 2021.

[RBS16] S. Roos, M. Beck, T. Strufe. Anonymous addresses for efficient and resilient routing in F2F overlays. In *INFOCOM*. 2016.