Robust Forwarding in Structured Peer-to-Peer Overlay Networks

Wang-kee Poon
Department of Computing
The Hong Kong Polytechnic University
cswkpoon@comp.polyu.edu.hk

Rocky K. C. Chang
Department of Computing
The Hong Kong Polytechnic University
csrchang@comp.polyu.edu.hk

ABSTRACT

Structured peer-to-peer overlay networks are widely exploited to build large-scale decentralized applications. Robustness to malicious nodes is an important and fundamental objective of designing an overlay network. The presence of even a small percentage of malicious nodes can present a serious threat to the stability of an overlay network. Existing approaches mainly utilize multiple paths to increase the probability of delivery. In this paper, we propose *Fence* that allows a source to diagnose possible forwarding faults injected by malicious nodes. Based on the feedback information, the source is able to identify malicious nodes and to achieve very robust message forwarding.

1. BACKGROUND

Structured peer-to-peer (p2p) overlay networks, such as CAN, Chord, Pastry, and Tapestry, provide distributed lookup services for large-scale decentralized applications. A major function of the overlay networks is to efficiently lookup the node corresponding to a key. While many lookup mechanisms are quite efficient, they are vulnerable in the presence of malicious nodes. Existing implementations do not address this problem directly, except for providing a time-out based retransmission scheme.

2. PROBLEM

A malicious node can disturb a normal delivery of a lookup message by message dropping, delayed forwarding, or incorrect forwarding. Only a small percentage of such malicious nodes is needed to seriously degrade the forwarding performance. In a simulation study of Pastry, the probability of a successful lookup is dropped to 65% when there are 10% malicious nodes [1]. Existing solutions mainly involve forwarding a message through multiple paths, with the hope to increase the probability of delivery. In this paper, we take another approach that involves first identifying the malicious nodes, and then performs appropriate actions to bypass them.

3. APPROACH

In this paper we propose *Fence* to diagnose *forwarding faults* on a forwarding path that are injected by malicious nodes. Forwarding faults include undesirable delay, message dropping, and forwarding to undesirable nodes. Once a source discovers a fault through Fence, it can perform immediate actions, such as fast recovery, fault isolation, and message blocking.

Fence allows a source to monitor each forwarding step of its own lookup messages. Using Fence, a source expects to receive from each forwarding node a *proof* for each successful forwarding of its messages. Based on the proof, the source can learn 1) the duration that the message has stayed at each forwarding node, 2) whether the forwarding is successful, and 3) the identity of the next forwarding node. It turns out that these three pieces of information are sufficient to allow the source to identify forwarding faults.

Based on the inference on the status of its messages, a source can perform the following actions to provide a very robust forwarding service in the presence of malicious nodes.

- **Fast Recovery** A source can roll back to the previous hop to continue forwarding immediately after identifying a forwarding fault.
- **Fault Isolation** A source, after identifying a list of malicious nodes, may inform other forwarding nodes about this list, so that these malicious nodes can be eventually isolated from the network.
- **Message Blocking** A node may ignore lookup messages sent from malicious nodes. In this case, the malicious nodes may include selfish nodes which do not dutifully provide forwarding service. Therefore, blocking the selfish nodes' messages would serve as an incentive for a more nonselfish behavior.

Further information can be accessed from:
http://www.comp.polyu.edu.hk/∼cswkpoon/fence

4. REFERENCES

[1] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach. Security for structured peer-to-peer overlay networks. In *5th Symposium on Operating Systems Design and Implementaion (OSDI'02)*, Dec. 2002.