Available online at www.sciencedirect.com
SCIENCE DIRECT
Computer Communications 29 (2006) 1516–1521
computer communications
www.elsevier.com/locate/comcom

Counter-based reliability optimization for gossip-based broadcasting

Tatsuhiro Tsuchiya *, Shinichi Ikeda, Tohru Kikuno

Osaka University, Graduate School of Information Science and Technology, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

Received 12 January 2005; received in revised form 15 September 2005; accepted 27 September 2005
Available online 15 November 2005

Abstract

Gossip-based broadcasting has recently gained popularity as a potentially effective solution for disseminating information in large-scale distributed applications. In this paper we propose a simple counter-based optimization for gossip-based broadcast protocols, aimed at enhancing reliability in the face of node failures. The basic idea is to have each node retransmit a broadcast message when the node has received the same message only a few times; a node thus can autonomously perceive the status of message dissemination and promote it when necessary. The usefulness of the technique is demonstrated through simulation results.

Keywords: Gossip; Epidemic; Broadcast; Overlay networks; Reliability; Distributed systems

1. Introduction

In the past few years, we have seen the rapid growth of peer-to-peer applications and the emergence of overlay infrastructures for the Internet. The scalability required from such applications has reached thousands or more participant nodes. *Gossip* or *epidemic-style broadcasting* has recently gained popularity as a potentially effective solution for disseminating information in such large-scale systems [1–3]. In this paper, we propose a simple optimization technique aimed at enhancing the robustness of gossip to node failures.

The essential characteristic of gossip is that information exchanges occur between sender nodes and randomly chosen gossip partners. Information is disseminated throughout the system by multiple rounds of such communication. This proactive use of redundant messages provides a means to ensure reliability in the face of failures. Also, it is shown that the load on each node increases only logarithmically with the size of the network [4,5]; so the gossip algorithms are scalable. Due to these properties, the gossip protocols have already been used in various contexts, such as replicated databases [6,7], distributed failure detection [8,9], or distributed information management [10].

In this paper, we propose a simple optimization technique that can further improve robustness to failures. Analytical and simulation results have shown that gossip-based broadcast protocols are highly reliable [11]. The robustness requirement arising from peer-to-peer applications is not fully addressed however, because compared to traditional distributed systems, these applications are much more subject to failures, due to their large scale and high churn rates.

The basic idea of our approach is to have each node retransmit a broadcast message when the node determines that the message has not been sufficiently disseminated. Each node autonomously makes the decision of retransmission by counting how many times it received the same message. A similar approach can be found in broadcasting protocols for ad hoc networks [12,13]. In ad hoc networks, a node can receive every message sent by its neighbors due to the nature of wireless communications; thus the node can easily sense the status of message dissemination. Clearly, this is not the case with wired networks. To our knowledge, no results from applying the counter-based approach to gossip protocols for wired networks exist in the literature.

The remainder of this paper is structured as follows. Section 2 describes the model of distributed systems and the concept of gossip-based broadcasting. Section 3 describes the proposed optimization technique, together with a specific gossip algorithm. Section 4 presents the simulation results. The results demonstrate that this counter-based optimization substantially improves the robustness to node failures. Section 5 concludes the paper with possible directions for future research.

* Corresponding author. Tel.: +81 66 879 4536; fax: +81 66 879 4539.
E-mail addresses: t-tutiya@ist.osaka-u.ac.jp (T. Tsuchiya), s-ikeda@ist.osaka-u.ac.jp (S. Ikeda), kikuno@ist.osaka-u.ac.jp (T. Kikuno).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.09.018

2. Model

We consider a distributed system consisting of a number of computing nodes. Membership information maintained by the nodes in the system determines the topology of a logical network: the link between nodes i and j exists in the network iff i and j know each other's address. This logical network is often called an *overlay network*. A node i is said to be a *neighbor* of another node j iff there is a link between them.

Membership is a fundamental issue underlying deployment of gossip algorithms, because the topology of the overlay network significantly affects the performance and/or reliability of gossip. Some gossip algorithms incorporate a membership management mechanism (e.g. [14]). In this paper, however, we focus our attention on message dissemination; we assume that the overlay network is maintained by a distinct membership protocol, such as in [15].

Although our proposed mechanism can be applied to gossip algorithms in general, we use a concrete algorithm to clearly illustrate our idea. We select the simple gossip algorithm described in Fig. 1 as the baseline. This is the algorithm called *blind counter rumor mongering* [16], with some parameters set to particular values.

In this algorithm when a node initiates broadcast, the node sends the message to f randomly selected neighbors. Message dissemination is carried out as follows: upon receiving a broadcast message for the first time, the node p randomly selects f neighbors as gossip partners and forwards copies of the message to all these selected nodes. The value f is usually referred to as a *fanout*. It should be noted that if p knows that a neighbor has already received m, then that neighbor will never be selected. In this paper we assume that when receiving m, the node can tell the sender node's address; thus the sender node is never chosen by p as a gossip partner.

3. Counter-based optimization

The reliability of gossip is dominated by the occurrences of failures; but for each node to comprehend which nodes are operational and have not left the system is difficult. This is especially true for peer-to-peer applications where node joins and leaves are common.

A simple way to improve resilience to failures in such a situation is to use a high fanout. When failed nodes are rare, however, this approach would result in unnecessarily high messaging cost. To dynamically adjust the fanout to an appropriate value is possible if nodes can accurately perceive the current status of the network. But substantial additional messaging cost is necessary to obtain such information in large-scale systems [17,18].

```
initiate broadcast of m:
  send m to f randomly chosen neighbors;

when a node p receives a message m:
  if (p has received m for the first time)
    p sends m to f uniformly randomly chosen neighbors
    that p knows who have not yet seen m;
```

Fig. 1. Ordinary gossip (rumor mongering).

Our goal is to improve the reliability of the broadcast by introducing into gossip algorithms an optimization technique that can adapt gossiping to the status of the system without explicit failure detection or aggregation.[1]

How does this adaptation work? Each node counts how many times it repeatedly receives each broadcast message. Nodes thus can perceive the status of the system indirectly; if the same message has been received many times, it is likely that many other nodes also have successfully received the message. On the other hand, if a broadcast message has arrived only a few times, it can be inferred with a high probability that there are other nodes that have failed to receive the message. Thus in the latter case, retransmission of the message would probably promote the message dissemination process.

This simple idea is the basis of the proposed mechanism. Fig. 2 shows the gossip protocol that incorporates this optimization. The protocol consists of three parts: initiation, forwarding, and retransmission.

To control the retransmission process, the protocol has two additional parameters: T and θ. T is the time in which a node waits to start retransmission, while θ is used to determine whether or not to perform retransmission: if a node has received the same message no more than θ times at time T after the first arrival of the message, retransmission is performed. In Fig. 2 *buff* is a buffer that maintains the messages currently being handled. Variable counter_m is used to count the number of times a message m is received. Variable known_m contains both the nodes from which m was received and those to which m was forwarded. The nodes in known_m will never be selected as the receiver nodes when retransmission is performed because they already received m (unless communication failures occurred).

4. Simulation

4.1. Settings

In this section we present the results of a performed simulation analysis. The simulation compares ordinary gossip and gossip with the proposed optimization. As in [14,19], we assume that the nodes in the system have a loosely synchronized clock and execute the protocol periodically. We let Δ represent the length of this period. We also assume that message delay is sufficiently small compared to the period.

We considered two types of network topologies: type 1 and type 2. Type 1 networks are such that every node has the same number of neighbors selected uniformly randomly from the other nodes. Topologies of this type are quite typical for overlay networks; existing systems that use this type of network topologies include, for example, LPBCAST [14] and PROOFS [20]. SCAMP [15] also constructs a similar network.

[1] *Aggregation* refers to a set of functions that provide global information about a distributed system.