IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING
IEEJ Trans 2010; **5**: 121–122
Published online in Wiley InterScience (www.interscience.wiley.com). DOI:10.1002/tee.20503

Letter

Dynamic Load-Aware Multicast Routing in *Ad hoc* Networks

Koichi Shimizu[*], Non-member
Takahiro Sawamura[**], Non-member
Md. Enamul Haque[*], Non-member
Noriko Matsumoto[*], Non-member
Norihiko Yoshida[*a], Member

In wireless *ad hoc* networks, collisions and congestion at some nodes cause degradation of network throughput. There have been some load-aware routing protocols proposed for load distribution and balancing by detouring congested nodes. However, they are for unicast communication. This paper proposes a load-aware protocol for the same purpose in multicast communication. The protocol has features to build a multicast tree with load distribution and balancing, and to reorganize the tree according to dynamic load changes. Simulation-based preliminary experiments result in that the number of highly loaded members in a multicast tree was smaller with the proposed protocol than that with a conventional one.

Keywords: wireless *ad hoc* networks, load-aware routing protocol, multicast communication

Received 30 December 2008; Revised 13 April 2009

1. Introduction

Most routing protocols for *ad hoc* networks construct a route as a shortest path, but do not consider route overlapping. This causes collisions and congestion at some nodes, and consequently causes degradation of network throughput. CSMA/CA in wireless communication forces transmission delay to avoid message collision. There is a report that the service quality is intolerably degraded when three routes overlap in VoIP applications or when two overlap in video streaming [1].

There have been some *load-aware* routing protocols proposed for load balancing and distribution by detouring congested nodes in wireless *ad hoc* networks. Typical examples, presented in Refs [2] and [3], are extensions of the DSR routing protocol, and operate in a decentralized manner as follows.

A route request (RREQ) packet has an additional header field (LOAD) for accumulated load information. A source node floods out an RREQ with the value 0 in the LOAD. A node receiving the RREQ adds its load value to the LOAD, and again floods it out. When the destination node receives more than one RREQ, the node compares LOAD values in them, and selects a route with the least load value.

Another protocol, which operates in a centralized manner, has also been proposed as an extension of cluster-based routing [4]. The central node, or a cluster head, holds topology information of a network around it, and builds an optimal route with detours if necessary, using a weighted Dijkstra algorithm.

They are, however, mostly for unicast communication. This paper proposes a load-aware protocol for the same purpose in multicast communication. There have already been many protocols proposed and used for multicast communication in wireless and wired networks such as DVMRP, MAODV, ODMRP, ABAM, MOSPF, PIM-DM, and PIM-SM to name a few. However, none of them are concerned with load awareness as far as we know. Our protocol has some features to build a multicast tree with load distribution and balancing, and to reorganize the tree according to dynamic load changes.

2. Load-Aware Multicast Routing

The proposed protocol constructs a source-based multicast tree as below.

2.1. Member joining phase A new node that wants to join a multicast tree follows the below procedure (Fig. 1).

(1) The new node floods out a join request (JREQ) packet. This packet contains the multicast address which the node are joining, and a header field (LOAD) for load information. The JREQ accumulates load values of nodes along a route while it is forwarded from node to node.
(2) When any member node of the multicast tree receives the JREQ, it waits for the same JREQ through other routes for a certain period. Then it selects a route with the least load, and send back a route reply (RREP) to the new node. When the

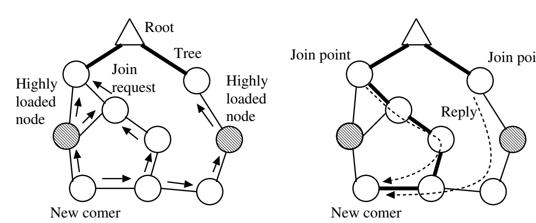

Fig. 1. Member joining phase

[a] Correspondence to: Norihiko Yoshida.
E-mail: yoshida@mail.saitama-u.ac.jp

[*] Graduate School of Science and Engineering, Saitama University, Saitama 338-8570, Japan
[**] NEC Informatec Systems, Kawasaki 213-0012, Japan

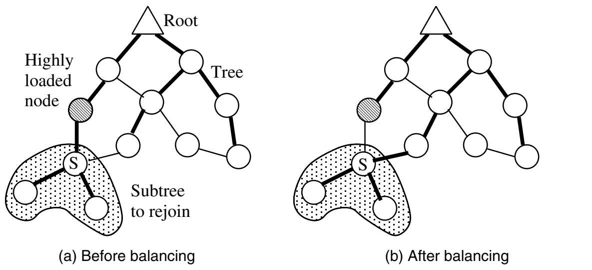

Fig. 2. Load balancing phase

new node receives an RREP, it waits for any other RREP from other members for a certain period. Then it selects a route with the least load, and it joins the multicast tree through the route.

2.2. Load balancing phase There is a possibility that the load of any member in the multicast tree increases dynamically due to another multicast tree or a unicast path overlapping the tree after the construction. Then transmission delay occurs not only at the highly loaded member but also at its downstream members.

Every multicast data packet in the tree accumulates load values of members along its path when it is forwarded from member to member. In this way, any member receiving a multicast packet gets the sum of load values of all the members along the path from the root to itself.

If a member finds that the sum of load values exceeds a certain threshold, the member grafts a subtree including itself and its downstream members to another member with a lower load. Actually, the member does the procedure in the member joining phase again, and reconstructs the tree so as to detour the highly loaded member (Fig. 2).

There are two cares that should be taken. Only the topmost node in the subtree under the highly loaded member should get in the member joining phase, but no other member in the same subtree should. In addition, the node in the member joining phase should prevent itself from linking to any member in its subtree, i.e. prevent the tree from having a loop. The data packet has a special header field used to take care of these.

2.3. Tree reconstruction phase There is a case that a member finds itself being cut apart from its parent member because of node mobility or lack of network dependability. Then the member does the same procedure as in the member joining phase, and reconstructs the multicast tree again. The estimated overhead in this procedure is as mush as in the DSR protocol.

3. Simulation Experiments

To verify the realizability and performance of the proposed proposal, we carried out some simulation-based preliminary experiments.

The simulator modeled a network of 400 nodes placed randomly within 400×400 area, and each node has a radio propagation range of 10. There is one multicast tree constructed, and 100 nodes which are randomly chosen emit the join request messages, while ten members which are randomly chosen leave the multicast group in turn as well. Every node is artificially assigned low, medium, or high load at first. A highly loaded node simulates a node under route conflict. The load value for each class is 1, 8, and 32 respectively, and the threshold value to invoke the load balancing phase is 60.

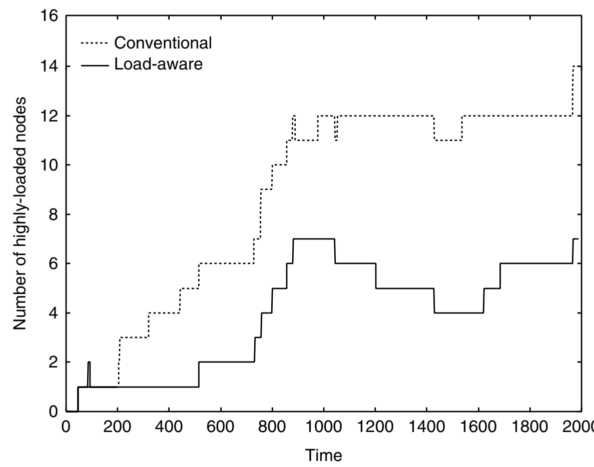

Fig. 3. Transition of number of highly loaded nodes

Figure 3 shows transition of the number of highly loaded members in the multicast tree. Compared with a conventional (load-unaware) protocol for an AODV multicast tree, the proposed protocol kept the number of highly loaded members relatively low, which means that highly loaded nodes were being eliminated from the tree, and loads among the members were well-distributed. Consequently, it exhibited better performance in regards to preventing load concentration and bringing load balancing.

4. Conclusions

This paper presented a multicast routing protocol with dynamic load distribution. Nodes construct and reconstruct dynamically a multicast tree transferring load information and detouring highly loaded nodes.

Simulation-based preliminary experiments resulted in that the number of highly loaded members in a multicast tree was smaller with the proposed protocol than with a conventional one. There has been no other protocol for *ad hoc* multicasts with adaptive reconstruction for load balancing as far as we know.

This protocol is originally for source-based multicast trees, however, it can be applied to core-based multicast trees in the case that there are more than one cores.

Although the experiments exhibited some promising results, we are still at a starting point in our research project, and we must conduct rigorous investigations and evaluations in further studies. One of the most important issues to address next is to extend our proposal to handle node mobility more precisely.

References

(1) Itaya S, Hasegawa J, Hasegawa A, Davis P, Kadowaki N, Yamaguchi A, Obana S. Experimental evaluation of media distribution over a wireless Ad hoc network. *Proceedings of the IPSJ/IEICE Forum on Information Technology* 2005; **4**:131–132 (in Japanese).

(2) Lee SJ, Gerla M. Dynamic load-aware routing in Ad hoc networks. *Proceedings on the IEEE International Conference on Communication* 2001; **10**:3206–3210.

(3) Tao X, Kunz T, Falconer D. Traffic balancing in wireless mesh networks. *Proceedings of the International Conference on Wireless Networks, Communicatin and Mobile Computation* 2005; **1**:169–174.

(4) Tanaka K. Routing with Conflict Resolution in Ad-hoc Networks. Master Thesis, Saitama University, 2006; (in Japanese).