Node Synchronization Based Redundant Routing for Mobile Ad-Hoc Networks[*]

Wonjong Noh, Yunkuk Kim, and Sunshin An

Computer Network Lab, Dep. Of Electronics Engineering, Korea University
Sungbuk-gu, Anam-dong 5ga 1, SEOUL, KOREA, Post Code: 136-701,
Phone: +82-2-925-5377, FAX: +82-2-3290-3674
{nwj, dbs1225, sunshin}@dsys.korea.ac.kr

Abstract. This paper proposes a new on-demand redundant-path routing protocol considering node synchronization based path redundancy as one of route selection criteria. Path redundancy implies how many possible redundant paths may exist on a route to be built up. Our proposal aims to establish a route that contains more redundant paths toward the destination by involving intermediate nodes with relatively mode adjacent nodes in a possible route. Our approach can localize the effects of route failures, and reduce control traffic overhead and route reconfiguration time by enhancing the reachability to the destination node without source-initiated route re-discoveries at route failures. We have evaluated the performance of our routing scheme through a series of simulations using the Network Simulator 2 (ns-2).

1 Introduction

A mobile ad hoc network [1] is a collection of wireless mobile nodes forming a temporary network without the aid of any established infrastructure. Adjacent nodes communicate directly between one another over wireless channels. However, since the transmission range of nodes is limited, the nodes that are not neighboring need routing supported by intermediate nodes for communications. Due to the mobility of nodes, the topology of connections between communicating nodes may be quite dynamic. In a dynamic ad hoc network, route re-discoveries due to route failures may incur heavy control traffic through the network and cause the increase of packet transmission delay. Hence, it is quite required to reduce the number of route re-discoveries by maintaining multiple redundant paths, establishing alternate route promptly and localizing the effect of the failures. Numerous ad hoc routing protocols have been developed [2, 3, 4, 5, 6, 7] but most of them don't deal with route reconfiguration by dynamic topology.

This paper presents a new on-demand redundant-path ad-hoc network routing protocol that provides dynamic and fast route reconfiguration using information about

---

[*] An unabridged version of this paper is available at the following URL: "http://nlab.korea.ac.kr/~angus/wons2003-full.pdf

redundant paths maintained at a source and intermediate nodes on initial route. Our protocol can establish a route consisting of intermediate nodes with relatively more node-synchronized neighboring nodes and hence it provides more redundant paths toward the destination We introduce a new routing metric, ‘path redundancy’, which implies how many possible redundant paths on a route to be built up. The remainder of this paper is organized as follows. The following section describes node synchronization concepts. Section 3 presents our routing procedures. Finally we present the simulation result and conclusions.

2 Node Synchronizations

2.1 Definitions

$\overrightarrow{v_i}$, $\overrightarrow{v_{ij}}$: Node(i)’s mobility vector, node(i)’s relative mobility vector with respect to node (j)

$T^e_{data,i}$: Expected whole data transmission time at node(i)

$T^e_{power,j} = \int_0^\infty P\left(T_{power,j} > t \mid T = T_c\right) dt$: Expected power alive time at node (j)

$$T^e_{link,i,j} = \int_0^\infty P\left(T_{link,i,j} > t \mid T = T_c\right) dt = \int_0^\infty P\left(\min\left\{ T \mid \left| \overrightarrow{d_{ij}}(T_c) + \int_{T_c}^{T_c+T} \overrightarrow{v_{ij}}(t)\, dt \right| \geq r_i \right\} > t \right) dt$$

: Expected link-alive time between neighboring two nodes, node(i) and node(j),

where, $\overrightarrow{d_{ij}}(T_c)$ is node(j)’s location vector with regard to node(i) at current time T_c .

r_i is transmission range of node(i)

$R^e_{comb,i}$: Expected combinatorially stable region

$R^e_{\theta,i,j}$: Estimated maximal pie-type sub-region inside node(i)’s neighborhood region

2.2 Synchronizations

When $T^{e}_{data,i}$ is less than or equal to $\min\left\{T^{e}_{link,i,j}, T^{e}_{power,j}\right\}$, node(j) is considered to be time synchronized to node(i). Spartial synchronization is related to each node's current position. When the node(j) is located in intersection area between combinatorially stable region [8] and estimated maximal pie-type sub-region inside neighborhood region, $R^{e}_{comb,i} \cap R^{e}_{\theta,i,j}$, it is considered to be spartially synchronized to node(i). Mobility synchronization is related to each node's mobility vectors. If the inequality $\left|\arccos\left(\vec{v_i}\cdot\vec{v_j} \Big/ \left|\vec{v_i}\right|\cdot\left|\vec{v_j}\right|\right)\right| \le \frac{\pi}{2}$ is satisfied, node(i) and node(j) is mobility synchronized.

2.3 Path Redundancy

$$\sum_{i\in path(p)} \sum_{j\in nbd(i)} sync(i,j) \tag{1}$$

$$sync(i,j) = \alpha(i,j)\cdot I_{time\_sync,i,j} + \beta(i,j)\cdot I_{spartial\_sync,i,j} + \gamma(i,j)\cdot I_{mobility\_sync,i,j} \tag{2}$$

$$\underset{path(p)\in paths(src,dst)}{\arg\max} \left\{ \sum_{i\in path(p)} min\left\{ \sum_{j\in nbd(i)} sync(i,j), UpperLimit \right\}\right\} \tag{3}$$

Formulation(1) denotes the path(p)'s redundancy degree. Where, function I is a synchronization indicator function. If node(i) and node(j) is synchronized with respect to time, spartial and mobility, $I_{\{time,spartial,mobility\}\_sync,i,j}$ is 1, otherwise 0. $\alpha(i,j)$, $\beta(i,j)$ and $\gamma(i,j)$ are weigh functions with respect to time, spartial and mobility synchronization. Formulation(3) shows how to choose an optimal route from source node to destination node. To prohibit the case of a path's redundancy degree is severely influenced by a specific node having especially large node redundancy degree, we use the upper limit on each node's redundancy degree.

3 Node Synchronization Based Redundant Routing

3.1 Route request process

A node initiates route establishment procedure by broadcasting a route setup message, Route Setup (RS) packet. An RS packet is flooded throughout the network as shown in Fig.1 and carries the information about redundancy degree and hop distance of nodes that it goes through. Any node that receives an RS packet does the following:

Case 1: If the node recognizes its own address as the intermediate node address, the node records the address of the neighbor node from which it received the RS packet as the upstream node. The recorded node address will be used to build a route during the route reply process. Then, it adds its own redundancy degree to that of the RS packet and broadcasts the updated packet to its neighbor nodes.

Case 2: If the node has already received the RS packet with the same identification, it records the address of the node from which it received the packet as a redundant upstream node and then drops that packet. The recorded node address will be used to build a redundant path if this node is involved in the selected route.

Case 3: If the node recognizes its own address as the destination address, it records the forwarding node address, hop count and path redundancy of the packet.

For the purpose of optimal route selection, the destination will wait for a certain number of RS packets to reach it after receiving the first RS packet. The destination node can receive several RS packets transmitted along different paths from the source node. An RS packet delivered along the shortest route will early reach the destination node and RS packets representing routes with more redundant links may come to later. The destination node adopts the RS packet that reached it later, but contained larger path redundancy per hop, and sends a Route Reply (RR) packet back to the source node via the node from which it received the RS packet.

3.2 Route reply process

A route containing redundant paths toward the destination is established during the route reply process. After selecting the optimal path using equation(3), the destination node initiates the route reply process by sending an RR packet back to the source node via the node from which it received the corresponding RS packet. An RR packet is forwarded back along the transit nodes the RS packet was traversed. An RR packet carries the hop distance from the destination to the node that received the RR packet. The hop distance is incremented by one whenever the RR packet is forwarded at each intermediate node. Any node that receives an RR packet does the following.

Case 1: If the node recognizes itself as the target node of the received RR packet, it records the forwarding node address of the packet as the next hop for the destination. Then, the node increments the hop distance of the received packet and sends the updated packet to its upstream node, which was recorded during the route setup process. Moreover, if the node has any redundant upstream node recorded, it generates and sends the Redundant Route Reply (RRR) packet to the redundant neighbor node(s). The hop distance of the RR packet is copied into the hop distance field of the RRR packet.

Case 2: If the node recognizes its own address as the source, it records the forwarding node address of the RR packet as the next hop for the destination in the route table.

Case 3: If the node is not targeted, the node discards the RR packet.

The RRR packet is used to setup a redundant path of a route. An RRR packet is originated from only nodes along a main route if they have redundant nodes in the upstream direction. RRR packets are forwarded at redundant nodes toward the source node. Any node that receives an RRR packet does the following:

Case 1: If the node exists along the main route, it creates the redundant route table (RRT) entry, which is a set of redundant neighbor nodes for the destination. A redundant next hop field of the RRT entry is filled with the forwarding node address of the RRR packet.

Case 2: If the node is along a redundant path and has not received the RRR packet, it records the forwarding node address of the RRR packet as a redundant next hop for the destination in the RRT entry. Then, it forwards the packet to the upstream nodes.

Case 3: If the node is along a redundant path and has already received the RRR packet with the same identification, it discards the packet. This means that a redundant path cannot have any redundant path for itself.

Case 4: If the node is not targeted, it discards the RRR packet.

3.3 Route reconfiguration

Failure notification is progressed when a failure-detecting node or a node that received a Failure Notification (FN) packet does not have any redundant path for the destination. Route failure information is carried using an FN packet and stored in the failure record. Route failure information is carried using a Failure Notification (FN) packet and stored in the failure record. Route failure information includes the information about a failure-detecting node, whether the failure-detecting node is along a main route or not, and intermediate transit nodes that an FN packet is propagated through. Any node that receives an FN packet does the following.

Case 1: If the node is along a main route (shortly, main node) and the FN packet originated from a main node, it records the failure information and finds an alternate redundant path.

Case 2: If the node is a main node and the FN packet originated from a node along a redundant path (shortly, redundant node), it removes the corresponding redundant path information from the RRT entry.

Case 3: If the node is a main node and the FN packet originated from a node along an active redundant path (shortly, active redundant node), it records the failure information and finds an alternate redundant path.

Case 4: If the node cannot find a redundant path, it adds its node address to the FN transit node list of the FN packet and propagates the updated packet. Moreover, it deletes all the information about the failed route.

Case 5: If the node is an active redundant node, it deletes the corresponding Routing Table (RT) entry and RRT entry and adds its node address to the FN transit node list of the FN packet and propagates the updated packet.

Case 6: If the node is an inactive redundant node, it deletes the corresponding RRT entry and broadcasts the packet.

4 Conclusion

This paper presented a new on-demand ad-hoc network routing protocol that can accomplish dynamic and fast route reconfiguration using information about redundant paths maintained at a source node and intermediate nodes on the main route. The proposed routing protocol can establish a main route with relatively more synchronized redundant paths using a new routing metric of path redundancy. Our approach can localize the effects of route failures and cut down route reconfiguration time by raising the reachability to destination nodes without route re-discoveries in the event of the route failures. We conducted a performance evaluation of our protocol through a simulation using NS-2 [9]. We measured performance comparing with DSDV, AODV, SMR and TORA. The simulation results say that the use of redundant routing considering node synchronization can reduce control traffic overhead and enhance packet delivery ratio and end-to-end delay in mobile ad-hoc networks. The details of simulation results are available at http://nlab.korea.ac.kr/~angus/wons2003-full.pdf

References

1. Introduction To Wireless and Mobile Systems, D. P. Agrawal, Qing-An Zeng, Brooks/Cole, a division of Thomson Learning, 2003.
2. J. Broch, D.B. Johnson, and D.A. Maltz, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," Internet Draft, draft-ietf-manet-dsr-00.txt, March 1998.
3. D. Jhonson and D. A. Maltz, Dynamic source routing in ad hoc wireless networks, in Mobile Computing, Kluwere Academic Publishers, 1996.
4. C. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced vector routing (DSDV) for mobile computers,” ACM SIGCOMM, Oct. 1994.
5. C. Perkins and E. Royer, “Ad Hoc On-Demand Distance Vector Routing,” Proceeding of IEEE WMCSA, Feb. 1999.
6. E.M. Royer and C.-K. Toh, “A Review of Current Routing Protocols for Ad-Hoc Mobile Networks," IEEE Personal Communications, vol. 6, no. 2, pp. 46-55, April 1999.
7. C-K Toh, “Associativity-based routing for ad hoc mobile networks,” Wireless Personal Communications, vol. 4, no. 2, pp. 1-36Mar. 1997.
8. Satyabrata Chakrabarti, Amitabh Mishra, “QoS issues in ad-hoc wireless networks”, IEEE communications magazine, Feb., 2001
9. The VINT Project, The UCB/LBNL/VINT Network Simulator-ns (version2), http://www.isi.edu/nsnam/ns.