PDAODMRP: An Extended PoolODMRP Based on Passive Data Acknowledgement

Shaobin Cai, Xiaozong Yang, and Ling Wang

Abstract: **An ad hoc network is a multi-hop wireless network. Its limited bandwidth and frequently changing topology require that its protocol should be robust, simple, and energy conserving. We have proposed PoolODMRP to reduce its control overhead greatly by its one-hop local route maintenance. However, PoolODMRP still has some shortcomings. In this paper, we propose PDAODMRP (passive data acknowledgement ODMRP) to extend PoolODMRP. Compared with PoolODMRP, PDAODMRP has the following contributions: (1) It knows the status of its downstream forwarding nodes by route information collected from data packets instead of BEACON signal of MAC layer; (2) it max simplifies the route information collected from data packets by pool nodes; (3) it adopts a dynamic local route maintenance to enforce its local route maintenance; (4) it adopts the route evaluation policy of NSMP (neighbor supporting multicast protocol). Compared with PoolODMRP, PDAODMRP has lower control overhead, lower data delivery delay, and lower data overhead.**

Index Terms: **Ad hoc network, local route recovery, multicast.**

I. INTRODUCTION

An ad hoc network [1] is a multi-hop wireless network, which can be rapidly deployed without any fixed infrastructure, and provide impromptu communication in hostile environment. The ad hoc network has its root in DARPA packet radio network [2], [3]. Any two un-neighboring nodes of the ad hoc network communicate by the packet relay of intermediate nodes. Hence, ad hoc network is a collection of mobile routers, which are interacted via wireless links and are free to move about arbitrarily.

Typical application areas of ad hoc network, which includes battlefields, emergency search, and rescue site, require lots of one to many and many to many communications. Therefore, more and more attentions are attracted by ad hoc multicast protocol. Compared with multiple unicasts, multicast makes full use of the inherent broadcast property of wireless communication, and minimizes link bandwidth consumption, source and router processing, and data delivery delay [4]. However, its limited battery and bandwidth, its fast changing topology resulted from speedy movement of its mobile nodes and the absence of central control point determine that the realization of multicast protocol for ad hoc network is more challenging than that of internet.

Lots of studies [5]–[11] on protocols for ad hoc network started with unicast protocols, and these unicast routing schemes can be classified into proactive routing scheme and reactive routing scheme based on their route determination system. The proactive routing scheme continuously makes routing decisions according to the variances of network topology. OLSR [5], DSDV [6], and WRP [7] are typical proactive routing protocols. The reactive routing scheme determines routes on an on-demand basis. In the reactive schemes, only when a node has packets to transmit, it queries network for its routes. TORA [8], DSR [9], AODV [10], and SSA [11] are typical reactive routing. In the proactive routing, routing information exchanges consume a great deal of radio resources, and these predetermined routes may rapidly lose their validity because the topology of ad hoc network changes rapidly. Studies [12]–[14] showed that the reactive protocols perform better than the proactive protocols.

Previous multicast protocols for ad hoc network, such as shared tree wireless network multicast [15], are proposed by adapting the existing internet multicast protocols. However, the adjusted protocols are not suitable for ad hoc network. Therefore, some multicast routing protocols, designed for ad hoc networks, have been proposed in the recent years [16]–[22]. The proposed multicast protocols for ad hoc network can be classified into two categories: Tree-based protocols and mesh-based protocols. In the tree-based schemes, a single shortest path between a source and a destination is selected out for data delivery. MAODV [16], AMRoute [17], and AMRIS [18] are typical tree-based schemes. In the mesh-based schemes, multiple paths are selected for data delivery. ODMRP [19]–[21] and CAMP [22] are typical mesh-based schemes.

Recent study [23] shows that the mesh-based schemes generally outperform the tree-based schemes. The study also shows that, among mesh-based schemes, ODMRP outperforms CAMP both in protocol efficient and data delivery ratio. Although ODMRP has some advantages, it still relies on periodically network-wide flooding, which are expensive operations in ad hoc networks [24], to maintain its forwarding mesh. According to that most link failure recoveries can be localized to a small region along previous routes [25], NSMP [26] and PatchODMRP [27] are proposed to save their control overhead by their local route maintenance systems.

Although PatchODMRP and NSMP reduce their control overhead by their local route maintenance, their local route maintenances is still large. In order to reduce local route maintenance scope further and acquire lower control overhead, we have proposed PoolODMRP [28], [29] to extend PatchODMRP by its pool node technology. PoolODMRP defines the un-forwarding neighbor nodes of forwarding nodes as pool nodes. And then the pool nodes collect route information from their received data packets to know the status of their neighbor forwarding nodes. PoolODMRP reduces its local route maintenance scope to one-hop with the aid of pool nodes, and reduces its control overhead greatly. However, it still has the following shortcomings: (1) It

Manuscript received July 26, 2004.

The authors are with the Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, email: caisb@hit.edu.cn, xzyang@hope.hit.edu.cn, lwang@ftcl.hit.edu.cn.

1229-2370/04/$10.00 © 2004 KICS

still knows the status of forwarding nodes by BEACON signal of MAC [30] layer; (2) its route collections from data packets consume many CPU resources; (3) its local route maintenance is weaker than that of PatchODMRP; (4) it selects all disjoint paths between a source and a member. Therefore, we have proposed PDAODMRP [31] (passive data acknowledge ODMRP). PDAODMRP extends PoolODMRP based on the passive acknowledgement function of data packets during their transmissions [3] to overcome the shortcomings of PoolODMRP.

The rest of the paper is organized as follows. First, we introduce the previous studies in Section II. Secondly, we analyze the characters of local route maintenance by mathematic analysis in Section III. Thirdly, we describe how PDAODMRP works in Section IV. And then, we analyze the performance of PDAODMRP by simulations in Section V. Finally, we draw a conclusion in Section VI.

II. OVERVIEW OF PREVIOUS STUDIES

In this section, the overview of ODMRP, NSMP, PatchODMRP, and PoolODMRP is given.

A. ODMRP

ODMRP is an on-demand ad hoc multicast protocol. In ODMRP, when a source has data packets to send out and the node doesn't know any route to destinations, it floods join query packets to set up its forwarding mesh. When a node receives a unduplicated join query packet, it updates its route table by information gotten from the received join query packet. And then, the node reduces TTL (time to live) value of the packet by 1, and relays the packet when the TTL value is still larger than 0. A member node of same group answers its received join query packets with a join reply packet.

When a node receives a join reply packet, it checks whether it is a downstream node defined in the downstream list of the packet. If the node is a downstream node, then the node marks itself as a new forwarding node. The new forwarding node broadcasts a new join reply packet, which is created according to its route table. Otherwise, the node discards the join reply packet. By the relay of forwarding nodes, the join reply packets reach the source. And, the nodes on the way, by which the join query packets reach the member, are marked as forwarding nodes.

B. NSMP

NSMP adopts a neighbor supporting local route discovery system to reduce its control overhead. In NSMP, a new source finds its route by broadcasting a FLOOD_REQ packet, including an upstream item to present which node deals with the packet last. When a node receives a FLOOD_REQ, it first computes the weight of the path by (1).

$$\text{Metric} = (1-\alpha)\times \text{FC} + \alpha\times \text{NC}, \quad 0 \le \alpha \le 1. \qquad (1)$$

In (1), FC presents the number of old forwarding nodes in the path, and NC presents the number of un-forwarding nodes in the path. When the value of α is low, there are more old forwarding nodes of same group in the new paths. Therefore, the efficiency of the path is higher, and the stability of the forwarding mesh is lower. When the value of α is higher, there are less forwarding nodes of other already existed paths of same forwarding group in the new path. Hence, the efficiency of the path is lower, and the stability of the forwarding mesh is higher. If the received packet is a non-duplicate packet, then the node records the upstream address of the FLOOD_REQ packet in its route table, and rebroadcasts out a new FLOOD_REQ packet with its address as new upstream address. Otherwise, the node discards the control packet, and only updates its route table when new path is more suitable.

When a member receives a FLOOD_REQ packet, it records its collected route information in its ReqCache and computes the weight of the path. If the member receives other FLOOD_REQ packets during its waiting time, it updates its ReqCache when new path is more suitable. After the waiting period, the member answers its received FLOOD_REQ packets with a REP packet with respect to its ReqCache.

When a node receives a REP packet, it checks whether it is a downstream node defined in the REP packet. If it does, the node marks itself as a forwarding node and broadcasts a REP packet of its own according to its routing table. Otherwise, it marks itself as a neighbor node, and discards the REP packet. By the relay of nodes, the REP packets arrive at the source node, and related nodes are marked as neighbor nodes and forwarding nodes. After forwarding mesh has been setup, normal source maintains its forwarding mesh mainly by periodically (by REQ_PERIOD) broadcasting LOCAL_REQ packet. The LOCAL_REQ packets are only relayed by forwarding nodes and neighbor nodes. A member answers its received LOCAL_REQ packets with a REP packet. The REP packet acknowledging to LOCAL_REQ packet works as that acknowledging to FLOOD_REQ packet.

Fig. 1 describes how NSMP works. In Fig. 1(a), node A is a source, node E is a receiver, nodes B and D are forwarding nodes, and nodes C, I, and K are neighbor nodes. Node A frequently broadcasts LOCAL_REQ packet to maintain its forwarding mesh. If a link between nodes B and D has broken during a flooding of LOCAL_REQ packet, then nodes B, C, D, E, K, and I relay the LOCAL_REQ packets (Fig. 1(b)). After receiving a LOCAL_REQ packet, member E answers the LOCAL_REQ packet with a REP packet. The REP packet arrives at node A by the relay of nodes D, C, and B. The REP packets mark the related nodes as forwarding nodes (B, C, and D) and neighbor nodes (K and I) (Fig. 1(c)). A new forwarding mesh is formed (Fig. 1(d)).

C. PatchODMRP

PatchODMRP extends ODMRP by its local route maintenance to prolong its join query interval, which is the period between two join query flooding. After setting up its forwarding mesh by network-wide flooding as ODMRP does, a source begins to send out its data packets. During transmission of data packets, forwarding nodes know the status of their neighbors by BEACON signal of MAC layer. When a forwarding node finds that a link between its upstream node and itself is broken, it floods an ADVT packet to do its local route maintenance.

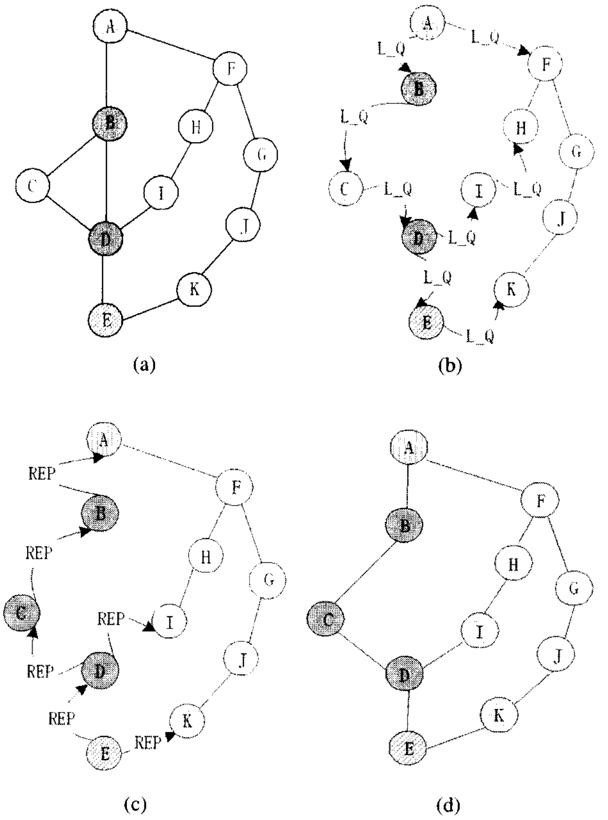


Fig. 1. An example of how NSMP protocol does its local route maintenance.

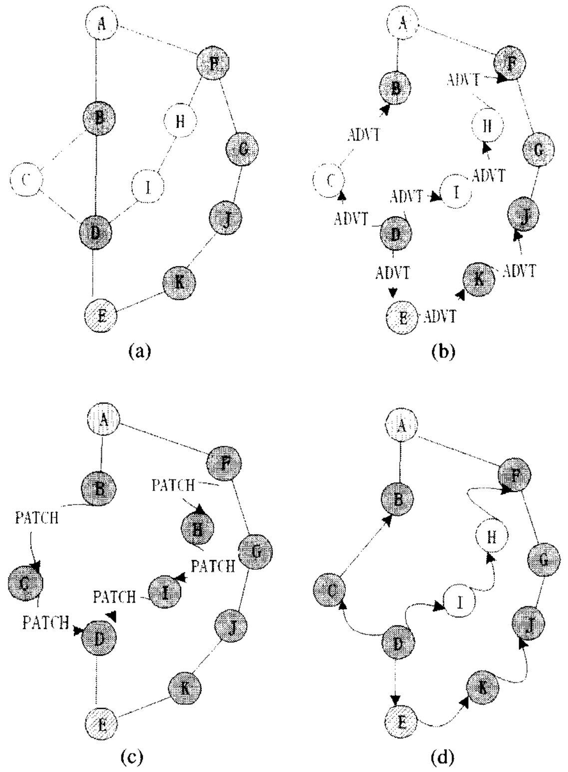


Fig. 2. An example of how PatchODMRP protocol does its local route maintenance.

When a forwarding node receives an ADVT packet, it checks whether it meets the following requirements of the ADVT packet: (1) The forwarding node and the ADVT source node belong to the same groups; (2) both the forwarding node and the ADVT source node relay data for same sources; (3) the forwarding node isn't farther from the sources than the ADVT source node is. If the forwarding node meets all the requirements, then it answers the ADVT packet with a PATCH packet. Otherwise, it reduces TTL value of the ADVT packet by 1, and relays the ADVT packet when its TTL value is still larger than 0.

A PATCH packet transmits in network as a join reply packet does. The PATCH packets arrive at the ADVT source node, and mark the nodes on paths temp forwarding nodes. The ADVT source node selects out the shortest path from the repaired paths, and informs the result to related node. The nodes, which are not on the shortest path, are not temporary forwarding nodes again.

Fig. 2 describes how PatchODMRP works. In Fig. 2(a), Multicast mesh is made up of three fractions: A source node A, a receiver node E, and six forwarding nodes (B, D, F, G, J, and K). Node B is upstream node of node D. When a link between node B and node D broke because of the movements of node B, node D can not receive BEACON signal from node B. And then, node D broadcasts an ADVT packet to do its local route maintenance. And then, nodes C, E, H, I, and K relay their received ADVT packet (Fig. 2(b)). Nodes B and F answer their received ADVT packet with PATCH packets. The PATCH packets reach node D by the relay of other nodes, and nodes C, H, and I are marked as temporary forwarding nodes during the transmission of PATCH packets (Fig. 2(c)). After receiving PATCH packets, node D selects out the shortest path, and informs the result to related nodes (Fig. 2(d)). Nodes H and I are not temp forwarding nodes again, and a new forwarding mesh is formed.

Table 1. Pool table.

Upstream Node	Birth	MG ID	Source	Count	Timer
			Source	Count	Timer
		MG ID	Source	Count	Timer
			Source	Count	Timer

D. PoolODMRP

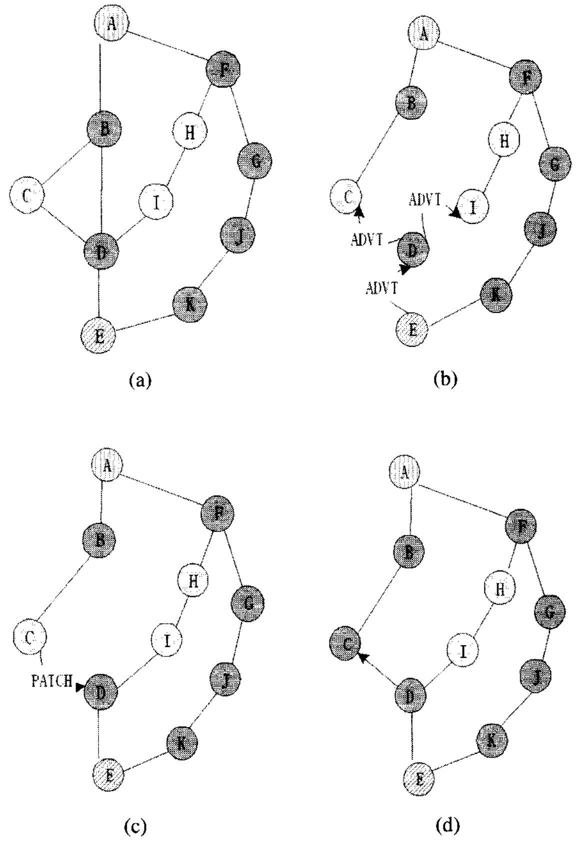
PoolODMRP extends PatchODMRP by its pool node technology, and reduces its local route maintenance scope to one-hop. PoolODMRP defines the neighbor un-forwarding nodes of forwarding nodes as pool nodes, and the pool nodes store route information collected from their received data packets into pool table Table 1.

Pool table exists in pool node, and is used to record route information gotten from a received data packet. In pool table, upstream node field records a node address, from which the pool node received data packet; birth field presents a time, when an entry is inserted in pool table; MG ID field is a multicast group address of a received data packet; source field records an address of a source node, which originates the data packet; count field presents the shortest distance between the source node and the node; timer field presents a time, when the data packet is received. Each subentry of the table has a lifetime. When a subentry can't be updated by a data packet, it expired, and is deleted. When all subentries of an entry are deleted, the entry is deleted too.

PoolODMRP sets up its forwarding mesh as ODMRP does. After setting up its forwarding mesh by network-wide flooding, a source begins to send out its data packets. When a forwarding node receives an unduplicated data packet, it relays the data packet. When a pool node receives a data packet, it records MG ID, source address, upstream address, receiving time, and how many times the data packet is relayed in its pool table.

In PoolODMRP, forwarding nodes still know the status of their neighbors by BEACON signal of MAC layer. When a forwarding node finds that a link between its upstream node and itself broke, it broadcasts out an ADVT packet to do its local route maintenance. When a pool node receives an ADVT packet, it checks whether one of its neighbor forwarding nodes meets requirements of the ADVT packet by its pool table. If one of its neighbor forwarding nodes meets the requirements, then the pool node answers the ADVT packet with a PATCH packet. Otherwise, the pool node discards the ADVT packet. After receiving PATCH packets, the ADVT source node selects the most stable path from these repaired paths, and informs the result to these related nodes. And then, the related nodes are marked as forwarding nodes.

Fig. 3 describes how PoolODMRP works. In Fig. 3(a), multicast mesh is made up of three fractions: A source node A, a receiver node E, and six forwarding nodes (B, D, F, G, J, and K). Node B is upstream node of node D. When a link between node B and node D broke because of the movements of node B, node D can not receive BEACON signal from node B. And then, node D broadcasts an ADVT packet to do its local route maintenance (Fig. 3(b)). After receiving an ADVT packet, pool node C, which satisfies requirements of the ADVT packet, answers the ADVT packet with a PATCH packet. The PATCH packet reaches node D (Fig. 3(c)). Node D informs local route maintenance result to node C. Node C is marked as forwarding nodes, and a new forwarding mesh is formed (Fig. 3(d)).



(a) (b)

(c) (d)

Fig. 3. An example of how PoolODMRP do its local route maintenance.

III. MATHEMATICAL ANALYSIS

In the Section II, the above three protocols amend their data forwarding mesh by different methods, and their local route maintenance methods have different local route maintenance scope. However, none of these protocols explain how large enough a local route maintenance scope should be to amend all link failures. In this section, we analyze the characters of local route maintenance by a random graph. The graph presents an ad hoc network, which is formed by 50 mobile nodes in a square (1000 m×1000 m), and each node has n' ($n' < 50$) neighbors.

In the ad hoc network, if the link failure is caused by the node's failure, then the possibility that the link failure can be amended is equal to the possibility that an i-link path exists between any two un-neighboring nodes.

Statement 1: In an ad hoc network, the possibility p' that there is only an i ($i > 1$)-link path between any un-neighboring nodes is

$$p'_i = \frac{n'^i \times (1 - \frac{n'}{n})^{\frac{i(i-1)}{2}}}{n}, (i > 1). \tag{2}$$

Proof: In the ad hoc network, the possibility that a node is a neighbor of another node is $\frac{n'}{n}$. The i ($i > 1$)-link path between any un-neighboring nodes, from a source to a destination, is made up of two fractions.

1. The $i - 1$-link path from the source to one of the neighbors of the destination.
2. The one-link path from the destination to one of its neighbors.

Then, the probability that an $i - 1$-link path exists between any two un-neighboring nodes is the product of the following three fractions.

1. The number of neighbors of the destination.
2. The probability p'_{i-1} that the source reaches one of the neighbors of the destination.
3. The probability $(1 - \frac{n'}{n})^{i-1}$ that the destination is a neighbor of the nodes, which is the first nodes on the link path.

Now, we can know that the probability p'_i that there is an i ($i > 1$)-link path between any un-neighboring nodes is

$$\begin{aligned} p'_i &= \frac{n'}{n}, i = 1, \\ p'_i &= p'_{i-1} \times n' \times \left(1 - \frac{n'}{n}\right)^{i-1} \end{aligned}$$

$$\begin{aligned} &= n'_{i-1} \times \left(1 - \frac{n'}{n}\right)^{\frac{i(i-1)}{2}} \\ &= n'_i \times \left(1 - \frac{n'}{n}\right)^{\frac{i(i-1)}{2}} \times \frac{1}{n}, i > 1. \end{aligned}$$

□

Example 1: In the ad hoc network, whose density is ρ, and whose node's communication radius is r, the probability that the link failure caused by the node failure can be recovered by local route maintenance is

$$p'_i = \frac{1}{50}(\pi r^2 \rho)^i \times \left(1 - \frac{\pi r^2 \rho}{50}\right).$$

If all link failures, caused by the node failure, can be amended by two-hop local route maintenance, then

$$p'_i = \frac{1}{50}(\pi r^2 \rho)^2 \times \left(1 - \frac{\pi r^2 \rho}{50}\right) \geq 1$$

$$\Longrightarrow r \geq 0.22 \text{ (km)}.$$

If all link failures, caused by the node failure, can be amended by three-hop local route maintenance, then

$$\frac{1}{50}(\pi r^2 \rho)^2 \times \left(1 - \frac{\pi r^2 \rho}{50}\right) + \frac{1}{50}(\pi r^2 \rho)^3 \times \left(1 - \frac{\pi r^2 \rho}{50}\right)^3 \geq 1$$

$$\Longrightarrow r \geq 0.153 \text{ (km)}.$$

When 0.2 (km), among the link failures caused by the movements of nodes, 69.1% can be amended in two hops and all of them can be amended in three hops.

Statement 2: In the ad hoc network, the possibility that there is another i $(i > 1)$-link path between any neighbor nodes is

$$p_i = \frac{n'^i \times (1 - \frac{n'}{n})^{\frac{(i+1)(i-2)}{2}}}{n}, (i > 1). \tag{3}$$

Proof: According to statement 1, we know the possibility that there is only an i $(i > 1)$-link path between any un-neighboring nodes. Then the possibility that there is an i $(i > 1)$-link path between any two nodes is

$$p_i = \frac{p'_i}{1 - \frac{n'}{n}}.$$

And thus, the probability that there is an i $(i > 1)$-link path between any two neighbor nodes is

$$\begin{aligned} p_i &= \frac{n'}{n} \times \frac{p'_i}{1 - \frac{n'}{n}} \\ &= \frac{n'^i \times (1 - \frac{n'}{n})^{\frac{(i+1)(i-2)}{2}}}{n}, (i > 1). \end{aligned}$$

□

When a forwarding node receives an unduplicated data packet, it relays the data packet. And then, it checks the status of its downstream nodes. If it can't receive data packet from one of its downstream nodes for 1 s, it thinks a link between it and its downstream forwarding node is broken, and broadcasts an ADVT packet to do its local route maintenance.

Example 2: In the ad hoc network, whose density is ρ , whose node's communication radius is r, the probability that the link failure, is caused by the node movement can be recovered by local route maintenance is

$$p'_i = \frac{1}{50}(\pi r^2 \rho)^i \times \left(1 - \frac{\pi r^2 \rho}{50}\right)^{\frac{(i+1)(i-2)}{2}}$$

If there exists at least one 2-link path between any two neighbors, then

$$p'_i = \frac{1}{50}(\pi r^2 \rho)^2 \geq 1$$

$$\Longrightarrow r \geq 0.212 \text{ (km)}.$$

If there exists at least one 3-link path between any two neighbors, then

$$\frac{1}{50}(\pi r^2 \rho)^2 + \frac{1}{50}\left(\pi r^2 \rho)^3 \times (1 - \frac{\pi r^2 \rho}{50}\right)^2 \geq 1$$

$$\Longrightarrow r \geq 0.105 \text{ (km)}.$$

When 0.2 (km), among the link failures caused by the movements of nodes, 79.2% can be amended in two hops and all of them can be amended in three hops.

We have analyzed the relationship between the possibility that a link failure can be amended and the scope of local route maintenance. Now, we analyze the control overhead of the local maintenance. The control overhead of local route maintenance is mainly determined by the control overhead created by its local flooding. The local flooding overhead is equal to the number of nodes, which broadcast the control packets, and the local flooding overhead is $\pi[(i-1)r]^2 \times 50, (i > 1)$. In the ad hoc network, which has 50 nodes with 0.2 km communication radius, when the local route maintenance scope is 1 hop, only one node broadcasts control packets, then its local flooding overhead is only 2% of global flooding overhead; when the local route maintenance scope is 2 hops, its local flooding overhead is only 12.56% of global flooding overhead; when the local route maintenance scope is 3 hops, then its local flooding overhead is only 50.24% of global flooding overhead.

IV. PDAODMRP

The characters of the local route maintenances of the three above protocols can be known from the above mathematical analysis. The local route maintenance of PatchODMRP is strong enough to amend link failures. However, its local route scope is larger and its local control overhead is higher. Compared with PatchODMRP, and NSMP has a little weaker local

Table 2. The modified ADVT packet.

MG ID	ADVT SrcAddr	Last Address	Next Address	Next-Next Address	TTL

Table 3. The modified forwarding table.

MG ID	T	Life Time	Next	Time	Next-Next
					Next-Next
			Next	Time	Next-Next
					Next-Next

Table 4. The modified pool table.

Upstream Address	Time

route maintenance because neighbor nodes move away sometimes. Among these protocols, PoolODMRP has the smallest local route maintenance scope, and its local control overhead is much lower. However, its local route maintenance is weaker, and its local route maintenance can't amend all link failures.

A. Data Structure of PDAODMRP

In order to realize its own local route maintenance and data route collection, PDAODMRP defines its own data structure and its own control packets: A modified join reply packet, a modified PATCH packet, a modified ADVT packet Table 2, a data ACK packet, a MEM_REQ packet, a MEM_LEV packet, a modified forwarding Table 3, and a modified pool Table 4.

Both modified join reply packet and modified PATCH packet have a new field, last-last address field. The item presents which node deals with the packet before the last one. Therefore, a node, which receives a join reply packet or a PATCH packet, can know two neighbor forwarding nodes on a path.

In the modified ADVT packet, MG ID field is the numeric identifier of a multicast group; ADVT SrcAddr field is the address of a node, which initiates the ADVT packet; last address field is the address of a node dealing the packet last; next address field is the address of a node, which is a downstream node of the ADVT source node; next-next address field is the address of a node, which is a downstream node of the downstream node.

In the modified forwarding table, MG ID is a numeric identifier of a multicast group; T item marks whether a forwarding node is a false forwarding node or not; life time item presents when a forwarding node expires; next item presents which node is a downstream node of the forwarding node; time item presents when a node receives a data packet from this downstream node; next-next item presents which node is a downstream node of the downstream node.

B. Data Packet Relay

PDAODMRP can select the most suitable path between a source and a member by its route evaluation policy as NMSP does. And, each forwarding node knows its downstream nodes and the downstream-downstream nodes by the route information collected from modified Join reply packets. After its forwarding mesh founded, a source begins to broadcast its data packets. When a node receives a data packet, it deals with the data packet as follows.

- When a forwarding node receives an unduplicated data packet, it relays the data packet. And then, it checks the status of its downstream nodes. If it can't receive data packet from one of its downstream nodes for 1 s, it thinks a link between it and its downstream forwarding node is broken, and broadcasts an ADVT packet to do its local route maintenance.
- When a forwarding node receives a data packet from one of its downstream forwarding nodes, it records its receiving time, and discards this data packet.
- When a pool node receives a data packet, it records both the last address and receiving time of the data packet.
- When a pure receiver receives an unduplicated data packet, it acknowledges the data packet by a data ACK packet.
- When a pure receiver receives an unduplicated data packet, it acknowledges the data packet by a data ACK packet.
- When a forwarding node (or a pool node) receives a data ACK packet, it records both the last address and receiving time of the data ACK packet.

Therefore, forwarding nodes know the status of its downstream forwarding nodes, determine whether it does its local route maintenance, and pool nodes know the status of their neighbor forwarding nodes.

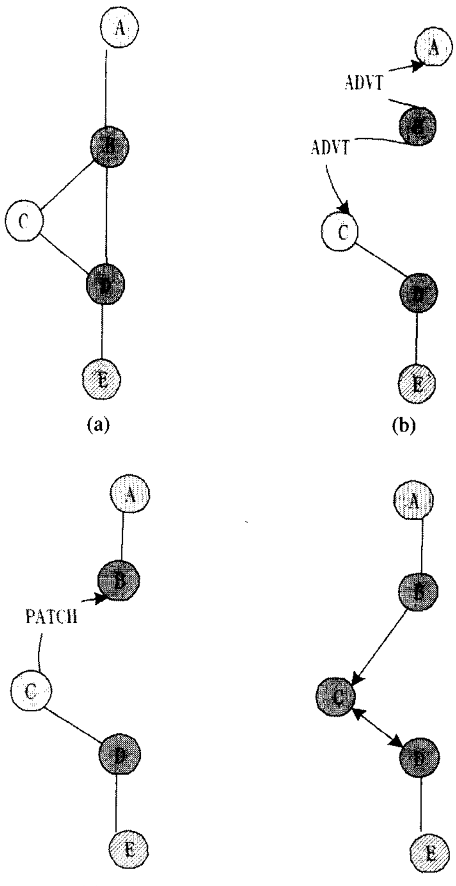
C. Local Route Maintenance

PDAODMRP adopts a dynamic local route maintenance policy. When a forwarding node does its local route maintenance, it first does its one-hop local route maintenance. If it can't receive a PATCH packet for a waiting time, it does its two-hop local route maintenance. By the dynamic local route maintenance, PDAODMRP acquires its stronger local route maintenance at the cost of little more control overhead.

In PDAODMRP, when a forwarding node knows that a link between one of its downstream nodes and itself is broken, it broadcasts an ADVT packet to do its local route maintenance. When a node receives an ADVT packet, it checks whether one of its neighbors is a downstream node (or downstream-downstream node) defined in the ADVT packet by its pool table. If it does, the pool node answers the ADVT packet with a PATCH packet. Otherwise, the node records the route of the ADVT packet, reduces the value of TTL by 1, and rebroadcasts the ADVT packet when its TTL value is still larger than 0.

The PATCH packet reaches the ADVT source node by the reverse way, by which the ADVT packet reaches the answered node. And, the PATCH packet marks the nodes on the paths as false forwarding nodes. The false forwarding nodes record the downstream node and downstream-downstream node, acquired from PATCH packets, in their forwarding table. The ADVT source node answers its first received PATCH packet, and informs the results to related nodes. The related nodes are marked as forwarding nodes, and a new forwarding node, which doesn't know its downstream-downstream node, acquires the address of its downstream-downstream node from its downstream node. At this time, the local route maintenance is finished.

Now, an example is given to explain how PDAODMRP works. In Fig. 4, when a link between node B and node D



(a) (b)

(c) (d)

Fig. 4. An example of how PDAODMRP protocol does its local route maintenance.

breaks because of the movements of node B, node B can't receive data packets from node D. Node B broadcasts an ADVT packet to do its local route maintenance (Fig. 4(b)). When its neighbor nodes A and C receive the ADVT packet, node C answers the ADVT packet with a PATCH packet (Fig. 4(c)). Node B answers its first received PATCH packet, and informs related nodes. Node C is marked as forwarding node, and node C acquires its downstream-downstream node address from node D (Fig. 4(d)). A new forwarding mesh is formed.

In Fig. 4, if another node L exists between nodes C and D, then PoolODMRP can't repair the link failure between nodes B and D by its local route maintenance. However, PatchODMRP and PDAODMRP can repair the link failure by their local route maintenance. Therefore, the local route maintenance of PatchODMRP and PDAODMRP is much stronger than that of PoolODMRP.

D. Joining and Leaving Group

When a node wants to join a group as a receiver, it broadcasts a MEM_REQ packet. The MEM_REQ packet transfers in network as a join query packet does. When a source node, a forwarding node or a pool node of the group receives a MEM_REQ packet, it sends out a join reply packet. The join reply packets acknowledging to the MEM_REQ packets are relayed toward the new receiver in the same way as join reply packets acknowledging to join query packet, and some nodes are marked as forwarding nodes.

When a member leaves a group, it sends out a MEM_LEV control packet to inform its upstream forwarding nodes. After receiving the MEM_LEV packet, the upstream node checks whether the member (or the downstream node) is its only one downstream node. If it does, the forwarding node sets itself a normal node, and broadcasts a new MEM_LEV control packet. Then, the forwarding nodes, which only relay data packet for the leaved member, are marked as normal nodes.

E. Join Query Interval and Lifetime of Forwarding Nodes

In all these five protocols, join query interval (or FLOOD_PERIOD) and forwarding nodes' lifetime are important factors that affect the performance of these five protocols. When the join query interval (or FLOOD_PERIOD) is too long, route information, acquired by network-wide forwarding mesh reconfiguration, can't match the rapidly changing topology of ad hoc network. When the join query interval (or FLOOD_PERIOD) is too short, frequent network-wide forwarding mesh reconfigurations create lots of control packets, and the performance of these protocols decreases greatly. When the lifetime of forwarding node is too long, there are too many old forwarding nodes in forwarding mesh. Although these large amount of old forwarding nodes can help amending some link failures, the large amount of old forwarding nodes increase the data overhead of these protocols greatly. When the lifetime of forwarding node is too short, there are too few old forwarding nodes to amend most of link failures, and data delivery ratio of these protocols decreases greatly.

According to definitions of NSMP [23], PatchODMRP [24], and PoolODMRP [28], [29], the join query interval and the lifetime of forwarding nodes of these protocols are given as follows.

- In ODMRP, the join query interval and the lifetime of forwarding nodes are set as 3 s and 9 s, respectively.
- In NSMP, the FLOOD_PERIOD, REQ_PERIOD, and the lifetime of forwarding nodes are set as 20 s, 3 s, and 9 s, respectively.
- In PatchODMRP, the join query interval, the lifetime of forwarding nodes and the lifetime of temp forwarding nodes are set as 200 s, 600 s, and 67 s, respectively.
- In PoolODMRP and PDAODMRP, the join query interval and the lifetime of forwarding nodes are set as 200 s and 270 s, respectively.

V. SIMULATION ANALYSIS

GloMoSim [32] is used here to realize the simulation of PDAODMRP protocol. In the simulations, 50 wireless mobile nodes, which move around over a square (1000 m×1000 m), form an ad hoc network. The communication radius of these nodes is 200 m. During a 1000 s simulation period, the nodes move according to the "random waypoint" model without pause time, and a multicast source generates 512-byte data packets with constant bit rate (CBR) of ten packets per second. In order to evaluate the performance of these protocols, we use the following metrics.

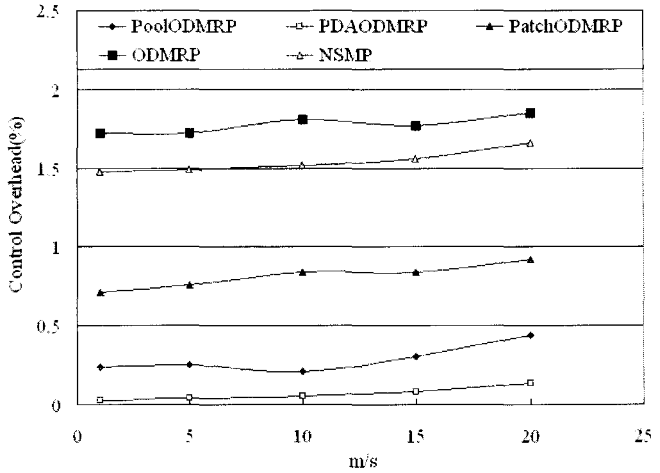


Fig. 5. Control overhead as a function of node speed.

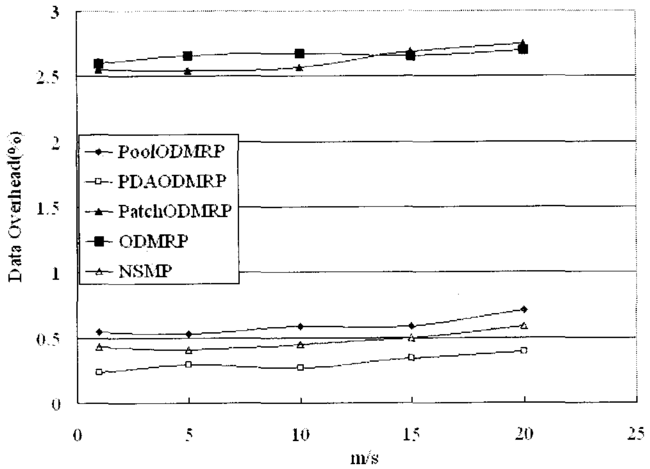


Fig. 6. Data overhead as a function of node speed.

- The data packet delivery ratio [14]. It represents the ratio between the average number of successfully received non-duplicate data packets at each receiver and the average number of data packets that should be received by each receiver. It is mainly determined by the robustness of forwarding mesh.
- The number of data transmissions per data packet delivered [14]. It represents the ratio between the number of data packets generated in the network and the number of successfully received unduplicated data packets at receivers. It is mainly determined by the efficiency of forwarding mesh.
- The number of control packets per data packet delivered [14]. It represents the ratio between the number of control packets issued in the network and the number of successfully received data packets at receivers. It is mainly determined by flooding period and local route maintenance scope.
- The data delivery delay between two nodes. It represents the average time, which a data packet uses to transmit from one forwarding node to another forwarding node. It is mainly determined by the robustness of forwarding mesh.
- The Number of Instructions, used to deal with control packets and data packets, per Data Packet Delivered. It represents the ratio between the number of instructions dealing with control packets and data packets and the number of successfully received data packets at each receiver.
- The number of message, buffered at nodes, per data packet delivered. It represents the ratio between the number of messages buffered at nodes and the number of successfully received data packets at receivers. It is mainly determined by the number of messages buffered at each node and the number of nodes, which buffer these messages.

Furthermore, all these metrics above are also greatly affected by the wireless bandwidth acquired by a data packet for its transmission.

A. Node Speed

Now, we test the impact of node speed on the performance of these protocols to evaluate the scalability of these protocols in this section. In order to test the impact, which the node speed enforces on the robustness of forwarding mesh, we set the multicast group as simple as possible. In following experiments, we set the number of sources as 1, and set the members of multicast group as 5. Therefore, there is enough wireless bandwidth for data packet transmission, and only the robustness of the forwarding mesh affects the performance of these protocols.

Compared with ODMRP and PatchODMRP, the forwarding mesh of PDAODMRP is as strong as that of ODMRP even though ODMRP reconfigures its forwarding mesh most frequently, not only because its routes are selected by its route evaluation policy as NSMP does but also because its local route maintenance is strong enough to amend all link failures, caused by its longer join query interval. Compared with other protocols, NSMP, and PoolODMRP have weaker forwarding meshes because they have weaker local route maintenance and NSMP has fewer paths.

Fig. 5 describes the impact of node speed on the average control overhead of these protocols. Among these protocols, PDAODMRP has the lowest control overhead as PDAODMRP doesn't use BEACON signals to know the status of forwarding nodes, in spite of the fact that the local route maintenance scope of PDAODMRP is little larger than that of PoolODMRP. Compared with NSMP and PatchODMRP, PoolODMRP has lower control overhead since it not only has the longest network-wide flooding interval but also has the smallest local route maintenance scope. Compared with ODMRP and NSMP, the control overhead of PatchODMRP, PoolODMRP, and PDAODMRP increases more as they have to do more local route maintenances to amend link failures caused by faster movements of nodes when the node speed increases.

Fig. 6 describes the impact of node speed on the average data overhead of these protocols. PDAODMRP has the lowest data overhead not only because of its most efficient forwarding mesh but also because of its shorter forwarding node lifetime. Compared with ODMRP and PatchODMRP, PoolODMRP have lower data overhead because of its shorter forwarding node lifetime; and its data overhead is similar to that of NSMP. Compared with ODMRP, the data overhead of NSMP, PatchODMRP,

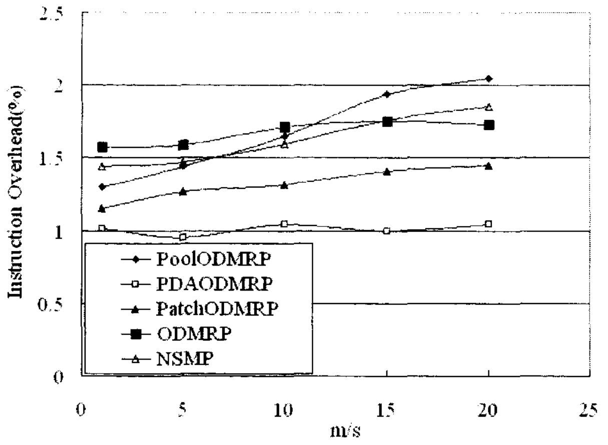


Fig. 7. Instruction overhead as a function of node speed.

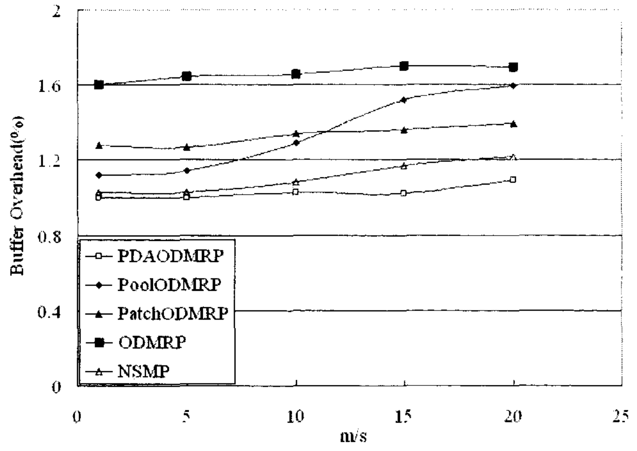


Fig. 8. Buffer overhead as a function of node speed.

PoolODMRP, and PDAODMRP increases more because their local route maintenances create more new forwarding nodes to amend link failures when node speed increases.

Figs. 7 and 8 describe the impact of node speed on the instruction overhead and buffer overhead of these protocols, respectively. In Figs. 7 and 8, the instruction overhead and buffer overhead of PDAODMRP are set 1 when node speed is 1 m/s. Among these protocols, PDAODMRP has the lowest instruction overhead and lowest buffer overhead not only because of its lowest control overhead and its lowest data overhead but also because its simplified route collection only collects and buffers few messages from data packets even though ODMRP, NSMP, and PatchODMRP do not collect and buffer route information from data packets. When the number of sources increases, the instruction overhead and buffer overhead of all these protocols increases; and the instruction overhead and buffer overhead of PatchODMRP, PoolODMRP, and PDAODMRP increase more because their control overhead and their data overhead increase more.

Figs. 9 and 10 describe the impact of node speed on the average data delivery ratio and data delivery delay of these protocols, respectively. ODMRP, PatchODMRP, and PDAODMRP have similar data delivery ratio because the robustness of their forwarding meshes is almost identical. Compared with other protocols, NSMP and PoolODMRP have lower data delivery ratio because of their weaker forwarding mesh. Since the forwarding mesh of all these protocols is strong enough, the data delivery delay of these protocols is mainly determined by the wireless bandwidth acquired by a data packet for its transmission. Therefore, PDAODMRP has the lowest data delivery delay because of its lowest control overhead.

In Figs. 9 and 10, when node speed increases, the data delivery ratio of these protocols decreases and the data delivery delay increases. In these protocols, ODMRP, PatchODMRP, and PDAODMRP have similar forwarding mesh; NSMP and PoolODMRP have weaker forwarding mesh. Therefore, the data delivery ratio of NSMP and PoolODMRP decreases more, and the data delivery delay of NSMP and PoolODMRP increases more, when node speed increases.

From the simulation results, a conclusion can be drawn that the forwarding mesh maintained by the local route maintenance of PDAODMRP is as robust as that of ODMRP and PatchODMRP. It is stronger than that of NSMP and PoolODMRP, and it is strong enough to stand against fast movement of the mobile nodes.

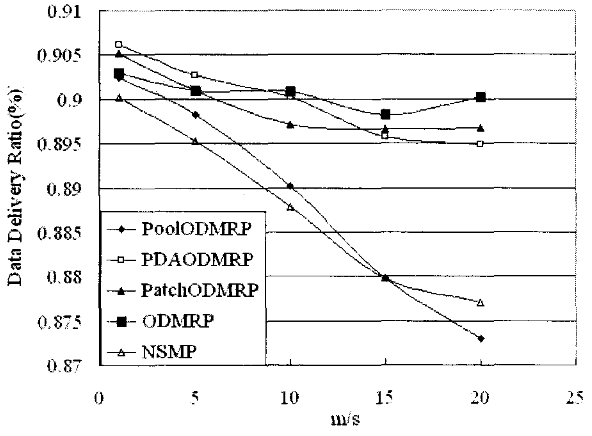


Fig. 9. Data delivery ratio as a function of node speed.

B. Sources

In this subsection, we test the impact of the number of the sources of a multicast group to evaluate the scalability of these protocols. In the following experiments, we set the max speed of nodes as 10 m/s, and set the number of multicast group as 20.

When there are many sources in multicast groups, the sources maintain their forwarding meshes by a large amount of control packets. The large amount of control packets occupies most of the limited wireless bandwidth, and data packets can not acquire enough wireless bandwidth for their transmission. Therefore, many data packets loss during their transmissions, and performance of these protocols is greatly affected.

Fig. 11 describes the impact of the source number of a multicast group on the average control overhead of these protocols.

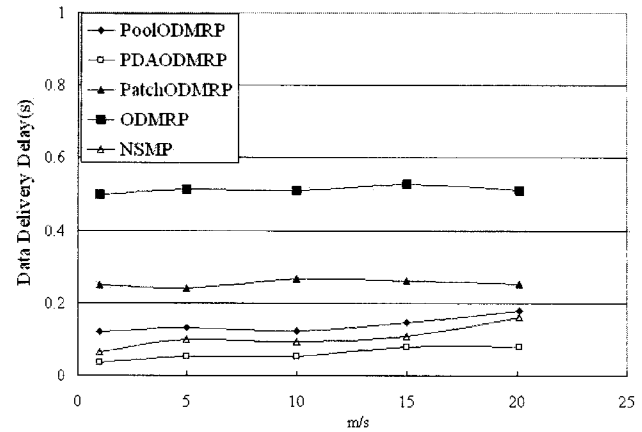


Fig. 10. Data delivery delay as a function of node speed.

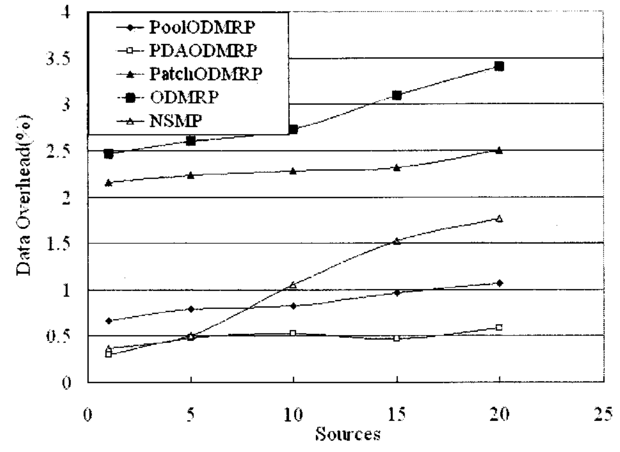


Fig. 12. Data overhead as a function of sources.

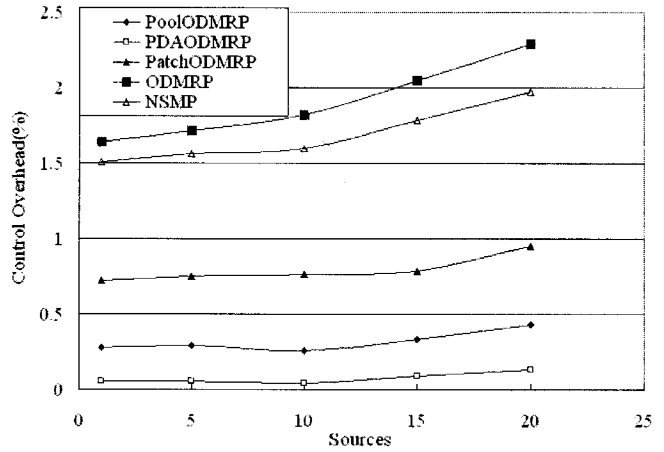


Fig. 11. Control overhead as a function of sources.

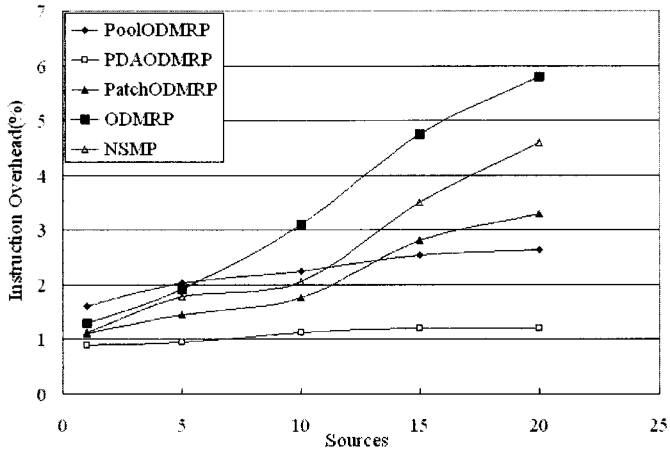


Fig. 13. Instruction overhead as a function of sources.

Among these protocols, PDAODMRP has the lowest control overhead because it doesn't use BEACON signals to know the status of forwarding nodes. When the number of sources increases, the control overhead of these protocols increases, and the control overhead of PDAODMRP increases the slowest because of its lowest control overhead. Therefore, the control overhead of PDAODMRP scales best when the number of sources increases.

Fig. 12 describes the impact of the source number of multicast group on the average data overhead of these protocols. PDAODMRP has the lowest data overhead because of its most efficient forwarding mesh and its shorter forwarding node lifetime. When the he number of sources increases, the data overhead of all these protocols increases, and the data overhead of PDAODMRP increases slowest because of its lowest control overhead. Hence, the data overhead of PDAODMRP scales best when the number of sources increases.

Figs. 13 and 14 describe the impact of the source number of multicast group on the average instruction overhead and buffer overhead of these protocols, respectively. PDAODMRP has the lowest instruction overhead and buffer overhead because of its lowest control overhead and its lowest data overhead even though ODMRP, NSMP, and PatchODMRP do not collect and buffer route information from data packets. When the number of sources increases, the instruction overhead and buffer overhead of all these protocols increases, and the instruction overhead and the buffer overhead of PDAODMRP increase the slowest because its control overhead and data overhead increase the slowest. Hence, the instruction overhead and buffer overhead of PDAODMRP extends best when the number of sources increases.

Fig. 15 describes the impact of the source number of multicast group on the average data delivery ratio of these protocols. It also can be divided into two cases for discussion.

- When there are few sources in a group, the data delivery ratio of these protocols is mainly determined by the robustness of forwarding mesh. ODMRP, PatchODMRP, and PDAODMRP have similar data delivery ratio because the robustness of their forwarding meshes is almost identical. Compared with other protocols, NSMP, and PoolODMRP have lower data delivery ratio because of their weaker forwarding mesh.

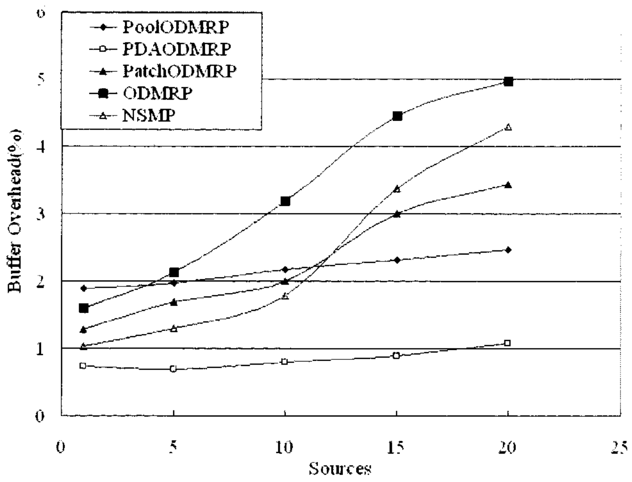


Fig. 14. Buffer overhead as a function of sources.

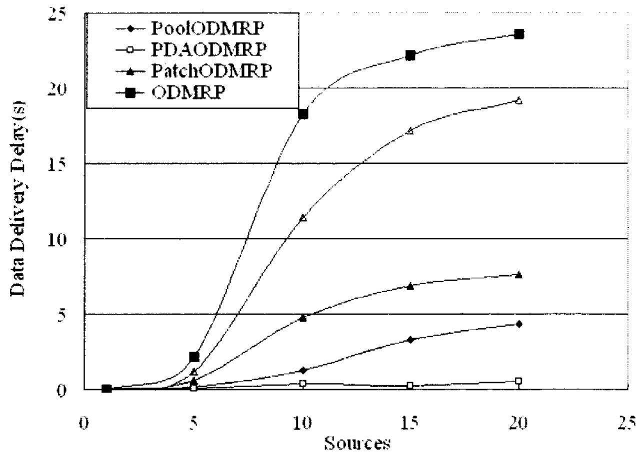


Fig. 16. Data delivery delay as a function of sources.

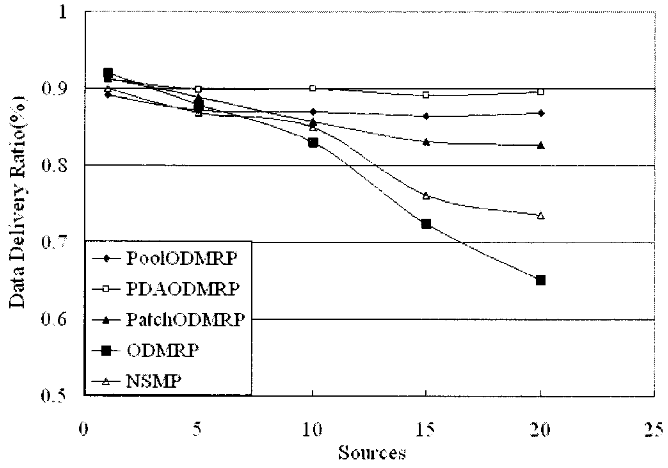


Fig. 15. Data delivery ratio as a function of sources.

- When there are many sources in a group, the data delivery ratio of these protocols is greatly affected by the wireless bandwidth for data packet transmission. PDAODMRP has the highest data delivery ratio among these protocols because of its lowest control overhead.

When the number of sources increases, the data delivery ratio of all these protocols decreases; and the data delivery ratio of PDAODMRP decreases the slowest because of its lowest control overhead. Hence, the data delivery ratio of PDAODMRP extends best when the number of sources increases.

Fig. 16 describes the impact of the source number of multicast group on the average data delivery delay of these protocols. The forwarding meshes of these protocols are strong enough for guaranteeing the data delivery, and their data delivery delay is mainly determined by the wireless bandwidth for data packet transmission. Hence, PDAODMRP has the lowest data delivery delay due to its lowest control overhead. When the number of sources increases, the data delivery delay of all these protocols increases; the data delivery delay of PDAODMRP increases the slowest since it has the lowest control overhead. Hence, the date delivery delay of PDAODMRP extends best when the number of sources increases.

From the simulation results above, a conclusion can de drawn that the control overhead of PDAODMRP is so small that it can stand best against the increase of the source number of the group. When the source number exceeds 10 in the group, its advantage is more prominent.

C. Members

Now, we test the impact of the members of groups on the performance of these protocols to evaluate the scalability of these protocols. In following experiments, the node max speed is 10 m/s; there are 3 groups in the ad hoc network, each of which only has 1 source.

When there are more members in groups, more nodes are marked as forwarding nodes. However, the forwarding nodes marked for new members do not increase the robustness of the forwarding mesh greatly because the old forwarding mesh is strong enough. When more nodes are marked as forwarding nodes, more forwarding nodes belong to all these three groups. Hence, more packets compete for the scare wireless bandwidth of these forwarding nodes. And, the performance of these protocols is greatly affected.

Figs. 17–20 describe the impact of members on the average control overhead, data overhead, instruction overhead, and buffer overhead of these protocols, respectively. PDAODMRP has the following characters.

- It has the lowest control overhead because it doesn't use BEACON signals to know the status of forwarding nodes.
- It has the lowest data overhead because of its most efficient forwarding mesh and its shorter forwarding node lifetime. Compared with ODMRP and PatchODMRP, PoolODMRP has lower data overhead because of its shorter forwarding node lifetime.
- It has the lowest instruction overhead and buffer overhead because of its lowest control overhead and its lowest data overhead.

When the members of groups increase, the control overhead, the data overhead, the instruction overhead and the buffer over-

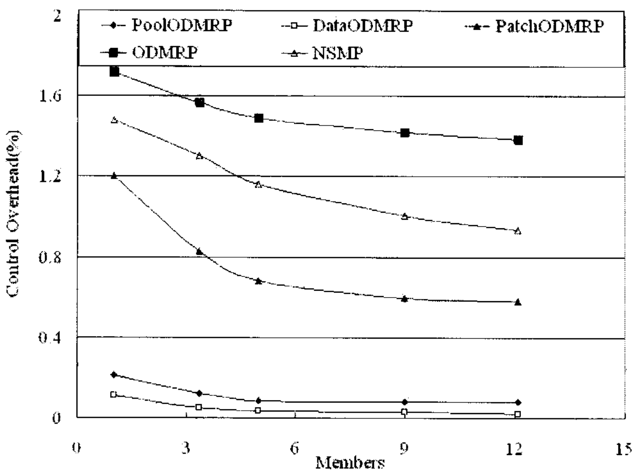


Fig. 17. Control overhead as a function of members.

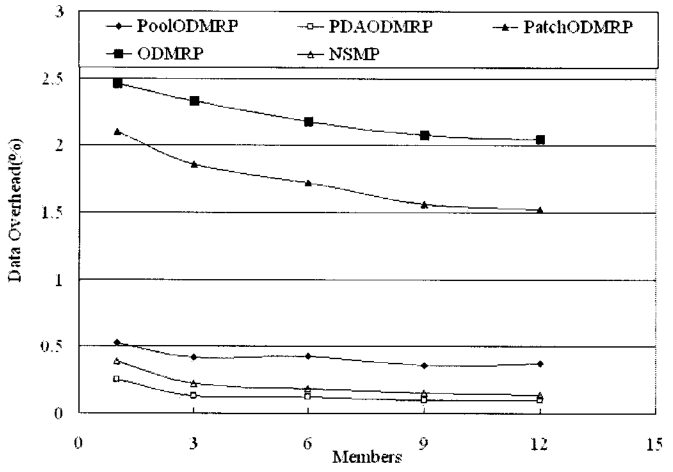


Fig. 18. Data overhead as a function of members.

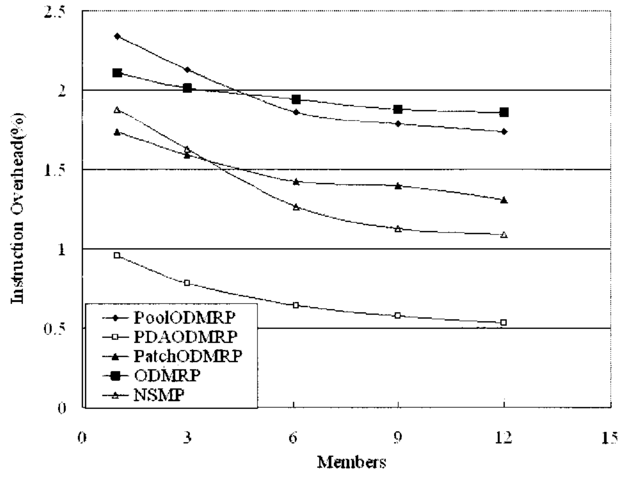


Fig. 19. Instruction overhead as a function of members.

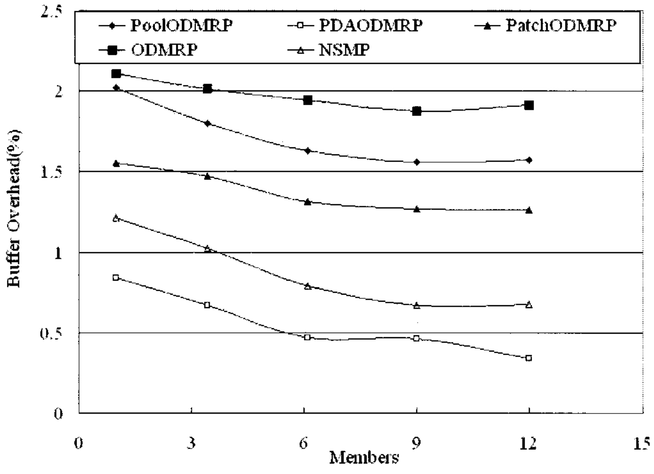


Fig. 20. Buffer overhead as a function of members.

head of all these protocols decrease because of the inherent broadcasting characteristics of wireless communication.

Figs. 21 and 22 describe the impact of the members of groups on the average data delivery ratio and data delivery delay of these protocols, respectively. PDAODMRP has lowest data delivery delay because of its lowest control overhead. The impact on the average data delivery ratio also can be divided into two cases for discussion.

- When there are few members in groups, the data delivery ratio of these protocols is mainly determined by the robustness of forwarding mesh. ODMRP, PatchODMRP, and PDAODMRP have similar data delivery ratio because the robustness of their forwarding mesh is similar.
- When there are many members in groups, the data delivery ratio of these protocols is greatly affected by the wireless bandwidth acquired by a data packet for its transmission. PDAODMRPP has the highest data delivery ratio among these protocols because of its most efficient forwarding mesh and shortest forwarding node lifetime.

In Figs. 21 and 22, when the members of groups increase, the data delivery ratio of all these protocols decreases and the data delivery delay of these protocols increases.

PDAODMRP has the most efficient forwarding mesh and the shortest forwarding node lifetime. Therefore, it scales best when member increases. Compared with ODMRP and PatchODMRP, PoolODMRP has shorter forwarding node lifetime, and NSMP has more efficient forwarding mesh. Hence, they scale better than ODMRP and PatchODMRP do when the members of groups increase.

From the simulation results, a conclusion can be drawn that PDAODMRP scales best when the members of groups increase because of its most efficient forwarding mesh and its shortest forwarding node lifetime.

VI. CONCLUSION

PDAODMRP is proposed as a new ad hoc network multicast protocol to extend PoolODMRP. Compared with PoolODMRP, PDAODMRP has overcome the shortcomings of PoolODMRP by its new technologies, and PDAODMRP has the following contributions: (1) It reduces its control overhead by the passive acknowledgement of data packets; (2) it adopts a dynamic

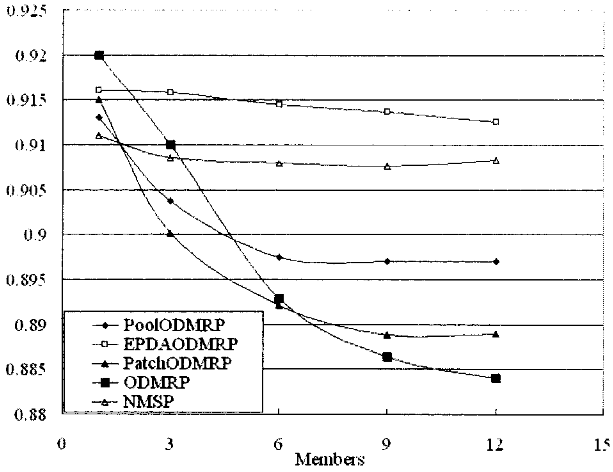


Fig. 21. Data delivery ratio as a function of members.

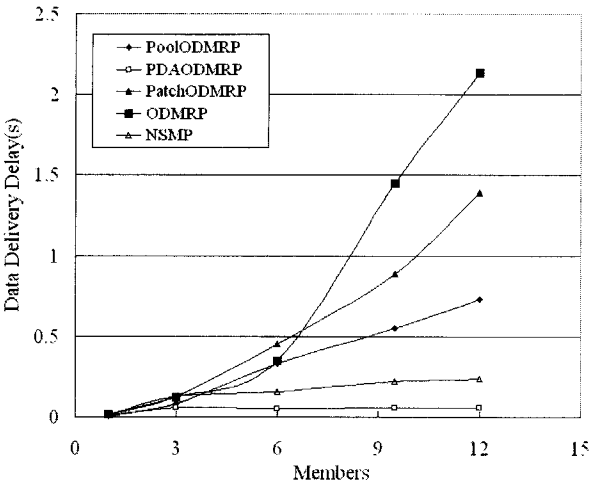


Fig. 22. Data delivery delay as a function of members.

local route maintenance to enforce its forwarding mesh; (3) it max simplifies the route information collected from data packets to reduce its instruction overhead and buffer overhead; (4) it adopts the route evaluation policy of NSMP to guarantee both the robustness and efficiency of its forwarding mesh. Therefore, among all these protocols, PDAODMRP scales best when node speed, sources and members increase.

REFERENCES

[1] Mobile Ad Hoc Network (MANET) Working Group, available at http://www.ietf.org/html.charters/manet-charter.html.

[2] B. M. Leiner, D. L. Neison, and F. A. Tobagi, "Issues in packet radio network design," *Proc. IEEE*, vol. 75, pp. 6–20, Jan. 1987.

[3] J. Jubin and J. D. Tornow, "The DAPAR packet radio network protocols," *Proc. IEEE*, vol. 75, pp. 21–32, Jan. 1987.

[4] S. Paul, *Multicasting on the Internet and Its Application*, Kluwer Academic Publishers, 1998.

[5] T. Clausen *et al.*, "Optimized link state routing protocol (OLSR)," *Internet Draft*, IETF, Oct. 2001.

[6] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. SIGCOMM'94*, vol. 24, Oct. 1994, pp. 234–244.

[7] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *ACM/Balzer Mobile Networks and Applications*, vol. 1, pp. 183–197, 1996.

[8] V. Park and S. Corson, "Temporally-ordered routing algorithm (TORA)," *Internet Draft*, IETF, Aug. 1998.

[9] J. Broch, D. B. Johnson, and D. A. Maltz, *The Dynamic Source Routing in Ad Hoc Wireless Networks*, Kluwer Academic Publishers, 1996.

[10] C. Perkins, E. M. Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," *Internet Draft*, IETF, June 1999.

[11] R. Dube, C. D. Rais, and K. Y. Wang, "Signal stability-based adaptive routing (SSA) for ad hoc wireless network," *IEEE Pers. Commun.*, vol. 4, pp. 36–47, 1997.

[12] J. Broch *et al.*, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. ACM/IEEE MobiCom'98*, Oct. 1998, pp. 25–30.

[13] P. Johansson *et al.*, "Scenario-based performance analysis of routing protocols for mobile ad hoc networks," in *Proc. IEEE MobiCom'99*, pp. 195–206.

[14] S. Corson and J. Macker, "Mobile ad hoc networking (MANET) : Routing protocol performance issues and evaluation considerations," *RFC 2501*, IETF, Jan. 1999.

[15] C. Chiang, M. Gerla, and L. Zhang, "Shared tree wireless network multicast," in *Proc. IEEE ICCCN'97*, Sept. 1997, pp. 28–33.

[16] E. M. Royer and C. E. Perkins, "Multicast ad hoc on-demand distance vector (MAODV) routing," *Internet Draft*, IETF, 2000.

[17] E. Bommaiah *et al.*, "AMRoute : Ad hoc multicast routing protocol," *Internet Draft*, IETF, Aug. 1998.

[18] C. W. Wu, Y. C. Tay, and C. K. Toh, "Ad hoc multicast routing protocol utilizing increasing id-numbers (AMRIS) functional specification," *Internet Draft*, IETF, Nov. 1998.

[19] S. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol (ODMRP) for ad hoc networks," *Internet Draft*, IETF, July 2000.

[20] S. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol," in *Proc. IEEE WCNC'99*, Sept. 1999, pp. 1298–1302.

[21] S. Lee, W. Su, and M. Gerla, "Ad hoc wireless multicast with mobility prediction," in *Proc. IEEE ICCCN'99*, Oct. 1999, pp. 4–9.

[22] J. J. Garcia-Luna-Aceves and E. L. Mdrguga, "The core-assisted mesh protocol," *IEEE J. Select. Areas Commun.*, vol.17, pp. 1380–1394, 1999.

[23] S. Lee *et al.*, "A performance comparison study of ad hoc wireless multicast protocols," in *Proc. INFOCOM 2000*, 2000, pp. 751–756.

[24] S. Ni *et al.*, "The broadcast storm problem in a mobile ad hoc network," in *Proc. ACM MobiCom'99*, Aug. 1999, pp. 151–162.

[25] G. Aggelou and R. Tafazolli, "RDMAR : A bandwidth-efficient routing protocol for mobile ad hoc networks," in *Proc. ACM IWWMM'99*, Aug. 1999, pp. 26–33.

[26] S. Lee and C. Kim, "A new wireless ad hoc multicast routing protocol," *Computer Networks*, vol. 38, pp. 121–135, 2002.

[27] M. Lee and Y. K. Kim, "PatchODMRP : An ad hoc multicast routing protocol," in *Proc. ICIN 2001*, 2001, pp. 537–543.

[28] S. Cai and X. Yang, "The performance of PoolODMRP," in *Proc. IEEE /IFIP MMNS 2003*, Sept. 2003, pp. 90–101.

[29] S. Cai, X. Yang, and W. Yao, "The comparison between PoolODMRP and PatchODMRP," in *Proc. IEEE ICON 2003*, Sept. 2003, pp. 729–735.

[30] IEEE Computer Society LAN MAN Standards Committee, Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, New York, 1997.

[31] S. Cai, L. Wang, and X. Yang. "An ad hoc multicast protocol based on passive data acknowledgement," *Computer Commun.*, vol. 27, pp. 1812–1824, 2004.

[32] Wireless Adaptive Mobility Lab. DEPT of Comp. SCI, UCLA, "GloMoSim: A scalable simulation environment for wireless and wired network system," available at http://pcl.cs.ucla.edu/projects/domains/glomosim.html.

Shaobin Cai was born in 1973. He received the B.S. and M.S. degree in computer science from Harbin Institute of Technology in 1996 and 1998. Now, he is a Ph.D. student of Department of Computer Science and Technology, Harbin Institute of Technology. His primary interests are ad hoc network protocols and distributed computing.

Xiaozong Yang was born in 1939. He is a professor in department of Computer Science and Technology of Harbin Institute of Technology. His current research interests are computing architecture, fault tolerant computing, fault injection, wireless network, and dependable computing.

Ling Wang received received her Ph.D. degree in electrical engineering from University of Nevada, Las Vegas, USA, in 2003. In 2004, she joined the faculty of the department of Computer Science and Technology in Harbin Institute of Technology as an assistant professor. Her primary interests are in VLSI design and various aspects of computer-aided design including wireless network, hardware-software co-design, high-level synthesis, and low-power system design.