モバイルコンピューティングと
ワイヤレス通信
（２００３．３．６）
24－1

Multiple-Route Adhoc Ondemand (MRAODV) Routing Protocol

Kenji Hasebe, Shingo Umeshima and Hiroaki Higaki
Department of Computers and Systems Engineering
Tokyo Denki University
E-mail: {namu, shin5, hig}@higlab.k.dendai.ac.jp

Several routing protocols, e.g. DSR(Dynamic Source Routing Protocol) and AODV(Ad-hoc On-Demand Distance Vector Routing Protocol), have been proposed for routing data packets in ad-hoc networks. These are on-demand routing protocols. Only when a node requires to transmit data packets, it searches a route to a destination node. In addition, these are single-path protocols which detect only one route. However, in an ad-hoc network, due to mobility of nodes and instability of communication links, multiple-route protocols are required. In this paper, we propose a novel multiple-route protocol MRAODV(Multiple-Route On-Demand Distance Vector Routing Protocol) where separated reverse path fragments are connected to achieve additional routes. According to simulation results, more additional routes are detected in our protocol than MNH protocol.

複数経路を用いた安定なメッセージ配送のためのアドホックルーティングプロトコル

東京電機大学 理工学部 情報システム工学科
長谷部 顕司　梅島 慎吾　桧垣 博章
E-mail: {namu, shin5, hig}@higlab.k.dendai.ac.jp

DSR や AODV といったアドホックルーティングプロトコルでは、送信元移動コンピュータから送信先移動コンピュータまでの単一の経路が検出される。しかし、この経路上にあるコンピュータの移動や電源断によって経路が無効化すると、再度、フラッディングを用いた経路探索が必要となる。各経路探索において複数の経路を検出する方法が提案されているが、検出経路数が少なく再経路探索までの間隔は十分長くない。本論文では、AODV を拡張し、経路要求メッセージのフラッディングで得られた枝経路を接続し、適切に方向を変更することでより多数の経路を検出できる MR-AODV を提案する。シミュレーションの実験の結果、経路探索の間隔は、AODV の 10.0、MNH の 1.67 倍に拡大されていることが明らかとなった。

1 Introduction

Recently, advanced computer and network technologies have led to the development of computer networks. Especially, mobile computers such as notebook PCs, handheld PCs and Personal Data Assistants (PDAs) have become widely available. With the help of wireless LAN protocols such as IEEE802.11 [1], mobile computers exchange messages with each other in order to communicate and cooperate for achieving network applications.

In most of computer networks including mobile computers, only routers connected to wired networks forward messages to transmit them from a source computer to a destination one. An *access point* AP is a gateway between a mobile computer M within a transmission range of AP and a wired network. However, it is difficult to apply methodologies of design and management developed for infrastructured networks to computer networks supporting fieldworks, disaster rescue, control of autonomous robots, sensor networks and so on. Hence, *ad-hoc networks* are required to be introduced. Here, there is no fixed infrastructure, i.e. the network is composed of only mobile computers and there is neither access point nor gateway. Each mobile computer has a function for routing messages. That is, on receipt of a message m, a mobile computer M_i finds a next hop mobile computer $next_{\langle M_i, R_{S \to D}\rangle}$ to reach m to a destination one D. In addition, since a mobile computer changes a location, even if a mobile computer M_i is in a transmission range of another mobile computer M_j and vice versa at a time, it is not always possible for M_i and M_j to exchange messages directly with each other.

In order to solve this problem, on-demand routing protocols have been designed and developed. Here, each mobile computer does not hold a routing table for all reachable mobile computers. Hence, no communication is required if no mobile computer exchanges an application message. Only when a mobile computer S requires to transmit an application message to another mobile computer D, S detects a message transmission route $R_{S \to D}$ by exchanging control messages among mobile computers. In addition, only routing information required for currently active communication is maintained. In some routing protocols called *source routing protocols* such as DSR [2] and LBSR [9], information of $R_{S \to D}$ is held only by S. $R_{S \to D}$ is kept in a header of a message and each mobile computer transmits the message according to $R_{S \to D}$ in the header. In the other protocols called *table-based routing protocols* such as AODV [8], information for $R_{S \to D}$ is held by mobile computers on $R_{S \to D}$. Each mobile computer M_i on $R_{S \to D}$ keeps a next hop in a routing table and forwards a message according to the routing table.

2 Related Works

Most of the conventional ad-hoc routing protocols depends on *flooding* of a control message which is the same as Message Diffusion Protocol in a wired network. Wireless communication media for wireless LAN have broadcast feature. Here, a message transmitted by a mobile computer M_i is received by mobile computers in a transmission range of M_i. A certain mobile computer S broadcasts a message m within a transmission range of S. If a mobile computer M_i receiving m from S or another mobile computer broadcasts m to all mobile computers within a transmission range of M_i, all reachable mobile computers receive m by multi-hop transmission. This method is called flooding.

In AODV [8], a route request message *RReq* is transmitted by flooding. On receipt of the first *RReq* message, each mobile computer M_i records a direct sender M_i^u of *RReq* as an *upstream mobile computer*. Here, it represents that M_i sets a *backward link* to M_i^u. Then, M_i broadcasts the *RReq*. When a destination mobile computer D receives the *RReq*, a set of backward link specifies a backward route from D to S. A route reply message *RRep* is transmitted along the backward route by unicast transmissions. On receipt of the *RRep*, each mobile computer M_i records the direct sender of the *RRep* into a routing table entry as a next hop for transmitting messages to D. Here, we call the sender as a *downstream mobile computer* of M_i. At the same time, a *forward link* is set from M_i to the downstream mobile computer. Then, M_i sends the *RRep* to the upstream mobile computer by unicast. Finally, a source mobile computer S receives the *RReq* and records the direct sender as a next hop and a downstream mobile computer. Now, a message transmission route $R_{S \to D}$ from S to D is detected, i.e. routing tables of S and all intermediate mobile computers on the route keeps next hop routers in the routing tables. A set of forward links represents a *forward route* from S to D. A mobile computer which receives an *RReq* message and does not receive a corresponding *RRep* message discards a record of an upstream mobile computer by using timeout.

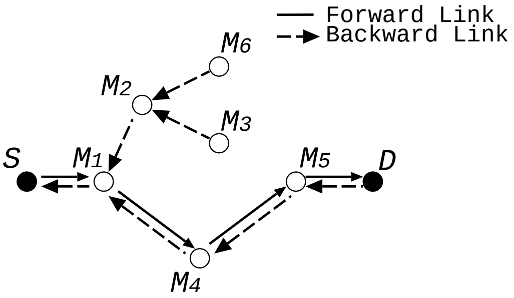

Figure 1: AODV protocol.

In an ad-hoc network, due to movement of mobile computers, limited battery capacity and noisy wireless communication environment, a communication link between mobile computers might be broken and $R_{S \to D}$ becomes unavailable. For achieving a stable communication in such an environment, two approaches have been proposed; *partial re-establishment* and *multiple route establishment*. In the partial re-establishment, if a mobile computer on an active message transmission route detects a broken link on the route, it fixes the route locally, i.e. detects a partial route to connect the divided two sub-routes, from a source mobile computer to an upstream mobile computer of the broken link and from a downstream mobile computer of the broken link to a destination mobile computer. AODV-BR [4] is this kind of protocol. On the other hand, in multiple route establishment, when a source mobile computer searched a message transmission route to a destination one, it detects multiple routes and keeps information of the routes. If a broken link on an active route is detected, the source mobile computer select one of the keeping route and make it active, i.e. messages are transmitted through the selected route. MultipathDSR [6], SMR [5], MNH [3] are multiple route establishment protocols.

MNH [3] is an extension of AODV to establish multiple routes. In AODV, each mobile computer records only a sender mobile computer of the first *RReq* message as an upstream mobile computer. Hence, only a single route is detected. In MNH, each mobile computer M_i records all the senders of *RReq* messages as upstream mobile computers except when it receives an *RReq* message from M_i. Hence, a mobile computer might receive multiple *RRep* messages and multiple routes might be detected.

3 MRAODV Protocol

Though MNH detects multiple message transmission routes, the number of detected routes is not so many. For example in Figure 2, two transmission routes $S \rightarrow M_1 \rightarrow M_4 \rightarrow M_5 \rightarrow D$ (a route detected by AODV) and $S \rightarrow M_1 \rightarrow M_3 \rightarrow M_5 \rightarrow D$ are possible. However, MNH detects only the former route in case that backward links are set as shown in Figure 2.

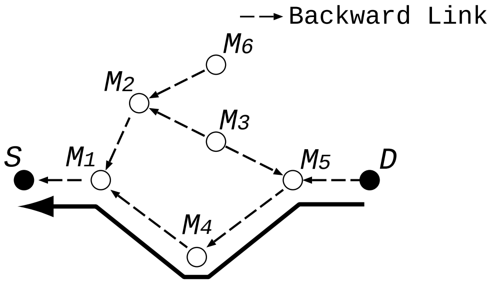

Figure 2: Less detected route in MNH.

By switching direction of backward link, e.g. as shown in Figure 3, a backward link $M_3 \rightarrow M_5$ is switched as $M_5 \rightarrow M_3$, this problem is solved and both routes are detected. By deciding whether direction of a backward link is switched or not, our proposed protocol *MRAODV* (Multiple Route AODV) assigns numbers to each mobile computer.

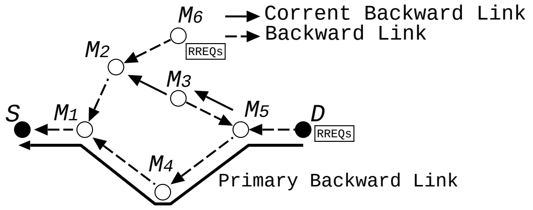

Figure 3: Connecting and reversing backward links.

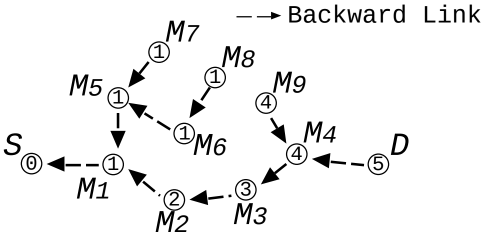

Figure 4: Assignment of branch numbers.

Let S and D be source and destination mobile computers, respectively.

[**Transmission of an** ***RREQ*** **message**]

A route request control message (an *RREQ* message) m_{req} for detecting a message transmission route $R_{S \rightarrow D}$ contains the following information fields:

- Source Address $m_{req}.src$: An address of the source mobile computer S.
- Destination Address $m_{req}.dst$: An address of the destination mobile computer D.
- Upstream Address $m_{req}.up$: An address of a mobile computer from which a sender $M^{s}_{m_{req}}$ of m_{req} received the first *RREQ* message for $R_{S \rightarrow D}$.
- Hop Count $m_{req}.hop$: A number of mobile computers along a route through which the *RREQ* message for $R_{S \rightarrow D}$ has been forwarded from S to $M^{s}_{m_{req}}$.

(Protocol in a source mobile computer S of an *RREQ* message)

1. On receipt of a request for a transmission of an application message to a destination mobile computer D, S searches a routing table for an entry of a message transmission route $R_{S \rightarrow D}$.
2. If S finds an entry of $R_{S \rightarrow D}$, S transmits the application message along $R_{S \rightarrow D}$. (end)
3. S broadcasts an *RREQ* message m_{req} to all mobile computers within a transmission range of S. Here, $m_{req}.src := S$, $m_{req}.dst := D$, $m_{req}.up := S$, $m_{req}.hop := 0$.
4. S stores a branch identifier in a routing table as $brid_{\langle S, R_{S \rightarrow D} \rangle} := 0$.
5. On receipt of an *RREQ* message m'_{req} where $m'_{req}.src = S$, S stores an entry $\langle M^{s}_{m_{req}}, down, \bot \rangle$ into a neighbor table.

(Protocol in an intermediate mobile computer M_i for an *RREQ* message transmission: Flooding)

1. On first receipt of an *RREQ* message m_{req} where $m_{req}.dst \neq M_i$ from a mobile computer $M^{s}_{m_{req}}$ within a transmission range of M_i,

1-1. M_i sets a backward link to $M^{s}_{m_{req}}$ for a message transmission route $R_{S \rightarrow D}$. Here, M_i stores an address of $M^{s}_{m_{req}}$ into a neighbor table as an upstream mobile computer for $R_{S \rightarrow D}$. That is, an entry $\langle M^{s}_{m_{req}}, up, \bot \rangle$ is stored into a neighbor table.

1-2. M_i broadcasts an *RREQ* message m'_{req} to all mobile computers within a transmission range of M_i. Here, $m'_{req}.src := m_{req}.src (= S)$, $m'_{req}.dst := m_{req}.dst (= D)$, $m'_{req}.up := M^{s}_{m_{req}}$ and $m'_{req}.hop := m_{req}.hop + 1$.

2. On receipt of an *RREQ* message m_{req} from $M^{s}_{m_{req}}$ and if M_i has already received another *RREQ* message m'_{req} where $m_{req}.src = m'_{req}.src (= S)$ and $m_{req}.dst = m'_{req}.dst (= D)$,
   - if $m_{req}.up = M_i$, M_i confirms a link from M_i to $M^{s}_{m_{req}}$ as part of $R_{S \rightarrow D}$. Here, M_i stores an address of $M^{s}_{m_{req}}$ into a neighbor table as a down-

stream mobile computer for $R_{S \to D}$. That is, an entry $\langle M^s_{m_{req}}, down, \perp \rangle$ is stored into a neighbor table.

- otherwise, M_i stores an address of $M^s_{m_{req}}$ into a neighbor table as a mobile computer of another branch. That is, an entry $\langle M^s_{m_{req}}, neighbor, \perp \rangle$ is stored into a neighbor table.

(Protocol in a destination mobile computer D for an *RREQ* message)

1\. On first receipt of an *RREQ* message m_{req} from a mobile computer M^s_{req} where $m_{req}.dst = D$,

1-1. D sets a backward link to M^s_{req} for a message transmission route $R_{S \to D}$. Here, D stores an address of M^s_{req} into a neighbor table as an upstream mobile computer for $R_{S \to D}$. That is, an entry $\langle M^s_{m_{req}}, up, \perp \rangle$ is stored into a neighbor table.

1-2. A branch identifier of D is set as $brid_{\langle D, R_{S \to D} \rangle} := m_{req}.hop + 1$.

1-3. D broadcasts an *RREP* message m_{rep} within a transmission range of D where $m_{rep}.src := m_{req}.src (= S)$, $m_{rep}.dst := m_{req}.dst (= D)$, $m_{rep}.up := M^s_{m_{req}}$ and $m_{rep}.hop := m_{req}.hop + 1$.

2\. On receipt of an *RREQ* message m_{req} from $M^s_{m_{req}}$ and if D has already received another *RREQ* message m'_{req} where $m_{req}.src = m'_{req}.src (= S)$ and $m_{req}.dst = m'_{req}.dst (= D)$, D stores an address of $M^s_{m_{req}}$ into a neighbor table as a mobile computer of a branch of $R_{S \to D}$. That is, an entry $\langle M^s_{m_{req}}, neighbor, \perp \rangle$ is stored into a neighbor table.

[Transmission of an *RREP* message]

A route reply control message (*RREP* message) m_{rep} for confirming a primary message transmission route $R_{S \Rightarrow D}$ and assigning a branch identifier to each intermediate mobile computer contains the following information fields:

- Source Address $m_{rep}.src$: An address of the source mobile computer S.
- Destination Address $m_{rep}.dst$: An address of the destination mobile computer D.
- Upstream Address $m_{rep}.up$: An address of a mobile computer from which a sender mobile computer $M^s_{m_{rep}}$ of m_{rep} received the first *RREQ* message.
- Hop Count $m_{rep}.hop$: A number of mobile computers along a route from S to $M^s_{m_{rep}}$ through which the first *RREQ* message for $R_{S \to D}$ has been forwarded. That is, a number of mobile computers from S to $M^s_{m_{rep}}$ along a primary message transmission route $R_{S \Rightarrow D}$.

(Protocol in a destination mobile computer D for an *RREP* message)

1\. On receipt of the first *RREQ* message m_{req} for detecting a message transmission route $R_{S \to D}$ from a mobile computer $M^s_{m_{req}}$, D broadcasts an *RREP* message m_{rep} within a transmission range of D where $m_{rep}.src := m_{req}.src (= S)$, $m_{rep}.dst := m_{req}.dst (= D)$, $m_{rep}.up := M^s_{m_{req}}$ and $m_{rep}.hop := m_{req}.hop + 1$.

2\. On receipt of an *RREP* message m_{rep}, D discards m_{rep}.

(Protocol in an intermediate mobile computer M_i for an *RREP* message transmission)

1\. On receipt of an *RREP* message m_{rep} from a mobile computer $M^s_{m_{rep}}$, if M_i has not yet received a corresponding *RREQ* message, M_i sets a timer and suspends.

- If the timer expires without receiving the *RREQ* message, M_i discards m_{rep}. (end)
- Otherwise, i.e. if M_i receives the *RREQ* message, M_i processes the *RREQ* message and resumes.

2\. If $M^s_{m_{rep}} = D$, M_i stores an address of D into a neighbor table as a neighbor mobile computer. That is, an entry $\langle D, neighbor, \perp \rangle$ is stored into a neighbor table.

3\. M_i refers an entry for $M^s_{m_{rep}}$ in a neighbor table.

- If $M^s_{m_{rep}}$ is a downstream mobile computer of M_i,

3-1. If M_i has already received another *RREP* message from an upstream mobile computer of M_i, M_i discards m_{rep}. (end)

3-2. M_i updates an entry in a neighbor table for $M^s_{m_{rep}}$ as $\langle M^s_{m_{rep}}, down, m_{rep}.hop \rangle$.

3-3. M_i confirms a link from M_i to $M^s_{m_{rep}}$ as part of a primary message transmission route $R_{S \Rightarrow D}$. Here, an address of $M^s_{m_{rep}}$ and a hop count $m_{rep}.hop - 1$ are stored into an entry of a routing table for $R_{S \Rightarrow D}$ as a next hop $next_{\langle i, S \Rightarrow R \rangle} := M^s_{m_{rep}}$ and a branch identifier $brid_{\langle M_i, S \Rightarrow R \rangle} := m_{rep}.hop - 1$, respectively.

3-4. M_i broadcasts an *RREP* message m'_{rep} to all mobile computers within a transmission range of M_i. Here, $m'_{rep}.src := m_{rep}.src (= S)$, $m'_{rep}.dst := m_{rep}.dst (= D)$, $m'_{rep}.up$ is an address of an upstream mobile computer which has been stored in a neighbor table and $m'_{rep}.hop := m_{rep}.hop - 1$.

- If $M^s_{m_{rep}}$ is an upstream mobile computer of M_i,

3-1. If M_i has already received another *RREP* message from a downstream mobile computer of M_i, M_i discards m_{rep}. (end)

3-2. M_i updates an entry in a neighbor table for $M^s_{m_{rep}}$ as $\langle M^s_{m_{rep}}, up, m_{rep}.hop \rangle$.

3-3. M_i confirms a link from $M^s_{m_{rep}}$ to M_i as part of a branch transmission route connecting to a primary transmission route $R_{S \Rightarrow D}$ at a mobile computer $M_k \in R_{S \Rightarrow D}$ with a branch identifier $brid_{\langle M_k, S \to D \rangle} = m_{rep}.hop$.

3-4. M_i broadcasts an *RREP* message m'_{rep} to all mobile computers within a transmission range of M_i. Here, $m'_{rep}.src := m_{rep}.src (= S)$, $m'_{rep}.dst := m_{rep}.dst (= D)$, $m'_{rep}.up := M^s_{m_{rep}}$ and

$m'_{rep}.hop := m_{rep}.hop$.

- Otherwise, i.e. if an entry $\langle M^s_{m_{rep}}, neighbor, \bot\rangle$ is in a neighbor table of M_i, M_i updates this entry as $\langle M^s_{m_{rep}}, neighbor, m_{rep}.hop\rangle$ and discards m_{rep}.

(Protocol in a source mobile computer S for an $RREP$ message transmission)

1. On receipt of the first $RREP$ message m_{rep} from a mobile computer $M^s_{m_{rep}}$,

1-1. S stores a next hop $next_{\langle S, S\to R\rangle} := M^s_{m_{rep}}$ into an entry for $R_{S\to D}$ in a routing table.

1-2. S broadcasts an $RREP$ message m'_{rep} to all mobile computers within a transmission range of S where $m'_{rep}.src := m_{rep}.src(= S)$, $m'_{rep}.dst := m_{rep}.dst(= D)$, $m'_{rep}.up := S$ and $m'_{rep}.hop := 0$.

2. On receipt of an $RREP$ message m_{rep} from $M^s_{m_{rep}}$ and if S has already received another $RREP$ message m'_{rep} where $m_{rep}.src = m'_{rep}.src(= S)$ and $m_{rep}.dst = m'_{rep}.dst(= D)$, S discards m_{rep}.

[Branch Identifier Comparison]

(Preprocessing)

1. If there is only one entry for an upstream mobile computer in a neighbor table in M_i, M_i clears the neighbor table, i.e. M_i removes the entry from the neighbor table, and broadcasts an $RDEL$ message to all mobile computers within a transmission range of M_i.
2. On receipt of an $RDEL$ message m_{del} from a mobile computer $M^s_{m_{del}}$,
   - if $M^s_{m_{del}}$ is a downstream mobile computer of M_i, M_i removes an entry for $M^s_{m_{del}}$ from a neighbor table.
   - otherwise, M_i discards m_{del}.

(Combination of two branch transmission routes)

- If a neighbor table in a mobile computer M_i includes an entry for an upstream mobile computer and neighbor ones, i.e. includes no downstream one,
  1. For each entry of a neighbor mobile computer M_k in a neighbor table in M_i,
     - if $brid_{\langle M_i, S\to D\rangle} < brid_{\langle M_k, S\to D\rangle}$, M_i updates the entry as $\langle M_k, down, brid_{\langle M_k, S\to D\rangle}\rangle$.
     - if $brid_{\langle M_i, S\to D\rangle} > brid_{\langle M_k, S\to D\rangle}$, M_i updates the entry as $\langle M_k, up, brid_{\langle M_k, S\to D\rangle}\rangle$.
  2. If number of entries of a neighbor mobile computer which becomes a downstream one of M_i is larger than number of entries of a neighbor mobile computer which becomes an upstream one of M_i, M_i forwards a $FORWARD$ message.

4 Evaluation

In this section, we evaluate performance of the proposed MRAODV protocol comparing with MNH and AODV-BR by simulation. Simulation parameters are as follows:

- Simulation area: 500m×500m
- Diameter of signal transmission range: 100m (constant)
- Number of mobile computers: 20, 40, 60, 80, 100
- Distribution of mobile computers: Uniquely Distributed
- Mobility of a mobile computer: 0–5km/h (uniquely distributed)
- Number of simulations: 100

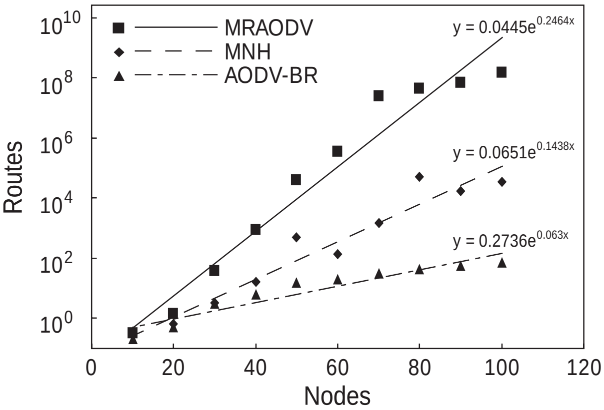

Figure 5: Number of detected routes.

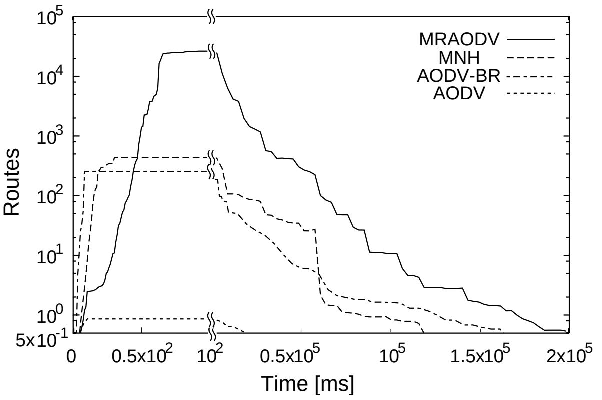

Figure 6: Number of available route.

Figure 5 shows numbers of message transmission routes detected by the three protocols. In AODV-BR, since sub-route search is not invoked before detecting a broken link, a number of possible sub-route which is 1 hop different from the primary route. As being increased density of mobile computers, the numbers of detected message transmission route are increased. However, MRAODV detects the most routes in the three protocols, since the direction of a link between two neighbor mobile computer is fixed at the receipt of *RReq* message in MNH and only 1 hop different sub-routes are searched in AODV-BR.

Figure 6 shows numbers of available transmission routes in AODV, MRAODV and MNH. This is a sim-

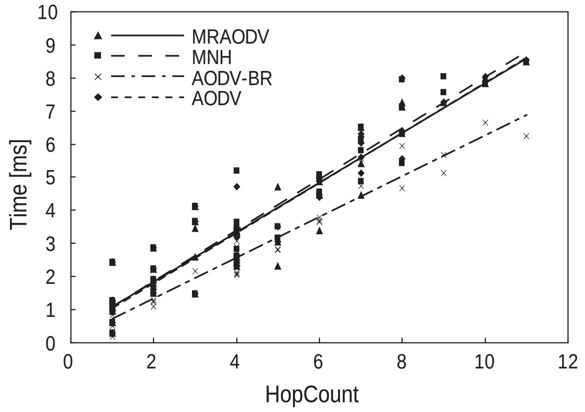

Figure 7: Required time for detecting the first route.

ulation result where a number of mobile computers is 60. First, a route detection protocol is executed and the number of available routes is increased. After terminating the protocol, due to the movement of mobile computers, the number of available routes is decreased. If there is no available route and the source mobile computer has more packets to transmit to the destination one, the route detection protocol should be invoked again. In the simulation result, the time duration between invocation of the protocol and the time when there is no available route in MRAODV is 200sec and the longest than the other two protocols (120sec in MNH and 20sec in AODV). Since time interval between two successive invocations of a route detection protocol in MRAODV is longer than the others, MRAODV reduces the overhead for route detection. In MRAODV, in order to determine a branch route ID, synchronization is required to receive *RRep* messages. Hence, from the beginning to 400ms, MNH detects more transmission routes than MRAODV. By relaxing the synchronization condition, it is possible to reduce the required time duration to detect all the possible route in MRAODV.

In order to start a transmission of an application message, only one route is needed. That is, on detecting the first route to a destination mobile computer, a source mobile computer starts to transmit an application message even if the route detection protocol is still under process. Figure 8 shows the required time to detect the first route in the three protocols. The hop count presents the number of routers between the source and the destination. For detecting the first route, an *RReq* message and a corresponding *RReq* message are transmitted through a primary route in all the three protocols. Hence, the difference of required time in the protocols is not depend on protocol overhead but depend on required processing time. The simulation result shows there is no meaningful difference among the processes.

5 Concluding Remarks

This paper has proposed a multi-route ad-hoc routing protocol MRAODV which is an extension of AODV. Here, multiple branch backword route connected to a primary backward route detected by AODV are connected to detect sub-routes. Here, direction of branch backword route is dynamically changed to detect as many routes as possible. According to the simulation results, MRAODV detects more routes than MNH and the time duration of successive procedures of route detection is extended. Hence, MRAODV reduces routing overhead.

In future work, the required time to detect sub-routes is also reduced by relaxing synchronization condition in the protocol.

References

[1] "Wireless LAN Medium Access control(MAC) and Physical Layer(PHY) Specifications," Standard IEEE 802.11 (1997).

[2] David, B., David, A., Hu, Y.C., Jorjeta, G. and Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," Internet Draft, draft-ietf-manet-dsr-04.txt (2000).

[3] Jiang, M.H. and Jan, R.H., "An Efficient Multiple Paths Routing Protocol for Ad-hoc Networks," Proc. of the 15th International Conference on Information Networking, pp. 544–549 (2001).

[4] Lee, S.J. and Gerla, M., "AODV-BR: Backup Routing in Ad hoc Networks," Proc. of IEEE Wireless Communications and Networking Conference, pp. 1311–1316 (2000).

[5] Lee, S.J. and Gerla, M., "Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks," Proc. of IEEE International Conference on Communications, pp. 3201–3205 (2001).

[6] Nasipuri, A. and Das, D.S., "On-Demand Multipath Routing for Mobile Ad Hoc Networks," Proc. of IEEE 8th International Conference on Computer Communications and Networks, pp. 64–70 (1999).

[7] Perkins, C.E. and Bhagwat, P., "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computerss," ACM SIGCOMM' 94, pp. 234–244 (1994).

[8] Perkins, C.E. and Royer, E.M., "Ad-hoc On-Demand Distance Vector Routing," Proc. of IEEE 2nd Workshop on Mobile Computing Systems and Applications, pp. 90–100 (1999).

[9] Sagawa, Y. and Higaki, H., "Loop-based Source Routing Protocol for Mobile Adhoc Networks," Proceedings of the 9th IPSJ Workshop on Multimedia Communication and Distributed Processing, pp. 157–162 (2001).