Loop-Based Source Routing Protocol for Mobile Ad-hoc Networks

Hiroyuki Unoki Tomonori Asano Yusuke Kurosawa Hiroaki Higaki
Department of Computers and Systems Engineering
Tokyo Denki University
E-mail: {unoki,tom,kuro,hig}@higlab.k.dendai.ac.jp

In an ad-hoc network, a routing protocol which detects a transmission route from a source mobile computer to a destination one is critical due to mobility and limited battery capacity of mobile computers. Here, a communication link between two mobile computers is not always bi-directional, i.e. uni-directional, since transmission power of mobile computers is not the same. Though some ad-hoc routing protocols, e.g. DSR, support routing with uni-directional links, multiple floodings (successive broadcasting) are used and communication overhead is high. This paper proposes a novel routing protocol LBSR supporting uni-directional links and based on combination of a single flooding and multiple unicast message transmissions. In LBSR, route cache mechanism works better than DSR in an environment with uni-directional links. Simulation results show the efficiency of route cache in LBSR.

閉路探索に基づくオンデマンドアドホックルーティングとその評価

東京電機大学 理工学部 情報システム工学科
卯木 宏幸 浅野 知倫 黒沢 祐介 桧垣 博章
E-mail: {unoki,tom,kuro,hig}@higlab.k.dendai.ac.jp

MANET(Mobile Ad-hoc Network) 環境では、移動コンピュータのバッテリ残量が均一ではないために、各移動コンピュータからの送信信号の出力が異なる。このため、すべての移動コンピュータ間の通信リンクが双方向であるとは限らず、片方向リンクを含むことが想定される。これまでに提案されているアドホックルーティングプロトコルの大部分は双方向リンクのみを用いるものである。しかし、片方向リンクを用いなければ経路を構築できない場合が考えられることから、移動コンピュータ間の経路検出の成功確率を高めるためには、片方向リンクをも用いるプロトコルの設計が望まれる。2組の独立なフラッディング (ブロードキャスト配送) を用いる DSR に対して、1組のフラッディングとユニキャスト配送の組合せで送信元と送信先を含む閉路を検出する LBSR を提案する。LBSR では、ブロードキャストメッセージ数が削減されている。また、経路キャッシュにより多くの情報が格納されるため、オーバヘッドは DSR よりも小さくすることが可能である。

1 Introduction

Recently, mobile computers not only handheld, laptop and parmtop personal computers (PCs), personal data assistants (PDAs) and personal information appliances (PIAs) but also computers in automobiles for intelligent transport systems (ITS) and computers for controlling autonomous mobile robots have become widely available. Since users of mobile computers request to access server computers for achieving internet services at any time and at any place, mobile computers are required to communicate with other computers through the Internet. Furthermore, for implementing a LAN (Local Area Network) to which mobile computers are connected by using wireless communication devices, wireless LAN protocols such as series of IEEE802.11 [2] and HIPERLAN [1] have been developed and standardized. According to network architectures, wireless LANs are classified into three categories; *infrastructured networks*, *multihop-access networks* and *ad-hoc networks*. In an infrastructured network, base stations are used as a gateway between a mobile computer and a wired network. A mobile computer m communicates with another computer c only when m is in a transmission range of a base station b and vice versa. A message exchanged between m and c is transmitted through b.

In a multihop-access network, if a mobile computer m is in a transmission range of a base station b, a message between m and another computer c is directly exchanged between m and b and transmitted through wired and/or wireless networks between b and c as in an infrastructured network. In addition, even if m is out of a transmission range of any base station, m exchanges a message with b if multi-hop message transmission by mobile computers between m and b is available. Here, a routing protocol for transmitting a message between m and b is required.

In the above two types of networks, a message from a mobile computer is always transmitted through a base station. However, for supporting temporary computer networks for disaster rescue, communication in conventions and conferences, a system consisting of a set of autonomous mobile robots controlled by micro computers, sensor networks and networks in a battle field, cost and overhead required for construction and maintenance of a wired network infrastructure and base stations are high. In addition, less flexibility is achieved due to a fixed infrastructure. In a mobile ad-hoc network (MANET), there is no base station and only mobile computers are connected to the network. Due to a bounded transmission range of a mobile computer m, m does not always exchange a message directly with another mobile computer m'. Thus, all (or most of) mobile computers are engaged in routing of a message and multi-hop transmission is required to exchange a message between m and m'. Here, a routing protocol for supporting mobility of computers is required. That is, a mobile computer is required to serve a role of router. Since not only m and m' but also intermediate mobile computers change locations, a routing protocol has to achieve a currently available route.

Until now, many kinds of ad-hoc routing protocols have been proposed such as DSDV [9], AODV [10] and TORA [8]. In these protocols, it is assumed that a message transmission range of mobile computers are the same and stable. That is, most of communication links are bi-directional and uni-directional links are omitted in these routing protocols. However, due to limited battery capacity, transmission power of mobile computers is not the same and changes. Thus, if only bi-directional links are used in an ad-hoc routing protocol, network connectivity gets lower. That is, some pairs of mobile computers cannot communicate even though there are message transmission routes with uni-directional communication links. Hence, it is required for multi-hop transmission to detect a route including uni-directional links for achieving higher network connectivity. That is, more pairs of mobile computers have message transmission routes between them. Though DSR (Dynamic Source Routing) [4] has this property, the protocol uses two independent floodings and communication overhead is high. CBRP [5] is another ad-hoc routing protocol supporting both bi-directional and uni-directional links. Here, a set of mobile computers configure a cluster in which one of them serves a roll of cluster head and the others communicate with the cluster head directly only through bi-directional links. Here, each cluster connects with another cluster through bi-directional or uni-directional links. Hence, a cluster contains only small number of mobile computers. Therefore, membership management for a cluster requires high communication overhead. ULSR [7] is an extension of CBRP. In CBRP, a cluster head has to communicate with another mobile computer in the same cluster directly. In ULSR, mobile computers which communicate one another by multi-hop message transmission only through bi-directional communication links form a cluster and each cluster connects with another cluster only through uni-directional communication links. Since more mobile computers are included in a cluster, communication overhead for membership management is reduced. However, both CBRP and ULSR assume only an environment with small number of uni-directional links.

This paper proposes a novel ad-hoc routing protocol LBSR (Loop Based Source Routing) in which looped routes including a source mobile computer are detected. One of the detected looped route contains a destination mobile computer if it is reachable from a source mobile computer. LBSR requires a single flooding and multiple unicast message transmissions and is designed for supporting an environment with many uni-directional communication links. In addition, route cache mechanism works more efficiently than DSR in an environment with uni-directional links.

2 Ad-hoc Routing Protocols

A mobile ad-hoc network $\mathcal{N} = \langle \mathcal{V}, \mathcal{L} \rangle$ is composed of a set $\mathcal{V} = \{M_1, \ldots, M_m\}$ of mobile computers and a set $\mathcal{L} \subseteq \mathcal{V}^2$ of communication links as shown in Figure 1. Message transmission from M_i to M_j through a communication link $\langle M_i, M_j \rangle$ is possible only if M_j is in a transmission range of M_i.

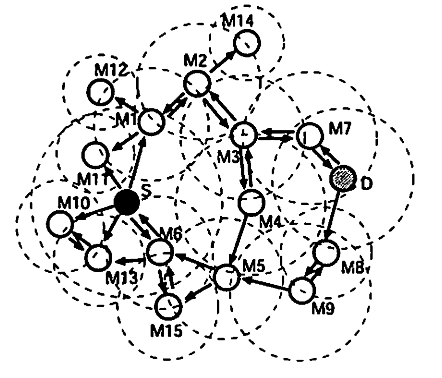

Figure 1: Ad-hoc network with uni-directional links.

Conventional ad-hoc routing protocols are classified into following two types; *topology management (proactive) routing protocols* and *on-demand (reactive) routing protocols*. By using the former, a routing table in each mobile computer is kept up to date to reflect any changes of network topology. Hence, control message transmissions are required even though no mobile computer communicates with another one. DSDV [9] is the most popular topology management protocol. On the other hand, by using the latter, a transmission route of a message from a mobile computer S to another one D is searched just before S transmits the message destined to D. DSR [4], AODV [10] and TORA [8] are on-demand routing protocols.

In addition, some routing protocols are based on an assumption that all available links are bi-directional, i.e. mobile computers M_i and M_j directly exchange messages only if $\langle M_i, M_j \rangle \in \mathcal{L}$ and $\langle M_j, M_i \rangle \in \mathcal{L}$ are satisfied. AODV is such a kind of protocol. Here, if a destination mobile computer D gets a route $R_{S \to D}$ from S to D, D also finds a reverse route $R_{D \to S}$ is available. Hence, by transmitting a message including $R_{S \to D}$ through $R_{D \to S}$, each mobile computer on $R_{S \to D}$ including S achieves $R_{S \to D}$ and a next hop mobile computer. Then, S starts transmission of application messages through $R_{S \to D}$. If DSR is applied in an environment with only bi-directional links, $R_{S \to D}$ is informed of S through $R_{D \to S}$ which is reverse of $R_{S \to D}$. Then, S starts transmission of application messages source routed with $R_{S \to D}$ by S. However, probability that $R_{S \to D}$ is detected is low since network connectivity is low due to existence of uni-directional communication links. On the other hand, the other protocols are based on an assumption that a transmission range of a mobile computer is not the same and changes. Here, even if a destination mobile computer D finds a transmission route $R_{S \to D}$, $R_{D \to S}$ is needed to be detected in order to transmit $R_{S \to D}$ to S. In an ad-hoc network shown in Figure 1, no transmission route from S to D is detected by the former protocol. However a route $\langle S, M_1, M_2, M_3, M_7, D\rangle$ is detected by using the latter protocol.

3 DSR protocol

In most of on-demand routing protocols, *flooding* [3] is used to detect a transmission route from a source mobile computer S to a destination mobile computer D. Flooding is based on a message diffusion protocol in a wired network [6]. Most of wireless communication media on which wireless LAN protocols depend is broadcast-based. A message broadcasted by a mobile computer M is received by all mobile computers within a transmission range of M. Suppose that a mobile computer S broadcasts a message *mes* to all mobile computers in a transmission range of S. If each mobile computer M_i which receives *mes* broadcasts *mes* to all mobile computers in a transmission range of M_i, all mobile computers with which S communicates by multi-hop message transmission receive *mes*. In DSR, in order to find a route from S to D, *Rreq* message is transmitted by flooding. In addition, in order to inform the detected route of S, *Rrep* message is also transmitted by flooding in an environment with uni-directional links.

1. A source mobile computer S broadcasts an *Rreq* message where $Rreq.seq \leftarrow \langle S\rangle$ and $Rreq.dst \leftarrow D$ to all mobile computers M_i within a transmission range of S.
2. On receipt of an *Rreq* message,
   - If M_i has already received the same *Rreq* message, M_i discards the message.
   - Otherwise, M_i appends an address of M_i to the end of *Rreq.seq* and broadcasts the *Rreq* message to all mobile computers in a transmission range of M_i.
3. By receiving an *Rreq* message, a destination mobile computer D appends an address of D to the end of $R_{S\rightarrow D}$ and detects a route $R_{S\rightarrow D}$ since $Rreq.dst = D$ and $Rreq.seq = R_{S\rightarrow D}$. D broadcasts an *Rrep* message containing $R_{S\rightarrow D}$ to all mobile computers in a transmission range of D.
4. On receipt of an *Rrep* message,
   - If M_i has already received the same *Rrep* message, M_i discards the message.
   - Otherwise, M_i broadcasts the *Rrep* message to all mobile computers in a transmission range of M_i.

   By receiving an *Rrep* message, S gets a sequence of addresses of mobile computers in $R_{S\rightarrow D}$ out of the *Rrep* message.

   S transmits an application message by source routing in accordance with $R_{S\rightarrow D}$.

[Example] As shown in Figure 2, a source mobile computer S broadcasts an *Rreq* message to M_1, M_6, M_{10}, M_{11} and M_{13} within a transmission range of S. Then, these mobile computers also broadcast an *Rreq* message. Since a communication link is uni-directional, S receives an *Rreq* message from M_6 and does not receive from M_1. By the successive broadcasts, i.e. flooding, a destination mobile computer D receives an *Rreq* message. Then, D gets a list $\langle S, M_1, M_2, M_3, M_7, D\rangle$ of addresses of mobile computers, i.e. $R_{S\rightarrow D}$.

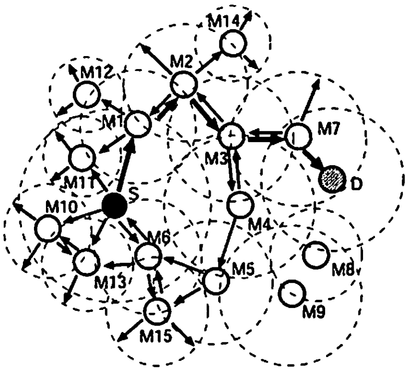

Figure 2: Flooding of Rreq in DSR.

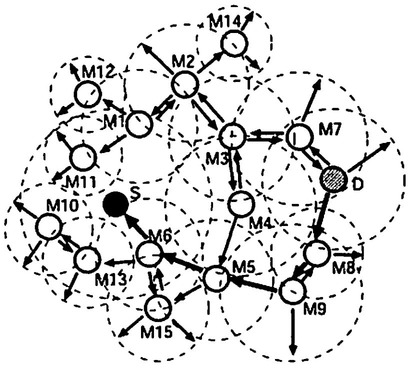

Figure 3: Flooding of Rrep in DSR.

Next, as shown in Figure 3, a destination mobile computer D broadcasts an *Rrep* message to M_7 and M_8. Then, these mobile computers also broadcast an *Rrep* message. By the successive broadcasts, i.e. flooding, a source mobile computer S receives an *Rrep* message containing $R_{S\rightarrow D} = \langle S, M_1, M_2, M_3, M_7, D\rangle$. □

In an environment only with bi-directional links, route cache works well in DSR. If a mobile computer M_i receives an *Rreq* message, M_i achieves transmission routes to all mobile computers included in *Rreq.seq*. In addition, an application message carries a list of addresses of mobile computers in a transmission route, a mobile computer on the route achieves routes to all mobile computers on the route. However, in an environment with uni-directional links, even if a mobile computer M_i receives an *Rreq* message, no route information is achieved since a route from M_i to M_j is not always available even though M_j is included in *Rreq.seq*. Here, only a mobile computer M_i on a detected route achieves message transmission route to a mobile computer M_j on the route where M_j is after M_i in the sequence of addresses of the route. Therefore, much less route information is stored in a route cache.

4 LBSR protocol

In an ad-hoc routing protocol using only bi-directional communication links, by detection of $R_{S\rightarrow D}$, $R_{D\rightarrow S}$ is achieved as a reverse route of $R_{S\rightarrow D}$. However, for achieving higher probability of successful transmission route detection between S and D, uni-directional communication links are also used to transmit messages. In DSR as discussed in the previous section, $R_{S \to D}$ and $R_{D \to S}$ are detected independently. In order to solve this problem, in LBSR, by combining detection of $R_{S \to D}$ and detection of $R_{D \to S}$, S detects a looped route $R_{S \to D} + R_{D \to S}$ containing both S and D. This is realized by flooding a control message *Lreq* and detecting a copy of the *Lreq* which is initiated by S, forwarded by D and received by S. While searching the looped route, S finds other looped routes which contain not D but S. These routes are used to reduce communication overhead caused by broadcast transmissions. If a mobile computer on an already detected looped route receives an *Lreq* message, it does neither broadcast nor discard but unicast the *Lreq* message to a next mobile computer on the looped route. By using this method, the copy of *Lreq* message is surely transmitted to a source mobile computer along the looped route without broadcast transmission. In order to achieve this unicast transmission, if a source mobile computer receives an *Lreq* message, i.e. a new looped route is detected, a confirmation message *Lconf* is transmitted along the looped route. The *Lconf* carries a sequence of addresses of mobile computers on the looped route, the *Lconf* is source routed and each mobile computer on the route gets an address of a next hop mobile computer to transmit future receiving *Lreq* messages to the source mobile computer.

0. Initially, $req\_flag_i \leftarrow false$, $stop\_flag_i \leftarrow false$, $next_i \leftarrow null$ and $hops_i \leftarrow \infty$ in each mobile computer M_i.
1. A source mobile computer S broadcasts an *Lreq* message where $Lreq.seq \leftarrow \langle S \rangle$ to all mobile computers M_i within a transmission range of S.
2. On receipt of an *Lreq* message, a mobile computer M_i ($\neq S$) processes the message as follows:
   - If $stop\_flag_i = true$, M_i discards the *Lreq* message.
   - If $M_i = D$ and $req\_flag_i = true$, M_i discards the *Lreq* message.
   - If $req\_flag_i = false$ and $stop\_flag_i = false$, $req\_flag_i \leftarrow true$ and M_i broadcasts the *Lreq* message to all mobile computers within a transmission range of M_i after appending an address of M_i to the end of *Lreq.seq*.
   - If $req\_flag_i = true$ and $stop\_flag_i = false$,
     - if $next_i = null$, M_i suspends the processing for the *Lreq* message. On receipt of an *Lconf* message, i.e., on storing an address into $next_i$, M_i resumes the processing from the beginning of step 2.
     - otherwise, i.e., an address has been stored in $next_i$, M_i appends an address of M_i to the end of *Lreq.seq* and transmits the *Lreq* message to a mobile computer whose address is $next_i$.
3. On receipt of an *Lreq* message, a source mobile computer S appends an address of S to the end of *Lreq.seq* and processes the message as follows:
   - If $detect\_flag = false$,
     - if an address of a destination mobile computer D is included in *Lreq.seq*, S sets *detect_flag* as *true* and transmits an *Lconf* message where $Lconf.seq \leftarrow Lreq.seq$ to a mobile computer whose address is just after an address of S in *Lconf.seq*.
     - otherwise, S transmits an *Lconf* message where $Lconf.seq \leftarrow Lreq.seq$ to a mobile computer whose address is just after an address of S in *Lconf.seq*.
   - Otherwise, S transmits an *Lstop* message where $Lstop.seq \leftarrow Lreq.seq$ to a mobile computer whose address is just after an address of S in *Lconf.seq*. □
4. On receipt of an *Lconf* message, a mobile computer M_i ($\neq S$) processes the message as follows:
   - If $next_i = null$, M_i stores an address which is just after an address of M_i in *Lconf.seq* and a number of addresses after an address of M_i in *Lconf.seq* into $next_i$ and $hops_i$, respectively, and transmits the *Lconf* message to a mobile computer whose address is just after an address of M_i in *Lconf.seq*.
   - Otherwise, i.e., an address has been stored in $next_i$,
     - if $hops_i$ is larger than a number of addresses after an address of M_i in *Lconf.seq*, M_i stores an address which is just after an address of M_i in *Lconf.seq* and a number of addresses after an address of M_i in *Lconf.seq* into $next_i$ and $hops_i$, respectively, and transmits the *Lconf* message to a mobile computer whose address is just after an address of M_i in *Lconf.seq*.
     - otherwise, M_i transmits the *Lconf* message to a mobile computer whose address is just after an address of M_i in *Lconf.seq*.
5. On receipt of an *Lstop* message, a mobile computer M_i ($\neq S$) sets *stop_flag* as *true* and transmits the *Lstop* message to a mobile computer whose address is just after an address of M_i in *Lstop.seq*.
6. On receipt of an *Lstop* message, S only discards it.

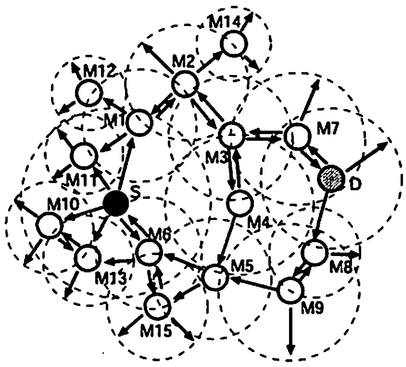

Figure 4: Flooding of Lreq in LBSR.

[Example] In Figure 4, an *Lreq* message is transmitted by using flooding. The message transmission is almost same as that for *Rreq* in DSR in Figure 2 except that D also transmits an *Lreq* message in LBSR. By the transmission of an *Lreq* message, some looped routes are detected as shown in Figure 5. Since a looped route $\langle S, M_6, S \rangle$ has been detected, when M_6 detects a part of looped route $\langle S, M_1, M_2, M_3, M_4, M_5, M_6 \rangle$ by receiving an *Lreq* message from M_5, M_6 does not broadcast but unicasts the *Lreq* message to S. Thus, S detects an additional looped route $\langle S, M_1, M_2, M_3, M_4, M_5, M_6, S \rangle$. Other looped routes $\langle S, M_6, M_{15}, M_6, S \rangle$ and $\langle S, M_1, M_2, M_3, M_4, M_5, M_{15}, M_6, S \rangle$ are detected by the same way. In addition, when M_5 receives an *Lreq* message from M_9, it does not broadcast

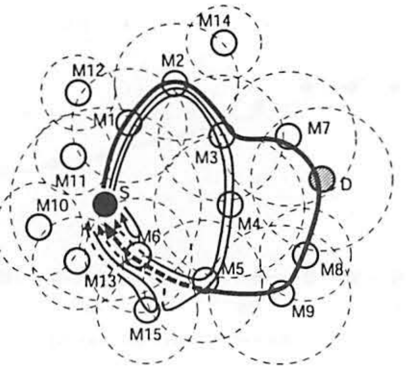

Figure 5: Unicasting of Lconf in LBSR.

but unicasts the *Lreq* to M_6. M_6 also unicasts the *Lreq* to S. Finally, S detects a looped route $\langle S, M_1, M_2, M_3, M_7, D, M_8, M_9, M_5, M_6, S\rangle$ containing both S and D. □

In LBSR, much more route information is stored into a route cache. A source mobile computer achieves message transmission routes to all mobile computers included in a detected looped route. In Figure 5, S gets routes to $M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9$ and D. By transmission of an *Lconf* message containing a list of addresses of mobile computers in a detected looped route, each mobile computer gets routes to all mobile computers in the list. For example in Figure 5, M_4 is included in a looped route $\langle S, M_1, M_2, M_3, M_4, M_5, M_6, S\rangle$. Hence, M_4 gets routes to S, M_1, M_2, M_3, M_5 and M_6 by receipt of an *Lconf* message.

5 Evaluation

This section discusses performance evaluation of LBSR comparing with DSR.

For detection of a transmission route from S to D in DSR, two independent floodings are required. In a single flooding, messages are transmitted through all the communication links. Therefore, the number of required messages is the same as the number of links $|\mathcal{L}|$. Hence, the total number of messages is $2|\mathcal{L}|$. On the other hand in LBSR, a single flooding and multiple unicast messages for transmitting *Lconf* messages are required. Let l_i be a number of mobile computers included in the ith detected looped route. Thus, the total number of messages in LBSR is $|\mathcal{L}| + \sum l_i$.

Figures 6 and 7 shows a simulation results for evaluation of number of control messages. Here, a simulation area is 500m × 500m and a distribution of diameters of a wireless transmission range is uniform between 20m and 200m. As show in Figure 6, the number of broadcast messages in LBSR is half of that in DSR. Figure 7 shows total numbers of messages. In LBSR, many unicast messages, i.e. *Lconf* messages, are transmitted. Especially, through a wireless communication link near S, an *Lconf* message is transmitted each time a looped route containing the link is detected. As mentioned in section 6, the total number of messages in LBSR is reduced by modifying the protocol.

Figure 8, 9 and 10 shows average numbers of cache entries in a mobile computer. Here, simulation assumptions are some as in Figures 6 and 7. Figure 8 shows average number of route cache entries in each mobile computer. In DSR as discussed in section 3, very few route cache entries are stored in an environment with uni-directional links. An average number of cache entries is 0.17 with 50 mobile computers. On the other hand in LBSR, much more cache entries are stored than that in DSR as mentioned in section 4 due to transmission of an *Lconf* message with an address sequence for a looped route. An average number of cache entries is 5.20 which is 31 times more than in DSR. The more a hop count between a source mobile computer and a destination one is, the more cache entries are stored in LBSR as shown in Figure 7. However in DSR, a number of cache entries is small and is not depend on a hop count between a source mobile computer and a destination one. Finally, Figure 8 shows relationship between ratio of uni-directional links and an average number of cache entries. As increasing the ratio of uni-directional links, an average number of cache entries is reduced since less looped routes are detected. However, even though 50% of communication links are uni-directional, 35 times more cache entries are stored in LBSR than in DSR. It is clear that much more route information is stored into a route cache in LBSR than DSR.

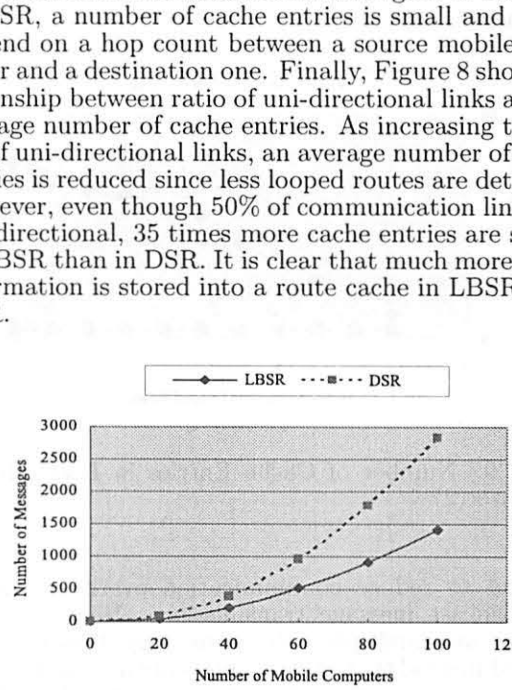

Figure 6: Number of Broadcast Messages.

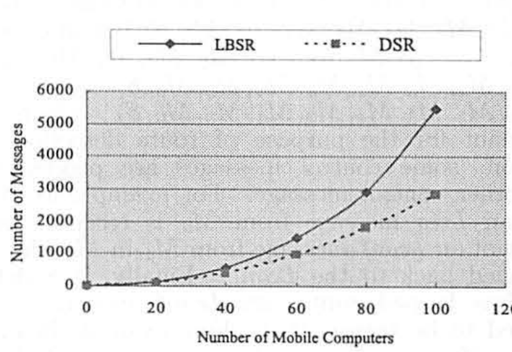

Figure 7: Total Number of Messages.

6 Concluding Remarks

This paper has proposed a novel ad-hoc routing protocol LBSR in which looped routes are detected to get a route from a source mobile computer to a destination

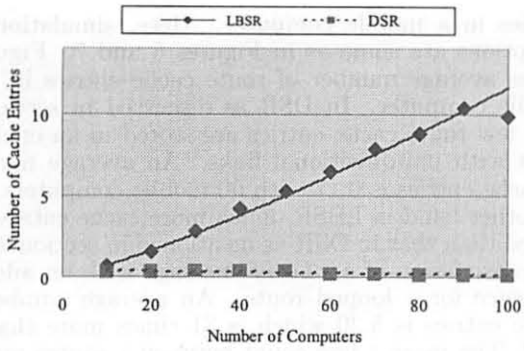

Figure 8: Number of Cache Entries in Each Mobile Computer(1).

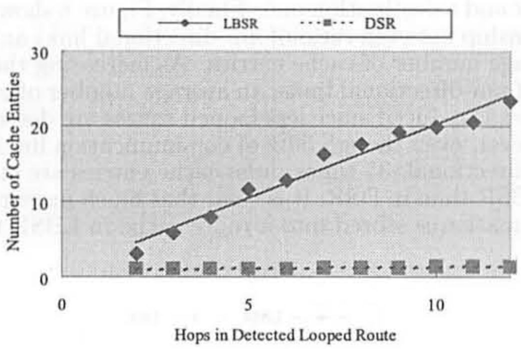

Figure 9: Number of Cache Entries in Each Mobile Computer(2).

one and to reduce communication overhead caused by broadcast message transmission. Here, a single flooding and multiple unicast message transmissions are used instead that two floodings are used in DSR. In addition, more route cache entries are stored in LBSR than in DSR through a simulation result. In order to reduce the number of messages in LBSR, detection of redundant looped routes should be avoided. For example in Figure 5, LBSR detects not only a looped route $\langle S, M_1, M_2, M_3, M_4, M_5, M_6, S\rangle$ but also looped routes $\langle S, M_1, M_2, M_1, M_2, M_3, M_4, M_5, M_6, S\rangle$, $\langle S, M_1, M_2, M_3, M_2, M_3, M_4, M_5, M_6, S\rangle$ and $\langle S, M_1, M_2, M_3, M_4, M_3, M_4, M_5, M_6, S\rangle$ which are redundant for the purpose of route detection. In addition, some control messages are piggied back to another control message. For example in Figure 4, if an *Lreq* message from M_9 is received before receipt of an *Lconf* message from M_4 in M_5, the *Lreq* is piggied back to the *Lconf*. Finally, if S detects a useless looped route, an *Lconf* message is not required to be transmitted. Fore example in Figure 5, even if S receives an *Lreq* message and detects a looped route $\langle S, M_1, M_2, M_3, M_4, M_5, M_{15}, M_6, S\rangle$ after detection of $\langle S, M_1, M_2, M_3, M_4, M_5, M_6, S\rangle$ and $\langle S, M_6, M_{15}, M_6, S\rangle$, S does not send an *Lconf* message since all the mobile computers on a newly detected looped route are included in some already detected looped route. In this case, these mobile computer have next hop mobile computers to unicast future receiving an *Lreq* message and there is no use to transmit an *Lconf* message along the looped route.

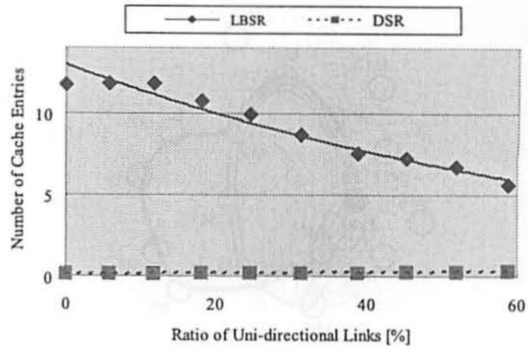

Figure 10: Number of Cache Entries in Each Mobile Computer(3).

By omitting detection of redundant looped routes and piggy back of control messages, communication overhead caused by unicast message transmissions in LBSR is reduced. In our future work, a modified protocol is designed and the performance is evaluated.

References

[1] "Radio Equipment and Systems(RES); HIPERLAN," TSI Functional Specifications (1995).

[2] "Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications," Standerd IEEE802.11 (1997).

[3] Corson, M.S. and Ephremides, A., "A Distributed Routing Algorithm for Mobile Wireless Networks," Wireless Networks, vol. 1, No. 1, pp. 61–81 (1995).

[4] 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).

[5] Mingliang, J., Jinyang, L., and Young, T., "Cluster Based Routing Protocol(CBRP) Functional specification," Internet Draft, draft-ietf-manet-cbrp-spec-01.txt (1999).

[6] Moses, Y. and Roth, G., "On reliable message diffusion." Proc. of the 8th ACM Symposium on Principles of Distributed Computing, pp. 119–128 (1989).

[7] Nishizawa, M., Hagino, H., Hara, T., Tsukamoto, M. and Nishino, S., "A Routing Method Considering Uni-directional Links in Ad-hoc Networks," Trans. of IPS Japan, vol. 41, No. 3, pp. 783–791 (2000).

[8] Park, V. and Corson, S., "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification," Internet Draft, draft-ietf-manet-tora-spec-04.txt (2001).

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

[10] Perkins, C.E. and Royer, E.M., "Ad-hoc On-Demand Distance Vector Routing," Proc. of International Workshop on Mobile Computing Systems and Applications, pp. 99–100 (1999).