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

Yousuke Sagawa, Tomonori Asano and Hiroaki Higaki
Department of Computers and Systems Engineering
Tokyo Denki University
E-mail: {sgw,tom,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 computers. Here, a communication link between two mobile computers is not always bi-directional since transmission power of mobile computers is not always the same. Though some ad-hoc routing protocols, e.g. DSR, support routing with uni-directional links, multiple independent 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 unicast message transmissions. According to simulation results, LBSR requires much messages than DSR. However, due to combination of a flooding and unicasts, much routes are detected and cached. Thus, communication overhead for each route is reduced.

アドホックネットワークにおけるループ型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
佐川 陽介　浅野 知倫　桧垣 博章
E-mail: {sgw,tom,hig}@higlab.k.dendai.ac.jp

アドホックネットワークは、移動コンピュータのみで構成されているため、移動コンピュータの電源容量の低下などにより移動コンピュータが必ずしも双方向の通信路で接続されるとは限らない。そのため、DSR などの幾つかのルーティングプロトコルは、ネットワーク内に片方向の通信路が存在する場合においても経路構築可能となっている。しかし、DSR では経路探索時に独立な複数のフラッディングを用いるために、通信オーバーヘッドが大きくなる。本論文では、片方向通信路を含むアドホックネットワーク内において 1 組のフラッディングと複数のユニキャストを組み合わせて経路探索を行なう新たなルーティングプロトコル LBSR (Loop-Based Source Routing) を提案する。シミュレーションの結果、1 回の経路探索に要するメッセージ数は LBSR の方が多いことが明らかとなった。しかし、DSR の 2 回のフラッディングが独立に行われるのに対し、互いに関係のあるフラッディングとユニキャストを組み合わせた LBSR では、90 倍の経路を同時に検出し、キャッシュすることができる。これによって、1 経路の検出に要するメッセージは、最大 20%に減少している。

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. In order 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 [1] and HIPERLAN [2] 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 access 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 for construction and maintenance of a wired network infrastructure and base stations are required. In addition, less flexibility is achieved due to a fixed infrastructure.

In an ad-hoc network, 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 a 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 AODV [6] and DSDV [5]. 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. Hence, for multi-hop transmission it is required to detect a route including uni-directional links for achieving higher probability to detect a route. Though DSR (Dynamic Source Routing) [3] has this property, the protocol uses two independent floodings and communication overhead is high.

This paper proposes a novel ad-hoc routing protocol LBSR(Loop-Based Source Routing Protocol) by which looped routes including a source mobile computer are detected, one of which includes a destination mobile computer, and detection of a target route is achieved by combination of a single flooding and multiple unicast message transmissions.

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. A communication link $\langle M_i, M_j \rangle$ is uni-directional, i.e. a message is transmitted from M_i to M_j, only while M_j is in a transmission range of M_i.

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

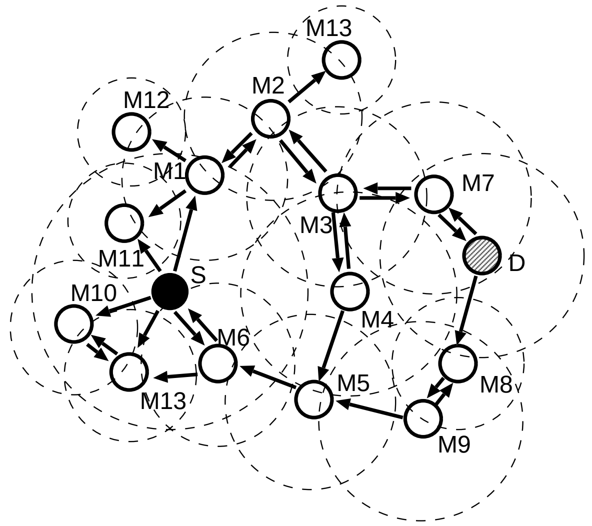

Figure 1: Ad-hoc network with uni-directional links.

each mobile computer is kept up to date to reflect any changes of a network topology. Hence, control message transmissions are required even though no mobile computer communicates with another one. DSDV [5] 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 messages destined to D. DSR [3] and AODV [6] 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 $R_{S \to D}$, D also finds a reverse route $R_{D \to S}$ is available. Hence, by transmitting a message including $R_{S \to D}$ through $R_{D \to S}$, S achieves $R_{S \to D}$ and starts to transmit application messages through $R_{S \to D}$. However, probability that $R_{S \to D}$ is detected is low. 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 $R_{S \to D}$, it is required to search $R_{D \to S}$ for transmission of $R_{S \to D}$ 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, a *flooding* is used to detect a transmission route from a source mobile computer S to a destination one D. A flooding is based on a message diffusion protocol in a wired network [4]. Most of wireless communication media on which wireless LAN protocols are implemented 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 transmissions receive *mes*. In DSR, in order to find a route from S to D, *Rreq* message is transmitted by a flooding. *Rreq* message keeps a sequence *Rreq.seq* of identifies of mobile computers having forwarded it. When D receives *Rreq*, $Rreq.seq = R_{S \to D}$. In addition, in order to inform the detected route of S, *Rrep* message carrying $R_{S \to D}$ is also transmitted by a flooding.

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 which is not a destination mobile computer in *Rreq.dst* has already received the same *Rreq* message, M_i discards the message.
   - Otherwise, M_i appends an identifier 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 detects a route $R_{S \to D}$ by appending its own identifier to the end of *Rreq.seq* since $Rreq.dst = D$ and $Rreq.seq = R_{S \to D}$. D broadcasts an *Rrep* message carrying $R_{S \to D}$ where $Rrep.dst \leftarrow S$ to all mobile computers in a transmission range of D.
4. On receipt of an *Rrep* message,
   - If M_i which is not a source mobile computer in *Rrep.dst* 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.
5. By receiving an *Rrep* message, S gets a sequence of addresses of mobile computers in $R_{S \to D}$ out of the *Rrep* message.
6. S transmits an application message by source routing in accordance with $R_{S \to D}$. □

[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,

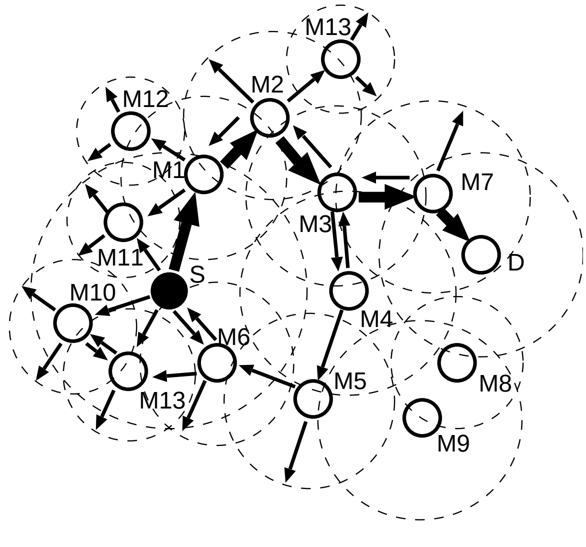

Figure 2: Flooding of Rreq in DSR.

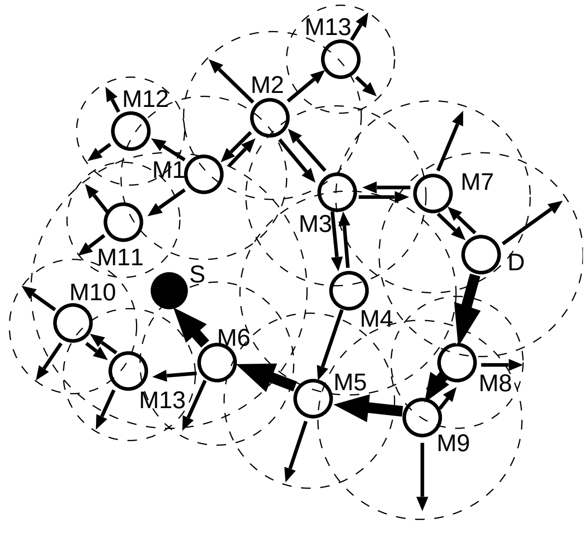

Figure 3: Flooding of Rrep in DSR.

S receives an *Rreq* message from M_6 and does not receive from M_1. By the successive broadcasts, i.e. a flooding, a destination mobile computer D receives an *Rreq* message. This message contains a sequence $\langle S, M_1, M_2, M_3, M_7 \rangle$ of identifiers of mobile computers, that is, D gets $R_{S \to D}$. 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. a flooding, a source mobile computer S receives an *Rrep* message carrying $R_{S \to D} = \langle S, M_1, M_2, M_3, M_7, D \rangle$. □

4 LBSR protocol

In an ad-hoc network including only bi-directional communication links, by detection of $R_{S \to D}$, $R_{D \to S}$ is achieved as a reverse route of $R_{S \to D}$. However, for achieving higher probability of success of detecting a transmission route from S to 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 LBSR, S detects a looped route $R_{S \to D} + R_{D \to S}$ containing both S and D. While searching the looped route, S finds other looped routes which contain not D but S. These routes are used to transmit *Lreq* message back to S with unicast transmission. That is, consider the case that a looped route $\langle S, L_1, \ldots, L_e, S\rangle$ which does not contain D has been detected. If $L_k (1 \leq k \leq e)$ receives an *Lreq* message which has been transmitted along a route $\langle S, R_1, .., R_r, L_k\rangle$, L_k detects another route $\langle S, R_1, .., R_r, L_k, .., L_e, S\rangle$. Hence, it is not required for L_k to broadcast the *Lreq*. By unicasting the *Lreq* along this route, S receives the *Lreq* and finds the looped route. Therefore, communication overhead caused by broadcast transmissions is reduced.

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 to $next_i$, 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 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 *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*.
4. On receipt of an *Lstop* message, a mobile computer M_i ($\neq S$) sets $stop\_flag_i$ as *true* and transmits the *Lstop* message to a mobile computer whose address is just after an address of M_i in *Lstop.seq*.
5. 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 M_i 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 M_i in *Lconf.seq*.
6. On receipt of an *Lstop* message, a source mobile computer S discards the *Lstop* message. □

[Example] In Figure 4, an *Lreq* message is transmitted by using a flooding. The message transmission is almost the 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 receives an *Lreq* message broadcasted by M_5 where $Lreq.seq = \langle S, M_1, M_2, M_3, M_4, M_5, M_6\rangle$, M_6 detects another looped route $\langle S, M_1, M_2, M_3, M_4, M_5, M_6, S\rangle$ and unicasts the *Lreq* message to S. Thus, S detects the looped route and a confirmation message *Lconf* is transmitted through the looped route. By the same way, on receipt of an *Lreq* message broadcasted by M_9, M_5 detects another looped route and the *Lreq* message is unicasted to S through M_6. Then, S detects an additional looped route $\langle S, M_1, M_2, M_3, M_7, D, M_8, M_9, M_5, M_6, S\rangle$. This is a

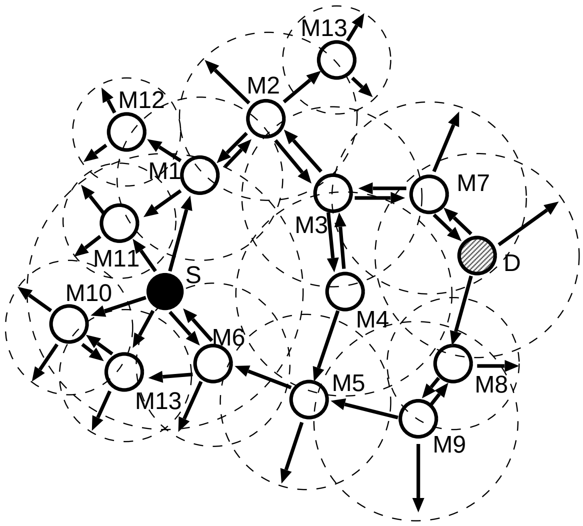

Figure 4: Flooding of Lreq in LBSR.

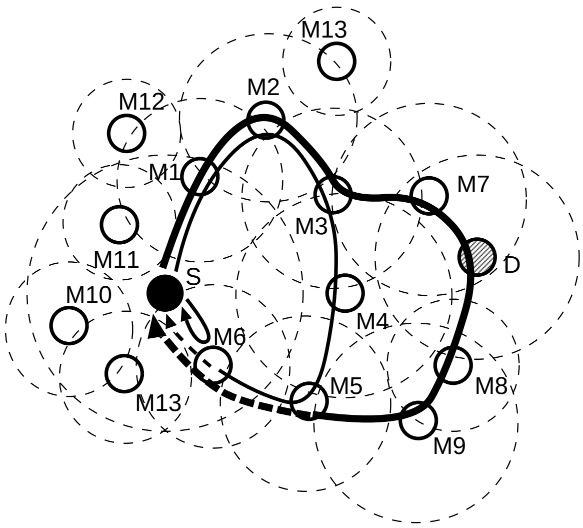

Figure 5: Unicasting of Lconf in LBSR.

target route $R_{S\to D} + R_{D\to S}$ since it contains D. □

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 maximum 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 a number of unicast messages for transmitting *Lconf* messages are required. Let l_i be a number of mobile computers included in the ith detected loop route. Thus, the total number of messages in LBSR is $|\mathcal{L}| + \sum l_i$.

Figures 6 and 7 show a simulation results. Here, a simulation area is 500m × 500m and distribution of diameter of wireless transmission ranges 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 wireless communication links near S, an *Lconf* message is transmitted each time a looped route containing the link is detected.

However, by the procedure of route detection, S detects route to all mobile computers included in detected looped routes. In addition, a mobile computer $M_i (\neq S)$ also detects routes to all mobile computers included in detected route containing S and M_i since on *Lconf* message is transmitted through a detected looped route. On the other hand in DSR, S only detects routes to mobile computers on $R_{S\to D} + R_{D\to S}$. Under an assumption of uni-directional links, even if a mobile computer M_i receives an *Rreq* and an *Rrep*, M_i gets no route. By transmitting an application message, M_i on $R_{S\to D} + R_{D\to S}$ gets routes to mobile computers downstream in the route since the message carries $R_{S\to D} + R_{D\to S}$ for source routing. Figure 8 shows a simulation result of distributions of number of cached routes in DSR and LBSR. In average, 0.1 and 9.0 routes are cached, respectively. Hence, average number of messages for detecting each route is 1597 and 323 in a case of 80 mobile computers. Hence LBSR reduces required communication overhead.

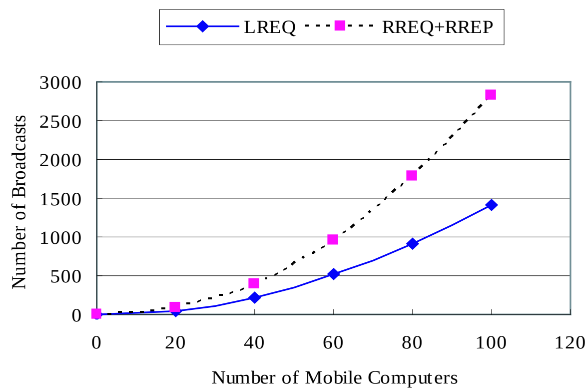

Figure 6: Number of Broadcast Messages.

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 one and to reduce communication overhead caused by broadcast message transmissions. Here, a single flooding and unicast message transmissions are used instead that two floodings are used in DSR. In future work,

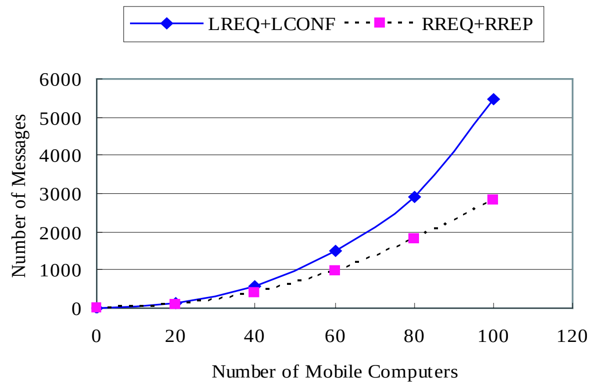

Figure 7: Number of Messages.

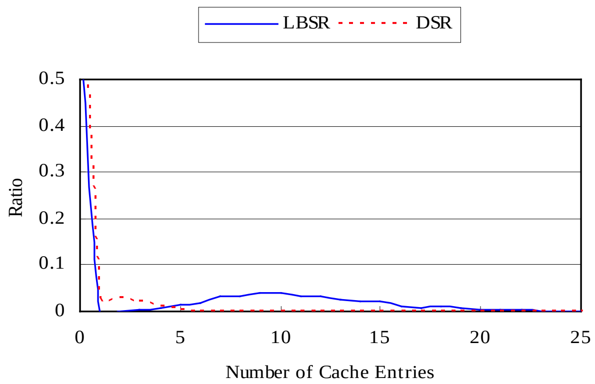

Figure 8: Distribution of Number of Cache entries.

the authors will evaluate the performance of LBSR in simulation and a prototype system.

References

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

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

[3] 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-07.txt (2002).

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

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

[6] 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).