Integrated protocol for optimized link state routing and localization: OLSR-L

Integrated Protocol for Optimized Link State Routing and Localization: OLSR-L

Hiroshi Mineno[a], Kazuyoshi Soga[b], Tomoya Takenaka[d], Yoshiaki Terashima[c], Tadanori Mizuno[d]

[a]*Faculty of Informatics, Shizuoka University, Hamamatsu, Shizuoka 432-8011, Japan*
[b]*Graduate School of Informatics, Shizuoka University, Hamamatsu, Shizuoka 432-8011, Japan*
[c]*Mitsubishi Electric Corporation, 5-1-1, Ofuna, Kamakura, Kanagawa, 247-8501, Japan*
[d]*Graduate School of Science and Technology, Shizuoka University, Hamamatsu, Shizuoka 432-8011, Japan*

Abstract

Localization protocol is important for estimating node positions in a wireless multi-hop network. Routing protocol is also important for controlling paths. In previous research, localization and routing protocols have been discussed and evaluated separately. In this paper, we propose an integrated protocol for optimized link state routing (OLSR) and OLSR based localization (ROULA). Our protocol enables simultaneous localization and routing. ROULA's localization is performed using OLSR overhead such as hello packets and routing tables. The routing overheads and the processing procedures can be efficiently integrated. We demonstrate that the integrated protocol for ROULA and OLSR enables simultaneous localization and routing.

Keywords: Localization, wireless multi-hop networks, routing protocol, location estimation.

1. Introduction

Wireless multi-hop networks, including ad-hoc and sensor networks, are discussed extensively in the literature. Wireless nodes can be used to construct an ad-hoc network without an infrastructure base-station by using multi-hop communication protocol. Therefore, they can be deployed easily compared with wired-infrastructure networks.

In the wireless multi-hop network, the data communication and positioning protocols are important techniques that mutually interact. Figure 1

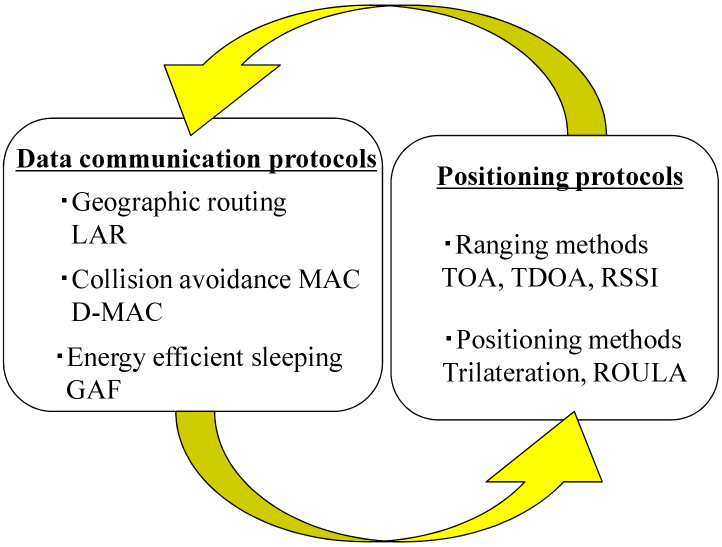


Figure 1: Relationship between data communication and positioning protocols [1].

shows the relationship between the two protocols [1]. For example, in location aided routing (LAR) [2], redundant flooding is controlled by determining the location information for a terminal. Geographical adaptive fidelity GAF [3] enables low power consumption by using location information. Furthermore, Y. Ko et. al. described a proposal in which a node uses a limited wireless range to send packets in D-MAC [4] with a directional antenna to a receiver when the transmitter knows the receiver's location. In all of these studied protocols, location information improves the data communication protocol. Therefore, a positioning protocol is necessary to acquire the location information. The global positioning system (GPS) might be used as a simple solution. However, GPS increases node hardware costs and it cannot be used indoors. Therefore, equipping all the nodes with GPS is not a practical solution. Hence, a positioning protocol is needed to obtain the position of a node autonomously without being dependent on GPS.

The positioning protocol consists of ranging (measurement of distance) and positioning (calculation of position) phases. The ranging protocols include time-of-arrival (TOA), time difference of arrival (TDOA), and received signal strength (RSS). Overhead naturally occurs with ranging. In the positioning protocols, including trilateration and the ROULA [6] that we previously developed, overhead, such as packet exchanging, naturally occurs with positioning. When a positioning protocol is used, this overhead has to be taken into account. Therefore, a data communication protocol that uses location information requires ranging and positioning protocols as shown in Figure 1. Moreover, the positioning protocols require data communication protocols to provide location information. However, the data communication and positioning protocols have been discussed separately in existing research [2, 3, 4]. Thus, we need to design a protocol that integrates the data communication and positioning protocols.

We propose an integrated protocol for optimized link state routing (OLSR) [5] and a localization protocol called OLSR-L that simultaneously conduct a localization technique called ROULA [6], which we previously developed. Our protocol enables simultaneous localization and routing. OLSR-L is based on original OLSR source code [7]. Hence, by only using the extended OLSR protocol, nodes can know the relative node positions of the ad-hoc network.

The contributions of this paper are as follows. First we analyze the two existing routing and localization protocols and show that they share several functionalities in the software programs. These overlapping functionalities are integrated efficiently in our proposed protocol so that localization and routing operate simultaneously. The overhead of our integrated protocol is reduced by exploiting the overhead in the routing protocol.

This paper is organized as follows. Related work on localization and routing is described in Section 2. Optimized link state routing and localization (OLSR-L) are described in Section 3. We explain the implementation of OLSR-L in Section 4. Section 5 presents an analysis of the source code and overhead evaluations of OLSR-L. Section 6 concludes the paper and mentions future work.

2. Related work

2.1. Localization protocol

Much research on localization techniques exists in the literature. Localization techniques can be categorized into two types [8]. One is range-based localization that uses ranging devices such as ultra-sound [9, 10, 11]. The other is range-free localization that does not use ranging devices [12, 13, 14, 15]. Ranging devices raise cost to nodes in wireless multi-hop networks. Range-free localization is attractive for wireless multi-hop networks because it only requires connectivity information. One example of range-free localization is

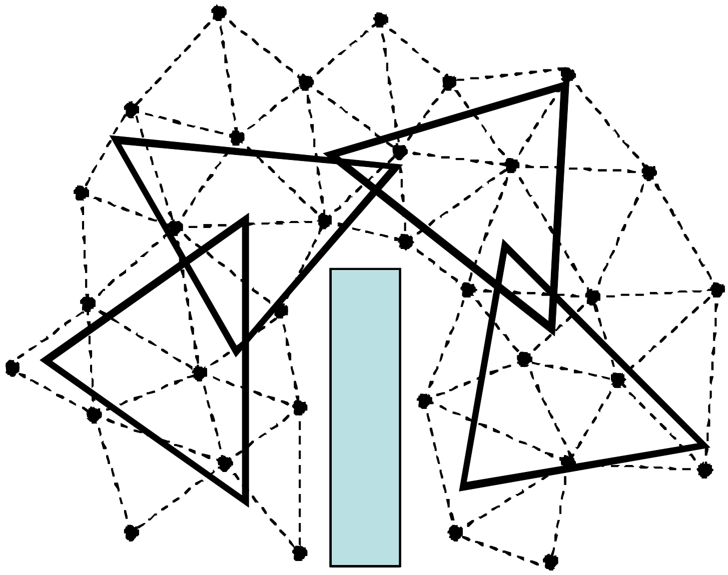


Figure 2: Conceptual representation of ROULA.

DV-HOP [12]. DV-HOP uses trilateration to calculate the hop-count distance from the positions of three or more anchor nodes. An anchor node's position is determined beforehand by using GPS or some other measurement means. The drawback of DV-HOP is that anchor nodes are needed in advance. Anchor-free localization (AFL) [13] does not require any anchor nodes in advance. Therefore, nodes can estimate their own positions using only the connectivity information of the ad-hoc network.

We previously developed an anchor-free localization technique called ROULA [6]. Figure 2 illustrates a conceptual representation of ROULA. Each point in the figure represents a node, and links are shown as lines. The basic idea is that nodes find relative coordinates of regular triangles in a network, and overlapping regular triangles are merged. Nodes then have a global relative coordinate system, shown in Figure 2. An extensive simulation experiment demonstrated that an approximation to regular triangles is effective in localization. ROULA chooses the farthest 2-hop node by using multi-point relay (MPR) nodes [5]. The farthest 2-hop node is a candidate to be a vertex of a regular triangle. Each node floods the network with TRI_NOTICE packets that each carry a farthest 2-hop node list. Nodes then calculate relative local coordinates by matching regular triangles on the basis of information in the

TRI_NOTICE packets. The algorithm is described as follows.

1. **MPR selection:** Nodes flood hello packets containing their own 1-hop node lists to their 1-hop nodes. Once a node has the same 2-hop nodes list, the node is selected as a MPR node.
2. **Farthest 2-hop node selection:** Each node selects the farthest 2-hop node from each MPR node.
3. **Matching regular triangle:** Nodes flood TRI_NOTICE packets to their farthest 2-hop nodes with their farthest 2-hop nodes list. Then, nodes that received TRI_NOTICE packets match regular triangles by using the received farthest 2-hop nodes lists. Next, nodes obtain local coordinates by merging their overlapping regular triangles.
4. **Merging local coordinates:** Local coordinates can be merged into one set of global coordinates. We assume a sink node merges all the maps of local coordinates in the network. A sink node floods MAP_REQ packets to all nodes in the network. Receiving nodes send back MAP_REP packets containing their local coordinates.
5. **Converting to absolute coordinates:** Although the global coordinates are relative coordinates, if at least three anchor nodes are in the network, the relative coordinates can be converted into absolute coordinates that have the correct network orientation. This phase is optional.

ROULA does not depend on anchor nodes and can deal with non-convex network topologies. Moreover, its performance has been evaluated in various network scenarios. A detailed algorithm and performance validation can be found in [6].

2.2. Routing protocol

A routing protocol is essential for wireless multi-hop networks. The basic functionality of routing is to relay data using multi-hopping in an ad-hoc network. Many network protocols exist for wireless multi-hop networks. The internet engineering task force (IETF) of the mobile ad-hoc network working group (MANET-WG) [16] standardizes routing control technology. MANET-WG standardized four routing protocols for request for comments (RFC): OLSR, ad hoc on-demand distance vector (AODV) [17], dynamic source routing (DSR) [18], and topology dissemination based on reverse-path forwarding (TBRPF) [19].

Routing protocols in wireless multi-hop networks are categorized into reactive, proactive, and position-based routing types. In the reactive routing protocol, when a node sends route requests, it checks for neighbor nodes and makes a routing table. AODV and DSR are categorized as reactive routing protocols. Because the routing is made after a route request is made, the reactive protocol incurs a certain delay before actual communication begins. The reactive routing protocol is only suitable for networks that can tolerate delays.

In the proactive routing protocol, a node makes a routing table in advance. Therefore, the node can start communication immediately after a data request. OLSR and TBRPF are categorized as proactive routing protocols. Link information needs to be exchanged to make a routing table. Nodes always send packets and must confirm neighbor nodes. The proactive protocol is suitable for networks with frequent communication. Routing tables are made in OLSR as follows.

1. **MPR selection:** Nodes flood hello packets containing their own 1-hop nodes list to their 1-hop nodes. Once a node has the same 2-hop nodes list, the node is selected as MPR node.
2. **Topology control:** Nodes periodically flood topology control (TC) packets to discover link state information throughout the network. Nodes in OLSR select the MPR nodes as relay nodes to make routing table. OLSR enables efficient dissemination of packets by using the MPR nodes.

In the position-based routing protocol, a node achieves efficient routing controls by using location information. Therefore, redundant flooding messages can be eliminated. However, the node has to know the position of the destination in advance. Therefore, the protocol is not compatible with ROULA.

Here, we discuss which protocol is most compatible with ROULA. Table 1 summarizes the characteristics of existing standardized MANET routing protocols. Reactive routing protocols, such as AODV and DSR, update the routing table when communication is demanded. They have an initial delay for starting actual data communication. On the other hand, proactive routing protocols, such as OLSR and TBRPF, update the routing table periodically so each node can start actual data communication at any time. The power consumption of each node is larger in proactive routing protocols than in reactive routing protocols because the node must exchange hello packets

Table 1: Comparison of standardized MANET routing protocols.

	AODV	DSR	OLSR	TBRPF
Type	Reactive	Reactive	Proactive	Proactive
Routing table	Updated when communication is demanded	Updated when communication is demanded	Updated periodically	Updated periodically
Initial delay	Slow	Slow	Early	Early
Power consumption	Small	Small	Large	Large
Influence of control packets on traffic	Large	Large	Little	Little

periodically to maintain the routing table. If the node does not move frequently, the interval between hello packets can be long in proactive routing protocols. This means the ratio of control packets to whole traffic is smaller than that for a reactive routing protocol, because a reactive routing protocol needs to send control packets, such as route requests and route responses during each communication. Therefore, the influence of control packets on traffic in reactive routing protocols is large and that in proactive routing protocols is small.

Therefore, we selected OLSR as the most suitable routing protocol for use with ROULA because ROULA requires neighbor node (i.e., connectivity) information to estimate node positions in a network. OLSR periodically holds and updates 1-hop neighbor information and ROULA uses OLSR's MPR node to estimate node distance accurately. We think ROULA can use OLSR's overhead to perform localization. Thus, we decided to select an existing OLSR protocol to achieve simultaneous localization and routing in the wireless multi-hop network.

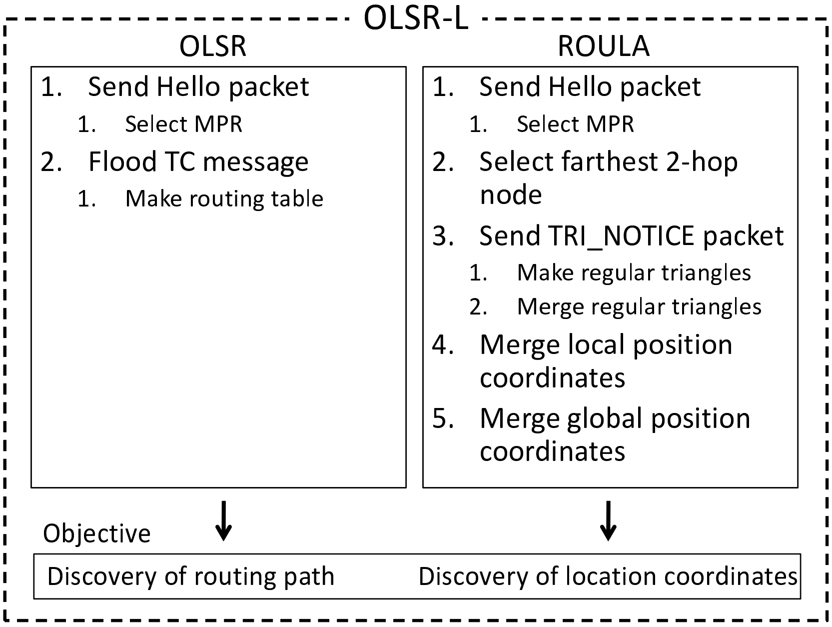


Figure 3: Flow of OLSR-L.

3. Optimized link state routing and localization (OLSR-L)

3.1. Overview

The objective of developing OLSR-L is to achieve simultaneous routing and localization in wireless multi-hop networks. When existing localization and routing protocols have been directly integrated into one protocol, communication overhead has doubled because the two protocols operate separately, and the overlapping functionalities in both protocols decrease the communication overhead. Therefore, we incorporated ROULA into OLSR and developed a technique that can conduct localization and routing simultaneously.

The OLSR protocol maintains connectivity information about the latest 2-hop node and the MPR node by continuously flooding hello packets. This characteristic implies that OLSR and ROULA are with each other. Therefore, efficient integration for localization and routing can be achieved on a wireless multi-hop network.

Figure 3 shows sequences of the two protocol algorithms, and we briefly describe the integration method. In OLSR, the node first sends hello packets and selects MPR nodes that relay them to a node for flooding. The node then floods topology control (TC) messages through the MPR nodes and constructs an overall routing table in the network. In ROULA, the node first sends hello packets and then selects the MPR nodes that are used to estimate the 1-hop node distance accurately. Next, the node selects the farthest 2-hop nodes and floods TRI_NOTICE packets to them. The next node finds regular triangles and merges them into global coordinates. Because both OLSR and ROULA use hello packets to sync neighbor nodes, they can be integrated. The TC packets are flooded periodically, and the TRI_NOTICE packet is incorporated in the TC packet. The results of the merged regular triangles are sent to a sink node. In original ROULA, we assume that a sink node, where the local coordinates information should be delivered, merges all the local coordinates in the network to the global map. The sink node conducts global localization on the basis of these results and floods its results.

As alternated, individual nodes can compute their own local maps using their local information, and then the local maps are merged to form a global map. For example, two local maps can be merged together based on their common nodes. The transformation is computed to transform the coordinates of the common nodes in one map to those in the other map. The matched and merged regular triangles and the global localization are included in the TC packets.

3.2. Sending hello packets

Next, we describe the OLSR-L operations sequentially in detail. OLSR nodes periodically send hello packets about themselves to other nodes in the network to build local link information. A node initially sends an adjacent node a hello packet containing its address as a statement of its existence. All nodes send these packets, so each node learns how many neighboring nodes it has.

In ROULA, nodes also hold 1-hop neighbor information by sending hello packets periodically. This connectivity information is used to choose the farthest 2-hop node from the farthest 2-hop node list. Each node can select the MPR list and the 2-hop node list by only using a hello packet. Therefore, we have found that a hello packet does not have to be changed from the original OLSR protocol.

3.3. MPR node selection

In the OLSR protocol, MPR information is periodically sent in the hello packets. Each node chooses MPR nodes from the neighbor node link list. Nodes insert address lists of MPR nodes into hello packets. A node that receives this information learns that it has been chosen as an MPR. A node that chooses itself to be an MPR is managed as an MPR selector node. Such a node knows whether it should relay packets as an MPR or delete them.

The distance characteristics of MPR nodes in OLSR decreases the distance estimation error [6]. The distance to an MPR node can be estimated accurately without changing the MPR selection in the OLSR selection procedure. The OLSR protocol is assumed to be in the network layer. Therefore, hello packet flooding and MPR selection can be merged into one operation by doing the processing in the network layer. In other words, changes do not need to be made to the MPR selection procedure.

3.4. Farthest 2-hop node selection

In the ROULA farthest 2-hop node selection, each node finds the farthest 2-hop node of all its 2-hop nodes among the MPR nodes. Note that nodes in the farthest 2-hop node selection do not require any connectivity information other than the MPR selection. The farthest 2-hop node becomes a candidate to be a vertex of a regular triangle. This is because, with a uniform node density, the connectivity between 2-hop nodes and the source node is smaller when the node distance is farther. On the basis of this assumption, nodes select the farthest 2-hop nodes.

Figure 4 illustrates that node S selects a node F as the farthest 2-hop node. The small solid circles are nodes, and the dashed circles are the communication ranges of nodes. The numbers in brackets show the number of connectivity for S's MPR nodes. The number of connectivity will be small if the distance to node S is large. Thus, the farthest 2-hop node is selected as a vertex of a regular triangle. Each node selects the farthest 2-hop node on this basis. Then, each node adds the farthest node list and farthest selector list to the local link information. The local link information is put into the TC packet.

3.5. Sending TC packets

Not only the hello packets but also the TC packets are frequently transmitted in the OLSR protocol. Each node sends hello packets only to its neighbor nodes, whereas it floods TC packets to the whole network. Each

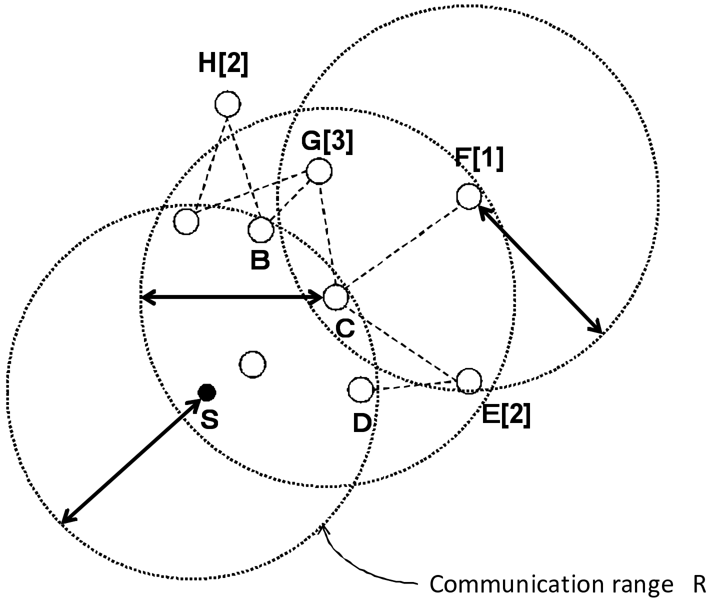


Figure 4: Example illustration of farthest 2-hop node selection. Node S selects node F as farthest 2-hop node. The numbers in brackets show the number of connectivity for S's MPR nodes.

node calculates its communication route and makes a routing table on the basis of the topology information. The topology does not consist of all link information; instead it is built from the MPR selectors list. Thus, the number of links that each node manages is considerably smaller than the actual number of links. All nodes chosen as MPRs send TC packets periodically. The MPR selector list and the node number are included in the TC packet. All nodes on the network learn of all MPRs and the MPR selector list this way. Hence, each node can learn the topology of the whole network from the TC packet. Each node calculates the shortest course and makes a routing table by referring to the topology. Each node communicates with all nodes quickly. The packet format of the original TC packet is shown in Figure 5, and the packet format of the modified TC packet is shown in Figure 6.

The modified TC packet includes the number of MPR selector nodes, the number of farthest 2-hop nodes, identification of a sink, and the amount of information. The farthest 2-hop node information is needed to generate approximate regular triangles. The number of MPR selectors and the number of farthest 2-hop nodes are needed. Also, the sink information is required

0	31
Advertised neighbor sequence number	Reserved
Advertised neighbor main address	
...	

Figure 5: Packet format of original TC packet.

0			31
Advertised neighbor sequence number		Reserved	
Number of MPR selector nodes	Number of farthest 2 hop nodes	Identification of a sink	Volume of information
Advertised neighbor main address			
...			
Farthest 2 hop node address			
...			
Information of sink node (or not sink node)			
...			

Figure 6: Packet format of modified TC packet.

to know whether the transmission came from a sink node. Information from each node is used by the sink node to create a map.

3.6. Matching regular triangles and merging local coordinates

Figure 7 illustrates the process of finding the nodes that form the vertices of regular triangles. Here, let us focus on matching node B to triangle ABC. The arrows show the direction of the farthest 2-hop nodes. For example, node A has three farthest 2-hop nodes AB, AC, and AC. Nodes A and C flood TRI_NOTICE packets, including the list of farthest 2-hop nodes from themselves to the farthest 2-hop nodes. Node B learns that nodes A and C chose it as a farthest 2-hop node. Then, node B finds the regular triangle by matching the list of farthest 2-hop nodes that receive the two combinations AC or CA of the triangle ABC.

The regular triangle has three MPR nodes and farthest 2-hop nodes. Local location coordinates are assigned to these nodes in accordance with the coordinates of a regular triangle.

A local map is generated as follows. When three or more points on two local maps include an overlapped regular triangle, the maps are merged into one. The original maps are eliminated, and the merged map becomes a new

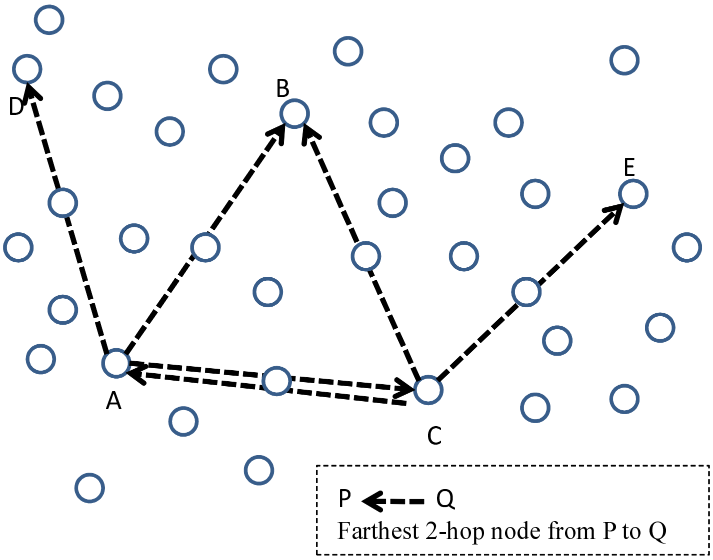


Figure 7: Matching of regular triangle.

local map. This process of merging two local maps is performed recursively until no more maps can be merged. The final map made by this procedure becomes the local map of the node.

4. Implementation

The implementation environment of OLSR-L is an ns-2 platform [20]. The source code of the OLSR is available on the Web [7]. Although ns-2 is a network simulation tool, the source code compiled in ns-2 can work as an actual wireless networking protocol on a personal computer (PC). Therefore, we used ns-2 code built by INRIA for the implementation. The localization ROULA functionalities are being incorporated into an OLSR protocol. The source code of ROULA is made based on the OLSR.

We first analyzed the original source code of OLSR with the UML modeling tool created by Enterprise Architect (EA) [21] and added classes that are required to conduct ROULA. The overall class relationship of the OLSR-L is shown in Figure 8.

The small boxes with the solid lines in Figure 8 represent the original OLSR class. Plain arrows represent a reference to a class, and arrows with white triangle heads represent a generalization to a class. The classes of

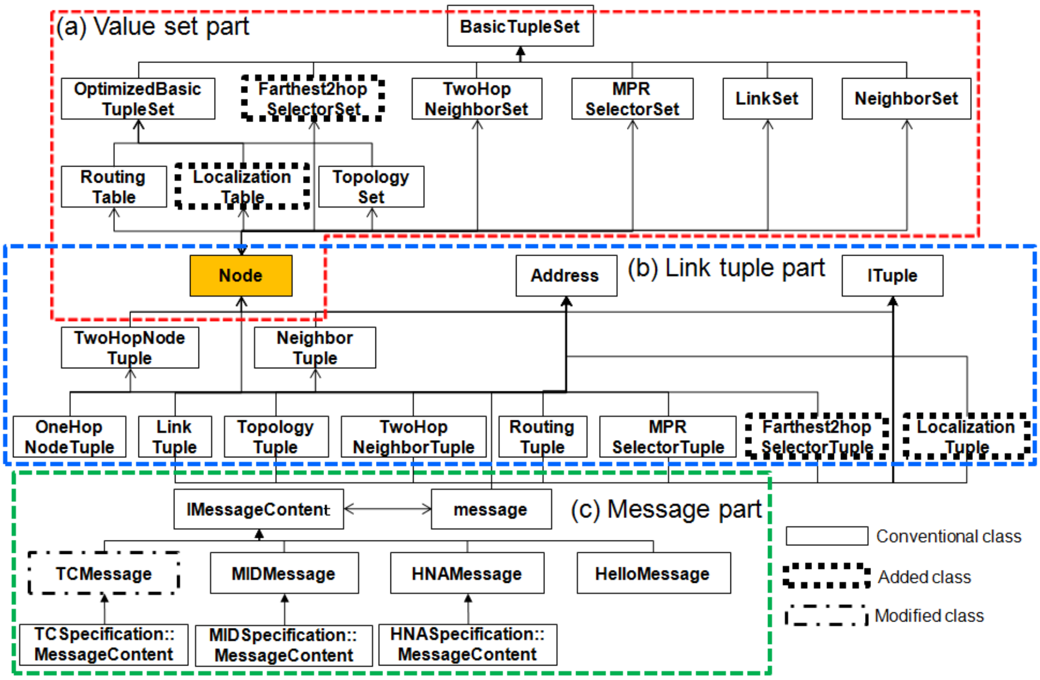


Figure 8: Modified class figure of OLSR-L visualized by using UML modeling tool of EA [21]. A small box with a solid line represents original OLSR class. Each bold box with a solid line represents (a) a value set part, (b) a link tuple part, and (c) a message part. Each box with a dash-and-dot line represents modified class, and each box with a dashed line represents an additional class from the original OLSR class.

OLSR-L are divided into three groups based on functionalities. Three bold boxes with solid lines show (a) value set part, (b) link tuple part, and (c) message part. Four classes surrounded by dashed lines show additional classes of OLSR-L from the original OLSR class. Two classes surrounded by dash-and-dot lines show modified classes. We describe the implementation details of OLSR-L class below.

4.1. Value set part

The "Value set part" in Figure 8 (a) is the set of class in which a node sets a value, such as MPR nodes and neighbor information. The "node" class is the main class in OLSR-L and the core of routing and localization are processed in this class.

The added classes are the "Farthest2hopSelectorSet" and "LocalizationTable". The "Farthest2hopSelectorSet" is a class that sets the list of the nodes that chose themselves to be the farthest 2-hop nodes. The "MPRSelectorSet" is a class that sets the list of the nodes that chose themselves to be the MPR nodes. The difference between these two classes is the address treated. The difference is whether they have a 1-hop neighbor node address or a 2-hop neighbor node address.

The "LocalizationTable" is a class that sets the map created by the "node" class. The "RoutingTable" is a class that sets that routing table created by the "node" class. The difference between these two classes is the address treated. The difference is in the map or route.

4.2. Link tuple part

The "Link tuple part" in Figure 8 (b) is a set of the classes that hold each value, such as MPR and neighbor information. The added classes are "Farthest2hopSelectorTuple" and "LocalizationTuple". "Farthest2hopSelectorTuple" is a class that holds a list of nodes that chose themselves to be farthest 2-hop nodes. "MPRSelectorSet" is a class that holds a list of nodes that chose themselves to be MPR nodes. The difference in these two classes is the address treated.

The "LocalizationTuple" is a class that holds each node's position information. The "RoutingTuple" is a class that holds the routing table. The difference between these two classes is the address treated in the map or route.

4.3. Message part

The "Message part" in Figure 8 (c) is the set of the classes that defines message formats. The class of the message part has not changed from the original OLSR class except for "TCMessage" class. There are four messages used in OLSR class.

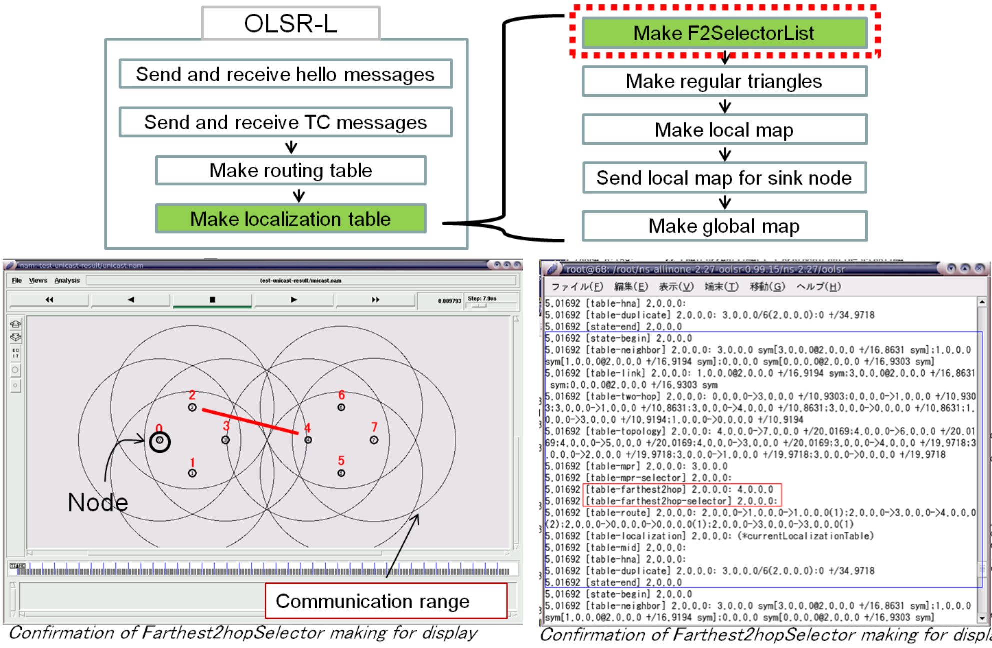
A multiple interface declaration (MID) packet is a message used when a node uses two or more interfaces. A host and network association (HNA) packet is a message used when the node is connected with a network outside a wireless multi-hop network. The hello packet was explained in Section 3.2. The hello message is not changed for OLSR and ROULA because choosing an MPR node does not require any modification. The TC packet was explained in Section 3.5. A new message for sending the farthest 2-hop selector and position information was not created because creating new messages increases the network overhead. We only modified the TC packet to include the farthest 2-hop node list and localization information instead of creating a new message. Therefore, the number of messages is not changed from the original OLSR protocol.

5. Evaluation

5.1. Example operation

In our previous work [6], ROULA was implemented on OMNET++ [22]. However, OLSR was implemented in ns-2 [20]. Hence, we developed OLSR-L by porting ROULA functionalities into the source code of ns-allinone-2-OLSR [7].

Screenshot examples of OLSR-L are shown in Figures 9 and 10. In Figure 9, we can check the function to make the farthest 2-hop selector node list. In topology like the left figure, node 2 and node 4 mutually become farthest 2-hop nodes. In Figure 10, we can check the function to make regular triangles. Nodes 0 and 5 are taken into account. For node 0, the 1-hop nodes are nodes 5 and 1, and the farthest 2-hop nodes are nodes 2 and 4. Nodes 1 and 5 are also MPR nodes. Because nodes 2 and 4 are each other's farthest 2-hop nodes, node 0 generates a regular triangle with nodes 0, 2, and 4 as vertices. Similarly, node 5 generates a regular triangle with nodes 5, 1, and 3 as vertices. This selection is checked by using two quadrangles as shown in Figure 10. All the nodes were similarly checked, and the check was done to ensure that no node operation was wrong.



Confirmation of Farthest2hopSelector making for display with ns-2

Confirmation of Farthest2hopSelector making for display with CUI

Figure 9: Check of F2Selector list generation (CUI).

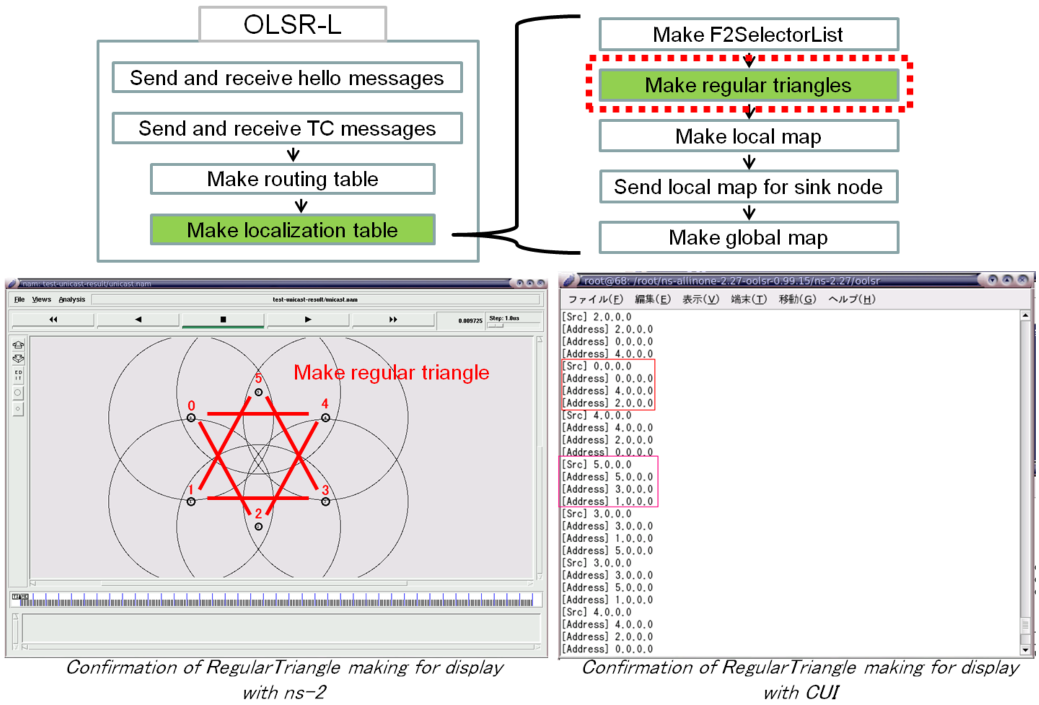
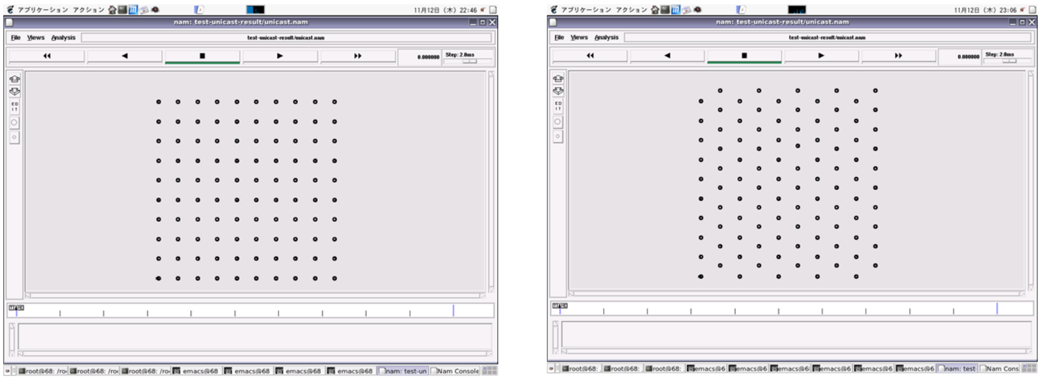


Figure 10: Check of regular triangle generation (CUI).



(a) Grid node placement

(b) Triangle lattice node placement

Figure 11: Visualizing (a) grid and (b) triangle lattice node placements using GUI of ns-2 [20].

5.2. Overhead

Next, let us describe an evaluation of the overhead of OLSR-L by using ns-2. One hundred nodes were placed in 1000×1000 field. The node placements were grid and triangle lattice as shown in Figure 11, and the communication range was 250. Each node was separated by 100. The MPR_COVERAGE parameter influences the number of MPRs in OLSR. By default, it is set to 1. Setting MPR_COVERAGE to 3 increases the number of MPRs and raises the localization accuracy [6]. Because MPR_COVERAGE has no influence on the routing control algorithm, its value in OLSR-L was also set to 3.

Figure 12 plots the number of times TC packets are generated by ROULA, OLSR, and OLSR-L. Figure 12 (a) (b) shows the number of messages on the grid node placement and Figure 12 (c) (d) shows of number of messages on the triangle lattice node placement. Figure 12 (a) (c) shows the number of TC packets that OLSR-L generated, and Figure 12 (b) (d) shows the number of TC packets generated when ROULA and OLSR protocols are operated separately. In these cases, OLSR-L generated about 40% fewer TC packets than the two protocols conducted separately.

About performance overhead compared with OLSR, data transfer delay and throughput are not affected by the modification. Because it operates the same way as general proactive routing protocols, OLSR-L makes routing tables in advance and updates them periodically, so there is no modification in data transfer part. The performance will be affected by node density

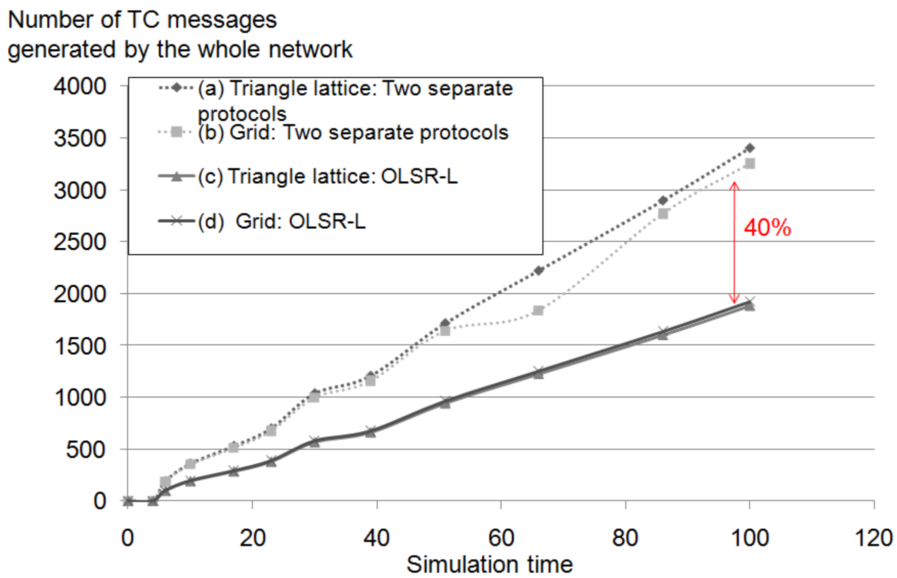


Figure 12: Number of TC packets generated by ROULA, OLSR and OLSR-L.

and the frequency of TC packets. This was evaluated by several studies in detail [24, 25], and localization performance of ROULA was evaluated in our previous work [6]. Therefore, we do not include the data transfer performance and localization performance evaluations in this work.

5.3. Protocol comparison

The software of OLSR, ROULA, and OLSR-L was compared. A comparison of the protocols is shown in Table 2. The OLSR-L is 24,419 lines. When OLSR and ROULA are integrated, the software code is 33,721 lines. By integrating OLSR and ROULA efficiently, 9302 lines of source code were eliminated, because the functionalities of the two protocols overlapped. In a wireless multi-hop network with various resource limitations, the source code needs to be reduced.

Complexity was measured using McCabe's cyclomatic complexity [23]. Cyclomatic complexity is calculated using the number of branches in a module. The max complexity and the average complexity of OLSR-L are lower than those of ROULA. In OLSR-L, the complexity of ROULA was suppressed down by using the existing functions of OLSR.

Depth is the number of modular calls. OLSR-L has a larger max depth than OLSR. Hence, our future work includes reduction of the max depth.

Table 2: Protocol comparisons for OLSR, ROULA and OLSR-L.

Protocol name	OLSR	ROULA	OLSR-L
Software name	OOLSR [7]	ROULA [6]	OLSR-L
Objective	Routing	Localization	Routing & localization
Developer	INRIA	Takenaka et al.	Soga et al.
Platform	ns-2 [20]	OMNeT++ [22]	ns-2
Language	C++	C++	C++
Lines	21,436	12,285	24,419
Avg. complexity	2.27	6.33	2.36
Max complexity	64	98	64
Avg. depth	6	9	9
Max complexity	1.44	2.53	1.57
Methods/class	6.44	64.75	6.59
Statements	5.3	20.9	5.8
Comments (%)	13.8%	29.7%	15.5%

6. Conclusion

We presented a method of integrating localization and routing protocols. OLSR-L enables simultaneous routing and localization in wireless multi-hop networks. We presented a detailed implementation of our OLSR-L protocol. We analyzed the source codes of OLSR and OLSR-L and evaluated the overhead of our OLSR-L protocol. The overhead of OLSR-L was about 40% less than a straightforward integration of the two protocols.

In future work, we need to evaluate the details of how the performance overhead is affected by node density and the frequency of TC packets through our platform. We plan to use laptop PCs to evaluate our OLSR-L and analyze overhead in an actual environment.

References

[1] T. Takenaka, H. Mineno and T. Mizuno, "A Study on Integrated Protocol for Communications and Positioning," *Proc. of International Workshop on Informatics (IWIN)*, pp.17-23, 2008.

[2] Y. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile

Ad Hoc Networks," *Proc. of ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pp.66-75, 1998.

[3] Y. Xu, J. Heidemann and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp.70-84, 2001.

[4] Y. Ko, V. Shankarkumar and N. H. Vaidya, "Medium Access Control Protocols Using Directional Antennas in Ad Hoc Networks," *Proc. of IEEE Computer Communications Societies (INFOCOM)*, pp.13-21, 2000.

[5] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," IETF RFC 3626, 2003.

[6] T. Takenaka, H. Mineno, Y. Tokunaga, N. Miyauchi and T. Mizuno, "Performance Analysis of Optimized Link State Routing-based Localization," *Journal of Information Processing Society of Japan (JIP)*, Vol. 48, No. 9, pp. 3286-3299, 2007.

[7] OOLSR - OLSR Implementation, http://hipercom.inria.fr/OOLSR/.

[8] T. He, C. Huang, B. M. Blum, J. A. Stankovic and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp.81-95, 2003.

[9] A. Savvides, C. Han and M. B. Strivastava, "Dynamic Fine-grained Localization in Adhoc Networks of Sensors," *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp.166-179, 2001.

[10] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. O. Anderson, A. S. Morse and Y. R. Yang, "Localization in Sparse Networks using Sweeps," *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp.110-121, 2006.

[11] J. Liu, Y. Zhang and F. Zhao, "Robust Distributed Node Localization with Error Management," *Proc. of ACM International Symposium on*

Mobile Ad Hoc Networking and Computing (MOBIHOC), pp.250-261, 2006.

[12] D. Nicolescu and B. Nath, "Ad-Hoc Positioning Systems (APS)," *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, pp.2926-2931, 2001.

[13] N. B. Priyantha, H. Balakrishnan, E. Demaine and S. Teller, "Anchor-free Distributed Localization in Sensor Networks," *Technical Report TR-892, MIT LCS*, 2003.

[14] Y. Shang, W. Ruml, Y. Zhang and M. P. J. Fromherz, "Localization from Mere Connectivity," *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pp.201-212, 2003.

[15] M. Li and Y. Liu, "Rendered Path: Range-free Localization in Anisotropic Sensor Networks with Holes," *Proc. of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp.51-62, 2007.

[16] IETF Mobile Ad-hoc Networks (MANET), http://www.ietf.org/html.charters/manet-charter.html.

[17] C. E. Perkins, E. Belding-Royer and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *IETF RFC 3561*, 2003.

[18] D. Johnson, Y. Hu and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," *IETF RFC 4728*, 2007.

[19] R. Ogier, F. Templin and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)," *IETF RFC 3684*, 2004.

[20] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/.

[21] Enterprise Architect (EA) Ver. 7.5, http://www.sparxsystems.jp/ea.htm.

[22] OMNeT++ Descrete Event Simulation System, http://www.omnetpp.org/.

[23] T. J. McCabe, “A Complexity Measure,” *IEEE Transactions on Software Engineering*, Vol. 2, No.4, pp.308-320, 1976.

[24] J. Härri, C. Bonnet and F. Filali, “OLSR and MPR: Mutual Dependences and Performances,” *IFIP International Federation for Information Processing*, Vol. 197, pp.67-71, 2006.

[25] Y. C. Huang, S. Bhatti and D. Parker, “Tuning OLSR,” *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp.1-5, 2006.