

Research Article

Location-Aware Source Routing Protocol for Underwater Acoustic Networks of AUVs

Edward A. Carlson, Pierre-Philippe J. Beaujean, and Edgar An

Department of Ocean and Mechanical Engineering, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA

Correspondence should be addressed to Pierre-Philippe J. Beaujean, pbeaujea@fau.edu

Received 9 May 2012; Revised 13 August 2012; Accepted 6 September 2012

Academic Editor: Paolo Casari

Copyright © 2012 Edward A. Carlson et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Acoustic networks of autonomous underwater vehicles (AUVs) cannot typically rely on protocols intended for terrestrial radio networks. This work describes a new location-aware source routing (LASR) protocol shown to provide superior network performance over two commonly used network protocols—flooding and dynamic source routing (DSR)—in simulation studies of underwater acoustic networks of AUVs. LASR shares some features with DSR but also includes an improved link/route metric and a node tracking system. LASR also replaces DSR's shortest-path routing with the expected transmission count (ETX) metric. This allows LASR to make more informed routing decisions, which greatly increases performance compared to DSR. Provision for a node tracking system is another novel addition: using the time-division multiple access (TDMA) feature of the simulated acoustic modem, LASR includes a tracking system that predicts node locations, so that LASR can proactively respond to topology changes. LASR delivers 2–3 times as many messages as flooding in 72% of the simulated missions and delivers 2–4 times as many messages as DSR in 100% of the missions. In 67% of the simulated missions, LASR delivers messages requiring multiple hops to cross the network with 2–5 times greater reliability than flooding or DSR.

1. Introduction

As autonomous underwater vehicles (AUVs) continue to become less expensive and more capable, they are being deployed in larger groups. As a result, the need to communicate between multiple, mobile underwater systems is growing as well. Underwater communication is best accomplished through the use of acoustic links, and interconnecting multiple underwater vehicles is best accomplished through the use of an acoustic network. Such a network, one using a shared medium and comprising mobile nodes, is called a mobile ad hoc network (MANET). It is difficult to efficiently forward data across a MANET because node mobility means network topology—the overall set of connections between nodes—changes over time. The network must spontaneously organize, learn the topology, and begin routing with a minimum of overhead traffic for route discovery and maintenance. There has been a great deal of attention paid to this problem, but almost exclusively as it applies to wireless radio networks [1–4].

In a network, a node is a communication endpoint able to send and receive data. When two nodes can communicate with one another, they are said to have a link between them. Links can be of varying quality: some links may deliver almost every message without error, others may deliver only a small fraction of the messages sent across them. In shared-medium communications like underwater acoustics, every transmission has exactly one sender but can have one or more receivers.

A message may have to be forwarded across one or more links to intermediate nodes before reaching its intended destination. Routing is the process of choosing the links that will comprise the route the message will follow across the network. A routing protocol is responsible for selecting the route. Most routing protocols collect, manage, and disseminate information about the network in order to function, for example, by monitoring network topology, specifying the next hop of a message, queuing messages awaiting routes, and tracking which messages have already been processed. Unlike in a traditional, wired network,

routing in a mobile ad hoc network (MANET) is complicated by the possibly rapid and unpredictable topological changes caused by movement of the nodes. A given routing protocol is typically intended for a particular type of network, and many have been developed specifically for MANETs [5–9].

Little of the existing research into MANET routing protocols addresses the specific limitations of underwater acoustics [10]. While few MANETs are as drastically low-bandwidth as an underwater acoustic network, many have little bandwidth when compared with wired networks, and some MANET techniques specifically address this by reducing protocol overhead [11–13]. The greater problem is that the existing research assumes—almost without exception—that wireless networks in general, and MANETs in particular, use radio links.

The particular problem is the speed of the nodes compared to the communication latency. Most advanced routing protocols need to propagate topology information throughout the network. The high latency of acoustic links means that the movement of underwater vehicles can change the network topology more quickly than updates can be propagated. This is especially a problem for protocols developed for radio MANETs, which overall assume a much slower rate of topology change compared to communication latency [11–17].

This paper describes the location-aware source routing (LASR) protocol, a network routing protocol specifically designed for use in low-bandwidth, high-latency underwater acoustic networks of mobile nodes. LASR is loosely based on the dynamic source routing (DSR) [9] protocol and is specifically designed for use in underwater acoustic networks where the topology changes frequently. The results presented here show that, in simulated underwater acoustic networks of AUVs, LASR outperforms both blind flooding and DSR in throughput and packet delivery ratio. Note that LASR is intended for use in missions where vehicle movement dominates energy consumption, so that it maximizes successful communication rather than energy conservation. A performance comparison between protocols in terms of energy consumption is not the focus of this publication, but it is an important future study.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. The new LASR protocol is described in Section 3. Specifics of handling routes and messages are covered in Section 4. Section 5 presents some results of LASR in a simulated underwater network. Section 6 summarizes our conclusions.

2. Literature Review and Background

2.1. Medium Access Control. Radio and acoustics are both shared medium techniques: multiple senders and receivers use the same medium (e.g., the water of the ocean) and there must be some sort of medium access control (MAC) to keep them from all “talking at once”. Inherent in shared-medium systems is the problem of collision—the interference among multiple, simultaneously-received signals. A large number of

MAC protocols have been developed, some better suited to mobile underwater acoustic use than others [10, 18–24].

Time-division multiple access (TDMA) divides the medium into time-slots [4]. Each node may use the entire bandwidth, but may only transmit according to a given schedule. LASR must use TDMA as its MAC protocol. The TDMA transmit-time information is what allows LASR to collect implicit time-of-flight information for the nodes in the network and is crucial for effective use of its tracking system.

2.2. Blind Flooding. Blind flooding is a network broadcasting protocol [4], and the simplest of the flooding protocols. It delivers its messages to every node in the network, and does so without knowledge of the topology. The basic operation is simple: the first time a node receives a given message, the node automatically rebroadcasts it. Because blind flooding does not require the topology to be known, many of the more-sophisticated routing protocols employ it before routes are known, for example, during route discovery. Blind flooding’s advantages include operation without topological information and low end-to-end delay. The main disadvantage of blind flooding is that it can produce a significant amount of unnecessary traffic, especially as the size of the network increases.

2.3. Shortest-Path Routing. Flooding delivers a message by network broadcast, and every node in the network receives the message. This is very inefficient when the destination is a single node. An alternative is shortest-path routing, where a message follows the path with the fewest hops. This is much more efficient: rather than every node in the network forwarding the message to all its neighbors by broadcast, each node along the shortest path forwards the message to the next hop by unicast. However, this makes it necessary for the network nodes to have at least partial knowledge of the network topology. It is also important to avoid routing loops, which occur when mismatches in topology information across several nodes cause messages to be routed in circles.

Examples of shortest-path routing include the Destination-Sequenced Distance Vector (DSDV) protocol [5], Ad hoc On-demand Distance Vector (AODV) [6], Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [8], and the Temporally-Ordered Routing Algorithm (TORA) [7]. Of particular interest here is the Dynamic Source Routing (DSR) protocol [9], a reactive protocol which, depending on the implementation, uses either distance-vector or link-state routing. In source routing, the entire route to the destination is determined by the originator (the source) and is carried along with the message. Routes are discovered as needed via a route-request/route-reply process, and there are no periodic updates.

2.4. Delay-Tolerant Routing. In some networks, there may never be an end-to-end connection. Instead, individual mobile nodes must hold data until a forwarding opportunity arises [25]. For example, a protocol can exploit vehicles’ non-random mobility patterns to improve routing performance

[26]. These routing techniques are not necessarily suitable to the cooperating-AUVs problem. When cooperating, the nodes will likely actively work to stay connected, that is, each node will maneuver such that it always stays within range of the network. More importantly, certain types of data do not need to be delivered immediately and can tolerate significant delay in their delivery, but when cooperating on short time-scales, some communication is very likely to be time-sensitive and delivery cannot wait long periods for an opportune vehicle motion to put it in range.

2.5. Position-Based and Location-Aware Routing. A routing protocol spends most of its time determining and tracking the network topology. With communication technologies such as radio and acoustics, which links are available largely depends on the distance between the various nodes. Some routing protocols use knowledge of the location of network nodes to provide or augment topology information. These are known as location-aware or position-based protocols.

Routing by absolute geographical location typically employs a locating service that is queried by nodes to look-up the current location of a destination node. Messages are routed to the neighbor that is geographically nearest to the destination. Routing by relative location typically requires both relative location (e.g., range and bearing) as well as traditional topology information. LASR routes by relative location.

A protocol similar to LASR is [27], which also estimates range from one-way time-of-flight using TDMA and uses it to discover network topology for routing via DSR. However, it includes pseudonoise probe patterns as a part of each frame because localization is of primary importance in that system. The network supports only very few nodes and the overall communication rate is extremely low.

3. New Protocol

3.1. Overview of the Specific Aspects of LASR. The new LASR protocol has been specifically designed to address the problems of routing in low-bandwidth, high-latency underwater acoustic networks of mobile nodes. It is loosely based on the DSR [9] protocol. Like DSR, LASR is a self-organizing, infrastructureless, distributed protocol. It learns and maintains only those routes that are in use.

LASR uses the source route principally as a means to communicate topology information. Each intermediate node updates the source route in every message it forwards, applying the route most likely to require the fewest transmissions (which does not necessarily correspond to the fewest hops) to reach the destination. Every message transmission is therefore routed according to the most current topological knowledge, rather than DSR's approach which routes according to the topological knowledge at the time the message was originated.

3.2. Assumptions. The LASR protocol is designed for small underwater networks using low-speed acoustic links. The

network should not contain more than 20–30 nodes, a reasonable assumption given typical multiple-AUV operations such as [28]. This network size limitation is due in lesser part to the source route header overhead in each message. The size of the source route grows linearly with the length of the longest path through the network. In greater part, this assumption is due to LASR's required use of TDMA, which does not scale well into large networks.

Nodes may move at any time and in any direction. The only restriction on node motion is that speeds should be in the range 0–3 m/s; this speed range is typical for most current AUVs. This assumption is necessary to limit the rate at which node motion can change the network topology.

All nodes must use identical LASR algorithms, and all must fully participate in the protocol, including forwarding the messages of others.

Every node must have accurate timekeeping, for example, by means of a low-drift clock. No two node clocks may differ by more than 50 milliseconds throughout a mission, although this network time may differ from true time by any amount. This is necessary for TDMA window timing. Equipped with the optional time synchronization feature, the FAU Dual Purpose Acoustic Modem (DPAM) fits this requirement over 8 hours using low-drift clocks [29]. Also, prior work [30] has shown that for LASR, this is the minimum timekeeping precision necessary to preserve the accuracy of the time-of-flight range estimates based on TDMA window timing.

The communication link endpoints should be identical acoustic modems, and these modems should be effectively omnidirectional. They must support overhearing—the reception of messages not specifically addressed to them. Overhearing is an important source of topology information. To allow the tracking system to function, each modem must report the time at which any incoming transmission is detected, regardless of whether or not the transmission can be successfully decoded. The detection time reporting must be accurate to within 30 milliseconds. As with the timekeeping precision, this reporting precision has been shown [30] to be the minimum necessary for time-of-flight range estimate accuracy.

LASR's implementation of ETX assumes that network links are bidirectional (acoustic modem links are traditionally bidirectional, albeit half-duplex) and symmetrical, meaning packets can cross the link between any pair of nodes in either direction with equal probability of success. In practice, the links are not perfectly symmetrical, but symmetry is a fair assumption so long as the transducer is assumed omnidirectional and the environmental conditions (and range between nodes) do not change significantly between two transmissions. The development of a nonsymmetrical and unidirectional version of LASR is beyond the scope of this article, but constitutes a future key for development of LASR. The links are assumed to be through a shared medium. The network must use TDMA as the MAC protocol so that implicit time-of-flight range estimate is possible.

The ETX implementation also assumes that a medium model exists for the modem, which can provide a reasonably accurate estimate of the frame-error rate (FER) between

two modems given the distance between them. The FER is the probability that a given transmission (a frame) on the link will be received in error. All nodes must use identical medium models and the FER estimate must be deterministic: every use of the model at every node must return the same FER for a given range. Note that the FER model includes other input parameters (sea state, ambient noise, water depth, bottom type . . .). A complete list is provided in [31]. The FER model used in the simulation was developed from field data [32]. For simplicity, the study assumes that every input parameter is constant, with the exception of range. These other parameters impact the FER, thus the LASR performance. At fixed range, the authors showed in [33] that the LASR performance drops with ambient noise and sea state, as the FER increases with these two parameters.

A range-only tracking system is assumed to be available at each node. Regular measurements of the distance from the local node to each of the various other nodes within detection range will be available from a combination of the modem's transmission detection and TDMA window timing. The tracking system must use those time-of-flight based range measurements to predict the current location of those nodes relative to the local node. Prior work [30] has shown that the tracking system must predict relative node position to within 200 m of the true relative node position. If the estimated prediction error exceeds this amount for a given node, the tracking system must cease reporting the predicted position of that node.

3.3. Link Metric. The expected transmission count (ETX) [34] estimates the number of times a node will have to transmit a message before it successfully receives an acknowledgment. The ETX of a route is simply the sum of the ETXs of each link in the route, and any two ETX route metrics are directly comparable. The ETX is calculated from a link's FER. The technique described in [34] to calculate the ETX uses probe messages sent periodically across a link—once a sufficient number of probe messages have been sent, it is possible to estimate the link's FER, and then to calculate the ETX.

In a MANET however, node motion can cause considerable variation in link quality over short time scales. This is a problem because, while ETX outperforms hop-count in a static network, hop-count can react more quickly to link changes and outperforms ETX when nodes are moving [35].

LASR uses expected transmission count (ETX), but overcomes this mobile-node measurement-delay problem by calculating the delivery ratio directly from the FER estimated by the medium model. LASR assumes symmetric links, so the probability that a message and its acknowledgement will cross a link successfully is $(1 - \text{FER})^2$, making the equation for ETX:

$$\text{ETX} = \frac{1}{(1 - \text{FER})^2}. \quad (1)$$

How LASR handles the ETX information is described in Appendix B.

3.4. Tracking System. Neighborhood topology is predicted by the tracking system based on information from both implicit and explicit communication. Combining the time-of-detection information from the modem with the current TDMA state provides both an estimated time-of-flight and the identity of the transmitter. The range to the transmitter can then be estimated using the medium model.

A series of range estimates to other nodes, coupled with knowledge of a node's own motion, can form the basis for localization and tracking of the other nodes. When combined with minimal information from the other nodes about their ranges to each other, the relative, progressive location of the other nodes can usually be uniquely determined to some accuracy.

A tracking system was not implemented as part of this work. The behavior of the tracking system was simulated based on the minimum established performance requirements. A recursive state-estimation filter, such as a particle filter, is expected to be able to localize and track some or all of the network nodes, depending on the amount of information available about each node. The more information that is available about another node, the more accurate tracking and location prediction can be. Even a low-order motion model (e.g., maximum, minimum, and typical speed and turning rate) will help constrain tracking and prediction uncertainty. A behavior model providing knowledge of the types of behaviors the node may exhibit (e.g., lawn mowing, line-following or hovering) can further reduce uncertainty.

Information for tracking can be characterized as either explicit or implicit. Explicit information is carried as overhead in network messages. The LASR source routes, for example, carry explicit link range information. Implicit information is communicated without overhead, simply by the act of communicating. An example of implicit information is the time-of-flight measured when a message is received.

Some modems, such as the FAU DPAM [31], preface each packet with a known sequence of symbols. The optional time synchronization feature of the FAU DPAM is used for TDMA communications and tracking [29]. This detection sequence is used by the receiver to identify an incoming transmission because, unlike the coded variable data in a message, the symbols in the detection sequence are known a priori, making them substantially easier to identify, even in very weak signals. It is frequently possible to correctly identify the detection sequence in transmissions from ranges far beyond the range at which there is sufficient signal to successfully decode the variable data.

Under such modems, incoming transmissions fall into three categories: strong enough to decode (providing implicit range and explicit data), strong enough to detect but too weak to decode (providing implicit range only), and too weak to detect (providing nothing). Because the detection sequence can be reliably identified even across a link with an extremely high FER, the second category includes transmissions from nodes far beyond the useful explicit range of the modem. A comparison of implicit and explicit data is shown in Figure 1.

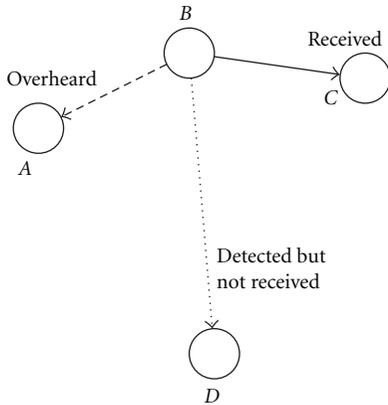


FIGURE 1: Implicit versus explicit data. Node B transmits a message for node C. Nodes A and C each detect and receive it (the message is intended for C, but A has overheard it). The detection provides an implicit range estimate to node B; the reception provides all of the explicit routing information contained in the message (e.g., in the source route). Node D detects but does not receive the message, thus D gains an implicit range estimate to node B but gets none of the explicit data.

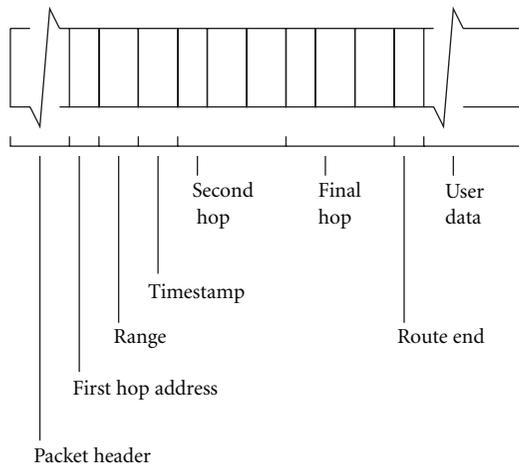


FIGURE 2: The LASR header is a variable-size stack of bits. This shows the source route portion of a three-hop route.

3.5. *LASR Packet Structure.* Each LASR packet contains one or more messages. A message can contain user data or protocol data. A user-data message contains a source-route in addition to the user data. There are several protocol message types; these are described in Appendix A.

Packets are small in a typical acoustic network, typically on the order of tens to hundreds of bytes only. This makes header overhead very expensive as even a small header can represent a large fraction of a packet. LASR uses a different header structure than DSR in order to reduce the size of the header as much as possible. LASR's header structure is shown in Figure 2. The number of bits added to the header by a given layer can change from message to message. To accommodate this, the header is implemented as a stack of bits.

A source route is structured as a series of triples followed by an end marker. Each triple is a hop in the route starting at the originator and ending one hop before the destination. A triple comprises the address of the node, the best-available estimate of the range from the node to the next hop (or the destination) and the timestamp of the range estimate. Both the range and its timestamp are quantized to conserve space in the header, see [30] for details on the quantization. The route end is the special network address zero, which is never a valid address. The network addresses are represented as the smallest number of bits that can represent the number of nodes in the network, plus one for the special zero address. For example, a 16 node network would require 17 unique addresses and would therefore require 5-bit addresses.

3.6. *Data Structures.* The protocol maintains several data structures at each node to maintain knowledge of the topology and the state of the various messages that the node must handle. The pending buffer holds messages awaiting transmission. The link cache stores network topology as information on links between pairs of nodes. The buffers are limited only by available memory. Tables such as the route-discovery, route-reply and route-request tables are used to limit the transmission of unnecessary or redundant protocol messages. The route-advice table prioritizes topology propagation messages. The data structures are described in Appendix C.

4. Management of Routes and Messages

4.1. *Route Discovery.* LASR is source-routed: when a message is sent across the network, the message must carry its route as part of its data. If a node, either the originator or an intermediate, has a message to transmit but is not able to find a suitable route in the data provided by its link cache and tracking system, it must defer sending the message and initiate route discovery by sending a route-request message.

The originated route-request message contains only an ID assigned by the originator. This ID, combined with the address of the originator, allows the route request to be uniquely identified in the network. A circular ID is used to conserve space in the message.

Before originating a route-request message to a given destination, a node must first check its route-discovery table. If a route request for that destination is already outstanding, the route request is not sent. This prevents a node from overloading the network with redundant route requests.

The route-request message is propagated through the network by flooding, as flooding is an efficient way to reach all the nodes in the network when the network topology is not known. When a node receives a route-request message, it is handled in one of three ways.

- (1) If the route-request's destination is the local node, it checks its route-reply table to determine if it should respond with a route-reply message. If it should, a route reply is originated with the destination set to be the originator of the route request. If not, the route request is dropped.

- (2) If the local node is not the destination and the route request has already been forwarded (if its unique ID appears in the route-request table), the route request is dropped.
- (3) Otherwise, the route request is enqueued for forwarding.

Whenever a route-request message is forwarded, it contains the best-available route from the originator to the forwarding node. This route is added to the link cache of each intermediate node and the destination node. In this way, route requests propagate topology information through the network, eventually providing the destination with a route back to the originator.

Due to the nature of flooding, a given route-request message can be duplicated many times as it propagates through the network and different copies can arrive at the destination at different times. When sending, LASR always chooses the best route currently available; if the same route is always best over multiple receptions of the same route request, multiple route replies would be sent, all containing the same data. To avoid such redundancy, a second or further route-reply message is originated only if a better route is available than the previous route reply to the same route request. This is determined by consulting the route-reply table (see Appendix C).

4.2. Route Maintenance. Route maintenance uses the implicit data from detected transmissions and the explicit data from received messages. A detected transmission provides an update to the tracking system. If a received message contains topology information, for example, the route in a route-request or route-reply message, this is used to update the link cache (see Appendix C). Received messages do not need to be for the local node to be useful. Overheard messages are an important source of topological information.

Received explicit topological information, whether from overheard messages or not, is processed by the link cache. Newer data are used to update the link cache. Older received data are used to update the route-advice table to help prioritize the sending of topology updates.

4.3. Send Algorithm and Route Handling. LASR sends a packet every time the local TDMA time-slot is open. It chooses the next message based from its fair-queue of user-data and protocol messages. If there are no pending messages, LASR will send a route-advice protocol message. This results in additional energy consumption, justified by the fact that LASR is designed to maximize successful communication rather than energy conservation. Again, LASR is intended for use in AUVs where vehicle movement dominates power consumption.

In DSR, a source route does not change except in special circumstances, for example, when a message is salvaged following the detection of a failed link. However, in an underwater acoustic network, the topology may change extremely quickly compared to the network latency, and a source route that was valid when a message was originated may become stale before the message has reached its destination.

To overcome this, LASR recomputes the route at every hop: the source routes serve mainly to disseminate topology information. When a message carrying a route is received, its route is read, the link cache is updated with any newer data, and older data are noted for future advising. The message then is queued without its route but with sufficient information for LASR to find for it a replacement route.

Immediately before a routed message is transmitted, it receives a completely new route from the cache. Timestamped link information ensures that a message always departs with the most up-to-date route information possible. When searching for a route, the cache first gets an updated prediction of the network topology from the tracking system. It then searches the updated network graph using Dijkstra's algorithm [36]. An example of an intermediate node updating a route is shown in Figure 3.

If a route cannot be found for a message, the resulting behavior depends on the message type. A user-data message remains in the queue, and route discovery begins to find a route to the destination. A route-request or a route-reply message with no route is silently discarded to prevent stale information from propagating through the network.

4.4. Receive Algorithm. The first step in processing a received message is to extract any topological information and use it to update the link cache. The link cache processes the data and updates the route-advice table as needed. The implicit data from the detection of the transmission is handled separately and is passed directly to the tracking system.

The next step is to discard overheard messages. Though a valuable source of topology data, nontopology-related message contents must not otherwise be processed or forwarded.

The remaining messages are those intended for the local node, either as an intermediate or as the destination. All of these messages update the acknowledgment-equivalence table to ensure they will be acknowledged properly. A message whose destination is the local node is handled and discarded. User data are passed up to the next-highest layer in the stack. Protocol data are processed, which may include originating a route-reply message in response to a route-request message. Other messages intended for the local node, but not destined there, are to be forwarded. These messages are passed to the pending buffer to be enqueued for transmission.

5. New Protocol Results

This section discusses the simulation results for the new LASR protocol for underwater acoustic networks. The new protocol has been tested under a variety of simulated underwater missions, each in several operational scenarios. For comparison purposes, these tests are also conducted with the flooding and DSR protocols. The results demonstrate that the LASR protocol provides improved network communication performance compared to flooding and DSR.

DSR is run without any of its optional features enabled as initial work demonstrated that each of the optional

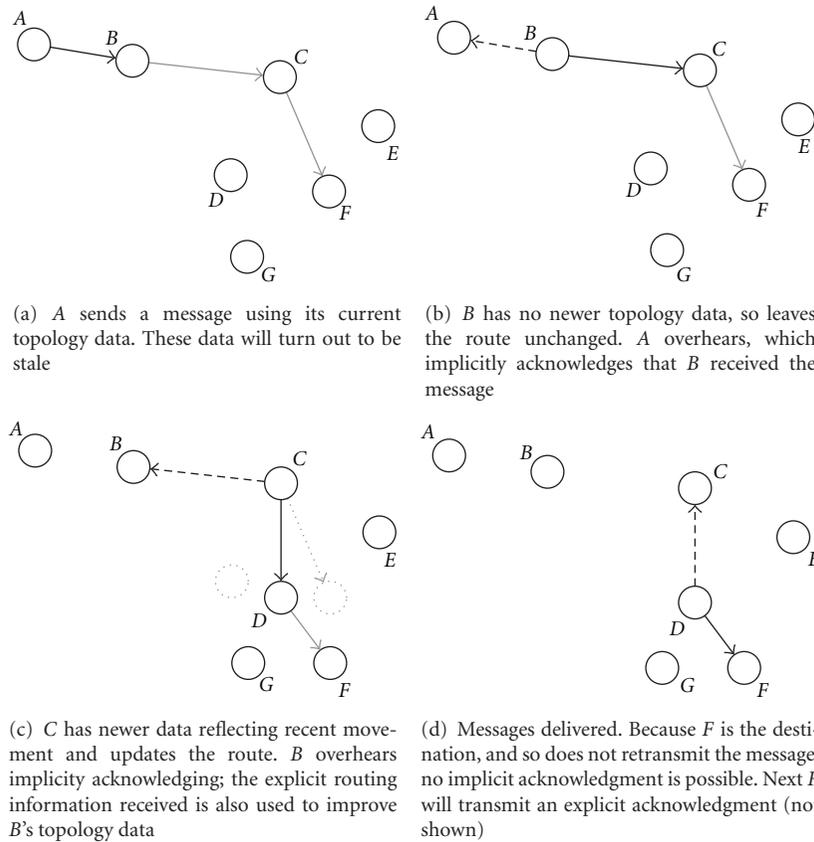


FIGURE 3: An example of LASR's route handling, where an inaccurate route is automatically corrected. For clarity, only the overheard message at the previous transmitter is shown. In actuality, some or all of the other nodes would detect or overhear each transmission, improving the topology data in those other nodes.

features negatively impacted DSR performance in an acoustic network. Three configurations of the LASR protocol are tested, which differ in number of retries and time spent waiting for acknowledgment. The LASR acknowledgment guarantee means that a receiver will acknowledge receipt within the specified time; this controls how much delay is introduced when a message, or its acknowledgment, fails to cross a link. The acknowledgment period is a multiple of the TDMA frame duration, to give each possible receiver some number of opportunities to transmit an acknowledgment (either implicit or explicit). The three LASR configurations are as follows.

- LASR-0+3: no retries, unacknowledged messages are never retransmitted. However, receivers are still obligated to send an acknowledgment within three TDMA frames.
- LASR-2+3: two retries, acknowledgment required within three TDMA frames.
- LASR-2+6: two retries, acknowledgment required within six TDMA frames.

5.1. Scenarios. Every scenario uses 16 vehicles, which is selected as an average network size for LASR. The parameters are exhaustively combined, with each combination defining

a scene. Each scenario contains all scenes. Due to the stochastic nature of the communication model, each scene is run 20 times and the results averaged to smooth the performance results. The authors limit the study to 20 runs per scene due to computation time. This paper shows only a small fraction of the extensive simulation results; full results are available in [33].

The vehicles originate messages containing arbitrary data and send the messages to randomly chosen destinations. Every node transmits at every opportunity. If no message is ready to be sent when the node's transmission time-slot opened, a new message is generated by either the application layer or a protocol layer. Here, we assume that there is always at least one packet in the buffer of each transmitting node, with the objective to discover the maximum possible throughput (in practice, the LASR performance is also related to the mean packet generation rate). The random selection of the destination node is according to a uniform distribution: each node (except the originating node itself) has an equal probability of being selected as the destination. This means the network had full utilization at all times: there is never a TDMA time-slot that passes without a transmission, either to forward a protocol or user-data message or to originate a new user-data message.

Each vehicle is equipped with an FAU dual-purpose acoustic modem (DPAM). Every modem uses Frequency-Hopped, M-ary Frequency Shift Keying (FH-MFSK) modulation with convolutional coding [31]. Packets are fixed-size, carrying 32 source bytes each. Each transmission takes 2.65 s and has a guard time of 2.35 s, for a total TDMA time-slot duration of 5 s.

The FER is determined at run-time using the FAU DPAM medium model [32], a stochastic model derived from the Nakagami model, which considers channel geometry, fading characteristics, background noise, bottom type, modulation, and error coding. The network simulation tool was developed at FAU and is described in detail in [32, 33]. The best-case conditions for communication are when Nakagami- m is 2.0 and noise PSD is 55 dB/ $\sqrt{\text{Hz}}$, the worst-case when Nakagami- m is 1.5 and noise PSD is 65 dB/ $\sqrt{\text{Hz}}$.

5.2. Graph Methodology. There are two graphed network metrics: messages-delivered versus range and message success ratio. These metrics measure different aspects of the network performance: messages-delivered measures throughput, success ratio measures reliability. Note that every message size is fixed to 32 bytes, so that the message-based analysis can be easily converted to a byte-based analysis. The graphs count as delivered or successful only unique user-data messages that reach the intended destination. User-data messages which never reached their destination, duplicate user-data messages received at the destination and protocol-only messages are not counted as delivered or successful. The uncounted messages are the protocol's message overhead.

The messages-delivered graphs show the total number of originated user-data messages that are successfully delivered versus the distance between the originating node and the delivery node at the time of message origination. It does not consider protocol messages (e.g., route requests and route replies) and counts only messages containing user data. The successful delivery of a protocol message is not counted towards messages-delivered, so in general, the greater a protocol's message overhead, the lower its messages-delivered count. These graphs provide a measure of throughput versus range. The messages-delivered graphs should be consulted if throughput is of primary importance, especially if the loss of packets can be tolerated.

The delivery success ratio graphs show the ratio of user-data messages successfully delivered to user-data messages originated. Again, only user-data messages are considered. This ratio is graphed versus the same distances as the messages-delivered graphs. Messages still in the network when the simulation ends are considered lost, and so reduce the success ratio. This metric provides a measure of reliability versus range, that is, the probability that a user-data message sent over a given range will eventually be delivered. The success ratio graphs should be consulted if assured delivery is of primary importance, especially if a loss of throughput can be tolerated.

Note that it is not valid to assume that delivering a greater volume of messages implies that messages are also delivered with greater reliability, or vice versa. They are

commonly *inversely* related because increasing the delivery reliability generally requires increasing protocol overhead, which reduces the total number of messages that can be delivered for a given network bandwidth. A protocol with little overhead may be able to send a tremendous number of user-data messages, losing most but still delivering a large number. On the other hand, a protocol with large overhead may be able to send only a few user-data messages, but may deliver almost all of them.

These metrics both count messages, not bytes. Larger packets would likely increase byte throughput but are also likely slightly decrease both messages-delivered and message success ratio because larger packets would take longer to transmit, thus lengthening the TDMA window, and would probably increase the FER of the links.

5.3. 16-Vehicle Line. The line scenario simulates a variety of possible real-life missions where the vehicles are arranged in a line, possibly with active station-keeping. The 16-vehicle line-scenario results are shown in Figures 4 and 5. The data in each figure are obtained using a specific average FER measured between any two adjacent nodes (noted FER_{NN}). The distance between neighboring nodes is fixed to provide a specific FER_{NN} for each scenario, these FER_{NN} values are 0.01 and 0.40. Under best-case communication conditions, $FER_{NN} = 0.01$ is achieved when neighbors are separated by 1875 m and $FER_{NN} = 0.4$ when separated by 3012 m. Under worst-case communication conditions, $FER_{NN} = 0.01$ is achieved when neighbors are separated by 492 m and $FER_{NN} = 0.4$ when separated by 1315 m.

LASR proves superior in total number of messages delivered, though not by a substantial margin except for $FER_{NN} = 0.01$ under the worst-case conditions, shown in Figure 4(c).

Generally, flooding in the line configuration benefits under poor communication conditions because the destination node, which does not retransmit, is often able to halt propagation of a message down the line in one direction. In contrast, flooding actually suffers under good communication conditions. Virtually every originated message reaches every node, as evidenced by its excellent delivery success ratio shown in Figure 4(c). However this also means that every node rebroadcast almost every message, producing message overhead that leaves the network so busy that very few messages can be originated, as shown by the small number of messages delivered shown in Figure 4(c).

Delivered message counts from DSR are consistently bad across the line scenarios due to poor handling of link failure in the high-latency acoustic network. The loss of a message forces route-error propagation in a congested network, which triggers numerous route-request/route-reply cycles, greatly increasing protocol packet overhead.

Three of the protocols do not retry unacknowledged transmissions: flooding which does not support retries, DSR which is found to perform spectacularly poorly when retries are enabled and so is used with retries disabled, and LASR configuration LASR-0+3. Delivery success for the 0.40 case is shown in Figures 5(a) and 5(c). This

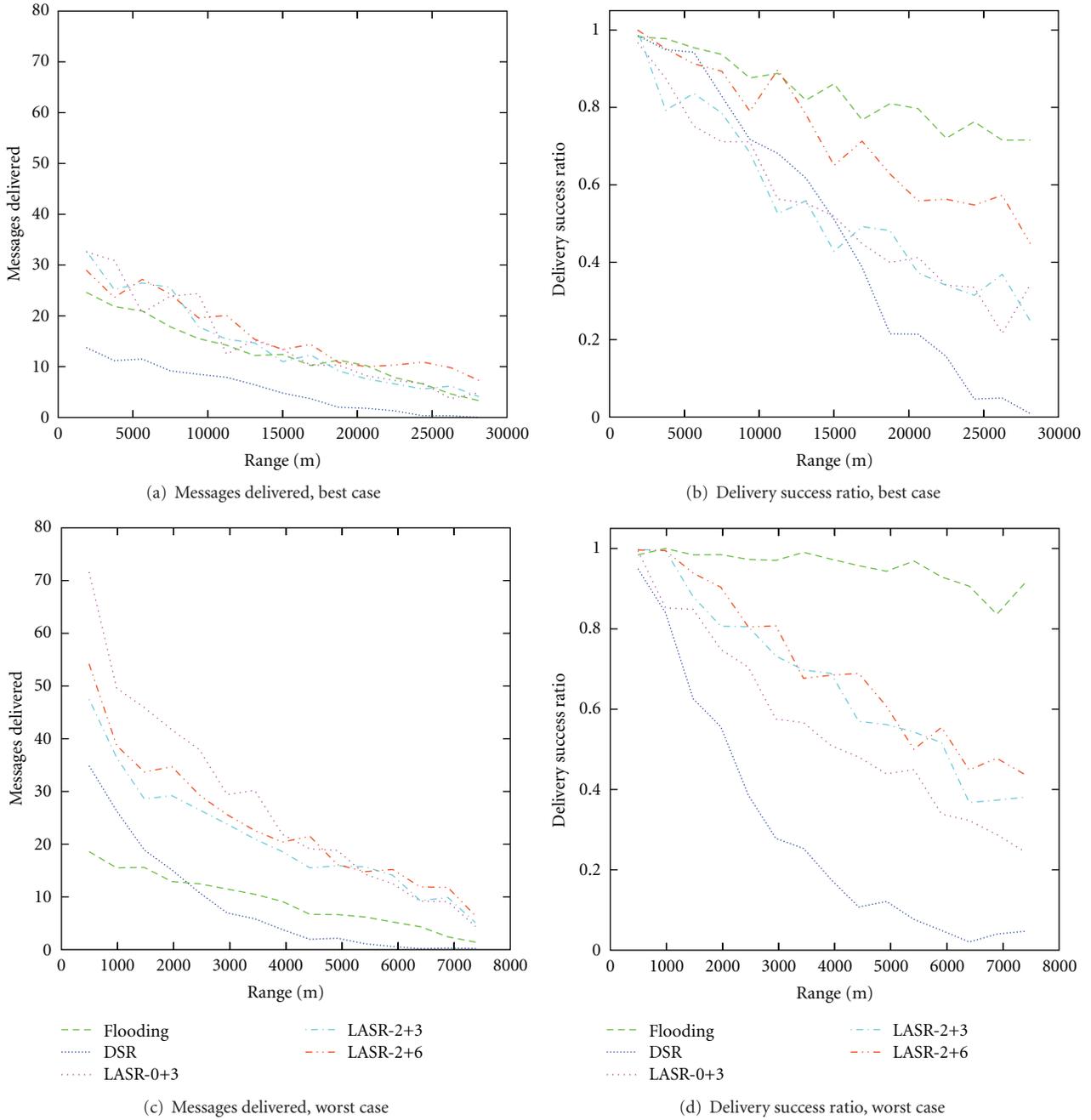


FIGURE 4: Messages delivered and message success: 16-vehicle line, $FER_{NN} = 0.01$.

shows that the paths used by each have very similar loss patterns: under any protocol, a message has a roughly equal probability of crossing a given distance in the network. This is as expected—in the 0.40 configuration of the line scenario, there is effectively only one path between any two nodes.

The LASR-2+3 and LASR-2+6 configurations do support retries. These two have similar delivery success across the scenario configurations, degrading only slightly as the communication conditions worsens. Using retries trades network latency and message origination rate for delivery success.

As communication becomes more difficult, the retrying LASR configurations suffer a greater decrease in number of delivered messages over short ranges where the nonretrying protocols profit from their greater origination rate. At the same time, retries become increasingly important to successful delivery over longer ranges, where the retrying protocols consistently maintain greater delivery success. Waiting longer for a retry (LASR-2+6 waiting six frames versus LASR-2+3 waiting three) also improves performance slightly as it requires fewer explicit acknowledgments, reducing protocol packet overhead.

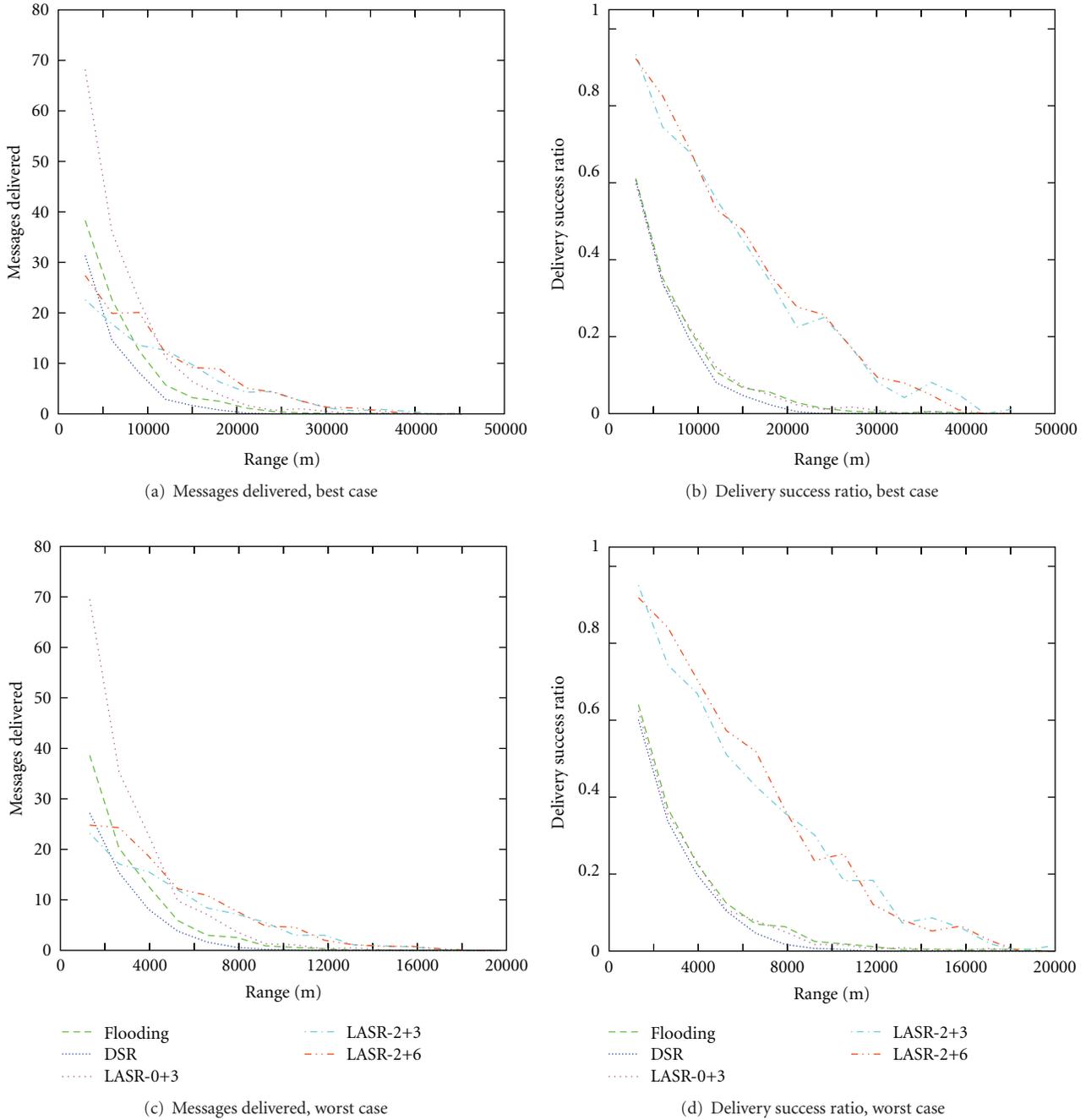


FIGURE 5: Messages delivered and message success: 16-vehicle line, $FER_{NN} = 0.40$.

5.4. 16-Vehicle Spiral. The spiral mission is a simulation of a search (or survey) mission in shallow waters. Each vehicle follows its own spiral path out from the center. On reaching a predetermined distance from the center, each vehicle follows another spiral back to the center. All vehicles begin and end the mission at the central launch point, and all vehicles remain at the same fixed depth throughout the mission. The spiral paths are calculated to maintain a fixed distance between any two paths, for example, to simulate complete coverage while bottom-mapping with side-scan sonar. All vehicles move at 1.5 m/s.

The FER_{NN} increases constantly as the vehicles spiral outwards from the central point, reaching a maximum at the turn-back radius, then decreases constantly as the vehicles spiral back to the center. The maximum FER_{NN} is noted as $FER_{NN,max}$. The spiral scenarios are identified by $FER_{NN,max}$, rather than by turn-back radius.

The message-success graphs for the spiral are averaged at 500-meter increments. This represents a change from the earlier graphs—nodes in the line and the grid topologies are stationary, remaining at a small number of fixed distances. The movement of the nodes in the spiral mission means

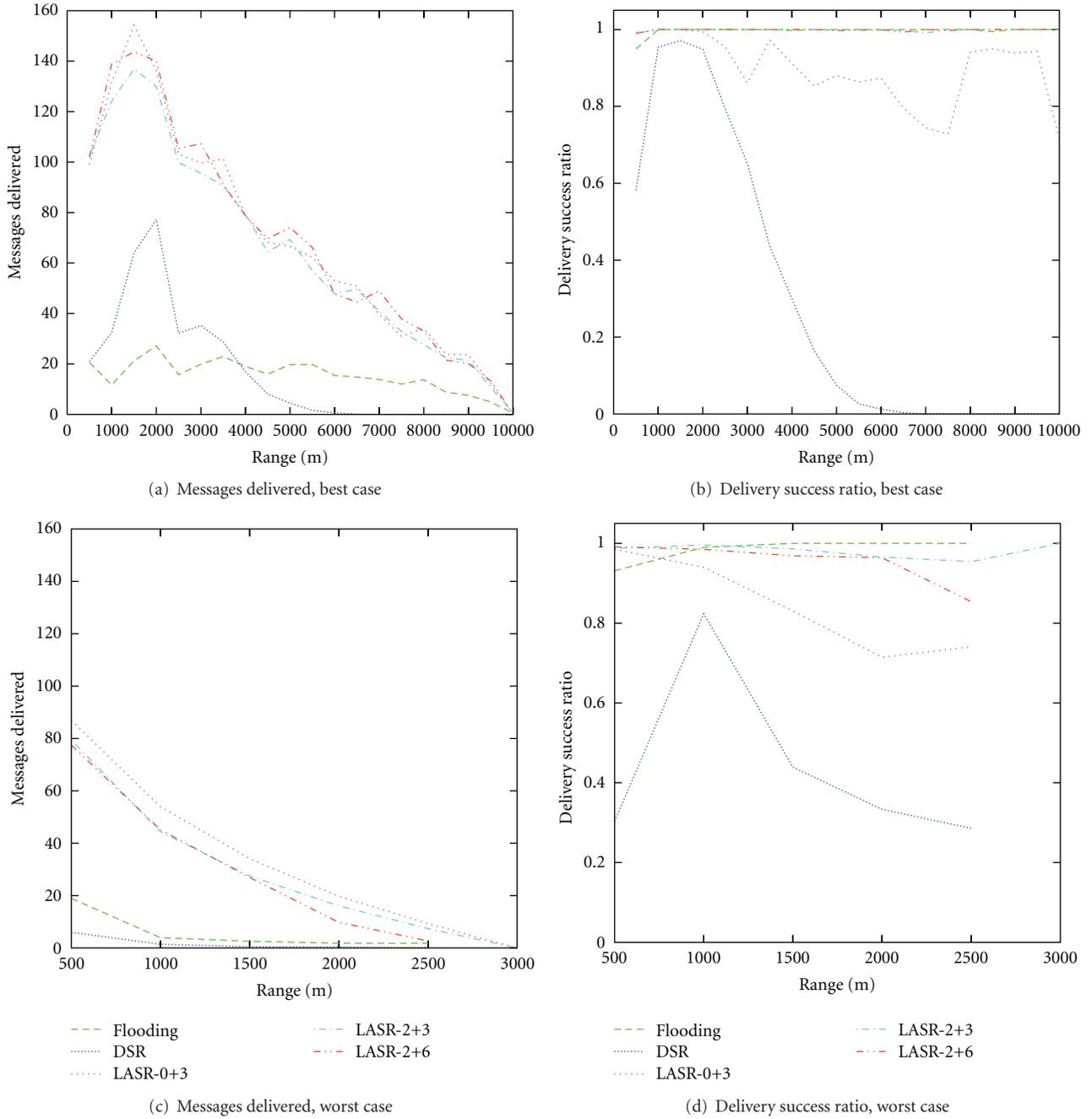


FIGURE 6: Messages delivered and message success: 16-vehicle spiral, $FER_{N,max} = 0.01$.

that source-destination distance at message origination is constantly changing. The 16-vehicle spiral results are shown in Figures 6 and 7.

The large peaks in the message-delivery graphs are caused by the motion of the nodes. As the nodes moved, there are certain ranges which occur more frequently than others. For example, as the spiral expands, a given range is encountered first in a link directly across the spiral, then to successively closer vehicles until reaching the neighboring node. This process is repeated as the vehicles return to the launch point. The shortest ranges are passed quickly while the longer

ranges do not occur until late in the spiral, and then suffer from the problems of increased path length and greater message loss. Medium ranges show pronounced peaks in the message-delivery graphs. Those ranges are present for a longer period of time than the shorter ranges. They also have lower network latency and smaller probability of message loss due to shorter paths than the longer ranges do.

At the beginning of the mission, all of the vehicles are very close together and the network is fully-connected: every vehicle can communicate directly with every other vehicle. DSR route discovery learns this topology and routes each

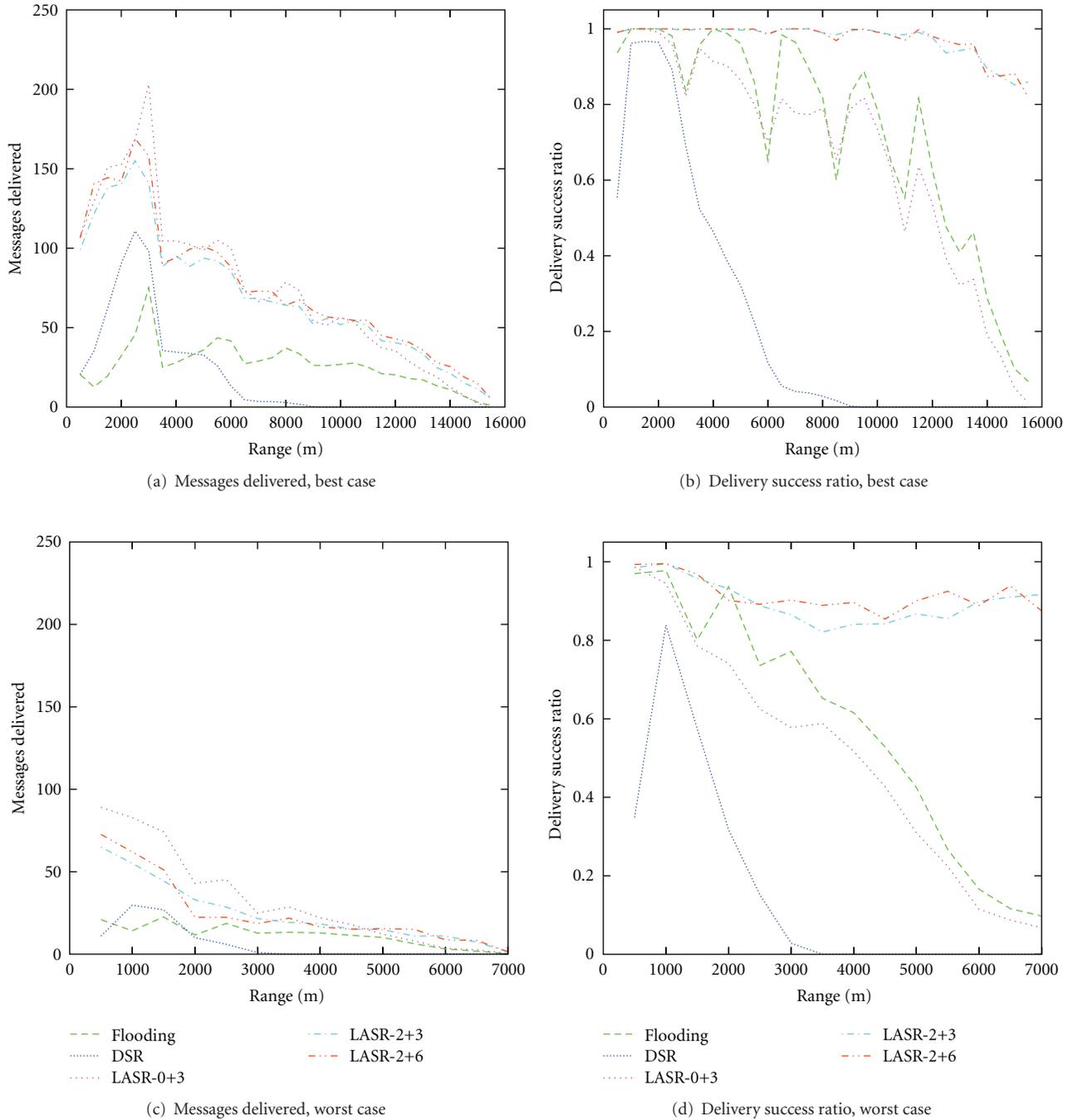


FIGURE 7: Messages delivered and message success: 16-vehicle spiral, $FER_{NN,max} = 0.40$.

message directly to its destination. During the expansion period of the mission, the longer links become untenable as the FER sharply increases and DSR begins to lose the messages sent across those links. The lost messages are never acknowledged, so DSR route maintenance marks the links as broken; any subsequent communication between distant links requires multiple hops. Since DSR must wait for an acknowledgment before it gives up on a link, the multiple hops introduce a delay between the actual failure of the link and the removal of the link from the vehicles' link caches.

This delay is responsible for the poor performance of DSR over mid- and long-ranges.

LASR is able to learn the initial, fully-connected network topology within the first TDMA frame. As the vehicles begin to spread out, LASR is able to consistently predict good network routes. The tracking system is kept updated by the constant distribution of network topology information in message source routes and by the regular estimation of node location by detected incoming packet headers. Unlike DSR, LASR does not suffer a lag in its topology information.

Flooding suffers a reverse problem as compared with DSR. When vehicles are close, many receive each transmission, causing the network to become saturated with messages that need to be forwarded. This means that when many reliable links are available, many fewer new messages can be originated, thus there are fewer messages that can be delivered. When vehicles are distant, flooding becomes more efficient. Message loss and a variation on the destination-blocking effect seen in the line scenarios help reduce the number of nodes that have to needlessly forward a message. This explains why flooding maintains largely consistent message delivery counts over all ranges.

5.5. LASR Benefits and Limitations. LASR provides an overall improvement in message delivery volume and reliability. Its tracking system gives it the ability to predict network topology with sufficient accuracy for effective routing. The use of the ETX metric means that LASR consistently selects good routes. The acknowledgment guarantee means that LASR can react quickly when transmission fails. In mobile networks, the guarantee also helps ensure that LASR completes link-level acknowledgment quickly, before network topology can change so dramatically that acknowledgment become impossible. The most significant drawback to the LASR protocol is its large header-overhead, which reduces the amount of data that can be contained in a given message. However, LASR have a relatively small message-overhead (most messages are user-data messages rather than protocol messages). Message overhead is a greater hindrance to message delivery in the simulated networks. The simulations demonstrate that LASR provides a better balance between message reliability and message delivery volume than either flooding or DSR.

5.6. LASR Performance Summary. In the context of the results presented in this paper, LASR delivers up to 2-3 times as many messages as flooding in 50% of the simulated missions, and delivers up to 2-4 times as many messages as DSR in 100% of the missions. In 50% of the missions with marginal communication conditions, LASR delivers messages requiring multiple hops to cross the network with up to 2-5 times greater reliability than flooding or DSR.

The results presented in this paper cover only a portion of the scenarios tested. A complete description of the tests is given in [33]. The full set of scenarios comprises exhaustive combinations of

- (i) line, grid, or spiral network configurations,
- (ii) 9 or 16 vehicles,
- (iii) $FER_{NN}/FER_{NN,max}$ of 0.01, 0.20, or 0.40, and
- (iv) best- or worst-case environmental conditions.

Over the full set of simulations, LASR delivers 2-3 times as many messages as flooding in 72% of the simulated missions, and delivers 2-4 times as many messages as DSR in 100% of the missions. In 67% of the missions with fair or marginal communication conditions, LASR delivers messages requiring multiple hops to cross the network with 2-5 times greater reliability than flooding or DSR.

6. Conclusion

The new location-aware source routing (LASR) is a reactive, link-state MANET routing protocol specifically designed to the constraints of an underwater acoustic network. It was intended for small underwater networks using low-bandwidth, high-latency acoustic links. Nodes were assumed to be mobile, moving at any time (including continuously) and in any direction with node speeds in the range of 0-3 m/s. LASR used the implicit information drawn from incoming transmissions to estimate ranges to neighboring nodes; these ranges were continuously fed into a tracking system which estimated local network topology. Other improvements included the addition of the ETX route metric to replace minimum hop-count, the use of an acknowledgment guarantee and aggressive, preemptive rerouting.

Simulated missions showed that the flooding protocol and DSR performed poorly in an underwater acoustic network. The flooding protocol provided reliable delivery at the cost of decreased message delivery counts. The DSR protocol regularly delivered the fewest messages with the least reliability.

The simulations showed the new LASR protocol to be superior to the DSR and blind-flooding routing protocols. In many of the simulated missions, LASR delivered more messages than flooding. In all of the simulated missions, LASR delivered more messages than DSR. Under fair or marginal communication conditions, LASR delivered messages requiring multiple hops to cross the network with greater reliability than flooding or DSR in more than half of the missions.

Appendices

A. Protocol Messages

A user-data message comprises: user data, source network address, destination network address, a source-assigned ID, and a route.

A route-request message comprises a source network address, destination network address, a source-assigned ID, and a route. It is flooded through the network and the route is built as the message travels.

A route-reply message comprises a source network address, destination network address, a source-assigned ID, and a route. It is sent in response to a route request and is routed to the originator of the request.

A route-advice message comprises topology data on one link. It is never forwarded.

An explicit-acknowledgment message comprises a source network address, a destination network address, and an identifying message-originator network address/message-ID pair. It is sent only when implicit acknowledgment is not possible.

B. Link Metric

LASR does not store the ETX, instead it stores the range and calculates the ETX as it is needed. This means the link quality

```

(1) for each msg in data-queue do
(2)   if IS-PENDING-TIME-EXPIRED(msg) then
(3)     ▷The message has been waiting too long for a route. It cannot be delivered.
(4)     DELETE-MESSAGE(msg)
(5)   else
(6)     if STATE(msg) = WAITING-FOR-ACK
(7)       and IS-ACK-TIME-EXPIRED(msg) then
(8)         STATE(msg) ← PENDING
(9)       end if
(10)      if STATE(msg) = PENDING then
(11)        route ← ROUTE-SEARCH (ORIG(msg), DEST(msg))
(12)        if route = ∅ then
(13)          ▷Enqueue a route-request message if one isn't already outstanding according to
(14)          the route-request table.
(15)          ROUTE-REQ-MAYBE (DEST(msg))
(16)        else
(17)          ▷Stop searching and return this message for transmission.
(18)          STATE(msg) ← WAITING-FOR-ACK
(19)          return msg
(20)        end if
(21)      end if
(22)    end if
(23)  end for
(24)  ▷There are no data messages to transmit at this time
(25)  return NO-MSG

```

ALGORITHM 1: The algorithm for obtaining the next user-data message from the data queue.

fields contain the distance in meters between the endpoints of the link, as estimated at the time of the timestamp.

In order to reduce the size of the protocol header, distances are stored as a 4 bit value to represent distances from 0–4080 meters in increments of 256 m. Link distance estimates are capped at 4080 m, the maximum range of the simulated acoustic modem (DPAM) [32]. An additional bit is used to flag the distance as estimated (by a time-of-flight measurement) or predicted (by the tracking system).

The timestamp communicates the age of the link data to ensure that only the newest link data are used and that stale link data are discarded. The timestamp represents the absolute age of the associated link data as a number of seconds before the start of the time window in which they are transmitted. On reception, timestamps are converted back to absolute times. Link data are considered fresh if 0–1019 seconds old, aging if 1020–2039 s old and stale if 2040 s or older. Fresh and aging link data are added to the receiver's link cache if no newer version already exists. If a newer version does exist, the link's advise priority is increased in the route-advice table. Link data are discarded if they become stale while in the cache.

C. Data Structures

The pending buffer contains all of the messages awaiting transmission: a data queue stores the outgoing user-data messages, a protocol queue stores all the protocol messages. Two data-queue messages are sent for every protocol-queue message sent.

User-data messages in the data queue exist in one of two states: pending or waiting for acknowledgment. A pending message is ready for transmission once one or more routes to the destination become available. The best route is systematically used.

Every transmitted message must be acknowledged. The message returns to pending status if the acknowledgment times out.

An unsent message in the queue will eventually expire and be removed. The algorithm for selecting the next message to transmit from the data queue is shown in Algorithm 1.

The protocol queue stores route requests and route replies. A pending route-request message or a route-reply message which does not have a route is silently discarded. If a protocol message is not acknowledged within the acknowledgment timeout, its state reverts to pending. The algorithm for selecting the next message to transmit (if any) from the protocol queue is shown in Algorithm 2.

- (1) Link Cache: The link cache stores all of the information currently available about the topology of the network. Information is added or updated whenever topology data become available and is removed when data become stale. When a route between two network nodes is needed, the first step is to search these topology data. The link cache communicates with the tracking system to receive predicted topology information that is used to supplement the cached information.

```

(1) for each msg in protocol-queue do
(2)   if STATE(msg) = WAITING-FOR-ACK
      and IS-ACK-TIME-EXPIRED(msg) then
(3)     STATE(msg) ← PENDING
(4)   end if
(5)   if STATE(msg) = PENDING then
(6)     if MSG-TYPE(msg) = ROUTE-REQUEST then
      ▷Route request message contain a route ending at the local node.
(7)       dest ← local-node
(8)     else
      ▷Route reply message contain a route ending at the message's destination.
(9)       dest ← DEST(msg)
(10)    end if
      ▷Both message type contain a route starting at message's originator.
(11)    route ← ROUTE-SEARCH (ORIG(msg), dest)
(12)    if route = ∅ then
      ▷A protocol message without a route is deleted.
(13)      DELETE-MESSAGE(msg)
(14)    else
      ▷Stop searching and return this message for transmission
(15)      STATE(msg) ← WAITING-FOR-ACK
(16)      return msg
(17)    end if
(18)  end if
(19) end for
      ▷There are no protocol messages to transmit at this time.
(20) return NO-MSG

```

ALGORITHM 2: The algorithm for obtaining the next user-data message from the data queue.

The cache is organized as a link cache: it stores data on the links between pairs of nodes. This organization allows route searches across all known and predicted topology data, potentially returning completely novel routes. Searching for routes is done with Dijkstra's algorithm [36], using weights calculated according to the link metric. Before searching for a route, the tracking system returns all predicted links and these are merged with the cached links into a temporary link table using a modified merge sort. When both the tracking system and the cache provide information about the same link, one of the two is chosen. If the link cache is unable to find a route between the requested nodes, a route request must be sent to obtain the link data necessary to build the route. A simplified version of the link data merge algorithm is shown in Algorithm 3.

The cache stores several pieces of data for each known link: endpoints, range, origin, and age. The link endpoints are the network addresses of the nodes at either end of the link. The link range is the distance between the endpoints. Origin is a flag indicating whether the distance was estimated (by a time-of-flight measurement) or predicted (by the tracking system). The link age is the time the distance was obtained: the time of the time-of-flight measurement for estimated distances or the time from the tracking system for a predicted distance. Currently, the tracking system reports the age of a prediction as the time of the most recent update to the data underlying that prediction. To prevent the use of stale data, the cache monitors age and automatically purges expired data.

Two types of data are used to update the link cache: explicit and implicit. The explicit data are extracted from received and overheard messages, including source routes, route requests, route replies, and route advice. The data include link endpoints, link range, and range age. Implicit data are recovered from received transmissions, but are not a part of the transmission data. When a transmission is received, the receiver is able to identify the transmitting node and estimate the range to that node using only on the medium model, the current time, and knowledge of the current TDMA time-slot. This provides data on the link between the receiving node and the transmitting node. These data are also used to update the tracking system.

The route-discovery table stores information about outstanding route-request messages. When a route request is originated to a given destination, an entry is added to the route-discovery table. This entry records the destination and the time the route-request message was sent. Further route requests for that destination will not be sent for as long as the entry remains in the table. This provides a delay between route requests to the same destination, allowing time for a route request to cross the network to the destination and for a route reply to return to the originator.

An entry is removed from the route-discovery table in one of two cases:

- (1) a route-reply message is received from that destination.

```

    ▷Links are sorted into ascending order of the node IDs of their endpoints.
(1)  $P \leftarrow \text{SORT-PREDICTED-LINKS}()$ 
(2)  $C \leftarrow \text{SORT-CACHED-LINKS}()$ 
    ▷Initialize the temporary merged link table.
(3)  $M \leftarrow \emptyset$ 
    ▷POP removes and returns the head element.
(4)  $p \leftarrow \text{POP}(P)$ 
(5)  $c \leftarrow \text{POP}(C)$ 
    ▷Loop until both link tables are empty.
(6) while  $p = \emptyset$  or  $c = \emptyset$ 
(7)   if  $p = c$  then
        ▷They represent the same link. Choose one based on origin and age.
(8)      $M \leftarrow \text{CHOOSE-LINK}(p, c)$ 
(9)      $p \leftarrow \text{POP}(P)$ 
(10)     $c \leftarrow \text{POP}(C)$ 
(11)   else if  $p < c$  then
(12)      $M \leftarrow p$ 
(13)      $p \leftarrow \text{POP}(P)$ 
(14)   else
(15)      $M \leftarrow c$ 
(16)      $c \leftarrow \text{POP}(C)$ 
(17)   end if
(18) end while

```

ALGORITHM 3: A simplified version of the algorithm used to merge the link data from the link cache and the tracking system. At the conclusion of the merge, M contains the merged link table.

```

    ▷Look for an earlier reply to route request  $req$ .
(1)  $prior\_route \leftarrow \text{LOOKUP-REPLY-ROUTE}(req)$ 
    ▷Get the best route from the local node to the originator of the route request.
(2)  $new\_route \leftarrow \text{ROUTE}(local\_node, \text{ORIG}(req))$ 
(3) if  $prior\_route = \emptyset$ 
        or  $\text{ETX}(new\_route) < \text{ETX}(prior\_route)$  then
        ▷No reply has ever been sent or a better route is now available.
(4)    $\text{UPDATED-REPLY-TABLE}(req, new\_route)$ 
(5)    $\text{ENQUEUE-ROUTE-REPLY}(\text{ORIG}(req))$ 
(6) end if

```

ALGORITHM 4: The route-reply algorithm.

- (2) the route-request message times out. Another route-request message may be sent after the timeout if the route is still needed.

The route-reply table stores information about outstanding route-reply messages. Each route request is uniquely identifiable in the network, but multiple copies of the same route request may be received by the destination. An entry in the route-reply table contains the unique ID of a given route request, the route sent in reply and the time the route request was first received.

If a route-request message is received with the local node as the destination, the node checks its route-reply table. If no entry exists for the unique ID of the route request, a route-reply message is returned to the request's originator. If an entry does exist, the best-available route is compared to the route stored in the entry. If the best-available route is better than the stored route, a new route reply is originated

containing the new route and the stored route is updated. When an entry expires, it is removed from the table. The route-reply algorithm is shown in Algorithm 4.

The route-request table ensures a node processes a given route-request message only once, regardless of the number of times it is received. When a route-request message is received it will be enqueued for resending if there is no corresponding entry in the route-request table. An entry containing that route request's unique ID and the time is then added to the table. If a route request is received and there is a corresponding entry in the table, the request is dropped. The time is used to determine the expiration of the entry, which depends on the size of the route request circular ID used by the network. An entry is removed from the table when it expires.

The route-advice table serves to prioritize the sending of route-advice messages. When topology data are received,

either implicitly or explicitly, they are processed by the link cache. If the received data are older than the data in the cache, the route-advice table is updated. The table stores link data along with a priority value; the priority is updated based on the age of the received data. When the opportunity arises to send a route-advice message, the message data are selected randomly from the route-advice table using the priority as a weight. The priorities ensure that it is more likely for older and more commonly used data to be advised first. Advice data selected for transmission are removed from the table.

References

- [1] J. J.-N. Liu and I. Chlamtac, "Mobile Ad Hoc networking with a view of 4G wireless: Imperatives and challenges," in *Mobile Ad Hoc Networking*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds., chapter 1, pp. 1–45, Wiley-Interscience, 2004.
- [2] J. P. Macker and M. S. Corson, "Mobile Ad Hoc networks (MANETs): routing technology for dynamic wireless networking," in *Mobile Ad Hoc Networking*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds., chapter 9, pp. 255–273, Wiley-Interscience, 2004.
- [3] R. Rajaraman, "Topology control and routing in Ad Hoc networks: a survey," *SIGACT News*, vol. 33, no. 2, pp. 60–73, 2002.
- [4] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds., *Ad Hoc Networking*, Wiley-Interscience, 2004.
- [5] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pp. 234–244, ACM Press, New York, NY, USA, 1994.
- [6] E. M. Belding-Royer and C. E. Perkins, "Evolution and future directions of the Ad Hoc on-demand distance-vector routing protocol," *Ad Hoc Networks*, vol. 1, no. 1, pp. 125–150, 2003.
- [7] R. Ogier, F. Templin, and M. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)," RFC 3684, IETF MANET Working Group, February 2004.
- [8] V. D. Park and M. S. Corson, "Highly adaptive distributed routing algorithm for mobile wireless networks," in *Proceedings of the 16th IEEE Annual Conference on Computer Communications (INFOCOM '97)*, pp. 1405–1413, April 1997.
- [9] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile Ad Hoc networks (DSR)," INTERNET-DRAFT, IETF MANET Working Group, July 2004.
- [10] E. M. Sozer, M. Stojanovic, and J. G. Proakis, "Underwater acoustic networks," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 72–83, 2000.
- [11] H. Ju, I. Rubin, K. Ni, and C. Wu, "A distributed mobile backbone formation algorithm for wireless Ad Hoc networks," in *Proceedings of the 1st International Conference on Broadband Networks (BroadNets '04)*, pp. 661–670, October 2004.
- [12] M. K. Marina and S. R. Das, "Performance of route caching strategies in dynamic source routing," in *Proceedings of the International Conference on Distributed Computing Systems Workshop*, pp. 425–432, 2001.
- [13] B. C. Seet, B. S. Lee, and C. T. Lau, "Optimisation of route discovery for dynamic source routing in mobile Ad Hoc networks," *Electronics Letters*, vol. 39, no. 22, pp. 1606–1607, 2003.
- [14] F. Bai, N. Sadagopan, B. Krishnamachari, and A. Helmy, "Modeling path duration distributions in MANETs and their impact on reactive routing protocols," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp. 1357–1373, 2004.
- [15] S. R. Medidi and K. H. Vik, "Quality of service-aware source-initiated Ad Hoc routing," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON '04)*, pp. 108–117, October 2004.
- [16] C.-H. Chuan and S.-Y. Kuo, "Cache management of dynamic source routing for fault tolerance in mobile Ad Hoc networks," in *Proceedings of the Pacific Rim International Symposium on Dependable Computing*, pp. 199–205, 2001.
- [17] W. P. Chen and J. C. Hou, "Dynamic, Ad Hoc source routing with connection-aware link-state exchange and differentiation," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 1, pp. 188–194, November 2002.
- [18] J. R. Potter, A. Berni, J. Alves, D. Merani, G. Zappa, and R. Been, "Underwater communications protocols and architecture developments at NURC," in *Proceedings of the OCEAN*, pp. 1–6, Santander, Spain, 2011.
- [19] J. Ahn, A. Syed, B. Krishnamachari, and J. Heidemann, "Design and analysis of a propagation delay tolerant ALOHA protocol for underwater networks," *Ad Hoc Networks*, vol. 9, no. 5, pp. 752–766, 2011.
- [20] M. S. Rahim, P. Casari, F. Guerra, and M. Zorzi, "On the performance of delay-tolerant routing protocols in underwater networks," in *Proceedings of the OCEAN*, pp. 1–7, Santander, Spain, 2011.
- [21] A. Patil and M. Stojanovic, "A node discovery protocol for Ad Hoc underwater acoustic networks," in *Proceedings of the OCEAN*, pp. 1–9, Spain, June 2011.
- [22] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: recent advances and future challenges," *Marine Technology Society Journal*, vol. 42, no. 1, pp. 103–116, 2008.
- [23] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research challenges and applications for underwater sensor networking," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '06)*, vol. 1, pp. 228–235, April 2006.
- [24] D. Pompili and I. Akyildiz, "Overview of networking protocols for underwater wireless communications," *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 97–102, 2009.
- [25] E. P. C. Jones and P. A. S. Ward, "Routing strategies for delay-tolerant networks," in *ACM Computer Communication Review*, 2006.
- [26] I. Cardei, C. Liu, J. Wu, and Q. Yuan, "DTN routing with probabilistic trajectory prediction," in *Proceedings of the Wireless Algorithms, Systems, and Applications (WASA '08)*, vol. 5258 of *Lecture Notes in Computer Science*, pp. 40–51, 2008.
- [27] M. Stojanovic, L. Freitag, J. Leonard, and P. Newman, "A network protocol for multiple AUV localization," in *Proceedings of the MTS/IEEE Oceans Conference*, vol. 1, pp. 604–611, Biloxi, Miss, USA, October 2002.
- [28] R. Bachmayer, J. A. Rice, R. Crebert, and C. Fletcher, "Navigation and control of multiple gliders in an underwater acoustic network," in *Proceedings of the IEEE/OES Autonomous Underwater Vehicles*, pp. 54–58, June 2004.
- [29] O. Normand, *Synchronous spatial array processing for underwater vehicle tracking [M.S. thesis]*, Florida Atlantic University, 2002.

- [30] E. A. Carlson, P. P. Beaujean, and E. An, "An improved location-aware routing protocol for mobile underwater acoustic networks," in *Proceedings of the MTS/IEEE Oceans Conference*, Vancouver, Canada, October 2007.
- [31] P. P. J. Beaujean and E. P. Bernault, "A new multi-channel spatial diversity technique for long range acoustic communications in shallow water," in *Proceedings of the MTS/IEEE Oceans Conference*, pp. 1533–1538, September 2003.
- [32] E. A. Carlson, P. - Beaujean, and E. An, "An Ad Hoc wireless acoustic network simulator applied to multiple underwater vehicle operations in shallow waters using high-frequency acoustic modems," *U.S. Navy Journal of Underwater Acoustics*, vol. 56, pp. 113–139, 2006.
- [33] E. A. Carlson, *Simulator and location-aware routing protocol for mobile Ad Hoc acoustic networks of AUVs [Ph.D. dissertation]*, Florida Atlantic University, 2009.
- [34] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 134–146, ACM Press, New York, NY, USA, September 2003.
- [35] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *Proceedings of the Conference on Computer Communications (ACM SIGCOMM '04)*, pp. 133–144, ACM Press, New York, NY, USA, September 2004.
- [36] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

