

Loop-Free and Localized Fast Reroute in Underwater Ad Hoc Networks

Haoran Tang* **, Rui Han* **

*National Network New Media Engineering Research Center, Institute of Acoustics,
Chinese Academy of Sciences, Beijing, China

**School of Electronic, Electrical and Communication Engineering,
University of Chinese Academy of Sciences, Beijing, China
tanghr@dsp.ac.cn, hanr@dsp.ac.cn

Abstract— Underwater ad-hoc networks (UANETs) suffer from slow propagation, high error rates, and sluggish control-plane reaction, making them highly vulnerable to sudden multi-node failures. Stale routes during convergence send packets along obsolete next hops, creating blind zones with cascading losses and severe energy waste, while existing recovery schemes rely on slow signaling, heavyweight computation, or non-trivial overhead that does not fit dynamic underwater conditions. This paper presents UW-LFRR, a lightweight, data-plane-driven fast-recovery mechanism. UW-LFRR constructs a purified local topology view, models failure impact via a severed-branch abstraction, and uses a two-stage relay selection strategy that first searches near the original shortest path and then performs a constrained fallback search inside the affected region. A minimal-overhead first-packet paving procedure installs temporary forwarding entries along the detour, enabling subsequent packets to bypass the damaged area without additional computation. Results from NS-3 simulations indicate that UW-LFRR achieves loop-free recovery using only local connectivity information with locality-bounded computation cost, while maintaining high packet-delivery ratios and low end-to-end delay with substantially reduced control overhead across diverse multi-failure scenarios, demonstrating its practicality for resilient underwater networks.

Keywords— Fast Reroute, Localized Repair, Loop-Free Forwarding, Underwater Ad Hoc Networks

I. INTRODUCTION

Underwater ad-hoc networks (UANETs) are key enablers for marine sensing and mission-critical underwater operations, but acoustic channels fundamentally differentiate them from terrestrial wireless networks: propagation is slow (~ 1500 m/s), bandwidth is only a few kbps, bit-error rates are high, and nodes have tight energy budgets [1]. These constraints lead to fragile link quality, long reaction cycles, and slow control-plane dissemination, so routing protocols designed for radio-based environments suffer severe performance degradation when directly ported to UANETs.

In array- and formation-based UANETs [2]-[6], nodes maintain quasi-static spatial relationships within a routing-control period, so link-state protocols can normally preserve a consistent global view. However, sudden multi-node failures — caused by battery depletion, equipment malfunction, or

localized physical damage — instantly change the physical topology, while the control plane needs several update cycles to converge. This mismatch creates a vulnerable Convergence Blind Zone (CBZ), during which packets are forwarded along pre-failure routes, leading to large-scale packet loss, queue buildup, and severe energy waste.

Traditional UANET routing protocols struggle to cope with CBZs. Proactive schemes such as HL-OLSR [7] suffer from long convergence delays due to large RTTs, whereas reactive schemes such as RB-AODV [8] trigger broadcast storms that further congest the acoustic channel. Terrestrial Fast Reroute (FRR) [9]-[11] designs are also ill-suited: precomputed backup paths [12] are fragile under multi-node failures, and failure-carrying packet schemes [13]-[14] impose heavy per-packet computation overhead on energy-constrained nodes.

These limitations reveal a fundamental gap: UANETs need a data-plane mechanism that performs immediate, low-overhead local repair without waiting for global convergence. To this end, we propose the Underwater Loop-Free and Localized Fast Reroute (UW-LFRR) scheme. UW-LFRR combines a topology snapshot with local failure perception to construct a Local Purified View, identifies the Branch-off Region K_u — the subgraph affected by failures — and confines the repair search to this region. A Two-Stage Relay Selection Strategy follows a “safety-first, fallback-as-needed” hierarchy: Phase I rapidly grafts traffic back to the backbone with complexity $O(|K_u|)$, while Phase II performs a bounded local search only when necessary to preserve connectivity.

The main contributions of this work are summarized as follows: 1) Two-Stage Relay Selection Strategy: We design a hierarchical repair algorithm that operates strictly within the local Branch-off Region K_u . By prioritizing a backbone-oriented fast search (Phase I) and using a branch-region search (Phase II) only when necessary, the strategy balances bounded path stretch and guaranteed connectivity while significantly reducing computational complexity compared to global re-routing. 2) Snapshot-Based Local Recovery Framework: We propose a CBZ-oriented recovery framework that uses topology snapshots and a First-Packet Paving technique to install temporary, loop-free forwarding entries using only local information, thereby eliminating network-wide

broadcasts and minimizing control overhead. 3) Analytical Guarantees and Performance Validation: We provide theoretical proofs showing that UW-LFRR guarantees loop-freedom and algorithmic efficiency, and NS-3 simulations demonstrating superior packet delivery ratios, lower end-to-end delay, and reduced computation overhead compared with state-of-the-art underwater routing and recovery protocols.

II. LOOP-FREE AND LOCALIZED FAST REROUTE IN UNDERWATER AD HOC NETWORKS

A. Failure Modeling and Local Purified View

We focus on the array-/formation-based UANETs described in Section I, where nodes move slowly with missions or currents and the relative topology remains almost unchanged within one convergence period. Under such quasi-static conditions, severe routing failures are mainly caused by sudden multi-node damage rather than high-speed mobility. Since extremely low data rates and long propagation delays make global routing convergence much slower than topology changes, UW-LFRR avoids synchronizing global state upon each failure and instead combines a topology snapshot with local perception to achieve fast, purely local recovery.

1) Network model and topology snapshot: The network is modeled as an undirected graph $G=(V,E)$, where V is the set of nodes and E is the set of links. Before failures, each node $u \in V$ maintains a network-wide topology snapshot G_{snap} and constructs a local shortest path tree (SPT) based on hop count:

$$T_u = \text{SPT}(G_{snap}, \text{root}=u)$$

G_{snap} is the perceived pre-failure view, while the post-failure physical topology is G_{real} , with $G_{real} \subset G_{snap}$. The discrepancy induces a Convergence Blind Zone: before the control plane updates the topology, the data plane still forwards according to G_{snap} , causing packet loss.

2) Dynamic failure set perception: When node u (the repair node RN) forwards along T_u , it detects failures via missing ACKs or failure information carried in packet headers. We define the dynamic failure set as

$$F_u(t) = \{v \in V \mid v \text{ is identified as failed by } u\},$$

which contains current and previously known failures, with $|F_u(t)| \geq 1$, and is updated whenever neighbor failures are detected or failure information is received.

3) Local purified topology and forbidden set: UW-LFRR constrains subsequent search by characterizing the failure-affected region. In tree-based routing, the failure of a node disconnects its downstream subtree. For any failed node $f \in F_u(t)$, we define its branch-off set

$$K(f) = \{v \in V \setminus f \mid \text{Path}_{T_u}(u, v) \cap f \neq \emptyset\},$$

i.e., all downstream nodes whose paths in T_u from u must pass through f (excluding f itself). For the entire failure set (Figure 1),

$$K_u = K(F_u(t)) = \bigcup_{f \in F_u(t)} K(f),$$

and the forbidden set

$$\text{Forbidden}_u = F_u(t) \cup K_u$$

contains all nodes that must not participate in forwarding: failed nodes and blocked branch nodes.

SPT of node 2 and severed branch at failed node 1

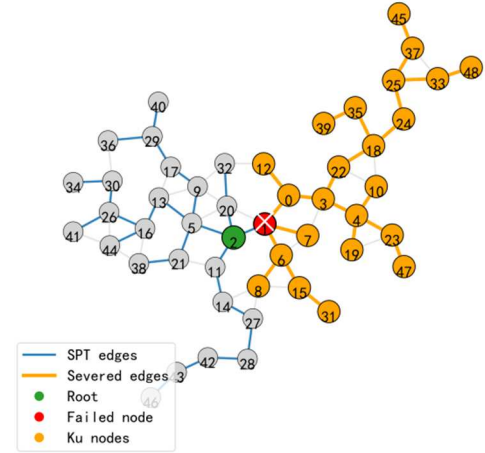


Figure 1. Formation of the Failure-Affected Region in UW-LFRR's SPT-based Model

Based on local G_{snap} and Forbidden_u , RN constructs the local purified topology

$$G'_u = G_{snap} \setminus \text{Forbidden}_u,$$

i.e., the subgraph obtained by removing all $v \in \text{Forbidden}_u$ and their incident edges from G_{snap} . All subsequent border relay (BR) selection and path computation are performed on G'_u , ensuring logical paths consist only of healthy, reachable nodes. This only requires locally marking failed nodes and their branch-off sets, without network-wide broadcasts, and incurs very low computation overhead.

The branch-off set K_u provides three key properties for the repair algorithms:

- Search space restriction. Constructing K_u has complexity $O(|V|)$, but typically $|K_u| \ll |V|$, so Phase I/II searches can be confined to K_u , avoiding full-graph traversal.
- Repair necessity test. For any destination DST , if $DST \in K_u$, it is unaffected and can be forwarded along the original SPT; repair is triggered only when $DST \notin K_u$.
- Grafting candidate set. Any valid grafting node GR must satisfy $GR \in K_u$, so the candidate size is at most $|K_u|$, ensuring both phases run efficiently within a well-bounded search space.

B. Two-Stage Relay Node Selection Strategy

Given $F_u(t)$, K_u , and G'_u , UW-LFRR designs a two-stage relay node selection strategy that balances path quality, routing stability, and computation/control overhead.

1) Problem Statement and Design Rationale: In the local purified topology $G'_u = G_{snap} \setminus \text{Forbidden}_u$, the repair node $RN=u$ is given the branch-off region K_u , dynamic failure set $F_u(t)$, and destination DST . It needs a relay pair (BR, GR) and a repair path

$$P_{rec}: RN \rightarrow \dots \rightarrow BR \rightarrow GR \rightarrow \dots \rightarrow DST$$

such that: (1) all nodes on the path are outside $F_u(t)$; (2) the path length is close to the pre-failure shortest path $d_{orig}(RN, DST)$; and (3) the number of temporary forwarding entries needed to realize this path is minimized.

The design follows two principles: (i) stay close to the original backbone, performing minimal detours around failures to control delay and energy; and (ii) when the backbone is completely cut, relax this constraint and search the branch-off region to guarantee connectivity. Accordingly, UW-LFRR adopts two phases: Phase I quickly finds a bounded-suboptimal, stable repair path near the backbone with complexity $O(|K_u|)$ for most failures, while Phase II is invoked only when Phase I fails and performs a relatively optimal search over the branch-off region with complexity approximately $O(|K_u| \log |K_u|)$, guaranteeing a solution whenever physical connectivity exists.

2) Phase I: Backbone-Oriented Fast Search with Bounded Stretch: Let $RN=u$, and denote the original shortest path based on T_u by

$$P_{orig} = \text{Path}_{T_u}(u, DST),$$

where failures cause some nodes on P_{orig} to fall into K_u .

Step 1: Backbone candidate set. Within K_u , select nodes that lie on P_{orig} and are not failed:

$$C_{main} = \{v \in K_u \cap P_{orig} \mid v \notin F_u(t)\}.$$

These nodes are on the original backbone downstream of the failures; once packets are grafted back to C_{main} , they naturally follow the SPT towards DST without revisiting the failure region.

Step 2: Candidate grafting pairs (BR, GR) . On G_{snap} , for each $GR \in C_{main}$, we examine neighbors $N(GR)$. For each neighbor BR , if $BR \in G'_u$ (i.e., $BR \in \text{Forbidden}_u$), edge (BR, GR) is a cross-boundary edge between the healthy backbone and the backbone-safe region, and we add $(BR, GR) \in P_{cand}$, $GR \in C_{main}$, $BR \in V(G'_u)$, $(BR, GR) \in E(G_{snap})$.

Step 3: Nearest relay pair (overhead minimization). We choose the pair minimizing the SPT distance from RN to BR :

$$(BR^\square, GR^\square) = \arg \min (BR, GR) \in P_{cand} d_{T_u}(RN, BR).$$

Temporary entries are needed only along $RN \rightarrow BR$, so minimizing $(d_{T_u}(RN, BR))$ minimizes the number of updated nodes and thus control overhead. Via single-hop $BR^\square \rightarrow GR^\square$, packets are then grafted back to the backbone and follow $GR^\square \rightarrow DST$ on the SPT.

Step 4: Output or failure. If $P_{cand} \neq \emptyset$ and we obtain $(BR^\square, GR^\square)$, Phase I outputs

$$(BR^\square, GR^\square, FN_{current})$$

as the first-packet repair instruction. Phase I fails only when $C_{main} = \emptyset$ or $P_{cand} = \emptyset$, indicating no usable backbone grafting point in (K_u) , and the system falls back to Phase II.

Implementation-wise, Phase I runs in $O(|K_u|)$ via a linear scan over K_u to build C_{main} and a bounded enumeration of cross-boundary neighbors.

3) Theoretical Properties of Phase I: Phase I prioritizes routing stability and low overhead by grafting traffic back onto the original shortest path whenever possible, achieving

bounded path stretch (≤ 2 extra hops) with minimal computational cost $O(|K_u|)$.

Lemma 1: Tree-distance constraint for neighboring nodes.

In the SPT (T_u) rooted at RN , if BR and GR are physically adjacent but belong to different branches, then

$$|d_{T_u}(RN, BR) - d_{T_u}(RN, GR)| \leq 1.$$

This follows from the triangle inequality and underpins the path-stretch analysis.

Theorem 2: Upper bound on path stretch.

If $GR \in P_{orig}$, then the logical repair path

$$P_{logic}: RN \rightarrow \dots \rightarrow BR \rightarrow GR \rightarrow \dots \rightarrow DST$$

has length L_{logic} satisfying

$$L_{logic} \leq d_{orig}(RN, DST) + 2.$$

Thus, Phase I yields a repair path with at most two extra hops.

Theorem 3: Backbone prioritization and recursion suppression).

If GR is chosen from non-backbone nodes in K_u , its SPT path to DST often returns to the backbone via failed nodes or their parents, easily triggering recursive repair. Constraining $GR \in P_{orig}$ keeps the post-grafting path away from the failure region and greatly reduces recursion probability.

Theorem 4: Closest-BR minimizes temporary routing state.

Among all feasible (BR, GR) pairs, choosing BR that minimizes $d_{T_u}(RN, BR)$ minimizes the number of temporary entries, since they only reside on $(RN \rightarrow BR)$, and is therefore optimal in terms of control overhead.

Summary: Properties of Phase I.

Phase I provides: (1) bounded path stretch (at most two extra hops); (2) high stability via backbone grafting and suppressed recursion; (3) minimal temporary state via Closest-BR; and (4) low complexity $O(|K_u|)$. In most failure scenarios, Phase I alone yields a sufficiently short and stable repair path at low cost.

4) Phase II: Branch-Off Fallback with Relatively

Optimal Search: When $C_{main} = \emptyset$, no backbone node in K_u is usable, and the backbone is locally cut. Phase II relaxes the backbone-first constraint and searches for a short detour over all of K_u . The idea is to run a shortest-path search on G'_u restricted to K_u and find a pair (BR, GR) minimizing total cost, with BR on a healthy backbone branch and GR inside the branch-off region.

Step 1: Fallback candidate set.

$$C_{fallback} = K_u$$

(failed nodes are excluded by definition). If $C_{fallback}$ is empty or highly disconnected, a physical partition may already exist.

Step 2: Dijkstra from DST on $C_{fallback}$.

On $C_{fallback}$, run Dijkstra (or BFS) from DST to obtain

$$d_{C_{fallback}}(v, DST), \quad \forall v \in V(C_{fallback}).$$

Step 3: Healthy backbone neighbors in the local SPT.

From T_u , select BR in $(G'_u \setminus T_u)$ that have at least one neighbor $GR \in C_{fallback} = K_u$ in G'_u . These form B_{cand}^{br} ; for each BR , neighbors $GR \in K_u$ form $G_{cand}(BR)$.

Step 4: Total cost evaluation.

For each feasible pair (BR, GR) , compute

$$Cost(BR, GR) = d_{G'_u}(RN, BR) + 1 + d_{C_{fallback}}(GR, DST),$$

and choose

$$(BR', GR') = \arg \min_{(BR, GR)} Cost(BR, GR).$$

Step 5: Output or declare failure.

If a feasible pair exists, Phase II outputs

$$(BR', GR', FN_{current})$$

for the first-packet repair instruction. If $B_{cand}^{br} = \emptyset$ or all $G_{cand}(BR)$ are empty, RN and DST are physically disconnected in the current view and no local repair is possible.

5) Theoretical Properties of Phase II: Phase II accepts some routing instability (potential recursion) to guarantee connectivity when the backbone is fully disrupted.

Theorem 5: Existence under physical connectivity.

If in the real topology G_{real} , RN can still reach DST after failures, and $K_u \sqcap F_u(t)$ is connected in G'_u and connected via at least one edge to healthy nodes outside the backbone, then Phase II is guaranteed to find a repair path in G'_u . Thus, as long as the network remains physically connected, Phase II will not miss an existing route.

Theorem 6: Complexity comparison of the two phases.

Let $|K_u|$ be the number of nodes and E_K the set of edges in the branch-off region. Phase I has time complexity $O(|K_u|)$, while Phase II has

$$O(|K_u| \log |K_u| + |E_K|),$$

which is approximately $O(|K_u| \log |K_u|)$ in sparse graphs. Compared with running Dijkstra once on the entire graph G with complexity $O(|V| \log |V|)$, the local search on K_u saves at least

$$\frac{|V|}{|K_u|} \sqcap \frac{\log |V|}{\log |K_u|}$$

in computation, which is crucial for energy- and computation-constrained underwater nodes.

6) Coordination of the Two Phases and Overall Framework:

UW-LFRR's two-stage relay selection forms a "safety-first, fallback-as-needed" hierarchy. Phase I, used by default whenever the backbone is still partially available, searches only within K_u , achieving bounded path stretch, low recursion risk, and minimal control overhead with complexity $O(|K_u|)$. Phase II is triggered only when Phase I fails due to complete backbone disruption; it performs a relatively optimal search over K_u with complexity $O(|K_u| \log |K_u|)$ and guarantees a solution under physical connectivity, providing a survival baseline even at the cost of some stability.

Operationally, when the repair node detects primary path failure, it first runs Phase I locally; only if Phase I has no solution does it invoke Phase II. If Phase II also fails, the system concludes that a physical partition has formed and leaves the partitioned state to higher-layer mechanisms.

Figure 2 illustrates the UW-LFRR local repair process.

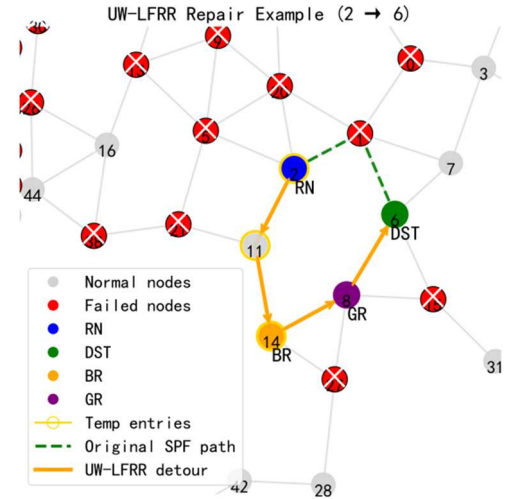


Figure 2. UW-LFRR Local Repair Process with Detour Path Construction

C. Data-Plane-Driven Lightweight Signaling Mechanism

UW-LFRR materializes the logical repair paths from Section II-B purely in the data plane, avoiding both complex source-routing headers and network-wide flooding. It adopts a first-packet-driven, on-demand installation mechanism: the first failure-affected packets carry a compact control triplet $\sqcap DST, BR, FN \sqcap$, which installs temporary forwarding entries along the detour; subsequent packets of the same flow then follow the repaired route transparently without extra control headers. The signaling scope is constrained to the upstream region around the failure, keeping overhead acceptable for UANETs.

Each participating node maintains minimal, local, soft-state entries scoped to a (destination, failure context) pair:

$$(DST, FN) \rightarrow (NextHop_{BR}, EntryTTL).$$

Here $NextHop_{BR}$ is the next hop on the $RN \rightarrow BR$ segment and $EntryTTL$ is a finite lifetime that decreases to zero, after which the entry is removed. When multiple failures occur, a node may hold several entries for the same destination (e.g., $(DST, FN1)$ and $(DST, FN2)$ with $FN2 \supset FN1$). Expired entries are purged, and the one with the largest remaining $EntryTTL$ always takes precedence, ensuring that the most recent failure view overrides older paths.

To limit state spread, entries are installed only at nodes that would otherwise misroute traffic. A node creates a temporary entry if and only if its original next hop toward DST lies in the failure region and differs from the next hop toward BR . Otherwise, it is treated as part of the healthy backbone and does not participate in detouring. Once a node observes that its next hops for DST and BR coincide, traffic has already rejoined the backbone; the node stops installing new entries and can strip the control triplet. With local-only installation, bounded lifetimes, and a latest-first policy, temporary state remains confined to a small area around the failure and converges naturally over time, while new failures simply trigger a fresh round of first-packet paving under the updated view.

Algorithm 1 formalizes the precise conditions for temporary entry installation and control triplet processing during the first-packet paving procedure.

Algorithm 1: First-Packet Paving at Node u

Input: Packet P

Output: Forwarding decision and routing table update

```

1: if  $P$  carries control triplet  $\square$  DST, BR, FN  $\square$  then
2:   extract  $\square$  DST, BR, FN  $\square$  from  $P$ .header
3:    $dst\_next \leftarrow next\_hop\_SPT(u, DST)$  // Original next hop to DST
4:    $br\_next \leftarrow next\_hop\_SPT(u, BR)$  // Next hop to boundary relay
5:    $path\_traverses\_FN \leftarrow check\_SPT\_path(u, DST, FN)$  // Check if
   SPT path traverses FN
6:   if (not  $path\_traverses\_FN$ ) or ( $dst\_next = br\_next$ ) then // Strip
   condition
7:     strip triplet from  $P$ .header
8:     forward  $P$  to  $dst\_next$ 
9:     return
10:  end if
11:  if  $path\_traverses\_FN$  and ( $dst\_next \neq br\_next$ ) then // Install
   condition
12:    install_temp_entry( $u, DST, FN, br\_next, TTL=default$ )
13:  end if
14:  forward  $P$  to  $br\_next$  // Follow repair path
15: else
16:  forward  $P$  according to local routing policy // Normal
   forwarding or repair initiation
17: end if

```

D. Recursive Multi-Failure Recovery Framework

In underwater environments, node failures often occur in correlated bursts, making it impractical to wait for global routing convergence after each event. UW-LFRR therefore decomposes complex multi-failure scenarios into a sequence of “single-failure + local repair” steps. Whenever a packet encounters a new failure that invalidates the current repair instruction, the node that first detects this failure promotes itself to a new repair node, extends the failure set, and recomputes a local detour.

Packet headers carry an accumulated failure list $FNlist$. Upon promotion, the new repair node RN' updates $FNlist = FNlist \square NHnext$, rebuilds the local purified view by pruning all nodes in $FNlist$, and reruns the two-stage relay selection algorithm on the updated topology. The resulting repair path is then instantiated using the data-plane signaling mechanism of Section II-C. Intuitively, “the node that first sees the new failure takes over,” so multi-failure events are handled as a chain of local repairs instead of a single global operation.

Before performing a new repair, RN' runs a lightweight Sanity Check. It recomputes the branch-off set K' under $FNlist$ and examines whether the destination DST remains in K' . If $DST \square K'$, both the original and current repair paths are obsolete, and RN' computes a new (BR', GR') pair on K' to attempt another detour. If $DST \square K'$, DST has escaped the shadow of the present failures and is likely reachable via ordinary SPT forwarding; RN' then discards all UW-LFRR headers and temporary entries and falls back to the underlying link-state routing.

Each repair segment ($RN \rightarrow BR$, $BR \rightarrow DST$, or $GR \rightarrow DST$) either follows a tree path in an SPT or a shortest path on a purified topology G'_u , and is therefore loop-free by construction. During recursive upgrades, every promotion step enlarges the failure set, updates the underlying view, and prunes all paths through newly failed nodes. Combined with finite-lifetime temporary entries and the latest-first selection rule, this ensures that the same packet cannot traverse a persistent loop under an unchanged rule set.

Overall, UW-LFRR’s recursive framework provides fast local recovery with constant-bounded path stretch when the backbone is partially intact, falls back to relatively short detours within the branch-off region when it is heavily damaged, and keeps computation and state overhead proportional to the size of K_u rather than the entire network V .

III. EXPERIMENTAL EVALUATION

A. Simulation Setup

UW-LFRR is implemented in NS-3.40 with a realistic underwater acoustic PHY and TDMA-based MAC. A 49-node grid topology (1000 m spacing, 1100 m transmission range) is deployed over a $10000m \times 10000m$ area (Figure 3) with 4 kbps effective bandwidth. The sound propagation speed is set to 1500 m/s. After the global routing state converges, we randomly select $N \square [0, 20]$ nodes and mark them as unavailable to emulate sudden multi-node failures. Each configuration is repeated 100 times (Monte Carlo method), and the averaged results are reported. Simulation parameters are summarized in Table 1.

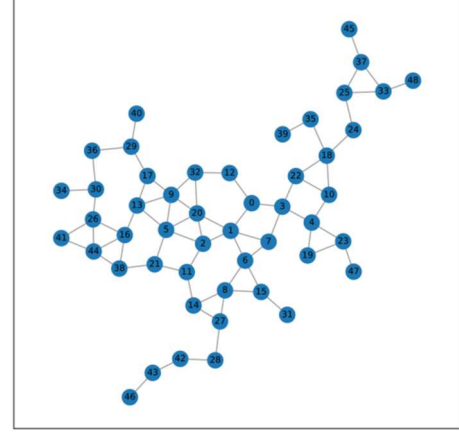


Figure 3. Randomly Generated Network Topology with 49 Nodes

TABLE 1. SIMULATION PARAMETER CONFIGURATION

Parameter	Value
Network Area	$10000m \times 10000m$
Total Nodes	49
Transmission Range	1100 m
Node Spacing	1000 m
Bandwidth	4 kbps
Data Packet Size	100 Bytes
Slot Duration	3 s
Slots per Frame	49
Traffic Patterns	All-to-All, Continuous P2P,

Random Burst

While direct energy consumption measurements would be ideal for underwater environments, the significant variations in hardware platforms, communication technologies, and computational architectures make standardized energy models challenging to establish. Therefore, we adopt computation counts and control overhead ratios as representative proxies for energy efficiency. These metrics provide a consistent, comparable foundation for protocol evaluation that directly correlates with energy consumption patterns.

We evaluate three traffic patterns representing common UANET workloads:

- All-to-All, reflecting cooperative high-load missions.
- Continuous P2P: modeling long-lived sensing flows toward a sink.
- Random Burst, mimicking emergency or event-driven transmissions.

The evaluation metrics include packet delivery ratio (PDR), end-to-end delay, control-overhead ratio, average hop count, and average computation overhead of the recovery mechanism. Baseline protocols are HL-OLSR, RB-AODV, FRP [12], FCP[13] and RTR [14].

B. Performance under Multi-Node Failures

After network stabilization with HL-OLSR, we simulate abrupt multi-node failures to replicate the steady-state-to-damage transition in UANETs. These failures precede routing topology updates, forcing packets to rely on stale routing information during convergence. Performance in this window thus reflects the efficacy of each protocol's local recovery mechanism. We evaluate three traffic modes using packet delivery ratio, end-to-end delay, control overhead, hop count, and computational cost.

1) Packet Delivery Ratio: UW-LFRR demonstrates high resilience across all traffic modes. In All-to-All and Random Burst scenarios, UW-LFRR's PDR closely tracks the theoretical upper bound set by flooding-based protocols (RB-AODV/FRP), maintaining connectivity even with 15 – 20 failed nodes. In Continuous P2P flows, UW-LFRR significantly outperforms HL-OLSR and RTR, proving that data-plane rerouting effectively rescues flows without global convergence (Figure 4).

2) End-to-End Delay: HL-OLSR shows the lowest average delay, but mainly because many long paths fail and are excluded from the statistics. Among recovery schemes, RB-AODV and FRP suffer the largest delays due to repeated route discovery and backup activation, which congest slow acoustic channels. UW-LFRR and FCP maintain moderate and stable delays in all traffic patterns; in all-to-all and random-burst scenarios, their delay is much lower than RB-AODV, FRP, and RTR while achieving comparable PDR. In the continuous P2P flow, UW-LFRR's delay stays close to FCP's and only slightly above HL-OLSR, but with substantially higher PDR (Figure 5).

3) Control Overhead: RB-AODV and FRP exhibit prohibitive overhead (>80% ratio) due to broadcast storms.

Conversely, UW-LFRR incurs negligible overhead, virtually identical to the baseline HL-OLSR. By embedding signaling in data headers and restricting state to the failure locality, UW-LFRR decouples recovery resilience from control-plane congestion (Figure 6).

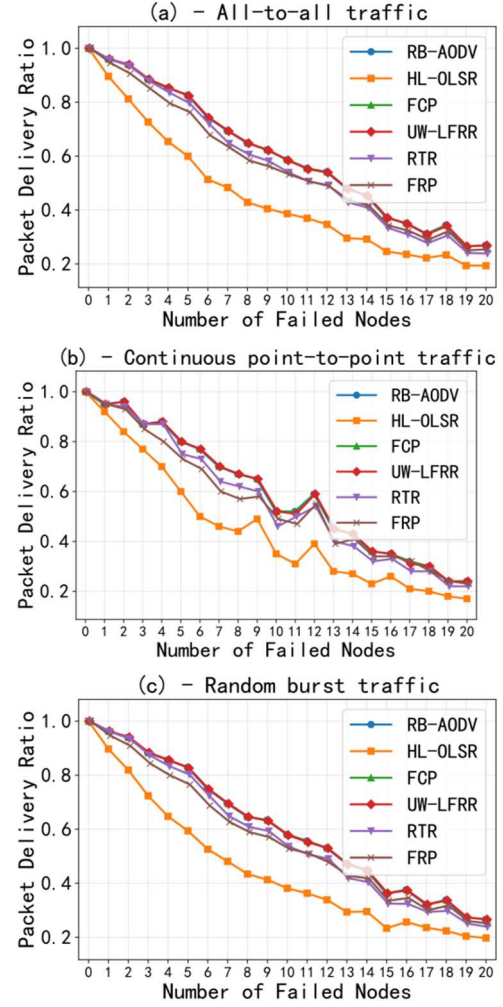
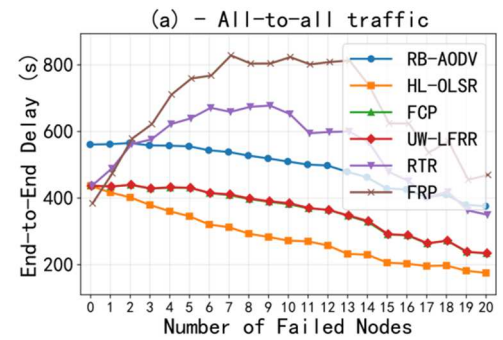


Figure 4. Packet delivery ratio under three traffic scenarios



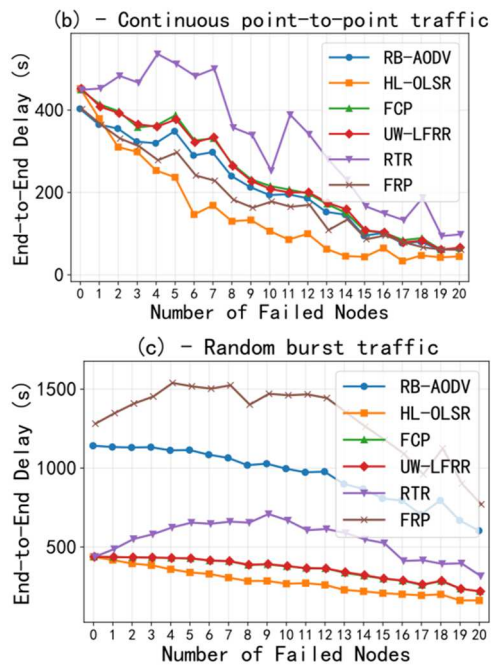


Figure 5. End-to-end delay under three traffic scenarios

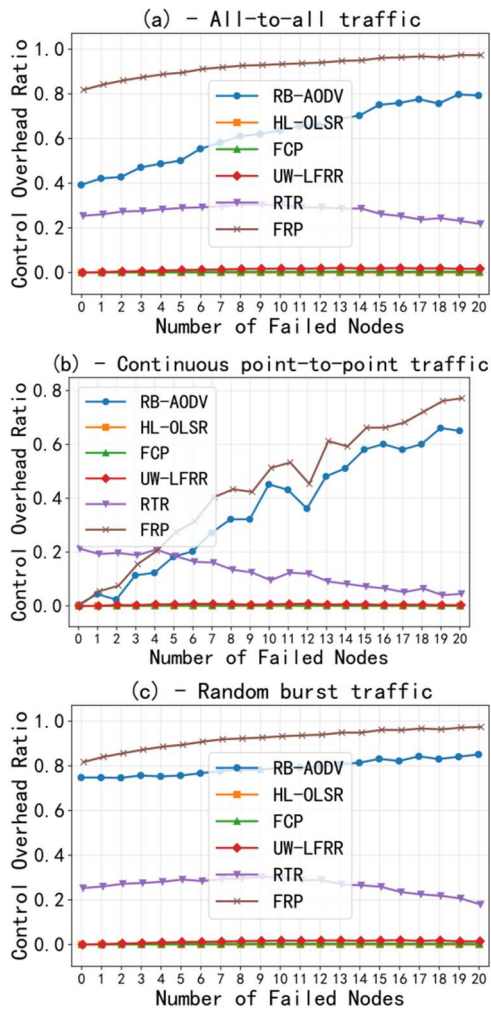


Figure 6. Control overhead ratio under three traffic scenarios

4) Average Hop Count: Fig. W shows that in all three scenarios, average hop counts decrease as node failures increase, as long paths break and surviving traffic becomes localized. HL-OLSR yields the lowest hop counts—an artifact of its high packet loss for long flows. In contrast, the recovery schemes (FCP, UW-LFRR, RB-AODV, FRP, RTR) produce closely clustered curves. UW-LFRR's hop count remains near FCP and within one hop of HL-OLSR even under high failures, confirming that its two-stage local repair avoids excessive detours and achieves bounded path stretch without global topology knowledge (Figure 7).

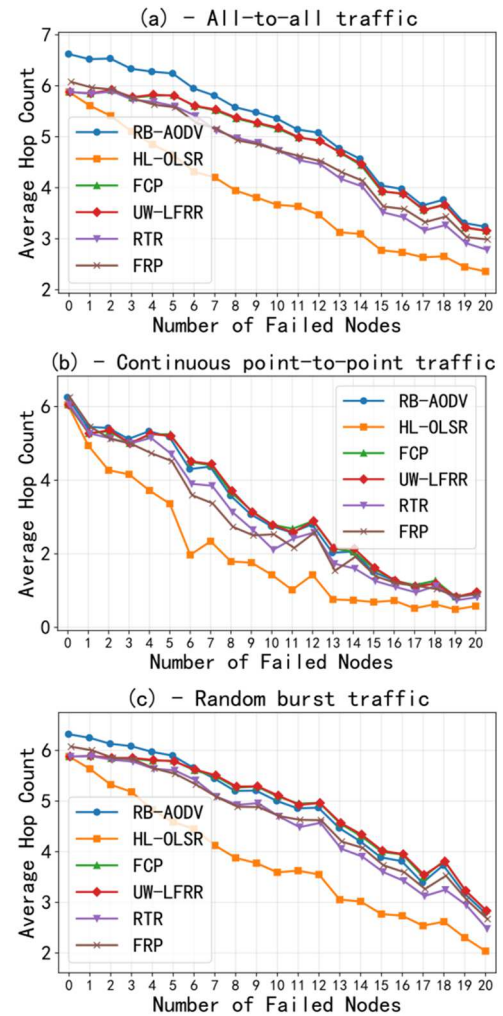


Figure 7. Average hop count under three traffic scenarios

5) Local Computation Overhead: FCP incurs significantly higher computation overhead (up to $4\times$ that of UW-LFRR) due to its per-hop recomputation mechanism. In FCP, each node that detects a failure must dynamically compute a new path based on the accumulated failure information in the packet header, leading to recursive calculations along the route. In contrast, UW-LFRR computes and installs a detour only upon the first packet, allowing subsequent packets to

bypass the failure without further computation. This efficiency is critical for energy-constrained nodes (Figure 8).

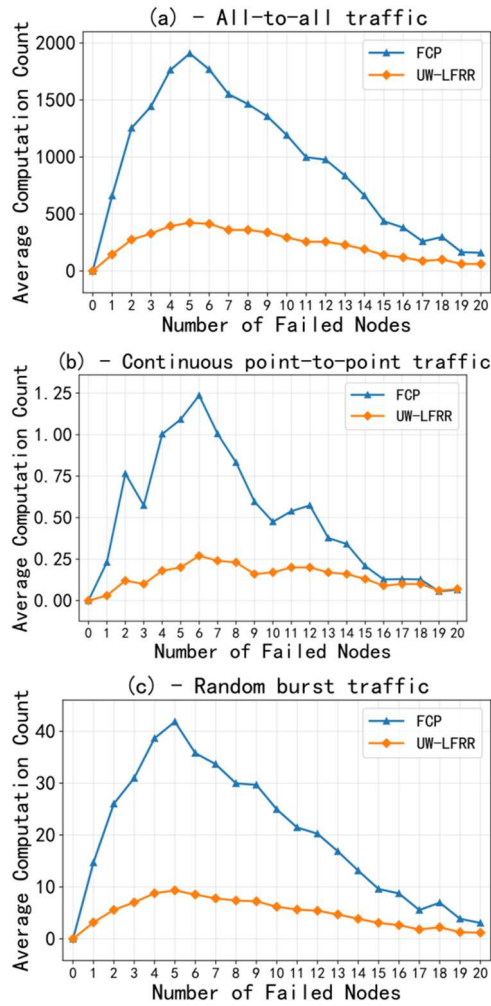


Figure 8. Average computation overhead of FCP and UW-LFRR under three traffic scenarios

C. Discussion

Across all metrics, UW-LFRR demonstrates strong resilience during convergence-blind windows caused by sudden multi-node failures. Its data-plane repair suppresses blackhole propagation and maintains near-normal delay before control-plane updates. Competing schemes—RB-AODV, HL-OLSR, RTR, and FRP—suffer from delays, stale-topology forwarding, probing overhead, or invalid backup activation. While FCP achieves similar robustness, it triggers SPT recomputation at every informed node, incurring high computational cost in energy-limited UANETs. Overall, UW-LFRR provides the most balanced reliability, efficiency, and resource usage among the evaluated recovery mechanisms.

IV. CONCLUSIONS

This paper proposed UW-LFRR, a data-plane-driven fast-recovery mechanism for underwater ad-hoc networks. By combining a purified local topology view, a two-stage relay selection strategy, and a first-packet paving mechanism, UW-

LFRR repairs multi-node failures within the data plane and eliminates convergence blind zones without network-wide flooding. The design guarantees loop-free forwarding with bounded path stretch and keeps computation and state overhead proportional to the size of the affected branch-off region rather than the entire network. NS-3 simulations confirm that UW-LFRR achieves high packet delivery, low latency, and very low control and computation overhead compared with representative UANET routing and recovery protocols, making it a practical and scalable basis for resilient underwater networking.

ACKNOWLEDGMENT

This work was supported by the self-deployed project of the Institute of Acoustics, Chinese Academy of Sciences (Project No. E555180401).

REFERENCES

- [1] A. Khan, I. Ali, A. Ghani, N. Khan, M. Alsaqer, A. U. Rahman, and H. Mahmood, "Routing protocols for underwater wireless sensor networks: Taxonomy, research challenges, routing strategies and future directions," *Sensors*, vol. 18, no. 5, p. 1619, 2018.
- [2] Y. Zhang, S. Wang, M. K. Heinrich, X. Wang, and M. Dorigo, "3D hybrid formation control of an underwater robot swarm: Switching topologies, unmeasurable velocities, and system constraints," *ISA Transactions*, vol. 136, pp. 345-360, 2023.
- [3] N. R. Rypkema and H. Schmidt, "Distributed Autonomy and Formation Control of a Drifting Swarm of Autonomous Underwater Vehicles," Ph.D. dissertation, Joint Program in Applied Ocean Science and Engineering, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, Cambridge, MA, USA, 2015.
- [4] X. Wang, Y. Su, and R. Fan, "SFDM: A time-synchronization-free detection mechanism for mobile underwater acoustic sensor networks," *Ocean Engineering*, vol. 311, p. 118983, 2024.
- [5] S. A. Hoff, J. Matouš, D. Varagnolo, and K. Y. Pettersen, "Communication-aware formation control for networks of AUVs," *European Journal of Control*, vol. 80, p. 101062, 2024.
- [6] Y. Yang, Y. Xiao, and T. Li, "A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 815-841, 2021.
- [7] J. Cui, T. Lv, J. Wang, Q. Tian, Y. Yao, and Z. Yan, "Research on Underwater OLSR Routing Protocol Based on Optimized MPR Nodes," in *2025 8th World Conference on Computing and Communication Technologies (WCCCT)*, 2025, pp. 133-137.
- [8] Q. Zhou, H. Huang, Y. Li, and L. Yin, "Quickening rebuilding routing protocol for underwater acoustic network," *Journal of Applied Acoustics*, vol. 33, no. 3, pp. 196-201, 2014.
- [9] H. Alwan and A. Agarwal, "A survey on fault tolerant routing techniques in wireless sensor networks," in *Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications*, 2009, pp. 366-371.
- [10] M. Chiesa, A. Kamisiński, J. Rak, G. Rétvári, and S. Schmid, "A survey of fast-recovery mechanisms in packet-switched networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1253-1301, 2021.
- [11] A. Kamisiński, "Evolution of IP fast-reroute strategies," in *Proceedings of the 2018 10th International Workshop on Resilient Networks Design and Modeling (RNDM)*, 2018, pp. 1-6.
- [12] S. Riaz, M. Rehan, T. Umer, M. K. Afzal, W. Rehan, E. U. Munir, and T. Iqbal, "FRP: A novel fast rerouting protocol with multi-link-failure recovery for mission-critical WSN," *Future Generation Computer Systems*, vol. 89, pp. 148-165, 2018.
- [13] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica, "Achieving convergence-free routing using failure-

- carrying packets," ACM SIGCOMM Comput. Commun. Rev., vol. 37, no. 4, pp. 241-252, 2007.
- [14] Q. Zheng, G. Cao, T. La Porta, and A. Swami, "Optimal recovery from large-scale failures in IP networks," in Proceedings of the 2012 IEEE 32nd International Conference on Distributed Computing Systems, 2012, pp. 295-304.