Multipath Routing with Limited Cross-Path Interference in Underwater Networks

Saiful Azad, Paolo Casari, *Senior Member, IEEE*, Michele Zorzi, *Fellow, IEEE*

***Abstract*—Multipath routing protocols trade off the effort of replicating data packets through multiple routes for, e.g., improved delivery ratio or end-to-end delay. These advantages are especially valuable for those underwater networking applications where reliable data delivery justifies a higher resource consumption. In this letter, we argue that choosing multipath routes according only to the node- and link-disjoint paradigms may lead to excessive interference in underwater networks, even in the presence of MAC protocols based on interference avoidance. We show that it is more convenient to directly choose multipath routes that cause little interference to one another, and propose a multipath routing protocol that distributedly implements this concept. We simulate our solution in underwater network scenarios, and show that it achieves better packet delivery ratio and fewer interference-induced packet losses with respect to standard multipath routing approaches, even when the latter are stacked on top of interference-avoiding MAC protocols.**

I. Introduction

Underwater acoustic networking can potentially support a variety of applications, from long-term environmental monitoring to coastal and open sea area security. Many of these applications require a certain degree of reliability and timeliness, to be achieved via a combination of robust point-to-point communications [1] and networking protocols. Where the communication system is fixed, the design of the network protocols must be steered in order to achieve the level of performance sought, keeping in mind that underwater acoustic communications pose mainly three challenges to network protocol design: i) the propagation delays over underwater links are typically comparable to or higher than packet transmission times, which in turn tends to impair handshaking procedures due to the long channel vulnerability time that results; ii) transmission is an energy-hungry operation: therefore, it is important to arrange communications so as to avoid useless packet replication and excessive interference among concurrent transmissions, which would lead to collisions, packet losses, and ultimately energy wastage; iii) underwater links are unlikely symmetric, which poses a further challenge to routing protocols. Issue i) can be dealt with as suggested in [2], where it is noted that collision-avoidance Medium Access Control (MAC) schemes based on RTS/CTS handshakes lead to low throughput with respect to a simple Carrier-Sense Multiple Access (CSMA) protocol, due to the long handshake duration and to the collision of control messages.

The novelty of our paper lies in dealing with issues ii) and iii) via a specific multipath routing scheme. Our protocol is based on a typical preliminary path discovery phase that inherently discards routes with unidirectional links, complemented by a strategy to limit the cross-path interference among the discovered multipath routes. We assume no a-priori knowledge of the network topology, hence convenient multiple paths are to be discovered on demand. Our mechanism to do so does not increase the signaling overhead. We realize that transmitting over multiple paths may lead to an overall increase of the bandwidth consumption in the network, during both path discovery and packet transmission. On the other hand, for some applications this is a very effective way to provide the required level of reliability, if not the only one. In fact, we show that our protocol, stacked on top of a simple CSMA MAC scheme, achieves better performance than traditional node- and link-disjoint multipath routing approaches, even when the latter are operated along with a MAC protocol based on collision avoidance.

S. Azad is with the Department of Computer Science, American International University–Bangladesh, Kemal Ataturk Avenue, Banani, Dhaka-1213, Bangladesh (email: sazadm684@aiub.edu). P. Casari (corresponding author) and M. Zorzi are with the Department of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova, Italy (e-mail: {casarip,zorzi}@dei.unipd.it) and with Consorzio Ferrara Ricerche, via Saragat 1, 44122 Ferrara, Italy. This work has been partially supported by the Italian Institute of Technology under the project SEED framework (NAUTILUS project) and by the US Office of Naval Research under Grant no. N62909-14-1-N127.

II. Related work

Multipath routing is typically approached by discovering link- or node-disjoint routes [3] via some knowledge of the network topology [4]. However, administering the transmission of packets through multiple known paths is as important as ensuring disjointness. For example, two transmissions over node-disjoint paths located physically near each other would likely lead to mutual interference. Most multipath protocols for terrestrial networks are designed by assuming interference-free transmissions at the link level. These are achieved, e.g., via a specifically designed Medium Access Control (MAC) protocol [5]–[7]. However, underwater propagation delays expose MAC protocols to long vulnerability periods [8], making it difficult to achieve collision-free communications without sacrificing throughput. If collisions cannot be assumed to be negligible, most approaches to multipath routing would perform much worse. For example, Multipath-DSR (M-DSR) [5] and Multipath-AODV [6], which extend their respective single-path versions by selecting multiple link-disjoint routes, would suffer from the loss of both data and control packets. Split Multipath Routing (SMR) [7], which requires an even higher control overhead, would be affected by the same problem.

A few multipath routing protocols have been specifically designed for underwater scenarios. For example, in [9] the authors detect multiple paths to two sinks via periodic sink beacons, and send data packets via source routing, where every data packet carries the description of its own path. However, the protocol proposed in [9] assumes bidirectional links, and becomes inefficient in the presence of significant link asymmetries. A multipath routing protocol for time-critical applications in underwater networks is proposed in [10]. The protocol employs node-disjoint multipath routing and packet combining at the destination in place of hop-by-hop retransmissions to privilege fast responsiveness. A power control technique at the physical layer complements the routing strategy. Random linear network coding is considered along with multipath routing in [11] to help recover transmission errors. Finally, an underwater multipath routing protocol is designed in [12] with the aim to limit the delivery delay incurred by data packets. This protocol employs locally node-disjoint routes and coordinates the transmission among relays having the same hop count. Unlike [9]–[12], in this work we explicitly target cross-path interference and propose an effective scheme to mitigate it, as discussed in the following section.

III. Multipath Routing with Limited Cross–Path Interference

Our protocol performs a source-initiated path discovery similar to AODV's and DSR's [13]. A node that knows no valid route towards its destination floods a `Path Discovery` packet, and waits for `Path Reply` packets. The `Path Discovery` describes the source and destination for which the routes are sought, carries a sequence number to discriminate among different requests for the same source-destination pair, lists the IDs of the nodes which forwarded the `Path Discovery` so far and maintains a Time-To-Live (TTL) field and a transmission timestamp. The TTL specifies the maximum number of hops the `Path Discovery` packet is allowed to travel, in order to reduce the scope of flooded requests. Every node that rebroadcasts the `Path Discovery` puts its own address in the hop list so that a `Path Reply` can be routed reversely back to the source. The same list is checked by intermediate relays in order to discard `Path Discovery` packets describing non-link-disjoint paths (up to the current relay) with respect to already processed `Path Discovery`s carrying the same sequence number and source-destination pair. Note that unlike in [6], where the `Path Discovery` keeps track only of the address of the second to last hop it travels, in this paper we keep track of its full path: although this increases the size of the `Path Discovery` header [9], it also makes it possible to safely achieve link disjointness.

The destination collects `Path Discovery` packets from the same source for a fixed amount of time, and sorts them in order of increasing end-to-end (E2E) delay required to travel from the initial source up to the destination.[1] Node-disjoint paths are singled out via a standard search algorithm [3] by giving priority to lower E2E delay paths and, *for each of them*, a `Path Reply` is sent through the reverse of the route stored in the respective `Path Discovery`. This procedure allows the network to inherently avoid paths with unidirectional links. Note that, in the `Path Reply`, the transmission timestamp is substituted by the E2E delay measured by the destination.

The key point of our protocol is that, in an underwater environment, node-disjointedness does not guarantee that two paths can be used in parallel, due to possible mutual interference. For this reason, our routing technique takes into account cross-path interference explicitly as follows, when establishing multiple paths. Before forwarding a `Path Reply` to the source, every relay checks if some of its neighbors had already transmitted a `Path Reply` for the same `Path Discovery` sequence number and source-destination pair. In this case, the relay flags the `Path Reply` accordingly. This procedure makes it possible to detect node-disjoint routes whose nodes are located physically close to one another, hence likely prone to mutual interference as the routes are traveled by transmitted packets. The source collects `Path Reply` packets, discards flagged ones, and uses *all* the routes described in all `Path Reply`s that survive as multiple paths towards its destination. If all are flagged, the `Path Reply` with the lowest E2E delay is employed. This technique is completely distributed and requires no action other than overhearing `Path Reply` transmissions. In addition, it transfers to the routing level some collision avoidance capability that would be inconvenient at the MAC level as per the discussion in Section I, and as proven by the results in Section IV. The route information is finally stored in a local table by the source in the form of (*source*, *destination*, *next_hop*, *expiration_time*) entries. The table is maintained via the periodic transmission of `Hello` packets. A next hop is assumed to be unavailable if no `Hello` is received for three transmission periods: in this case, a `Neighbor Unavailable` message is propagated in the network.

We name our protocol L-CROP (short for Limited CROss-Path interference). The following section discusses its performance compared to the standard multipath link-disjoint (MLD) and node-disjoint (MND) routing approaches [3]. We include a single-path routing protocol (SP) in the comparison: for fairness, SP employs the same path discovery process as L-CROP, MND and MLD, and finally chooses the route with the lowest E2E delay.

All protocols are simulated in conjunction with both the CSMA and the MACA-TR [14] MAC protocols. The latter is based on an interference avoidance mechanism, which can be used to reserve the channel for the transmission of up to 5 packets, and should help reduce the mutual interference caused by simultaneous transmissions over multiple paths, while at the same time compensating the handshake overhead by transmitting multiple data packets per channel access. Counter-intuitively, we will demonstrate that the simple and practical solution offered by L-CROP in conjunction with CSMA represents a much better tradeoff.

IV. Simulation scenarios and results

We simulate all protocols using the ns2/NS-MIRACLE-based DESERT Underwater network simulation framework [15], along with the WOSS libraries [16], which enable the simulation of realistic underwater channels via the Bellhop ray tracing software [17]. Based on the channel realizations provided by WOSS, DESERT computes the signal-to-interference-and-noise ratio (SINR) for every received packet by assuming the interference process to be Gaussian and white; the SINR is then used to compute the probability of bit error (in this case we consider a BPSK modulation scheme), whence the packet error probability is derived by assuming independent errors across the packet. With reference to Fig. 1, we consider a network of 12 bottom-mounted nodes deployed in an area of 9 km $\times$ 16 km, located near the coordinates $(55.51°\text{N}, 6.14°\text{E})$. The area is divided into 12 cells; one node is placed at random within each cell. Two sinks are located at opposite sides of the network area, at a random location along one of the short sections. The sinks are assigned the same anycast address, so that one path discovery process can discover routes towards both sinks. We consider two use cases for this network: *Case 1* (environmental monitoring), where the nodes generate packets according to a Poisson process

[1]This requires the source and destination to at least loosely share a common time reference.

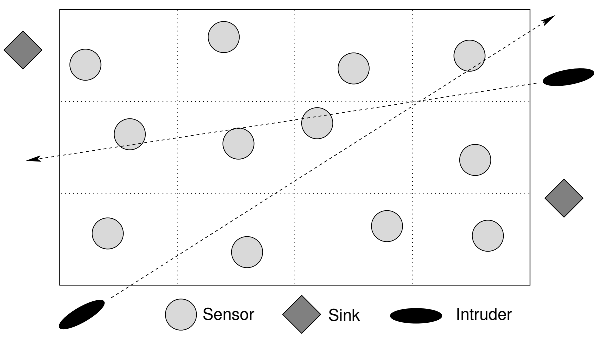

Figure 1. Underwater network with 12 nodes and 2 sinks. Two intruders cross the area, following a linear trajectory.

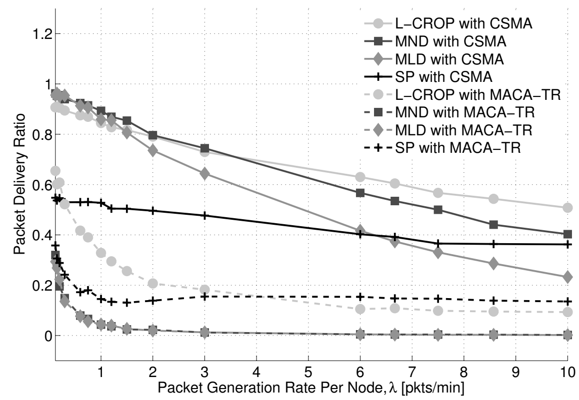

Figure 2. PDR vs. λ, *Case 1*.

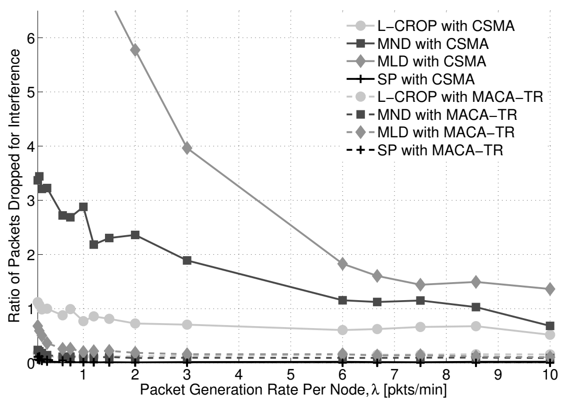

Figure 3. PDIR vs. λ, *Case 1*.

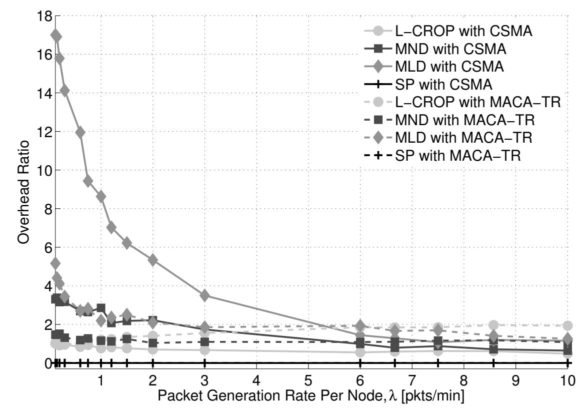

Figure 4. Overhead ratio vs. λ, *Case 1*.

of given rate λ; *Case 2* (event mode), where two "intruders" cross the network area following a linear trajectory with a random direction, and generate noise of given level within the communications band; an intruder is detected by a network node when it enters a detection range of 2 km from that node; upon detection, a node generates packets at a rate of $\lambda = 2$ packets per minute times the number of intruders within its detection range. The data packets payload is 512 bits. A packet is considered delivered if any sink receives it correctly within 300 s of its generation. The nodes communicate in the 4-8 kHz band, at a bit rate of 256 bps. The transmit source level is set to 180 dB re 1 μPa^2 at 1 m from the source. The results are obtained by averaging over 100 realizations of the network topology.

We start from *Case 1*. Figs. 2 and 3 respectively show the packet delivery ratio (PDR, defined as the number of unique packets correctly delivered to any sink divided by the total number of packets generated), and the packets dropped for interference ratio (PDIR, defined as the number of packets lost due to interference from concurrent transmissions divided by the total number of packets generated). Unlike the PDR, the PDIR numerator includes relayed packets. The curves are drawn as a function of the packet generation rate λ, in packets per minute per node. In both figures, L-CROP performs generally better than both MLD and MND, due to its capability to single out routes that do not interfere too much with one another. We observe that the protocols rank roughly depending on the number of routes selected after the path discovery process: MLD discovers the largest number of routes, which leads to many packet replicas and would imply the highest reliability. However, the opposite is in fact true, as these transmissions create significant interference to one another (Fig. 3), and the overall effect is a low PDR (Fig. 2). Hence, MND (which discovers fewer routes) outperforms MLD. L-CROP, which poses further constraints on cross-path interference, outperforms both for $\lambda > 3$, and has comparable performance otherwise. This argument is supported by Fig. 4, which reports the overhead ratio, defined as the average number of additional copies received per correctly received packet. SP, instead, remains limited to a lower PDR by the lack of redundancy due to single-path transmissions. Notably, all protocols achieve a worse PDR when coupled with MACA-TR. This happens regardless of the improved level of transmitter-side coordination achieved by MACA-TR, and testified by the lower PDIR in Fig. 3. The main reason behind this result is that MACA's handshaking procedure increases the channel access delay, and thus causes some packets to exceed their delivery deadline and to be counted as lost. While the few packets that make it enjoy a low level of interference, this is not sufficient to achieve a sufficiently high PDR when using MACA-TR, as seen from Fig. 2. We also observe that the PDIR decreases for increasing packet generation rate, because deafness to incoming transmissions (e.g., when the desired receiver is also transmitting) becomes a significant reason for packet loss as well. To conclude this first discussion, we note that SP incurs a generally low PDR; however, the lack of contention from different multipath routes makes it achieve a higher PDR than both MLD and MND

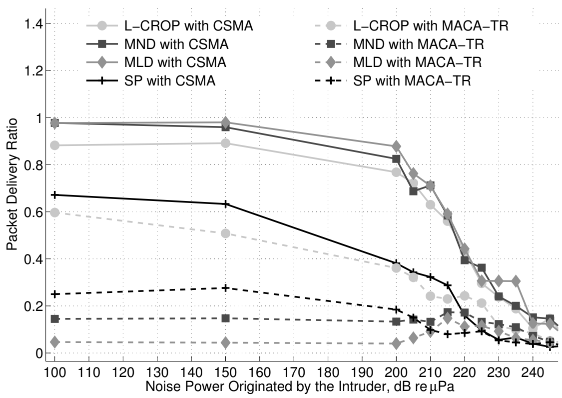

Figure 5. PDR vs. intruder noise level, *Case 2*.

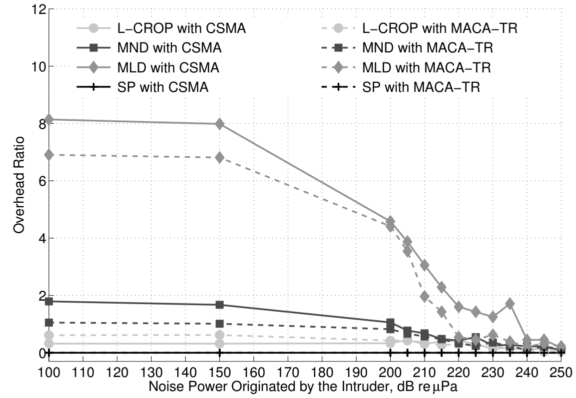

Figure 6. PDIR vs. intruder noise level, *Case 2*.

when coupled with MACA-TR, and to slightly outperform L-CROP for sufficiently high λ. In any event, all protocols perform better when coupled to CSMA: this is in line with the observation in Section I, which spurred the thrust behind this paper.

We now turn to *Case 2*. Figs. 5 and 6 respectively show the PDR and PDIR of all routing algorithms. Unlike in *Case 1*, the performance metrics are plotted as a function of the noise level originated by one intruder within the communications band (we recall that the packet generation rate is not controllable in *Case 2*). When the noise level is sufficiently low, the PDR of all protocols levels between 0.9 and 1: this is due to the event-based packet generation, which makes transmissions more erratic, hence subject to lower cross-interference despite the high packet generation rate of 4 pkt/min/intruder in range. For this same reason, the PDR achieved by L-CROP is similar to that of MLD and MND: however, L-CROP requires much fewer packet replicas to achieve this result, in line with the outcome of simulations for *Case 1*. The PDR of SP with CSMA is lower than observed for the other protocols. The use of MACA-TR, on the contrary, significantly increases the PDIR for all protocols (due to many data-data and control-data packet collisions), and even gives SP some advantage over MLD and MND in terms of PDR. Again, such PDR is a combination of a positive factor (lack of cross-path interference) and a negative factor (lack of redundancy from multipath transmissions). For high intruder noise level ($\geq$ 220 dB re μPa^2) L-CROP performs as well as MLD and MND, due to the higher cross-path interference among routes discovered by the latter approaches. This is confirmed by Fig. 6. In any event, the high level of noise causes all protocols to achieve worse performance in this regime. Finally, we observe that all protocols achieve better performance when run on top of the CSMA MAC protocol rather than MACA-TR. This is consistent with the conclusions related to *Case 1*.

V. Conclusions

In this letter, we propose L-CROP, a multipath routing protocol where the discovered routes cause limited interference to one another, thanks to a distributed algorithm based on the overhearing of `Path Reply` packets. L-CROP (used along with CSMA), shows the ability to cope with typical underwater networking issues. The main features of L-CROP are that i) it operates on top of a MAC protocol which achieves better performance in underwater networks, with respect to handshake-based protocols [2]; ii) it avoids wasting energy in transmissions that would interfere with each other, by limiting the cross-interference of discovered routes; and iii) it makes it unlikely that routes with unidirectional links are employed. This makes our solution a good candidate for the constrained scenarios found in several underwater networking applications.

References

[1] M. Chitre, S. Shahabudeen, and M. Stojanovic, “Underwater acoustic communications and networking: Recent advances and future challenges,” *Marine Tech. Soc. Journal*, vol. 42, no. 1, pp. 103–116, 2008.

[2] P. Casari and M. Zorzi, “Protocol design issues in underwater acoustic networks,” *Elsevier Comp. Commun.*, vol. 34, no. 17, pp. 2013–2025, 2011.

[3] R. G. Ogier, V. Rutenburg, and N. Shacham, “Distributed algorithms for computing shortest pairs of disjoint paths,” *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 443–455, Mar. 1993.

[4] C. Lal, V. Laxmi, and M. S. Gaur, “A node-disjoint multipath routing method based on AODV protocol for MANETs,” in *Proc. IEEE AINA*, Fukuoka, Japan, Mar. 2012.

[5] A. Nasipuri, R. Castaneda, and S. R. Das, “Performance of multipath routing for on-demand protocols in mobile ad hoc networks,” *ACM/Kluwer MoNet*, vol. 6, no. 4, pp. 339–349, 2001.

[6] M. K. Marina and S. R. Das, “Ad hoc on-demand multipath distance vector routing,” *Wireless Communications and Mobile Computing*, vol. 6, pp. 969–988, 2006.

[7] S. J. Lee and M. Gerla, “Split multipath routing with maximally disjoint paths in ad hoc networks,” in *Proc. IEEE ICC*, Helsinki, Finland, 2001.

[8] A. Syed, W. Ye, B. Krishnamachari, and J. Heidemann, “Understanding spatio-temporal uncertainty in medium access with ALOHA protocols,” in *Proc. of ACM WUWNet*, Montréal, Canada, Sep. 2007.

[9] M. Goetz, S. Azad, P. Casari, I. Nissen, and M. Zorzi, “Jamming-resistant multi-path routing for reliable intruder detection in underwater networks,” in *Proc. of ACM WUWNet*, Seattle, WA, Dec. 2011.

[10] Z. Zhou, Z. Peng, J.-H. Cui, and Z. Shi, “Efficient multipath communication for time-critical applications in underwater acoustic sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 1, pp. 28–41, Feb. 2011.

[11] Z. Guo, B. Wang, P. Xie, W. Zeng, and J.-H. Cui, “Efficient error recovery with network coding in underwater sensor networks,” *Elsevier Ad Hoc Networks*, vol. 7, no. 4, pp. 791–802, Jun. 2009.

[12] Y.-S. Chen, T.-Y. Juang, Y.-W. Lin, , and I.-C. Tsai, “A low propagation delay multi-path routing protocol for underwater sensor networks,” *Journal of Internet Technology*, vol. 11, no. 2, pp. 153–165, Mar. 2010.

[13] C. Perkins, E. Royer, S. Das, and M. Marina, “Performance comparison of two on-demand routing protocols for ad hoc networks,” in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000.

[14] N. Chirdchoo, W.-S. Soh, and K. C. Chua, “MACA-MN: A MACA-based MAC protocol for underwater acoustic networks with packet train for multiple neighbors,” in *Proc. of IEEE VTC Spring*, Singapore, 2008.

[15] R. Masiero *et. al*, “DESERT Underwater: an NS–Miracle-based framework to DEsign, Simulate, Emulate and Realize Test-beds for Underwater network protocols,” in *Proc. IEEE OCEANS*, Yeosu, Korea, 2012.

[16] F. Guerra, P. Casari, and M. Zorzi, “World Ocean Simulation System (WOSS): a simulation tool for underwater networks with realistic propagation modeling,” in *Proc. ACM WUWNet*, Berkeley, CA, 2009.

[17] M. Porter *et al.*, “Bellhop code.” [Online]. Available: http://oalib.hlsresearch.com/Rays/index.html