Proceedings Paper:
Mahieu, Florian, Mitchell, Paul Daniel, Morozs, Nils et al. (2024) Acknowledgement strategies for the JANUS medium access control protocol. In: OCEANS 2024. OCEANS 2024 Halifax, 23-26 Sep 2024 Oceans. IEEE, CAN.

https://doi.org/10.1109/OCEANS51537.2024.10682249

Acknowledgement strategies for the JANUS medium access control protocol

Florian Mahieu*, Paul D. Mitchell, Nils Morozs, Tim C. Tozer
School of Physics, Engineering & Technology
University of York
York, YO10 5DD, United Kingdom
*florian.mahieu@york.ac.uk

***Abstract*—The JANUS protocol, as the first standardized underwater communication protocol, aims to become the reference means of communication between underwater devices. For compatibility and ease of implementation, its initial Medium Access Control (MAC) layer specification is left as plain as possible. In this paper, we present an extension of the JANUS protocol involving data packet acknowledgments (ACKs) and retransmissions to improve the reliability of underwater acoustic networks (UAN) powered by this protocol. This extension will enable more complex applications with back-and-forth messaging or usage of the protocol in more challenging acoustic environments. As part of this work, an initial Logical Link Control (LLC) module and three different ACK integration strategies are proposed that produce the following three JANUS Link Layer protocol variants: JANUS-ACK, JANUS-Quick-ACK (JANUS-QACK), and JANUS-Interrupt-ACK (JANUS-IACK). These three variants of the JANUS protocol enhance it by integrating the ACK functionality at the Link Layer. In this work, we implement and simulate these protocols in multiple scenarios with a range of different parameters to provide a general overview of their comparative performance, as well as comparing them with ALOHA and Time Division Multiple Access (TDMA) benchmark protocols. JANUS-QACK and JANUS-IACK show promising results with up to a 50% throughput improvement compared to JANUS-ACK, and provide a good trade-off amongst throughput, latency and number of re-transmissions required for successful delivery compared with ALOHA and TDMA.**

***Index Terms*—Acknowledgement, Communications, JANUS, JANUS-ACK, JANUS-IACK, JANUS-QACK, LLC, MAC, Underwater Acoustic Network**

I. Introduction

Underwater communication technologies find applications in an increasing number of systems in charge of ocean, sea monitoring or coastal surveillance and monitoring [1], and more generally in underwater acoustic sensor networks (UASN) and the underwater internet of things (UIOT) [2]. They enable remote control, telemetry and cooperation between autonomous underwater vehicles (AUV). However, the underwater environment is a difficult transmission medium for conventional radio-based communication technologies. They suffer from extremely high signal absorption limiting their range in most practical underwater deployment scenarios [3]. Hence alternative communication techniques are adopted, with acoustic technologies being the most practical choice. However, due to the use of acoustic waves with slow propagation speeds, acoustic systems suffer from high latency and low throughput [4]. Moreover, acoustic propagation varies significantly from one underwater environment to another [5]. Conventional communication protocols previously used for radio systems were not designed with these extreme properties in mind, and new solutions and bespoke protocols have to be developed.

Typical underwater acoustic networks implement a simplified network stack stripped to its bare minimum to reduce overheads as much as possible. A key component of these networks is Medium Access Control (MAC), which is in charge of scheduling packet transmissions to provide sufficient throughput with minimal packet loss and latency [6]. Typical MAC protocols found in underwater acoustic networks can be split into two main categories. First are contention-free MAC protocols, in which channel resources are pre-allocated and their availability guaranteed to specific nodes or links. The fundamental contention-free schemes include Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA), and Time Division Multiple Access (TDMA) [7]. In contrast to that, another class of MAC is based on the idea of channel resource contention. In these systems, the channels are either accessed randomly (ALOHA [8]) or with transmission scheduling rules based on carrier sensing [9] or channel reservation with the help of handshaking by exchanging Request-to-Sent (RTS) and Clear-to-Send (CTS) signals prior to any data transmission [10]. These MAC protocols work particularly well in applications requiring little traffic, because at higher loads, due to packet collisions, their efficiency degrades rapidly. They are also useful in situations where flexibility is paramount, especially for networks with dynamic topologies and in vehicle-to-infrastructure applications, since these MAC schemes do not require any kind of clock synchronization or channel resource pre-allocation.

A modern approach for general-purpose underwater acoustic communication involves the JANUS protocol, the first standardized underwater acoustic protocol approved by NATO [11]. As an open standard with a focus on simplicity and interoperability, the ambition behind the JANUS protocol is to become at large the reference underwater acoustic communication system [12] that can integrate itself in a diverse landscape of modem manufacturers and underwater communication technologies. Due to this desire for simplicity, its MAC approach is simple too, and it currently does not specify important

link layer functionality, such as ACKs and retransmission management for reliable communication. As its default MAC, JANUS employs a Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) algorithm combined with a slot election-based binary exponential back-off (BEB) scheme to try to minimize packet collisions [11]. In other words, during back-off, transmissions are not scheduled for a random timeout delay. Instead, a transmission window is randomly elected to begin the transmission. If any energy is detected on the channel before a transmission window is reached, the probability of a successful election for the following transmission windows is reduced until success.

This paper proposes a Logical Link Control (LLC) based extension to the JANUS protocol to enhance it with packet reception acknowledgments (ACK) and add a retransmission mechanism in case of packet delivery failure. Packet loss caused by collisions or corruption can be mitigated with this extension, increasing the reliability of a JANUS-based network. Thanks to that, complex applications with back-and-forth messaging or deployment of networks in more challenging communication settings with heavy signal degradation such a those found in constrained underwater environments [13] become possible. More specifically, in this paper, we describe a simple but modular and parameterized LLC module as well as three JANUS MAC protocol variants, each testing a different LLC-MAC integration strategy to try and optimize ACK packet scheduling and minimize their impact on the overall performance due to the added overhead. These three MACs, JANUS-ACK, JANUS-QACK, and JANUS-IACK, are named after their respective differences in their ACK and data scheduling procedures. To evaluate their relative impact on an acoustic network, we implement and simulate them in various operating scenarios with different settings along with an ALOHA-ACK and TDMA-ACK implementation for reference purposes.

The rest of this paper is organized as follows: section II, defines the network and its components that are used for this study, and section III describes the LLC component that manages ACKs and data retransmissions as well as the different MAC integration strategies used to process ACK packets in each JANUS variant. Section IV presents simulation results for each tested scenario. Finally, section V extracts conclusions from this study and proposes useful directions for further work.

II. Study setup

To obtain an insight into the performance and run-time characteristics of the proposed new MAC protocols and their LLC module, we set up a simulation of a UAN comprising uniformly randomly placed nodes representing a generic oceanic, sea, or coastal area with activity between multiple static devices or AUVs, with different tasks or operations. In this use-case scenario, as shown on Fig. 1, the UAN lets all active devices within hearing range share information about their state and contribute to their situational awareness. In such a network, only small data transfers are expected, but the position and quantity of different nodes can be highly random and dynamic. This kind of situation is perfect for a JANUS-based system as opposed to a contention-free protocol that relies on channel resource pre-allocation.

To represent this use-case scenario, we adopt many instances of a network topology comprised of 10 nodes that are within single-hop range and therefore can all communicate (or interfere) with each other as shown in Fig. 2. The nodes are spread across a square area of 1500 by 1500 meters, resulting in a maximum node distance of 2121 meters, which at a constant sound propagation speed of 1500 m/s, translates into a maximum transmission delay of 1414 milliseconds. For simplicity, and to isolate the effect of the LLC and MAC modules, we assume that acoustic propagation phenomena

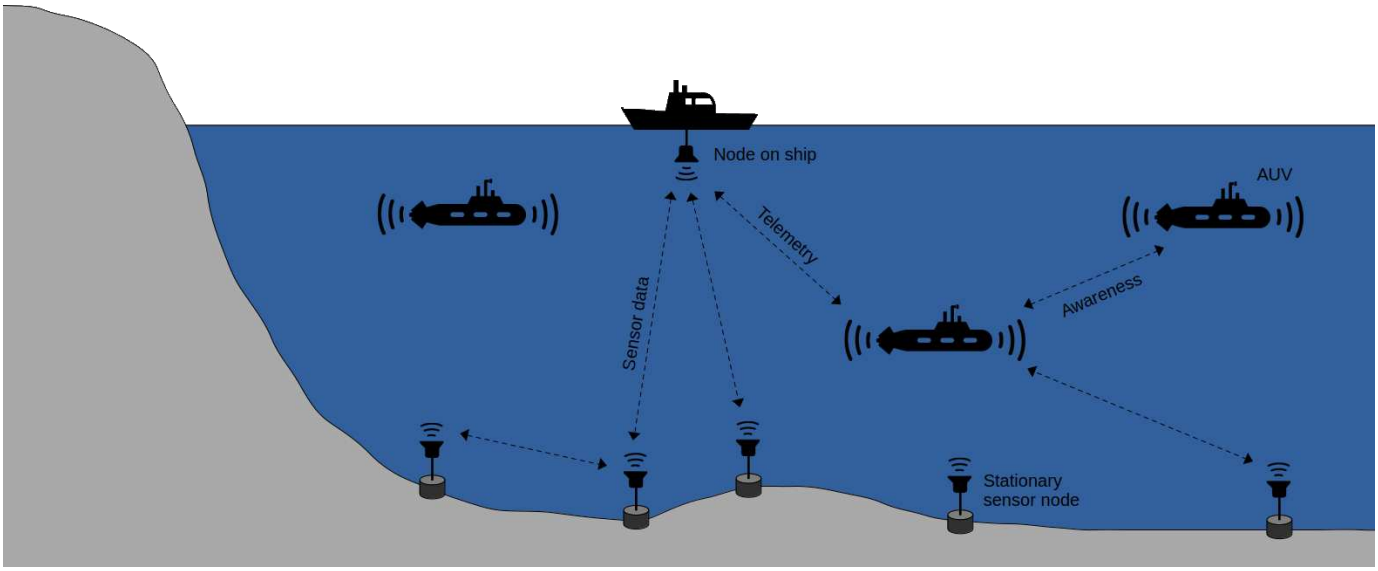


Fig. 1: Example of an underwater acoustic network application

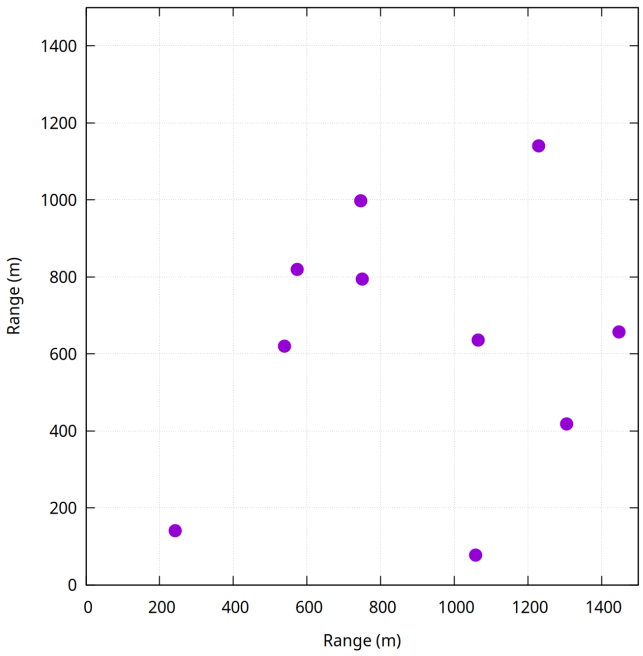


Fig. 2: Top view representation of the fully connected network topology with randomly spread nodes used in the study, all nodes can hear and transmit to each other.

such as multipath fading, Doppler shift, and variable sound speed profiles are dealt with by the physical layer. For the same reasons, the modems of the simulated nodes are all omnidirectional half-duplex transceivers and packet collisions are modelled as a boolean check: if packets at the receiver overlap in time, then all afflicted packets are assumed to be lost. Thanks to these simplifications, the only remaining source of data loss is packet collisions which are a direct consequence of the MAC protocols.

We also assume that each node implements a generic application layer in charge of generating packets at random times following the Poisson distribution [14]. The created data packets are assigned to a single random destination node which cannot be the emitting node itself. In the context of the simulation, the created data packets have a total length of 64 bits and are therefore compliant with the JANUS packet length specified in the standard. This application layer module also acts as the sink for any data packets arriving at their destination node. Only data packets that reach the application layer are counted in network performance statistics like throughput, retransmissions number or unsent packet proportion.

Overall, as shown in Fig. 3 every node in this study implements a compact network stack consisting of an application layer, a data-link layer consisting of an LCC and MAC sub-layers, and a physical layer. Since all nodes can hear each other, routing and a dedicated network layer are outside the scope of this study. So are the presentation, session, and transport layers. All layers and sub-layers within a node are connected via a priority-based packet transmission queue of

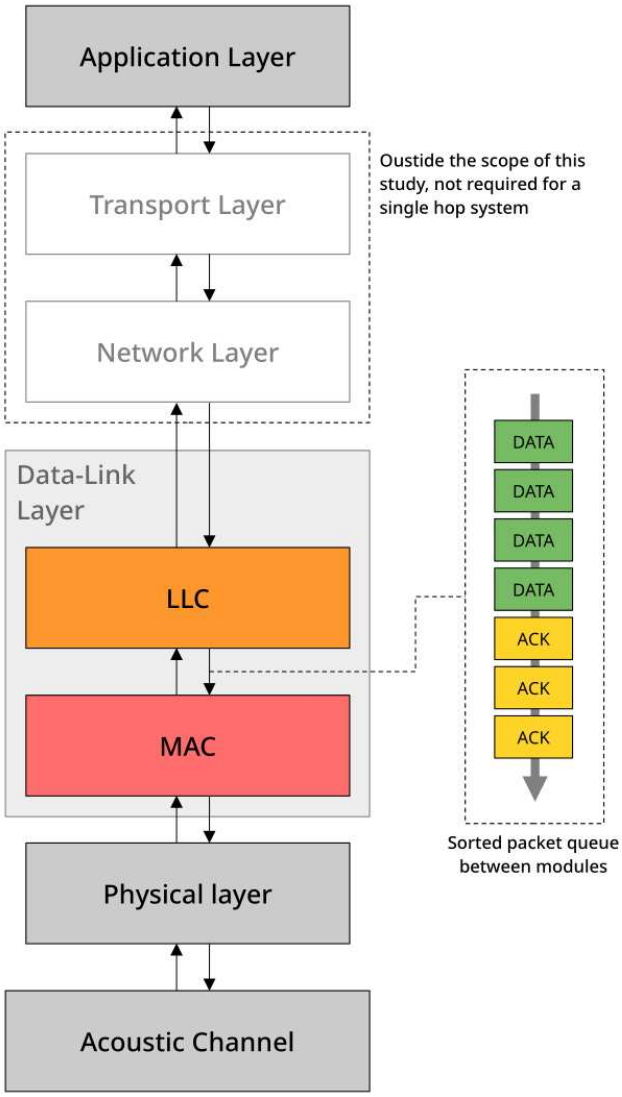


Fig. 3: Representation of the implemented network stack in this study. On the right is a representation of the packet queue present between each layer and sub-layer

unlimited size. Inside the queue, packets are sorted by a priority level assigned depending on their type. ACK packets have a higher priority than data packets. Within a same-priority segment of the queue, packets are sorted and processed using the First-In-First-Out method. Additionally, the data-link layer is capable of probing the physical layer to determine if it's busy receiving or transmitting a signal (carrier sensing).

III. Integrating ACKs into the JANUS Protocol

In this section, we describe the proposed ACK extension to the JANUS protocol. First, we define the component that manages ACKs and data retransmissions: the LLC sub-layer. Then we introduce three integration strategies of the new LLC module into the JANUS MAC. Each of these strategies

provides different ACK scheduling rules at the MAC level, thus modifying the behavior of the network. The resulting protocols, JANUS-ACK, JANUS-Quick-ACK (JANUS-QACK), and JANUS-Interrupt-ACK (JANUS-IACK) are named after these ACK integration strategies.

A. LLC sub-layer

At the heart of the three new JANUS variants is the LLC data-link sub-layer in charge of creating and receiving ACK packets as well as initiating re-transmissions in case of data packet delivery failure or in case of a missing ACK. It does not however provide a handshaking mechanism like RTS-CTS. This component sits on top of the MAC sub-layer. In this study, we construct a generic LLC module with an ACK timeout and a slotted BEB to delay re-transmission attempts of data packets that do not receive an ACK within the timeout period. Its operational steps can be broken down into two algorithms, one for passing packets downstream from the application layer to the MAC, and one for the opposite packet flow direction, from the MAC to the application layer.

The gist of the LLC process is the following: once a source node has a data packet to transmit, it sends it into the LLC's input queue. In turn, the LLC module, if idle, will grab and remove the packet from the queue and hold it until its downstream process is complete. This process, detailed in Algorithm 1, will add a unique packet identifier in the packet's payload and then send it down to the MAC layer to be scheduled for transmission. After that, it waits for a confirmation by the MAC layer that the packet was properly transmitted at the physical layer, which triggers a timeout countdown set at an arbitrary initial value T_t. Because this parameter's optimal value is highly dependent on the scenario, its value will be set and customized in section IV. If a corresponding ACK is not received in time, the module enters a back-off state, and waits for a random number of time slots $S \in [0, 2^R]$, with R being the number of times a packet has been re-transmitted. The length of an individual time slot T_s is set at twice the transmission time of a 64-bit data packet to match the sensing period of the standard JANUS MAC protocol. It then re-sends a copy of the packet to the MAC layer and begins the countdown anew. If the re-transmission limit $R_{max} = 6$ is reached, the packet is abandoned. Instead, this algorithm needs to be interrupted and reset back to its initial state by the upstream process, whose steps are outlined in Algorithm 2. In short, when the data packet created by the source node is received at the destination node, it is directly sent from the physical layer to the LLC layer. If the received packet contains data, its unique identifier is extracted and used to set the payload of a newly created matching ACK packet. The created ACK packet is then sent to the MAC layer to schedule its transmission. Finally, when the source node receives the ACK packet, it will try to match its payload to the packet held its the downstream algorithm. In case of a match, which means that the data-ACK exchange has been successful, the downstream process is reset and a new packet

Algorithm 1 Downstream LLC process

```
Receive data packet from the Application layer and load it into short term memory
Assign a unique packet identifier to the data packet
Initiate re-transmission counter R
while R < R_max do
    Send copy of the data packet to the MAC
    Initiate ACK timeout countdown T_t
    Wait for transmission confirmation by the MAC
    while T_t > 0 do
        Decrement T_t and listen for matching ACK to exit
    end while
    — Back-off sub-procedure start —
    Select random back-off slot S ∈ [0, 2^(R+1)]
    Initiate back-off countdown T_b = S * T_s
    while T_b > 0 do
        Decrement T_b and listen for matching ACK to exit
    end while
    — Back-off sub-procedure end —
    Increment re-transmission counter R
end while
— Past this point, the packet transmission is considered to have failed. —
Abandon packet transmission
Reset and wait for new packet
```

Algorithm 2 Upstream LLC process

```
Receive packet from the MAC
if packet type = data then
    if data packet destination address = node address then
        Generate ACK packet
        Set data packet source address as ACK packet destination address
        Set data packet unique identifier as ACK packet payload
        Send ACK packet to the MAC
        Send data packet to the Application layer
    else
        Discard data packet
    end if
else if packet type = ACK then
    if ACK payload value = data packet unique identifier from algorithm 1 then
        Reset algorithm 1 back to its initial state
    else
        Discard ACK packet
    end if
else
    Discard unsupported packet type
end if
Reset and wait for new packet
```

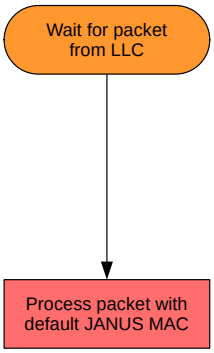
can be processed if any are queued. i.e. this is the *Stop-and-Wait* flow control [15].

B. ACK scheduling strategies

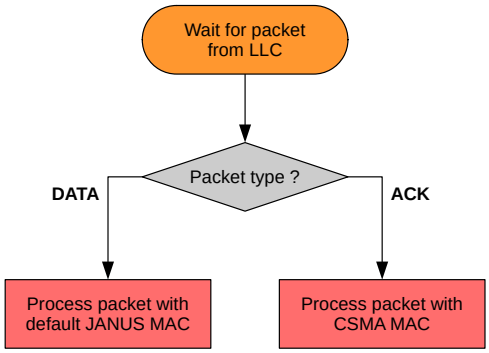
While the LLC component implements ACK features, it is up to the MAC sub-layer to properly schedule data and ACK transmissions. Here we explore three different ACK scheduling methods to be added to the default JANUS MAC protocol. The names of the resulting MACs are named after each ACK scheduling method.

The simplest way to integrate ACKs into the JANUS MAC protocol is to simply schedule ACK packets in the same way as regular data packets, with a full sensing period and potential back-off in case of channel activity. With this method, hereafter called JANUS-ACK, there is no special processing done for the ACK packets. Instead, the MAC will simply deal with all incoming packets from the LLC sub-layer one by one and it will not discriminate between data and ACK packets. In a sense, it is just the default JANUS MAC protocol with the LLC sub-layer added on top of it.

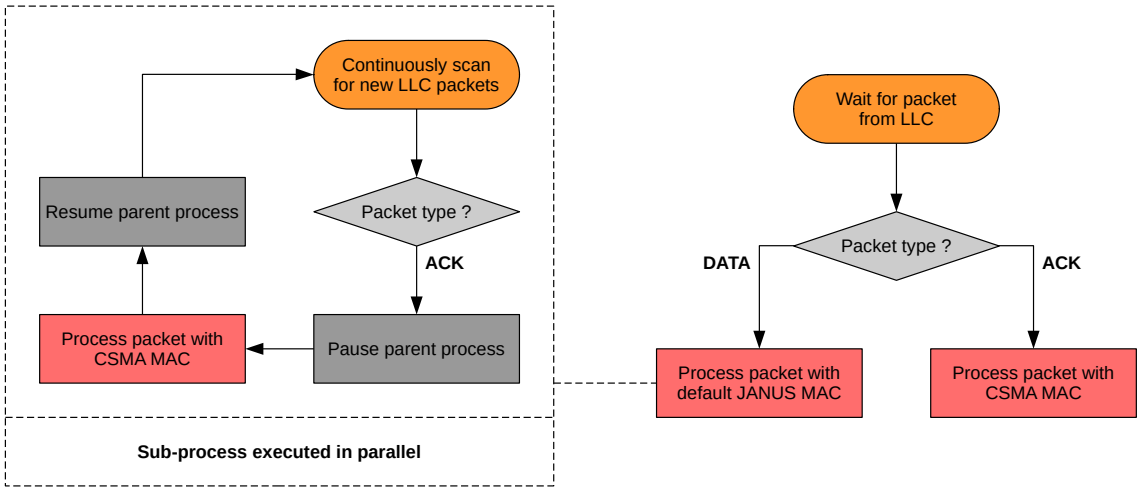
In contrast to that, JANUS-Quick-ACK, or JANUS-QACK, is a proposed hybrid MAC protocol that adds a separate CSMA branch specifically for ACK packets. This means that regular data packets will go through the standard MAC JANUS algorithm while ACK packets will follow a more lightweight CSMA procedure, whereby the transmission is only delayed by immediate activity on the acoustic channel. As soon as the channel becomes free again, the ACK packet is transmitted. Thanks to this modification, delays caused by the sensing period and the back-off of the standard JANUS procedure are



(a) JANUS-ACK



(b) JANUS-QACK



(c) JANUS-IACK

Fig. 4: Representation of the different ACK integration strategies implemented in the three JANUS variants

replaced by a single and immediate check of the physical layer activity for ACK transmissions. If the physical layer is busy receiving or transmitting, the MAC will wait until it's free again, then it will transmit the packet. And if the physical layer is idle, then the ACK packet is sent out immediately. Thus the transmission of ACK packets is accelerated. Thanks to that, the average time it takes for the MAC layer to transmit an ACK is reduced, also reducing the probability of ACK timeout at the node that sent the initial data packet.

Both JANUS-ACK and JANUS-QACK process data and ACK packets sequentially, meaning that these MACs will wait for the physical layer to transmit a given packet, before starting to process a new one. This may cause an issue with ACK timeouts, even with the accelerated JANUS-QACK MAC variant. When a node is waiting for an ACK after data transmission, it may happen that the receiving node, instead of transmitting the appropriate ACK, is busy with a data packet it generated itself. And because the JANUS MAC protocol back-off procedure is based on slot election following a probability rule, there can be situations when an ACK is not sent out in time because the MAC layer was busy processing the previous data packet. To this end we propose JANUS-IACK, or JANUS-Interrupt-ACK, which addresses this issue by allowing ACK packets to interrupt the standard JANUS contention procedure. This means that even if the MAC layer is busy waiting to transmit a data packet (sensing period + back-off), if it detects an ACK packet from the LLC waiting in its queue, it can pause the contention procedure of the data packet, send out the ACK following the simple CSMA procedure described for JANUS-QACK, and then resume working with the data packet as if nothing happened in-between. There can be as many interruptions as there are ACK packets that need to be sent out, therefore higher average data packet delay is expected, but the experienced ACK delay in exchange should be minimal, and thus help minimize the regularity of ACK time-outs.

Fig. 4 visualizes the ACK packet scheduling differences between the three new JANUS MAC protocols variants, and how each protocol integrates with the LLC component. As shown in Fig. 4c, the JANUS-IACK representation is split into two processes. A main one, similar to JANUS-QACK which processes packets one by one. And a secondary one that only runs when the main process is executing the "DATA" branch. This second child process continuously scans for incoming ACK packets and can pause or resume the MAC process of its parent.

IV. Simulation study

We simulated a large number of scenarios similar to the one shown in Fig. 2 and within the bounds defined in Section II but with randomized node positions. These simulations were split into two batches. The first batch, utilises 16-bit 'performance' ACKs, while the second batch utilises 64-bit ACKs. The latter is considered for compatibility purposes with the existing JANUS data packet structure. That way, if ACK packets are received by a JANUS node not supporting the extension, they can still be properly detected by that node and then just be silently discarded without raising an error at the physical layer. The modem bit rate is set to a low 100-bit/second to represent a worst-case communication scenario in which ACKs are useful. This is also the main reason as to why the offered traffic values in the subsequent plots are expressed in packets/min/node since a single data packet already consumes 640 milliseconds of channel time. With 64-bit data and ACK packets, it takes a minimum of 4108 milliseconds to do a full data + ACK exchange between two nodes at the maximum distance defined in Section II. Therefore, the ACK timeout time T_t mentioned in section III-A is set to 20 seconds, approximately 5 times the data + ACK exchange delay to leave enough room for the ACK transmission to avoid as many missed ACKs as possible due to heavy node activity.

Additionally, for benchmark comparison, we also add to the study ALOHA-ACK and TDMA-ACK MAC protocols. Both of these protocols will make use of the same LCC module as the studied JANUS MAC variants, with the same parameters. It should be noted that in the following simulations, TDMA-ACK MAC will use 10 slots (exactly one per node) with an individual length of 3700 and 4108 milliseconds (for 16-bit and 64-bit ACKs respectively), just big enough to accommodate both data and 16-bit-ACK transmission within the same slot.

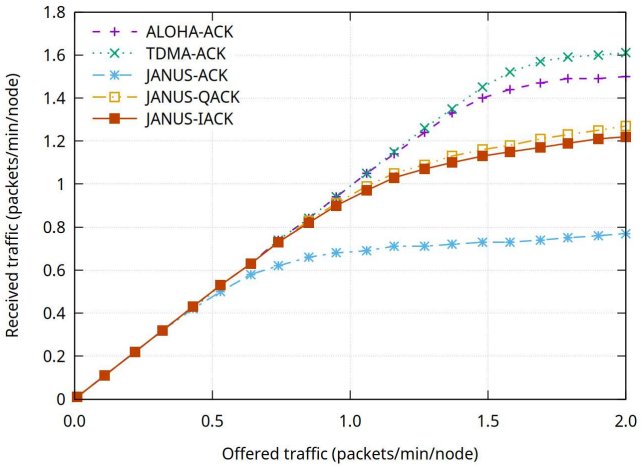
A. Throughput

Fig. 5a and Fig. 5b reveals the average data throughput of ALOHA-ACK, TDMA-ACK, JANUS-ACK, JANUS-QACK, and JANUS-IACK at a range of traffic loads with 16-bit or 64-bit ACKs. Only the data packets that successfully reach the application layer at the intended destination node are counted. At first glance, JANUS-ACK offers a comparatively low throughput, especially compared to ALOHA-ACK and TDMA-ACK. The main reasons for that are the sensing period and potential back-off delays from the JANUS MAC protocol, but this performance loss is partially mitigated by the ACK acceleration implemented in JANUS-QACK and JANUS-IACK.

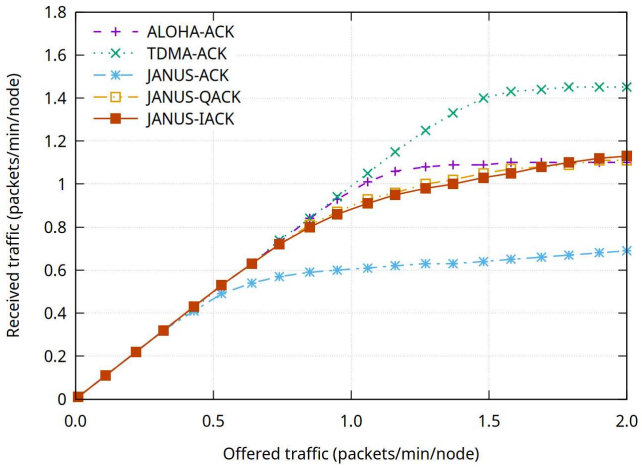
The throughput also decreases with bigger ACK packets. From 16-bit to 64-bit ACKs. At an offered traffic load of 2 packets/min/node, a throughput reduction of 12.5%, 8.3%, and 8.3% is visible for JANUS-ACK, JANUS-QACK and JANUS-IACK respectively. However, ALOHA-ACK gets hit with a bigger 26.7% throughput loss, and due to that, performs no better than JANUS-QACK and JANUS-IACK at higher loads. This stark difference in throughput loss between the JANUS variants and ALOHA-ACK is thanks to the core JANUS-MAC protocol that helps mitigate collisions. Collisions that happen more frequently with larger packets or ACKs.

B. Re-transmissions

The additional throughput capability offered by faster ACKs, is gained at the expense of increased packet loss caused by the ACKs interfering more often with other packets on the network. The result is twice the number of re-transmissions for JANUS-QACK and JANUS-IACK compared with with

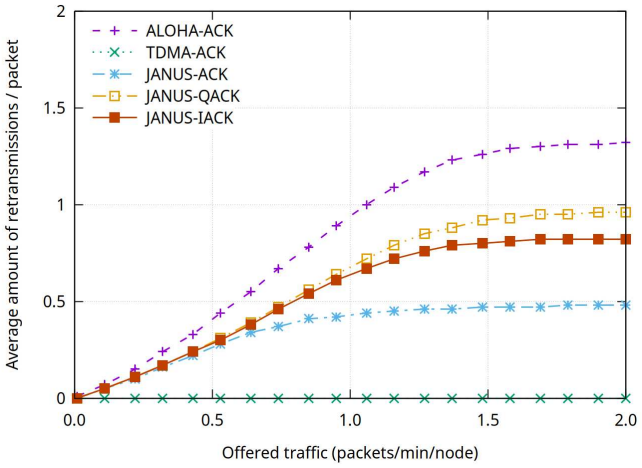


(a) 16-bit ACKs

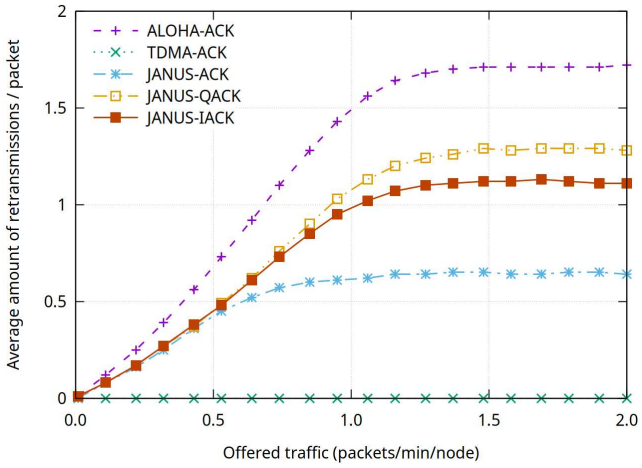


(b) 64-bit ACKs

Fig. 5: Throughput (3 hours of simulated time)

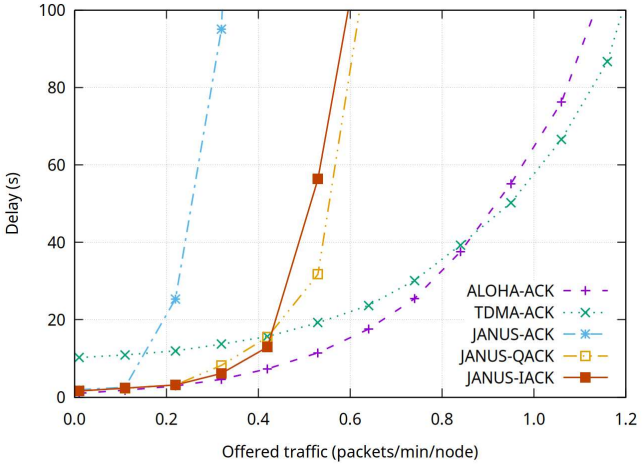


(a) 16-bit ACKs

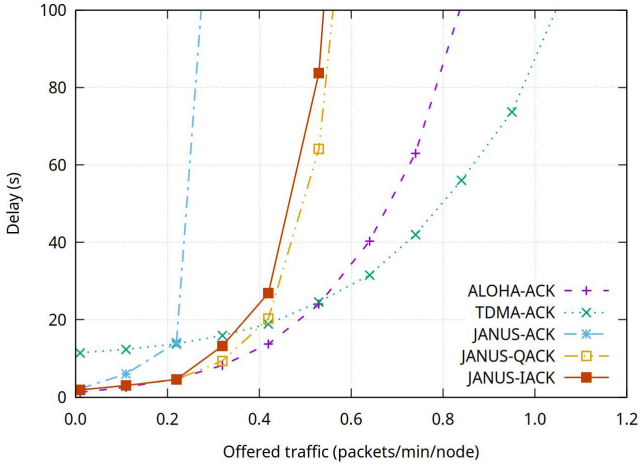


(b) 64-bit ACKs

Fig. 6: Re-transmissions (3 hours of simulated time)

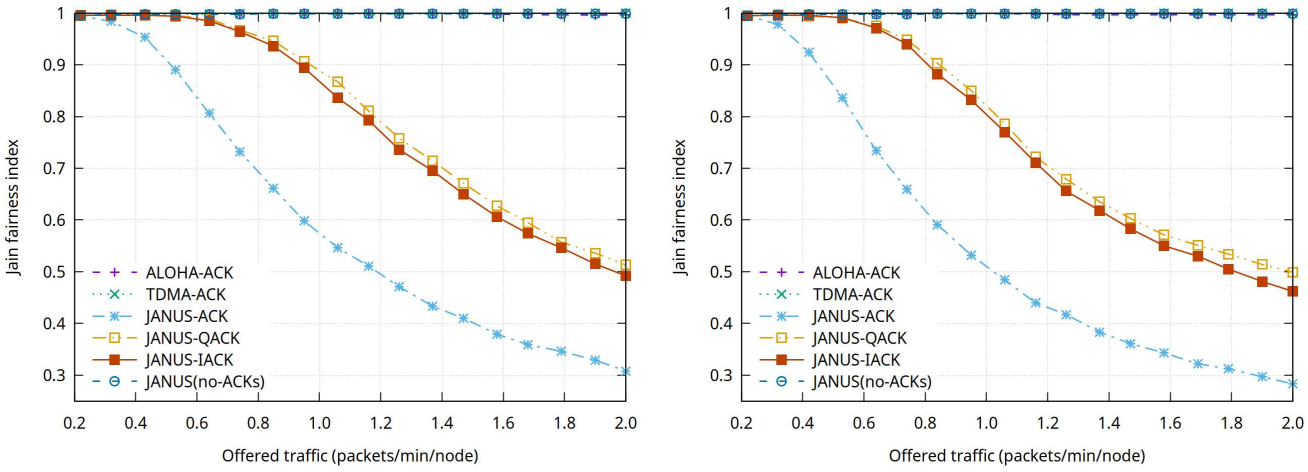


(a) 16-bit ACKs



(b) 64-bit ACKs

Fig. 7: Latency (3 hours of simulated time)



(a) 16-bit ACKs

(b) 64-bit ACKs

Fig. 8: Jain's fairness index (15 hours of simulated time)

JANUS-ACK as shown in Fig. 6a and Fig. 6b. JANUS-IACK is also slightly better than JANUS-QACK in that regard. But more importantly, on this metric, all JANUS variants show big improvements compared to ALOHA-ACK that re-transmits on average up to 2.6 more times than JANUS-ACK with 16-bit ACKs, and up to 2.9 more times than JANUS-ACK with 64-bit ACKs.

C. Latency

Fig. 7a and Fig. 7b show the average latency of a full data + ACK transmission, from the time the data packet is generated on the application layer until the time the corresponding ACK is received. These figures also reveal the real throughput limits, past which, the MACs protocols hit their throughput limits, and the packet queues start building up indefinitely, as marked by a "near-vertical" increase in latency. Respectively, for JANUS-ACK, JANUS-QACK, and JANUS-IACK we note a throughput limit of around 0.2, 0.4, and 0.4 packets/min/node or 120, 240, and 240 packets network-wide per hour for 16 bits ACKs. Similar values are found for 64-bit ACKs. In other words, the JANUS-QACK and JANUS-IACK techniques provide traffic limits up to 2 times higher than the simpler JANUS-ACK.

Similarly to the throughput figures, the JANUS variants cope with bigger packets better than ALOHA-ACK. Finally, TDMA-ACK has an average fixed time cost due to using a slotted system. Because the JANUS variants are not restricted by slotting unless they're in back-off mode, they exhibit a data + ACK round-trip latency 10 times smaller than TDMA-ACK (on average) when the overall network traffic load is low.

D. Fairness

However, when measuring Jain's fairness index [16] of the different protocols, we discovered a fairness issue at higher traffic loads that damages the overall throughput capability. Past the throughput limits defined in subsection IV-C, the simulated networks display inequalities in channel usage as shown by Fig. 8a and Fig. 8b. All JANUS MAC protocol variants are affected, but JANUS-QACK and JANUS-IACK show significant improvements in fairness over JANUS-ACK. It should be stressed that this issue is only present in the JANUS protocol variants with ACKs, and not in the standard ACK-less JANUS MAC protocol. Solving this fairness issue is key to unlocking the full potential of the ACK extension of the JANUS MAC protocol.

V. Conclusions and Further Work

We have proposed an LLC sub-layer that integrates flow control, ACKs, and re-transmission management with the JANUS protocol to improve the reliability of underwater acoustic networks powered by it. We proposed three LLC-MAC integration strategies: a simple ACK flow with no extra scheduling rules; a "Quick ACK"; and an "Interrupt ACK" strategy. We demonstrated the variation in performance these strategies present, JANUS-QACK and JANUS-IACK can provide a significant increase in throughput compared to JANUS-ACK which is a simple addition of the LLC sub-layer without additional packet processing rules. This is made possible thanks to their hybridization of JANUS with the CSMA MAC protocol to push out ACKs faster, especially at higher loads. However, this throughput increase is gained at the cost of a greater number of ACK collisions, and by extension more re-transmissions. Still, all proposed JANUS variants create significantly fewer re-transmissions than ALOHA-ACK thanks to the collision avoidance capabilities of the core JANUS MAC protocol that is always present for data packet transmissions. Moreover, all JANUS variants also demonstrate lower data + ACK latency than TDMA at low traffic. But all of the proposed JANUS variants are hindered by fairness issues that appear at higher throughput loads, even if the JANUS-QACK and JANUS-IACK MAC protocols are less affected by them than JANUS-ACK.

This is an initial study and a more in-depth investigation of these new protocols is needed. In future work, we plan to develop the proposed LLC sub-layer further, and, crucially, solve the fairness issues in JANUS-ACK, JANUS-QACK and JANUS-IACK. Moreover, each LLC parameter, for example the ACK timeout length, can also be fine-tuned to further reduce wasted time between each transmission. Finally, because this work focused on evaluating the network performance of the integrated JANUS LLC-MAC protocols and their resulting behavior, it does not include yet a full breakdown of the ACK packet bit level structure, nor the bit allocation of the data packets that request an ACK. Defining this packet structure is another important direction for further work, which will help with wider adoption of the proposed schemes in JANUS systems worldwide.

REFERENCES

[1] N. Morozs, P. D. Mitchell and Y. Zakharov, "Target Detection Using Underwater Acoustic Networking," OCEANS 2023 - Limerick, Limerick, Ireland, 2023, pp. 1-5, doi: 10.1109/OCEANSLimerick52467.2023.10244266.

[2] N. Morozs, P. D. Mitchell and R. Diamant, "Scalable Adaptive Networking for the Internet of Underwater Things," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10023-10037, Oct. 2020, doi: 10.1109/JIOT.2020.2988621.

[3] Heidemann John, Stojanovic Milica and Zorzi Michele, 2012, "Underwater sensor networks: applications, advances and challenges", Phil. Trans. R. Soc. A.370158–175, doi: 10.1098/rsta.2011.0214

[4] "Sonar Acoustics Handbook", STO CMRE, Viale San Bartolomeo, La Spezia, Italy, 2016

[5] James Preisig. 2007. Acoustic propagation considerations for underwater acoustic communications network development. SIGMOBILE Mob. Comput. Commun. Rev. 11, 4 (October 2007), 2–10. https://doi.org/10.1145/1347364.1347370

[6] S. Jiang, "State-of-the-art Medium Access Control (MAC) protocols for underwater acoustic networks: A survey based on a MAC reference model," IEEE Commun. Surveys Tuts., vol. 20, no. 1, pp. 96–131, 2018.

[7] Yongxin Zhong, Jianguo Huang and Jing Han, "A TDMA MAC protocol for underwater acoustic sensor networks," 2009 IEEE Youth Conference on Information, Computing and Telecommunication, Beijing, 2009, pp. 534-537, doi: 10.1109/YCICT.2009.5382438.

[8] M. Schwartz and N. Abramson, "The Alohanet - surfing for wireless data [History of Communications]," in IEEE Communications Magazine, vol. 47, no. 12, pp. 21-25, Dec. 2009, doi: 10.1109/MCOM.2009.5350363.

[9] Rahman MA, Lee Y, Koo I. An adaptive network allocation vector timer-based carrier sense multiple access with collision avoidance medium access control protocol for underwater acoustic sensor networks. International Journal of Distributed Sensor Networks. 2017;13(1). doi:10.1177/1550147716687762

[10] R. Diamant, P. Casari, F. Campagnaro and M. Zorzi, "A Handshake-Based Protocol Exploiting the Near-Far Effect in Underwater Acoustic Networks," in IEEE Wireless Communications Letters, vol. 5, no. 3, pp. 308-311, June 2016, doi: 10.1109/LWC.2016.2549530.

[11] J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen and K. McCoy, "The JANUS underwater communications standard," 2014 Underwater Communications and Networking (UComms), Sestri Levante, Italy, 2014, pp. 1-4, doi: 10.1109/UComms.2014.7017134.

[12] J. Alves et al., "Moving JANUS forward: A look into the future of underwater communications interoperability," OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 2016, pp. 1-6, doi: 10.1109/OCEANS.2016.7761094.

[13] Tao, Qy., Zhou, Yh., Tong, F. et al. Evaluating acoustic communication performance of micro autonomous underwater vehicles in confined spaces. Frontiers Inf Technol Electronic Eng 19, 1013–1023 (2018). https://doi.org/10.1631/FITEE.1700841

[14] W. J. Thompson, "Poisson distributions," in Computing in Science & Engineering, vol. 3, no. 3, pp. 78-82, May-June 2001, doi: 10.1109/5992.919271.

[15] A. A. Alsebae, M. S. Leeson and R. J. Green, "SimEvents-based modeling and simulation study of Stop-and-Wait protocol," 2013 5th International Conference on Modelling, Identification and Control (ICMIC), Cairo, Egypt, 2013, pp. 239-244.

[16] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, "A quantitative measure of fairness and discrimination," Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, 1984, doi: 10.48550/arXiv.cs/9809099