

A Transmission Scheme for Continuous ARQ Protocols over Underwater Acoustic Channels

Mingsheng Gao¹, Wee-Seng Soh¹ and Meixia Tao²

¹Dept. of Electrical & Computer Engineering, National University of Singapore, Singapore 117576

²Dept. of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China 200240

Emails: elegaom@nus.edu.sg; elesohws@nus.edu.sg; mxtao@ieee.org

Abstract—Due to the *half-duplex* property of the underwater acoustic channels, the classic stop-and-wait ARQ (SW-ARQ) and its variants are generally thought to be the only class of ARQ protocols that can be applied in underwater. When combined with the large propagation delay property of the underwater acoustic channels, the use of SW-ARQ and its variants makes the throughput performance of underwater acoustic communication systems very inefficient. In this paper, we propose a transmission scheme that takes advantage of the long propagation delay in underwater to enable the use of *continuous* ARQ protocols over underwater acoustic channels. Simulation results show that our proposed transmission scheme allows much higher throughput to be achieved than both the classic SW-ARQ and its variants, even when simple *continuous* ARQ protocols are used.

I. INTRODUCTION

Forward-error-correction (FEC) and automatic repeat request (ARQ) are the two main error-control techniques that ensure the reliability of data transmission in underwater acoustic links [1], [2]. Recently, there has been some research on how FEC schemes can be applied to eliminate burst errors in underwater acoustic communication systems [3], [4]. However, in any FEC scheme, the number of erroneous symbols in a received codeword cannot exceed a certain bound, beyond which the errors cannot be corrected. Therefore, there are always some application scenarios that prefer the use of ARQ techniques to achieve the desired reliability [5].

In terrestrial RF communications whereby propagation delays are short, only the stop-and-wait ARQ (SW-ARQ) protocol and its variants can be used when the links are half-duplex [6]. In order to improve the throughput performance of half-duplex communication systems, several variants of the classic SW-ARQ have been proposed over the years. For example, Morris [7] proposes that the transmitter sends a group of M packets, and waits for the receiver to reply with the ACKs/NAKs for these packets as a group. Those packets that are NAK'ed are then retransmitted with other new packets to form a new group of M packets during the next cycle. In this manner, as many as M packets can be transmitted during one round-trip time and, hence, the throughput efficiency is increased. However, if ordered delivery of packets to the upper layer is needed, the receiver has to store the packets until they can be rearranged. In [8], Turney proposes a scheme in which the transmitter also sends out a group of M packets and waits for their ACKs/NAKs, but only those packets that are NAK'ed are retransmitted in the next cycle, i.e., no new

packet is added. The advantage over the previous approach is that the receiver now does not need the capability to buffer more than M packets for ordered delivery.

For the case of underwater acoustic communications, since the modems are mostly half-duplex, it is generally thought that only SW-ARQ and its variants would be feasible. For instance, in [9], Stojanovic improves the SW-ARQ variant in [7] by selecting the optimal packet size, which is a function of the communication range, transmission rate, and bit error rate. In essence, these variants have merely optimized the classic SW-ARQ; while they have retained the SW-ARQ's suitability for half-duplex links, they have also retained its undesirable properties of low efficiency and high packet delay.

In the underwater environment, however, the underwater acoustic waves have a low propagation speed of approximately 1500 m/s, which results in significant propagation delay. This is also the key reason why the SW-ARQ protocol and its existing variants perform even more badly in underwater, since a transmitter always idles for one round-trip time while waiting for the receiver's ACK(s)/NAK(s). In a natural manner, if a methodology can be devised that would enable a *continuous* ARQ despite of the channel's half-duplex property, the transmission efficiency could be greatly improved. An important but often neglected property in underwater acoustic communication is that, it is actually possible for a pair of nodes to send packets that cross each other in the medium, while each packet can still be correctly received by the other node, so long as each node has finished its transmission and has switched to listening mode by the time the packet arrives. Motivated by this property, we present a transmission scheme where a transmitter alternates between transmitting a data packet to the receiver, and listening for an earlier packet's ACK/NAK, while one or more other packets are still in transit. This enables *continuous* ARQs to be implemented over underwater acoustic channels. Although the approach is unthinkable in half-duplex terrestrial wireless systems due to the much shorter propagation delay, it has now become feasible in underwater by riding on its long propagation delay.

It should be mentioned that, although many studies have been performed on *continuous* ARQs for terrestrial communication systems, to the best of our knowledge, no attempt has been made to imitate this continuous behavior in half-duplex underwater acoustic communication systems so far. In this paper, we explain how our proposed transmission scheme

can enable *continuous* ARQ protocols to be implemented over half-duplex underwater acoustic channels. We also provide simulation results to show that, even if we merely apply simple *continuous* ARQ protocols that are now enabled by the proposed transmission scheme, they can already achieve much higher throughput than the *classic* SW-ARQ and its variants over underwater acoustic channels.

The rest of this paper is organized as follows. In Section II, we describe the transmission scheme in detail. In Section III, we explain how the system parameters of the transmission scheme can be determined. We then show in Section IV the simulation results that we have obtained when we implemented several *continuous* ARQ protocols using our proposed transmission scheme, and compare them with the SW-ARQ and one of its variants. Finally, Section V concludes the paper.

II. OUR PROPOSED TRANSMISSION SCHEME

Consider an underwater communication system whereby a transmitter has an estimation of the initial propagation delay (denoted by p_0) to its receiver, as well as the upper bound of the relative radial velocity (denoted by v_r) between the two nodes. Here, we define v_r to be positive when the two nodes are approaching each other. For convenience, we utilize the following notations in our model description:

- v_p : The propagation speed of underwater acoustic waves.
- δ and γ : The transmission times of a data packet and an ACK/NAK packet, respectively. Note that their reception times are assumed to be the same as their transmission times, because v_r is usually much smaller than v_p .
- p_i : The propagation time of the i^{th} data packet from the transmitter to the receiver.
- p'_i : The propagation time of the ACK/NAK packet for the i^{th} data packet from the receiver to the transmitter.
- t_i : The idle period at the transmitter after sending the i^{th} data packet. These idle periods are introduced to avoid possible collisions between the data transmissions and the ACK/NAK receptions by the transmitter when $v_r > 0$.
- t'_i : The idle period at the receiver after acknowledging the i^{th} data packet with either an ACK or a NAK. These idle periods are introduced to avoid possible collisions between the receptions of data packets and the ACK/NAK transmissions by the receiver when $v_r > 0$.
- s : The maximum permissible number of transmitted, but not yet acknowledged, data packets; this is also referred to as the “window size”.
- k_0 : The maximum allowable total number of data packets that can be transmitted in the current session (including retransmissions); if the transmitter still has some remaining data packets to be sent when k_0 is reached, it has to initialize a new session before it can continue to transmit them. Therefore, k_0 should be set to a value that is sufficiently large to accommodate the current session.

As mentioned earlier, the classic SW-ARQ’s low throughput and high packet delay become much more pronounced in the underwater acoustic channel, due to the long propagation delay. The proposed transmission scheme, on the other hand,

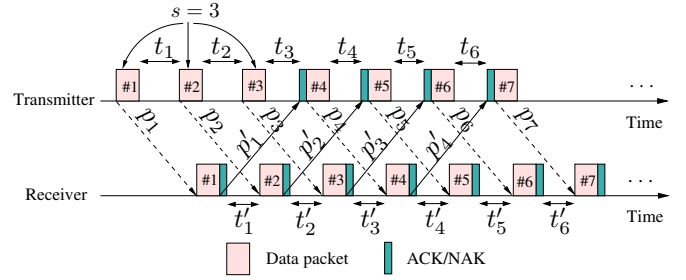


Fig. 1. The flow of the transmission scheme.

combats the effects of long propagation delay by having more than one packet to be on-the-fly within the channel between the transmitter and the receiver, at any instant. The flow of the transmission scheme for $s = 3$ is illustrated in Fig. 1. Here, we assume that the transmitter has already acquired the control of the channel successfully, and has a series of packets to send to the receiver. Also, for the ease of explanation, we assume that the ACKs/NAKs are received error-free¹. After sending s data packets, the transmitter switches to a “stop-and-wait” mode, in which it waits until it has received an ACK/NAK before transmitting the next data packet. This eliminates the need for explicit time-slot synchronization between the transmitter and the receiver. The key difference from the classic SW-ARQ protocol is that the ACK/NAK that the transmitter expects is not the response for the most recently transmitted data packet, but for an earlier data packet. For example, after transmitting packet #3, the transmitter waits to receive the ACK/NAK for packet #1. For the receiver, it transmits the ACK/NAK immediately after receiving each data packet. Apart from the time it spends on transmitting the ACK/NAK, the receiver listens for data packet at other times.

Our novel approach is similar to the juggling of objects between two hands; when juggling, a hand cannot be throwing one object while receiving another object at the same time, which is similar to the half-duplex property of the underwater acoustic channel. In addition, multiple objects can be juggled between two hands, much like the transmission scheme’s attempts to inject multiple data packets and ACKs/NAKs into the medium simultaneously. We shall refer to this transmission scheme as the juggling-like stop-and-wait (JSW) scheme. Note that it is merely a transmission scheme, rather than an ARQ protocol. Its transmission strategy allows other continuous ARQ protocols to be applied in half-duplex underwater acoustic links. Although it may appear that our transmission scheme requires an ACK/NAK for every packet, it can actually be further optimized. For example, we can group multiple packets together to form a packet train, for which the receiver only needs to return a single “ACK” packet that contains cumulative/selective ACK information.

It can be observed from Fig. 1 that the JSW scheme can function properly if and only if (i) $t_i \geq \gamma$ for $1 \leq i \leq s - 1$, (ii) $t_i \geq 0$ for all $i \geq s$, and (iii) $t'_i \geq 0$ for all $i \geq 1$.

¹Although it is possible to extend our protocol to include scenarios that consider corrupted ACKs/NAKs, we will explain how this can be addressed in our future paper, due to space limitation.

Conditions (i) and (ii) ensure that the transmitter is always in listening mode when an ACK/NAK arrives; on the other hand, condition (iii) ensures that the receiver is always in listening mode when a data packet arrives.

Our main focus will be on the case where $v_r > 0$. We will explain at the end of Section III how our proposed scheme works when $v_r \leq 0$. Also, note that v_r is merely an upper bound, meaning that the instantaneous relative radial velocity can still be negative even when v_r is positive. For $v_r > 0$, it is possible for both t_i and t'_i to decrease over time, with the fastest rate of decrease occurring when the relative radial velocity is consistently at its upper bound, i.e., v_r . Therefore, given p_0 , v_r , and k_0 , it is crucial to choose an appropriate value for s , as well as the appropriate inter-packet spacings to transmit the first s packets, so that the above three conditions are always met throughout the entire transmission session.

III. DETERMINATION OF THE SYSTEM PARAMETERS

We now present in detail how the system parameters of the JSW scheme can be calculated. For the ease of presentation, we assume that the switching time between transmit and receive modes, t_{sw} , is negligible. Note, however, that it can be fully accommodated in the following calculations by incorporating it within the transmission time of an ACK/NAK, i.e., we can replace the parameter γ with γ' , where $\gamma' = \gamma + t_{sw}$.

A. Calculation of s

For simplicity, the inter-packet spacings to transmit the first s packets (i.e., t_1, t_2, \dots, t_{s-1}) are made equal, with a value denoted by t_0 . We will show in Section III-B how its appropriate value can be calculated. From Fig. 1, we have

$$p_1 + p'_1 + \delta = s\delta + \sum_{i=1}^{s-1} t_i + t_s. \quad (1)$$

Since $t_s \geq 0$, the above can be further rewritten as

$$p_1 + p'_1 \geq (s-1)\delta + \sum_{i=1}^{s-1} t_i. \quad (2)$$

Thus, s can be expressed as

$$s = \left\lfloor \frac{p_1 + p'_1}{\delta + t_0} \right\rfloor + 1, \quad (3)$$

where p_1 and p'_1 are given by (4) and (5), respectively:

$$p_1 = \frac{v_p}{v_p + v_r} p_0 \quad (4)$$

$$p'_1 = \max \left\{ p_1 - \frac{v_r}{v_p + v_r} (p_1 + \delta), 0 \right\}. \quad (5)$$

B. Calculation of t_0

From Fig. 1, for all $k \geq 2$, we can obtain

$$p_k = \max \left\{ p_1 - \frac{v_r}{v_p + v_r} \left[(k-1)\delta + \sum_{i=1}^{k-1} t_i + (k-s)\gamma.U(k-s-1) \right], 0 \right\} \quad (6)$$

and

$$p'_k = \max \left\{ p_1 - \frac{v_r}{v_p + v_r} \left[p_1 + k\delta + \sum_{i=1}^{k-1} t'_i + (k-1)\gamma \right], 0 \right\} \quad (7)$$

where $U(x)$ is a unit step function that returns a 0 if $x < 0$, and returns a 1 otherwise. It can be noted from the above that, if the relative distance between the transmitter and the receiver remains static, we have $p_k = p'_k = p_0$ for all $k \geq 1$.

We can also make the following observations about p_k and p'_k for all $k \geq 2$:

$$p_k = p_1 + \sum_{i=1}^{k-1} (t'_i - t_i) + [\min(k, s) - 1]\gamma \quad (8)$$

$$p'_k = (s-1)\delta + \sum_{i=1}^s t_i + \sum_{j=1}^{k-1} (t_{s+j} - t'_j) - p_1 \quad (9)$$

The above two relations can be used to find t_k for all $k \geq s$, and t'_k for all $k \geq 1$. Thus, we have

$$t_s = p_1 + p'_1 - (s-1)\delta - \sum_{i=1}^{s-1} t_i \quad (10)$$

$$t'_k = p_{k+1} - p_k + t_k - \gamma U(s-k-1) \quad (11)$$

$$t_{s+k} = p'_{k+1} - p'_k + t'_k \quad (12)$$

For a given $v_r > 0$, we can see from (6) and (7) that, both p_k and p'_k are strictly decreasing functions with respect to k for $1 \leq k \leq k_0$. In light of (11) and (12), we have

$$t_{s+k} < t_k, \quad \text{for } k \geq s \quad (13)$$

$$t'_{s+k} < t'_k, \quad \text{for } k \geq 1 \quad (14)$$

Equations (13) and (14) tell us that, both the sequences $t_k, t_{s+k}, t_{2s+k}, \dots$ and $t'_k, t'_{s+k}, t'_{2s+k}, \dots$ (for any $k \in [1, s]$) are monotonically decreasing sequences. These imply that, for any given $t_0 \geq \gamma$, if it can result in $t_k \geq 0$ and $t'_k \geq 0$ for all $k \in [k_0 - s + 1, k_0]$, then $t_k \geq 0$ and $t'_k \geq 0$ also hold for all $k \in [1, k_0]$. Such t_0 ($\geq \gamma$) always exists for any given combination of p_0 , v_r , and k_0 that yields $s \geq 2$. Note that for the case where $s = 1$, the transmission scheme degrades to the conventional transmission mode that is only suitable for the classic SW-ARQ and its variants, and t_0 no longer exists.

In order to obtain a value for t_0 that satisfies the above conditions for a given set of p_0 , v_r , and k_0 , we use a numerical method as summarized in Fig. 2. As an illustrated example, we explore the values of s and t_0 when the upper bound of the relative radial velocity, v_r , varies from 0 to 6 m/s, along with the following parameters: $p_0 = 0.6667$ s (corresponding to an initial separation of 1 km), $\delta = 0.064$ s, $\gamma = 0.005$ s, $k_0 = 1000$, and $\Delta = 0.001$. From Fig. 3, we can see that t_0 increases monotonically as v_r varies from 0 m/s to around 5.6 m/s. Fig. 4, on the other hand, shows that the window size, s , decreases monotonically as v_r increases. When v_r goes beyond 5.6 m/s, the transmission scheme degrades to the classic stop-and-wait behavior, with $s = 1$. In this case, t_0 no longer exists, as can be seen in Fig. 3. By comparing Fig. 3 with Fig. 4 closely, we also observe that the discontinuities in Fig. 3 occur at those points where s changes.

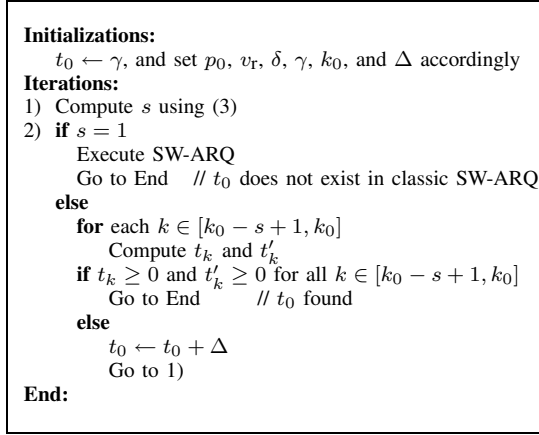


Fig. 2. The algorithm for determining the initial inter-packet spacing, t_0 .

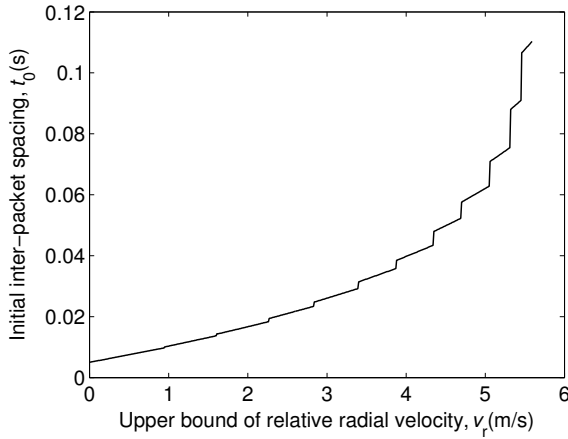


Fig. 3. The initial inter-packet spacing t_0 versus the relative radial velocity's upper bound v_r for $p_0 = 0.6667$, $\delta = 0.064$, $\gamma = 0.005$, and $k_0 = 1000$.

From the discussion above, the following remarks can be made. Firstly, our scheme does not require accurate knowledge of the propagation delay, since we can always bias towards a smaller p_0 , which will only result in a smaller s . Secondly, it can be easily inferred that our scheme can work well when $v_r > 0$. For $v_r \leq 0$, the transmitter and the receiver may either remain static relative to each other (if $v_r = 0$), or they may drift further apart as time goes by. In either case, the conditions (i), (ii) and (iii) in Section II will always be met so long as all t_i are initialized to γ for $1 \leq i \leq s - 1$, and s is given by

$$s = \left\lfloor \frac{2p_0}{\delta + t_0} \right\rfloor + 1. \quad (15)$$

IV. SIMULATION RESULTS

We have performed simulations using programs written in C++ to evaluate the benefits of our proposed scheme. In our simulations, we only consider point-to-point communications, and we assume that the two points are within each other's range throughout the session. The switching time between transmit and receive modes are assumed to be negligible. We have also assumed that the data rate is 8 kbps, the data packet length is 512 bits, and the control packet length (i.e., ACK/NAK) is 40 bits. The performance metric used is

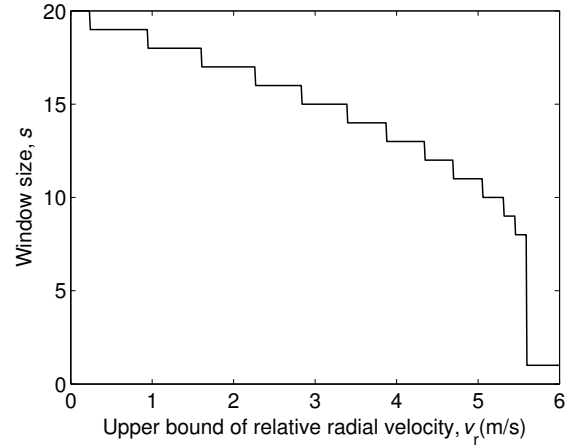


Fig. 4. The window size s versus the relative radial velocity's upper bound v_r for $p_0 = 0.6667$, $\delta = 0.064$, $\gamma = 0.005$, and $k_0 = 1000$.

throughput, which we define as the ratio of the *total useful time* to the *total simulation time*, where the *total useful time* refers to the total transmission time of all the data packets that are correctly received during the *total simulation time*. We have implemented two classic continuous ARQ protocols, namely, go-back- N (GBN-ARQ) and selective-repeat (SR-ARQ), which are now made possible in half-duplex underwater acoustic channel by our JSW scheme. We compare their throughput with the classic SW-ARQ and T-ARQ, under both static and dynamic conditions, to demonstrate the loss in efficiency if we were to stick to the previous perception that only non-continuous ARQ protocols may be applied in half-duplex underwater acoustic channels. We choose not to compare with the scheme in [9], which optimizes the packet length for [7], because its throughput can be increased by having a larger group size, M . The comparison will not be fair since this is achieved at the expense of having a much larger average resequencing delay at the receiver. In addition, the optimal packet size calculation in [9] loses its optimality when the nodes move.

First, we consider the static case (i.e., the relative radial velocity is zero), where $p_k = p'_k = p_0$ for all $k \geq 1$. Here, p_0 is assumed to be 0.6667 s, and t_i is set to γ for $1 \leq i \leq s - 1$. We also assume that there is an infinite data source at the transmitter. The total simulation time is 1,000,000 s, and all the results presented are averaged over 10 simulation runs. Fig. 5 illustrates the throughput of these protocols versus packet error rates (P_e) ranging from 0.001 to 0.2. Here, the classic GBN-ARQ and SR-ARQ are implemented based on our JSW transmission scheme, while the classic SW-ARQ and T-ARQ (with $M = 5$) are based on the conventional stop-and-wait transmission scheme where the transmitter-receiver pair are not allowed to transmit simultaneously. As expected, T-ARQ with $M = 5$ provides higher throughput than classic SW-ARQ. This is because T-ARQ sends multiple data packets in each stop-and-wait cycle, while SW-ARQ sends only a single data packet in each cycle. However, both of them have much lower throughput than GBN-ARQ and SR-ARQ. The superior performance of the latter two protocols can be

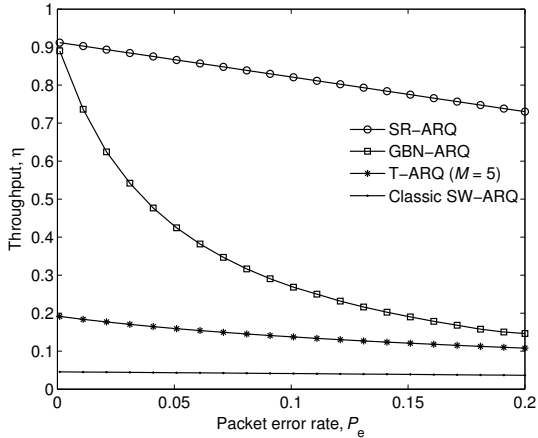


Fig. 5. Throughput vs. packet error rate comparison among the four different ARQ protocols for $p_0 = 0.6667$.

largely attributed to their use of our JSW transmission scheme, which allows these continuous ARQs to operate in an almost continuous manner without the need for a full-duplex channel.

Next, we consider a dynamic scenario whereby the radial velocity ranges between 0 and 6 m/s. Fig. 6 shows its impact on the throughput of each of the four different ARQ protocols that we have simulated. At each data point, we hold the radial velocity constant, and measure the ARQ's throughput, assuming that there are 500 data packets to be transmitted, while p_0 is 0.6667 s, P_e is 0.05, and k_0 is 1000. From the figure, we can make several interesting observations. Firstly, both GBN-ARQ and SR-ARQ can achieve much higher throughput than T-ARQ and SW-ARQ when the radial velocity is less than 5.6 m/s. However, their throughput degrades to that of SW-ARQ when the radial velocity goes beyond 5.6 m/s. This can be attributed to the fact that when the radial velocity is less than 5.6 m/s, the window size s is greater than one; thus, both GBN-ARQ and SR-ARQ can benefit from our JSW scheme, and operate as continuous ARQ protocols. On the other hand, when the radial velocity exceeds 5.6 m/s, s degrades to one, essentially making these protocols operate like SW-ARQ. Secondly, for $s > 1$, the throughput of both GBN-ARQ and SR-ARQ experiences stair-like descents as the radial velocity increases. By comparing Fig. 4 and Fig. 6, we observe that the discontinuities in their throughput occur at those radial velocities where s decreases. This is because a smaller s implies that less packets can be placed into the pipeline, causing a noticeable drop in the throughput. On the contrary, the throughput of both T-ARQ and SW-ARQ increases as the radial velocity increases, because less time is wasted on waiting as the round-trip delay decreases faster.

V. CONCLUSION

In this paper, we propose a transmission scheme for underwater acoustic channels that enable the use of continuous ARQ protocols without the need for a full-duplex channel. Our scheme exploits the long propagation delay in underwater, and allows the transmitter and the receiver to inject multiple data packets and ACKs/NAKs into the medium simultaneously

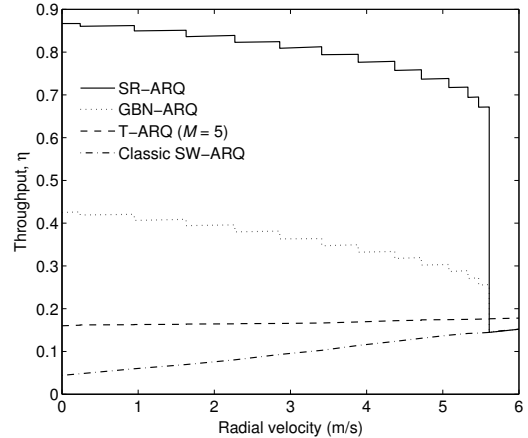


Fig. 6. Throughput vs. relative radial velocity comparison among the four different ARQ protocols for $p_0 = 0.6667$ and $P_e = 0.05$.

in a juggling-like manner, even though the channel is half-duplex. We have also considered the case whereby the radial velocity between the transmitter-receiver pair is non-zero, and shown that the scheme works correctly so long as the maximum radial velocity and the initial propagation delay are known. It also does not require any clock synchronization between the transmitter-receiver pair. Our simulation results show that the proposed transmission scheme allows much higher throughput to be achieved than the existing stop-and-wait ARQ approaches, even when simple continuous ARQ protocols, namely GBN-ARQ and SR-ARQ, are used. This implies that more sophisticated continuous ARQ protocols that are now made possible by our transmission scheme could potentially achieve even better throughput.

ACKNOWLEDGMENT

This research was supported by the Ministry of Education of Singapore, under AcRF Grant No. R-263-000-370-112.

REFERENCES

- [1] I. F. Akyildiz, D. Pompili, and T. Melodia, "State-of-the-art in protocol research for underwater acoustic sensor networks," in *Proc. WUWNet'06*, Los Angeles, California, USA, Sep. 2006.
- [2] J. Proakis, E. Sozer, J. Rice and M. Stojanovic, "Shallow water acoustic networks," *IEEE Commun. Mag.*, vol. 39, no. 11, pp.114-119, Nov. 2001.
- [3] D. Pompili, T. Melodia, and I. Akyildiz, "Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks," in *Proc. ACM Mobicom*, 2006.
- [4] A. F. Harris III, D. G. B. Meneghetti, and M. Zorzi, "Maximizing channel utilization for underwater acoustic links," in *Proc. IEEE OCEANS*, June 2007.
- [5] M. Zorzi, "Some results on error control for burst-error channels under delay constraints," *IEEE Trans. Veh. Technol.*, vol. 50, no. 1, pp. 12-24, Jan. 2001.
- [6] S. Lin, D. J. Costello, Jr., and M. J. Miller, "Automatic-repeat-request error control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5-17, Dec. 1984.
- [7] J. M. Morris, "Optimal blocklengths for ARQ error control schemes," *IEEE Trans. Commun.*, vol. 27, pp. 488-493, Feb. 1979.
- [8] P. F. Turney, "An improved Stop-and-Wait ARQ logic for data transmission in mobile radio systems," *IEEE Trans. Commun.*, vol. 29, pp. 68-71, Jan. 1981.
- [9] M. Stojanovic, "Optimization of a data link protocol for an underwater acoustic channel," in *Proc. IEEE OCEANS*, pp. 68-73, Jun. 2005.