# On the use of rate-less codes in underwater acoustic file transfers

Mandar Chitre, and Mehul Motani, *Member, IEEE*

***Abstract*— The network protocols developed for electromagnetic wireless communications in air cannot be directly used in underwater acoustic networks due to differences in channel characteristics, propagation speed, bandwidth and the half duplex nature of most acoustic modems. To ensure efficient use of the underwater acoustic channel, the protocol design needs to take propagation delay into account. In this paper, we address a common practical problem of transferring a data file reliably from one underwater node to another. Although this point-to-point model is simple, it captures relevant characteristics of the underwater acoustic channel and has immediate practical applications. We compare the efficiency of various file transfer protocols addressing this common need, where the efficiency is measured by the time taken to transfer the file reliably.**

**The simplest of protocols to address reliable file delivery involves splitting the file into smaller data packets, transmitting each packet and ensuring reliable delivery of the packet via an acknowledgement from the receiving node. When the propagation delay is large, the protocol spends significant amounts of time waiting for acknowledgements and not utilizing the channel. This results in a poor efficiency. The efficiency of the protocol can be improved at the cost of complexity by combining multiple acknowledgements.**

**We also consider a protocol based on rate-less codes, a class of erasure correcting codes where virtually an infinite number of symbols can be generated from the source data. The source data can be reconstructed from any set of the generated symbols provided the set contains a minimum number of independent symbols. This allows a file transfer protocol to be designed where the individual packets do not have to be acknowledged. This reduces the need for acknowledgements dramatically and hence the protocol efficiency is less dependent on the propagation delay.**

***Index Terms*— Channel coding, communication systems, protocols, underwater acoustic communication.**

## I. INTRODUCTION

To reduce the cost and increase the coverage of ocean monitoring, researchers have been looking towards autonomous sensors and systems. Although these systems provide tremendous benefits, they pose a new challenge – efficient communications with these systems. As electromagnetic waves do not propagate well in seawater, acoustics provides the natural channel for communication. However, the acoustic bandwidth available is limited by the rapid absorption at high frequencies. Efficient use of this bandwidth is therefore important. As acoustic waves travel about 105 times slower in seawater as compared to electromagnetic waves in air, the network protocols developed for electromagnetic wireless communications in air cannot be directly used in underwater acoustic networks. To ensure efficient use of the underwater acoustic channel, the protocol design needs to take propagation delay into account. Practical considerations of bandwidth availability and dynamic range have led to the development of half-duplex acoustic modems. This imposes an additional constraint on the protocols.

In this paper, we address a common practical problem of transferring a data file reliably from one underwater node to another. Furthermore, we assume that the two nodes are the only users of the acoustic channel in the geographical area and frequency band. Though this point-to-point model is simple, it captures relevant characteristics of the underwater acoustic channel and is a stepping-stone towards multi-node underwater networks. Moreover, the point-to-point file transfer function has immediate practical applications and can be part of more complex approaches.

In this paper, we compare the efficiencies of various protocols for the point-to-point file transfer problem. We measure the efficiency of a protocol by the time taken to transfer the file reliably using the protocol – the less the transfer time, the more efficient the protocol. More specifically, the efficiency is defined as the ratio of the average file transfer speed and the link transmission rate.

The simplest of protocols to address reliable file delivery involves splitting the file into smaller data packets, transmitting each packet and ensuring reliable delivery of the packet via an acknowledgement from the receiving node. When the propagation delay is large, the protocol spends significant amounts of time waiting for acknowledgements and not utilizing the channel. This results in a poor efficiency. The efficiency of this protocol can be improved at the cost of complexity by combining multiple acknowledgements. Multiple packets could be sent in a batch (or cluster) without waiting for an acknowledgement between packets. A single acknowledgement packet with a bit field for each data packet would be sent per cluster of data packets. This protocol is only slightly more complex than the simple protocol but can be made much more efficient.

Manuscript received March 24, 2007. This work was supported in part by the Defence Science & Technology Agency (DSTA), Singapore and the National University of Singapore.

M. Chitre is with the Acoustic Research Laboratory, Tropical Marine Science Institute, National University of Singapore, 12A Kent Ridge Road, Singapore 119223 (e-mail: mandar@arl.nus.edu.sg).

M. Motani is with the department of Electrical & Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576 (e-mail: motani@nus.edu.sg).

We then consider a protocol based on rate-less codes [1, 2] (also known as fountain codes [3, 4]). Rate-less codes are a class of erasure correcting codes where virtually an infinite number of symbols can be generated from the source data. The source data can be reconstructed from any set of the generated symbols provided the set contains a minimum number of unique symbols. This allows a file transfer protocol to be designed where the individual packets (rate-less code symbols) do not have to be acknowledged. The transmitter is only required to know when the receiver has received enough packets to reconstruct the source data (i.e., the file to be transferred). This reduces the need for acknowledgements dramatically and hence the protocol efficiency is less dependent on the propagation delay.

We point out that there are various approaches to the file transfer problem that have been explored in the literature. Early works focused on wired networks (e.g., the Internet) and these have been extended to the wireless network scenario. Recent work has considered reliable file transfer in challenged network environments [5, 6]. However, as described above, underwater channels pose special challenges to network designers. To the best of our knowledge, these issues have not been addressed adequately in previous works. The two simple point-to-point file transfer protocols we use as a benchmark have been studied in the underwater network scenario in [7], which studies the optimization of the protocol parameters for underwater channels. The aim of this paper is to further [7] by utilizing rate-less codes in the protocol to provide the reliability through redundancy. We envision that a point-to-point file transfer protocol based on rate-less codes can be a building block for a reliable protocol in network scenario.

In this paper, we compare the efficiencies of the three simple reliable point-to-point file transfer protocols described above through analytical and numerical studies. We also briefly discuss the implications of these findings on protocols for transferring infinite data streams rather than finite data files.

## II. ACKNOWLEDGEMENT BASED PROTOCOLS

### A. Individual Packet Acknowledgement

A very simple robust file transfer protocol consists of fragmenting the file into small packets, transmitting each packet and waiting for an acknowledgement before the next transmission. A timeout is imposed on the reception of the acknowledgement, with the assumption that the packet was lost if an acknowledgement was not received. The timeout can be computed based on the expected propagation and processing delays. The receiver only acknowledges packets that were successfully received. The receiver silently drops erroneous packets, which may be detected using checksums. It also silently drops duplicate packets, which may be detected using sequence numbers.

Let $H$ be the length of a packet header and $L$ be the length of a data payload in a packet in bits. We assume that a data packet consists of a header and data payload, while the acknowledgement packet only consists of a header. Let $R$ be the link transmission rate, $D$ the distance between the two communicating nodes and $c$ be the average sound speed in the channel.

To transfer a file of size $F$, the file has to divided into $N$ packets such that

$$N = \left\lceil \frac{F}{L} \right\rceil. \tag{1}$$

Let $p$ be the probability of bit error in the packet. In case of error correction coding, $p$ is measured at the output of the decoder so that the metric of probability of error is similarly applied to coded and uncoded communications. For a data packet of length $H$+$L$, the probability that the packet is received correctly is

$$P_D = (1-p)^{H+L}. \tag{2}$$

For an acknowledgement packet of length $H$, the probability of correct reception is

$$P_{1A} = (1-p)^{H}. \tag{3}$$

The probability that both the data and acknowledgement is received correctly is

$$P_1 = P_D P_{1A} = (1-p)^{2H+L}. \tag{4}$$

Assuming negligible processing time, the transmission of a packet and reception of the corresponding acknowledgement takes time $t_1$ such that

$$t_1 = \frac{H+L}{R} + \frac{H}{R} + 2\frac{D}{c}. \tag{5}$$

With a timeout of exactly $t_1$, the average time required to transfer the file is

$$\begin{aligned} T_1 &= \frac{N}{P_1} t_1 \\ &= \frac{1}{(1-p)^{2H+L}} \left\lceil \frac{F}{L} \right\rceil \left( \frac{2H+L}{R} + \frac{2D}{c} \right). \end{aligned} \tag{6}$$

In most underwater modems, feasible packet lengths are limited by the data speed and time coherence of the channel. Typical files to be transferred are usually much larger than the packet length i.e. $F >> L$. Hence,

$$\left\lceil \frac{F}{L} \right\rceil \approx \frac{F}{L}. \tag{7}$$

The average file transfer speed can thus be computed as:

$$S_1 = \frac{F}{T_1} = \frac{L(1-p)^{2H+L}}{\left( \frac{2H+L}{R} + \frac{2D}{c} \right)}. \tag{8}$$

By comparing this speed with the link transmission rate, we can estimate the efficiency of the protocol:

$$\eta_1 = \frac{S_1}{R} = \frac{(1-p)^{2H+L}}{1 + \frac{2}{L}\left( H + \frac{DR}{c} \right)}. \tag{9}$$

The efficiency is dominated by the second term in the denominator when $L$ is small. The efficiency is poor due to the overhead from the header and the propagation delay. Increasing $L$ helps improve efficiency by reducing this term. At large values of $L$ this term becomes negligible and the numerator dominates the efficiency. For $p > 0$, the denominator reduces with increasing $L$, causing the efficiency to drop. An optimum exists at some intermediate packet length such that the efficiency is maximal. This is clearly seen in Fig. 1 where we plot efficiency curves for $H$ = 16 bits, $R$ = 10 kbps, $c$ = 1500 m/s and various values of $D$ and $p$.

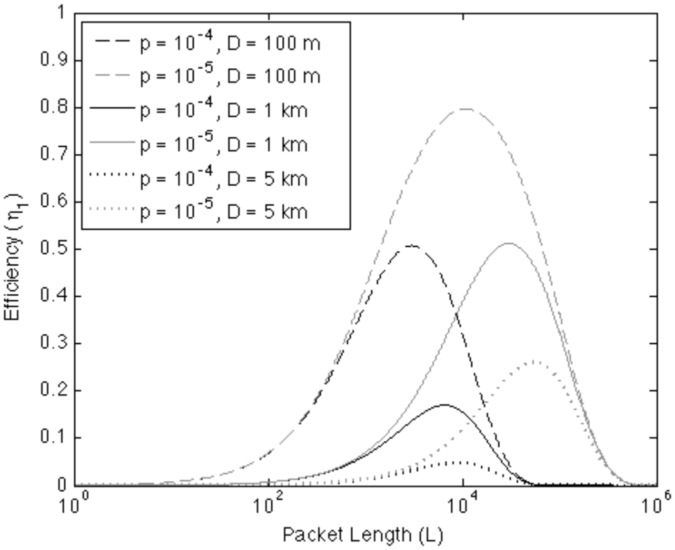

Fig. 1. Efficiency of individual packet acknowledgement protocol

At short ranges of 100 m, efficiencies up to 0.8 can be reached with packet lengths of about $10^4$ bits provided $p = 10^{-5}$. At a higher value of $p = 10^{-4}$, the optimal packet length is lesser and the maximal efficiency is only about 0.5. At a longer range of 1 km, the maximum efficiency reduces to about 0.5 and 0.15 at $p = 10^{-5}$ and $p = 10^{-4}$ respectively. At an even longer range of 10 km, the maximum efficiency reduces even further to only about 0.25 and 0.05 at $p = 10^{-5}$ and $p = 10^{-4}$ respectively. The required packet lengths also increase with range and can become impractically large at long ranges and low values of $p$. The protocol efficiency at long ranges is very poor, as the protocol requires the sender to wait for an acknowledgement every time it transmits a data packet. The file transfer speed is thus severely limited by the propagation delay in the channel.

### *B. Packet Cluster Acknowledgement*

A simple change to the protocol described in the previous sub-section can improve its performance significantly. Instead of transmitting a single packet and then waiting for an acknowledgement, the sender could transmit a cluster of packets before requiring an acknowledgement. This would reduce the average time per packet spent waiting for an acknowledgement and hence improve the efficiency of the protocol.

Let $M$ be the number of data packets in a cluster. After sending $M$ packets, the sender waits for an acknowledgement from the receiver. The acknowledgement consists of a bit-field of $M$ bits where each bit indicates if the receiver successfully received a particular data packet in the cluster. The packets are uniquely identified using the sequence number in the packet's header.

For an acknowledgement packet of length $H$+$M$, the probability of correct reception is

$$P_{CA} = (1-p)^{H+M}. \tag{10}$$

The probability that both the data and acknowledgement is received correctly is

$$P_C = P_D P_{CA} = (1-p)^{2H+L+M}. \tag{11}$$

Assuming negligible processing time, the transmission of a cluster and reception of the corresponding acknowledgement takes time $t_C$ such that

$$t_C = M\left(\frac{H+L}{R}\right) + \frac{H+M}{R} + 2\frac{D}{c}. \tag{12}$$

With a timeout of exactly $t_C$, the average time required to transfer the file is

$$\begin{aligned} T_C &= \frac{1}{P_C}\left\lceil \frac{N}{M} \right\rceil t_C \\ &= \frac{\left\lceil \frac{1}{M}\left\lceil \frac{F}{L} \right\rceil \right\rceil}{(1-p)^{2H+L+M}} \left( \frac{M(H+L)+H+M}{R} + \frac{2D}{c} \right). \end{aligned} \tag{13}$$

Assuming $F >> L$ and $N >> M$,

$$\left\lceil \frac{1}{M}\left\lceil \frac{F}{L} \right\rceil \right\rceil \approx \frac{F}{LM}. \tag{14}$$

The average file transfer speed can thus be computed as:

$$S_C = \frac{F}{T_C} = \frac{LM(1-p)^{2H+L+M}}{\left( \frac{LM+HM+H+M}{R} + \frac{2D}{c} \right)}. \tag{15}$$

By comparing this speed with the link transmission rate, we can estimate the efficiency of the protocol:

$$\eta_C = \frac{S_C}{R} = \frac{(1-p)^{2H+L+M}}{1 + \frac{1}{L}\left( H + \frac{H}{M} + 1 + \frac{2DR}{Mc} \right)}. \tag{16}$$

This efficiency is plotted in Fig. 2 for $H$ = 16 bits, $R$ = 10 kbps, $c$ = 1500 m/s, $p = 10^{-5}$ and various values of $D$ and $M$. It is clearly seen that much higher efficiencies can be obtained by introducing clustered acknowledgements. The optimal packet length reduces with increasing $M$ and is only about $10^3$ bits for large values of $M$. As $M$ increases, the efficiency increases, with 0.9 or better efficiencies obtained at $M$ = 1000.

The file transfer speed can be significantly improved using this protocol as compared to the individual packet acknowledgement protocol at the cost of algorithmic complexity and memory requirements at the communicating nodes.

## III. A PROTOCOL BASED ON RATE-LESS CODES

### *A. Motivation*

The protocols described in the previous section require a feedback channel from the receiver to the sender in order to convey which packets were successfully received. This enables the sender to resend packets that were not received.

However, the half-duplex nature of the underwater modems causes the feedback to incur a round-trip penalty and consequently reduces the efficiency of the protocols.

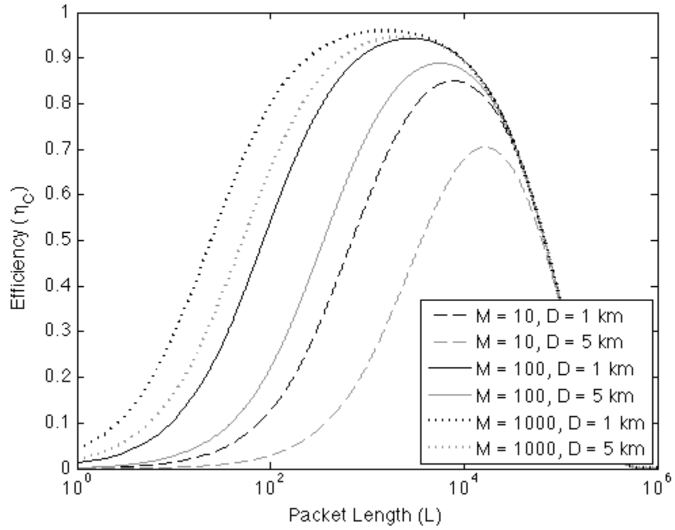

Fig. 2. Efficiency of packet cluster acknowledgement protocol

At alternative approach is provided by the use of rate-less codes, where the sender does not need the knowledge of the successful packets. The sender generates a virtually infinite series of encoded packets based on the file to be transferred and transmits them. When the receiver has received sufficient number of packets, it is able to decode them and recover the file.

As a feedback channel is not required in this protocol, the round-trip penalties are avoided and the protocol does not suffer from a reduced efficiency at longer ranges. However, a practical implementation of this protocol requires a strategy for the sender to stop transmitting additional packets. Either the sender can stop transmitting after it has decided that the probability that the receiver has not received enough packets is smaller than a preset threshold, or a feedback channel is needed to inform the sender of successful decoding of the file.

### *B. Rate-less Codes*

Elias introduced the erasure channel with independent constant probability $\delta$, of packet loss for each packet [8]. He proved that the channel capacity of such a channel is 1-$\delta$. Elias' codes, Reed Solomon codes and Tornado codes can be used to transmit information over such channels. However, the design of these codes requires the probability $\delta$ to be known beforehand. A more general erasure channel with no constraint on the erasure probability is known as a free erasure channel. A class of codes known as rate-less codes has been developed for communications in such channels. Many different codes such as LT codes [3], Raptor codes [4] and online codes [1, 2] belong to this category. In this paper we use online codes for performance analysis, as they are near optimal and can be encoded in constant time and decoded in linear time.

### *C. Performance Analysis*

Online rate-less codes can be decoded from any set of $N(1+3\varepsilon)$ coded packets with an asymptotic success probability of $(\varepsilon/2)^{q+1}$ where $\varepsilon$ and $q$ are parameters of the code [1]. Although the success probability can be made arbitrarily small by choosing small $\varepsilon$ and large $q$, the corresponding increase in complexity of the algorithm limits practical choices. A value of $\varepsilon = 0.01$ and $q = 3$ have been recommended in literature. These values yield a probability of failure of about $10^{-9}$ with only 3% increase in required packets for large $N$. With a finite $N = 1000$, this probability increases to $10^{-8}$, which is still quite acceptable.

In order to decode the file with a high probability, the receiver must have successfully received $N(1+3\varepsilon)$ packets. On an average, the transmitter will have to transmit $N_R$ coded packets so that $N(1+3\varepsilon)$ packets are successfully received, such that

$$N_R = \frac{N(1+3\varepsilon)}{P_D}. \tag{17}$$

Assuming negligible processing times, the average time required to transfer the file successfully is then

$$\begin{aligned} T_R &= N_R \frac{H+L}{R} + \frac{D}{c} \\ &= \left\lceil \frac{F}{L} \right\rceil \frac{(1+3\varepsilon)(H+L)}{(1-p)^{H+L} R} + \frac{D}{c}. \end{aligned} \tag{18}$$

Assuming F >> L, the average file transfer speed can thus be computed as:

$$S_R = \frac{F}{T_R} = \left[ \frac{(1+3\varepsilon)(H+L)}{LR(1-p)^{H+L}} + \frac{D}{Fc} \right]^{-1}. \tag{19}$$

By comparing this speed with the link transmission rate, we can estimate the efficiency of the protocol:

$$\begin{aligned} \eta_R &= \frac{S_R}{R} \\ &= R^{-1} \left[ \frac{(1+3\varepsilon)(H+L)}{LR(1-p)^{H+L}} + \frac{D}{Fc} \right]^{-1} \\ &= \left[ \frac{(1+3\varepsilon)(H+L)}{L(1-p)^{H+L}} + \frac{DR}{Fc} \right]^{-1}. \end{aligned} \tag{20}$$

As we choose $\varepsilon \ll 1$, we can ignore the $3\varepsilon$ term. Assuming a large file size $F$, the term $DR/Fc$ becomes small and can be ignored. Hence we have an asymptotic result,

$$\eta_R \approx \frac{(1-p)^{H+L}}{1+\dfrac{H}{L}}. \tag{21}$$

The efficiency of this protocol is independent of the range of transmission, sound speed and link data rate. This makes the protocol attractive for use in many environments. The numerator limits the efficiency at large $L$ while the denominator limits the efficiency when $L$ is small. The maximum efficiency is obtained at intermediate values of $L$.

Fig. 3 shows a plot of efficiency against packet length for $H$ = 16 bits and two different values of $p$. The optimal packet lengths are in the range of $10^2$-$10^3$ bits and exhibit high efficiency. For $p = 10^{-5}$, the maximum efficiency is about 0.98.

Although the efficiency of the rate-less code based protocol is very high for large files, it reduces for smaller files. The number of packets required for the asymptotic result to hold approximately is $O(10^3)$. With packet length of $O(10^2$-$10^3)$ bits, the file size required is $O(10^5$-$10^6)$ bits or 10-100 kbytes.

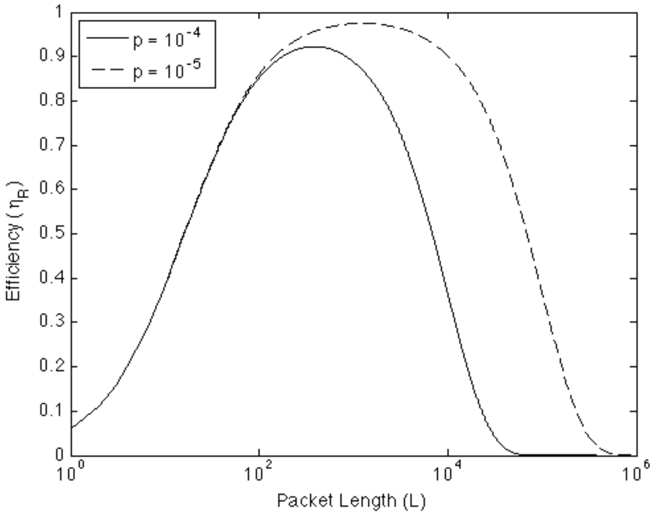

Fig. 3. Efficiency of the rate-less code based protocol

## IV. DISCUSSION

In Fig. 4, we compare the performance of single packet acknowledgement protocol, packet cluster acknowledgement protocol and the rate-less code based protocol for $D$ = 5 km and $p = 10^{-4}$. The performance of the single packet acknowledgement protocol is very poor with a maximum efficiency of less than 0.1. The cluster acknowledgement protocol performs much better with maximum efficiency in the range of 0.6 to 0.8 depending on the cluster size. The larger the cluster size, the better the performance. With a cluster size of 1000 packets, optimal performance requires a packet length of about 1000 bits. The rate-less code based protocol performs even better with a maximum efficiency of 0.9 at packet sizes of only 100 bits.

From the above analysis it is clear that rate-less codes are well suited to underwater file transfers due to the half duplex nature of the modems and the slow sound speed. Although the analysis presented assumes large file sizes, the protocol described can be used with smaller file sizes too. Keeping the number of packets constant, the packet length could be reduced with a loss of efficiency estimated as seen in Fig. 4. Alternatively the packet length could be set to the optimal value and the number of packets could be reduced. This also results in a loss in efficiency that can be estimated using Monte-carlo simulations.

The single packet acknowledgement protocol is inefficient but exhibits single packet latency. This low latency is particularly useful for streaming applications. The cluster acknowledgement protocol detects lost packets only at the end of a cluster. Hence the average latency is half the cluster size. While a large cluster size improves the efficiency, it also increases the latency incurred in the protocol. The rate-less code based protocol recovers all the data in a file only at the end of the transfer. Although this protocol uses the bandwidth very efficiently, it suffers from latency equal to the file size. This is undesirable in most streaming applications. One way to address this issue is to divide the data stream into blocks of $10^5$-$10^6$ bits. Each data block is sent after the earlier one is transferred, thus utilizing the channel efficiently and maintaining a constant latency of one block length. Due to the low data rates in underwater communications, this latency may translate to several 100 seconds in a streaming application. If this latency can be tolerated, the channel can be used very efficiently. The effective data rate for streaming can thus be traded against the latency by using block size as a tunable parameter.

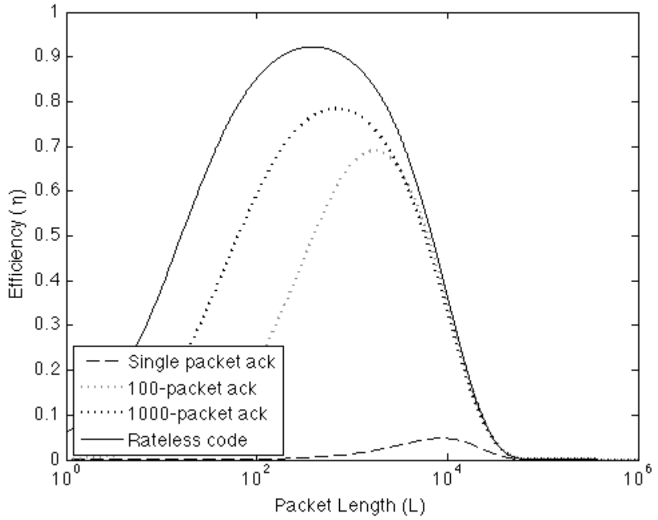

Fig. 4. Comparison of efficiency of various protocols (single packet acknowledgement protocol, cluster acknowledgement protocol for $M$ = 100 & $M$ = 1000 and rate-less code based protocol) for $D$ = 5 km and $p = 10^{-4}$

## V. CONCLUSION

In this paper, we have studied three different protocols for file transfer in an underwater communications context. We have shown that a packet-by-packet acknowledgement results in very poor utilization of the channel, especially at long ranges. By combining acknowledgements for a cluster of packets, the channel can be used more efficiently. The larger the cluster size, the better the efficiency. However, a completely different approach using rate-less codes allows us to achieve even higher efficiencies. Moreover the packet length required to achieve optimal performance are smaller in case of rate-less codes and therefore easier to implement in many practical applications.

Although the file transfer application studied in this paper is simple, it contains many of the key features in underwater networking. It also has many direct applications today.

However, an even larger set of applications requires underwater networking with more than two communicating nodes. A natural extension of this work will include the study of rate-less codes in such scenarios.

## REFERENCES

[1] P. Maymounkov and D. Mazieres, "Rateless Codes and Big Downloads," in *2nd International Workshop on Peer-to-peer Systems*, Berkeley, 2003.

[2] P. Maymounkov, "Online Codes," New York University Technical Report TR2002-833, October 2002.

[3] M. Luby, "LT Codes," in *IEEE Symposium on the Foundations of Computer Science (FOCS)*, 2002, pp. 271-280.

[4] A. Shokrollahi, "Raptor Codes," *IEEE Transactions on Information Theory,* vol. 52, pp. 2551-2567, 2006.

[5] H. Wang and N. B. Maandayam, "Opportunistic file transfer over a fading channel under energy and delay constraints," *IEEE Transactions on Communications,* vol. 53, pp. 632-644, 2005.

[6] C. H. Yu, Y. C. Chen, and L. J. Chen, "Effective File Transfer for Opportunistic Networks," Academia Sinica Technical Report TR-IIS-06-010, IIS, 2006.

[7] M. Stojanovic, "Optimization of a Data Link Protocol for an Underwater Acoustic Channel," in *IEEE OCEANS'05*, Brest, France, 2005.

[8] P. Elias, "Coding for Two Noisy Channels," in *Information Theory, Third London Symposium*, 1955, pp. 61-76.