A Propagation-delay-tolerant Collision Avoidance Protocol for Underwater Acoustic Sensor Networks

Xiaoxing Guo, Michael R. Frater, Michael J. Ryan
School of Information Technology and Electrical Engineering,
University College, the University of New South Wales, Canberra, ACT 2600, Australia
Tel: (+61) 2 6268 8052 Fax: (+61) 2 6268 8443
E-mail: {x.guo, m.frater, m.ryan}@adfa.edu.au

Abstract— **Underwater acoustic sensor networks can be employed by a vast range of applications, retrieving accurate and up-to-date information from underneath the water surface. Although widely used by radios in terrestrial sensor networks, radio frequencies do not propagate well underwater. Acoustic channels are therefore employed as an alternative to support long-distance and low-power communication underwater, even though such channels suffer from long propagation delay and very limited bandwidth. In this paper, we investigate the impact of the large propagation delay on the throughput of selected classical MAC protocols and their variants, and show that protocols need to be revised to accommodate large propagation delay in order to achieve good throughput. We then introduce a propagation-delay-tolerant collision avoidance protocol named PCAP and show that by taking into account the propagation delay, PCAP offers higher throughput than the protocols that are widely used by conventional wireless communication networks.**

I. Introduction

Recent advances in embedded system and communication technologies have pushed a higher level of functionality into ever-smaller devices capable of sensing and wireless communication. Networks formed by these devices, known as wireless sensor networks, have been attracting a tremendous amount of research efforts for their huge potential in both civil and military domains. For example, terrestrial sensor networks, which are enabled by RF radios, have been widely discussed in the wireless communication community.

As over 70 percent of the earth's surface is covered by water, it is desirable to deploy underwater sensor networks to support oceanic research. However, RF signals deliver very poor performance in the underwater environment, providing transmission ranges of merely a few metres at the power used by typical terrestrial sensors. Consequently, acoustic networks become ideal alternatives since sound waves propagate well underwater and require less power than RF radios. Nonetheless, acoustic signals travel much more slowly than RF (approximately 1500 m/s underwater), and have very limited bandwidth delivering data rates of merely a few kbps. Moreover, underwater acoustic sensor networks (UASNs, for short) are envisaged to monitor much larger areas then those on the ground. As a result, propagation delay becomes significant in UASNs. Coupled with low data rates, the long propagation delay means that UASNs greatly challenge networking concepts developed for their terrestrial counterparts enabled by RF radios, where propagation delay is usually negligible.

Several key aspects of UASNs have been investigated by Akyildiz *et al* [1], where they concentrated on general network architectures and channel characteristics. Xie and Gibson [7] proposed a networking protocol that employs one or more master nodes to coordinate and manage other ordinary sensors. Foo *et al* [2] proposed a networking protocol for underwater acoustic networks, which is modified from the AODV protocol [4]. Rodoplu and Park [5] introduced an energy-efficient MAC protocol, which schedules listening and transmission for sensors. Shahabudeen and Chitre [6] studied the performance of several classical MAC protocols that are of different levels of functionality.

The rest of this paper is organised as follows. In Section II we describe the network structure used in our simulations. In Section III we review some of the classical MAC protocols that are widely used in RF networks, and then evaluate the performance of these protocols in terms of throughput. We then propose a new collision avoidance protocol called PCAP in Section IV and investigate its throughput accordingly. Section V concludes this paper.

II. Simulation Network Structure

Underwater acoustic sensor networks can be employed by a vast range of applications, such as ocean sampling, environmental monitoring and undersea explorations [1]. However, because acoustic channels suffer from both significant propagation delay and very limited bandwidth, transferring large amounts of data is generally not attractive in UASNs. Therefore, UASNs will more likely be utilised by applications that only request numerical readings, such as conductivity, temperature and depth measurements. Moreover, since the long propagation delay is physically unavoidable, applications in UASNs must be tolerant of delay.

The network topologies used in the simulations throughout this paper are shown in Fig. 1. We assume that two networks of small (5 nodes) and moderate (20 nodes) scales are deployed in the same depth of water. All nodes of a network are randomly deployed in an area of 4500 by 4500 square metres, resulting in a maximum possible node distance of 6363 metres. All underwater sensor nodes carry half-duplex omni-directional transceivers of a uniform type that allow them to transmit and receive data. All nodes are able to hear each.

1-4244-0138-0/06/$20.00 ©2006 IEEE

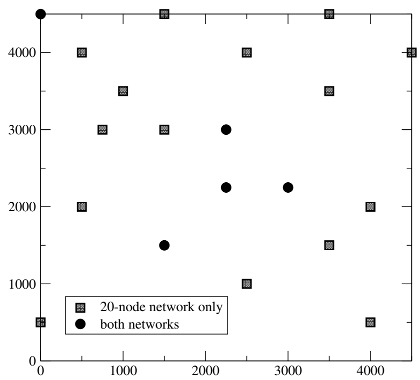

Fig. 1. Topologies of the two networks.

We assume that each node maintains a transmit queue of unlimited size. Moreover, the acoustic channel is of zero-error rate and zero-path loss. Consequently, the only packet loss is caused by collisions. *Data frames* are generated at each node in a network in accordance with a Poisson distribution, and each data frame is to be transmitted to a dedicated destination node located at (0,4500) in the 5-node network and (3500,1500) for 20 nodes. Offered load and throughput are employed to measure the performance of various protocols, which are defined by (1) and (2) respectively [6].

$$offered\ load = \frac{\frac{total\ data\ frames\ transmitted}{simulation\ time}}{\frac{signalling\ rate}{dataframe\ size}} \quad (1)$$

$$throughput = \frac{\frac{total\ data\ frames\ received}{simulation\ time}}{\frac{signalling\ rate}{dataframe\ size}} \quad (2)$$

A data frame comprises the actual user data to be transmitted by the network between a *source* and a *destination* node. *Control frames*, including RTS/CTS and ACK frames, are employed to coordinate the transmission in the network. Data and control frames are 3200 and 96 bits respectively. Nodes in the network that are neither a source nor a destination for a data frame are potential *interferers* of that frame. A data frame is processed by a node if the node is either transmitting it or performing handshaking or sensing the carrier for such a frame.

III. Classical MAC Protocols Investigated

In this section, we investigate the throughput of some of the well-known MAC protocols and variants in the presence of large propagation delay.

Pure aloha (ALOHA) was implemented. Due to the high propagation delay of underwater acoustic signal however, neither channel listening nor retransmission was implemented because a half-duplex transceiver is not able to listen to the channel feedback while transmitting and the channel feedback may be heard after a fairly long time due to the long propagation delay.

In *Aloha with carrier sense (ALOHA-CS)*, a sensor senses the carrier for a period of time (corresponding to the data frame length for low delay channels or the average propagation delay for high delay channels) prior to sending the next queued data frame. A node will not transmit if it finds the channel is occupied.

Aloha with acknowledgement (ALOHA-ACK) requires an acknowledgement (ACK) frame from the destination node to determine whether or not the transmission is successful. A node will not transmit the next queued data frame until it receives the ACK frame for the current one or times out if it has not received the ACK frame after twice of the maximum single trip time since the transmission of the current data frame is completed.

Handshaking (HDSK) is a more complex protocol that employs short request-to-send (RTS)/clear-to-send (CTS) frames before actually sending a data frame. A source performs handshaking with the destination by sending an RTS and receiving a CTS frame. On reception of such RTS/CTS frames, interferers are blocked for a certain period of time that is enough long to ensure they will not garble the on-going transmission. A node will not perform any action for the next queued data frame if the on-going transmission is not completed.

Handshaking with acknowledgement (HDSK-ACK) requires an ACK frame from the destination to conclude if the transmission of the data frame is successful. If the transmission has failed, it retransmits until a maximum number of retries (≥ 0, set by user) is reached. As with HDSK, a node only processes one data frame at a time.

A. Performance of classical protocols with no acknowledgement and retransmission

We investigated the throughput of the protocols when the signalling rates are 200, 2400 and 9600 bps. This is because varying the signalling rate effectively changes the ratio of the propagation delay to the lengths of both the data and control frames. This is equivalent to changing the significance of the propagation delay in the simulation. The performance of the protocols with no acknowledgement and retransmission can be seen in Fig. 2.

The throughput of ALOHA roughly matches the theoretical values and peaks at approximately 0.18 when the offered load is 0.5. The performance remains the same when the network scales from 5 to 20 nodes and does not change with different signalling rates.

ALOHA-CS delivers results that are slightly worse than ALOHA when the signalling rate is 200 bps. However, its throughput becomes worse when the signalling rate increases, peaking at 0.03 when the offered load is about 0.1 and the signalling rate is 9600 bps. This is because ALOHA-CS lingers over sensing the carrier, resulting in lower offered load and throughput.

HDSK offers good throughput under 200 bps because the propagation delay (2 s on average) is relatively small compared to the length of the data frame (16 s). However, its throughput drops when the propagation delay becomes more significant. This is due to the fact that when processing a data frame,

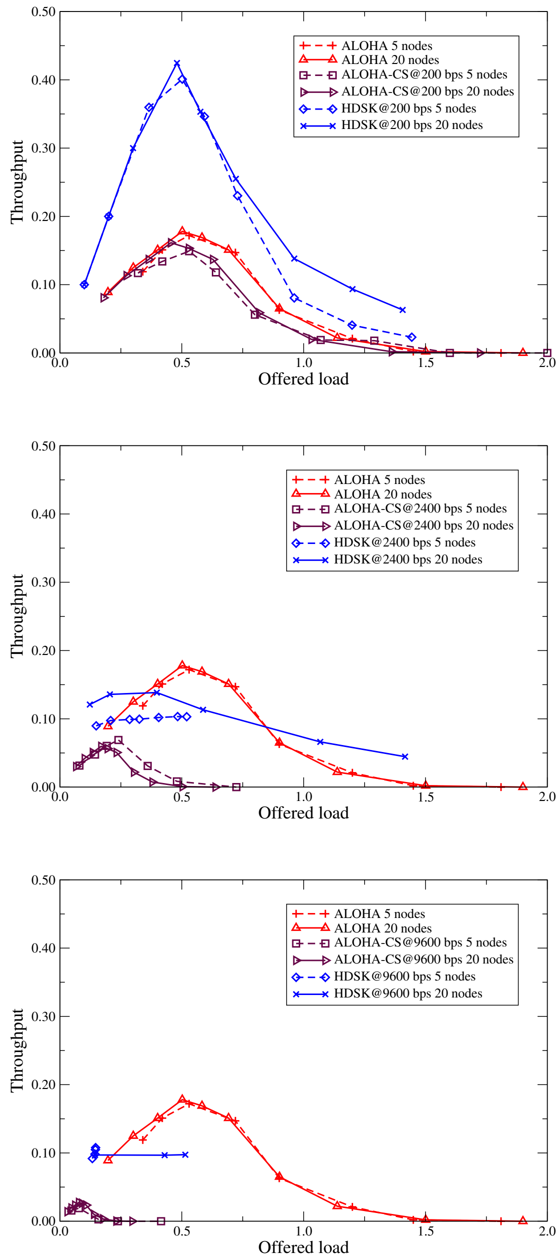

Fig. 2. Throughput of the classical protocols without ACK and retransmission.

the node is stalled, waiting for the RTS/CTS frame to be propagated, and the next data frame cannot be processed while the previous transmission is still live. The throughput is also slightly lower when the network size grows.

Furthermore, HDSK has very poor throughput when the signalling rate is higher than 2400 bps. In a network where the propagation delay is negligible or very low, receipt of RTS/CTS frames at interferers is virtually instantaneous. Since the RTS/CTS frames carry enough information to inform the interferers to not to interfere with the on-going transmission, collisions can be reduced. However, if the propagation delay is dominant, RTS/CTS frames may only reach certain interferers when the transmission of the data frame has commenced. Worse, due to the propagation delay, interferers will not have enough knowledge from the RTS/CTS frames to conclude when they should be blocked. Consequently, this elegant collision avoidance mechanism for RF networks becomes an obstacle that holds off the network from processing data frames efficiently because transferring control frames involves significantly more time if the propagation delay is large.

Carrier sense and handshaking are efficient means to coordinate transmission of different nodes in a network when the propagation delay is negligible or of least significance. Both carrier sense and handshaking become less useful when the propagation delay is considerably larger than the data frame length. As the propagation delay increases, much time is spent on transmitting/waiting control frames or sensing the carrier to avoid potential collisions and hence the throughput is reduced.

B. Performance of classical protocols with acknowledgement and retransmission

Since the acoustic channel does not have immediate feedback, from which the source node of a data frame can conclude if the transmission is interrupted, an ACK frame sent by the destination is engaged to inform the source of a successful transmission. If the source has not receive the ACK frame after twice of the maximum single trip delay since the transmission of the data frame is completed, it regards the transmission as failed and retransmits until a maximum number of retries is reached. The performance of ALOHA-ACK and HDSK-ACK is presented in Fig. 3.

The offered load of ALOHA-ACK can only be varied within a very limited range. This is because that no matter how fast the data frames are injected into the queues maintained by the nodes in the network and no matter what the signalling rate is, the source node has to wait for the ACK frame or times out. Each unsuccessful transmission will result in a time-out that contributes twice of the maximum single trip time to the time spent on processing the data frame. As a consequence, this degrades both the effective offered load and the throughput of the network. Meanwhile, the number of retries allowed for each data frame also affects the offered load and throughput in the same way. More retries generally mean the network needs to spend longer time on processing a data frame. Although the number of data frames successfully transferred by the network increases by using retries, the throughput is impaired due to the longer processing time.

As with ALOHA-ACK, HDSK-ACK encounters the same situation that the time spent by the network on transferring each data frame is significantly protracted. In spite of the fact that more data frames are successfully carried by the network, the throughput and offered load are reduced.

ACK and retransmission degrade the throughput for all the classical protocols studied if the propagation delay is significant. Although retransmission improves the success rate of transferring queued data frames, the greatly impaired

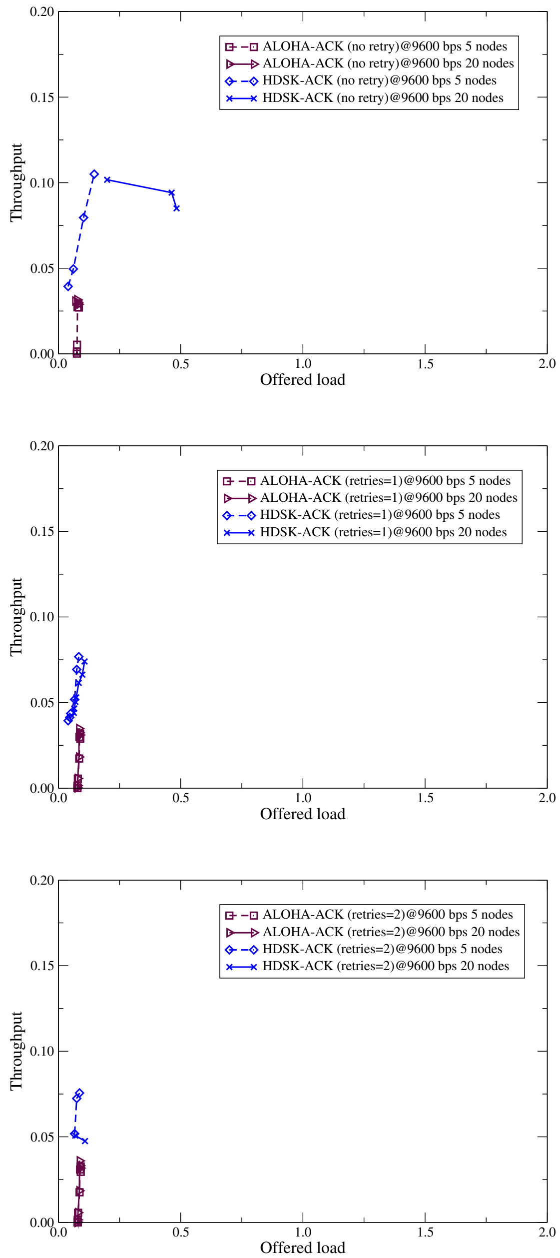

Fig. 3. Throughput of the classical protocols with ACK and retransmission. Maximum numbers of retries are given in brackets.

throughput implies that this effort seems to be worthless under the high delay environment.

Handshaking and acknowledgement schemes become less efficient as the growth of the propagation delay. Specifically, handshaking by RTS/CTS frames does not work effectively when the propagation delay is considerable. In an RF network, RTS/CTS frames block interferers virtually instantaneously because radio waves propagate very quickly. In an UASN, however, they are not able to block interferers effectively because they may reach some interferers when the transmission of the data frame has already commenced. ACK and retransmission add significant delay to the time spent by the network to transfer a data frame, they therefore impair the throughput much more than the benefit they bring to it. It is obvious that the classical protocols waste considerable time on transferring control frames and waiting for responses. Therefore, it would be desirable if a protocol can utilise the time spent on waiting for acoustic signals to propagate.

IV. A Propagation-Delay-Tolerant Collision Avoidance Protocol

In this section, we propose a propagation-delay-tolerant collision avoidance protocol (PCAP). The objective of PCAP is to fix the time spent on setting up links for data frames, and to avoid collisions by scheduling the activity of sensors.

A. *Overview of PCAP*

Devised along the same line as HDSK, PCAP exchanges short RTS/CTS frames between source and destination nodes before the data frame is transmitted. Once a source node has a data frame to transmit, it sends a short RTS frame, and waits for a CTS from the desired destination node. An RTS frame includes information such as the IDs of the source and destination, a time stamp, information about the data frame to be transmitted, and the proposed start/end time of transmitting the data frame. In addition to the information in the RTS frame, a CTS frame consists of information such as the signal strength of the received RTS frame and a time stamp. When the source node receives the replied CTS frame, it starts transmitting the data frame at the proposed time. Since an RTS/CTS frame also indicates the proposed start/end time of the transmission of the actual data frame, it blocks the interferers who receive it to avoid collisions when the transmission is in progress.

Assume that there are two nodes a and b, and a has a data frame to send to b. If a does not know the distance to b, it assumes that the CTS frame replied by b may arrive at any time. Consequently, a has to keep listening to the channel and make itself available until the CTS frame arrives or the RTS frame times out. Since the propagation delay of the acoustic channel is large, the CTS frame may not arrive for several seconds depending on the distance between the nodes. As a result, a may spend large amounts of energy and time while listening to the channel and waiting. This also provides an explanation for the poor performance of the classical MAC protocols investigated in Section III.

The basic idea of PCAP is to allow a to take other actions (e.g. to transmit another data frame or perform handshaking for the next queued data frame) while waiting for the CTS frame from b and subsequently remain available to receive it at a proper time. This is achieved by forcing b to wait for a certain period of time before sending the CTS frame to a. Since we assume that all nodes in a network are quipped with the same type of transceivers offering the same transmission range, the maximum propagation delay for all nodes can then be calculated. Let T_d denote the maximum propagation delay for the acoustic signal from a to reach b; t_{tx} denote the absolute time the RTS frame is transmitted, and t_{rx} denote the time the frame is received. The actual time spent by the RTS frame from a to reach b is then $t_{rx} - t_{tx}$. Instead of sending a CTS frame immediately when the RTS frame is received, b then waits for an additional period of time T_a calculated as follows:

$$T_a = 2(T_d - (t_{rx} - t_{tx})) \tag{3}$$

As a result, a receives the CTS frame after $2T_d$ as if b was located at the maximum transmission range of a. Consequently, a is able to devote itself to other actions that can be finished by the time the CTS frame arrives.

Since (3) uses the absolute time difference $t_{rx} - t_{tx}$ to calculate T_a, PCAP requires accurate timekeeping. A commercial RTC/TCXO/crystal integration device can provide accuracy of ± 2.0 ppm from 0 to 40 degrees Celsius [8]. Since the time difference is only calculated between neighboring nodes, timekeeping error does not propagate in the network. Calibration can be employed on a regular basis to compensate for the timing error introduced by clock drift. Any error is also very small compared to the propagation delay.

Once the RTS/CTS handshaking is finished, the link is established and a will be consequently ready to send the actual data frame to b. We define the link establishment time to be the time between the RTS frame is transmitted and the CTS frame is received. Clearly, forcing b to wait additional T_a time increases the link establishment time. If nodes in a UASN periodically have data to send, they are able to establish and reserve links prior to the next data frame being generated, and consequently the delay incurred by employing PCAP need not affect the transmission of data frames.

Compared with the classical protocols investigated in Section III, PCAP makes the propagation delay predictable by allowing a node to perform other actions while waiting for the CTS frame to establish the link. Since the propagation delay of underwater acoustic channels is long and unavoidable, it is possible for a node to be involved in transmitting other data frames instead of wasting its time when the signals propagate. In other words, with PCAP, a node transmits data frames as if it had a pipeline for transmitting. Since most actions are scheduled when a new data frame is processed from the queue, the source node fits an action in its schedule once it identifies that the action can be completed within its idle time. As a result, there can be more than one simultaneous data frame being processed at a node and scheduling also becomes feasible.

B. Performance of PCAP

We first implemented PCAP without acknowledgement, in which case nodes also do not retransmit if data frames fail in transmission. The results are presented in Fig 4.

It can be seen that PCAP offers comparable throughput to ALOHA when the signalling rate is 200 bps, in which case the average propagation delay is roughly 1/8 of the length of the data frame. And the scale of the network does not distinctly affect the throughput. Similar to the protocols investigated in

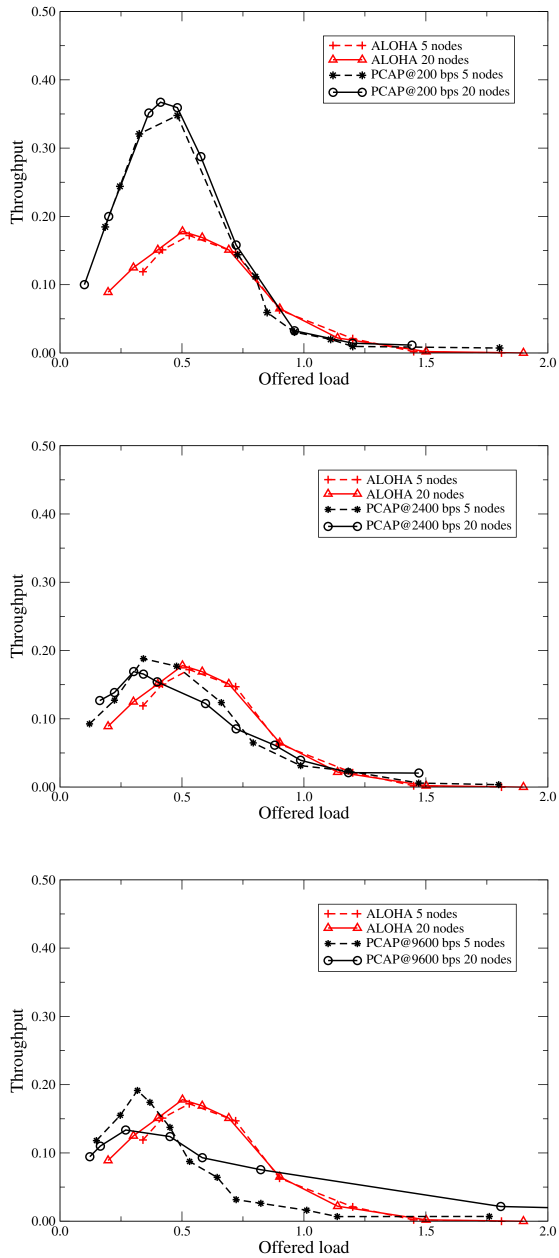

Fig. 4. Throughput of PCAP without ACK and retransmission.

Section III, the overall throughput of PCAP drops while the signalling rate increases, which indicates that the propagation delay becomes more significant.

For both network scales, the peak throughput of PCAP is roughly 20% higher than HDSK when the data is transmitted at 2400 bps and above. It reaches its peak when the offered load is between 0.3 and 0.4 for both the 2400 bps and 9600 bps cases. While the peak throughput of PCAP attenuates with the growth of the network scale and the throughput decreases less steeply as the offered load increases. This is due to the fact that for a larger network, the average node distance is less than for the smaller network (leading to shorter overall delay when transmitting data frames), but more nodes may contend for the channel at the same time (resulting in a higher probability of having collisions). Clearly, PCAP outperforms other protocols investigated when the propagation delay becomes more significant.

We then implemented PCAP with acknowledgement and retransmission. In addition to scheduling the time to receive CTS and data frames, the protocol also reserves time to receive the ACK frames. A retransmission is scheduled if the source does not receive the ACK frame after twice of the maximum single trip time between itself and the destination. The results are shown in Fig. 5.

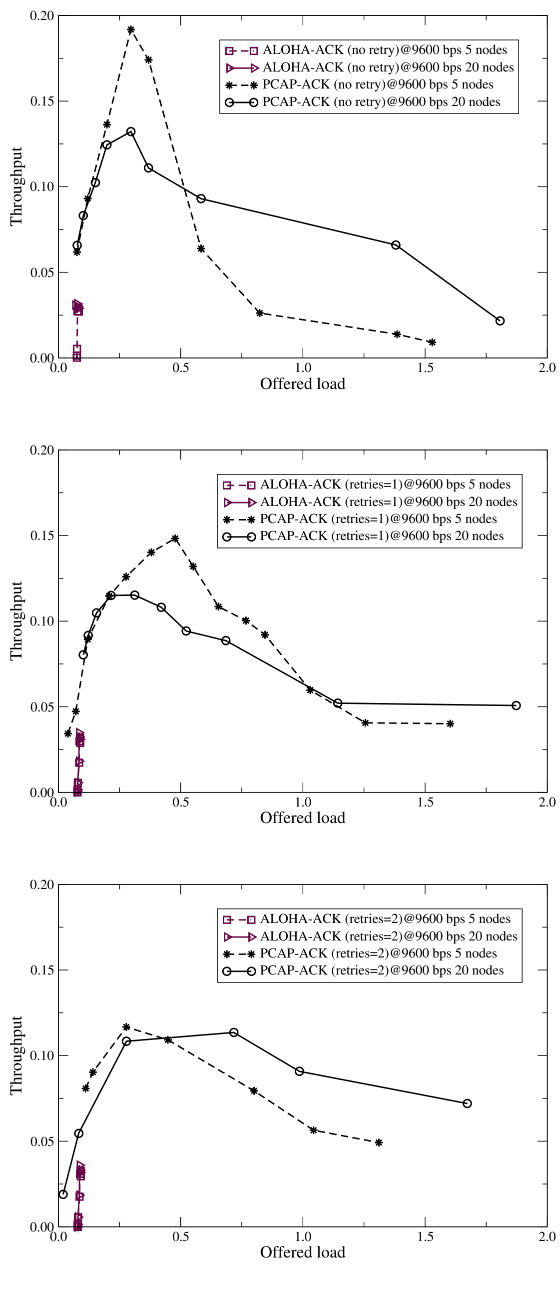

Fig. 5. Throughput of PCAP with ACK and retransmission. Maximum numbers of retries are given in brackets.

It is clear that the engagement of ACK frames does not significantly impair the performance of PCAP, as it does to the other protocols in Section III. Like the other protocols, more retries mean longer time to process per data frame, and hence lead to a lower throughput. Except for the lower peak throughput, PCAP with ACK and retransmission exhibits the same trend as PCAP without these mechanisms.

Clearly, PCAP offers higher overall throughput than the other protocols we have studied in this paper when the propagation delay becomes dominant.

V. Conclusions

In this paper, we show that the performance of a variety of classical MAC protocols is very poor in circumstances where there is a large propagation delay, and have demonstrated the direct relationship between this poor performance and the propagation delay. A new protocol, the propagation-delay tolerant collision avoidance protocol (PCAP), is proposed that overcomes these limitations by allowing stations to schedule actions for multiple data frames, reducing the amount of time wasted in waiting for signals to travel over high-delay channels.

References

[1] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. *Elsevier's Journal of Ad Hoc Networks*, Vol. 3, Issue 3, pp. 257 – 279, 2005.

[2] K. Y. Foo, P. R. Atkins, T. Collins, C. Morley, and J. Davies. A routing and channel-access approach for an ad hoc underwater acoustic network. *Proc. MTS/IEEE OCEANS'04*, pp. 789 – 795, 2004.

[3] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. *ACM SIGOPS Operating Systems Review*, Vol. 34, No. 5, pp. 93 – 104, 2000.

[4] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90 – 100, 1999.

[5] V. Rodoplu and M. K. Park. An energy-efficient MAC protocol for underwater wireless acoustic networks. *Proc. MTS/IEEE OCEANS'05*, 2005.

[6] S. Shahabudeen and M. A. Chitre. Design of networking protocols for shallow water peer-to-peer acoustic networks. *Proc. IEEE/OES Oceans'05 Europe*, 2005.

[7] G. G. Xie and J. H. Gibson. A network layer protocol for UANs to address propagation delay induced performance limitations. *Proc. MTS/IEEE OCEANS'01*, pp. 2087 – 2094, 2001.

[8] Maxim Integrated Products. `http://www.maxim-ic.com`