This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Adaptive Multimode Medium Access Control for Underwater Acoustic Networks

Shiraz Shahabudeen, *Member, IEEE*, Mandar Chitre, *Senior Member, IEEE*, and Mehul Motani, *Member, IEEE*

***Abstract*—Standards such as 802.11 have played a key role in the success of terrestrial radio wireless communications. Similar standardization will be needed in underwater acoustic networks (UANs) of the future. One of the important aspects of standardization is UAN medium access control (MAC). Since no single protocol can satisfy the diverse requirements of a general UAN MAC, we explore the possibility of combining multiple MAC protocols into a suite. We also consider physical-layer adaptation techniques as they are closely related to the MAC adaptation. The suite's key protocol mode called MACA–EA is a novel, enhanced adaptation of multiple access with collision avoidance (MACA) as used in 802.11. The suite uses two other modes—a centrally polled mode called MACA–C and a simple DATA–ACK protocol. Using both simulations and mathematical analysis, we compare saturated throughput performance and waiting time performance (in case of Poisson arrivals) of the different MAC protocol modes. We also benchmark the performance against ideal time-division multiple access (TDMA) performance. We present suitable adaptation techniques to switch between the protocol modes based on network requirements, traffic intensity, and quality-of-service QoS) requirements such as maximum allowed waiting time for reliable transfer. We propose an adaptation algorithm for automatically varying the batch size in MACA–C and MACA–EA for optimum performance. A key observation is that for *ad hoc* UANs, the adaptation of the multiple modes can lead to near-optimal performance across a wider range of traffic intensity, as compared to what any single protocol can achieve.**

***Index Terms*—Adaptive protocol suite, medium access control (MAC), multimode, standardization, queueing analysis, underwater acoustic networks (UANs), waiting time.**

I. Introduction

UNDERWATER modems today use diverse physical-layer standards and medium access control (MAC) protocols in overlapping frequency bands and are unable to coexist or communicate with each other over standardized protocols. A few initiatives have sought to standardize underwater acoustic network (UAN) physical-layer and MAC protocols for global interoperability [1], [2]. As part of the JANUS initiative [2], a suitable candidate for an adaptive MAC protocol suite for UAN was previously proposed [3]. This paper presents important enhancements and an in-depth analysis of this adaptive protocol suite, which will henceforth be referred to as adaptive multimode MAC (MAC–AMM).

In a general UAN scenario, a single MAC protocol and a single physical-layer type cannot cater to differing deployment and traffic requirements. The first problem is the existence of numerous underwater acoustic modems that are incompatible with each other. How can we provide adaptation mechanisms to allow them to coexist and communicate, if deployed in the same geographical area? The answer lies in standardization. The second problem, quite independent of the first, is how MAC protocols can autonomously switch between centralized topology and distributed topology. For example, in a centralized topology network, the master controller (MC) could fail. Can the client nodes communicate with the other nodes in the absence of the MC? Or in a network currently operating in distributed mode, if a gateway buoy is introduced to operate as an MC, can the MC take charge of the MAC coordination seamlessly? Related to this, there is the question of how the centralized topology performance compares with the distributed mode. The third problem is related to traffic intensity. As we will see in Section V-F, the relative performance of protocols is dependent on traffic intensity. As traffic varies, can modems adaptively switch between protocol modes to provide the best performance? The need for a robust and flexible heterogeneous UAN exists, and the above three MAC problems require a unified solution.

It is necessary for a universal robust UAN MAC to be able to work without time synchronization but at the same time be able to utilize time synchronization when available, to improve performance. The need for *ad hoc* functionality in the network, i.e., nodes joining and departing is important. The issue of geographical scalability and robustness also cannot be ignored. In the hunt for a basic protocol model for use in UANs, MACA-based schemes stand out as a good choice from among the many alternatives [4], [5]. Other protocols proposed for UANs such as the distance-aware collision-avoidance protocol (DACAP) [6] and the propagation-delay-tolerant collision-avoidance protocol (PCAP) [7] are also based on MACA. The choice of MACA-based protocols for UANs has also been discussed at length in [8].

The MAC–AMM protocol has two levels of operation: level-1 to achieve coexistence, and level-2 to achieve communications among heterogeneous assets. In this adaptive protocol, nodes dynamically adapt their physical-layer and MAC protocol modes based on node capability, network scenario, and traffic intensity. Level-1 MAC has a single mode: DATA–ACK, where there is no request-to-send/clear-to-send

Manuscript received August 07, 2011; revised March 17, 2012 and November 26, 2012; accepted June 03, 2013.
Associate Editor: J. Gomes.

S. Shahabudeen was with the Electrical and Computer Engineering Department, National University of Singapore, Singapore 117576, Singapore. He is now with the Technology Development Center, NeST Software, Kerala 695581, India (e-mail: shirazs@gmail.com).
M. Chitre is with the Electrical and Computer Engineering Department and the Acoustic Research Laboratory (ARL), National University of Singapore, Singapore 117576, Singapore (e-mail: mandar@arl.nus.edu.sg).
M. Motani is with the Electrical and Computer Engineering Department, National University of Singapore, Singapore 117576, Singapore (e-mail: motani@nus.edu.sg).

Digital Object Identifier 10.1109/JOE.2013.2268273

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

(RTS/CTS) handshake before sending DATA and an acknowledgement (ACK) is sent to indicate correct reception of DATA. Level-2 MAC has three modes: a distributed MACA-based mode called MACA–EA (see Section II-C), a centrally controlled polled mode called MACA–C (see Section II-D), and a low traffic DATA–ACK mode (see Section II-E). Our main focus here is on the level-2 MAC that allows full-fledged communications among all nodes in a network. Various modes, as outlined above, are described in Section II, along with details of how the dynamic adaptation between these modes works. The analysis model and performance measures are outlined in Section III. Following that, a throughput performance analysis is presented in Section IV. Mode adaptation based on traffic intensity is illustrated in Section V, using waiting time as the key performance metric. The primary contributions of this paper are:

- a comprehensive adaptive multimode MAC to address a heterogeneous UAN;
- analytical characterization of the DATA–ACK protocol mode with queueing of incoming data and reliable communication with retries;
- analytical characterization of the MACA–C protocol mode for reliable communications;
- waiting time comparison between time-division multiple access (TDMA), MACA–EA, MACA–C, and DATA–ACK;
- illustration of the transition traffic intensity between MACA–EA and DATA–ACK modes for minimizing waiting time;
- a practical algorithm for traffic-based adaptation between the two distributed modes.

II. Protocol Modes in MAC–AMM

A. Level-1 Compliance

Coexistence is defined as the ability to perform communications among modems of the same physical-layer type, while operating in a region where there are modems using different physical layers. In the coexistence mode, all modems should adopt the same detection preamble for a given frequency band. If that is not possible, modems must implement an alien signal detection feature, i.e., the ability to detect a signal in its frequency band that is not of its physical-layer type. This may be based on energy detection. Wakeup tones may be included as part of the preamble structure, such as those being proposed in the JANUS initiative, and these may be used for alien signal detection as well. Energy detection can also be used alongside the detection preamble to monitor the signal following the preamble, to determine end of the packet, for example. Such a minimal compliance at the physical layer is termed level-1. This concept is illustrated in Fig. 1.

With such a minimal compliance at the physical layer, it is possible to use a DATA–ACK protocol to communicate among modems of the same type, while operating in an environment consisting of alien nodes. In this mode, nodes use a random backoff before transmitting a DATA packet. Upon reception of a DATA packet, the receiver sends back an ACK. More details of the DATA–ACK protocol are described in Appendix A. In

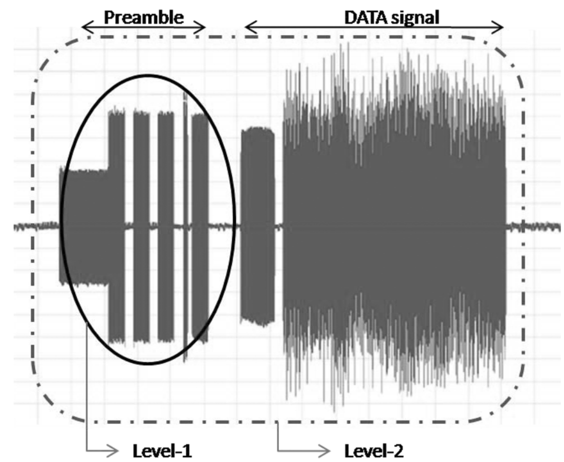


Fig. 1. A sample physical-layer packet structure shows the preamble and data signal portion. Physical-layer compliance levels are as indicated.

fact, this mode is similar to the Basic Access Scheme in 802.11 [9]. At high traffic intensity, this protocol has lower performance compared to other options as we will see. However, at this level of minimal physical-layer compliance, it offers perhaps one of the best solutions for communication and to deal with interference between alien modem types.

B. Level-2 Compliance

To enable communications among heterogeneous assets operating in the same geographical area, modems will be required to implement a standardized physical layer (along with proprietary physical layers, if desired, in addition to the alien signal detector or detection preamble). Such a globally standardized physical layer may evolve through initiatives such as JANUS. This is termed level-2 compliance, as illustrated in Fig. 1.

Level-2 MAC has a distributed MACA-based mode (MACA–EA), a centrally controlled polled mode (MACA–C), and a low traffic DATA–ACK mode that nodes dynamically adopt based on deployment, node configuration, and traffic, as explained in Section II-F. In the centralized mode, a cell is defined to consist of a MAC controller (MC) and the nodes within its control. Many of the centrally controlled MAC protocols use a polling scheme, where the MC polls the client nodes [5]. Some of the distributed protocols include ALOHA, carrier-sense multiple access (CSMA), medium access with collision avoidance (MACA), and floor acquisition multiple access (FAMA) [10], [11]. Among distributed protocols, some protocols such as MACA and FAMA involve handshaking using control packets before data transmission. Centrally controlled modes may offer better performance than distributed modes by eliminating contention. However, in a generic network environment with heterogeneous nodes, a centrally controlled protocol alone may not be usable and distributed modes may be required. Prior work addressing such large-scale *ad hoc* dynamic underwater networks includes the Seaweb project [4]. The terrestrial IEEE 802.11 family of protocols also uses such combination protocols in the form of point coordination function (PCF) and distributed coordination function (DCF). An overview on the choice of suitable MAC protocols for UANs can be found in [8]. It outlines why time-domain protocols are well suited for

UANs and how MACA-based protocols serve well in *ad hoc* UANs without the need for time synchronization.

C. Distributed Mode of Level-2 MAC: MACA–EA

The MACA–EA protocol [12], [13] forms the basis of the distributed mode in the level-2 MAC protocol suite. The protocol is based on MACA using an RTS/CTS exchange [10]. The basic model used is RTS/CTS/DATA–BATCH/ACK. The transmitter sends an RTS and the receiver sends back a CTS. The RTS specifies the number of packets in the batch. The transmitter then sends a batch of DATA packets (DATA–BATCH). The receiver then sends a single ACK which indicates failed packets in the batch. Similar protocols with batch DATA packets that employ ACKs after every packet (RTS/CTS/DATA/ACK/DATA/ACK. . .) are not efficient for UANs due to the two-way propagation delay overhead, and, thus, we use a single ACK for the entire batch of DATA packets. Similar RTS/CTS handshake models have been previously explored in other protocols [6].

In the RTS contention algorithm, a node starts with a uniform probability distributed backoff in a contention window W. Once the contention timer expires, the timer t_A starts and an RTS is sent. If timer expires before reception of the CTS, the RTS backoff procedure starts again. Once a CTS is received, the DATA–BATCH is sent, and the transmitter waits for an ACK. If an ACK is not received, the RTS cycle repeats. Reception of RTS/CTS packets and possible DATA packets while waiting to send an RTS triggers virtual carrier sense (VCS). Successful DATA transmission for any one node restarts the RTS contention cycle for all. Note that 802.11 uses freezing backoff [9] whereas this protocol uses a constant window. Also, this protocol does not use physical carrier sense (PCS) whereas it is used in 802.11. All nodes use the same contention window W.

To make reliable transfer more efficient, we use two variations with regards to acknowledgments and retransmissions to handle failed data packets. After a batch of DATA packets is received, an ACK is sent by the receiver. In typically used retry models, if an ACK fails to reach the transmitter, the RTS/CTS-based contention cycle and the batch DATA transmission processes repeat. Two enhancements were made to this retry process. First, instead of sending one ACK packet, we send i ACK packets, a feature termed multi-ACK. This reduces the probability of ACK loss. As a second enhancement, when the sender of the DATA batch does not receive an ACK, the RTS with the same unique identification number (UID, incremented only for an RTS for a new batch of DATA packets) is repeated. The receiver sends back an ACK instead of a CTS for the repeated RTS. Together with the multi-ACK feature, this is called the early-multi-ACK model. The retry mechanism uses constant backoff with infinite retries (other options include exponential increase exponential decrease, maximum retries capped, etc.)

D. Centralized Mode of Level-2 MAC: MACA–C

In this mode, an MC controller controls the collision domain or "cell." A request-to-receive (RTR) initiates all communication sequences for the uplink (to MC or between nodes in the same cell). All nodes monitor for MC control packets to detect presence of a controlling MC and then switch to the centralized MAC mode. Channels not mentioned in RTR can be assumed to be uncontrolled by the MC and nodes may make use of them as they wish (e.g., using MACA–EA). The nodes that operate as MC may be software preconfigured (e.g., radio buoys). For the uplink, MACA–C operates in few different modes.

- RTR–DATA–ACK: The intended node responds with DATA using the control channel if it is meant for the MC and uses power control information inferred from RTR's received power, assuming bidirectional validity of power information. MC then closes with an ACK. Multiple ACKs may be used to increase receive probability. An ACK may include earliest next RTR timing and help reduce uncertainty.
- RTR–RTS–CTS–DATA–ACK: If the destination is another node (not the MC), or if a node wishes to use a different modulation of forward error correction (FEC) scheme, a node sends out an RTS once the RTR is received. That is followed by a CTS–DATA–ACK just as in MACA–EA. This mode has some similarities to a previously published protocol [5]. The CTS may provide FEC and power control information as described earlier.

For the downlink, the MC uses an RTS–CTS–DATA–ACK sequence just as in MACA–EA.

E. Level-2 Distributed Mode With No Handshaking: DATA–ACK

A DATA–ACK reliable data transfer mode without RTS/CTS exchange is also necessary for the distributed mode. As we show in Section V, for low traffic intensity, RTS/CTS handshaking is not necessary and increases waiting time. Instead, the DATA can be sent using the same backoff process as RTS in MACA–EA. The receiver sends back an ACK when it receives the DATA packet.

F. Adaptive Multimode MAC

In MAC–AMM, a modem assesses its neighborhood and traffic intensity and switches to an appropriate MAC mode from the above choices. When modems do not sense dissimilar modems, they are free to use any physical-layer and MAC protocols. This allows the usage of proprietary technologies and protocols in isolated environments. If modems with only level-1 compliance detect alien signals (hear a standardized preamble followed by indecipherable packet or based on energy detection), they should automatically switch to level-1 DATA–ACK protocol that uses random backoff.

Level-2 adaptation is possible in two ways: modems implement only the standardized physical layer or they implement the standardized physical layer alongside any proprietary modulation/FEC scheme and have mechanisms to switch between them. For nodes using the compliant physical-layer and MAC protocol, there is no change in behavior required. For nodes using compliant physical-layer and noncompliant MAC protocol in isolation hear packets belonging to the standardized MAC protocol (as identified by the type field), they should switch to the standardized level-2 MAC protocol for compliance. Nodes with multiple physical layers (one of which is compliant) operating in nonstandard physical layer in isolation should switch to compliant mode upon detecting alien signals.

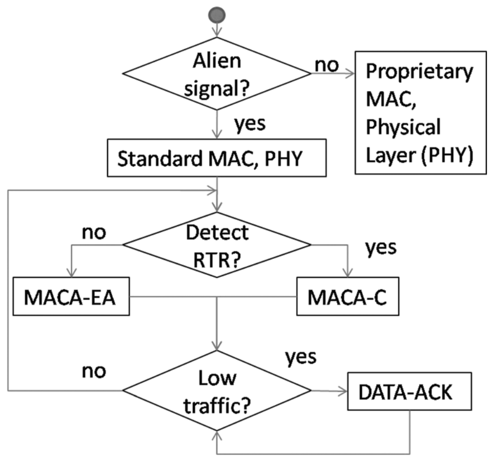


Fig. 2. MAC–AMM adaptation.

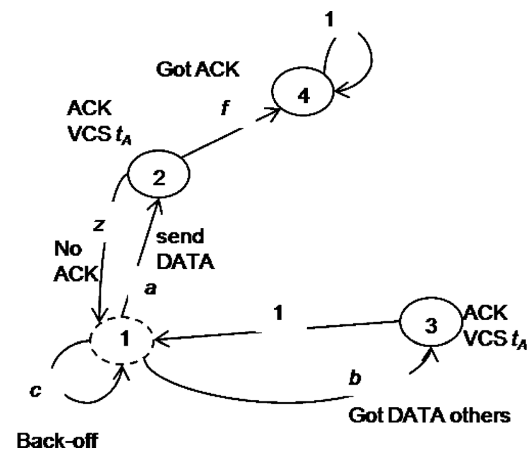


Fig. 3. Markov chain for computing expected service time for DATA–ACK protocol.

In level-2 MAC, there are three modes: MACA–EA, MACA–C, and DATA–ACK. The nodes determine the presence of an MC through RTR messages. If they hear RTR, they use MACA–C. If they do not hear RTR messages, they use MACA–EA. In other words, the switching between MACA–C and MACA–EA is decided by the network deployment and is not automatic. The network automatically switches to DATA–ACK mode if the network is operating in very low traffic intensity (sporadic DATA). In MACA–C and MACA–EA modes, batch size B also needs to be adapted based on traffic intensity. Fig. 2 captures this process for level-2 compliance.

More details on the behavior of the protocol modes with respect to traffic intensity are presented in Section V. The modes discussed here are all for reliable communications, i.e., with retries for failed packets. A brief discussion on broadcasts and unreliable transmissions is provided in Section VI.

III. Analysis Model and Performance Measures

Queuing theory is commonly used in the modeling and analysis of wireless networks. First, we assume a saturated load to model file transfer and other high load applications, and analyze the throughput performance of the network. Then, we model sporadic arrivals such as those generated by random events in underwater sensors, using a Poisson arrival process. A Markov chain analysis is used to study the behavior of the system. Important common metrics derived are service time distribution and its expected value, saturated throughput, expected steady-state queue length, and expected total waiting time. We show that the service time is approximately exponentially distributed. Although for a specific deployment the traffic arrival pattern may be different (e.g., bursty or uniform), the two models that we present provide key insights into the performance of the network for high- and low-load situations.

Each packet is assumed to have a fixed length detection preamble at the start. Detection probability P_d is dependent on the nature of the preamble. Packet decoding probability P is determined by the bit error rate (BER) of the physical layer, the number of bits in the packet, and the coding scheme. The probability that a packet is detected and decoded correctly is

$$k = P_d P. \tag{1}$$

Control and data packets may use different modulation, coding, and packet length, as robustness is of key importance to control packets while data rate is of importance in data packets. To model this, we allow the decoding probability P_D of data packets to differ from that of control packets. Therefore, the overall data packet success probability k_D is

$$k_D = P_d P_D. \tag{2}$$

The time duration of a control packet is L and that of a data packet is L_D. Note that instead of long single DATA packets, we use a batch DATA model. We assume static channel conditions and fixed FEC [13]. Let D be the maximum propagation delay, and N be the number of nodes in a collision domain (assuming no hidden nodes [11]). In a multihop scenario, N can be viewed as the number of neighboring nodes that each node effectively contends with.

We define the mean packet service time s_p as the expected delay from the time a packet is intended for transmission until it is successfully delivered, i.e., until the ACK (with retries) shows successful reception of the specific packet. We define mean batch service time s_b as the average delay from the time a batch is intended for transmission (in MACA–EA and MACA–C) until it is successfully transmitted, i.e., until the first ACK is received for the batch. The difference between the definitions of s_p and s_b is important for the queuing analysis.

An important performance metric for reliable transfer is throughput. In radio network literature this is sometimes termed as saturation throughput—the throughput of the network when the queue is saturated or always has data to transmit [9]. Such a measure is suitable for file transfer applications. This is also a measure of efficiency or channel utilization. We define normalized throughput T as the number of packets successfully transferred per unit time normalized by the system capacity, which is one packet in time L_D. For batch mode protocols such as MACA–EA and MACA–C, B packets are sent as a batch in time s_b by definition, and of these, on an average only $k_D B$ succeed due to decoding and detection losses. Thus, the normalized throughput T per node is

$$T = \frac{k_D B / s_b}{1/L_D}. \tag{3}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

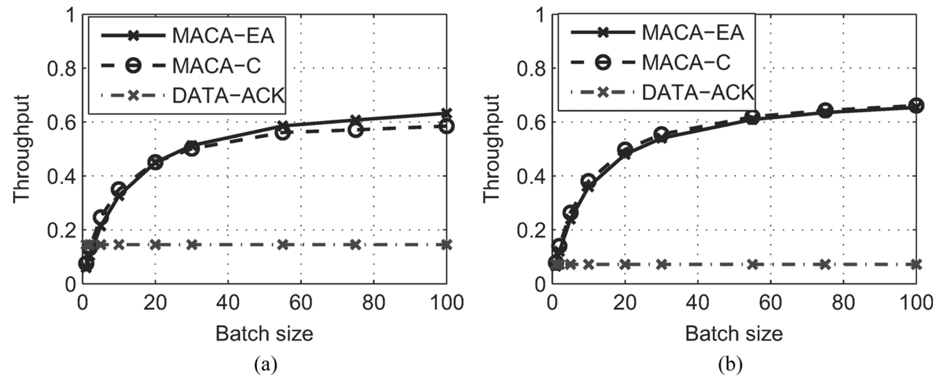


Fig. 4. Network throughput of MACA–EA, MACA–C, and DATA–ACK (packet duration L =0.5 s, L_D = 1.0 s, detection and decoding probability $k = 0.81, k_D = 0.72$, one-way propagation delay $D = 0.5$ s, number of nodes $N = 7$, contention window $W = 17$, multi-ACK $i = 3$). (a) Simulation. (b) Analysis.

Due to the retry process, it takes on an average $1/k_D$ batch transmissions for a particular packet to be delivered. Hence, as shown in [13]

$$s_p = s_b/k_D. \tag{4}$$

Therefore, we can rewrite $T = (B/s_p)/(1/L_D)$. For the DATA–ACK protocol, $B = 1$, and, therefore

$$T = \frac{1/s_p}{1/L}. \tag{5}$$

Note that for the DATA–ACK protocol, L_D does not appear in equations, as DATA packets are also of the same duration as control packets ($L_D = L$). If we characterize the service behavior with mean service times s_p and s_b and the service time cumulative distribution function (CDF), other queuing metrics such as waiting time and queue length under nonsaturated conditions can be derived using queuing analysis. The total waiting time W_T includes the waiting time in the queue W_Q and the mean service time s_p per input packet, i.e., $W_T = W_Q + s_p$.

IV. Throughput Analysis of Level-2 MAC

The throughput performance of DATA–ACK, MACA–EA, and MACA–C for saturated traffic with respect to system and environment parameters (such as batch size B for MACA–EA and MACA–C) are presented in this section. Simulations and mathematical analysis are undertaken for all the three modes. Although the analysis uses some simplifying assumptions, the analysis and simulation results match reasonably well. Therefore, the mathematical analysis can be used for comparative studies in the future. Impact of parameter variations other that those presented in this paper can also easily be studied using the analytical models derived here.

The simulator is based on Omnet++ [14], an established discrete event simulation system. The simulator accurately models propagation delays and a half-duplex physical (PHY) layer with packet loss due to collision, decoding and detection errors. By abstracting out details of modulation, error correction coding, and channel characteristics, the simulator provides a general framework that can be used in a wide variety of underwater scenarios. As demonstrated in [15], the MAC layer behavior can be simulated well using this model, by choosing appropriate PHY parameters in the simulation. In all simulations presented in this paper, the nodes are randomly spread in a 2-D area whose dimensions are chosen to match the required maximum propagation delay D. The arrival model is chosen to be saturated or Poisson depending on the scenario being tested. An overview of the simulation framework can be found in [15]. However, note that the simulator has since been enhanced, such as the ability to use different durations and loss probability for DATA and control packets, more abstracted out method for physical-layer and channel simulations (uses defined decoding and detection probabilities directly without using channel loss model).

A. DATA–ACK

The analysis used here follows a similar approach as the MACA–EA analysis in [12] and [13]. Note that this mode has queueing of incoming data instead of discarding packets if the server is in the process of sending DATA or waiting for an ACK. The standard ALOHA-based analysis for such protocols does not model queueing, uses a model where DATA is transmitted as soon as it arrives, and does not model the ACK from the receiver. There is also no retry at the MAC level for lost DATA in such models. The model here is for reliable transmission, and, therefore, no DATA packets are discarded. The protocol can be represented using the model in Fig. 3.

The details of the analysis are given in Appendix A. The important result that we use here is that the service time

$$s_p = \left(\frac{lW' + t_A}{k^2}\right)\left(\frac{W'}{W'-1}\right)^{(N-1)} + \frac{t_A(N-1)}{k} \tag{6}$$

where $l = L + D, W' = (W+1)/2, W$ is the contention window, and t_A is the timer for ACK (or CTS) reception. The throughput can be computed using (5) and multiplied by N to yield network throughput. This analytical result matches simulations reasonably well, as shown in Fig. 4. Although there is no batch size in the DATA–ACK protocol, the same throughput is shown across all values of B for ease of representation.

B. MACA–EA

Mathematical analysis and simulation results for throughput for MACA–EA protocol are presented in [12] and [13]. The batch service time is [13]

$$s_b = (1 + (1-k)^i)V + Nt_B \tag{7}$$

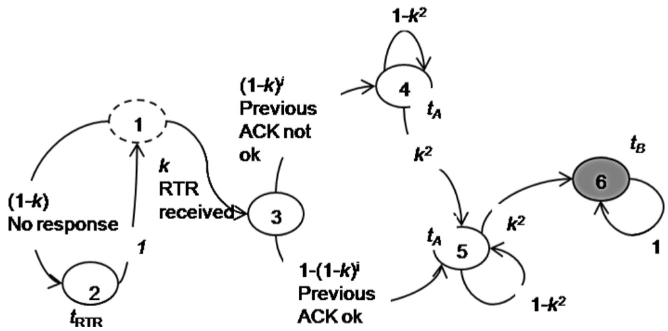


Fig. 5. Markov chain for the MACA–C protocol.

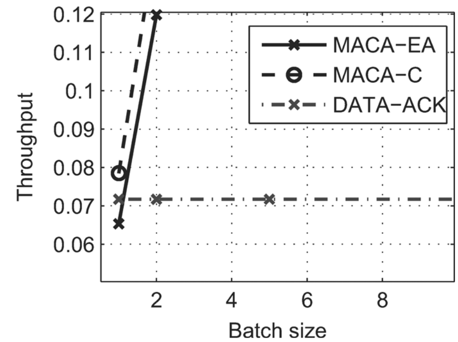


Fig. 6. Network throughput of MACA–EA, MACA–C, and DATA–ACK: behavior at low batch size, showing that DATA–ACK is better than MACA–EA protocol with $B = 1$ (simulation). Parameters: $L = 0.5$ s, $L_D = 1.0$ s, $k = 0.81, k_D = 0.72, D = 0.5$ s, $N = 7, W = 17, i = 3$.

where V is

$$V = \frac{l}{k^2 W'} \left(\frac{W'}{W'-1} \right)^N (W'^2 + W' - 2) + \frac{2(N-1)l}{k} \tag{8}$$

and

$$t_B = BL_D + t_A + (i-1)L. \tag{9}$$

The throughput can be computed using (3), and multiplied by N for network throughput. Simulation results shown in Fig. 4 show how the performance of MACA–EA improves with packet batch length. For high batch sizes, the reliable delivery throughput approaches k_D [13].

Apart from batch size, the number of ACKs i can also be optimized. The optimum number of ACKs i is given by [13]

$$i = \left\lceil \left(\log_e \frac{-Z}{\log_e(q)V} \right) \Big/ (\log_e(q)) \right\rceil .$$

This can be obtained by rewriting (7) as $s_b = (1+(1-k)^i)V + Y' + (i-1)Z = q^i V + Y + (i-1)Z$, where $q = (1-k) < 1$, V is from (8), $Y' = N(BL_D + t_A)$, $Z = NL$, and $Y = V + Y'$.

C. MACA–C

Next, we present the performance of the distributed data transfer topology RTR–RTS–CTS–DATA–ACK mode of the MACA–C protocol, for a fair comparison with MACA–EA and the DATA–ACK protocols. The MC can be considered to be acting as the channel access arbitrator for N nodes, which need to communicate to any other node.

The MACA–C protocol can be analyzed as shown in Fig. 5. The process starts with the MC sending an RTR in state s1. If RTR is received successfully, it goes to state s3, else to state s2. In s2, the response timeout $t_{\rm RTR}$ needs to be set such that it allows sufficient time to detect post-RTR packets such as RTS, CTS, or DATA to ensure that the RTR was received, and at the same time not too long so as not to waste bandwidth when the RTR is lost. The analysis and simulation used $t_{\rm RTR} = 2L + BL_D + 2D$ seconds or $2L + 12L_D + 2D$ seconds, whichever is lower. This allows for the packets in a batch, the RTS, the CTS, and the propagation delay, capped at $B = 12$. If RTR is received, the node will proceed from s3 to an early-ACK mode in state s4 [12] if the previous ACK was not received or directly to state s5 otherwise. In s4, if the RTS and early-ACK is received with a combined probability of k^2, it proceeds to state s5. In s5, if RTS is received by recipient and CTS by the sender with a combined probability of k^2, it proceeds to send DATA and wait for multi-ACK in state s6. This sequence repeats for N nodes. From the Markov chain, it can be ascertained that the expected service time to send a batch is

$$s_b = N \left(l + \left(\frac{1}{k} - 1 \right) t_{\rm RTR} + (1-k)^i \frac{1}{k^2} t_A + \frac{1}{k^2} t_A + t_B \right). \tag{10}$$

In (10), the first l (slot length $l = L + D$) is for the RTR packet itself. $((1/k) - 1)t_{\rm RTR}$ is the time to go from state s1 to s3, since expected passes through s2 are $(1/k - 1)$ (expected passes through s1 are $1/k$, since probability s1 $\rightarrow$ s3 is k). $(1-k)^i (1/k)^2 t_A$ is the expected time from s3 to s5 (path s3 $\rightarrow$ s4 $\rightarrow$ s5 only, since s4 $\rightarrow$ s5 has no delay), $(1-k)^i (1/k)^2 t_A$ is the delay in the RTS/CTS process in s5, and finally the batch transmission delay in s6 is t_B.

Note that the batch service time s_b is defined as the time from the start of the sending process until reception of the ACK. In the model used here, if the ACK after the batch transmission is not received (assumed to be sent by receiver always with probability 1) [12], the ACK may be received as an early-ACK in the next cycle in state s4 as shown. Even so, the result in (10) holds. The throughput can be computed using (3) and multiplied by N for network throughput.

Simulation and results are shown in Fig. 4. They show that MACA–C performance is nearly the same as that of MACA–EA. The MACA–C protocol eliminates collisions, but there are still overheads in arbitrating through an MC, and the throughput is restricted. Another cause of the similarity between MACA–C and MACA–EA throughput performance is that for higher batch sizes, the performance is dominated by the batch sending, and the contention period in MACA–EA or the RTR process in MACA–C becomes a less significant factor in the overall service time. Thus, one would expect the greatest percentage difference between MACA–C and MACA–EA at low batch sizes, especially $B = 1$. MACA–C service time is clearly lower for $B = 1$ or similar low batch sizes. However, as elaborated further in the following discussion, there is little utility in using MACA–C or MACA–EA in very low batch sizes as the simpler DATA–ACK mode gives good performance. It can be noted that the DATA–ACK mode is better than MACA–EA when $B = 1$, as shown in Fig. 6.

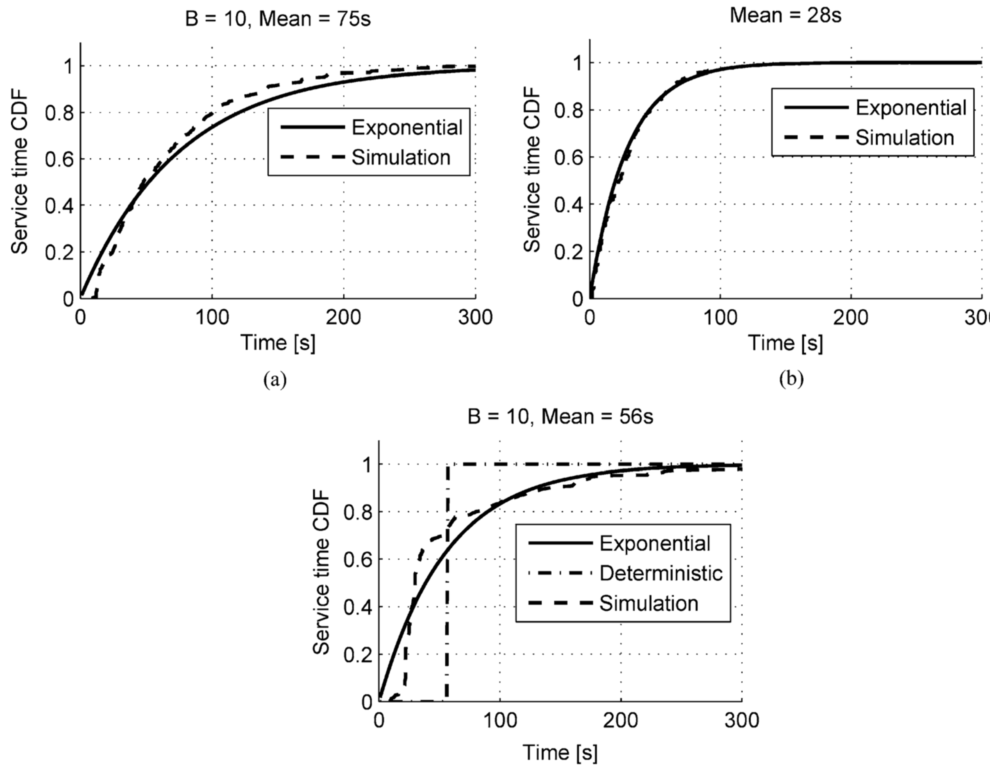


Fig. 7. Service time distributions of (a) MACA–EA, (b) DATA–ACK, and (c) MACA–C. Parameters: $L = L_D = 0.5$ s, $N = 4$, $D = 0.5$ s, $i = 3$ (not applicable to DATA–ACK), $k = k_D = 0.81$.

V. Mode Adaptation Based on Traffic Intensity

In this section, we look at unsaturated traffic scenario where packet arrivals are modeled as Poisson distributed. The choice of protocol to use is also related to the traffic intensity $\rho = \lambda/\mu$ where λ is the arrival rate (let $\delta = 1/\lambda$ be the arrival delay) and μ is the average service rate of the MAC protocol. In general, $\lambda < \mu$ for a stable system. Each protocol variant has an upper limit to the service rate and, hence, a maximum permissible λ. Waiting time in the system is used as the metric for comparing the performance of the different modes. Here, a reference TDMA protocol (referred to as TDMA–REF) is used for comparison with the three modes of MAC–AMM and to provide an upper bound for performance (lowest possible waiting time). Apart from mode switching based on traffic intensity, batch size B also needs to be automatically adapted for both MACA–C and MACA–EA. For the results presented in this section, all arrival delays are given per node. The average network delay is a fraction $1/N$ of this.

A. Service Time Distribution

To analyze the waiting time, the service time distributions of the modes have to be characterized. The service time distribution of the MACA–EA protocol has been shown to be near-exponential [13]. A plot of the MACA–EA service time distribution for $N = 4$ is shown in Fig. 7(a), and it shows the analytical exponential fit. The service time distribution of the DATA–ACK protocol was obtained via simulations and was found to be exponential, as shown in Fig. 7(b). The service time behavior of the MACA–C protocol was also obtained via simulations and is plotted in Fig. 7(c). The exponential fit for the MACA–C service time is poorer. Other typically used distributions do not give a good fit either. In our waiting time analysis that follows, we consider the exponential and deterministic service time distributions, as the actual distribution lies somewhere in between. We will see that exponential approximation yields conservative results while the deterministic service model offers optimistic results. In the waiting time queueing analysis for the MACA–EA and DATA–ACK modes, we use a Markov model as an approximation and find that it yields a good match with simulation results. Note that the mean of the DATA–ACK protocol is for the service time to send one packet, while that for the MACA–EA or the MACA–C protocol is for a batch of packets B as indicated. TDMA–REF service time is nearly deterministic. Normal TDMA is deterministic, but the retry-based reliability process introduces a small degree of nondeterminism, as discussed later in this section. Batch service time is defined as time until ACK is received, and, hence, retries have to be included in its computation.

B. MACA–EA

Since the service time distribution is nearly exponential, an $\mathrm{M/M}^B/1$ model is used. The waiting time analysis of the MACA–EA mode uses the analysis model shown in Appendix B [13]. Simulation results for waiting time are shown in Fig. 8(a) and (c). The analytical results are shown in Fig. 8(b) and (d).

C. MACA–C

To analyze MACA–C, we consider the $\mathrm{M/M}^B/1$ and $\mathrm{M/D}^B/1$ models as approximations. The $\mathrm{M/M}^B/1$ analysis

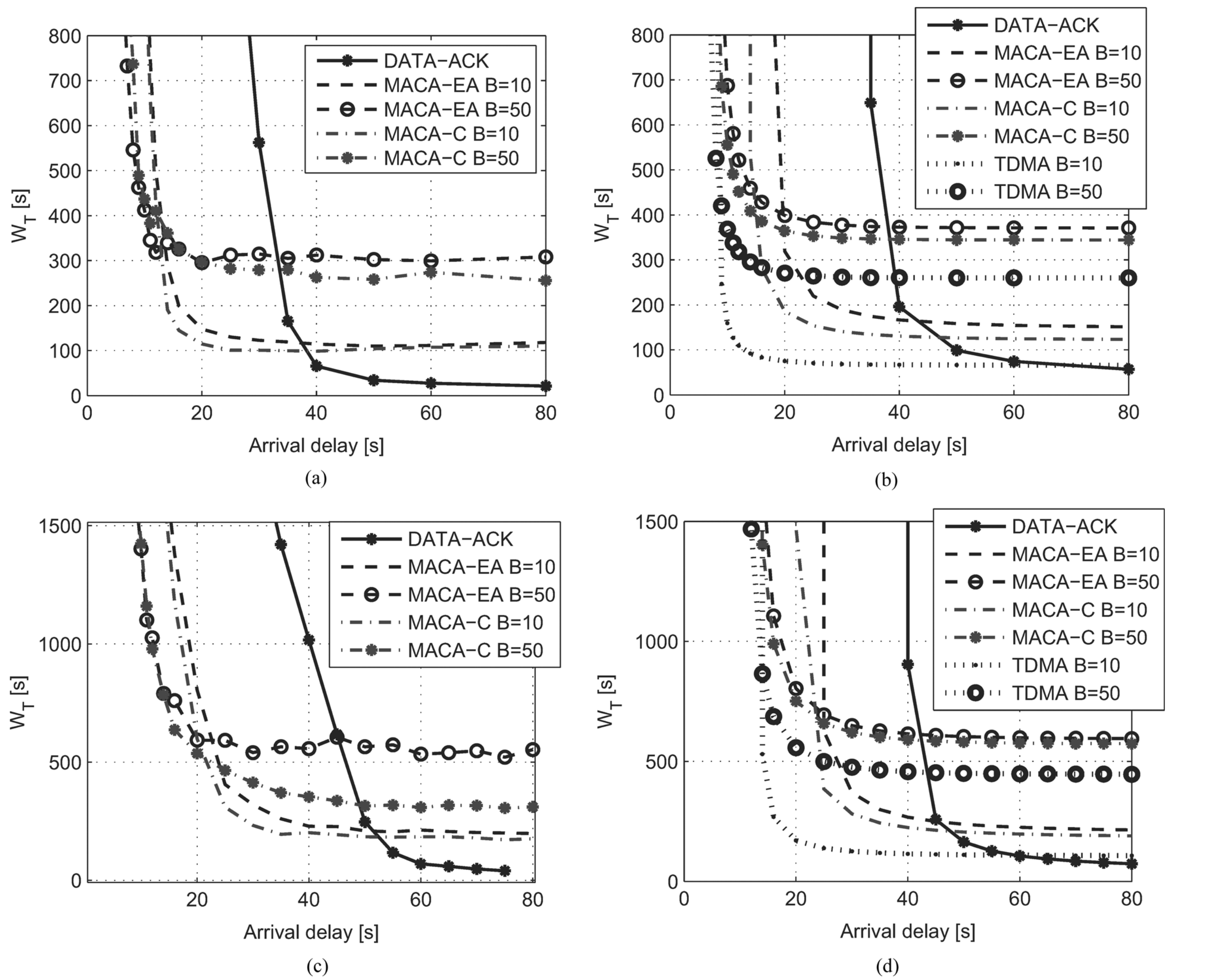


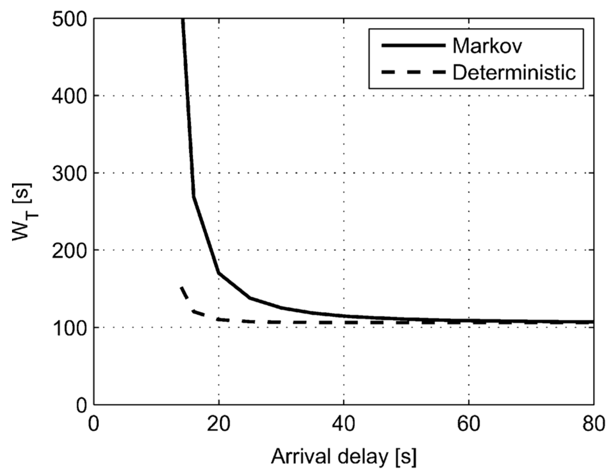
Fig. 8. Waiting time for the different modes. (MACA–C uses exponential service). Parameters: $L = 0.5$ s, $L_D = 1.0$ s, $i = 3, k = 0.81, k_D = 0.72$. (a) Simulation $N = 4, W = 11, D = 1.0$ s. (b) Analysis $N = 4, W = 11, D = 1.0$ s. (c) Simulation $N = 7, W = 17, D = 0.5$ s. (d) Analysis $N = 7, W = 17, D = 0.5$ s.

is presented in Appendix B while the $\mathrm{M/D}^B/1$ analysis is in Appendix C. Simulation results are shown in Fig. 8(a) and (c). The corresponding analytical results using the $\mathrm{M/M}^B/1$ model are shown in Figs. 8(b) and (d). The overall trends of the results compare quite well, though the analysis gives higher floor values (for higher delays) at $B = 50$ (large batch size). When arrival delay is high, there is a chance that upon successful completion of RTS–CTS, there is no DATA to send. In the analysis, when RTR succeeds, the subsequent RTS–CTS–DATA process is assumed to take place. In simulation, it is possible that both RTS and CTS were missed by the MC, and thereafter when no DATA packets are detected (DATA-based VCS), the MC will resend the RTR to the next node, and this node may have DATA. This will help reduce the overall waiting time. Such intricate behavior is not captured in the simplified analysis model and makes it give conservative estimates.

To assess how the queueing performance for MACA–C varies with different service time distribution models, we compare the Markov model $\mathrm{M/M}^B/1$ with the deterministic $\mathrm{M/D}^B/1$ model in Fig. 9(b). It shows that the $\mathrm{M/M}^B/1$ analysis gives a higher waiting time estimate.

It can be seen that the waiting time performance of this implementation of MACA–C is comparable to that of MACA–EA (except for larger number of nodes and at large batch sizes). The case of saturated throughput earlier also presented comparable performance. On the whole, for an equivalent distributed topology operation, MACA–C may offer marginal advantage compared to MACA–EA. However, MACA–C may bring greater advantage to cases where the uplink mode RTR–DATA–ACK and the downlink mode RTS–CTS–DATA–ACK can offer better performance in a star topology communication between clients and MC. The choice of MACA–C over MACA–EA is thus primarily dependent on the traffic pattern demand of the application. MAC–AMM easily allows the switch between the centralized and distributed modes, by introducing an MC in the network neighborhood. All the nodes can automatically switch modes based on RTR detection, as discussed in Section II-F.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



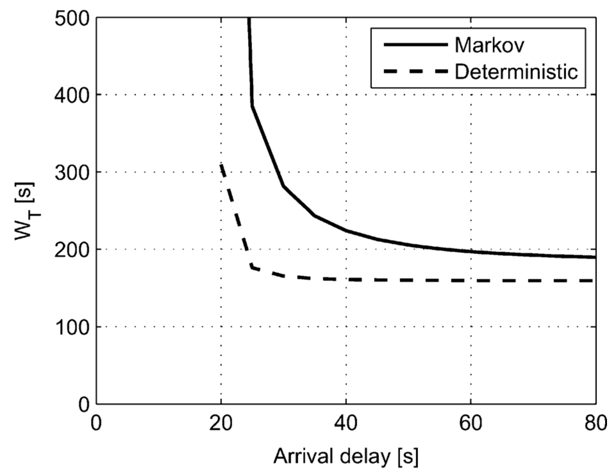


Fig. 9. Comparison of deterministic and Markov models for (a) TDMA–REF and (b) MACA–C analysis. Parameters: $B = 10, L = 0.5$ s, $L_D = 1.0$ s, $N = 7, D = 0.5$ s, $W = 17, i = 3, k = 0.81, k_D = 0.72$.

D. DATA–ACK

For the DATA–ACK mode, an $\mathrm{M/M/1}$ model is used. Using $\mu = 1/s_p$, the waiting time is [16]

$$W_T = \frac{1}{\mu - \lambda}. \tag{11}$$

Simulation results are shown in Fig. 8(a) and (c), and they match reasonably well with the analytical results shown in Fig. 8(b) and (d), and yield reasonably correct relative performance comparison with the other modes. We do note that the rapid rise below 40 s (this is close to the instability region) in the analysis is different from simulations, but the overall performance is reasonably well predicted by analysis. It can be seen that, at high arrival delay, it can outperform MACA–EA, MACA–C, and even TDMA–REF, depending on the batch sizes chosen.

E. TDMA–REF

Here we try to benchmark the three modes of MAC–AMM with a reference TDMA-based protocol named TDMA–REF. The TDMA–REF protocol has no contention process. Just as in any static TDMA-based protocol model, transmission frames are statically assigned to all nodes in a given sequence. In each frame, the assigned node transmits a batch of packets. To have a comparable scheme that allows for retry-based reliability, ACK from the receiver is sent within each frame. To enhance performance, the multi-ACK technique from MACA–EA is used.

Accounting for the two-way propagation delay, the service time for N nodes is

$$s_b = \frac{1}{1 - (1 - k)^i} N(BL_D + D + 3L + D) \tag{12}$$

where the factor $1/(1 - (1 - k)^i)$ accounts for the expected retries when the ACK is lost. If the ACK is lost and the whole batch is retried, the packet service time s_p is

$$s_p = 1/(1 - (1 - k_D)^{(1/k_D)}) s_b. \tag{13}$$

The reason for resending an entire batch when the ACK is lost is that in such a TDMA-based protocol, there is no RTS/CTS contention process and the associated early-ACK provision to resend the ACK. Hence, each node has no option but to retransmit the entire batch in the next frame when an ACK is not received.

Since the whole batch is retransmitted each time an ACK fails, the receiver can receive a particular packet in any one of the batches. Thus, the factor $1 - (1 - k_D)^{(1/k_D)}$ accounts for this success probability and the consequent retries. The TDMA–REF protocol used here is sophisticated enough to provide a good benchmark.

Analysis for the general service model $\mathrm{M/G/1}$ or $\mathrm{M/G}^B\mathrm{/1}$ shows that waiting time measures are related primarily to the mean and variance of the service time distribution [16]. Since the variance is zero for the deterministic service model, it provides the best case queueing performance for a given mean. Thus, we can use an $\mathrm{M/D}^B\mathrm{/1}$ model for the TDMA–REF protocol to provide a benchmark, although the possibility of retries when ACK is lost causes the batch service time to not be strictly deterministic.

The $\mathrm{M/D}^B\mathrm{/1}$ analysis is described in Appendix C. The results are shown in Fig. 8(b) and (d).

To assess how the queueing performance for TDMA–REF varies with difference service time distribution models, we compare the Markov model $\mathrm{M/M}^B\mathrm{/1}$ with the deterministic $\mathrm{M/D}^B\mathrm{/1}$ model in Fig. 9(a). The analysis method for $\mathrm{M/M}^B\mathrm{/1}$ queueing model for TDMA–REF is the same as the MACA–EA analysis (shown in Appendix B). The results are plotted in Fig. 9(a). It shows that the Markov model produces higher waiting time at arrival delays close to saturation as expected and illustrates how the deterministic case produces the best case for TDMA–REF, thus giving us a good benchmark to compare against.

We can see from Fig. 8(b) and (d) that TDMA–REF outperforms other protocols for the same batch size. Using the analysis models presented here we can assess the performance cost in not using TDMA–REF. However, the choice of TDMA–REF versus other protocols depends on requirements of *ad hoc* functionality, scalability, and robustness (especially to time synchronization).

F. Effect of Traffic Intensity

There are two key points related to traffic intensity. The first is that a given mode has a maximum service rate beyond which it will have rapidly increasing waiting time. The second is that each protocol has a lower bound for waiting time as δ increases, for a fixed set of environment and protocol parameters.

There is another perspective to the results. We can define the quality-of-service (QoS) requirement as a maximum allowable waiting time for a given arrival delay. It is possible that different modes can satisfy the QoS requirement. For example, if we say that for an average arrival delay of 40 s, we require the waiting time to be no more than 100 s for reliable transfer, Fig. 8(b) shows that DATA–ACK, TDMA–REF, MACA–EA, and MACA–C all can satisfy it with a suitable batch size. When arrival delays become lower, the lower performing protocols will no longer be able to satisfy the QoS requirement. For example, at an average arrival delay of 20 s with a 100-s maximum waiting time, the DATA–ACK mode can no longer be used (for this particular set of parameters). This perspective is important in assessing the relative merits.

Except for high arrival delay, TDMA–REF can provide the lowest waiting time for a given batch size as seen. It can also support the lowest arrival delay (or the highest arrival rate). However, it should be noted that, in TDMA–REF, the batch size B has to be fixed at deployment and it needs to be tuned to a certain minimum arrival delay. Higher B will give better throughput efficiency, and support a lower delay, but also suffer increased waiting time. However, in TDMA–REF, it is not easily possible to adaptively tune the batch size, especially if required frequently. Neither MACA–C nor MACA–EA presents this drawback, and can vary the batch size at every transmission as required. Also, dynamic node joining and departures (*ad hoc* capability) cannot easily be supported in TDMA–REF. It is also critically dependent on time synchronization, which may not always be available in a general UAN scenario.

MACA–C has both *ad hoc* capability and the ability to adaptively vary B to suit arrival delays. Since it can support additional uplink and downlink modes also, it may be a good choice in some UAN scenarios. However, MACA–C will have to contend with scalability issues and interference between neighboring cells for multihop networks [3]. If it is a small network in isolation, such issues may not be present and MACA–C will be a good choice. Without elaborating on the scenarios where such protocols are useful, the purpose of the analysis in this paper is to understand their performance limitations and, most importantly, the transition traffic intensity to change to and from the DATA–ACK protocol mode.

Comparing MACA–EA and DATA–ACK, the latter is best in the high arrival delay region. Beyond a certain arrival delay, DATA–ACK will have a lower waiting time. But at lower arrival delay, DATA–ACK cannot support the traffic intensity and only MACA–EA can provide stable service and a reasonable waiting time. It should be noted that the possibility of RTS/CTS-based protocols and random access protocols being dynamically adapted based on traffic intensity had been previously discussed [17]. However, the paper does not contain a detailed simulation and mathematical analysis as we have presented here, using metrics such as normalized throughput or waiting time for reliable transfer of DATA.

This crossover arrival delay ($\delta = 1/\lambda$) can be judged as follows. It has been shown that for $1/\lambda > (k_D(\alpha+\beta)^2 + \beta(\alpha+\beta))/\beta$, $B = 1$ is the optimum batch size [13], where α and β are as follows:

$$\alpha = (1+(1-k)^i)\gamma + N(t_A + (i-1)l), \qquad \beta = NL_D. \quad (14)$$

Since at $B = 1$, DATA–ACK protocol performs better than MACA–EA, we can use $1/\lambda$ as the crossover arrival delay. In other words, for

$$\delta > \frac{k_D(\alpha+\beta)^2 + \beta(\alpha+\beta)}{\beta} \quad (15)$$

the protocol suite must switch to the DATA–ACK mode. Another key factor is batch size B adaptation with respect to δ when MACA–EA is in operation. Optimum batch size has been estimated for MACA–EA as follows [13]:

$$\overline{B_{\text{opt}}} = 1 + \frac{k_D\lambda(\alpha+\beta)^2}{2\beta(1-\lambda(\alpha+\beta))}. \quad (16)$$

Higher batch size is suited for lower δ to ensure stability and to minimize waiting time. With such a combined batch size and protocol mode adaptation mechanism, this suite can deliver optimum performance at any traffic intensity for an *ad hoc* UAN.

G. Adaptation Algorithm

Next, we present an algorithm that can dynamically adapt the batch size of the MACA–EA protocol and enable the switch to the DATA–ACK mode based on traffic intensity. The same algorithm can easily be adapted to switch between MACA–C and DATA–ACK.

Fig. 10 shows how DATA–ACK compares with MACA–EA in terms of waiting time for a range of batch sizes. It shows how between 30- and 40-s arrival delay, the DATA–ACK protocol gets better than MACA–EA in terms of waiting time, for the set of parameters used here ($N = 4, L = L_D = 0.5, D = 0.5, k = k_D = 0.81, i = 3$). Fig. 10(c) shows the approximate delay where the two match. The match between simulations and analysis for MACA–EA waiting time is also confirmed in Fig. 10. Fig. 11 shows how DATA–ACK performance compares with $B = 5$ and $B = 2$ for MACA–EA. It also shows roughly 30-s arrival delay as the transition point. Note that the small difference between the DATA–ACK modes in Figs. 10 and 11 is due to the former being obtained from the mathematical model, and the latter from simulations.

The adaptation mechanism is as follows. Use the average batch size occupancy to decide batch size, and average the batch size occupancy over multiple batch transmissions (e.g., three transmissions). If the batch occupancy is greater than 80% of

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

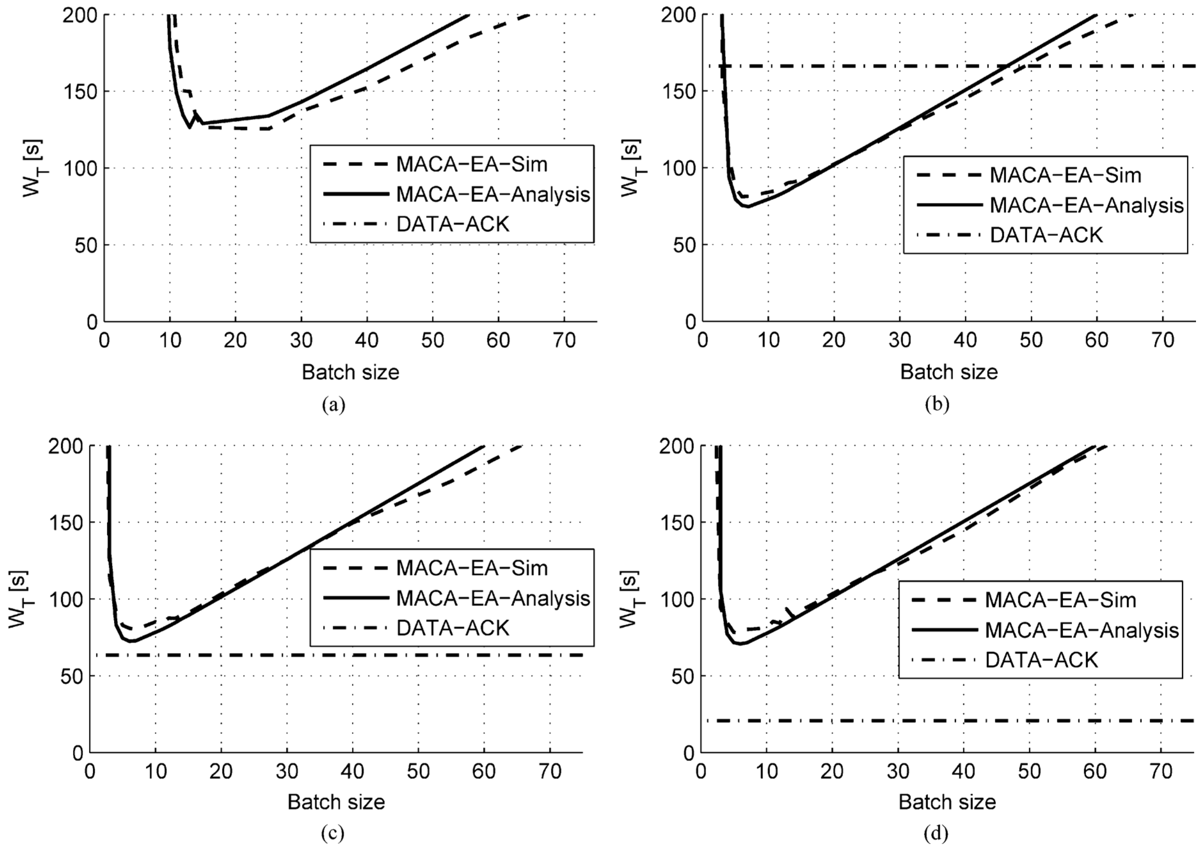


Fig. 10. Varying batch size B, DATA–ACK (simulated), and MACA–EA at a given arrival delay. Parameters: $L = L_D = 0.5$ s, $N = 4, D = 0.5$ s, $W = 11, i = 3, k = k_D = 0.81$. (a) Average arrival delay = 10 s. (b) Average arrival delay = 30 s. (c) Average arrival delay = 35 s. (d) Average arrival delay = 40 s.

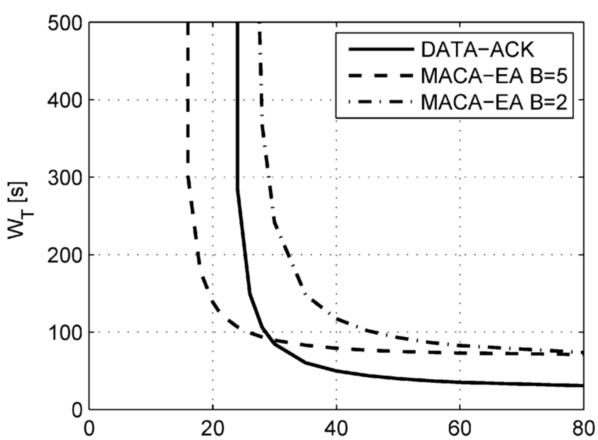


Fig. 11. Comparing $B = 5$ and $B = 2$ MACA–EA with DATA–ACK (all analytical) at different arrival delays. Parameters: $L = L_D = 0.5$ s, $N = 4, D = 0.5$ s, $W = 11, i = 3, k = k_D = 0.81$.

B, increase B by S. The step S is set to 10% of the batch size. If the batch occupancy falls below 50% of B, then decrease batch size by S. Simulations showed that this helps adapt the MACA–EA protocol batch size to its optimum value, as discussed in Section V-F. If the adapted optimum B in MACA–EA drops to less than $B = B_{\min}$ ($B_{\min} = 5$), then switch to DATA–ACK mode. In DATA–ACK mode, monitor the queue size; if queue size exceeds a certain threshold, switch back to MACA–EA with $B = B_{\min}$.

Simulation of the MACA–EA adaptation described above has been done. For an arrival delay of 10 s, it gave an adapted batch size of $B \approx 15$ with $W_T = 148$ s and for an arrival delay of 35 s, it gave an adapted batch size of $B = 5$ (which is $B_{\min}$, to switch over to DATA–ACK mode) with $W_T = 85$ s. In both cases, the simulations were started off with $B = 50$. The values are in reasonable agreement with the optimum batch size seen in Fig. 10(a) and (c), respectively (the approximate batch size at the lowest point in the curves). The switch over from MACA–EA to DATA–ACK should be done once adapted batch size reaches $B_{\min}$, as mentioned above. Some of the parameters of the above algorithm are currently heuristics and require further analytical investigation.

If data arrive in a burst in distributed mode, MACA–EA should be used. If the system is in DATA–ACK mode at the time, it switches to MACA–EA mode. If operating in centralized mode, the protocol can directly take care of burst data. The DATA–ACK mode is suitable for situations with low sporadic load. The case of low load could arise in many situations such as underwater sensor network deployments which need to report sporadic information. The DATA–ACK mode gives the least possible latency in this case and makes information and consequent potential actions more timely at the receiver.

VI. Discussion and Future Work

We discuss a few important points regarding the use and behavior of MAC–AMM. If data arrive in a burst in distributed mode, MACA–EA should be used. If the system is in DATA–ACK mode at the time, it switches to MACA–EA mode. If operating in centralized mode, the protocol handles burst data well. Unreliable messaging and broadcasts can be easily achieved in MAC–AMM. In the level-1 MAC, the random backoff method applies to all packets equally, including unreliable messages and broadcasts. In level-2 communications mode, if unreliable short messaging (no ACKs) is required, the same contention logic as RTS can be used to send single short DATA packets using control packet FEC. This is essentially the DATA–ACK protocol without the ACK. In MACA–C uplink, RTR–DATA format can be used for unreliable short messaging (no ACKs). Broadcasts are achieved via the use of these unreliable modes. Unreliable broadcast mode can be used for beacons such as those proposed in JANUS. JANUS beacon and similar concepts attempt to allow nodes to broadcast useful information about itself to neighboring nodes. Such broadcast packets should also be transmitted under the control of a MAC protocol to avoid interference in a given acoustic frequency band. Rather than using batches of short DATA packets, it is possible to use long DATA packets. The tradeoffs in this approach are discussed in [13].

FEC and power control have a significant role in achieving optimum performance in UAN MAC protocols. RTS/CTS exchanges can carry information to do both dynamic FEC and power control, and this is an important advantage offered by the protocols adapted from MACA, which may more than compensate for the loss in round trip delay in the handshake. The decoding of control packet RTS could help estimate the BER and the CTS can specify the FEC scheme, power, and other parameters to be used for batch DATA. We note that, as a simplifying assumption, bidirectional validity of power information (or, equivalently, the same success probability) was used in the simulations. Power control, to a certain extend, may be able to handle bidirectional inequalities in sea trials and help achieve similar success probabilities for packets in both directions. These aspects are to be further studied as part of future work.

The protocol overhead is captured in the normalized throughput metric. Based on (3), we can say that T/k_D is the efficiency of the protocol, since k_D is the saturation throughput (mentioned in Section IV-B). $1 - T/k_D$ thus gives the protocol overhead, which becomes negligible as B gets large. The loss due to propagation delay is also captured within this, but this is incurred through the RTS–CTS process and, hence, associated directly with the protocol overhead.

A. Multihop

The work here is primarily aimed at single-hop networks. Multihop extensions should be done as part of future work. A brief discussion on possible methodology for multihop analysis is outlined in [13]. As mentioned in Section III, in a multihop scenario, N can be viewed as the number of neighboring nodes that each node effectively contends with. As mentioned in Section V-F, when using MACA–C in multihop networks, inter-MC interference may be experienced, if there are multiple cells in the neighborhood controlled by different MCs. Mechanisms will be required to address this, and it was briefly discussed in [3]. This is a general challenge for the centralized topology, as discussed in [8].

VII. Conclusion

In this paper, we presented a comprehensive MAC protocol suite to address a diverse and heterogeneous underwater network with multiple levels of compliance which also allows for proprietary protocols to be used in isolation. The MAC protocol in level-2 operations (communications among heterogeneous assets) of the suite has both distributed and centralized operating modes. A novel adaptation scheme chooses between the modes based on deployment, environment, and system parameters. Analytical throughput and waiting time performance results were shown which have been verified through simulations. Traffic intensity was shown to be a key parameter, especially to determine the transition to the low traffic DATA–ACK mode. An algorithm to dynamically adapt the batch size in MACA–C and MACA–EA, as well as to enable the automatic switch to the DATA–ACK mode was presented. The key vision is a self-organizing network, with nodes able to dynamically adapt to any scenario through environment discovery. A key utility of this work is the reasonably accurate analytical characterization of the relative performance of the different modes in this protocol suite. Users planning to deploy such networks can also understand the performance limitations and ensure that communication requirements are within feasible limits.

Appendix A
DATA–ACK Throughput Analysis

The transmitter sends DATA and the receiver sends back ACK. The contention algorithm for sending DATA packets is the same as MACA–EA RTS sending. A node starts with a uniform probability distributed backoff in a contention window W. When the backoff timer expires, a DATA is sent. When DATA transmission begins, ACK timer t_A starts. (Note that here the DATA–ACK process is analogous to the RTS–CTS process for MACA–EA, described in Section II-C, and the DATA packets are essentially control packets with some user data payload.) The timer used to wait for ACK (t_A) (as well as CTS in MACA–EA protocol mode) is related to D and control packet time duration L to give enough time for the round trip delay as

$$t_A = 2D + 2L. \tag{17}$$

If the timer expires before reception of ACK, DATA backoff procedure starts again. If ACK is not received, the cycle repeats. Reception of DATA–ACK packets while waiting to send DATA triggers VCS. Note that 802.11 uses freezing backoff [9], whereas a constant window is used here. All nodes use the same contention window W at any given time.

In line with the definition used by [18], during DATA contention phase, the slot duration l is defined as

$$l = L + D. \tag{18}$$

The following analysis is adapted from the Markov analysis for RTS contention in [12]. A node starts with a uniformly selected backoff time slot in the integer range $[1, W]$. The actual contention window time period is Wl. For simplicity of analysis, it is assumed that no collisions happen during the ACK period, assuming VCS starts due to DATA reception (results showed that this simplification did not have significant impact on the analytical predictions). So, in our analysis model, ACK loss will only be due to decoding and packet detection probability. If the transmitter does not get an ACK, it restarts the contention window for DATA. Any other node which had received the DATA does a VCS for an ACK. It resets and restarts contention if an ACK does not arrive. Thus, until one node gets an ACK, this process will continue.

The protocol can be represented using the model in Fig. 3. Circles with enclosed numbers are states. Transition probabilities are shown along the arrows. The duration spent in state 1 is 1, and for others states, it is t_A as indicated. In the analysis, state transitions will be represented as a pair such as (g, h) for a transition from state g to h. State transition probability will be represented as $P(g, h)$.

The start of the DATA contention cycle is at state 1. The probability of a node sending a DATA at the start of a new slot is modeled as $P(1, 2) = a = 2/(W + 1)$. This is because the expected value of the uniformly distributed contention window is $W' = (W + 1)/2$, and that is used as the expected value of a geometric process for transition (1, 2) to satisfy Markov chain requirements. Once a DATA is sent, the node is in state 2, waiting for t_A time slots for the ACK to arrive. If the ACK arrives, the process terminates.

The probability that the DATA transmitted in a given slot has no collision from any other node is $(1 - 1/W')^{N-1}$, i.e., no other node transmits a DATA in that slot. The ACK will be successfully received if, apart from having no collisions, DATA is received at the receiver (probability k) and the ACK, in turn, is received at the transmitter (probability k) with a combined probability of k^2. This is shown in Fig. 3 as $P(2, 4) = f = k^2(1 - (1/W'))^{N-1}$. If the ACK is not successfully received, transition (2, 1) happens as shown with probability $z = 1 - f$.

If DATA is not sent (probability $1 - 1/W'$), the current node counts down the DATA timer by one slot. During this backoff period, the probability that one of the $N - 1$ neighbors has a successful DATA transmission is $y = (N - 1)(1/W')(1 - 1/W')^{N-2}$, using the same arguments as in the last paragraph. And k being the DATA detection probability, the current node could receive DATA from another node with probability ky. Thus, the transition (1, 3) with $P(1, 3) = b = (1 - (1/W'))ky$ occurs as shown.

In state 3, it awaits an ACK for time t_A. Thereafter, it goes back to state 1 with probability 1.

If system is backing off and either DATA or ACK from others is not received as stated above, it goes back to state 1, as shown with $P(1, 1) = c = 1 - a - b$.

A Markov matrix $\mathbf{M}$ [16] represents this as follows using $P(a, b)$, as shown in Fig. 3:

$$\mathbf{M} = \begin{bmatrix} c & a & b & 0 \\ z & 0 & 0 & f \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} c & a & b \\ z & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{19}$$

$\mathbf{Q}$ is the transient state matrix

The fundamental matrix $\mathbf{F}$ [16] is $\mathbf{F} = (\mathbf{I} - \mathbf{Q})^{-1}$. Let $E(m, n)$ be the expected number of times the system is in state n after starting from state m. $E(1, n)$ is the expected number times the state n will be visited if the chain starts in state 1. Using the standard Markov chain theory [16]

$$E(1, n) = F_{1,n}, \quad E(1, 1) = \frac{1}{k^2} \frac{W'}{\left(\frac{W'-1}{W'}\right)^{N-1}}$$
$$E(1, 2) = \frac{1}{k^2} \frac{1}{\left(\frac{W'-1}{W'}\right)^{N-1}}, \quad E(1, 3) = \frac{N-1}{k}. \tag{20}$$

Let the time until successful reception of ACK from state 1 to state 4 of Fig. 3 be s_p. This gives

$$s_p = (l)E(1, 1) + t_A E(1, 2) + t_A E(1, 3). \tag{21}$$

Using (20), s_p can be simplified as

$$s_p = \left(\frac{lW' + t_A}{k^2}\right)\left(\frac{W'}{W'-1}\right)^{(N-1)} + \frac{t_A(N-1)}{k}. \tag{22}$$

Appendix B
M/M^B/1 Waiting Time Analysis for MACA–EA and MACA–C

The probability generating function of the steady-state probability distribution $P(z)$ for M/M^B/1 (can be inferred from results in [16]) is $P(z) = (1 - r_0)/(1 - r_0 z)$. Service rate μ_b is defined as $1/s_b$. The constant r_0 is the root of the characteristic equation of M/M^B/1 [16], as shown in the following (Δ is the shift operator):

$$\left(\mu_b \Delta^{B+1} - (\lambda + \mu_b)\Delta + \lambda\right) p_n = 0, \qquad n \geq 0. \tag{23}$$

The root r_o needs to be found numerically, except for small values of B. When $B = 1$, (23) becomes quadratic. Roots of (23) can analytically be found up to $B = 4$, but expressions are cumbersome. For $B >= 5$, analytical roots are not always obtainable as is generally known in polynomial theory.

The required moments such as expected queue length and waiting time can be computed as follows. Expected system total length L_T (includes the packets in service)

$$L_T = P'(z)|_{z=1} = \frac{r_0}{1 - r_0}. \tag{24}$$

Using Little's law [16], the expected waiting time W_Q (excluding service) can be found as follows [s_b from (7)]:

$$W_Q = \frac{L_T}{\lambda} - \frac{1}{\mu_b} = \frac{r_0}{(1 - r_0)\lambda} - s_b. \tag{25}$$

The total waiting time W_T (the sum of the queuing time W_Q and the service time of one packet s_p) is then [s_p from (22)]

$$W_T = \frac{r_0}{(1 - r_0)\lambda} - s_b + s_p. \tag{26}$$

The average intake batch size is reduced to Bk, since retries will occupy $(1-k)B$ on average for every batch, and r_0 needs to be computed using this modified batch size in (23).

Appendix C
M/D^B/1 Waiting Time Analysis for MACA–C and TDMA

The characteristic equation for the M/D^B/1 system is given by [19]

$$z^B e^{B\rho(1-z)} - 1 = 0 \tag{27}$$

where $\rho = \lambda/\mu$. The roots z_i of (27) can be found through a numerical technique [20]. The total system size is [19]

$$L = \frac{1 - B(1-\rho)^2}{2(1-\rho)} + \sum_{1}^{B-1} (1 - z_i)^{-1}. \tag{28}$$

Using Little's law [16], the expected waiting time W_Q (excluding service) can be found as follows [s_b from (12) for TDMA and from (10) for MACA–C]:

$$W_Q = \frac{L_T}{\lambda} - s_b. \tag{29}$$

Total waiting time W_T (the sum of queuing time W_Q and service time of one packet s_p) [from (13) for TDMA and from (4) for MACA–C] is then

$$W_T = W_Q + s_p. \tag{30}$$

References

[1] L. Freitag and S. Singh, "Multi-user frequency hopping underwater acoustic communication protocol," Woods Hole Oceanogr. Inst. (WHOI), Woods Hole, MA, USA, Tech. Rep., 2000.

[2] K. McCoy, "JANUS: From primitive signal to orthodox networks," in *Proc. Underwater Acoust. Meas.*, Nafplion, Greece, 2009, pp. 1–8.

[3] S. Shahabudeen, M. Chitre, J. Potter, and M. Motani, "Multi-mode adaptive MAC protocol suite and standardization proposal for heterogeneous underwater acoustic networks," in *Proc. Underwater Acoust. Meas.*, Nafplion, Greece, 2009, pp. 1–8.

[4] J. Rice, B. Creber, C. Fletcher, P. Baxley, K. Rogers, K. McDonald, D. Rees, M. Wolf, S. Merriam, R. Mehio, J. Proakis, K. Scussel, D. Porta, J. Baker, J. Hardiman, and D. Green, "Evolution of seaweb underwater acoustic networking," in *Proc. MTS/IEEE OCEANS Conf.*, 2000, vol. 3, pp. 2007–2017.

[5] A. Kebkal, K. Kebkal, and M. Komar, "Data-link protocol for underwater acoustic networks," in *Proc. OCEANS Eur. Conf.*, 2005, vol. 2, pp. 1174–1180.

[6] B. Peleato and M. Stojanovic, "Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks," *IEEE Commun. Lett.*, vol. 11, no. 12, pp. 1025–1027, Dec. 2007.

[7] X. Guo, M. R. Frater, and M. J. Ryan, "Design of a propagation-delay-tolerant MAC protocol for underwater acoustic sensor networks," *IEEE J. Ocean. Eng.*, vol. 34, no. 2, pp. 170–180, Apr. 2009.

[8] S. Shahabudeen, M. Chitre, and M. Motani, "Dynamic TDMA and MACA based protocols for distributed topology underwater acoustic networks," in *Underwater Acoustic Sensor Networks*. Boca Raton, FL, USA: CRC Press, 2010, ch. 8, pp. 197–224.

[9] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[10] P. Karn, "MACA—A new channel access method for packet radio," in *Proc. ARRL/CRRL Amateur Radio 9th Comput. Netw. Conf.*, 1990, pp. 134–140.

[11] J. J. Garcia-Luna-Aceves and C. L. Fullmer, "Performance of floor acquisition multiple access in ad-hoc networks," in *Proc. 3rd IEEE Symp. Comput. Commun.*, 1998, pp. 63–68.

[12] S. Shahabudeen and M. Motani, "Performance analysis of a MACA based protocol for adhoc underwater networks: Short paper," in *Proc. 4th ACM Int. Workshop UnderWater Netw.*, Berkeley, CA, USA, Nov. 3, 2009, pp. 13:1–13:4.

[13] S. Shahabudeen, S. Motani, and M. Chitre, "Analysis of a high performance MAC protocol for underwater acoustic networks," *IEEE J. Ocean. Eng.*, 2013, DOI: 10.1109/JOE.2013.2246741.

[14] Omnet++, "Discrete event simulation system," [Online]. Available: http://www.omnetpp.org/

[15] S. Shahabudeen, M. Chitre, M. Motani, and A. Low, "Unified simulation and implementation software framework for underwater MAC protocol development," in *Proc. MTS/IEEE OCEANS Conf.*, Biloxi, MS, USA, Oct. 26–30, 2009.

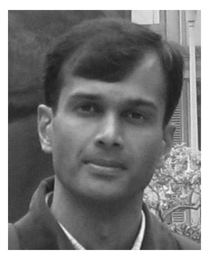
[16] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*. New York, NY, USA: Wiley, 1998, pp. 122–127.

[17] X. Peng and J. H. Cui, "Exploring random access and handshaking techniques in large-scale underwater wireless acoustic sensor networks," in *Proc. OCEANS Conf.*, 2006, DOI: 10.1109/OCEANS.2006.306945.

[18] M. Molins and M. Stojanovic, "Slotted FAMA: A MAC protocol for underwater acoustic networks," in *Proc. MTS/IEEE OCEANS Conf.*, 2006, DOI: 10.1109/OCEANSAP.2006.4393832.

[19] M. L. Chaudhry and J. G. C. Templeton, *A First Course in Bulk Queues*. New York, NY, USA: Wiley, 1984, pp. 196–204.

[20] A. Janssen and J. V. Leeuwaarden, "Analytic computation schemes for the discrete-time bulk service queue," *Queueing Syst., Theory Appl.*, vol. 50, no. 2–3, pp. 141–163, 2005.



Shiraz Shahabudeen (S'05–M'06) received the B.Eng. degree in electrical engineering from the National University of Singapore (NUS), Singapore, in 1998 and the M.S. degree in telecommunication engineering from Melbourne University, Parkville, Vic., Australia, in 2003.

He worked in the telecommunication software industry from 1998 to 2002 and at the Infocomm Development Authority of Singapore (IDA) from 2003 to 2004 as a Wireless Technology Specialist. He joined the Acoustic Research Laboratory (ARL), NUS, in 2004 and served as a Research Fellow until 2010, where his research interests included underwater acoustic communications, networking, and autonomous underwater vehicles. He currently works as a Senior Research Engineer at NeST Software, Kerala, India.

Mandar Chitre (S'04–M'04–SM'11) received the B.Eng. and M.Eng. degrees in electrical engineering from the National University of Singapore (NUS), Singapore, in 1997 and 2000, respectively, the M.Sc. degree in bioinformatics from the Nanyang Technological University (NTU), Singapore, in 2004, and the Ph.D. degree in underwater acoustic communications from NUS in 2006.

From 1997 to 1998, he worked with the Acoustic Research Laboratory (ARL), NUS. From 1998 to 2002, he headed the Technology Division of a regional telecommunications solutions company. In 2003, he rejoined ARL, initially as the Deputy Head (Research) and is now the Head of the laboratory. He also holds a joint appointment with the Department of Electrical and Computer Engineering, NUS, as an Assistant Professor. His current research interests are underwater communications, autonomous underwater vehicles, model-based signal processing, and modeling of complex dynamic systems.

Dr. Chitre has served on the technical program committees of the IEEE OCEANS Conference, the International Conference on Underwater Networks and Systems (WUWNet), and the Defense Technology Asia (DTA) Conference, and has served as reviewer for many international journals. He was the chairman of the student poster committee for the 2006 IEEE OCEANS Conference in Singapore. He is currently the Vice Chairman of the IEEE Oceanic Engineering Society (OES; Singapore chapter) and the IEEE Technology Committee Co-Chair of Underwater Communication, Navigation & Positioning.

Mehul Motani (S’92–M’00) received the B.S. degree from Cooper Union, New York, NY, USA, in 1992, the M.S. degree from Syracuse University, Syracuse, NY, USA, in 1995, and the Ph.D. degree from Cornell University, Ithaca, NY, USA, in 2000, all in electrical and computer engineering.

Currently, he is an Associate Professor at the Electrical and Computer Engineering Department, National University of Singapore (NUS), Singapore. He has held a Visiting Fellow appointment at Princeton University, Princeton, NJ, USA. Previously, he was a Research Scientist at the Institute for Infocomm Research, Singapore, for three years and a Systems Engineer at Lockheed Martin, Syracuse, NY, USA, for over four years. His research interests are in the area of wireless networks. Recently, he has been working on research problems which sit at the boundary of information theory, networking, and communications, with applications to mobile computing, underwater communications, sustainable development, and societal networks.

Dr. Motani received the Intel Foundation Fellowship for his Ph.D. research, the NUS Faculty of Engineering Innovative Teaching Award, and placement on the NUS Faculty of Engineering Teaching Honours List. He has served on the organizing committees of the International Symposium on Information Theory (ISIT), the Wireless Network Coding Conference (WiNC), and the International Conference on Computational Science (ICCS); and the technical program committees of the International Conference on Mobile Computing and Networking (MobiCom), the International Conference on Computer Communications (Infocom), the International Conference on Network Protocols (ICNP), the International Conference on Sensing, Communication, and Networking (SECON); and several other conferences. He participates actively in IEEE and the Association for Computing Machinery (ACM) and has served as the Secretary of the IEEE Information Theory Society Board of Governors. He is currently an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY and an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.