Cross Layer Power Adaptive CSMA/CA for Sink Powered Underwater Acoustic Sensor Networks

Alper Bereketli\textsuperscript{*}, Hasan Türken\textsuperscript{†}, and Semih Bilgen\textsuperscript{*}
\textsuperscript{*}Department of Electrical and Electronics Engineering
Middle East Technical University, Ankara, Turkey
\textsuperscript{†}ASELSAN Inc.
Defense Systems Technologies Division, Ankara, Turkey
Email: \{balper.bilgen\}@eee.metu.edu.tr
Email: htturken@aselsan.com.tr

Abstract—The term Sink Powered Underwater Acoustic Sensor Networks (SPUASN) refers to a special configuration within the recent Remotely Powered Underwater Acoustic Sensor Networks (RPUASN) paradigm, where the data sink supplies energy to battery-free sensors that constitute the network. In this paper, we introduce the Cross Layer Power Adaptive CSMA/CA (X-PACCA) protocol for SPUASN. Fully free from energy constraints of traditional protocols, X-PACCA integrates MAC, network, and transport layer functionalities. Packet relaying/routing is based on CSMA/CA backoff window size adjustment according to harvested power levels at nodes. End-to-end reliability is enhanced via acknowledgments sent by the sink. Congestion is avoided via prevention of redundant packet forwarding. Neither global network information nor synchronization is required at nodes. It is shown that, with appropriate selection of protocol parameters, X-PACCA achieves low end-to-end latency and high packet delivery performance.

I. INTRODUCTION

Remotely Powered Underwater Acoustic Sensor Networks (RPUASN) are composed of a large number of battery-free nodes, which harvest and store the power supplied by an external acoustic source, indefinitely extending their lifetime [1]. Sink Powered Underwater Acoustic Sensor Networks (SPUASN) constitute a special configuration of RPUASN, where the data sink supplies energy to sensors.

Removal of the energy constraint, and the characteristics of underwater acoustic channels [2], such as high propagation delay, ambient noise, and attenuation, raise the need for the design of new communication protocols for SPUASN. For the collaborative operation of SPUASN nodes deployed in a volume powered by the sink, it is imperative that sensed event information is reliably transferred over multiple hops by exploiting the shortest path to the data sink.

A number of solutions have been proposed for the medium access problem in conventional underwater acoustic sensor networks [3]. Deterministic MAC schemes such as TDMA, FDMA, or CDMA cannot be directly adopted in the underwater environment due to problems such as narrow channel bandwidth, vulnerability to fading and multipath, dependence on network-wide clock synchronization, handling long propagation delays, optimizing energy consumption, difficulty of power control at each node to avoid the near-far problem, and scalability with number of nodes [4]. Therefore, majority of the solution efforts have led to random access protocols, mostly based on carrier sense multiple access with collision avoidance (CSMA/CA).

The handshaking protocol in [5] requires synchronization of all nodes. A node is allowed to reserve multiple time slots for transmission, and the scheduled transmission cannot be canceled even if another communicating neighbor is detected. In [6], an improved RTS/CTS handshaking solution that does not need clock synchronization is proposed. RTS/CTS handshaking brings extra delay to MAC operation. The protocol assumes that all nodes use the same transmission power, which is not valid for SPUASN. To solve any uncertainty, coordination of medium access is carried out by a centralized controller in [7]; however, deciding on a network-wide collision-free transmission order requires knowledge of relative locations of all nodes. It is also proposed to choose a transmission order at the receiver side [8]. The receiver has to wait until it receives an RTS from all possible contenders, and this decreases channel utilization seriously. [9] combines carrier sensing with a packet exchange between sender and receiver before transmission. Nodes rely on global time synchronization, and a node is allowed to transmit only at the beginning of a time slot.

Reservation-based medium access is another design alternative [10]-[13]. Nodes estimate propagation delays for scheduling transmissions in [10]. However, this estimation is more applicable when nodes are static and no new node joins the network. [11] is a MAC solution for single-hop underwater networks, where each node has to count how many other nodes contend for the channel. [12] is a TDMA implementation, requiring synchronization of transmission schedules through periodic message exchange. Similarly, in [13], receivers periodically start packet transfers, generating continuous packet exchange overhead in the network. Furthermore, for a dynamic topology under channel fluctuations, it is not very practical to determine when to initiate transmission at each node.

The main objective of traditional underwater routing solutions is the efficient use of limited battery power at nodes [14]. Most of the proposed methods require location information at each node [15]-[18]. Nodes periodically exchange their locations, and routing decisions depend on the location of the sink. For example, in [15], a virtual pipe is formed towards the sink, and only nodes that are close to the pipe can forward data. Protocols in [17] and [18] use periodic beacons broadcast from multiple sonobuoys acting as data sinks. These protocols
require location information for the two-hop neighborhood. Each node is equipped with special hardware to measure its own depth. Next node selection depends upon sequence number, hop count, and depth information exchange among nodes and sinks. Moreover, energy consumption at the special hardware equipment for pressure sensing can lead to lifetime issues. In [19], nodes contain pressure sensors, and a routing decision is made by comparing local depths and forwarding the data packets to the node with the lowest depth. Depth information is used to determine a transmission order via packet holding times, and collision avoidance is not taken into account.

Regarding the transport layer, only a few solutions have been proposed [4]. Those solutions discuss the applicability of error control techniques for the selection of links that optimize power attenuation and transmission delay.

We introduce the Cross Layer Power Adaptive CSMA/CA (X-PACCA) protocol for SPUASN, integrating MAC, network, and transport layer functionalities. It is based on the determination of initial backoff window sizes to be used for packet relaying, according to harvested power levels at nodes. Access priorities are given to nodes closer to the sink, resulting in a loop-free path to reduce end-to-end packet delay. While waiting for medium access to transmit a packet, if a node hears the successful forwarding of the packet by another node, it cancels the transmission. Thus, congestion is avoided via prevention of redundant packet forwarding. Finally, end-to-end reliability is enhanced via acknowledgments sent by the sink upon reception of packets. X-PACCA does not require global network information or clock synchronization at nodes; protocol operation proceeds via local decisions of individual nodes based on harvested power levels.

The remainder of the paper is organized as follows. The operation of the X-PACCA protocol is described in Section II. Performance evaluation of X-PACCA and comparison to some available alternatives in terms of end-to-end delay, packet delivery ratio, and throughput is presented in Section III. Section IV concludes the paper.

II. PROTOCOL DESCRIPTION

X-PACCA is responsible for the following MAC, network, and transport layer functions:

- Organizing the access of SPUASN nodes to the shared medium.
- Routing of data packets along a path from the event region to the sink.
- Congestion avoidance through prevention of redundant packet forwarding.
- End-to-end reliability enhancement via ACKs sent by the sink.

Using the relations given in [1], a node knows its power budget harvested from the sink, \( P_{\text{harv}} \). The transmission power level of the omnidirectional communication transducer of a node is a portion of the harvested power level, and it is denoted as \( MYPL = \beta P_{\text{harv}} \) (\( 0 < \beta < 1 \)). Each node has a unique identifier (ID), denoted by \( MYID \).

Each node maintains the following queue structures:

- A MAC Transmit Queue (MTQ), holding packets to be transmitted and operating according to CSMA/CA with a unique starting window size for each node.
- A list (IGS) of packets to be ignored (already processed and/or relayed), used to store the IDs of packets (PIDs) that must not be re-processed or re-relayed. If PID is already a member of IGS, the list is not modified upon re-insertion.
- A list (WFA) of packets waiting for ACK together with their detection times, maintained only at event data source nodes.

The format for X-PACCA data packets is shown in Fig. 1. A single bit TYPE field is used to identify packets sent by the sink. SID represents the ID of the data source node that has sensed an event, and EVID is the sequence number of the sensed event, which is repeated in a sufficiently long time for sensor network data pipeline capacity. These two fields form the packet ID, \( PID = SID + EVID \). The transmission power level (TPL) is the power level with which the packet was transmitted, either from the original sensing node or when relayed. Event data necessary for the application level is carried in the PAYLOAD bits, and CHK contains the checksum bits for error detection and correction.

![Fig. 1. X-PACCA data packet format.](image)

The operation of X-PACCA upon event sensing is given in Algorithm 1. A node that senses an event constructs a packet with \( PID = MYID + EVID \), \( TPL = MYPL \), \( TYPE = 0 \), filling the corresponding PAYLOAD and CHK fields. It inserts the packet into MTQ, and starts CSMA/CA transmission, applying the backoff procedure as given in Algorithm 3, with a fixed initial backoff window size \( W_s \). Hence, regardless of where an event is sensed, the data source gets a fair chance for medium access. After successful transmission, the packet is deleted from MTQ, the node enters PID and local timestamp value into WFA, and PID is inserted into IGS, meaning that the node has processed the packet. Thus, if the node hears the same packet again from a neighbor, it simply discards the newly arrived copy.

The node periodically checks the (PID, timestamp) pairs in WFA, and those still unacknowledged after \( WFA_{\text{thresh}} \) are re-sent in case the corresponding event EVID is still active.

The behavior of a node upon packet reception is presented in Algorithm 2. When a node receives a packet with \( TYPE = 1 \), that is an ACK coming from the sink, PID is searched and removed, if found, from WFA. The node also deletes PID from MTQ if found, and cancels pending CSMA/CA if not yet completed, because there is no need to relay a packet that has already been received successfully by the sink. Outdated packets are removed from the queues eventually to avoid indefinite growth of queue size; however, PID is not erased from IGS immediately. Instead, the node
Algorithm 1 X-PACCA pseudo-code for event sensing

1:   procedure EVENTSENSED(EVID)
2:     p = CreatePkt(0, MYID, EVID, MYPL, payload, chk)
3:     MTQ.insert(p)
4:     Backoff(Ws, p) ▷ CSMA/CA backoff, initial window size Ws
5:     MTQ.delete(p)
6:     WFA.insert(p, P.ID, timestamp(EVID)) ▷ Wait for ACK
7:     IGS.insert(PID) ▷ If you hear this event again, do not relay packets
8:     while WFA is not empty do ▷ WFA is checked for ACK from the Sink
9:         for all PID ∈ WFA do
10:            if myClock ≥ WFA.timestamp(EVID) + WFA.bthresh then
11:                ▷ No ACK has arrived yet
12:                if SenseEvent(EVID) then
13:                    ▷ The event is still active; re-transmit
14:                        p = CreatePkt(0, MYID, EVID, MYPL, payload, chk)
15:                        MTQ.insert(p)
16:                        Backoff(Ws, p)
17:                        MTQ.delete(p)
18:                        WFA.update(p, P.ID, timestamp(EVID))
19:                else
20:                    ▷ The event is not active anymore
21:                        WFA.delete(p, P.ID, timestamp(EVID))
22:                end if
23:            end for
24:         end for
25:     end while
26:   end procedure

keeps PID in IGS for another duration of D, which is the propagation delay of the longest hop in the network. Hence, duplicate forwarding is avoided in case delayed packet arrivals occur from other neighbors.

When an SPUASN node receives a packet from any neighbor (TYPE = 0), the node compares its own transmission power level (MYPL) with that (TPL) of the sender of the packet. If TPL < MYPL, the packet is coming from a neighbor that is further from the sink, and the node may relay the packet. It ignores the packet if PID ∈ IGS, showing that it has processed and/or relayed the packet before, and this is a duplicated arrival. Otherwise, the node must relay the packet with TPL = MYPL update, and it enters PID into IGS and MTQ. To minimize end-to-end delay, nodes closer to the sink have a higher priority to access the medium for relaying packets. Hence, at relay nodes, the initial backoff window size is adapted according to the power differential between the previous sender and this relay node, and determined as Wmin = k/(MYPL − TPL). Effects of the choice of the backoff constant k on performance is investigated in Section III. Then, the node starts exponential backoff, as detailed in Algorithm 3. When CSMA/CA does succeed in transmission (i.e., no collision) after backoff, the nodes deletes PID from MTQ.

If TPL ≥ MYPL for the received packet, this means the node is hearing a packet relayed ahead of itself, from a node closer to the sink. The node inserts PID into IGS if PID /∈ IGS. The packet has already been relayed by a node closer to the sink, so there is no need to relay a duplicate. The node cancels the pending backoff or transmission procedure for that packet and removes it from MTQ.

Whenever a packet is received at the sink, it is acknowledged by simply setting the TYPE bit and broadcasting it, using Ws as the starting window size. That is, we are assuming that, as the sink can supply power to all sensors in a single hop, the sink can also be heard by all sensors in a single hop. Hence TYPE = 1 packets are not relayed, they simply cause sensors to drop acknowledged packets from their transmission lists.

III. PERFORMANCE EVALUATION

In this section, the performance of X-PACCA is evaluated in terms of design parameters and compared with other protocols that have been proposed for underwater sensor networks. Namely, MAC performance of X-PACCA is compared with UWAN-MAC [12] and Slotted FAMA [9], and the routing performance is compared with DBR [19] and VBF [15].

Algorithm 2 X-PACCA pseudo-code for packet reception

1:   procedure PACKETRECEIVED(p)
2:     if p.Type == 1 then ▷ Sink packet
3:         WFA.find(p, P.ID) == TRUE then ▷ Check WFA
4:         WFA.delete(p, P.ID, time(p, EVID)) ▷ ACKed; do not wait
5:         end if
6:     if MTQ.find(p, P.ID) == TRUE then ▷ Check MTQ
7:         MTQ.delete(p, P.ID) ▷ Cancel retransmission
8:         end if
9:     if IGS.find(p, P.ID) == TRUE then ▷ Check IGS
10:        IGS.scheduleDelete(p, P.ID, D) ▷ Delete PID from IGS at currentTime + D
11:     end if
12:     if IGS.find(p, P.ID) == FALSE then ▷ I have not processed this pkt
13:        if IGS.find(p, P.ID) == FALSE then ▷ Packet from a relay node, p.Type = 0
14:            IGS.insert(p, P.ID) ▷ Packet relayed from another relay node
15:        end if
16:       if p.TPL < MYPL then ▷ Closer to the sink; may relay p
17:          p.TPL = MYPL ▷ Update p.TPL field
18:          MTQ.insert(p) ▷ Packet waiting for transmission
19:          Wmin = k/(MYPL − p.TPL) ▷ Initial backoff window size
20:          Backoff(Wmin, p) ▷ Backoff(Wmin, p)
21:         MTQ.delete(p)
22:       end if ▷ Packet relayed by a node closer to the sink
23:       if MTQ.find(p, P.ID) == TRUE then ▷ Cancel retransmission
24:           MTQ.delete(p, P.ID)
25:       end if
26:     end if
27:     end if
28:     if MYID == SinkID then ▷ I am the sink
29:         if p.Type = 1 then ▷ ACK broadcast from the sink
30:            BroadcastPkt(p) ▷ Broadcast packet
31:         end if
32:     end if
33:     end procedure

Algorithm 3 X-PACCA backoff algorithm

1:   procedure BACKOFF(W, p)
2:     i = 0 ▷ Backoff stage
3:     txFlag = 0 ▷ Transmission flag
4:     txAttempts = 0 ▷ Number of transmission attempts
5:     maxAttempts = the maximum number of attempts
6:     while txFlag == 0 && txAttempts ≤ maxAttempts do
7:         txAttempts = txAttempts + 1 ▷ Window size, stage i
8:         Ws = 2iW ▷ Window size, stage i
9:         cnt = unsf(0, Ws − 1) ▷ Counter
10:        while cnt ≠ 0 do
11:            repeat
12:                if MTQ.find(p, P.ID) == FALSE then ▷ Maximum backoff stage check
13:                   EXIT
14:                else
15:                    if p.TPL < MYPL then ▷ If MTQ.find(p, P.ID) == FALSE then
16:                        if i ≠ m then ▷ Maximum backoff stage check
17:                           i = i + 1
18:                        end if
19:                    end if
20:                    else
21:                        txFlag = 1 ▷ Transmission successful
22:                    end if
23:            end if
24:         end while
25:     end while
26:     end procedure
A. Simulation Settings

All simulations are performed using Aqua-Sim [20], which is a network simulator for underwater sensor networks. Aqua-Sim has the ability of simulating underwater environment for three-dimensional network deployment [22] and implemented as a patch over NS-2 [21]. It adopts a realistic model of the underwater acoustic channel, including multipath effects, time-varying delay, attenuation, and Doppler scaling. The validity of the channel modeled in Aqua-Sim is verified through comparisons with real testbed results [20].

In all simulations, the sink is the external acoustic power source placed at the water surface with the input power of \( P_{elec} = 10 \, \text{kHz} \), operating at the frequency \( f = 10 \, \text{kHz} \) with an electro-acoustic power conversion efficiency of \( \eta = 0.5 \). We assume that the level of harvested power is constant during packet transmission at each node, as the sink continues to supply energy in an uninterrupted fashion. Communication frequency among nodes is set to 25 kbps. Spherical spreading is assumed for acoustic signals.

Unless otherwise stated, default values for X-PACCA and SPUASN node parameters are given in Table I and Table II, respectively.

![Spherical cone deployment for SPUASN.](image)

**Fig. 2.** Spherical cone deployment for SPUASN.

### Table I. X-PACCA Protocol Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data sink/source starting window size ( W_s )</td>
<td>3</td>
</tr>
<tr>
<td>Maximum backoff stage ( (m) )</td>
<td>5</td>
</tr>
<tr>
<td>Maximum number of retransmissions ( (m_{Atmpt}) )</td>
<td>5</td>
</tr>
<tr>
<td>ACK waiting time ( (WFA_{thresh}) )</td>
<td>20 s</td>
</tr>
<tr>
<td>Ignore timeout ( (I) )</td>
<td>50 s</td>
</tr>
<tr>
<td>Aqua-Sim receive threshold</td>
<td>( 8.7 \times 10^{-8} ) W</td>
</tr>
</tbody>
</table>

Unless explicitly stated otherwise, simulations are repeated until all reported average end-to-end delay (latency) values are within \( \pm 10\% \) confidence interval of the actual value with a confidence level (i.e., probability) of 99%. The packet delivery probabilities, however, simply reflect the ratio of successfully delivered packets that constitute the basis for the reported average latencies. That is, in conformance with the commonly accepted convention in networking literature, the statistical confidence of the reported packet loss characteristics are not assessed, as obvious from the relatively inconsistent nature of those results. The consistency of the delay characteristics, however, are taken to be sufficient for an overall evaluation of the performance of the proposed protocol.

### Table II. SPUASN Node Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required power for a node to operate ( (P_{req}) )</td>
<td>0.5 W</td>
</tr>
<tr>
<td>Number of harvesting hydrophones ( (n) )</td>
<td>5</td>
</tr>
<tr>
<td>Receiving voltage sensitivity ( (RVS) )</td>
<td>(-130 , \text{dB re V} / \mu\text{Pa})</td>
</tr>
<tr>
<td>Hydrophone impedance ( (R_H) )</td>
<td>125 , \text{Ω}</td>
</tr>
<tr>
<td>Harvesting efficiency</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Fig. 2 shows the topology for the default simulation parameters. As can be seen from the figure, a spherical cone shaped network is formed and nodes, randomly distributed in the medium and satisfying minimum harvested power requirement, are actively participating the network.

### B. Impact of Design Parameters

First, the impacts of network and protocol parameters over the performance of X-PACCA are examined.

1) **Slot Length:** Slot length is one of the most crucial parameters in terms of delay and throughput. Fig. 3(a) and 3(b) show delay and packet delivery probability for a single 600 B packet over an SPUASN with an average number of 60 active nodes. In this case, a single source is placed at a farthest point to the sink, which is 777 m for a vertex angle of \( \theta = 20\ degrees \) and \( P_{elec} = 10 \, \text{kHz} \) sink power. Nodes continue retransmitting until successful delivery.

Shorter slot lengths reduce the performance by increasing the probability of collision and longer slot lengths cause longer delays. Decreasing the slot length decreases observed delays until a point where collisions begin to dominate. Fig. 3 shows that the effect of \( k \) diminishes with shorter slot lengths.

2) **Density of Nodes:** Effect of node density, again on end-to-end delay and delivery ratio, is investigated for different values of the backoff constant, \( k \).

Data source node is placed at the farthest point to the sink, which is 535 m away for \( \theta = 30\ degrees \) and \( P_{elec} = 10 \, \text{kHz} \). At each run, a 600 B packet is transmitted from the data source to the sink. Packets are retransmitted until no collision occurs; however, the maximum backoff stage is \( m = 5 \), as stated in Table I. According to the results depicted in Fig. 3, slot length in the backoff is taken as 0.1 s.

Fig. 4(a) shows that the choice of \( k \) has a more critical effect on end-to-end delay for low node densities, but proper selection of this constant leads to lower latencies. Higher sensor densities also result in higher packet delivery ratio as expected, as can be seen in Fig. 4(b). However, packet delivery ratio is very close to 100% for all investigated node densities.

3) **Effect of Source Window Size \( W_s \):** A spherical cone shaped topology is formed by the acoustic source as a result of \( \theta = 20\ degrees \) and \( P_{elec} = 10 \, \text{kHz} \). In the simulations, 5 and 10 source nodes are deployed randomly inside this cone such that all sources can deliver packets to the sink in a single hop. All nodes other than the sink are data sources and send 50 B packets with independent but identically distributed Poisson traffic. Backoff constant \( k = 1 \) and \( \beta = 0.025 \) guarantees single
hop access to the sink, according to the range that directly depends on the transmission power [1]. Slot length is 0.005 s. To avoid congestion from multiple sources, the maximum number of transmission attempts is limited to $\text{maxAttempts} = 5$, as given in Table I.

Fig. 5(a) and Fig. 5(b) show latency and delivery ratio values for different source window sizes and different average number of source nodes, respectively. As can be seen from these figures, small window sizes result in better performance at low traffic rates, whereas larger window sizes are needed with higher traffic rates.

### C. Comparison of MAC Performance

MAC performance of X-PACCA is compared with UWAN-MAC [12] and Slotted FAMA [9] in terms of throughput and delay. To get rid of routing effects, network configuration is set such that transmitted packets can reach the sink in a single hop. A spherical cone shaped topology is formed by the acoustic source with $\theta = 30^\circ$ and $P_{\text{elec}} = 10$ kW. An average of 9 active nodes are randomly placed inside this cone, and packet size used is 50 B.

X-PACCA slot length is set to 0.005 s and $k = 0.1$. $\beta$ is set to 0.025 to satisfy single hop requirement of this simulation with the objective of evaluating only the MAC performance of X-PACCA. Maximum backoff stage for Slotted FAMA is set as 15, and maximum burst is taken as 30. Average and deviation of cycle period parameters used in UWAN-MAC simulations are 1 and 0.1, respectively.

Fig. 6(a) shows the receive throughput and Fig. 6(b) shows the delay obtained with all three protocols. As can be seen from figures, UWAN-MAC achieves lower throughputs and higher delays at high packet generation rates compared to the other two protocols. This is an expected result of duty cycling applied in UWAN-MAC, which primarily aims to enhance energy efficiency. In UWAN-MAC, two different nodes might have the same transmitting period resulting a collision, so when traffic rate increases, collisions dominate and throughput will no longer increase. Slotted FAMA and X-PACCA have similar throughputs at low packet generation rates, but when this rate increases Slotted FAMA achieves better throughput as a result of its RTS/CTS handshaking mechanism and trains of packets technique (burst of packets sent with a single handshake). RTS/CTS mechanism in Slotted FAMA decreases collision probability hence results in better throughput values but significant delays in packet transmissions can be observed when compared with X-PACCA.

To sum up, UWAN-MAC, which is recommended mainly for delay-tolerant applications, favors energy efficiency and has lower throughput and high delay at high traffic rates. X-PACCA and Slotted FAMA display similar throughputs at low
packet generation rates; but when this rate increases, the effect of collisions lowers throughput values of X-PACCA. On the other hand, X-PACCA performs much better than both others in terms of delay.

D. Comparison of Routing Performance

In this set of simulations, packet delivery ratio and average end-to-end delay values of X-PACCA are compared with DBR [19] and VBF [15]. The scenario used for simulations is the same as that in [19] except for the topology, operating X-PACCA in a spherical cone shaped geometry, as shown in Fig. 2. All simulations last for 5000 s with 50 packets and each data point is a result of 20 simulations (average of 1000 packets). Slot length is 0.005 s, $k = 0.001$, and packet size is 50 B.

Transmission ranges in DBR and VBF simulations are fixed for all nodes and set as 100 m. In X-PACCA, transmission ranges are proportional to the power harvested by each node.
[1]. Using $P_{elec} = 10$ kW and $\theta = 30^\circ$ for the sink with $\beta = 0.0007$ at the nodes, the average transmission range of nodes in X-PACCA is 95 m, which is slightly less than the fixed transmission range used in DBR and VBF - i.e., favoring DBR and VBF at the expense of X-PACCA. The values of $k$ and $\beta$ used in this simulation guarantee multi-hop operation within the volume of deployment.

Average end-to-end delay and packet delivery ratio performances are plotted in Fig. 7(a) and Fig. 7(b), respectively. Since SPUASN nodes have different transmission ranges increasing with power harvested by node, X-PACCA can send via a smaller number of hops which means smaller delays. X-PACCA also performs best in terms of packet delivery ratio when compared with single sink simulations. Only multi-sink DBR performs better at low number of nodes.

E. Cross Layer Performance of X-PACCA

Finally, we present the latency and delivery success performance achieved when MAC, relay and transport layer functionalities are collectively effective. In the simulations for each data point shown in Figure 8(a) and 8(b), out of a total of 1200 randomly placed nodes, an average of 200 or 400 were within the spherical cone receiving energy from the sink. The electrical power at the sink was $P_{elec} = 10$ kW and $\theta = 30^\circ$, resulting in $R_{max} = 535$ m as the depth of this cone. As in the previous subsection, assuming $\beta = 0.0007$ gives an average transmission range of $r_t = 95$ m at nodes. 20 or 40 data source nodes randomly selected with equal probability among these were generating Poisson traffic of 50 B packets, contributing to the total packet generation rates depicted in the figures. Each simulation was repeated until the latency results were within the targeted level of confidence. X-PACCA slot length was taken as 0.005 s, and the backoff constant was $k = 0.001$. The assumed values of $k$ and $\beta$ enable multi-hop operation, as in the case of routing performance evaluation of X-PACCA. It is shown in Fig. 5 that increasing the starting window size helps in resolving the contention which results from adding more sources to the network. Therefore, $W_s$ was set to 30 to maintain a high success probability when the number of sources is changed between 20 and 40.

With the investigated configuration, packets generated by 40 sources are delivered over a maximum distance of 535 m with a delay less than 2 s, at a delivery ratio more than 80%. Populations of 200 and 400, as analyzed in [1], are above the minimum number of sensors to achieve 1-coverage in the given volume, and hence, even better performance may be obtained with networks of sparser populations.

IV. Conclusion

We have presented a new, cross-layer, power adaptive CSMA/CA protocol (X-PACCA) that caters for MAC, routing and transport layer needs of sink powered underwater acoustic sensor networks. For packet relaying and routing, the protocol exploits differences of the level of power harvested at sensors according to their distance from the data sink which is also the remote supplier of acoustic energy. MAC layer operates according to CSMA/CA rules, and transport layer reliability is achieved through packet acknowledgment by the data sink. Duplicate forwarding, while possible, is effectively suppressed by individual relays dropping packets as soon as relay from a more powerful node, i.e. one that is closer to the sink, is detected. Sink acknowledgments cause not only original data sources to avoid repeated transmission of event information, but also permit intermediate relays to cancel any pending forwarding of already received packets. It is shown that appropriate selection of protocol parameters ensure acceptable latency and delivery performance. As protocol operation is based purely on local decisions, sensor mobility, provided that it is not at severely high levels, would not have a detrimental effect on performance, but the conditions for this need further study. Also deserving further study are issues such as operation with multiple data sinks and possible configurations of RPUASN with multiple energy sources.

REFERENCES


