AN EFFICIENT DATA COLLECTION PROTOCOL FOR UNDERWATER WIRELESS SENSOR NETWORKS

Khaled Day[1], Faiza Al-Salti[2], Abderezak Touzene[1] and Nasser Alzeidi[1]

[1]Department of Computer Science, Sultan Qaboos University, Muscat, Oman
[2]Muscat College, Muscat, Oman

ABSTRACT

This paper presents the design and evaluation of a new data collection protocol for Underwater Wireless Sensor Networks called the Data Collection Tree Protocol (DCTP). It uses an efficient distributed algorithm to proactively construct and maintain a data collection tree rooted at the sink node. The pre-constructed and maintained data collection tree allows the efficient selection of a single forwarding node at each hop when routing a data packet. We prove the correctness of the constructed data collection tree and we show that under some stability conditions, the constructed tree converges to an optimal shortest-path tree. Results of extensive simulations show a big improvement in terms of packet delivery ratio, end-to-end delay and energy consumption compared to the well-known VBF protocol. The simulated cases show increases in the packet delivery ratio between 20% and 122%, reductions in the average end-to-end-delay between 15% and 55% and reductions in the energy consumption between 20% and 50%. These results clearly demonstrate the attractiveness of the proposed DCTP protocol.

KEYWORDS

Underwater Wireless Sensor Networks, Data Collection, Routing Protocols, Performance Evaluation.

1. INTRODUCTION

Underwater wireless sensor networks (UWSNs) are used for underwater data collection for various applications such as undersea exploration of natural resources, natural disaster monitoring, military surveillance, and mines detection. Many aspects of underwater wireless sensor networks have been studied including medium access [1], deployment [2], localization [3], routing [4], energy conservation [5] and void handling [6].

Routing is an important problem for UWSNs. It is more energy-efficient to perform routing in UWSNs using several shorter hops compared to one longer hop [7]. Many routing protocols have been proposed for UWSNs [8][9][10]. One of the well-known routing protocols is the Vector-Based Forwarding (VBF) protocol proposed in [11]. In this protocol a routing pipe from the source to the destination is defined and only the nodes located inside this pipe can contribute to the forwarding of packets from the source to the destination. Another protocol is the Depth-Based Routing (DBR) protocol proposed in [12]. In DBR, a node forwards a received packet only if it is located at a lower depth than the depth of the previous forwarder. In [13] the Focused Beam Routing (FBR) protocol has been introduced. In this protocol only the nodes which are located inside a cone of a certain angle between the source and the destination are allowed to perform forwarding. In [14] the authors have proposed a Multi-Path Routing (MPR) protocol. MPR routes packets from a source to a destination over a multi-path. A multi-path is a sequence of 2-hop sub-paths each using a different relay node for reducing data collision.

DOI: 10.5121/ijcnc.2020.12501

We propose a new approach based on using an efficient distributed algorithm for proactively constructing and maintaining a data collection tree connecting the sensor nodes to the sink node. We call the proposed protocol the Data Collection Tree Protocol (DCTP). Attributed to the availability of the pre-constructed data collection tree, the DCTP protocol outperforms considerably the well-known VBF protocol in terms of energy consumption, packet delivery ratio and end-to-end delay as clearly demonstrated by the obtained simulation results.

2. THE DCTP DATA COLLECTION TREE PROTOCOL

Figure 1 outlines the details of the DCTP protocol and the following is an overview description of the protocol. DCTP proactively constructs and periodically updates a data collection tree which has the sink node as its root. The tree is used for forwarding data packets from the sensor nodes to the sink node. The sink node initiates a distributed algorithm for the construction of the data collection tree by setting its tree LEVEL to zero (root level) and initializing a sequence (period) number SEQ to zero. Periodically (every τ seconds), the sink node increments its SEQ number (indicating a new tree updating period) and sends a beacon packet containing its SEQ and LEVEL values to all sensor nodes in its transmission range. A sensor node receiving a beacon packet makes use of the LEVEL and SEQ values contained in the beacon packet to determine if it is sent by a potential parent node (a node one hop closer to the sink) and whether it is from an old period, the current period or a new period. If it is from a current or a new period then the node makes use of the information in the beacon packet to update its local SEQ value, its tree LEVEL value and its set of potential parents. After processing a beacon packet, a sensor node either discards the packet (if it is from an old period) or sends it to all nodes in its transmission range after inserting its updated LEVEL and SEQ values. This allows each sensor node to maintain a set of potential parent nodes (each one hop closer to the sink node) from which it selects a forwarder node when it needs to forward a data packet towards the sink node. This selection can for example use a round-robin scheme, a random selection or a remaining energy based selection. This allows to balance the data forwarding load among the sensor nodes and maximize the sensors lifetime. When a sensor node has no parent node, it buffers the data packets it generates or it receives for forwarding in a QUEUE until a parent node becomes available in which case it calls a FLUSH_QUEUE function to clear the buffered packets.

When a sensor node wants to send or forward a data packet to the sink node, it invokes the function SEND_DATA_PACKET. In this function, the node sends the packet to the PARENT node. If the PARENT node does not acknowledge reception of the data packet within a certain timeout period, the sensor node removes the PARENT node from the set of potential parent nodes PARENT_SET and selects another parent node from this set unless the set has become empty in which case the node buffers the data packet in its local QUEUE. Upon receiving a data packet, a sensor node sends an acknowledgment back to the sender and invokes the SEND_DATA_PACKET function to forward the packet towards the sink node.

Sink Node s:

- Initialize SEQ_s to 0 and set $LEVEL_s$ to 0
- Every period (every τ seconds) do:
  - SEQ_s = SEQ_s+1 //new period
  - Send a BEACON<SEQ_s, $LEVEL_s$> packet to all sensor nodes in range.
- When a data packet is received from a sensor node
  - Send ACK packet to sender
  - Consume received data packet

Sensor Node u

- SEQ_u = 0; PARENT_SET$_u$ = ∅; $LEVEL_u$ = ∞
- $PARENT_u$ = NULL; $QUEUE_u$ =∅
- When the node *u* receives a BEACON<SEQ_v, $LEVEL_v$> packet from a sender node *v*:
  - If (SEQ_v>SEQ_u) then // new period
    - SEQ_u = SEQ_v; $LEVEL_u$ = $LEVEL_v$+1
    - PARENT_SET$_u$ = {*v*}; $PARENT_u$ = *v*
    - Send BEACON<SEQ_u, $LEVEL_u$> to all sensor nodes in range
    - Call the FLUSH_QUEUE() function
  - else if (SEQ_v = SEQ_u) //received beacon from current period
    - if ($LEVEL_v$< $LEVEL_u$-1) //a better parent (closer to the sink)
      - PARENT_SET$_u$ = {*v*}; PARENT = *v*; $LEVEL_u$ = $LEVEL_v$+1
      - Send BEACON<SEQ_u, $LEVEL_u$> to all nodes in range
      - Call the FLUSH_QUEUE() function
    - else if ($LEVEL_v$ = $LEVEL_u$-1) //an alternative parent
      - PARENT_SET$_u$ = PARENT_SET$_u$∪ {*v*}
      - $PARENT_u$ = SELECT(PARENT_SET$_u$)
      - Call the FLUSH_QUEUE() function
    - else discard received BEACON packet //sender not closer to sink
  - else discard received BEACON packet //an old beacon
- When the sensor node wants to send a data PACKET to the sink node:
  - Call the SEND_DATA_PACKET(PACKET) function
- When the sensor node receives a data PACKET from another sensor node
  - Send ACK packet to sender
  - Call the SEND_DATA_PACKET(PACKET) function
- function SEND_DATA_PACKET(PACKET)
  - while(true)
    - if ($PARENT_u$ = NULL)
      - Add PACKET to $QUEUE_u$ // no parent, queue the packet
      - return false //packet not sent, it was queued
    - Send PACKET to $PARENT_u$ //attempt to forward packet towards sink
    - Wait for ACK (set timer)
    - if (timeout) //no ACK received from parent
      - PARENT_SET$_u$ = PARENT_SET$_u$ – {$PARENT_u$} //remove this parent
      - if (PARENT_SET$_u$ = ∅)
        - $PARENT_u$ = NULL
        - Add PACKET to $QUEUE_u$
        - return false //packet not successfully forwarded, it was queued
      - else $PARENT_u$ = SELECT(PARENT_SET$_u$)
    - else //packet successfully forwarded, update parent to distribute load
      - $PARENT_u$ = SELECT(PARENT_SET$_u$)
      - return true //packet successfully forwarded
- function FLUSH_QUEUE()
  - flag = true
  - while ($QUEUE_u$ ≠ ∅) and (flag = true)
    - extract PACKET from $QUEUE_u$
    - flag = SEND_DATA_PACKET(PACKET)
    - if (flag = false) Add PACKET to $QUEUE_u$ // packet not successfully sent

Figure 1.The proposed DCTP protocol

3. ON THE CORRECTNESS AND OPTIMALITY OF THE TREE

We first prove that the node-parent relation established by the DCTP protocol defines a correct directed tree rooted at the sink node. Therefore, DCTP delivers packets to the sink node following paths along the tree without looping. We then address the issue of optimality of the constructed tree. We show that after a tree update, the resulting tree is a shortest-path tree provided that any connected nodes at the start of the update remain connected for the amount of time needed to complete the update.

Definition 1: Let s denote the since node. At any given time, let $G = (V_G, E_G)$ be the directed graph given by:

$V_G = \{s\} \cup \{\text{all sensor nodes}\}$
$E_G = \{(u, v), u \in V_G, v \in V_G \text{ and } \text{PARENT}_u = v\}$
Definition 2: At any given time, let $T = (V_T, E_T)$ be the directed sub-graph of G given by:
$V_T = \{s\} \cup \{\text{all sensor nodes } u \text{ for which there is a path in } G \text{ from } u \text{ to } s\}$
$E_T = \{(u, v) \in E_G \text{ such that } u \in V_T \text{ and } v \in V_T\}$
We shall prove that at any time, T is a rooted directed tree with root$s$. We first establish the following preliminary results. In the remainder of this section we will use the symbol S instead of SEQ and the symbol L instead of LEVEL for clarity of the proofs.
Lemma 1: For any node u and at any given time, if $v \in$PARENT_SET$_u$, then either ($S_v > S_u$) or ($S_v = S_u$and $L_v < L_u$).

Proof: We prove that for any sensor node u, the property is initially true and that it remains true after any modification that affects this property assuming that the property was true before the modification. Based on the statement of the claimed property, the only modifications that can affect the property are those that modify: (a) PARENT_SET$_u$, (b) S_v, for any $v \in$PARENT_SET$_u$, (c) S_u, (d) L_v, for any $v \in$PARENT_SET$_u$, or (e) L_u. Since PARENT_SET$_u$ is initially set to empty, the claimed property is therefore initially vacuously true. Now we show that the property remains true after any of the modifications (a) to (e) assuming it was true before the modification.

PARENT_SET$_u$ is modified by the DCTP protocol only in the following four situations:

(i) PARENT_SET$_u$ is modified when u receives a BEACON from a node v with $S_v > S_u$. In this case, S_u is set to S_v, PARENT_SET$_u$ is set to $\{v\}$ and L_u is set to L_v+1. After these settings the claimed property is true since after the modification $S_v = S_u$ and $L_u = L_v+1$, hence $L_v < L_u$.

(ii) PARENT_SET$_u$ is also modified when u receives a BEACON from a node v at a time when $S_u = S_v$ and $L_v < L_u$-1. As in case (i), in this case PARENT_SET$_u$ is set to $\{v\}$, L_u is set to L_v+1, and S_u remains equal to S_v. After these settings the property remains true for the same reasons as in (i).

(iii) PARENT_SET$_u$ is also modified when u receives a BEACON from a node v at a time when $S_u = S_v$ and $L_v = L_u$-1, in this case, v is added to PARENT_SET$_u$, while S_u is not modified and remains equal to S_v, and L_u is not modified and remains equal to L_v+1. After these settings the claimed property remains true since for the only added node v to PARENT_SET$_u$, we have $S_v = S_u$ and $L_u = L_v+1$ (hence $L_v < L_u$). No other node w in PARENT_SET$_u$ is affected and hence the property ($S_w > S_u$) or ($S_w = S_u$ and $L_w < L_u$) remains true since it was true before the modification.

(iv) $PARENT\_SET_u$ is also modified by the SEND_DATA_PACKET function when there is a timeout (no ACK received from $PARENT_u$). At this time the node $PARENT_u$ is removed from the set $PARENT\_SET_u$ but the property ($S_v>S_u$) or ($S_v = S_u$ and $L_v<L_u$) remains true for any node v not removed from $PARENT\_SET_u$ since it was true for that node v before the modification. If $PARENT\_SET_u$ becomes empty after removing $PARENT_{u,}$, then the property becomes vacuously true after the modification.

(b) For any $v \in PARENT\_SET_u$, S_v is only modified when node v receives a BEACON packet from a node w with $S_w>S_v$. In this case, S_v is set to a higher value S_w. Since the property was true before the modification, then we must have had either ($S_v>S_u$) or ($S_v = S_u$ and $L_v<L_u$) before the modification. In both cases, we will have $S_v>S_u$ after the modification since S_v is set to a higher value S_w, hence the property remains true.

(c) S_u is only modified when node u receives a BEACON packet from a node v with $S_v>S_u$. In this case, S_u is set to S_v and L_u is set to L_v+1. Therefore, after the modification we have $S_v = S_u$ and $L_v<L_u$.

(d) For any $v \in PARENT\_SET_u$, L_v is modified only in the following two situations:

(i) L_v is modified when node v receives a BEACON packet from a node w with $S_w>S_v$. In this case S_v is set to a higher value S_w and L_v is set to L_w+1. Since the property was true before the modification, then we must have had either ($S_v>S_u$) or ($S_v = S_u$ and $L_v<L_u$) before the modification. In both cases, we will have $S_v>S_u$ after the modification since S_v is set to a higher value S_w, hence the property remains true.

(ii) L_v is also modified when$v$ receives a BEACON from a node w with $S_w = S_v$ and $L_w<L_v$-1. In this case,S_v is not modified and L_v is set to L_w+1 which is smaller than its previous value. Since S_v is not modified and L_v is decreased, the property ($S_v>S_u$) or ($S_v = S_u$ and $L_v<L_u$) remains true, assuming it was true before the modification.

(e) L_u is modified only in the following two situations:

(i) L_u is modified when node u receives a BEACON packet from a node v with $S_v>S_u$. In this case S_u is set to S_v, $PARENT\_SET_u$ is set to $\{v\}$ and L_u is set to L_v+1. After these settings, node v is the only node in $PARENT\_SET_u$ and it satisfies ($S_v = S_u$ and $L_v<L_u$). Hence the claimed property is satisfied.

(ii) L_u is also modified when$u$ receives a BEACON from a node v with $S_v = S_u$ and $L_v<L_u$-1. In this case,S_u is not modified and remains equal to S_v, $PARENT\_SET_u$ is set to $\{v\}$ and L_u is set to L_v+1. As in the previous case, after these settings node v is the only node in $PARENT\_SET_u$ and it satisfies ($S_v = S_u$ and $L_v<L_u$). Hence the claimed property is satisfied. QED

Lemma 2: For any sensor node u and at any time, if $PARENT_u = v$ for some node v, then either ($S_v>S_u$) or ($S_v = S_u$ and $L_v<L_u$).

Proof: As shown in Figure 1, we have either $PARENT_u$ = NULL or $PARENT_u \in PARENT\_SET_u$ at all times including initially and after any modification to $PARENT_{\underline{u}}$ or $PARENT\_SET_u$. Therefore, by Lemma 1 the claimed property is satisfied. QED

Proposition 1: The graph G is a directed acyclic graph (DAG).

Proof: Let R be the binary relation on the set of nodes V_G defined as follows: u R v if and only if ($S_v>S_u$) or ($S_v = S_u$ and $L_v<L_u$). We prove that R is a strict partial order on the set V_G and then derive based on Lemma 2 that G is a directed acyclic graph (DAG).

To prove that R is a strict partial order we have to show that R is (a) irreflexive, (b) transitive and (c) asymmetric.

(a) *R is irreflexive:* Assume u R v for some nodes u and v in V_G. Then either $S_v>S_u$ or $L_v<L_u$. Therefore, $u \neq v$. Hence R is irreflexive.
(b) *R is transitive:* Assume u R v and v R w for some nodes u, v and w in V_G. Since u R v then ($S_v>S_u$) or ($S_v = S_u$ and $L_v<L_u$). Since also v R w, therefore ($S_w>S_v$) or ($S_w = S_v$ and $L_w<L_v$). There are four cases:

   (i) If ($S_v>S_u$) and ($S_w>S_v$), then $S_w>S_u$, hence u R w.
   (ii) If ($S_v>S_u$) and ($S_w = S_v$ and $L_w<L_v$), then $S_w>S_u$, hence u R w.
   (iii) If ($S_v = S_u$ and $L_v<L_u$) and ($S_w>S_v$), then $S_w>S_u$, hence u R w.
   (iv) If ($S_v=S_u$ and $L_v<L_u$) and ($S_w=S_v$ and $L_w<L_v$), then ($S_w=S_u$ and $L_w<L_u$), hence u R w.

Therefore, u R w in all cases. Hence R is transitive.

(c) *R is asymmetric:* Assume u R v for some nodes u and v in V_G. Then either ($S_v>S_u$) or ($S_v = S_u$ and $L_v<L_u$). If $S_v>S_u$ then neither $S_u>S_v$ is true nor $S_u = S_v$ is true. Therefore, v R u is not true. On the other hand, if ($S_v = S_u$ and $L_v<L_u$) then neither $S_u>S_v$ nor $L_u<L_v$. Therefore, v R u is also not true in this case. Hence R is asymmetric.

Therefore, R defines a strict partial order on V_G. Furthermore, by definition of the graph G, for any edge (u, v) in E_G, we have $\text{PARENT}_u = v$ and hence by Lemma 2, we have u R v. Therefore, G cannot possibly contain any cycles. Hence G is a directed acyclic graph (DAG). QED
Corollary 1: The sub-graph T of the graph G is a directed tree.

Proof: By the definitions of G and T, we can infer that for any sensor node u in V_T, there is a path in T from u to s. Therefore, T is a connected graph. Furthermore, T is a sub-graph of G and G is acyclic (by Proposition 1). Therefore, T is also acyclic. Hence T is a directed tree. QED

Now we show that after a tree update, the resulting tree is a shortest-path tree provided that connected nodes remain connected for the amount of time needed to propagate to all reachable nodes a beacon issued by the sink node at the start of the update.

Definition 3: For any sensor node u in V_G, let OPT-DIST_u be the length of the shortest path from u to the sink node s in the graph G.

Proposition 2: If during a tree updating period, any two connected nodes in G remain connected, then we will have $L_u = \text{OPT-DIST}_u$, for any node u in V_T, by the end of the period, assuming the period is long enough to allow for the beacon packet issued by the sink node at the start of the period to be propagated to all sensor nodes.

Proof: (by induction on L_u) As induction basis, the only node u in V_T with $L_u = 0$ is the sink node s and we obviously have $L_s = \text{OPT-DIST}_s = 0$. Assume that during a given tree updating period, any two connected nodes in G remain connected. Assume also that this period is long enough to allow for the beacon packet issued by the sink node at the start of the period to be propagated to all reachable nodes. As induction hypothesis, assume that by the end of the period, $L_w = \text{OPT-DIST}_w$ for any node w in V_T satisfying $L_w<k$ (for some $k> 0$). Now as induction step, consider a node u in V_T with $L_u = k$. Let v be the node selected by u as its parent by the end of the current period. Let S be the sequence number (period number) of the beacon packet issued by the sink node at the start of the period. Since this beacon packet had enough time to be propagated to all reachable sensor nodes before the end of the period, all these nodes will have their sequence number equal to S by the end of the period. Therefore, $S_u = S_v = S$ by the end of the period. Furthermore, by Lemma 2, either ($S_v>S_u$) or ($S_v = S_u$ and $L_v<L_u$). Since $S_u = S_v = S$, we must have $L_v<L_u = k$. Hence by induction hypothesis we have L_v = OPT-DIST$_v$ by the end of the period. In addition, there is no sensor node v' in G within transmission range of u and such that $L_{v'}<L_v$, otherwise DCTP would have selected v' as parent instead of v when v' has sent the beacon packet of the current period to all sensor nodes in its transmission range. Therefore, OPT-DIST$_u$ = OPT-DIST$_v$ + 1 = L_v + 1 = L_u. QED

Corollary 2: If any two connected nodes in *G*at the start of a tree update remain connected until completion of the update then the resulting tree *T* is a shortest-path tree.

4. PERFORMANCE EVALUATION OF DCTP

We have simulated the DCTP protocol using the Aqua-Sim [15] simulator. Aqua-Sim is a special simulator for underwater networks which supports 3D deployment.

Table I lists the simulations settings used in the evaluation. We have assumed the technical specifications of a real underwater acoustic sensor modem. More specifically, the values of the transmission range, transmission power, reception power, idle power, frequency, bit rate and bit error rate are of the acoustic modem UWM2000H [16], which is available in the market.

According to [17], underwater sensor nodes move with water currents typically with speeds ranging between 3 and 6 km/h (i.e., around 0.8-1.6 m/s). The node speeds used to evaluate our protocol are 0, 0.5, 1.0 and 1.5 m/s.

A CSMA-based MAC protocol is assumed which has been used by several other studies such as [11] and [18]. We have also assumed the Reference Point Group Mobility (RPGM) mobility model [19] which is suitable for UWSNs [20]. In each simulation run a set of source nodes is selected randomly to inject data packets according to an exponential distribution. We have averaged 25 runs and presented the results with 95% confidence intervals.

The simulation settings listed in Table 1 are similar to the settings we have used in a previous study [21] for evaluating the performance of a grid-based routing protocol.

We have chosen the VBF [11] routing protocol to compare DCTP against since it is one of the most widely cited routing protocols in UWSNs, and it has been used in simulation and comparison with other protocols.

We have measured the packet delivery ratio, the amount of consumed energy and the average packet delivery delay for both DCTP and VBF and compared their performance. We have assessed the impact on these measures of the following parameters: (a) the number of nodes (with values 54, 162 and 270), (b) the traffic load (with sending probability values 0.3, 0.5 and 0.7) and (c) the node mobility speed (with values 0,1 and 1.5 m/sec).

Table 1. Simulation settings

Parameter	Value
Underwater region	3 km × 3 km × 3 km
Transmission range	1 km
Transmission power	8.0 W
Reception power	0.8 W
Idle power	0.008 W
Frequency	35.695 kHz
Bit rate	17.8 kbps
Bit error rate	10e-9
MAC protocol	CSMA-based
Mobility model	RPGM
Number of nodes	54, 162, 270
Initial energy	300 J
Data packet size	150 Bytes
Traffic injection rate	0.07 packets/s
Sink beacon period	(R/2)/max speed
Sensor beacon period	2 × beacon period
Maximum speed	(0, 0.5, 1.0, 1.5) m/sec
Sending probability	0.3, 0.5, 0.7
Pipe width (for VBF)	400 m
Simulation run	2000 seconds
Parameters for the RPGM Mobility Model	
Pause time	20 s
Distribution of nodes	10 nodes/group
Probability of group changes	0.01
Maximum distance to group centre	2.5
Standard deviation	2.0

4.1. Effect of the Number of Nodes

The results of Figure 2 show how the Packet Delivery Ratio (PDR) varies as a function of the number of deployed sensor nodes. Clearly, the PDR for each of DCTP and VBF increases when increasing the number of sensor nodes. This is justified by the increase in the probability of finding candidate forwarders. However, DCTP has increased PDR compared to VBF by more than 20% in all tested cases. For instance, when the number of sensor nodes is set to 270, DCTP has improved PDR by 21% compared to VBF. This percentage has reached 42% for 162 nodes and 122% for 54 nodes. This can be explained by the fact that in DCTP each node selects one forwarding node at a time while in VBF data packets are broadcasted in each hop, which increases the channel congestion and the probability of dropping packets.

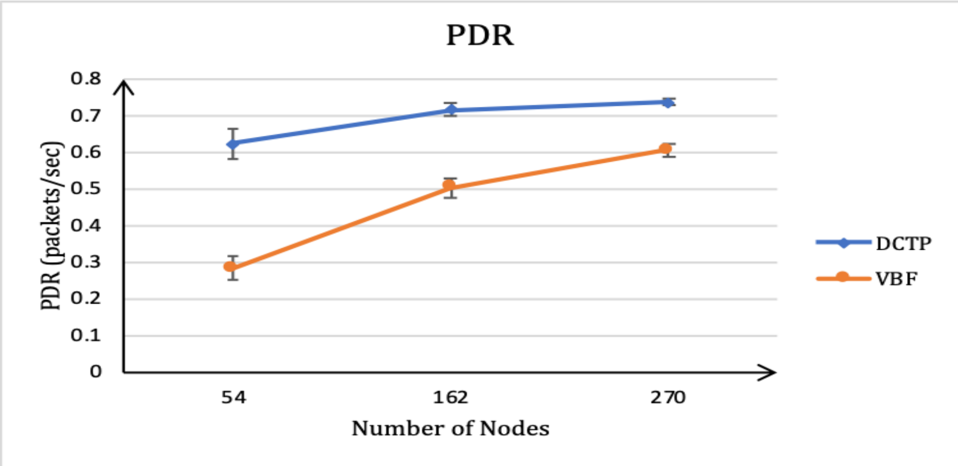

Figure 2. PDR vs the number of nodes

Figure 3 shows how the average end-to-end delay is affected by the number of sensor nodes. The two protocols exhibit different trends. In VBF, the end-to-end delay goes up when the number of deployed sensor nodes is increased. This is because when there are more sensor nodes, the number of forwarders for the same packet increases causing congestion and long waiting times for channel access. In DCTP on the other hand, the average end-to-end delay decreases with the increase in the number of nodes. This is due to the increase in the probability of finding a forwarder for relaying a packet. Furthermore, the congestion in DCTP is lower compared to VBF as only one node participates in the packet forwarding. Figure 3 shows that DCTP has reduced the end-to-end delay compared to VBF by 15%, 50% and 55% for 54, 162 and 270 nodes respectively.

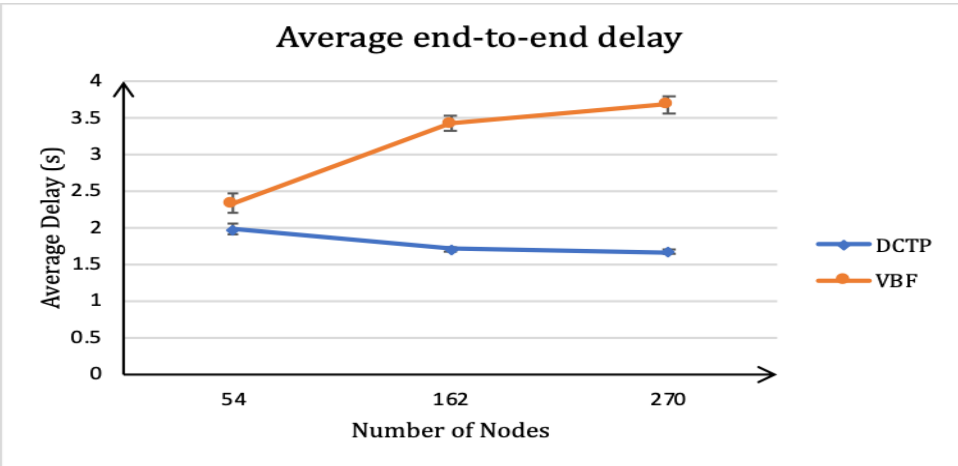

Figure 3. Average end-to-end delay vs the number of nodes

Figure 4 shows how the total energy consumed by the deployed nodes varies with the number of sensor nodes on. When the number of nodes goes up, the two protocols consume more energy due to the increase in the number of transmissions and receptions of packets. However, VBF exhibits a faster increase of the energy consumption compared to DCTP. This can be justified by the increase in the number of nodes broadcasting data packets in VBF, and hence, multiple copies of the same packet are propagated in the network. On the other hand, a data packet in DCTP is forwarded to only one node at a time. Although DCTP uses beacon control packets to proactively build and maintain the data collection tree, but this overhead becomes substantially lower than the number of duplicate data packet retransmissions in VBF when the number of nodes is higher. As shown in Figure 4 DCTP saves 35% of energy compared to VBF when the number of sensor nodes reaches 270 nodes.

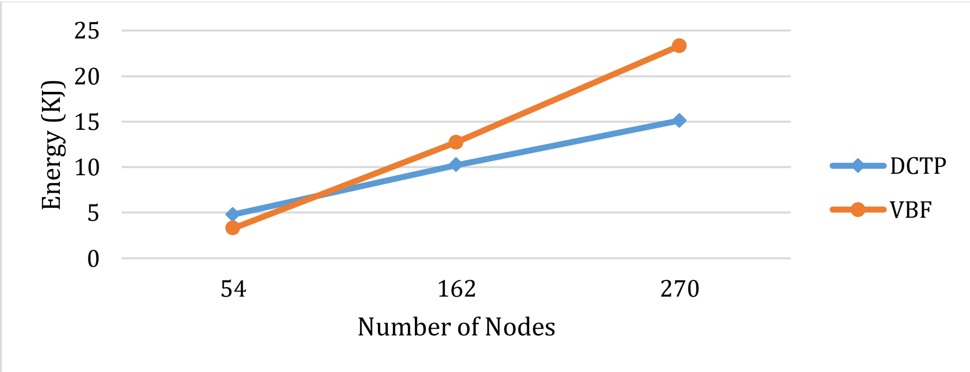

Figure 4. Energy consumption vs the number of nodes

4.2. Effect of the Traffic Load

Figure 5 shows how the sending probability impacts PDR. When the sending probability is increased PDR decreases for both DCTP and VBF. The reason is that when the sending probability increases the number of data packets propagated in the network increases leading to channel access conflicts. This causes more collisions among packets and hence a lower delivery ratio. Though, DCTP has achieved higher PDR than VBF. This is due to the efficient forwarding mechanism of DCTP which sends only one copy of the data packet each time. As an example, when the sending probability is 0.7, DCTP has improved PDR by 43% compared to VBF.

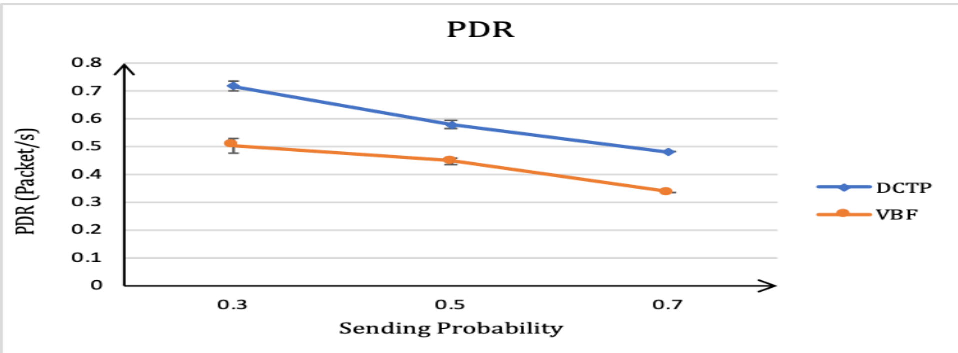

Figure 5. PDR vs the sending probability

Figure 6 shows that DCTP outperforms VBF in terms of the average end-to-end delay for all tested values of the sending probability. DCTP has reduced the end-to-end-delay compared to VBF by over 50%. The reason is that in VBF there are multiple forwarders of a packet at each hop. This increases traffic and hence the queueing delays. On the other hand, in DCTP only one forwarder is selected at each hop.

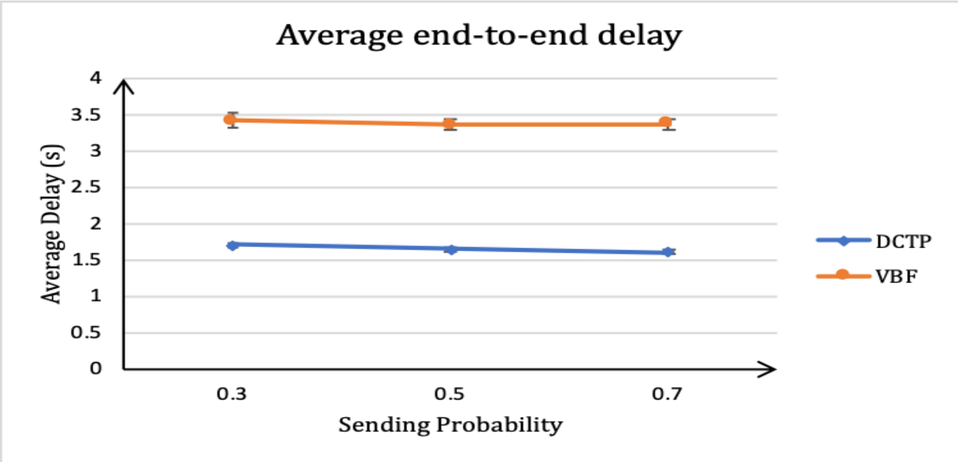

Figure 6. Average end-to-end delay vs the sending probability

As demonstrated in Figure 7, the energy consumption increases with the increase in the nodes sending probability for the two protocols. The reason is that as the sending probability increases, the number of source nodes generating data packets increases. This leads to the increase in the number of propagated data packets. However, DCTP has consumed less energy than VBF. For example, with a sending probability of 0.7, DCTP has reduced the energy consumption by 20% compared to VBF. This is due to the fact that a forwarder in VBF broadcasts the packet to all nodes within its transmission range. Hence, in VBF multiple copies of a single data packet are forwarded in the network which consumes more energy.

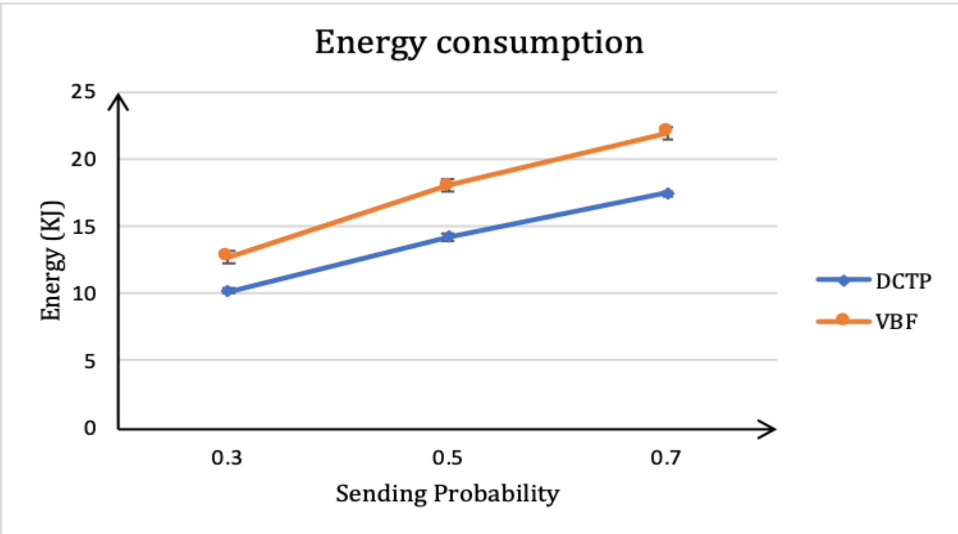

Figure 7. Energy consumption vs the sending probability

4.3. Effect of the Nodes Mobility

The PDR achieved by the two protocols as a function of the nodes' speed is shown in Figure 8. For both protocols the PDR is not affected much by the nodes speed. This may be attributed to the relatively low speeds used in the simulation. Nevertheless, DCTP has outperformed VBF in delivering data packets for all tested node speeds. For example, with a node mobility speed of 1 m/sec, DCTP has improved the PDR by 24% compared to VBF.

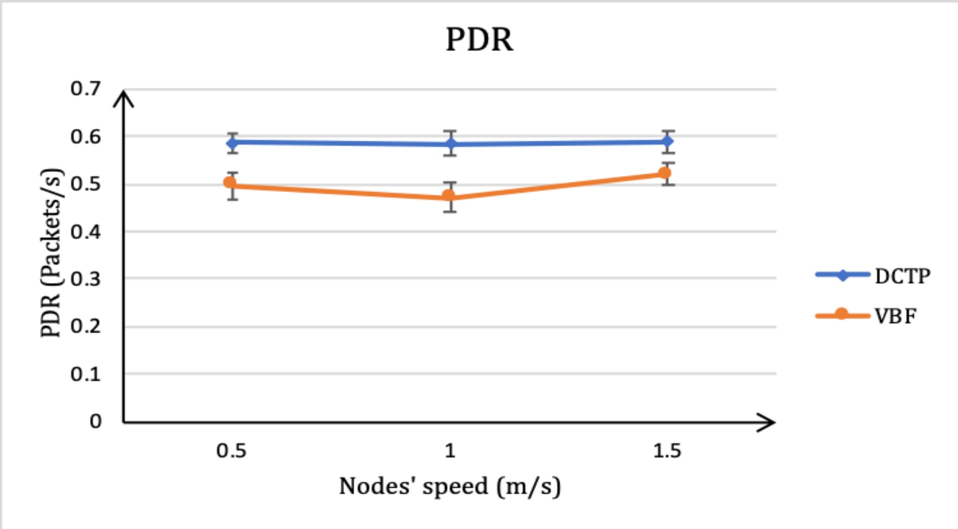

Figure 8. PDR vs the nodes' mobility speed

The average delay experienced by the data packets received by the sink node as the nodes speed varies as illustrated in Figure 9. With the increase in the speed of the nodes, there is almost no variation in the average end-to-end delay for both protocols VBF and DCTP. However, DCTP incurred lower average end-to-end delay than VBF for all tested node speeds. For instance, with a mobility speed of 1 m/sec, DCTP incurred49% lower end-to-end delay than VBF.

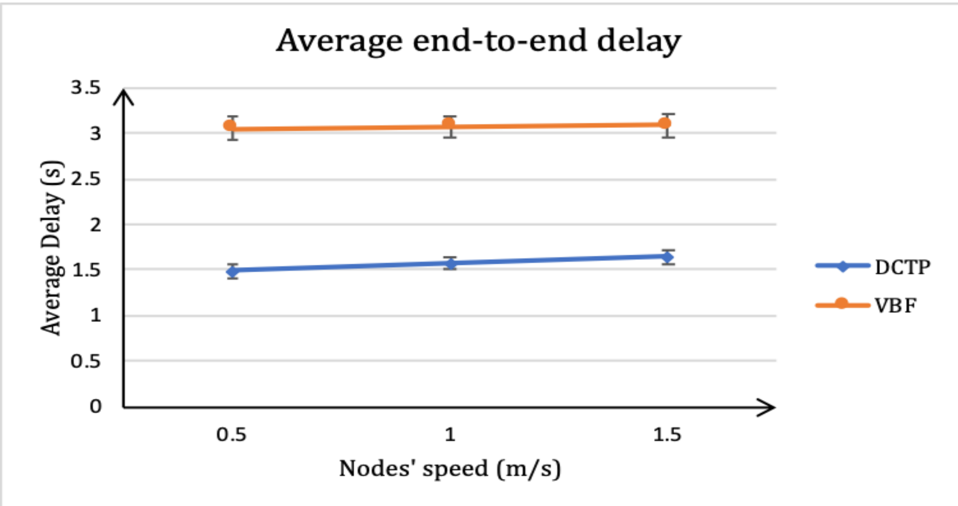

Figure 9. Average end-to-end delay vs the nodes' mobility speed

Figure 10 shows the energy consumption of the two protocols for different nodes mobility speeds. DCTP has outperformed VBF in saving energy for all tested mobility speeds. For example, with a node mobility speed of 1m/sec, DCTP has consumed nearly 50% less energy than VBF.

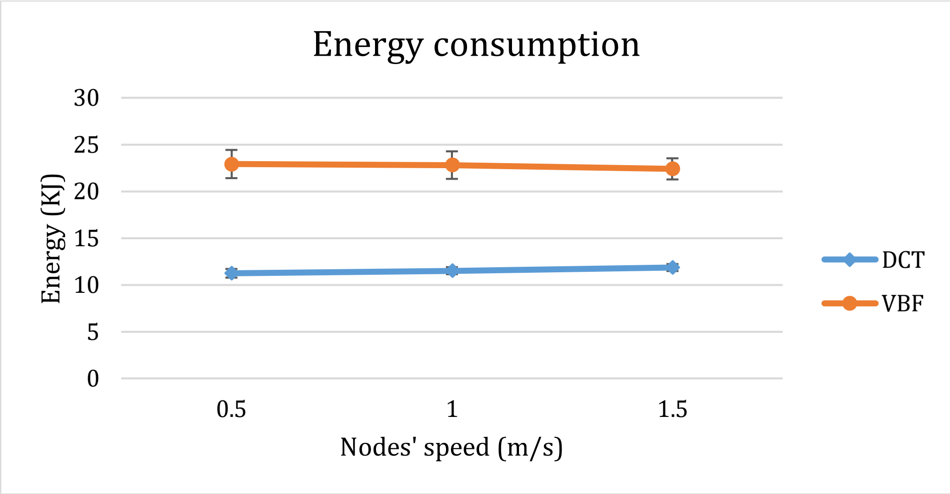

Figure 10. Energy consumption vs the nodes' mobility speed

Despite the fact that DCTP incurs some communication overhead (beacon packets) to build and maintain the data collection tree, it has outperformed VBF in all three measures of delivery ratio, communication delay and energy consumption. This is attributed to the efficient forwarding mechanism of DCTP which selects a single forwarder closer to the sink node at each routing step.

5. CONCLUSION

A new data collection protocol for Underwater Wireless Sensor Networks called DCTP is proposed and evaluated. An efficient distributed algorithm is used to proactively construct and maintain a data collection tree connecting the sensor nodes to the sink node. The pre-constructed and maintained data collection tree allows a fast and efficient selection of a single forwarding node at each hop when routing a data packet. We have proved the correctness of the data collection tree construction and we have shown that under some stability conditions, the constructed tree converges to an optimal shortest-path tree. Results from extensive simulations have shown that DCTP considerably improves energy consumption, packet delivery ratio and end-to-end delay compared to the well-known VBF protocol.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

ACKNOWLEDGEMENTS

Supported by Sultan Qaboos University grant No. IG/SCI/COMP/19/01.

REFERENCES

[1] K. Chen, M. Ma, E. Cheng, F. Yuan, and W. Su, A Survey on MAC Protocols for Underwater Wireless Sensor Networks, IEEE Comm. Surveys & Tut., vol. 16, no. 3, pp. 1433–1447, 2014.
[2] G. Tuna, G andV.C. Gungor, A Survey on Deployment Techniques, Localization Algorithms, and Research Challenges for Underwater Acoustic Sensor Networks. Int. J. Commun. Syst., Vol. 30, Issue 17, pp. 1-21, 2017.
[3] Z. Wang,X. Feng, G. Han, Y. Sui, and H. Qin, EODL: Energy Optimized Distributed Localization Method in 3DUnderwater Acoustic Sensors Networks. Computer Networks 2018, 141, 179–188.

[4] F. Al Salti, N. Alzeidi, and B. Arafeh, EMGGR: An Energy-Efficient Multipath Grid-Based Geographic Routing Protocol for Underwater Wireless Sensor Networks, Wireless Networks, volume 23, no. 4, pp. 1301–1314, May 2017.

[5] R. Gomathi andJ.M.L. Manickam, Energy Efficient Shortest Path Routing Protocol for Underwater Acoustic Wireless Sensor Network. Wireless Personal Commun. 2018, 98, 843–856.

[6] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, Void-Handling Techniques for Routing Protocols in Underwater Sensor Networks: Survey and Challenges. IEEE Communications Surveys Tutorials 19, 2 (Secondquarter 2017), 800–827, 2017.

[7] Z. H. Jiang, Underwater Acoustic Networks-Issues and Solutions, International Journal of Intelligent Control and Systems, vol. 13, no. 3, pp.152–161, 2008.

[8] M. Ahmed, Routing Protocols for Underwater Wireless Sensor Network Based on Location: A Survey, Ad Hoc & Sensor Wireless Networks, 38:67-101, 2018.

[9] T. Islam, andY. K. Lee, A Comprehensive Survey of Recent Routing Protocols for Underwater Acoustic Sensor Networks. Sensors, 19(19), 4256. doi:10.3390/s19194256.

[10] G. Han, J. Jiang, N. Bao, L. Wan, and M. Guizani, Routing Protocols for Underwater Wireless Sensor Networks, IEEE Communications Magazine, pp. 72-78, November 2015.

[11] P. Xie, J.-H. Cui, and L. Lao, VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks, in Proceedings of IFIP Networking'06, Coimbra, Portugal, 2006, pp. 1216–1221.

[12] H. Yan, Z. J. Shi, and J.-H. Cui, Depth Based Routing for Underwater Sensor Networks, in Proc. 7th International IFIP-TC6 Networking Conference on Ad-Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, Singapore, 2008, pp. 72-86.

[13] J. M. Jornet, M. Stojanovic, and M. Zorzi, Focused Beam Routing Protocol for Underwater Acoustic Networks, I Proc. International Conference on Mobile Computing and Networking, ACM International Workshop on Underwater Networks, San Francisco, 2008. pp. 75–82.

[14] Y.-S. Chen, T.-Y. Juang, Y.-W. Lin, and I.-C. Tsai, A Low Propagation Delay Multi-Path Routing Protocol for Underwater Sensor Networks,J. of Int. Technology, vol. 11, no. 2, 153–165, 2010.

[15] P. Xie et al., Aqua-Sim: An NS-2 Based Simulator for Underwater Sensor Networks, in Proceedings of MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges (OCEANS 2009), 2009, pp. 1–7.

[16] LinkQuest: Underwater Acoustic Modem Models. [Online]. Available: http://www.link-quest.com/html/models1.htm. [Accessed: 10-Apr-2018].

[17] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, The challenges of building mobile underwater wireless networks for aquatic applications, *IEEE Networks*, vol. 20, no. 3, pp. 12–18, May 2006.

[18] Z. S. Peng Xie, Zhong Zhou, Zheng Peng, Jun-hong Cui, Void Avoidance in Mobile Underwater Sensor Networks, in WUWNet’07, 2007.

[19] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, A Group Mobility Model for Ad Hoc Wireless Networks, in Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWiM ’99, 1999, pp. 53–60.

[20] Z. Zhou, Z. Peng, J.-H. Cui, Z. Shi, and A. Bagtzoglou, Scalable Localization with Mobility Prediction for Underwater Sensor Networks, IEEE Trans. Mob. Comput., vol. 10, no. 3, pp. 335–348, Mar. 2011.

[21] Faiza Al-Salti, Nasser Alzeidi , Khaled Day , Abderezak Touzene, An Efficient and Reliable Grid-Based Routing Protocol for UWSNs by Exploiting Minimum Hop Count, Computer Networks 162 (2019) 106869.

AUTHORS

Khaled Day received his PhD degree in computer science from the University of Minnesota (USA) in 1992. Heis holding a Professor position atthe Department of Computer Science at Sultan Qaboos University. His research interests are related to parallel and distributed computing and networks. He is a senior member of IEEE.

Dr. Faiza Al-Salti received her BSc, MSc and PhD degrees in computer science from Sultan Qaboos University in 2012, 2015 and 2019, respectively. She is currently holding an Assistant Professor position at Muscat College (Oman). Her research interests include communication protocols and terrestrial and underwater wireless sensor networks.

Abderezak Touzene received the BS degree from the University of Algiers in 1987, the M.Sc. degree from Orsay Paris-Sud University in 1988 and the Ph.D. degree from the Institute Polytechnique deGrenoble (France) in 1992 (all in computer science). He is holding a Professor position at the Department of Computer Science at Sultan Qaboos University. His research interests include Cloud Computing, Parallel and Distributed Computing, Wireless and Mobile Networks, Network On Ship (NoC), Cryptography and Network Security, Interconnection Networks, Performance Evaluation, Numerical Methods. Dr. Touzene is a member of the IEEE, and the IEEE Computer Society.

Nasser Alzeidi received his PhD degree in computer science from the University of Glasgow (UK) in 2007. He is currently an AssociateProfessor of computer science and the Director of the Center for Information Systems at Sultan Qaboos University. His research interests include performance evaluation of communication systems, wireless networks, interconnection networks, System on Chip architectures and parallel and distributed computing. He is a member of the IEEE.