TBR: Tree-Based Routing over a 3D Grid for Underwater Wireless Sensor Networks

Khaled Day, Hussein Al-Moqbali, Nasser Alzeidi, and Abderezak Touzene
Sultan Qaboos University, Muscat, Oman
Email: {kday, alzidi, touzene}@squ.edu.om; halmaqbali@gmail.com

Abstract —The paper presents a tree-based routing approach for Underwater Wireless Sensor Networks (UWSN). The proposed approach uses pre-constructed shortest-path trees in a virtual 3D grid topology to obtain routing paths between source and sink nodes in the UWSN. Due to the availability of these pre-constructed shortest-path trees connecting the occupied cells of the 3D grid, the proposed protocol offers higher routing path stability compared to other protocols that rely on costly reactive mechanisms for path establishment and maintenance.

Index Terms—Tree-based routing approach, underwater Wireless Sensor Networks (UWSN), 3D grid, routing protocols

I. Introduction

Underwater Wireless Sensor Networks (UWSNs) are used for communication underwater for military and non-military applications such as undersea exploration of natural resources, tactical supervision and mines detection. Many related research problems have attracted attention including deployment strategies [1], reliable communication, routing [2], medium access [3], localization [4] and energy conservation [5].

Routing protocols for UWSNs should take into consideration several constraints including dynamic topology, limited energy, low bandwidth and high propagation delay. Routing packets over multiple short hops has been proven to be more energy-efficient for UWSNs than using a single long hop [6]. However, routing over more hops ultimately degrades the end-to-end reliability especially for the harsh underwater environment.

Several routing protocols for UWSNs have been proposed (see [7] and [8] for some examples of protocols). In [9] the authors have proposed a protocol called Vector-Based Forwarding Protocol (VBF). In VBF, each packet carries the positions of the sender, the target and the forwarder (i.e., the node which forwards this packet). The forwarding path is specified by the routing vector from the sender to the target. Upon receiving a packet, a node computes its relative position to the forwarder by measuring its distance to the forwarder. Recursively, all the nodes receiving the packet compute their positions. If a node determines that it is close to the routing vector enough, it puts its own computed position in the packet and continues forwarding the packet; otherwise, it simply discards the packet. Therefore, the forwarding path is virtually a routing “pipe” from the source to the target and only the sensor nodes inside this pipe are eligible to forward packets.

In [10] the authors propose an improvement to VBF called Hop-by-Hop Vector-Based Forwarding (HH-VBF). Unlike the VBF approach, HH-VBF uses a routing vector for each individual forwarder in the network, instead of a single network-wide source-to-sink routing vector. The hop-by-hop vectors of HH-VBF allow overcoming two problems of VBF related to low delivery ratio for sparse networks and sensitivity to the routing pipe radius threshold.

In [11] the authors propose a depth based protocol called DBR. It requires knowledge of the local depth at each node which can be obtained from a depth sensor. The authors propose to forward the packets greedily (based on the depth of each sensor node) to the receivers at the surface of the water. When a node receives a packet, it forwards it if its depth is smaller than the one embedded in the packet otherwise it discards it.

In [12] the Focused Beam Routing Protocol is introduced. When a node A wants to send a packet to a node B, node A issues a multicast Request to Send (RTS) to its neighbors. All the nodes that receive A’s multicast RTS first calculate their location relative to the AB line. Nodes which lie within a cone of a specified angle emanating from the transmitter towards the final destination are candidates for forwarding.

In [13] a routing protocol for UWSNs, called multi-path routing protocol (MPR), is proposed to improve the transmission delay. It uses a multi-path during the path construction from the source node to the destination node. A multi-path is composed of a series of sub-paths. Each sub-path is from a sending node to a two-hop neighboring node.

Most existing protocols use costly mechanisms for establishing and maintaining routing paths. In [2] however, the authors proposed a geographic routing protocol which uses pre-constructed disjoint paths to route in a virtual 3D grid topology which has reduced the cost of establishing and maintaining routing paths. Similarly, we propose a Tree-Based Routing (TBR) protocol which builds and makes use of shortest path

Manuscript received August 13, 2017; revised October 25, 2017.
This work was partially supported by The Research Council (TRC) of the Sultanate of Oman under Grant No. RC/SCI/COMP/15/02.
Corresponding author email: kday@squ.edu.om.
doi:10.12720/jcm.12.10.579-584

trees connecting the occupied cells of a virtual 3D grid (instead of using disjoint paths as in [2]) for routing packets. Due to the availability of the pre-constructed shortest path trees, the proposed TBR protocol offers higher stability of the routing paths compared to other protocols. This has a positive impact on the performance of the protocol in terms of reducing communication overheads (hence energy consumption) and increasing the delivery ratios. The rest of this paper is organized as follows. Section II presents some definitions and introduces the idea of using shortest-path trees over a virtual 3D grid for routing in UWSNs. Section III presents the routing tree construction algorithm. Section IV describes the proposed routing protocol. Section V discusses some performance aspects of tree-based routing over a virtual grid, and Section VI concludes the paper.

II. Routing Trees over a Virtual 3D Grid

The proposed TBR protocol views the physical area of the UWSN as a virtual 3D grid of equal size cells (see Fig. 1). A packet travels from a source sensor node to a sink node at the surface by hopping repeatedly from a grid cell to a neighboring grid cell following a path on a pre-constructed tree connecting the grid cells. In each cell a selected cell-head is responsible of relaying the packets that need to cross that cell to neighboring cells. The cell-head responsibility in a given cell is shared among the nodes located in that cell in the sense that any node in a given cell is a potential cell-head as viewed by nodes in neighboring cells. Different nodes in neighboring cells of a given cell C may each select (independently) a different node to be the cell-head for C. This has the advantage of minimizing the cell-head selection overhead and avoiding to overload a particular node in a cell with all the packet-forwarding requests coming from neighboring cells.

A routing path from a source sensor node to a sink node is a sequence of grid cells that need to be crossed in order to go from the cell hosting the source node to the cell hosting the sink node. TBR constructs at each node a tree rooted at the node's cell connecting it to all occupied cells of the 3D grid.

The information needed by the routing algorithm is maintained at each node in the form of three tables, namely, a Location Table (LT), a Neighbors Table (NT) and a Routing Tree Table (RTT). The Location Table contains for each sensor node, its node id and its location. The location of a node is specified by a grid cell address. A grid cell address is a triple (x, y, z) of coordinates in the 3D grid (assuming the address of the bottom right front corner cell is (0, 0, 0) as illustrated in Fig. 1). The Neighbors Table (NT) at a given node contains, for each of its neighboring cells (including the local cell), a list of ids of all nodes located in that cell. The Routing Tree Table (RTT) contains one entry RTT[C] for each cell C in the 3D grid (where C is a cell address). The entry RTT[C] contains a (Boolean) field RTT[C].occupied indicating whether cell C is occupied or non-occupied. An occupied cell is a cell that hosts at least one sensor node. We also refer to an occupied cell as a non-empty cell and to a non-occupied cell as an empty cell. If C is occupied, a field RTT[C].next indicates the neighboring cell to go to from the current cell (i.e. the next hop) in order to reach cell C. The next hop is determined based on a breadth-first search tree rooted at the node's cell connecting it to all reachable occupied cells. The algorithm for building this routing tree is presented in the next section.

Fig. 1, Fig. 2 and Table I show respectively an example of an UWSN viewed as a virtual 3D grid, a routing tree rooted at node S of Fig. 1, and the representation of this tree in the form of a Routing Tree Table (RTT).

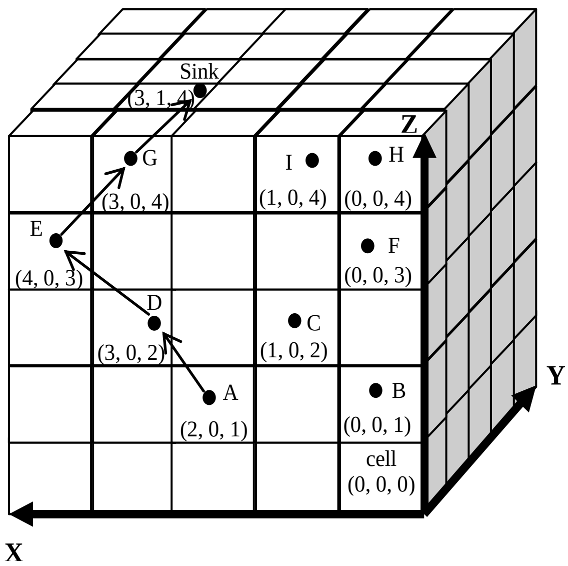

Fig. 1. A virtual 3D grid view of the UWSN.

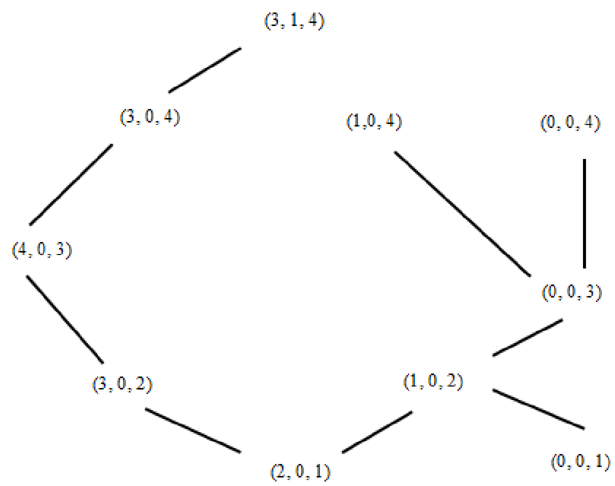

Fig. 2. Shortest path tree rooted at cell (2, 0, 1).

Table I: Routing Tree Table (RTT) at node A of Fig. 1.

Cell	(0,0,1)	(2,0,1)	(1,0,2)	(3,0,2)	(0,0,3)	(4,0,3)	(0,0,4)	(1,0,4)	(3,0,4)	(3,1,4)
Parent	(1,0,2)	NULL	(2,0,1)	(2,0,1)	(1,0,2)	(3,0,2)	(0,0,3)	(0,0,3)	(4,0,3)	(3,0,4)
Next	(1,0,2)	NULL	(1,0,2)	(3,0,2)	(1,0,2)	(3,0,2)	(1,0,2)	(1,0,2)	(3,0,2)	(3,0,2)

III. Routing Tree Construction

At each node, TBR constructs a tree connecting the node's cell (root of the tree) to all reachable occupied cells. The tree is used to determine the next hop fields of the Routing Tree Table (RTT). The tree is a breadth-first search tree of the graph of occupied cells. The vertices of this graph are the occupied cells and two occupied cells are connected by an edge if they are neighboring cells in the 3D grid. The breadth-first search algorithm is outlined in Fig. 3. It uses a FIFO queue storing a list of occupied cells not yet visited in the search. The breadth-first search algorithm starts by enqueueing the root cell in the initially empty queue and then loops dequeueing at each iteration one cell from the queue and enqueueing its unvisited occupied neighboring cells until the queue becomes empty. In each iteration the links between the dequeued cell and the enqueued neighboring cells are recorded in the routing tree table as child-parent links of the routing tree. Finally, the algorithm sets for each cell C the field RTT[C].next to be the neighboring cell leading to C from the local cell along the tree.

```
Routing Tree Construction Algorithm:
  create a FIFO queue Q and initialize it to empty
  for every occupied cell C in the grid do
        RTT[C].next = NULL
  enqueue the local node's cell (root cell)
  while Queue is not empty
        dequeue one cell C from Q
        for each neighboring cell C' of C do
              if (RTT[C'].occupied) and (RTT[C'].next = NULL)
                    RTT[C'].parent = C
                    enqueue C'
  for every cell C other than the local node's cell do
        RTT[C].next = C
        while RTT[C].next is not a neighbor to local node's cell do
                    RTT[C].next = RTT[RTT[C].next].parent
```

Fig. 3. Routing tree construction algorithm in TBR.

The routing tree table is rebuilt when there is a change in the occupied/non-occupied status of any cell. To reduce the number of times the table is rebuilt, TBR uses a Boolean flag valid indicating whether the current table is valid or not depending on whether there have been changes to the occupied status of any grid cell since the table was built the last time or not. A node does not rebuild its routing tree table until it has to route a data packet and the flag valid is false. The value of the flag valid is updated based on exchange of some control packets as explained in the next section.

IV. The TBR Protocol

A. Control Packets in TBR

TBR uses the following control packets:

a) *Empty_to_Non-Empty (ENE)* packet*:* when a node enters an empty cell C, it broadcasts (using cell-based broadcasting) an Empty_to_Non-Empty (ENE) packet. In cell-based broadcasting of a packet only cell head nodes rebroadcast the packet. The ENE packet contains the cell address of the entered cell C. Each node receiving this packet sets the RTT[C].occupied field in its routing tree table to true and sets the valid flag to false (indicating that the routing tree table has to be rebuilt).

b) *Non-Empty_to_Empty (NEE)* packet*:* when a cell-head leaves a cell C and there is no other node left in that cell, it broadcasts (cell-based broadcasting) a Non-Empty_to_Empty (NEE) packet. This packet contains the address of the cell C that has become empty. Any node recieving this packet sets the RTT[C].occupied field in its routing tree table to false and sets the valid flag to false (indicating that the routing tree table has to be rebuilt).

c) *EXIT* packet*:* when any node moves out of its current cell to a neighboring cell, it rebuilds its routing tree table and it transmits an EXIT packet to tell the neighbors about its new location. Each node in a neighboring cell receiving this packet updates its Neighbors Table (NT) accordingly.

d) *INFO* packet*:* when a cell-head node receives an EXIT packet from a node entering its cell from a neighboring cell, it replies by sending a unicast INFO packet to that node containing its NT table. The sender of the EXIT packet replaces its NT table with the received one.

e) *Sink Location Request (SLOCREQ)* packet*:* Every cell-head rebroadcasts this packet to neighboring cell heads. Sink Location Reply (SLOCREP) packet: when a sink node receives a SLOCREQ packet, it broadcasts (cell-based broadcasting) its location (cell address) using a SLOCREP packet to all nodes. Any node receiving this SLOCREP packet updates its Location Table accordingly.

B. Discovering the Location of a Sink Node

When a node wants to send data packets to a destination (sink) node and there is no information about its location (cell address) in the Location Table, then it broadcasts a Sink Location Request (SLOCREQ) packet. Only one node in each cell (the cell-head) participates in rebroadcasting the SLOCREQ. This mechanism is called cell-based broadcasting. The SLOCREQ packet carries the location (cell-address) of the previous node which is used to update the Location Table and the Neighbors Table. There are three possible cases for a node receiving the SLOCREQ:

Case 1: Sink Node. If it is a sink node, then it first records the cell location information of the previous hop node in the Neighbors Table (NT) and in the Location Table (LT) and then it broadcasts (using cell-based broadcasting) a Sink Location Reply (SLOCREP) packet. Once the source node receives the SLOCREP packet, it starts forwarding data packets to the sink node along the routing tree guided by the next hop fields of the routing tree tables.

Case 2: Cell-Head Node: If it is not a sink node but it is a cell-head node, then it first records the cell location information of the previous hop node in the NT and LT tables and then it re-broadcasts the SLOCREQ packet. If any node receives a previously processed SLOCREQ (based on a sequence number maintained in LT), it discards it.

Case 3: Non-Cell-Head Node. If it is not a sink node and it is not a cell-head, then it records the cell location information of the previous hop node in the Neighbors Table and in the Location Table and then discards the SLOCREQ packet.

C. Cell-Head Selection in TBR

In TBR, any sensor node A selects for each of its neighboring cells C, a node H_C^A located in that cell C (the information about which nodes are located in C is obtained from the Neighbors Table) to act as the cell-head responsible of relaying the packets that A needs to send across cell C. Node A selects H_C^A randomly among all sensor nodes located in C. This independent random selection allows to have multiple nodes in a given cell serving as cell-heads for different nodes in neighboring cells. This reduces the cell-head selection overhead and avoids overloading a particular node in C with all the packet-forwarding requests from neighboring cells.

D. Proactive and Reactive Layers of TBR

Any node joining the UWSN starts by determining its location (i.e., in which cell of the 3D Grid it is located). It continues monitoring its location periodically until it leaves the UWSN. TBR operates in two layers: a proactive layer and a reactive layer. The main function of the proactive layer is to maintain up-to-date the Routing Tree Table and the Neighbors Table. Fig. 4 outlines the proactive layer mechanisms.

The mechanisms of the reactive layer of TBR are activated when a node needs to send a pack and it has no information in its Location Table about the destination (sink node). These mechanisms allow to track the location of the sink node (the cell in which it is located) and to ensure this information is known by all nodes. Fig. 5 outlines the mechanisms of this reactive layer.

Empty to Non-Empty
If a node enters an empty cell C:
- Initiate cell-based broadcasting of an Empty-to-Non_Empty packet identifying cell C
- Any node receiving this packet performs the following:
  - set RTT[C].occupied to true
  - set the valid flag to false (indicating that the routing tree has to be rebuilt)

Non-Empty to Empty
If a node leaves a cell C and no other node is left in C:
- Initiate cell-based broadcasting of a Non-Empty-to-Empty packet identifying the concerned cell C
- Any node receiving this packet performs the following:
  - set RTT[C].occupied to false
  - set the valid flag to false (indicating that the routing tree has to be rebuilt)

Exit from Cell
- If a node leaves its current cell then it rebuilds its routing tree table and it sends an EXIT packet to all nodes in its transmission range
- When the cell-head of the newly entered cell receives the EXIT packet, it unicasts an INFO packet to the new node containing a copy of the Neighbors Table (NT). Upon receiving this INFO packet, the new node replaces its NT table with the received one

Build Tree:
- If a node has to route a packet and the valid flag is false, then it rebuilds its routing tree using the tree construction algorithm of Figure 3 and sets its valid flag to true

Fig. 4. The proactive layer mechanisms of TBR.

Reactive Layer Mechanisms of TBR:
- If a node has to send a packet and there is no information about the destination (sink) node in the Location Table, then it initiates cell-based broadcasting of a SLOCREQ packet
- If a cell-head receives a SLOCREQ packet (for the first time) then it retransmits the SLOCREQ packet
- If a non-cell-head and non-sink node receives a SLOCREQ then it records the cell location information of the previous hop node in the Neighbors Table and Location Table and then discards the packet
- If a sink node receives a SLOCREQ then it initiates cell-based broadcasting of a SLOCREP packet. Any node receiving this SLOCREP packet updates its Location Table accordingly

Fig. 5. The reactive layer mechanisms of TBR

V. Performance of Tree-Based Routing

In order to demonstrate the merits of the proposed approach of tree-based routing over a virtual grid, we present in this section simulation results we obtained in a previous study [14] on routing in mobile ad-hoc networks using this same approach. These results illustrate the attractiveness of this approach in terms of reducing communication overheads and increasing delivery ratios especially in dense networks.

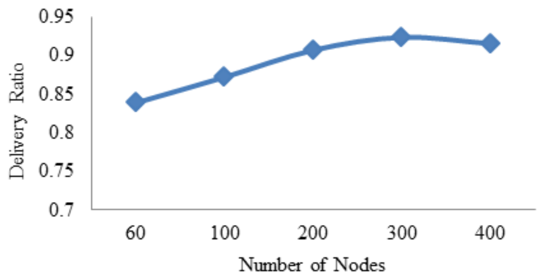

Fig. 6. Delivery ratio versus number of nodes

Fig. 6 shows how the delivery ratio (probability of packet delivery) increases as the number of nodes in the network increases. As the number of nodes increases, more and more grid cells become occupied and hence the probability of finding of a routing path (sequence of adjacent occupied cells) from source node to sink node also increases. When at least one such path exists, delivery of packets from source to destination will be successful.

Fig. 7 shows an inverse relationship between the number of sensor nodes and the normalized number of control packets (average number of control packets per transmitted data packet). The number of proactive layer control packets (Empty-to-Non-Empty and Non-Empty-to-Empty) decreases when the number of nodes increases and vice-versa. Increasing the number of nodes reduces the need to send these control packets because they are only sent when the status of grid cells changes from empty to non-empty or vice versa. Such changes are reduced when the number of nodes increases.

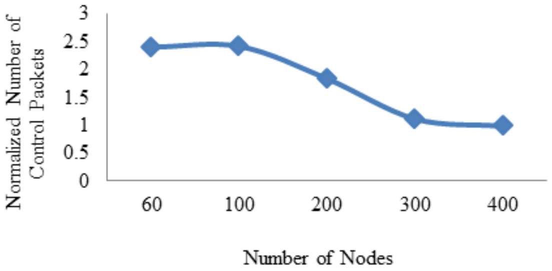

Fig. 7. Number of control packets versus number of nodes

VI. Conclusions

We have presented a Tree-Based Routing protocol for Underwater Wireless Sensor Networks called TBR. It is based on building shortest path trees spanning the occupied cells of a virtual 3D grid representing the underwater region. During routing, the paths to the destination (sink) nodes are readily obtained from the constructed shortest path trees. Due to the availability of these pre-constructed shortest path trees, the proposed TBR protocol offers higher stability of the routing paths compared to other protocols which rely on costly reactive routing path establishment and maintenance mechanisms.

Acknowledgment

This work was supported in part by a grant from The Research Council (TRC) of the Sultanate of Oman under the open research grant number RC/SCI/COMP/15/02.

References

[1] G. Han, C. Zhang, L. Shu, N. Sun, and Q. Li, "A survey on deployment algorithms in underwater acoustic sensor networks," *Int'l Journal of Distributed Sensor Networks*, vol. 2013, pp. 1–11, 2013.

[2] F. A. Salti, N. Alzeidi, and B. Arafeh, "EMGGR: An energy-efficient multipath grid-based geographic routing protocol for underwater wireless sensor networks," *Wireless Networks*, vol. 23, no. 4, pp. 1301-1314, May 2017.

[3] K. Chen, M. Ma, E. Cheng, F. Yuan, and W. Su, "A survey on MAC protocols for underwater wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1433-1447, 2014.

[4] G. Han, C. Zhang, L. Shu, and J. Rodrigues, "Impacts of Deployment strategies on localization performance in underwater acoustic sensor networks," *IEEE Trans. on Industrial Electron*ics, vol. 62, no. 3, pp. 1725–1733, March 2015.

[5] M. Boulaiche and L. Bouallouche-Medjkoune, "EGGR: Energy-aware and delivery guarantee geographic routing protocol," *Wireless Networks*, vol. 21, no. 6, pp. 1765–1774, August 2015.

[6] Z. H. Jiang, "Underwater acoustic networks-issues and solutions," *International Journal of Intelligent Control and Systems*, vol. 13, no. 3, pp.152–161, 2008.

[7] M. Ayaz, I. Baig, A. Abdullah, and I. Faye, "A survey on routing techniques in underwater wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1908–1927, Nov. 2011.

[8] S. Sarafiabadi, A. Berqia, and S. Parvaneh, "Survey of routing protocols in underwater WSNs for mine detection," in *Proc. 4th International Conference on Computer Modeling and Simulation*, Hong Kong, 2012, pp. 234–237.

[9] P. Xie, J. H. Cui, and L. Lao, "VBF: Vector-Based forwarding protocol for underwater sensor networks," in *Proc. of IFIP Networking'06*, Coimbra, Portugal, 2006, pp. 1216–1221.

[10] N. Nicolaou, A. See, P. Xie, J. H. Cui, and D. Maggiorini, "Improving the robustness of location-based routing for underwater sensor networks," in *Proc. OCEANS 2007-EUROPE*, Aberdeen, 2007, pp. 1–6.

[11] 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.

[12] J. M. Jornet, M. Stojanovic, and M. Zorzi, "Focused beam routing protocol for underwater acoustic networks," in *Proc. International Conference on Mobile Computing and Networking, ACM International Workshop on Underwater Networks*, San Francisco, 2008, pp. 75–82.

[13] 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," *Journal of Internet Technology*, vol. 11, no. 2, pp. 153–165, 2010.

[14] H. Almaqbali, K. Day, M. Ould-Khaoua, A. Touzene, and N. Alzeidi, "TGRP: A new hybrid grid-based routing approach for manets," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 7, pp. 601-609, 2016.

Khaled Day received his PhD degree in computer science from the University of Minnesota (USA) in 1992. He has joined Sultan Qaboos University in 1996 where he is currently holding a Professor position. His research interests include interconnection networks, parallel computing, distributed systems and wireless and mobile networks. He is a senior member of IEEE.

Nasser Alzeidi received his PhD degree in Computer Science from the University of Glasgow (UK) in 2007. He is currently an Assistant Professor of computer science and the Director of the Center for Information Systems at Sultan Qaboos University, Oman. 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.

Abderezak Touzene received the BS degree in Computer Science from the University of Algiers in 1987, M.Sc. degree in Computer Science from Orsay Paris-Sud University in 1988 and Ph.D. degree in Computer Science from Institute Polytechnique de Grenoble (France) in 1992. He is an Associate Professor in the Department of Computer Science at Sultan Qaboos University in Oman. 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.