CUMAC-CAM: Addressing Triple Hidden Terminal Problems for Multi-channel Transmission in Underwater Sensor Netw....

Conference Paper · November 2017

DOI: 10.1145/3148675.3148707

Mohammad Shah Alam
Bangladesh University of Engineering and Technology

CUMAC-CAM: Addressing Triple Hidden Terminal Problems for Multi-channel Transmission in Underwater Sensor Networks

Purobi Rahman
Institute of Information and Communication Technology
Dhaka, Bangladesh
purobi.rahman@gmail.com

Mohammad Shah Alam
Institute of Information and Communication Technology
Dhaka, Bangladesh
smalam@iict.buet.ac.bd

Shamim Ara Shawkat
Dept. of Electrical Engineering and Computer Science
Knoxville, TN, USA
mshawkat@vols.utk.edu

Mohammad A Hoque
Department of Computing
Johnson City, TN, USA
hoquem@etsu.edu

ABSTRACT

As multi-channel communication is becoming more and more common in Radio Frequency (RF) arena, acoustic communication protocols have also started to adopt the same concept to utilize multiple channels in Underwater Sensor Networks (UWSN). Although the deployment of multi-channel increases throughput significantly, it also opens up the possibility of collision occurrence due to the hidden terminal problem. In particular, the "Triple Hidden Terminal (THT)" problems, a phenomenon characterized by collision occurrence due to multi-hop, multi-channel communication with long propagation delay, persists more dominantly in UWSN. Existing MAC protocols try to mitigate the adverse effect of THT without utilizing the information of propagation delay that may be exploited to improve the performance of UWSN significantly. The current work proposes a Cooperative Underwater Multi-Channel MAC protocol with Channel Allocation Matrix (CUMAC-CAM). A new Channel Allocation Matrix (CAM) has been introduced for estimating propagation delay to ensure enhanced channel utilization. In this scheme, each node maintains a delay mapping database, based on which senders and receivers perform a scheduling algorithm before initiating any transmission. This mapping helps a node to predict whether it's upcoming packet transmission will collide with other nodes' transmission or not. In brief, the objective is to ensure successful transmission by mitigating triple hidden terminal problems in multi-channel underwater sensor networks as well as to enhance the channel utilization with the benefit of delay mapping and channel allocation assessment. Simulation results, carried out for performance analysis, show that the proposed MAC protocol is more efficient in terms of network throughput, energy consumption, end to end delay and packet delivery ratio compared to the contemporary CUMAC protocol.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

Underwater Sensor Networks; propagation delay; multi-channel hidden terminal problem; Channel Allocation Matrix (CAM); delay map; transmission scheduling; collision detection; channel assessment

ACM Reference Format:
Purobi Rahman, Mohammad Shah Alam, Shamim Ara Shawkat, and Mohammad A Hoque. 2017. CUMAC-CAM: Addressing Triple Hidden Terminal Problems for Multi-channel Transmission in Underwater Sensor Networks. In *WUWNET'17: WUWNET'17: International Conference on Underwater Networks & Systems, November 6–8, 2017, Halifax, NS, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3148675.3148707

1 INTRODUCTION

In recent times, there has been a great interest on the subject of UWSN. The extension of WSN to the UWSN has opened up new opportunities as these networks are supporting smart, reconfigurable and fault tolerant sensor nodes deployment.UWSN is projected to be implemented in numerous applications such as, environmental monitoring, underwater explorations, disaster prevention, assisted navigations and tactical surveillance.A range of studies have been conducted on MAC protocols in underwater network. Most of the studies focused on single-channel networks in underwater and acoustic communication protocols have started to utilize multiple channels in underwater sensor network for the last couple of years. The organization of the rest of this paper is as follows. Section 2 presents the related works, section 3 explains the Triple Hidden Terminal problems in UWSN, section 4 gives an overview of proposed protocol CUMAC-CAM. Then, performance of CUMAC-CAM has been investigated in section 5 in terms of performance parameters such as throughput, end to-end delay, energy consumption and packet delivery ratio. Finally, section 6 concludes the paper.

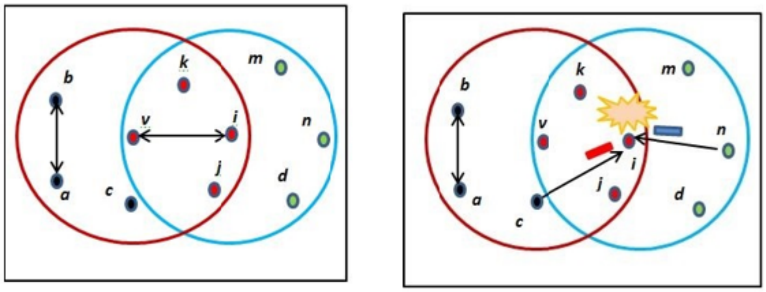
2 RELATED WORKS

Underwater acoustic channels are characterized with long propagation delay,low data rate and limited bandwidth [1–3, 7]. Furthermore, in underwater acoustic communication, transmission is almost 100 times more expensive compared to reception in terms

---

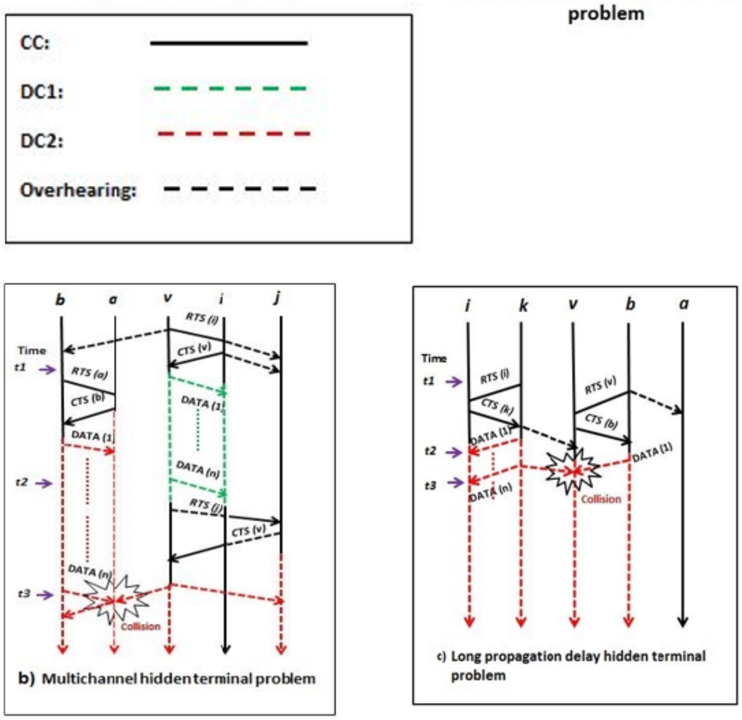
WUWNET'17, November 6–8, 2017, Halifax, NS, Canada

ACM ISBN 978-1-4503-5561-2/17/11…$15.00
https://doi.org/10.1145/3148675.3148707



Network Topology

a) Multi-hop hidden terminal problem



b) Multichannel hidden terminal problem

c) Long propagation delay hidden terminal problem

Figure 1: The Illustration of Triple Hidden Terminal (THT) problems

of energy consumption [7, 9]. To minimize the impact of long propagation delay and limited bandwidth, a Delay-aware Opportunistic Transmission Scheduling (DOTS) MAC protocol [6] is proposed for underwater sensor networks. DOTS exploits propagation delay and provides an efficient mechanism to support the concurrent transmission over a single channel with preventing the possibility of collisions. Moreover, multichannel transmission in UWSN also compensates the adverse effect of long propagation delay and low data rate. However, utilizing multichannel in underwater sensor networks, which suffers long propagation delay introduce hidden terminal problem more dominantly [11]. To mitigate the triple hidden terminal problems, CUMAC is proposed in [12] where collision detection scheme is employed with a simple tone device by utilizing the cooperation of neighbor nodes. Another multichannel MAC protocol, named as UMMAC, based on slot reservation is proposed in [8] where fixed length of slot duration makes it impossible to avoid collision.

3 TRIPLE HIDDEN TERMINAL PROBLEMS

Researchers came up with a CUMAC protocol, focusing on Triple Hidden Terminal (THT) problems in underwater sensor networks [12]. THT problems include three kinds of hidden terminals in underwater sensor networks and these are: a) multi-hop hidden terminal problem which is the traditional hidden terminal problems in multi-hop networks; b) multi-channel hidden terminal and c) long-delay hidden terminal problem. An illustration of THT is given in Figure 1 (Figure 1 adopted from [5]).

A sample network topology is presented here which involves number of sensor nodes (*a, b, c, d, v, k, i, j, m, n*), one control channel

(CC) and two data channels (DCs). The effect of traditional multi-hop hidden terminal problem is depicted in the Figure 1(a). Then, from the Figure 1(b) it is observed that, when node v has a data for i then it puts the possible data channels and reservation information to RTS and send to *i* on the CC. After RTS/CTS handshaking, both nodes switch to the selected data channel (DC1) around the time t1 and carry out the data transmission. During the period (t1, t2), *b* has a data for *a*. Next, both nodes (*b, a*) switch to another idle channel (DC2) after reservation. As *v* and *i* are not overhearing on CC over the period (t1, t2), *v* and *i* still assume that DC2 is idle. Around the time t3, one situation causes packet collisions at a or b. When *v* finishes sending data to *i* and *v* has data for *j*. Now, if *v* also selects DC2 channel, that node *a* and *b* are still occupying, then a collision happens and this collision causes due to the effect of multi-channel hidden terminal problem. Similarly, long-delay hidden terminal problem is depicted in the Figure 1(c). As shown in the Figure, node *k* starts handshaking process with node *i* and then selects channel DC2 for communication. Later, node *b* and *v* also negotiate on the CC for their transmission. Let assume that, CTS message of node k arrives at node v after it selects its own data channel (DC2) and send CTS message back to *b*. In this case node *v* does not know that the same channel that is DC2 is already occupied by node *k* and thus create a collision.

4 PROPOSED SOLUTION

In this research work we have focused on the enhancement of earlier work on CUMAC with incorporation of propagation delay mapping and channel allocation assessment.

4.1 System Model

For simplicity, it is assumed that, underwater sensor nodes in the proposed CUMAC-CAM protocol are static and nodes are uniformly distributed within a fixed area. As per assumption of CUMAC, there is one control channel and multiple data channels with equal bandwidth. If there is no data to send or receive then every node listens to the common control channel. Each node has only one acoustic transceiver and a node can work either on control channel or data channel but not on both at a time.It is assumed that, every node knows its own position information by some localization algorithms [4, 10]. In addition, the enhancement of the CUMAC-CAM demands that, every node in the network will maintain a Channel Allocation Matrix (CAM) and delay map database.

4.2 Protocol Description

The key methods of CUMAC-CAM: CAM and Propagation Delay Map Database, Cooperative Update on Channel Allocation and Transmission Scheduling with Collision Detection and Channel Assessment are presented in this section followed by a discussion how CUMAC-CAM resolves the triple hidden terminal problems of multichannel underwater sensor networks.

4.2.1 *Channel Allocation Matrix (CAM) and Propagation Delay Map Database:*

In CUMAC-CAM, all nodes maintain Channel Allocation Matrix (CAM) and delay map information as shown in Figure 2. Channel Allocation Matrix (CAM) is used to keep the details of data channel allocation, which must contain the information of occupied channel, source and intended receiver for which the channel is observed as reserved, timestamp the time at which the MAC frame is sent and finally, the transmission time duration for the MAC frame. On the other hand, each node maintains delay map by passively overhearing neighbor nodes transmissions. The delay map database consists of source and destination information of the observed MAC frame and the estimated propagation delay between the source and the destination. In this context, the proposed CUMAC-CAM protocol makes the assumption of time synchronization among all nodes in the network as [6], in order to precisely estimate the transmission delay between nodes with the measurement of propagation delay.

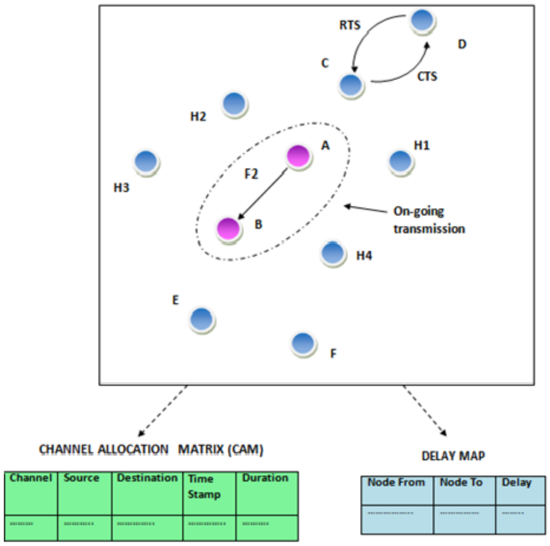


Figure 2: Nodes are maintaining Channel Allocation Matrix (CAM) and delay map database by overhearing neighbor nodes transmission

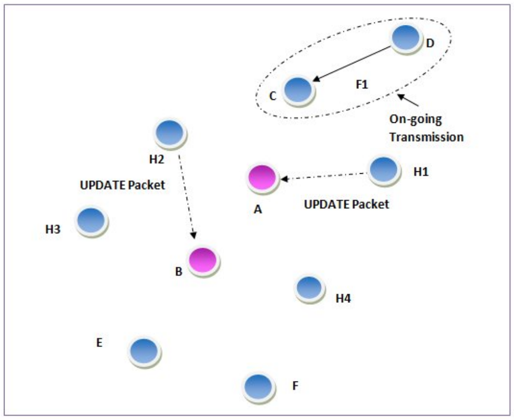


Figure 3: Update packets sent by neighbor nodes

4.2.2 *Cooperative Update on Channel Allocation:*

Cooperative Update on Channel Allocation scheme is introduced to address the multi-channel hidden problem in underwater sensor network. Figure 3 depicts that, node A and node B were not aware of handshaking and upcoming transmission on F1 due to multi-channel hidden terminal problem. But neighbor node of node A and B have the handshaking information between node D and node C. In addition, neighbor nodes also know that when the on-going transmission between node A and node B is going to complete. Accordingly, after completion of each on-going transmission, neighbor nodes act as a helper node by cooperating communication pairs with UPDATE control packets. The objective is to provide updated information on channel allocation status to alleviate the multi-channel hidden terminal problem in underwater sensor networks. Furthermore, in order to mitigate UPDATE packet collision in the event of cooperative update on channel allocation, neighbor nodes send UPDATE packets through random back-off algorithm.

4.2.3 ***Transmission Scheduling with Collision Detection and Channel Assessment:*** This section presents the details scheme of free channel assessment, sender and receiver end collision detection along with the implementation of transmission scheduling algorithm (Algorithm 1).

Algorithm 1 TRANSMISSION SCHEDULING ALGORITHM BASED ON DELAY MAP

```
/*Transmission scheduling with sender end collision detection*/
for all nodes ∈ delay map do
    if frame type is DATA then
        DATA arrival time at neighbor node <- timestamp + trans.time+(2*prop.ctrl.delay)+prop.data.delay
        if DATA transmission time at sender node ∈ DATA arrival time at neighbor node then
            return collision detected
        end if
    end if
end for
/*Transmission scheduling with receiver end collision detection*/
For all nodes ∈ delay map do
    if frame type is DATA then
        DATA arrival time at receiver node <- timestamp + trans.time+(2*prop.ctrl.delay)+prop.data.delay
        if DATA arrival time at receiver node ∈ DATA transmission time at neighbor node then
            return collision detected
        end if
    end if
end for
```

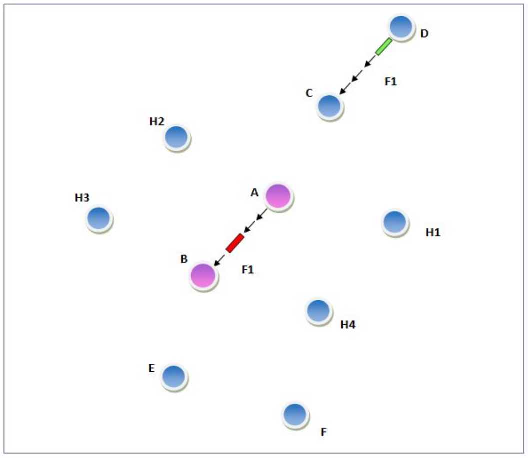


Figure 4: Collision Detection at sender end- Node A transmits a packet to node B over the channel F1 before its one hop neighbor node C receives a packet from node D over the same channel

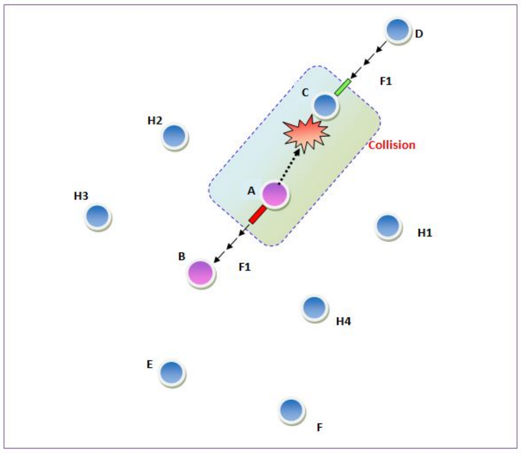


Figure 5: Collision Detection at sender end- Collision occurs if node A transmit a packet to node B over the channel F1 without considering collision at one-hop neighbor node C′s packet reception on the same channel

Free Channel Assessment: If any data channel is found free by verifying the CAM that is not occupied by any other nodes, then the sender node will initiate RTS/CTS handshaking process over the control channel. The goal of this handshaking is to take initiative of upcoming packet transmission over the free data channel. Thus the free channel assessment ease the process of getting data channel status and select a data channel by node itself with the help of CAM.

Collision Detection at Sender End: In case of unavailable free data channel, at first sender node checks the occupied channels information from CAM. Then it will run the transmission scheduling algorithm aligning with delay map database. It will guide the sender node to take a decision to move forward for upcoming packet transmission without hampering its one hop neighbor nodes packet reception over the desired channel, which is occupied by that one hop neighbor node.

Considering the Figure 4 and Figure 5 (Figure 5 adopted from [12]) as an example, a transmission is going on over a data channel F1 and the channel is occupied by a communication pair node D (sender) and node C (receiver). Node C is one-hop neighbor node of node A. Suppose, node A wants to transmit a packet to node B. In this scenario, node A will move forward for data transmission over the data channel F1 [Figure 4], if it get assurance from transmission scheduling that, node A must be able to transmit data packet before its one hop neighbor node C receives the transmitted packet from node D otherwise collision will occur as depicted in the Figure 5. In short, a node will move forward for data transmission over any occupied channel if it estimates that its upcoming transmission will not hamper or collide with one-hop neighbor nodes packet reception.

Collision Detection at Receiver End: After getting the RTS, receiver will also check the CAM and runs transmission scheduling algorithm based on the delay map. Taking the Figure 6 and Figure 7 (Figure 7 adopted from [12]) as an example, where a transmission is continuing over the data channel F1 between the communication pair node E (sender) and node F (receiver). E (sender) is a one-hop neighbor node of node B. In this scenario, say, node B receives a RTS packet from node A for the upcoming transmission over the same channel F1. As per node B also estimates whether or not its packet reception from node A will be interfered with one hop neighbor node Es ongoing packet transmission. Node B will reply with CTS if and only if it predicts by running transmission scheduling algorithm that it will receive the incoming packet from node A after its one hop neighbor node E completes transmitting packets to the intended receiver, node F [Figure 6]. Or else, collision will occur as shown in the Figure 7. In addition, before replying CTS node B will wait up to maximum propagation delay according to its delay map database. The purpose is to alleviate collisions, occurred for long delay hidden terminal problem in underwater sensor networks.

Subsequently, when receiver node replies with its control message CTS to the sender, both nodes switch to the desired data channel. The receiver node starts a timer and keeps waiting for the incoming packet. If it does not receive any packet before the time out it switch back to the control channel, updates its CAM accordingly and broadcast a CANCEL control message to cancel the channel reservation. On the other hand, if sender node does not receive any CTS after sending its RTS packet then after random back-off period sender node retransmit the RTS packet to initiate the transmission by following the proposed scheme.

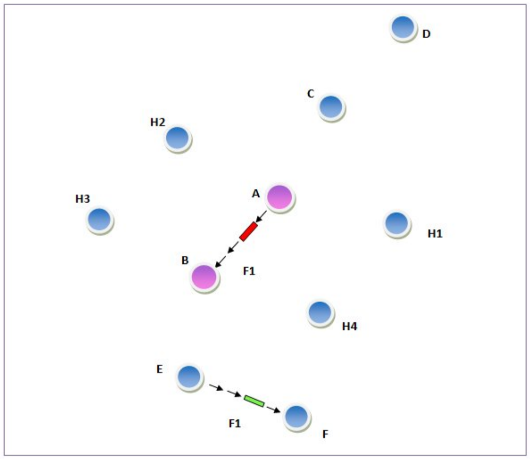


Figure 6: Collision detection at receiver end- Node B will receive packet from node A on F1 channel after node B′s one hop neighbor node E transmits a packet to its intended receiver - node F over the same channel

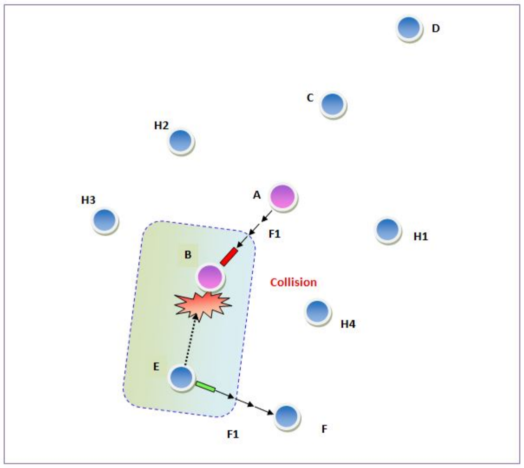


Figure 7: Collision detection at receiver end- Collision occurs if node B receives a packet over the channel F1 while its one hop neighbor node transmitting a packet to node F over the same channel

4.2.4 ***Discussion:*** As per reference of CUMAC and research work in literature, the traditional multi-hop hidden terminal problem can be easily alleviated by RTS/CTS handshaking process. Therefore, the major challenge is to handle multi-channel and long delay hidden terminal problems in underwater sensor networks.

In CUMAC-CAM, CAM has been introduced and each node maintains CAM and delay map by passively overhearing the ongoing transmission of neighbor nodes. Cooperative update on channel allocation scheme effectively addresses the multichannel hidden terminal problem in underwater sensor networks. As this scheme helps the communication pair to get the update channel allocation information. Moreover, each node evaluates the channel and run transmission scheduling algorithm mapping with delay map to have collision free transmission at both sender and receiver end. Additionally, receiver node waits up to maximum propagation delay before replying CTS to its respective sender to proceed for data transmission on the selected data channel. Hence, before the waiting time out, the receiver node may have a chance to get the channel status from the long delayed neighbor nodes for which the same data channel is already occupied. Thus CUMAC-CAM efficiently mitigates the triple hidden terminal problems in underwater sensor networks.

5 PERFORMANCE EVALUATION

In this section, we evaluate the performance of CUMAC-CAM protocol and compare it with CUMAC. We compare these two protocols with following metrics: i) Average network throughput ii) Average energy consumption iii) End to end delay iv) Packet delivery ratio (PDR)

To evaluate the performance of these two MAC protocols, we implement CUMAC and CUMAC-CAM protocol in Aqua-sim simulator which is an NS-2 based simulator for Underwater Sensor Networks.In this set of simulations, random network is examined. CUMAC-CAM protocol is implemented in a random network where maximum 20 static nodes are uniformly distributed in 500m X 500m area.

Table 1: System Parameters

Parameters	Value
Transmission range of every node	100 m
Maximum number of channels	8
Acoustic Propagation Speed	1500m/s
Maximum data packet length	600 bytes
Control packet length	32 bytes
Data rate	1 kbps
Transmitting power	0.6 Watt
Receiving power	0.2 Watt
Idle listening power	0.02 Watt

5.1 Average Network Throughput

Average network throughput can be defined as average number of successfully transmitted data bytes per second.

$$Average\ Network\ Throughput = \frac{\text{Average of total transmitted data}}{\text{Network operation time}} \tag{1}$$

5.1.1 Impact of input traffic: As depicted in Figure 8, for the both protocols CUMAC and CUMAC-CAM, the throughput upturns significantly with the input traffic. The throughput of CUMAC-CAM is about 175 bytes per second whereas CUMAC can achieve its maximal throughput 165 bytes per second when input traffic is 0.04 packets per second. As Figure 8 shows that, the throughput of CUMAC-CAM protocol reaches to the highest peak 185 bytes per second when input traffic is 0.05 packets per second. But after that, its throughput decreases slowly but provides quite steady performance than the CUMAC.

5.1.2 Impact of number of channels: In this set of simulations, we evaluate the performance of CUMAC and CUMAC-CAM protocol with varying number of channels. Input traffic of every node is fixed to 0.02 packets per second and number of channels is varied from 2 to 8 channels. Figure 9 represents that, for both protocols network throughput improves with the number of channels. As net input traffic to every data channels in the network decreased with

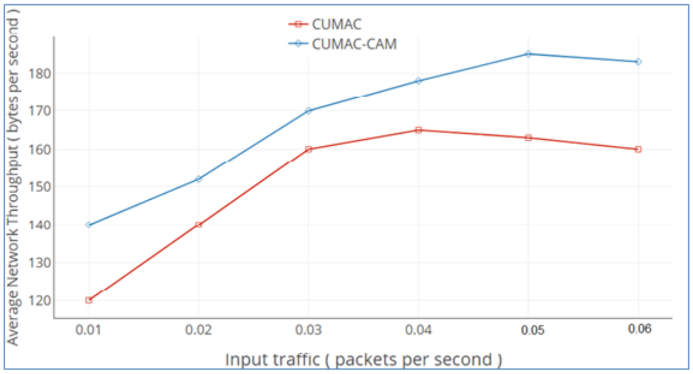


Figure 8: Performance comparison of CUMAC-CAM with CUMAC in terms of impact of input traffic on average network throughput

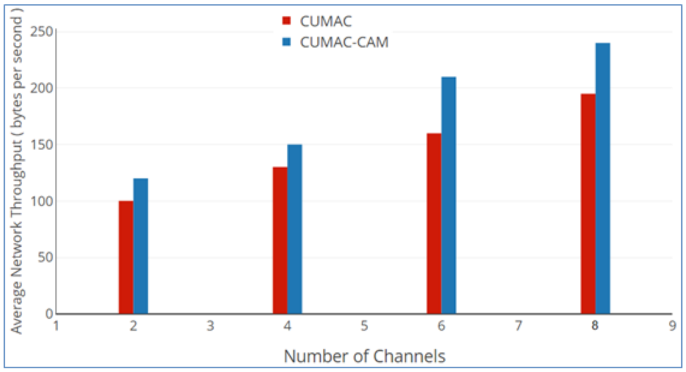


Figure 9: Performance comparison of CUMAC-CAM with CUMAC in terms of impact of no. of channels on average network throughput

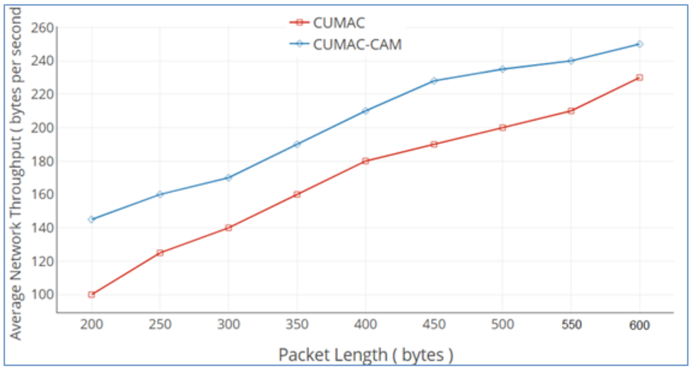


Figure 10: Performance comparison of CUMAC-CAM with CUMAC in terms of impact of packet length on average network throughput

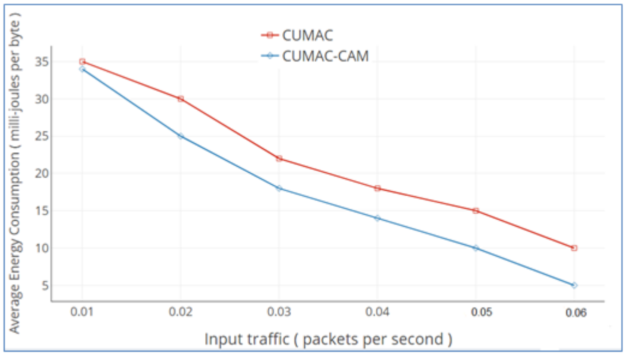


Figure 11: Performance comparison of CUMAC-CAM with CUMAC in terms of impact of input traffic on average energy consumption

the number of channels. Moreover, Figure 9 reflects that, CUMAC-CAM protocol is more efficient compared to CUMAC. This is because, channel allocation assessment, channel utilization, collision detection along with transmission scheduling algorithm scheme of CUMAC-CAM results less collision probability, efficient channel allocation and finally provides higher throughput.

5.1.3 Impact of data packet length: For the performance comparison in terms of varying packet length, input traffic is set to 0.02 packets per second and data packet length is changed from 200 to 600 bytes. As shown in Figure 10, our proposed CUMAC-CAM protocol also achieve higher throughput compared to CUMAC. As longer data packet may incur higher collision probability but CUMAC-CAM offers less collision. Therefore, CUMAC-CAM protocol provides better performance compared to CUMAC.

5.2 Average Energy Consumption

Average energy consumption is obtained by dividing the overall energy consumption in the network by the successful transmitted data bytes. It is measured by milli-joules per byte.

$$E_{avg} = \frac{E_{consumption}}{\text{Total transmitted data}} \tag{2}$$

Here, E_{avg} is the average energy consumption per byte and $E_{consumption}$ is the total energy consumption in the network.

5.2.1 Impact of input traffic: Figure 11 plots that, the average energy consumption per byte decreases with the increase of input traffic for the both two protocols. As it is seen from the performance evaluation, CUMAC-CAM achieves higher energy efficiency than the CUMAC protocol. CUMAC implements cooperative collision detection and tone pulse sequence to suppress the triple hidden terminal problems of underwater sensor networks. On the other hand, the channel allocation matrix along with propagation delay map database and transmission scheduling with collision detection and channel assessment mechanism of CUMAC-CAM defeats the triple hidden problems efficiently. As a consequence, CUMAC-CAM provides better results compared to CUMAC.

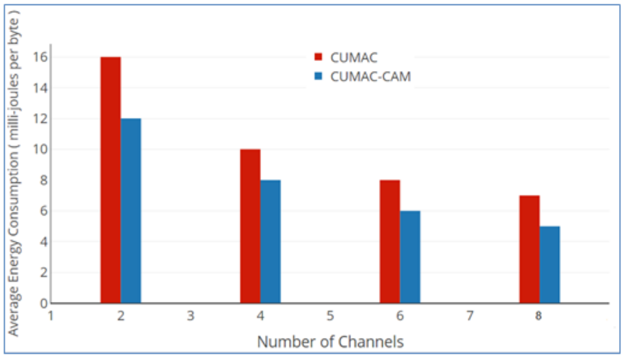


Figure 12: Performance comparison of CUMAC-CAM with CUMAC in terms of impact of no. of channels on average energy consumption

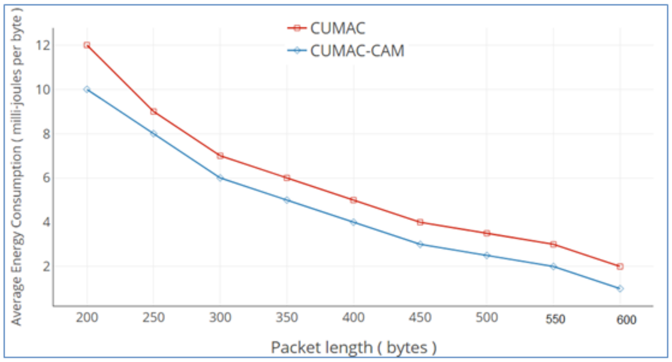


Figure 13: Performance comparison of CUMAC-CAM with CUMAC in terms of impact of packet length on average energy consumption

5.2.2 Impact of number of channels: In this set of simulation, input traffic is set to 0.02 packets per second and number of channels is changed to 2, 4, 6, and 8. Figure 12 shows that, CUMAC-CAM can attain much higher energy efficiency than CUMAC. In view of the performance comparison, when the number of data channels is 2, average energy consumption for CUMAC is about 16 milli-joules per byte and 12 milli-joules per byte for CUMAC-CAM. If there are maximum 8 data channels in the network then average energy consumption for CUMAC reduces to 7 milli-joules per byte and for CUMAC-CAM it stands around 5 milli-joules per byte. This is because, with association of number of data channels, collision probability reduces and impact emulates in the network performance.

5.2.3 Impact of data packet length: It is also observed from the Figure 13 that, the average energy consumption reduces with the increase of data packet length. Therefore, it reflects that, the network will have higher energy efficiency with the longer data packet length. As per the simulation result, CUMAC-CAM achieves higher energy efficiency in comparison of CUMAC protocol.

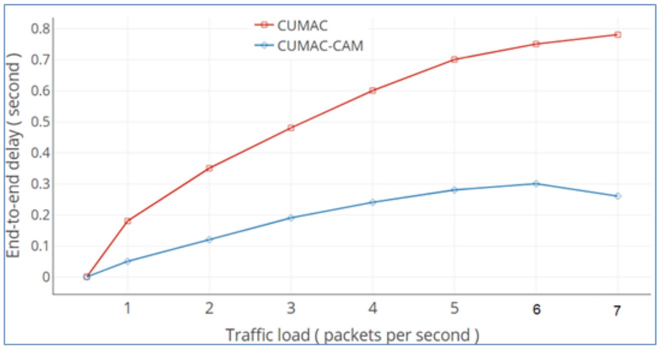


Figure 14: Performance comparison of CUMAC-CAM with CUMAC in terms of impact of traffic load on end-to-end delay

5.3 End to End Delay

The end-to-end delay signifies the average time taken by each packet to reach from source to destination. It comprises of all the various delays experienced during the trip from sender to receiver.

$$End\_to\_End\_Delay = \frac{\sum_{n=1}^{N}(R_n - S_n)}{\mathrm{N}} \qquad (3)$$

Here,
S_n= Time at which nth packet is sent; R_n= Time at which nth packet is received; N=Number of packets received

Simulation result represents that [Figure 14], CUMAC-CAM achieves a lower end-to-end delay over CUMAC protocol. As the graph shows, for the both protocols delay upturns gradually. The reason is, at first node gets free data channel to transmit data to the intended receiver. Then data channels started to be occupied and after a certain period of time it releases again. Therefore, the curve gradually upturns with the increase of data channel access contention and traffic load.
Compared with CUMAC, the improvement of CUMAC-CAM comes from an aspect.That is CUMAC-CAM allows concurrent transmission by channel allocation assessment mapping with propagation delay and provides better output for delay performance.

5.4 Packet Delivery Ratio (PDR)

The packet delivery ratio is the ratio of successfully delivered packets at the destination to the packets generated by the source. It can be represented as:

$$PDR = \frac{\text{Number of received packets}}{\text{Number of generated packets}} \times 100 \qquad (4)$$

TABLE 2 shows that, packet delivery ratio degrades when number of nodes is increased. CUMAC-CAM also provides better performance than the CUMAC protocol in terms of packet delivery ratio.

Therefore, evaluating all the performance metrics we can conclude that CUMAC-CAM offers significant improvement over the CUMAC protocol.

Table 2: Performance comparison in terms of Packet Delivery Ratio

Number of Nodes	CUMAC PDR %	CUMAC-CAM PDR %
5	87.5714	91.7143
10	87.5620	91.0014
20	86.0012	90.0004

6 CONCLUSIONS

Our proposed protocol CUMAC-CAM provides a solution to triple hidden terminal problems providing an efficient resolution of collision detection and channel selection. With the benefit of propagation delay, CUMAC-CAM focuses on increasing the chances of concurrent transmission while preventing the likelihood of collisions. Results from performance analysis show that CUMAC-CAM is more efficient in terms of energy consumption, network throughput, end to end delay and packet delivery ratio compared to the contemporary CUMAC protocol.

ACKNOWLEDGMENT

The authors would like to thank Amit Karmaker of IICT, BUET for his contribution in this research work.

REFERENCES

[1] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia. 2004. Challenges for efficient communication in underwater acoustic sensor networks. *ACM Sigbed Review* 1, 2 (2004), 3–8.
[2] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia. 2005. Underwater acoustic sensor networks: research challenges. *Ad hoc networks* 3, 3 (2005), 257–279.
[3] Keyu Chen, Maode Ma, En Cheng, Fei Yuan, and Wei Su. 2014. A survey on MAC protocols for underwater wireless sensor networks. *IEEE Communications Surveys & Tutorials* 16, 3 (2014), 1433–1447.
[4] Guangjie Han, Jinfang Jiang, Lei Shu, Yongjun Xu, and Feng Wang. 2012. Localization algorithms of underwater wireless sensor networks: A survey. *Sensors* 12, 2 (2012), 2026–2061.
[5] Ramiro Liscano, Guanlilng Chen, Daqing Zhang, and Xingshe Zhou. 2010. *Ubiquitous Intelligence and Computing*. Springer.
[6] Youngtae Noh, Uichin Lee, Seongwon Han, Paul Wang, Dustin Torres, Jinwhan Kim, and Mario Gerla. 2014. DOTS: A propagation delay-aware opportunistic MAC protocol for mobile underwater networks. *IEEE Transactions on Mobile Computing* 13, 4 (2014), 766–782.
[7] Jim Partan, Jim Kurose, and Brian Neil Levine. 2007. A survey of practical issues in underwater networks. *ACM SIGMOBILE Mobile Computing and Communications Review* 11, 4 (2007), 23–33.
[8] Yishan Su and Zhigang Jin. 2016. UMMAC: A multi-channel MAC protocol for underwater acoustic networks. *Journal of communications and networks* 18, 1 (2016), 75–83.
[9] Affan A Syed, Wei Ye, and John Heidemann. 2008. T-Lohi: A new class of MAC protocols for underwater acoustic sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 231–235.
[10] Peng Xie, Jun-Hong Cui, and Li Lao. 2006. VBF: vector-based forwarding protocol for underwater sensor networks. In *International Conference on Research in Networking*. Springer, 1216–1221.
[11] Yang Yu, Jie Shi, and Ke He. 2014. A multichannel MAC protocol for underwater acoustic sensor networks. In *Proceedings of the International Conference on Underwater Networks & Systems*. ACM, 2.
[12] Zhong Zhou, Zheng Peng, Jun-Hong Cui, and Zaihan Jiang. 2012. Handling triple hidden terminal problems for multichannel MAC in long-delay underwater sensor networks. *IEEE Transactions on Mobile Computing* 11, 1 (2012), 139–154.