BMF-MAC: A Bidirectional Multi-flow MAC Protocol for Multihop Underwater Acoustic Sensor Networks

Jenifar Rahman
Institute of Information and Communication Technology
Dhaka, Bangladesh
jenifar_shithi@yahoo.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

Due to the atypical characteristics of the physical media in underwater acoustic sensor networks (UW-ASN)–mostly because of long propagation delay, low bandwidth and high error rate–several challenges arise while designing the MAC protocol. In this paper, we propose a Bidirectional Multi-flow MAC protocol (BMF-MAC) for UW-ASN, to efficiently handle multi-hop multi-flow traffic load patterns such a way that multiple streams of data transmissions concurrently proceed while adapting with varying traffic condition. BMF-MAC supports constitution of multi-hop flows by considering all pending packets in routing layer buffer and all flow setups requests from neighbors when setting up a flow, contrary to other underwater MAC protocols. The proposed MAC introduces a data transmission technique using the bidirectional multi-flow packet method for sending multiple data packets of the same flow in the reverse direction and thus improve channel utilization. The protocol is aimed to schedule more data transmission over multiple multi-hop flows, thus permitting rapid distribution of data and reduction of latency. Results show that BMF-MAC protocol outperforms existing CMRT protocol in terms of network throughput and packet delivery latency.

KEYWORDS

Medium Access Control Protocols; Under Water Acoustic Sensor Network (UW-ASN); Handshake based MAC

ACM Reference Format:
Jenifar Rahman, Mohammad Shah Alam, Shamim Ara Shawkat, and Mohammad A Hoque. 2017. BMF-MAC: A Bidirectional Multi-flow MAC Protocol for Multihop Underwater Acoustic 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/10.1145/3148675.3148708

1 INTRODUCTION

Underwater acoustic sensor network (UW-ASN) provides a diverse set of applications for vehicles and vessels navigating below the surface of the water. For instance, disaster recovery, military surveillance, environmental monitoring, and resource investigation are some of the widely used applications worth mentioning [1]. The speed of sound under water is nearly five orders of magnitude lower than a radio signal 's propagation speed [2]. Furthermore, in underwater acoustic communication, transmission is almost 100 times more expensive compared to reception in terms of energy consumption [3]. Therefore, the physical media used in acoustic networks are characterized with long propagation delay, low data rate and high packet loss [4]. In order to mitigate the adverse effects of the above mentioned limitations, significant amount of research efforts have been directed towards improving the efficiency and channel utilization of existing MAC protocols by reducing handshaking significantly.

2 RELATED WORK

The slotted floor acquisition multiple access (Slotted-FAMA) protocol, one of the pioneer MAC schemes, combines both carrier sensing and handshaking mechanisms that prevents collisions; however, the throughput performance is significantly reduced by the excessive length of the slot [5]. Bidirectional concurrent MAC (BiC-MAC) protocol [6] improves the channel utilization by transmitting data packets to a sender-receiver pair simultaneously. Therefore, data transmission only happened between one sender and receiver. Whereas, in MACA-MN [7], the communication with multiple neighbors to request for data transmission is established by the sender in each successful handshake. In [8], the author introduces the Reverse opportunistic packet appending (ROPA) is a hybrid of MAC protocol which can be initialized by both sender and receiver. ROPA enables a sender to coordinate multiple neighbors by opportunistically transmitting their data packets which improves channel utilization. However, ROPA faces more collisions as more control packet exchange is needed. In multi receiver MAC (MR-MAC) [9] protocol more than two nodes can communicate in one handshake holding by a main receiver. The protocol schedules the packet transmission time by sending the data packet in a packet train manner thus the receiver can receive data packet without collision. But the

---

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

ACM ISBN 978-1-4503-5561-2/17/11…$15.00
https://doi.org/10.1145/3148675.3148708

protocol is too complicated to effectively improve network throughput and need too much control packets, which will influence the network performance. Additionally, reserving multi-hop channels at once in cascading manner, Cascading multi-hop reservation and transmission (CMRT) protocol [2] delivers the data packets to the destination by relaying data packets progressively. Furthermore, CMRT adopts a method based on packet train model in order to improve utilization of channel [7]. However, CMRT is based on single packet flow setup. It cannot support multiple flow construction from a single node and transmit data simultaneously over multiple streams, thus poorly responding to heavy traffic loads. Inspired by the problems mentioned above, we propose the design and evaluation of a low latency medium access control protocol, Bidirectional Multi-Flow MAC protocol (BMF-MAC) for UW-ASNs aimed to minimizing end-to-end delay while maximizing throughput in varying traffic conditions compared to existing handshake based MAC protocol for UW-ASNs, CMRT.

3 BMF-MAC PROTOCOL DESCRIPTION

In this paper we recommend a multi-flow MAC protocol in static underwater sensor networks. We consider that every node is equipped with an omni-directional half-duplex acoustic modem. It is assumed that, nodes estimate the propagation delay using information obtained from their two-hop neighbors. While the network is initialized, the distance between nodes are calculated with the help of control packets that measure round-trip time (RTT) or by information sharing between neighboring nodes [2]. Moreover, we consider all nodes have the routing tables which aid to relay through multi-hop nodes.

3.1 Flow Setup Communication

BMF-MAC is designed to schedule multiple bidirectional packets over multiple flows, each of which requires multi-hop communication with per round channel reservation similar to [10]. During scheduling of allocation of channels, the proposed MAC protocol takes into account all outstanding packets in routing layer while considering all pending requests for flow setup. Therefore, the MAC layer detects the variations of traffic load as well as the reserve time for packet transmission. In BMF-MAC, nodes can setup communication by means of multi-flow setup packets (MFP) which is uniquely structured to efficiently addresses multiple destinations. MFP is a series of control frames, across multiple hops through multiple flows. An MFP serves both as an RTS and as a CTS packet to the destination and source nodes, respectively, similar to [10]. Furthermore, some cross-layer information like the address of the final destination and the hop count for the current flow are enclosed in MFP as well. The network layer is responsible to pass down this final destination address. Moreover, MFP includes the number of data packets to be transmitted, batch size B_{size}, the busy duration of Node S, $d_{busy,S}$ like [2]. Furthermore, it has the ability to address several destinations i.e. one MFP can operate as an RTS up to k different destinations. Additionally, the number of flows, flow ID and reverse flag are enclosed in MFP packet.

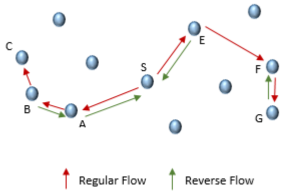

Figure 1: Topology

3.2 Foundation of States

In BMF protocol, a node shifts between fourteen different states. Six states are defined in the same way as CMRT protocol which are described in [2]. Here we introduce a state named *Tx_MFP* where a node sends requested MFP packet to relay node. Furthermore, *Rcv_MFP* state is defined where a node receives confirmed MFP from receiver node. *Wait_Flow* state is introduced where a receiver waits for forward data transmission to be finished over a flow. The receiver enters the Wait_Flow state directly after transmitting batch data packets and stays there until it starts transmitting CTS for reverse data transmission to be received.

A sender waits for CTS from a receiver in state *Reverse_Wait_Resp*. After forwarding data, if a sender finds its reverse flag is set, it waits to start its data transmission of the same flow over reverse direction. Another state *Reverse_Data* is presented where a sender delays data transmission for avoiding possible collisions caused by the bidirectional data transmission. Meanwhile sender receives a CTS from the receiver, the sender enters the *Reverse_Data* state and remains there until it finishes the transmission of reverse data packets. Moreover, a state called *Reverse_Rx* is designed where a receiver receives data packets over reverse flow direction.

Finally, two states *Tx_Retry* and *Wait_Retry* are defined to handle any missing control packet transmission. While a receiver node does not find data packet from its corresponding sender, it transmits another control packet named Retry Packet in *Tx_Retry* state. If any sender node misses its confirm MFP, it waits for Retry Packet(RP) from the receiver node in *Wait_Retry* state.

3.3 Protocol Description

Figure 1 and 2 illustrate the topology and the operation of the BMF protocol correspondingly. In Figure 1 it is assumed that node S denote a sender. Suppose node S has batch packets for destination nodes C and G. Node S sends packets to destination node C through node A and B and destination node G through E and F respectively. Relaying process of multi-flow establishes when the source node S starts transmitting MFP to relay nodes E and A simultaneously. After transmitting MFP, sender S enters in *Wait_Resp* state like protocol [2]. Node S starts the handshake by transmitting MFP to node E and A with the first destination address set to node E, and the second destination address set to node A. The scheduling of addresses is depended on node distances. Short distant node will be prioritized first.

The first destination node E, sends an MFP to relay node F in the forward direction, node S overhears MFP in the backward direction. At the same time, the second destination, node A, sends an MFP packet to relay node B in the forward direction, Node S overhears MFP in the backward direction. Utilizing the propagation delay between nodes due to distances, these two flows can take place

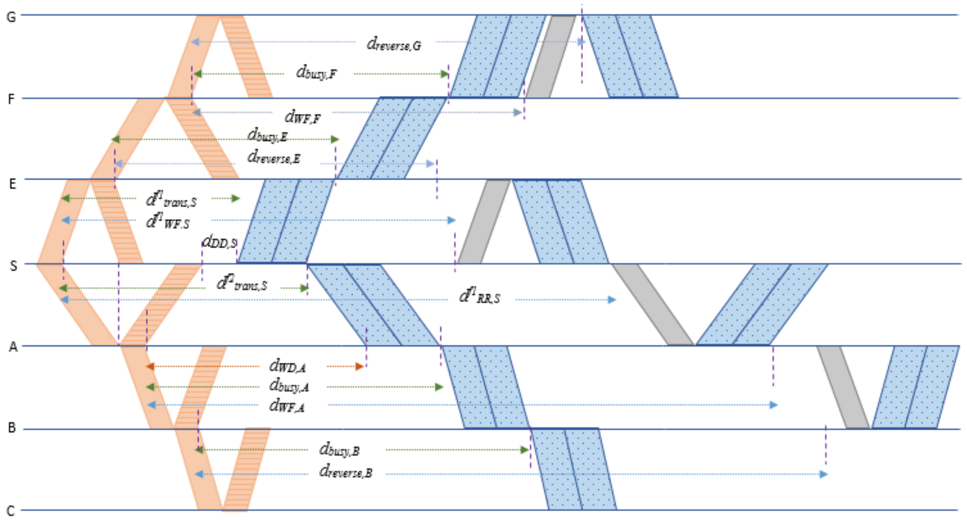

Figure 2: Operation of the bidirectional multi-flow MAC protocol

simultaneously in our proposed protocol. After relaying MFP, node S enters in *Delay_Data* state. Node A and E go to state $Wait\_Data$. Upon receiving S's MFP, A performs the same steps as E.

Concurrently receiving E's MFP, F performs the same actions as E. This process of receiving an MFP and immediately transmitting another MFP continues until either the final destinations node G and C have received the MFP or the end of the current MFP period is reached. Here, the MFP performs a fundamental action for reserving information through multiple hops over multiple flows in parallel in cascading way. Therefore, it can reduce handshaking and data delivery times in efficient manner.

As first address has given to node E it has the first scheduling priority for both regular and reverse flow data transmissions. If node S earlier received the confirmation MFP from its next hop E, S relays the data frame to E after the time interval of *Delay_Data* ($D^{f1}_{DD,S}$) like [2]. The relay E can transmit a train of data packets to the next relay F continuously without having any time duration between packets. Correspondingly, F also forwards the train of data packets without interval as the multi-hop channels are reserved directed toward the destination nodes.

To stay away from possible collisions from data transmission over other flows, S delays from transmitting data for *Delay_Data* ($D^{f1}_{DD,S} + d_{data}$). The data delivering process of the second flow works as the same way as the first one. This data frame sending process continues at each hop until the final destination C is reached. Thus data relaying process is very much like a pipeline process. The data from two different steams can be delivered simultaneously in our proposed protocol.

3.3.1 *Transmission of Data Adopting the Reverse Packet Method.*

Control packet MFP adopts a data transmission technique using the reverse multi-flow packet method where multiple data packets are sent from the same flow in reverse direction. Hence channel utilization is increased.

Figure 2 illustrates how the reverse packet method is employed in BMF protocol. After successful transmission of data packets from source S to G and S to C, the data transmissions of reverse packets take place. If any node has packet from similar flow in reverse direction it sets its reverse flag to 1 when transmitting MFP packet. When a node observes that it has packet from reverse flow direction, it simply sends a CTS packet to sender and sender then sends the packet to the destination node.

Suppose, for the first flow from S to G, there are reverse packet from node E to S and node G to F. Therefore, while transmitting MFP from S to E, S sets its reverse flag to 1. Moreover, F transmits MFP in the same manner to node G. After the transmission of data packets from node S to E and S to A, node S enters in $Wait\_Flow$ state and transmits a CTS to node E and E immediately transmits data packet to S. Furthermore, as there is no transmission going on one hop distance of node F, after completion of forward transmission of data packets from nodes F to G, F sends a CTS to G and G sends reverse data packet to F instantly.

Imagine that node A has batch packets for destination node S and node B has batch packets for destination node A. As node A is set as second destination, after end of regular and reverse transmission of node E, node A starts its reverse flow data packet transmission. Node S transmits a CTS to node A. and A immediately transmits data packet to S. The data sending process from node B to A performs in the same way as well.

3.3.2 *Transmission of Data Adopting Request Packet Method.*

Scenario 1: Sender misses a confirmation: Suppose, the sender node S does not receive confirmation MFP packet to its requested MFP of node A. After $d_{WD,A} + T_w$ duration of time when receiver A does not receive data from S, node A assumes that its confirmation MFP packet does not received by S. Therefore, node A sends RP and S begins sending data packets for second flow to node A.

Scenario 2: Sender misses a confirmation: In the second scenario, the sender node S does not receive confirmation to its request of both nodes E and A. After $d_{WD,E} + T_w$ duration of time when

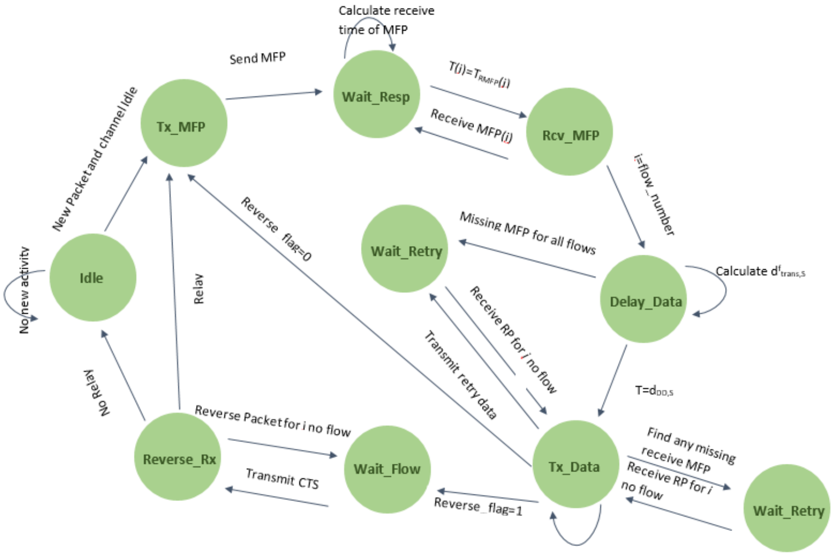

Figure 3: State transition diagram of a sender of the bidirectional multi-flow MAC protocol

receiver E does not receive data from S, E sends RP packet and node S immediately starts sending data packets for first flow to E. In the same way, $d_{WD,A} + T_w$ period later when receiver A does not receive data packet from S, A sends RP packet and S starts sending data packets for second flow to A instantly.

Scenario 3: Intermediate node misses a confirmation: Assume that, due to packet collision the relay node E does not receive confirmation MFP packet of node F. When F does not receive data frame E, E transmits RP packet and starts relaying data packets to F immediately.

Scenario 4: Immediate destination node misses a confirmation: Suppose, the node B does not receive confirmation to its request of destination node C. After $d_{WD,C} + T_w$ duration of time when C does not receive data from relay B, C sends RP and B immediately starts transmitting data to destination C.

Scenario 5: Finally, suppose the relay node fails to receive the requested MFP while S has sent MFP to the relay. Hence, the relay does not wake up, and the sender waits for reception of RP from the relay. As S does not receive RP packet, it will infer that its request is lost and after $Wait\_Retry$ state S will send MFP packet again.

3.4 State Transition Diagram for Sender Node

The state transition diagram of a sender in our proposed MAC protocol BMF-MAC is interpreted in Figure 3. When a sender has no activity to do, it remains in *Idle* state. If a sender generates new packets and channel is idle, it moves to the *Tx_MFP* state. In *Tx_MFP* state, the sender sends MFP to the relay nodes over multi-flows. After transmitting MFP packet, sender goes to the $Wait\_Resp$ state where it waits for the time for receiving MFP from that corresponding relays. In this state, the sender node calculates T^i_{RMFP} the receiving time of MFP. If time is equal to the time of receiving of a MFP for i number flow, the sender moves to the *Rcv_MFP* state. After receiving MFP for i number flow, the sender moves to $Wait\_Resp$ state. The sender waits for the waiting time of next receiving MFP T^{RMFP}_{i+1} in $Wait\_Resp$ state. Then if receiving time of MFP occurs, it moves to *Rcv_MFP* state again for receiving MFP of $i + 1$ number flow. This switching of states $Wait\_Resp$ and *Rcv_MFP* continues until the number of flow i is less than the flow construction number for multi-flow data transmission.

If $f = total\_flow\_number$, the sender moves to the $Delay\_Data$ state. In $Delay\_Data$ state, the sender calculates $d^i_{trans,S}$ time to transmit data packet for different flows i. The sender node waits for d_{DD} time in this state. From state $Delay\_Data$ a sender can move to two different states according to the cases. In case 1) If the sender receives all corresponding response MFP, it starts transmission of data packets to the relay nodes in Tx_Data state. The sender stays in Tx_Data state until it sends data packets over all flows in forward direction. In case 2) If the sender misses MFP for all flows, it waits for receiving retry packet RP in $Wait\_Retry$ state. After receiving RP for i number of flow, the sender moves to the Tx_Data state for transmitting data packet for that specified RP packet. This shifting of states $Wait\_Retry$ and Tx_Data progresses up till the sender waits for RP packets for all constructing flows.

In Tx_Data state three situations can take place. Case 1) If the sender does not miss MFP for any flows and $reverse\_flag$ is one in confirm MFP, the sender waits for $d^i_{WF,S}$ time in the state $Wait\_Flow$ for avoiding collision. Then after this duration of time, the sender transmits CTS packet and moves to $Reverse\_Rx$ state. In $Reverse\_Rx$ state, the sender receives reverse data packets over that similar flow. After receiving reverse data packets, it goes to the state $Wait\_Flow$ again and waits for $d^{i+1}_{trans,S}$ time for receiving reverse data transmission over next flow $i + 1$. The sender switches between these two states $Wait\_Flow$ and $Reverse\_Rx$ as far as the data transmission is finished over reverse flow for all f number of flows. Case 2) If the sender misses response MFP for any flow the sender waits for time T_w in $Wait\_Retry$ state. After receiving RP

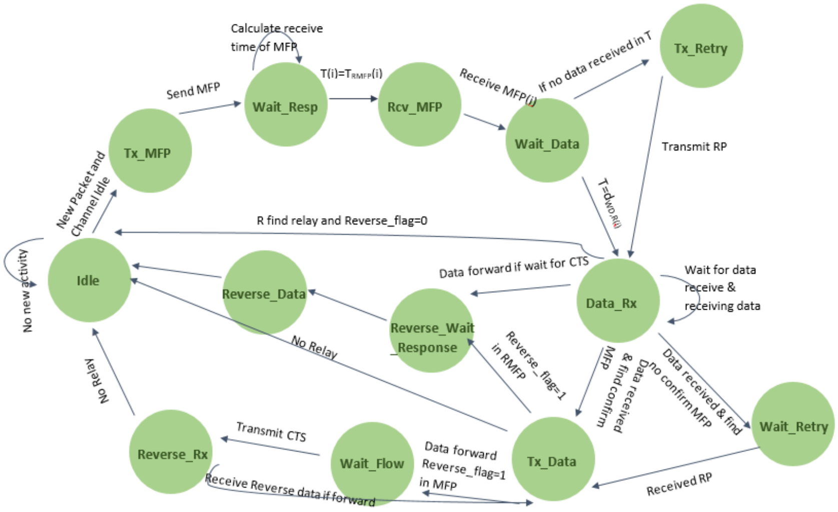

Figure 4: State transition diagram of a relay of the bidirectional multi-flow MAC protocol

packet for particular MFP, sender node then enters in *Tx_Data* state for receiving data from the relay node. If more than one response MFP is missed then node shifts between this two states *Tx_Data* and *Wait_Retry*. Then if node has reverse data to receive it goes to *Wait_Flow* state, transmits CTS and moves to *Reverse_Rx* state. Case 3) If sender recognizes that $reverse\_flag = 0$, sender understands that there is no data transmission over reverse flow direction, therefore it goes to the *Idle* state.

3.5 State Transition Diagram for Relay Node

Figure 4 illustrated the state transition diagram of a relay node of BMF-MAC protocol. When a relay has no activity to do, it remains in *Idle* state. In *Idle* state, if a relay node wants to relay packets and channel is idle, it moves to the *TX_MFP* state. Then relay sends the request MFP to the next relay $R_{(i+1)}$ node over multiple flows. After transmitting MFP packet, relay enters in the *Wait_Resp* state where it waits for the time T_{RMFP} for receiving MFP from the next relay. In this state, the relay node calculates the receiving time of response MFP T_{RMFP}. If time is equal to the time of receiving of a MFP, the relay moves to the *Rcv_MFP* state where it receives the confirm MFP packet.

After receiving the MFP packet from next relay, the relay node enters in *Wait_Data* state. In *Wait_Data* state relay nodes calculate the waiting time d_{WD,R_i} and wait for the duration to receive the data packets from relay $R_{(i-1)}$ or the sender node. In *Wait_Data* state two different circumstances may occur. Case 1) If $T = d_{WD,R_i} + T_w$ and no data is received in time T, the nodes moves to the *Tx_Retry* state, transmits RP packet and goes to *Data_Rx* state. Case 2) If the $T = d_{WD,R_i}$ and data is received in time T, the nodes moves to the *Data_Rx* state and starts receiving relaying data from its corresponding relay $R_{(i-1)}$ or sender node.

From state *Data_Rx* relay can move to four different states according to the cases. Case 1) In case that, the relay node receives data and finds confirm MFP, it moves to *Tx_Data* state where the relay starts transmission of data packets to its next relay node. Case 2) On the other hand, if the relay receives data but misses confirm MFP, it moves to *Wait_Retry* state. Then the node waits for receiving retry packet RP. After receiving RP from corresponding relay node, the node moves to the *Tx_Data* state for transmitting data packet for that specific RP packet. Case 3) If the relay receives data and $reverse\_flag = 1$, the node moves to *Reverse_Wait_Resp* state. Case 4) Finally, if the relay receives data and relay node is the last node over the flow and $reverse\_flag = 0$, the node moves to *Idle* state.

A node transmits data packet to its next relay node in *Tx_Data* state. In *Tx_Data* state three situations may occur. Case 1) In case that, the relay node forwards data and finds $reverse\_flag = 1$ in requested MFP, it moves to *Wait_Flow* state. The relay calculates the time d_{WF,R_i} and waits for d_{WD,R_i} time duration for avoiding collision in *Wait_Flow* state. After sending CTS packet to the interrelated relay node, the node moves to the state *Reverse_Rx* state. In *Reverse_Rx* state the node receives data over reverse flow direction. If data have to forward then the node moves to *Tx_Data* state. After forwarding data in *Tx_Data* state, if there is no data to relay, the node enters in *Idle* state. Case 2) In other case, while the node forwards data and finds $reverse\_flag = 1$ in responded MFP, it enters in *Reverse_Wait_Resp* state. In this state, the node calculates the time $d_{Reverse,R_i}$ and waits for $d_{WD,R_i} + T_{CTS}$ time duration for avoiding collision. After receiving response packet CTS from the specific relay node, the node moves to *Reverse_Data* state. The node transmits reverse data in *Reverse_Data* state. If there is no relay node, the relay moves to the *Idle* state. Case 3) If relay has no data to relay, it goes to *Idle* state.

3.6 Calculation of Time Duration Parameters

Here we consider that τ_{max} is the maximum propagation delay between nodes. $T_{control}$ is the common transmission time of all control packets. Assume d^{f1}_{data} and d^{f2}_{data} are the transmission time of batch data packets for flow one and two respectively. The busy duration of node S for first flow in Figure 2 is as follows

$$d^{f1}_{busy,S} = 2\tau_{max} + T_{control} + d^{f1}_{DD,S} \tag{1}$$

The waiting and busy duration of Node E is

$$d_{WD,E} = 2\tau_{max} + T_{control} \tag{2}$$

$$d_{busy,E} = d_{WD,E} + d^{f1}_{data} \tag{3}$$

Assume T_{data} is a single data packet transmission time and B_{SIZE} is the batch data size. The busy duration of Node F is given by:

$$\begin{aligned} d_{busy,F} &= (d_{busy,E} - T_{control}) + B_{size}T^{f1}_{data} \\ &= 2\tau_{max} + 2B_{size}T^{f1}_{data} \end{aligned} \tag{4}$$

Thus the busy duration of relay R_i for first flow can be generalized.

$$d_{busy,R_i} = 2\tau_{max} + iB_{size}T^{f1}_{data} - (i-2)T_{control} \tag{5}$$

The busy duration of Node S for second flow in Fig. 1 is as follows

$$d^{f2}_{busy,S} = d^{f1}_{busy,S} + d^{f1}_{data} \tag{6}$$

The waiting and busy duration of Node A is

$$d_{WD,A} = 2\tau_{max} + T_{control} + d^{f1}_{data} \tag{7}$$

$$d_{busy,A} = d_{WD,A} + d^{f2}_{data} \tag{8}$$

Like Node A, the busy duration of B is:

$$d_{busy,B} = (2\tau_{max} + d^{f1}_{data}) + B_{size}T^{f2}_{data} \tag{9}$$

Imagine f is the total number of flows. Therefore, we can derive the busy duration of relay node R_i for second flow as well.

$$d_{busy,R_i} = 2\tau_{max} + d^{f1}_{data} + iB_{size}T^{f2}_{data} - (i-2)T_{control} \tag{10}$$

The waiting time for ongoing first flow of node S is given by:

$$d^{f1}_{WF,S} = d^{f2}_{busy,S} + d^{f2}_{data} \tag{11}$$

Assume T_{CTS} is the transmission time of a CTS packet. Hence, the delay time for reverse flow of node E for first flow is given by:

$$d_{reverse,E} = d^{f1}_{WF,S} + T_{CTS} \tag{12}$$

The waiting time of node F is:

$$d_{WF,F} = d_{busy,F} + d^{f1}_{data} \tag{13}$$

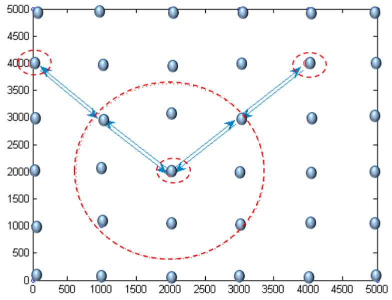

Figure 5: The network topology for analysis

Table 1: Systems Parameters

Parameters	Value
Acoustic propagation speed	1500 m/s
Transmission rate	9600 bps
Buffer Capacity(N_{max})	300 packets
Data packet size	1200 bits
Control packet size	128 bits

4 PERFORMANCE EVALUATION

For our analytical analysis we consider a multi-hop topology of 36 static nodes which are placed in a 5000 × 5000 m^2 square area which is illustrated in Figure 5. The distance between two nodes are 1000 m in grid spacing. The transmission range of node is 1.5 times the grid spacing that is 1500 m. Here we assume every node has the same transmission power. All of the nodes are assumed have exactly eight neighbors within its range which is indicated by the dotted circle in Figure 5. The average transmitting and receiving power is 2W and 20 mW of the acoustic transceiver. The acoustic channel is assumed to be error-prone.

BMF-M is a version of BMF protocol, where only multi-flow data transmission is considered. On the contrary, BMF-R contemplates multi-flow bidirectional data transmission. The performance of both version of BMF-MAC protocol in terms of throughput and latency is evaluated using simulation tool MATLAB. The network parameters have been set for BMF-MAC are shown in Table 1.

4.1 Throughput

We study the performance of throughput according to different offered loads, different distances and number of flows. In Figure 6, it is recognized that, BMF-MAC exhibits the best performance in terms of throughput in all offered load conditions. The analysis result proves CMRT is inefficient, because it only transmits data from sender to receiver over a single flow with per-round channel reservation, which suffers from underutilization of the channel when the propagation delay is high. However, our channel reservation mechanism allows a single sender to transmit data packets to multiple nodes of different flows with per round of channel reservation and can reduce the total channel reservation overhead greatly and thus can improve channel utilization. As a result, BMF-MAC has better data throughput than CMRT. Figure 6 reveals that, in case of high traffic load 3 packets/s, BMF-M can achieve the highest increase of data throughput around 39.2% higher compared to CMRT protocol as well as BMF-R can achieve the highest increase of data throughput around 67.5% higher compared to CMRT protocol. In low traffic load 0.5, BMF-M protocol can achieve throughput around 20% and BMF-R achieve 46.7% higher than that of CMRT.

Throughput of proposed BMF-MAC protocol and existing CMRT protocol in the variation of inter nodal distances with BER of 10^{-3} is shown in Figure 7. Here offered load is set to 0.8 packets/s. It is examined that the performance of each of the protocols in terms of throughput degrades with the increase of the inter-nodal distance. This is due to the rising of the distance-related communication overhead. As the distance enhances, the busy duration along with the handshaking time raises by reason of prolonged propagation delay. Therefore, with the increase of the extended busy duration the Silence state length enlarges. This causes a node to have less opportunity to exchange control packets. More specifically, from Figure 7 it is seen that for smaller inter nodal distance 1km, the throughput of BMF-R and BMF-M protocols can achieve around 83% and 45% higher compared to CMRT protocol respectively. For high distant node 8km, BMR-R can achieve the highest decrease of throughput around 19% high compared to CMRT whereas BMF-M gains the maximum decrease of throughput around 15% large compared to CMRT. Accordingly, BMF-MAC surpasses CMRT in respect to throughput with variable inter nodal distant nodes.

The throughput of proposed BMF-MAC protocol and existing CMRT protocol with the increase of number of flows is illustrated in Figure 8. for BER 10^{-3} and offered load 3.2 packets/second. Figure 8 depicted that the performance of both CMRT and BMF-MAC protocols in terms of throughput degrades with the increase of the number of flows. As with the increase of number of flows more time is required to deliver packet. Moreover, multiple flows cause the traffic pattern more complex and hard to handle by MAC protocol in underwater sensor networks scenario. From Figure 8 it is observed that for data transmission over double flow, the throughput of BMF-R and BMF-M protocols can achieve around 70.1% and 40.1% higher compared to CMRT protocol respectively. In case of six number of flows, BMF-R and BMF-M MAC protocol can accomplish throughput around 39.5% and 14.5% greater than that of CMRT protocol respectively. Accordingly, BMF-MAC exceed CMRT in respect to throughput in different flows condition.

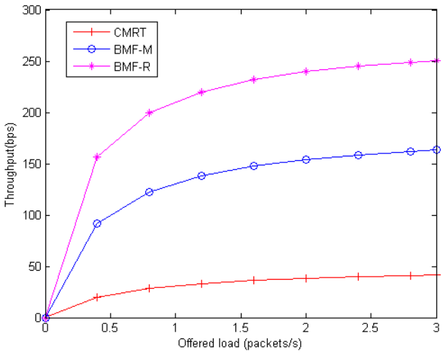

Figure 6: Throughput versus offered load

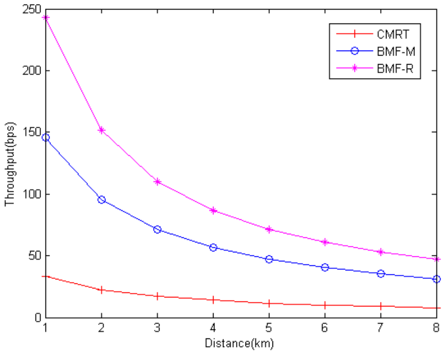

Figure 7: Throughput versus inter nodal distance

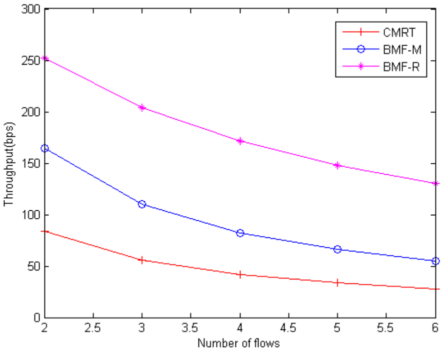

Figure 8: Throughput versus number of flows

4.2 Latency

We study the performance of latency according to different number of hops, different distances, number of flows. End to end packet delay of proposed BMF-MAC protocol and existing CMRT protocol in the variation of number of hops with BER of 10^{-3} is presented in Figure 9. As can be seen from the figure, in case of medium number of hop 5, BMF-M MAC protocol provides 25% less packet delay compared to CMRT. Whereas in high number of hop count 10, BMF-M MAC protocol can achieve latency around 50% lower than that of CMRT protocol. On the other hand, in medium number of hop 5, BMF-R MAC protocol provides 35% less packet delay compared to CMRT protocol. Furthermore, in high number of hop number 10, BMF-R MAC can achieve the significant reduction of latency around 72% lower compared to CMRT protocol. Therefore, it is observed that, BMF-MAC protocol outperforms the existing CMRT protocol in terms of latency with the increase of number of hops. Packets experience longer delay while using CMRT scheme compared to BMF-MAC scheme with the increase of number of hops. That is because, CMRT only transmits train of data per round handshake to multi-hop relaying nodes in a single flow. In BMF-MAC exchange of control packets and data packets in different flows can be held simultaneously. The protocol permits more scheduled transmissions per round handshaking. Hence, the protocol is capable to significantly reduce the time spent in data transmission and handshaking by means of bidirectional data-transmission over multiple flows. Therefore, it is observed that, BMF-MAC protocol outperforms the existing CMRT protocol in terms of latency with the increase of number of hops. Figure 10 reveals the end-to-end delay of BMF-MAC and CMRT with the inter nodal distance of node. As the distance enhances, the busy duration along with the handshaking time raises by reason of prolonged propagation delay. Therefore, with the increasing of the extended waiting duration the $Wait\_Flow$ state length enlarges. This causes a node to need more time to transmit bidirectional packets in multi-flow scenario. From Figure 10 it is shown that, while the distance is 1km, BMF-M

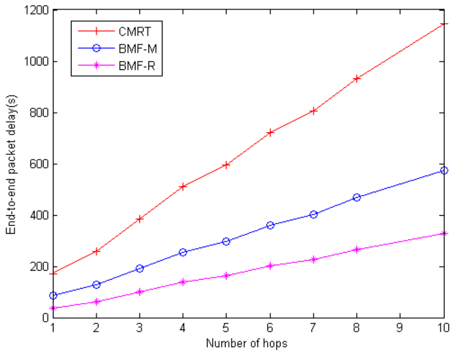

Figure 9: End-to-end delay versus number of hops

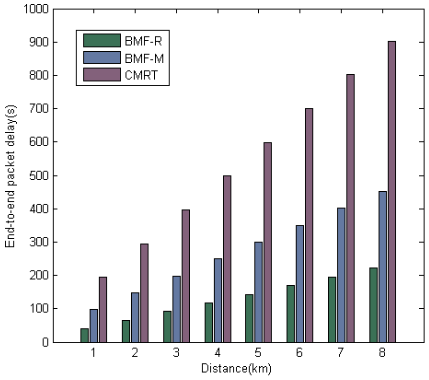

Figure 10: End to end delay versus inter nodal distance

MAC protocol provides 10% less packet delay compared to CMRT protocol. While the distance is increased to 8km, BMF-M MAC can obtain the highest reduction of latency around 69% lower compared to CMRT protocol. Hence, it is observed that, BMF-MAC can provide better result to handle variable traffic patterns in multi-hop underwater sensor networks comparing with CMRT.

Figure 11 depicted that the performance of both CMRT and BMF-MAC protocols in terms of end-to-end delay declines with the increase of the number of flows. Literally from Figure 11 it is shown that, data transmission over double flows, BMF-M MAC protocol provides 34% less packet delay compared to CMRT. Whereas in high number of flow number 4, BMF-M MAC protocol can provide latency around 76% lower than that of CMRT protocol. Additionally, data transmission over double flows, BMF-R MAC protocol gains 40% less packet delay compared to CMRT protocol. Furthermore, for number of flow 4, BMF-R MAC can obtain the apical degradation of latency around 88% lower compared to CMRT protocol.

5 CONCLUSION

In our protocol data transmission with bidirectional multi-flow packet method is developed to allow sender to send multiple MFP to different receivers with parallel reservation of channels. Additionally, pioneer transmission of a CTS is sufficient for transmission

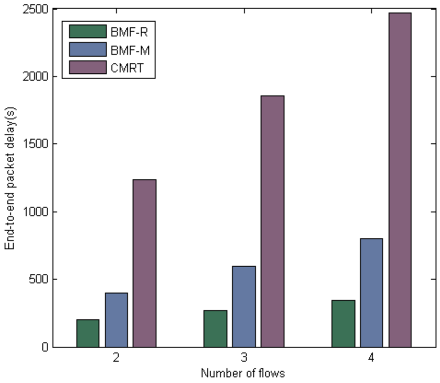

Figure 11: End to end delay versus number of flows

of data packets in reverse flow direction without exchanging of control packets thus reducing control packet overhead. In order to show the efficiency of the proposed scheme, the performance of BMF-MAC protocol with the existing CMRT protocol has been compared. The analysis shows that the proposed MAC protocol performs better by decreasing the end to end latency while increasing the throughput in multi-hop UW-ASNs. Therefore, the proposed BMF-MAC protocol surpasses existing CMRT protocol by complying simultaneous bidirectional data transmission over multiple streams.

ACKNOWLEDGMENT

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

REFERENCES

[1] Gkikopouli, A., Nikolakopoulos, G., and Manesis, S., "A Survey on Underwater Wireless Sensor Networks and Applications," in *Control & Automation (MED) Mediterranean Conference on, IEEE* , pp. 1147-1154 , Jul. 2012.

[2] Jae W. L., and Ho S. C, "Cascading Multi-Hop Reservation and Transmission in Underwater Acoustic Sensor Networks," *Sensors,* Vol. 14, pp. 18390-18409, 2014.

[3] Partan, J., Kurose, J., and Levine, B. N., "A survey of practical issues in underwater networks," *ACM SIGMOBILE Mobile Computing and Communications Review, 11(4)*, pp. 23-33, 2007

[4] Chen, K., Ma, M., Cheng, E., Yuan, F. and Su, W., "A Survey on MAC protocols for Underwater Wireless Sensor Networks," *IEEE Communications Surveys & Tutorials , Vol. 16, No. 3,* Third Quarter 2014.

[5] Molins M., and Stojanovic M, "Slotted FAMA: A MAC Protocol for Underwater Acoustic Networks," *IEEE OCEANS '06 Asia Pacific, Singapore*, pp. 16-19, May 2006.

[6] Ng H.H., Soh W.S., and Montani M, "BiC-MAC: Bidirectional-Concurrent MAC Protocol with Packet Bursting for Underwater Acoustic Networks," in *MTS/IEEE OCEANS Seattle, WA, USA*, pp. 20-23 September 2010.

[7] Chirdchoo N., Soh W.-S., and Chua, K. C., "MACA-MN: AMACA-based MAC protocol for underwater acoustic networks with packet train for multiple neighbors," in *Vehicular technology conference*, 2008.

[8] Ng, H.H., Soh, W.S., and Montani, M, "An Underwater Acoustic MAC Protocol using Reverse Opportunistic Packet Appending," in *Computer Networks*, pp. 2733-2751, 2013.

[9] Wen H. L., and Ssu C. K., and Yu C. L, "A Receiver-Initiated MAC Protocol with Packet Train Design for Underwater Acoustic Sensor Networks," in *Springer*, New York, January 2015.

[10] Mohamed S. H., Turkmen C., and Ashfaq K, "CL-MAC: A Cross-layer MAC protocol for heterogeneous wireless sensor networks," *Ad Hoc Networks,*pp. 213-225, 2013.