Underwater Wireless Sensor Network Layer Design for Reef Restoration

T. T. Manikandan, Rajeev Sukumaran

Abstract—Coral Reefs are very important for the majority of marine ecosystems. But, such vital species are under major threat due to the factors such as ocean acidification, overfishing, and coral bleaching. To conserve the coral reefs, reef restoration activities are carried out across the world. After reef restoration, various parameters have to be monitored in order to ensure the overall effectiveness of the reef restoration. Underwater Wireless Sensor Network (UWSN) based monitoring is widely adopted for such long monitoring activities. Since monitoring of coral reef restoration activities is time sensitive, the QoS guarantee offered by the network with respect to delay is vital. So this research focuses on the analytical modeling of network layer delay using Stochastic Network Calculus (SNC). The core focus of the proposed model will be on the analysis of stochastic dependencies between the network flow and deriving the stochastic delay bounds for the flows that traverse in tandem in UWSNs. The derived analytical bounds are evaluated for their effectiveness using discrete event simulations.

Keywords—Coral Reef Restoration, SNC, SFA, PMOO, Tandem of Queues, Delay Bound.

I. Introduction

WHEN the prominence of coral reefs in the ocean is concerned, it is evident that thousands of species have symbolic interactions with the coral reeds. The corals themselves tend to establish a symbolic relationship with zooxanthellae, their algal partner, which results in the brownish color of corals. When the corals are stressed, they expel algae which in turn changes its natural brownish color to white. Even though corals can sustain such short-term bleaching, corals as well as the thousands of species dependent on coral reefs will be in danger when bleaching continues for a longer period of time.

Coral Reef Restoration is one of the global initiatives to preserve the coral reefs and the aquatic ecosystem dependent on them. The core idea behind coral reef restoration is growing the individual corals in the monitored external space and then replanting those corals in the area where corals are destroyed due to bleaching. Post-coral replantation environment of those corals is to be continuously monitored so that the coral reef formation is not affected by the environmental parameters. Usually, set of individual corals take around 1-2 years to transform itself into a coral reef. So during this long period, environmental parameter monitoring should be done. But doing such long-term monitoring with human effort is impossible.

Here come the underwater sensor networks which with the help of sensors that are delayed in the field of coral replantation, environmental factors such as light, water temperature, water current, and salinity are sensed. The quality of water will also be sensed by the sensors due to the direct connection it has with the local coral bleaching events.

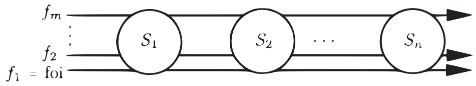

Fig. 1 Tandem of n Servers

The information collected by the underwater sensor is then communicated to the buoys at the surface of the ocean. Then the information collected from the sensors is compared with the information collected from the ocean weather stations for a detailed study. But the main issue here is the communication delay with the packets at the network layer in the sensor-buoy communication phase which is vital to a critical reef restoration monitoring application. The information about the corals that are collected from a particular underwater region needs to be delivered to the sink sensor nodes without any delay because the underwater parameters change quickly over time the time of delivery of data decides their usefulness. So UWSNs need to provide a strict QoS with respect to delay for coral reef restoration application. So this research focuses on mathematically bounding the delay of packets.

Prediction of packet delays in UWSNs is not so easy due to the highly variable nature of the underwater environment and the sensitivity it carries with respect to the overall network performance. When packet delay in UWSNs is concerned, the delay acquired through queuing of packets acts as a major source. But it is hard to predict the queuing delay of packets in UWSN since the packets traverse along tandem queues from source to destination. In this research, we consider the set of flows that traverse through the tandem of queues together as shown in Fig. 1, and focus on estimating the delay bound for a particular flow from the set of flows.

To analytically model the delay of a particular flow from the set of network flows, SNC is adopted as a mathematical framework. The main reason behind the adoption of SNC is the support it offers for modeling per flow bounds which is not the case with conventional queuing theory [1], [2], [3]. In stochastic network calculus, the delay bounds can be calculated either based on moment generating functions or tail bounds. But the mathematical results using both these methods are not available for complex networks. If the existing literature is concerned, the work in [2] have used MGF based method for modeling the delay bounds considering the

T. T .Manikandan and Rajeev Sukumaran are with the Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India (e-mail: abilash545@gmail.com, rajeevcbe@gmail.com).

topology represented in Fig. 2. But as per the work in [2] it is not possible to transform from the tandem queue scenario considered in this research to the one represented in Fig. 2 because of the stochastic dependency the flows carries between them. This violates the assumptions of the model in [2].

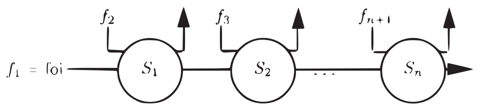

Fig. 2 Tandem in [2]

SNC based mathematical model proposed in this research have adopted holder inequality to relax the assumptions of [2] so that the stochastic single flow analysis can be done using MGF. But, by considering the pay-multiplexing-only-once technique of DNC, much better delay bounds for UWSN can be achieved. The core idea behind PMOO is the convolution of servers which is followed by the subtraction of cross flows. PMOO is exactly the opposite of SFA.

For the purpose of network layer modeling, apart from considering the tandem of queues illustrated in Fig. 1, it is also assumed that arbitrary multiplexing is followed with respect to per-flow analysis. Due to this assumption, the flow of interest is provided the lowest priority in tandem with queue setup in order to achieve uncompromised delay bound estimation. So that results of the proposed model can be easily adopted for the case where each of the individual servers of a tandem of servers follow priority.

When set of flows arriving into the network, there will be dependencies among one or more of such flows and this model provides that flexibility to the flows by not imposing any condition on the flows to be independent. Moreover, the proposed model also models the scenario where some flows of the network are dependent and some are independent. Due to these features the proposed model stands unique among the models proposed in the past for network layer of UWSNs. The evaluation of the proposed model is done based on the fractional Brownian motion (fBm) traffic model which is capable of modeling long-range network flows with dependencies. Since the MGF is available for fBm it fits well into the proposed network layer analytical framework. So in this research, we will focus on analyzing some of the interesting relations with respect to queuing models.

II. Basic Notations of Stochastic Network Calculus

In this section, we will deal with the basic notations and definitions of SNC. As per the definitions of SNC, the stochastic process A_p defines the flow arriving into the network. The arrival flow in the network is represented as $A(i,t) := \sum_{k=i+1}^{t} z(k)$ where z(k) represents the traffic increment process in time slot k. With respect to the coral reef restoration monitoring application, estimation of delay per flow is very much vital. So in this research, we use MGF based SNC framework for estimating delay bounds per flow. To be specific we try to bound the probability by which the delay value can exceed a certain value. The relation between the MGFs and the probability bounds is established using the Chernoff bound which can be defined as:

$$\Pr(X > z) \leq e^{-\theta z} E\left[e^{\theta X}\right] \tag{1}$$

where the moment generating function of the random variable X is denoted by $E\left[e^{\theta X}\right]$. In the following lines some of the basic definitions of SNC framework which will be used throughout this article are presented.

A. Convolution and Deconvolution Functions

The min-plus convolution and deconvolution of two functions x(i,t) and y(i,t) which are real valued and bivariate in nature can be defined as:

$$\begin{aligned} (x \otimes y)(i,t) &:= \min_{i \leq k \leq t} \{x(i,k) + y(k,t)\} \\ (x \oslash y)(i,t) &:= \max_{0 \leq k \leq i} \{x(k,t) - y(k,i)\} \end{aligned} \tag{2}$$

where $\otimes$ represents the convolution and $\oslash$ represents the deconvolution operation in min plus algebra. The dynamic S-Server whose definition will be presented in the following subsection will capture the characteristics of the service process.

B. Dynamic S-server

We assume that the service element in the network has an input arrival flow denoted by A_p and the flow coming out of the service element is denoted by D_p. Let S(i,t), 0≤i≤t, represents the stochastic process which is increasing with time t and non negative in nature. Then the service element can be considered as a dynamic S server iff for all t ≥0 it holds:

$$D_p(0,t) \geq A_p \otimes S(0,t) = \min_{0 \leq k \leq t} \{A_p(0,k) + S(k,t)\} \tag{3}$$

C. Leftover Service

As per the assumption made with respect to multiplexing in introductory section, the multiplexing is arbitrary and based on its worst case analysis it is assumed that the flow of interest will always have the lowest priority at the given dynamic server [4]. We consider that the network flow f_2 has been prioritized over the network flow f_1, in this case the leftover service for the arrival flow of f_1 denoted by A_1 can be represented as $[A_2 - S]^+$.

D. Virtual Delay

In a network at time t ≥0 virtual delay can be defined as:

$$d(t) := \inf\{s \geq 0 : A_p(0,t) - D_p(0,t+i) \leq 0\} \tag{4}$$

E. Output and Delay Bound

For a dynamic server S(i,t) with the arrival process A_p(i,t): The departure process D_p for any 0≤i≤t is upper bounded as per

$$D_p(i,t) \leqslant (A_p \oslash S)(i,t) \tag{5}$$

and the delay at time t ≥0 is upper bounded by

$$d(t) \leq \inf\{s \geq 0 : (A_p \oslash S)(t+i,t) \leq 0\} \tag{6}$$

Now the bound on the probability of violation of stochastic delay bound denoted by T can be represented as:

$$\Pr(d(t) > T) \leq E\left[e^{\theta(A_p \oslash S(t+T,t))}\right] \tag{7}$$

In (7), the stochastic delay bound has been estimated with the deconvolution operators. But to avoid the use of deconvolution operators we use the following inequalities proposed in [2]:

$$E\left[e^{-\theta(X \otimes Y(i,t))}\right] \leq \sum_{k=i}^{t} E\left[e^{-\theta(X(i,k)+\theta Y(k,t))}\right] \tag{8}$$

$$E\left[e^{-\theta(X \oslash Y(i,t))}\right] \leq \sum_{k=0}^{i} E\left[e^{\theta(X(k,t)-Y(k,i))}\right] \tag{9}$$

When the dependencies between the flows in the network are permitted it leads to many products inside the expectation, to tackle this issue we will use Holder inequality which is defined in the following subsection.

F. Holder Inequality

Let X_1,.......,X_n be the random variables such that $X_i \,\epsilon\, L^{p_i}$ then we have the following inequality:

$$E\left[\prod_{i=1}^{n} |X_i|\right] \leq \prod_{i=1}^{n} E\left[|X_i|^{p_i}\right]^{\frac{1}{p_i}} \tag{10}$$

where, $\sum_{i=1}^{n} \frac{1}{p_i} = 1$ and $p_i > 0$.

III. Network Layer Model for Monitoring of Reef Restoration

With the definitions of delay bounds for single server cases based on the SNC framework [5] presented in the previous section, we now move towards deriving the delay bounds for the tandem of servers. Initially, we start the illustration with the tandem of two servers where it is assumed that the flows are stochastically independent in nature. The illustration of 2 server tandem is provided in Fig. 3.

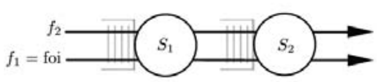

Fig. 3 Tandem with 2 Servers

Initially we assume the scenario illustrated in Fig. 3 where there are two servers with arrival of two flows. It is also assumed that both the servers are work conserving in nature and they are manually independent of the arrivals. At first, all the arrivals are considered to be independent later on this assumption is relaxed.

Here, in this research we present two different strategies namely Separated Flow Analysis (SFA) and Pay Multiplexing Only Once (PMOO) for estimating the delay bounds in the case of a tandem of servers.

A. Separated Flow Analysis for 2 Tandem Servers

In SFA, the delay bound estimation is done server by server by computing leftover service with arbitrary multiplexing. Once this is done, to convert the delay bound estimation to single server single flow case [6] all the servers along the tandem are convoluted. If Fig. 3 is considered f_2 is the flow that is prioritized over f_1 where as f_1 is the flow we are interested in analyzing. Now, under SFA strategy, as an initial step the cross flows of S_1 are subtracted which can be represented as $[S_1 - A_2]^+$. In the second step the cross flows of S_2 are subtracted, now the important thing to consider is the fact that the cross flow of S_2 is traversed through S_1, so we have to take into account its output which can be represented as $[S_2 - (A_2 \oslash S_1)]^+$. Now based on this the overall leftover service can be estimated as:

$$[S_1 - A_2]^+ \otimes [S_2 - (A_2 \oslash S_1)]^+ \tag{11}$$

By applying the output and delay bound estimation in Section II-E and leftover service definition in Section III-A the delay bound for 2 server tandem setup illustrated in Fig. 3 can be estimated as:

$$\begin{aligned}
&\Pr(d(t) > T) \\
&\overset{(7)}{\leq} E\left[e^{\theta A_{\text{foi}} \oslash S_{\text{I.o}}(t+T,t)}\right] \\
&\overset{(9)}{\leq} \sum_{z_0=0}^{t} E\left[e^{\theta(A_{\text{foi}}(z_0,t) - S_{\text{I.o}}(z_0,t+T))}\right] \\
&= \sum_{z_0=0}^{t} E\left[e^{\theta\left(A_1(z_0,t) - \left([S_1-A_2]^+ \otimes [S_2-(A_2 \oslash S_1)]^+\right)(z_0,t+T)\right)}\right] \\
&\leq \sum_{z_0=0}^{t} E\left[e^{\theta A_1(z_0,t)}\right] \\
&\quad \cdot \sum_{z+T}^{t+T} E\left[e^{-\theta\left([S_1(z_0,z_1) - A_2(z_0,z_1)]^+\right)}\right. \\
&\quad \left. \cdot e^{-\theta\left([S_2(z_1,t+T) - (A_2 \oslash S_1)(z_1,t+T)]^+\right)}\right]
\end{aligned} \tag{12}$$

In (12), inside the expectation we have stochastic dependency between the two exponentials. So to neglect this dependency we apply the definition of Section II-F and arrive at the delay bound:

$$
\begin{aligned}
.... \leq \sum_{k_0=0}^{t} E\left[e^{\theta A_1(z_0,t)}\right] \sum_{z_1=z_0}^{t+T} \Bigg(E\left[e^{p_1\theta A_2(z_0,z_1)}\right]^{\frac{1}{p_1}} \\
\cdot E\left[e^{-p_2\theta S_1(z_0,z_1)}\right]^{\frac{1}{p_1}} \\
\cdot \left(\sum_{z_2=0}^{z_1} E\left[e^{p_2\theta A_2(z_2,t+T)}\right] E\left[e^{-p_2\theta S_1(z_2,z_1)}\right] \right)^{\frac{1}{p_2}} \\
\cdot E\left[e^{-p_2\theta S_2(z_1,t+T)}\right]^{\frac{1}{p_2}} \Bigg)
\end{aligned}
\tag{13}
$$

where,

$$\frac{1}{p_1} + \frac{1}{p_2} = 1$$

B. Pay Multiplexing Only Once Analysis for 2 Tandem Servers

PMOO-based modeling of 2 tandem servers just does the reverse of SFA. Here as a first step, all the servers are convoluted with cross flows, and then the cross flows are subtracted. For the case illustrated in Fig. 3, this works perfectly but for the n tandem case due to the dependency, it is more complicated. But the major benefit of this method lies in the fact that it considers the end-to-end perceptive of tandem scenarios.

Now if we consider 2 tandem server cases PMOO at the first step convolutes both the servers which can be represented as $S_1 \otimes S_1$ which is then followed by the subtraction of cross flows which is represented as $[(S_1 \otimes S_1)$ - $A_2]^+$. When PMOO-based delay calculation is compared with the one based on SFA, A_2 appeared only once where as in the case of SFA it appeared twice which adds to the ineffectiveness of the acquired bound.

Now PMOO based delay bound can be derived as:

$$
\begin{aligned}
&\Pr(d(t) > T) \\
&\overset{(7)}{\leq} E\left[e^{\theta A_{\text{foi}} \oslash S_{1.o}(t+T,t)}\right] \\
&\overset{(9)}{\leq} \sum_{z_0=0}^{t} E\left[e^{\theta\left(A_1(z_0,t) - [S_1 \otimes S_2(z_0,t+T) - A_2(z_0,t+T)]^+\right)}\right] \\
&\leq \sum_{z_0=0}^{t} \Bigg(E\left[e^{\theta A_1(z_0,t)}\right] \\
&\quad \cdot E\left[e^{\theta(A_2(z_0,t+T) - S_1 \otimes S_2(z_0,t+T))}\right]\Bigg) \\
&= \sum_{z_0=0}^{t} \Bigg(E\left[e^{\theta A_1(z_0,t)}\right] E\left[e^{\theta A_2(z_0,t+T)}\right] \\
&\quad . \sum_{z_1=z_0}^{t+T} E\left[e^{-\theta S_1(z_0,z_1)}\right] E\left[e^{-\theta S_2(z_1,t+T)}\right] \Bigg)
\end{aligned}
\tag{14}
$$

From the delay bound acquired for both SFA and PMOO it is evident that only for SFA based delay bound estimation there comes the necessity to apply holder inequality which is not in the case of PMOO. This analogy becomes even more true when the number of servers in tandem becomes n.

C. Delay Bound Estimation for N Tandem Servers

In this section, the delay bounds estimated using SFA and PMOO methods for 2 tandem servers are tended to n tandem servers.

1) SFA Based Delay Bound Estimation for N Tandem Servers: Here, we assume that there are n servers in tandem and m flows arrives into the network which is actually the general network scenario depicted in Fig. 1.

$$
\begin{aligned}
&\Pr(d(t) > T) \\
&\leq E\left[e^{\theta(A_{foi} \oslash S_{\text{l.o.}}(i,t))}\right] \\
&\leq \sum_{z_0=0}^{t} \Bigg(E\left[e^{\theta A_1(z_0,t)}\right] \sum_{z_0 \leq z_1 \leq t+T} \cdots \\
&\quad \sum_{z_{n-2} \leq z_{n-1} \leq t+T} E\left[e^{p_1\theta \sum_{j=2}^{m} A_j(z_0,z_1)} e^{-p_1\theta S_1(z_0,z_1)}\right]^{\frac{1}{p_1}} \\
&\quad \cdots E\Big[e^{p_n\theta\left(\left(\left(\sum_{j=2}^{m} A_j\right) \oslash S_1\right) \ldots\right) \oslash S_{n-1}\right)(z_{n-1},t+T)} \\
&\quad \cdots e^{-p_n\theta S_n(z_{n-1},t+T)}\Big]^{\frac{1}{p_n}} \Bigg)
\end{aligned}
\tag{15}
$$

where, $\sum_{i=1}^{n} \frac{1}{p_i} = 1$.

2) PMOO Based Delay Bound Estimation for N Tandem Servers: Similar to the case of SFA, we assume that there are n servers in tandem and m flows arrives into the network which are independent from each other. For such cases the delay bound using PMOO is as follows:

$$
\begin{aligned}
&\Pr(d(t) > T) \\
&\leq E\left[e^{\theta(A_{foi} \oslash S_{\text{l.o.}}(t+T,t))}\right] \\
&\leq \sum_{z_0=0}^{t} \Bigg(E\left[e^{\theta A_1(z_0,t)}\right] \prod_{j=2}^{m} E\left[e^{\theta A_j(z_0,t+T)}\right] \\
&\quad \cdot \sum_{z_0 \leq z_1 \leq t+T} \sum_{n-2} \leq z_{n-1} \leq t+T \\
&\quad \cdots E\left[e^{-\theta S_n(z_{n-1},t+T)}\right] \Bigg)
\end{aligned}
\tag{16}
$$

So far we have considered n servers in tandem with all the flows independent of each other. Now we consider the case of n tandem servers with dependencies. We assume that there are m flows in the network and there are n servers where all the flows are dependent, now the delay bound using PMOO is as follows:

$$\Pr(d(t) > T) \leq \sum_{z_0=0}^{t} \left(E\left[e^{p_1 \theta A_1(z_0,t)} \right]^{\frac{1}{p_1}} \cdot \left(\prod_{j=2}^{m} E\left[e^{p_2 p_{j+1} \theta A_j(z_0,t+T)} \right]^{\frac{1}{p_{j+1}}} \cdot \sum_{z_1=z_0}^{t+T} \cdots \sum_{z_{n-1}=z_{n-2}}^{t+T} \left(E\left[e^{-p_2 \theta S_1(z_0,z_1)} \right] \cdots E\left[e^{-p_2 \theta S_n(z_{n-1},t+T)} \right] \right) \right)^{\frac{1}{p_2}} \tag{17}$$

where,

$$\frac{1}{p_1} + \frac{1}{p_2} = 1$$
$$\frac{1}{p_3} + \cdots + \frac{1}{p_{m+1}} = 1$$

When the flows are dependent, even PMOO can avoid the need for applying holder inequality similar to SFA.

IV. Results and Analysis

In this section the analytical model which is created in the previous section is evaluated using discrete event simulations. Specifically, we focus on analysing different aspects of delay bounds for per flow in case of tandem of server queues. The simulation scenarios are created for the tandem of server case with SFA and PMOO techniques respectively in Riverbed Simulator.

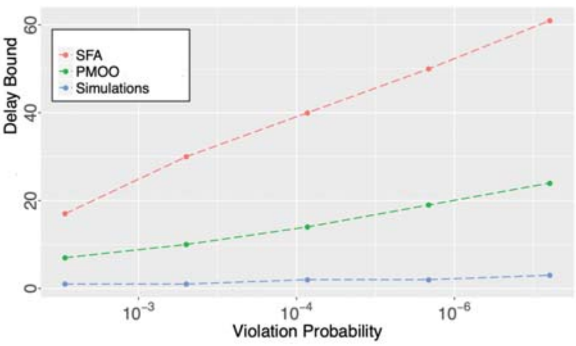

Fig. 4 SFA vs PMOO for 2 Server and 2 Flows with FBM

For the purpose of simulation modeling, the following parameter values are considered with respect to arrival model of FBM: The average rate of flow denoted by λ is set to the value of 0.5, the burstiness parameter denoted by σ is considered to have the value of 1 and the hurst parameter which determines the degree of long range dependencies of a low is set to have the value of 0.7. All the servers which are considered for the simulation are work conserving constant rate servers and the service rate of the servers is set equal to the number of flows. For simplicity, it is assumed that the number of network flows traversing through the tandem of servers is equal to number of servers.

In first phase of the simulation we have setup the scenario in Riverbed simulator with FBM traffic. The n arrival flows which are independent of each other enters the tandem of servers parallely. In this case the delay for one of the flow which we are interested in is recorded. In order to allow the simulation to have higher probability of violation, the simulation times and the service rate are feasibly altered. The delay results of this particular scenario is compared with the the analytical delay bounds obtained through SFA and PMOO methods and the graphical illustration of the same is presented in Fig. 4.

From the observation it is clear that the delay attained through simulations are on a lower side when compared with the delay bounds of SFA and PMOO. But in comparison between the two methods, PMOO is very much closer to the simulation results which shows the superiority of PMOO strategy for per flow delay bound estimation.

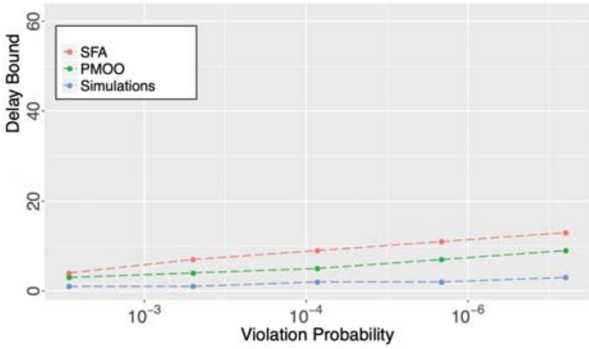

Fig. 5 SFA vs PMOO for 2 Server and 2 Flows with Exponential Distribution

To analyze further, the same scenario that we dealt with during phase 1 of simulations is considered and the simulations are carried out but this time with the exponential traffic distribution instead of FBM. The value of λ is now set to 1.8. The graphical illustration of this phase of simulation is presented in Fig. 5, from which it is evident that the delays obtained through simulations are much more closer than in the previous case with FBM. Even in this scenario, PMOO is the one which performs the best when compared with its counterpart SFA. From this we can very much understand that the SNC based analytical model created using PMOO method is very much suitable with respect to the closeness with the simulation delay bounds. So, we recommend the proposed Network layer model based on PMOO method for adoption in underwater coral reef restoration monitoring application for better estimation of delay bounds for the flows traversing through n tandem of servers.

V. Conclusion

In this research we have considered the case of n servers in tandem in UWSN and proposed a SNC based for estimation of stochatic delay bounds for the individual flow of interest. In the process, we have adopted two different methods for delay bound estimation; namely i) SFA and ii) PMOO. The stochastic delay bounds for n tandem servers are analytically derived using both the methods and for evaluating the closeness of the derived bound we have created a simulation model using Riverbed Simulator. The simulations are conducted for both SFA and PMOO with two different traffic models namely factorial Brownian Motion and exponential distribution separately. The results of the simulations show that the delay bounds estimated using PMOO method performs superiorly than its counterpart SFA. So, based on the extensive research and analysis we recommend to use PMOO based delay bound estimation with exponential traffic model for underwater coral reef restoration application which demands strict delay bounds from the underwater network layer.

REFERENCES

[1] C.-S. Chang, *Performance guarantees in communication networks*. Springer Science & Business Media, 2000.

[2] M. Fidler, "An end-to-end probabilistic network calculus with moment generating functions," in *200614th IEEE International Workshop on Quality of Service*. IEEE, 2006, pp. 261–270.

[3] F. Ciucu and J. Schmitt, "Perspectives on network calculus: no free lunch, but still good value," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, 2012, pp. 311–322.

[4] H. Wang, F. Ciucu, and J. Schmitt, "A leftover service curve approach to analyze demultiplexing in queueing networks," in *6th International ICST Conference on Performance Evaluation Methodologies and Tools*. IEEE, 2012, pp. 168–177.

[5] Y. Jiang, Y. Liu *et al.*, *Stochastic network calculus*. Springer, 2008, vol. 1.

[6] Y. Jiang, "Stochastic service curve and delay bound analysis: A single node case," in *Proceedings of the 2013 25th International Teletraffic Congress (ITC)*. IEEE, 2013, pp. 1–9.

T.T.Manikandan received his B.Tech in Computer Science and Engineering from SASTRA University, ME in Computer Science and Engineering from College of Engineering, Anna University and he is currently perusing PhD in Computer Science and Engineering at SRM Institute of Science and Technology. His research interest is in the area of Underwater Communication.

Rajeev Sukumaran is an Engineering Epistemologist and a Learning Researcher. He received his Ph.D in Computer Engineering and is an active researcher in Modeling Wireless Communication Networks and is with the Directorate of Learning and Development, SRM Institute of Science and Technology, Chennai, India.