SNC Based UWSN Network Layer Design for Monitoring Coral Bleach

Manikandan T T [1], Rajeev Sukumaran [2]

[1]Research Scholar, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India.

[2]Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India.

Corresponding Author: abilash545@gmail.com

Abstract: - Coral reefs around the world are under serious threat to their survival and have already led to degradation and destruction in many places. To preserve coral reefs and the underwater ecosystems dependent on them, various measures are being taken across the world. Monitoring of coral bleach activities and data collection from the location of coral reefs plays a vital role in coral reef recovery measures. For more precise information about the coral reefs, infield monitoring is preferred over other monitoring techniques such as satellite imaging. Underwater Wireless Sensor Networks (UWSN) play an important role in such infield monitoring of coral bleaching events. Once the information is collected from the underwater environment the way it is communicated to the offshore data center is very much vital so that the corresponding measures can be taken on time. This research focuses on the analytical modeling of network layer communication using Stochastic Network Calculus (SNC). With SNC we can represent a communication network in convolution form using which the tight end-to-end delay bounds of the network layer communication can be derived. The data flow in UWSNs is subject to transformations as they move towards the destination. This kind of data flow transformation can be analytically represented using scaling elements but it is limited to the case of fixed-sized data packets. But in reality, the packet size will be variable. So, in this research, we have designed an SNC-based analytical model of UWSN network layer communication for coral bleach monitoring application. The end-to-end delay is analytically derived for network layer packets of variable size subject to transformation using SNC for effective network layer communication in coral bleach monitoring application and the derived delay bounds are evaluated with discrete event simulations.

Key Words— ***SNC, Variable Length Packets, Flow Transformations, Packetizer, End to End Delay Bound.***

I. INTRODUCTION

In a highly dynamic network like UWSN which involves multiple sensor nodes, representation of the network in convolution form is very much vital to derive the precise end-to-end delay bounds. If the lower bound process S_k(i,t) represents the service provided by the node k for time $0 \le i \le t$, then the lower bound of the service processes in the case of n sensor nodes as tandem can be represented as follows:

$$S_1 \otimes S_2 \otimes \ldots\ldots \otimes S_n \tag{1}$$

where (min,+) convolution is denoted by ⊗ as:

$$S_1 + S_2 = \inf_{0 \le i \le t} \{S_1(0,i) + S_2(i,t)\} \tag{2}$$

By applying above defined convolution form the end-to-end performance analysis can be done for the single node case. But, the main limitation imposed by this convolution form is that the network flows needs to be unaltered throughout the transmission. But in reality, the data flows always tend to transform due to transmission loss, aggregation of data during the transmission, and routing of a part of the information to another destination. To deal with this kind of flow transmission the concept of scaling element is introduced in Deterministic

Manuscript revised August 31, 2022; accepted September 01, 2022. Date of publication September 03, 2022.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

Network Calculus [1] to analytically represent such transformations. Then subsequently in [2] the stochastic variant of the scaling element was introduced using which the actual flow transformations happening in the network can be captured in convolution form. The key idea behind this is to commute the dynamic server element along with the scaling element recursively to get the network's single-node representation. But the main limitation of this model is that it can be used only in the case of fixed-sized packets or bits. Since almost all the modern-day networks including UWSNs deal with variable length packets and the operations like sending and receiving cannot be represented in the form of bits this model cannot be used for the representation of UWSNs.

In this research, we have introduced a new scaling element that treats network flow as packets of variable length instead of bits. To ease the process of representing such flow transformations without affecting the convolution from the representation of multiple nodes networks the most widely accepted transformation operation demultiplexing is considered for the representation of flow transformations at the packet level.

In the past, there has been some research focusing on the representation of flows with variable length packets like [3][4] which introduced packetizer element into existing network calculus. Then the authors in [5][6] extended the packetizer element into SNC. We have used a packetizer along with the scaling element for the purpose of modeling transformations of flow at the packet level which is the first of its kind.

II. Basic Notations of SNC

In this section, some of the basic concepts of Stochastic Network Calculus (SNC) and the notations that will be used in this article are presented.

Throughout this article notations $A_p(t)$, $S_p(t)$, and $A_p^*(t)$ are used for denoting the arrival, service, and departure processes in UWSN. Usually, in SNC, all the processes arriving in the network are considered nonnegative processes, and all the negative processes arriving in the network are always an increasing function. Based on the above assumptions a flow in the network is represented as follows:

$$F = \{f(.): \forall 0 \le i \le t, 0 \le f(i) \le f(t)\} \tag{3}$$

at a given time t = 0

$$A_p(0) = A^*{}_p(0) = S_\rho(0) = 0 \tag{4}$$

for any, 0≤i≤t,

$$A_p(i,t) \equiv A_p(t) - A_p(i) \tag{5}$$

$$A_p^*(i,t) \equiv A_p^*(t) - A_p^*(i) \tag{6}$$

$$S_p(i,t) \equiv S_p(t) - S_p(i) \tag{7}$$

Traditional algebra has two common operations namely addition and multiplication represented by + and x respectively. Similarly, SNC has min plus algebra where the addition operation in traditional algebra becomes the computation of minimum which is represented using, likewise the multiplication operation becomes addition and it is represented using +. In the case of SNC, assume there exists a set S then if there exists the element in the set which is less than or equal to all the elements set then it is called infimum which is abbreviated as inf. Similarly, in a set S if there exits the least element which is greater than or equal to all the elements in the set then it is called supremum which is abbreviated as sup. The set of properties is defined for the min plus algebra as follows:

The (min, +) convolution of two functions i and j are represented as follows:

$$(i \otimes j)(z) = \inf_{0<l<k}[q] \tag{8}$$

where,

$$q = i(l) + i(k - l) \tag{9}$$

The (min, +) deconvolution of two functions i and j are represented as follows:

$$(i \oslash j)(z) = \sup_{t>0}\{q\} \tag{10}$$

where,

$$q = i(z + s) = i(t) \tag{11}$$

Generally, in UWSNs, a flow can be expanded in case of the addition of data from another flow, and also a flow can be divided into many sub-flows when some part of the flow is routed to a different destination, in this case, the demultiplexing operation is carried out on that particular flow. In such a case the arrival process of the original flow is divided into two arrival processes namely $A_p^{(1)}(t)$ and $A_p^{(2)}(t)$ satisfying:

$$A_p(t) = A_p^{(1)}(t) + A_p^{(2)}(t) \tag{12}$$

for all t ≥ 0. In a network the bit level splitting can be represented as an indicator function 1. The indicator function is true if the the particular bit goes to the destination 1 and the term will be false in other case. This indicator function for bit k can be represented a X_k, with this the scaling element can be defined as a random process denoted by $X = (X_k)_{k\geq 1}$. Then the scaled arrivals denoted by $A_p^X(t)$ can be defined as follows:

$$A_p^{\mathbf{X}}(t) = \sum_{k=1}^{A_p(t)} X_k, \forall t \geq 0 \tag{13}$$

Now this scaling element is being used to model the demultiplexing operation at bit level. This demultiplexing operation is assumed to happen instantly without any queue for scaling element. But in modern day networks like UWSN the flow transformations tend to happen at packet level instead of bits. In case of demultiplexing at packet level we only have the information of routing probability of the whole packet to the destination. So, in order to model the packet level demultiplexing operation the scaling element need to extended. But before doing that we would like to introduce the analytical representation of variable sized packets into SNC.

The length of the packets in UWSN can be denoted as l_1,l_2,.... representing as sequence of positive random variables. The cumulation of these random variables form a packet process denoted by $L_p(n)$ for all, n≥1 as $L_p(n) = l_1 + l_2 + \ldots + l_n$ with $L_p(0) = 0$ and $l_n = L_p(n) - L_p(n-1)$. Now with the below definition of packetizer we model the packet flow in UWSN. Figure 1 illustrates the network elements used for the proposed model.

2.1 Packetizer

With the packet process denoted by $L_p(n)$ and Arrival Process $A_p(t)$ the new network element named L Packetizer can de defined as a function $PK^L(.)$ satisfying for all $A_p(t), t \geq 0$:

$$PK^L(A_p(t)) = L_p(N_t), \tag{14}$$

where,

$$N_t = max\{m: L_p(m) \leq A_p(t)\} \tag{15}$$

A. Packet Scaling Element

A flow in the network is said to be L packetized for any t≥0 if $A_p(t) = PK^L(A_p(t))$. So the corresponding packet flow is also considered to be L packetized arrival process. Now by taking into consideration the demultiplexing of packet flow the scaling element is defined as follows:

$$A_p^{\mathbf{X}}(t) = \sum_{k=1}^{N_t} l_i X_k \tag{16}$$

Now the packet flow transformations can be modelled with the above definition of packet scaling element especially in the case of demultiplexing. In a network the queuing system processes the packet flow either before or after performing demultiplexing. As a packet flow get the required service from each node in the network the output will always be in the form of packets i.e the bits are assumed to be packetized by the packetizer PK^L.

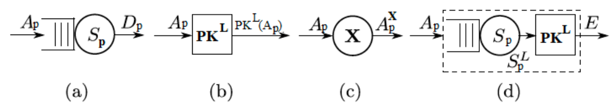

Fig.1. Network elements: (a) dynamic server, (b) packetizer, (c) packet scaling element, (d) packetized server.

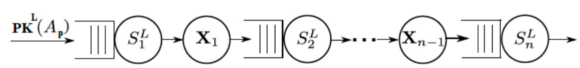

Fig.2. A network model consisting of packetized arrivals, services and packet scaling elements.

2.2 Dynamic Server

In SNC the network queuing system can be represented as a dynamic server which is denoted as $S_p(i,t)$ for all 0≤i≤t. This is not the actual server by the property of sever which defines the lower bound process on the provided network service such that for all t ≥0 there holds the following convolution inequality:

$$D_p(t) \geq A_p \otimes S_p(t) := \inf_{0\leq i\leq t}\{A_p(i) + S_p(i,t)\} \tag{17}$$

The dynamic server is said to be exact when the inequality holds with equality. When two dynamic servers are concatenated then with convolution it is also a dynamic server.

2.3 Packetized Server

In the network the packetizer is followed by the packetized server which is actually a bit server and such packetized server has a dynamic server denoted by S_p^L(i,t). Given,S_p^L and L_p of the of S_p and with the assumption that the maximum packet size denoted by l_{max} is known, we get the possible S_p^L as follows:

$$S_p^L(i,t) = \left[S_p(i,t) - l_{\max}\right]_+ \quad (18)$$

2.4 Packet Delay

Now with all the above definition of the network elements the packet delay denoted by W_p(t) can be defined for all t≥0 as follows:

$$W_p(t) = \inf\left\{d \geq 0: PK^L(A_p(t)) \leq PL^L(D_p(t+d))\right\} \quad (19)$$

Here, the packet arriving at time t is assumed to get the FIFO service and the delay experienced by the packet is a virtual delay.

III. Network Layer Modelling for Monitoring of Coral Bleach

In this section we propose SNC based network layer model design for UWSN through which the end to end delay for the UWSN with multiple demultiplexing operations are analytically derived. As per the literature there are two ways to estimate the end to end delay i) first way of estimation is by commuting the scaling elements and the service ii) In case of FIFO scheduling is used, end to end delay can be estimated based on the leftover service of the flow of concern [7]. In this research we follow the first way of end to end delay estimation. With this way, we can either directly observe the bit flow with packetizers for the estimation of delay bound or else we can estimate the delay bound by normalizing the bit level service and the packet flow with the packet size so that irrespective of the size the observation is done directly on the packet. Here we use the normalization based technique for end to end delay estimation.

We directly move our focus from bit level flow observation to packet level flow observation as shown in Figure 2. Here irrespective of size each packets in the network are considered as single data units.

Then we re represent the service received by the packets followed by the direct end to end delay estimation based on [2]. Assume that the UWSN has an arrival with packets for which the arrivals are defined in their last bit. This arrivals times can be modelled with the help of packet size distribution and the counting process and it can be expressed as follows:

$$A_p(t) = \sum_{k=1}^{N_t} l_i \quad (20)$$

where the counting process is defined as $\{N(t), t \geq 0\}$. Now we obtain the arrival process of the packets and this approach is called normalization of bit flow by packet sizes. Now the sequence of packets will be serviced with the combination of service element of bitwise service and the packetizer. Here we define the service size normalization represented by:

$$S_p^{norm}(i,t) = \sum_{j=i}^{t} c(j) \quad (21)$$

for all $0 \leq i \leq t$. Here, the time varying capacities of the serving a packet in the network at time t is represented by c(j).

In this research we propose the method for calculation of end to end delay in case of flows of variable length packets. Consider that the bitwise server in the network S $_p$(t) is offering the work conserving service with varying rates which can be represented as S $_p$(t) ≥ Ct for any t ≥ 0. Now we get the MGF bound $M_{S_p(t)}(-\theta) \leq e^{-\theta\bar{C}t}$ for $\theta > 0$. Then we can represent c(j) ≥ C/l $_x$ where the packet length is represented by l $_x$. Since the packet size have a maximum limit which can be represented by l $_{max}$ we can represent c(j) . Now the lower bound of the normalized server can be represented as

$$S_p^{norm}(i,t) \geq C/l_{max}(t,i) \quad (22)$$

Now the dynamic server can be represented with normalized capacity as C/l_{max}. Along with this with the assumption at each bit server that $M_{S_k(t)}(-\theta) \leq e^{-\theta C_k t}, k \geq 1$. The stochastic bound for the end to end delay can be derived as follows:

$$Pr(W_p > d) \leq K^n b^d \quad (23)$$

The main difference between our result and the results of theorem 1 presented in [2] is the MGF bound of each service :

$$M_{S_j^{\text{norm}}(i,t)}(-\theta) \leq e^{-\theta\frac{C_j}{l_{\max}}(t-i)}. \quad (24)$$

Since the proposed model does the delay analysis for the packet only when the last bit of the packet is serviced by the bit server this analysis is valid irrespective of the fact that the packetizer is present or not.

3.1 Theorem 1

With the proposed model, when the maximum length of the packet l_{max} is assumed to be $< \infty$ the steady state end to end delay bound for all $d \geq 0$ and $\theta_j > 0, j = 1,2, \dots, n$ can be expressed as follows:

$$Pr(W_p > d) \leq e^{\left(\sum_{j=1}^{n} \theta_j + \theta_1\right) l_{\max}} K^n b^d \tag{25}$$

IV. Results And Discussion

For the purpose of evaluating the analytical model proposed in this research for estimation of delay bounds for packets at the network layer of variable length the simulation is carried pout using Riverbed simulator. The comparison between the number of scaling and the delay is illustrated in Figure 3. From the figure it is very clear that the delay grows in the order of 0(n).

From the graph it is understood that the results of the Theorem 1 is very much closer to the simulation than the normalized flow this is because of the fact that, even though both the methods assumed the maximum packet size a l_{max} for the normalization based method the form is $C_j / l_{\max} \cdot t$ whereas in case of theorem 1 it is $\left[C_j \cdot t - l_{\max}\right]_+$. Since the division results in thing higher precision than the subtraction operation the deviation is obvious.Since it it the first attempt from our end to model the flow transformations of variable length packets we focused on the representation of such network elements in the SNC setting. We will focus on deriving the more tighter bounds in the subsequent attempts.

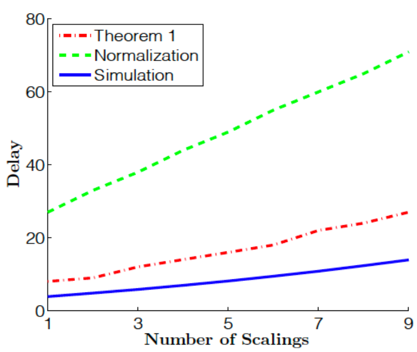

Fig.3. Delay bounds with Theorem 1, normalized flow, and simulation

In the Figure 4 the comparison between the theorem 1 and the simulations results are illustrated. From the illustration it is clear that more the number of flows is there during transformation higher the bustiness at the subsequent node will become.

Since the extra latency is considered after each packet is being serviced by the server the gap between the curves of the theorem and the simulations are larger.But in actual scenario this latency will be much smaller This way of treatment extends the sensitivity of the derived results, more the number of flows that passes the ratio of loosing the tightness of bound will also increase.

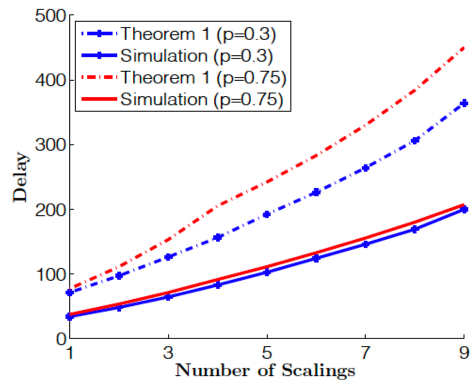

Fig.4. Delay bounds with Theorem 1 and the simulation.

V. Conclusion

In this research we have proposed a mathematical model for network layer operations of UWSN for usage in monitoring of coral bleaching. As a main contribution of this work we have extended Stochastic Network Calculus for deriving the delay bounds of variable length packet under flow transformations. We have defined the scaling element using SNC that works at operated at packet level rather than bit level which most of the research in the past have concentrated on. With the proposed model we have derived the end to end delay bounds of the packets of variable lengths under flow transformations and the results of the same is evaluated for its effectiveness using discrete event simulations. The model based on the SNC with new scaling element have very closer results with respect to the simulations. However improvement of the delay bound tightness remains the challenge especially in the case of variable length packet flow transformations. But we will focus on the improvement of the same using more accurate representation of packetized service and dynamic server in future.

REFERENCES

[1]. Fidler, M., & Schmitt, J. B. (2006, June). On the way to a distributed systems calculus: An end-to-end network calculus with data scaling. In *Proceedings of the joint international conference on Measurement and modeling of computer systems* (pp. 287-298).

[2]. Ciucu, F., Schmitt, J., & Wang, H. (2011, April). On expressing networks with flow transformations in convolution-form. In *2011 Proceedings IEEE INFOCOM* (pp. 1979-1987). IEEE.

[3]. Le Boudec, J. Y., & Thiran, P. (Eds.). (2001). *Network calculus: a theory of deterministic queuing systems for the internet*. Berlin, Heidelberg: Springer Berlin Heidelberg.

[4]. Chang, C. S. (2000). *Performance guarantees in communication networks*. Springer Science & Business Media.

[5]. Liebeherr, J., Burchard, A., & Ciucu, F. (2012). Delay bounds in communication networks with heavy-tailed and self-similar traffic. *IEEE Transactions on Information Theory*, *58*(2), 1010-1024.

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

[7]. Wang, H., Ciucu, F., & Schmitt, J. (2012, October). A leftover service curve approach to analyze demultiplexing in queueing networks. In *6th International ICST Conference on Performance Evaluation Methodologies and Tools* (pp. 168-177). IEEE.