PERFORMANCE ANALYSIS PROBLEMS IN TSN NETWORKS

Umar Nisar

***Abstract*—Time Sensitive Networking (TSN) is a collection of standards aimed at improving the real time peformance of ethernet interface. The TSN standards, however, lack the formal latency guarantees. For that different models are proposed employing methodologies such as Network Calculus, Compositional Performance Analysis and Machine Learning. In this paper we review different performance analysis techniques at our disposal and draw comparison with respect to their methodology, completeness, accuracy and scalability.**

***Index Terms*—TSN, Performance Analysis, Machine Learning,**

I. INTRODUCTION

THe latency requirement for industrial applications are of the order of a few millisecond, traditional networks, however, have latency of tens of millisecond [1]. Time Sensitive Networking (TSN) task group [2] is aimed at standardizing Ultra-Low Latency (ULL) network stack on the standard ethernet interface, in order to incorporate real-time awarness on to an otherwise time agnostic ethernet data link layer protcols. There is a need of comprehensive timing analysis of such protcols to determine schedulability of dynamic time-sensitive traffic in a TSN. There are several approaches that have been explored in order to provide the scheduling entity with the knowledge on the schedulability of a certain traffic. These methods can be broadly categorized in to two categories, mathematical based approach and simulation based approach. Mathematical approaches formulate a time aware mathematical model of a TSN protocol in order to compute latency and other characterstics such as maximum buffer size of scheduled traffic given the network traffic scenario and topology. A simulation model mimics the behaviour of the real system through a set of rules. Simulation based approach has several drawbacks such as it does not usually show the worst-case latencies and design space exploration is cumbersome. In section II we review TSN standards. In section III we review two mathematical models primarily for Audio Video Bridging (AVB) traffic in a TSN network. Furthermore, we also review a Machine Learning (ML) based approach to determine schedulability of AVB traffic.

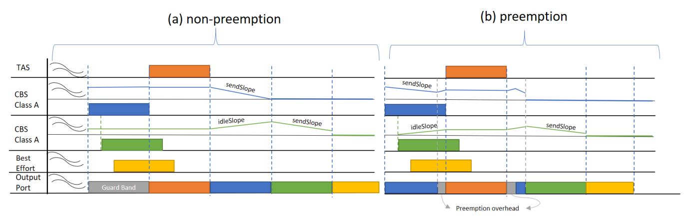

Fig. 1. Flow control at an output port employing TSN CBS and TAS with and without preemption

II. BACKGROUND

TSN employs different standards that are beriefly reviewed here in order to understand the flow control within a node of a TSN network. Reader interested in a more detailed survery can consult [3].

A. TSN Concepts

A unicast or multicast connection from one end station (talker) to another end station (listener) is called as a *flow*, or a *stream*.

In [3], the TSN standards are classified under properties of the flow, such as flow synchronization, flow management, flow control and flow integrity. Performance analysis or schedulability although considers different standards associated with the flow holistically but it is more focussed on standards associated with the flow control. In the following we take a brief look on TSN flow control.

1) Flow Control: The flow control in general deals with the schedule of the flows in order to comply with the QoS requirements primarily latency.

There are different standards that play their role in scheduling different classes of traffic in a TSN.

IEEE 802.1Qav [4] deals with queuing and forwarding of time sensitive streams. The standard divides the time sensitive traffic in to two classes i-e Class A (tight Delay Bound) and Class B (Lose Delay Bound). It also employs Credit Based Shaper (CBS) in order to avoid starvation of Best Effort (BE) traffic that does not have delay bound. CBS is a credit-based mechanism that only allows a stream to transmit if it has non-zero, positive credit. The CBS is governed by idle slope and send slope. The CBS of a particular class is increased by idle slope if there is at least one frame (of that class) in waiting and is decreased by send slope during the transmission.

Time Aware Shaper (TAS) [5] make use of Gate Controlled List (GCL) for a queue and it make use of global clock synchronization [6]. GCLs dictate opening and closing of the gate for each queue of output port in a bridge. The gate is opened for low priority traffic if it can be closed before transmission of TAS queues. The timer period of no-activity at the gate before time critical frame is transmitted is referred to as guard band.

IEEE 802.1Qbu [7] introduces low priority traffic frame preemption with an overhead. The guard band here gets reduced to the shortest low priority frame fragment.

III. Analysis

As discussed, TSN employs broad range of standards aimed at different use-cases with stringent latency requirements. In the flow control in order to schedule a packet dynamically, the scheduling entity needs to determine the schedulability of the frame i-e the latency of scheduled packet should be less than the worst case latency for the traffic. In the presence of multitude TSN standards at a node the analysis to determine the schdulability becomes involved. In this section, we review three different techniques aimed at the timing analysis primarily latency of dynamic traffic of a network.

A. ML Based Analysis

This methodology significantly differs from the other techniques discussed here as it does not explore underlying mathematical timing analysis to determine schedulability of a configuration. The paper [8] develops a machine learning model to ascertain feasibility of different configurations of a real time ethernet network as available under TSN standards. It goes on to compare performance metric, i-e accuracy and computation time, of the developed model with two network calculus based models referred to as approximate and precise in the paper.

1) Model: Four TSN based configurations are considered while evaluating feasible configurations for a real time ethernet network given the traffic and their associate timing constraints. The configuration are named as FIFO, Manual, Concise Priority (CP8) and Manual with traffic shaping, for details consult [8]. The configuration are proposed while keeping TSN standards in mind while consorting to the standard static priority scheduling algorithms in order to compare the ML model results with so called “approximate analysis” and “precise analysis based off [9]–[11].

A generic topology of network is given for which three classes of traffic is scheduled namely Audio, Video and Command and Control (C&C). The ML model that is used to determine feasibility of a configuration is K Nearest Neighbors (KNN), K being an arbitrary number of neighbors. The learning complexity of the model is O(N), N being the training data set, as it only needs to store the training data set. Each member of training data set of length N, is D dimensional, D being the number of features of a configuration having maximum impact on the end result i-e if a particular configuration having certain values of D features is feasible to schedule or not. The decision sought from the ML model is binary i-e schedulable or not schedulable. In the training phase of the model, the labelled training data is stored in an N x D array, for instance. The decision of the feasibility against these data indices is stored in a N dimensional array where entries 0 and 1 represent non-feasibility and feasibility respectively. Once a prediction against a new configuration, i-e a D dimensional data point, is sought from the ML model, an L2 norm of the data point is calculated from the N data points stored as training data set and K closest neighbors of the data point are identified. Based on majority voting of feasibility status of K neighbors, the decision of feasible or not feasible is taken against the new configuration. Four such models are developed against the solutions proposed earlier. The computational complexity here is of order of O(NKD) for each new configuration. In order to have accurate prediction, the training data set should be sufficient to populate D dimensional space of data points adequately. The trade off here is clear; more features of a configuration mean more apt characterization of the configuration; however, this also implies that more data points are needed to populate now higher dimensional space of the data set in order to have accurate predictions. This phenomenon is referred to as curse of dimensionality. In the ML model developed in the [8], following five features of a configuration are selected based on the empirical results: the number of critical flows, the number of audio flows, the number of video flows, the maximum load of the network and Gini Index. Gini Index [12] is a quantitative characterization of unbalancedness of link loads.

2) Results: The training data set is generated by labelling randomly generated data of the given topology by the accurate analysis in order to avoid false negatives. The hyperparameters such as K neighbors, K fold cross validation are fine-tuned by experiments. A training data set of N=5200 was deemed optimum against the accuracy, where any increases in N afterwards plateaued the accuracy curve. As can be seen in Table 2 of [8] the overall accuracy reduces with increasingly complex model. True Positive Rate (TPR) captures the percentage of accurate prediction of the overall prediction deemed to be feasible by the model. The TNR is just the converse of TPR. The lower percentage of TPR in the case of FIFO is correlated by Kappa Index and F Index. The reason behind that is low coverage of D dimensional space for the case of feasible solutions. As most of the training data set contained not feasible solutions indicated by F index for the case of FIFO scheduling. The figure 6 in [8] shows the comparison of the performance of ML model against the "approximate" and "accurate" analysis in the case of Manual Scheduling, the accuracy of the model is slightly worse than the two other two analysis. However, there is a significant difference of computation time only for the case $10\hat{6}$ TSN configurations. The table 3 of [8] summarizes the accuracy result and make a comparison against the other two analyses. The interesting point to note is that the analyses do not yield False Positive Rate as opposed to the ML model, which might indicate potential room of improvement in the Approx. analysis as it is pessimistic. For the case of CP8 there

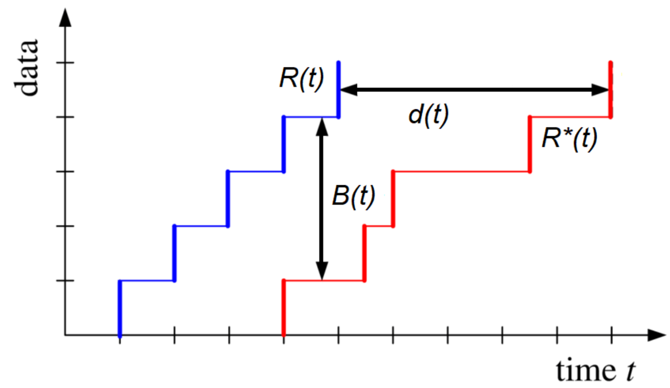

Fig. 2. Arrival and Departure Functions at an arbitrary node

is an improvement of overall accuracy in the case of ML model

3) Conclusion: The paper was premised around infeasibility of computational time in the case of accurate analysis however the choice of ML model is not consistent with the premise, since the prediction time of the model will increase linearly with the training data set. Another potential improvement could be feature selection, there is no evidence of different features selection for different schedulability in the paper, as can be seen in table that the accuracy has degraded with more complicated solution. Perhaps different features selection for each solution would customize the characterization of a configuration with respect to that solution and may drive the overall accuracy higher.

Furthermore, it does not provide a worst-case estimate on the latencies which remains a major draw back even for a relatively accurate model. As it does not guarantee a worst-case latency, hence it is best suitable as a verification method for the timing constraints.

B. Network Calculus Based Analysis

Network calculus is a toolset that is used in deterministic networking to compute bound on queuing, delays, buffers etc. In this paper [13] Network Calculus is used to derive an upper bound on the latency of AVB flow of TSN, which can then be used to ascertain shcedulability of the traffic.

1) Network Calculus Background: In network calculus, a system is characterized by two important functions that are arrival function $R(t)$ and departure function $R^*(t)$ as shown in the figure 2. The arrival function $R(t)$ is bounded by an arrival curve

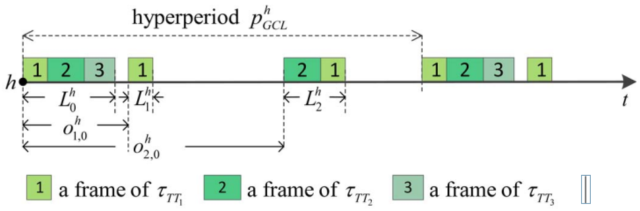

Fig. 3. An example of scheduled TT traffic at an output port h

α, which essentially is a limit on the incoming flow of the data. Similarly departure function $R^*(t)$ is bounded by a service curve β. The backlog $B(t)$ is simply a subtraction of arrival function $R(t)$ and departure function$R^*(t)$ at a given point in time. The latency is bounded by maximum horizontal deviation between the graphs of arrival function and departure function as shown in the figure 2

The paper primarily derives expressions for arrival and service curve in order to derive a bound on arrival and departure function which then can be used to compute latency of a particular flow.

2) Model: The worst case analysis for an AVB traffic flow in presence of TT traffic is carried out by calculating the arrival and service curves of the flow. As TT traffic takes priority over any AVB flow, the paper ingeniously makes use of the arrival curves of TT traffic in order to compute service curve of the AVB flow. This makes sense because when a node is not engaged in TT traffic and credit of the particular flow is positive only then the traffic of that flow is scheduled at an output port. Since TT traffic is scheduled offline assuming in a periodic fashion as show in figure 3, this period is referred to as hyperperiod, the arrival curve for TT traffic of a particular port h would simply be the summation of the bits transferred in each TT traffic window in a hyperperiod. The equation below summarizes the intuition:

$$\alpha^h_{TT,i}(i) = \sum_{j=i}^{i+N^h-1} L^h_j.C.\left\lceil \frac{t - o^h_{j,i}}{\rho^h_{GCL}} \right\rceil \quad (1)$$

The equation represents an aggregate of arrival curves where each arrival curve is with reference to each TT traffic window.

In the case of absence of IEEE 802.1Qbu, i-e no preemption of the low priority frames, the guard period needs to be catered to in the arrival curve. For the worst case the length of guard band would be either the maximum transmission time of AVB frames or just idle period if that is shorter than the aforementioned time period. Each guard band is appended in front of each TT traffic window. Following the same logic as before, the guards bands can be nicely assimilated in to the equation by extending and adjusting the time windows of TT traffic by their corresponding guard bands as captured in the equation of arrival curve for GB and TT given by as follows:

$$\alpha^h_{GB+TT,i}(i) = \sum_{j=i}^{i+N^h-1} (L^h_j + L^h_{GB,j}).C.\left\lceil \frac{t - o^h_{j,i} + L^h_{GB,j} - L^h_{GB,i}}{\rho^h_{GCL}} \right\rceil$$

Now, considering an AVB flow of class M, an interval Δt at an output port h can be broken down as: $\Delta t = \Delta t^+ + \Delta t^- + \Delta t^0$ where Δt^+ refers to the time spent in waiting, Δt^- is the time duration of the transmission and Δt^0 refers to the time duration of TT and GB windows. As service curve is a bound on the departure function, it can be well characterized by bits served in Δt^-. This knowledge along with arrival curves of the TT and GB windows help derive the service curve for the AVB flow of class M.

The arrival curves of the AVB flow are derived by the characteristics of the traffic associated with that flow and is expressed as summation of the bursts, σ, associated with class M in an output port h and their long-term rate of arrival, ρ, expressed as:

$$\alpha^h_{AVB\_M}(t) = \sum_{\tau_{AVB\_M_k} \in h} \sigma^h_{AVB\_M_k} + \sum_{\tau_{AVB\_M_k} \in h} \rho^h_{AVB\_M_k}.t \quad (2)$$

As discussed earlier, the maximum horizontal deviation of arrival and service curve of an output port h draws an upper bound on the latency, $D^h_{AVB\_M_k,i}$. The arrival curve of subsequent nodes can be estimated by summing the arrival curve of the current node with upper bound of latency, D, for that node. Since service curve is dependent on reference TT windows, i, this yield multiple upper bounds for the latency of the flow $\tau_{AVB\_M_k}$, the worst-case upper bound is the maximum latency, $\max_i D^h_{AVB\_M_k,i}$ for the output port h.

3) Results: The paper has incorporated the analysis on a topology of 31 End Switches (ES), 15 SWs and 39 routers connected by dataflow links transmitting at 1 Gbps. In order to show the scalability of the analysis, the Worst Case Delays (WCD) of 34 AVB flows are compared against 15 and 100 TT flows. For each AVB flow the WCD in presence of 100 TT flows was greater than that of 15 TT flow, which makes sense since the service curve of AVB flow reduces with an increased value of arrival curve of TT and GB.

Similarly, the analysis goes on to show that WCD in the case of non-premption of AVB flows are always greated than premption mode due to the penalty incurred by GB. The analysis also noted that the WCD of AVB flow reduce with an increased value of idle slope. However, the degree of decrease in WCD reduces at a certain point yielding diminished returns. The analysis also explore the effect of the routing path in a TSN topology and discovers significant impact of routing on the WCD of AVB flows.

4) Conclusion: The paper derives expression for WCD of AVB flows in the presence of TT traffic by empoying NC toolset. The analysis captures the behaviour of a TSN traffic in a node and results shown are consistent with that behaviour. The analysis presented in the paper is comprehensive as it incorporates TT traffic along with premption and non-premption mode. The analysis claims to remove the pessismism of the results obtained in [14].

C. Compositional Performance Analysis (CPA) of AVB Ethernet Traffic

This technique [15] differs from the NC one primarily because of the mathematical approach towards tackling the timing analysis of the network traffic. This paper makes use of CPA toolset to derive an upper bound on the latency of a traffic stream of class AVB.

1) CPA Background: CPA model is composed of a set of tasks that are then processed by the a set of resources. For example an output port is a resource that executes tranmission of different flows in a network that are reqarded as tasks. Each task, τ_i, is activated by their respective events, q, which are frames of a particular stream that keep the task busy referred to as busy time $B(q)$, part of the busy time is the execution time of the task on the resource associated with the task. Each task has a profile of events referred to as maximum and mimimum arrival curves, these arrival curves provide the information regarding the arrival of the events associated with the task in a given time interval.

2) CPA based Analysis: Busy time activated by q event is given by the equation:

$$B_i^+(q, a_i^q) \leq t_{transfer}(q) + I_{LPB} + I_{SPB}(a_i^q) + \\ I_{TSB}(a_i^q) + I_{HPB}(B_i^+(q, a_i^q))$$

where: $t_{transfer}(q) = q.C_i^+$ is maximum tranfer time for back to back q events and C is the time it takes for a frame to transmit at an output port. $I_{LPB} = \max_{j \in lp(i)} C_j^+$ is worst-case time taken by a lower priority task that just started execution before arrival of the q-event. $I_{SPB}(a_i^q) = \sum_{j \in sp(i)} (\eta_j^+(a_i^q).C_j^+)$ is worst-case time taken by same priority events of task τ_j arrived until arrival time of q-event of τ_i given by a_i^q. $I_{HPB}(B_i^+(q, a_i^q)) = \sum_{j \in hp(i)} (\eta_j^+(B_i^+(q, a_i^q) - C_i^+).C_j^+)$ is worst-case time taken by a high priority task, τ_j, following the same logic as same priority task however the subtraction of C_i^+ is due to the non-premption nature of onging low priority task, I_{TSB} is worst-case time taken by traffic shaper or insufficient credit which is dependent on other terms of the expression as blocking of τ_i due to any reason would result in an increase of the credit. For details consult [15] The upper bound on latency, l_p^+, of stream p is given by the summation of the worst-case response time, R_i^+, at each node along the path. The derived equation is as follows:

$$l_p^+ \leq \delta^-_{first(p)}(s) + \sum_{j \in Tasks(p)} R_j^+ + O_{routing}(p)$$

where $\delta^-_{first(p)}(s)$ is the worst-case arrival interval between the last and the first frame and $O_{routing}$ is the path delay associated between the nodes. In order to compute worst-case response time, R_i^+, at each node one has to find the worst-case time interval between the foremost q event arrival and the last q event departure of the same busy time characterized by $R_i^+ = \max_{q \in Q_i} \{ \max_{a_i^q \in A_i} \{B_i^+(q, a_i^q) - a_i^q\}\}$. Different values of arrival times a_i^q of qth event have to be considered which in return affect the busy time, leading to fixed point iterations.

So far in the analysis, the effect of traffic shaper has not been incorporated into the calculation of arrival times of the events, incorporating such could reduce interference of same priority tasks vying for the same resource. For that, an upper bound on the traffic shaped by a shaper on a resource is computed which is then incorporated in to the output event model for each task. For detailed derivation consult [15].

3) Results: Having developed the model for AVB traffic, the paper goes on to evaluate the model on different topologies by comparing the worst-case latencies of AVB traffic with that of strict priority. A general trend of much higer latencies in the case of AVB are observed which is explainable due to the maximum transmission bound imposed by shaper credit. It was seen that setting the idle slope of AVB traffic 31 times than the required bandwidth achieves the same latency as of strict priority. It was noted that in the case of linear topology the rate of increase of latencies with respect to the number of nodes was much higher than that observed in star topology. This is due to the intermediate shapers at each node that accumulated the effect of shaper at the output node.

4) Conclusion: The paper makes use of CPA analysis to compose a model of AVB ethernet traffic and evaluates the result in different topologies. The model is shown to be scalable with the plausible results.

However, with respect to TSN standards the analysis leaves much to be desired as it does not: 1) incorporate premption of low priority frames as standarized in IEEE 802.1Qbu, 2) explicitly include effect of TT traffic on to the credit shaper.

IV. Conlcusion

We have reviewed TSN standards particularly associated with the flow control. We then discussed three different approaches aimed at providing the schedulability analysis of AVB traffic flow in a TSN network. We have discussed and compared the underlying mathematical principles of the aforementioned approaches. Furthermore, we also highlighted the shortcomings in the completeness of the analyses with regards to TSN standards.

References

[1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[2] "Time-sensitive networking (tsn) task group." [Online]. Available: https://1.ieee802.org/tsn/

[3] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-low latency (ull) networks: The ieee tsn and ietf detnet standards and related 5g ull research," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 88–145, 2019.

[4] "Ieee standard for local and metropolitan area networks - virtual bridged local area networks amendment 12: Forwarding and queuing enhancements for time-sensitive streams," *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)*, pp. C1–72, 2010.

[5] "Ieee standard for local and metropolitan area networks – bridges and bridged networks - amendment 25: Enhancements for scheduled traffic," *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015)*, pp. 1–57, 2016.

[6] "Ieee draft standard for local and metropolitan area networks - timing and synchronization for time-sensitive applications in bridged local area networks," *IEEE P802.1AS/D7.7, November 2010*, pp. 1–296, 2010.

[7] "Ieee standard for local and metropolitan area networks – bridges and bridged networks – amendment 26: Frame preemption," *IEEE Std 802.1Qbu-2016 (Amendment to IEEE Std 802.1Q-2014)*, pp. 1–52, 2016.

[8] T. Mai, N. Navet, and J. Migge, "On the use of supervised machine learning for assessing schedulability: application to ethernet tsn," 11 2019, pp. 143–153.

[9] A. Bouillard and r. Thierry, "An algorithmic toolbox for network calculus," *Discrete Event Dynamic Systems*, vol. 18, no. 1, p. 3–49, 2007.

[10] M. Boyer, J. Migge, and N. Navet, "An efficient and simple class of functions to model arrival curve of packetised flows," 11 2011.

[11] R. Queck, "Analysis of ethernet avb for automotive networks using network calculus," in *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, 2012, pp. 61–67.

[12] F. Farris, "The gini index and measures of inequality," *American Mathematical Monthly*, vol. 117, pp. 851–864, 12 2010.

[13] L. Zhao, P. Pop, Z. Zheng, and Q. Li, "Timing analysis of avb traffic in tsn networks using network calculus," in *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2018, pp. 25–36.

[14] S. M. Laursen, P. Pop, and W. Steiner, "Routing optimization of avb streams in tsn networks," *SIGBED Rev.*, vol. 13, no. 4, p. 43–48, Nov. 2016. [Online]. Available: https://doi.org/10.1145/3015037.3015044

[15] J. Diemer, D. Thiele, and R. Ernst, "Formal worst-case timing analysis of ethernet topologies with strict-priority and avb switching," in *7th IEEE International Symposium on Industrial Embedded Systems (SIES'12)*, 2012, pp. 1–10.