Applying Quality of Service Prediction in WDM Optical Networks

Panagiotis G. Sarigiannidis
Department of Engineering Informatics and Telecommunications
University of Western Macedonia
Kozani, Greece
sarpan@uowm.gr

***Abstract*—A dynamic prediction scheme is presented in this paper, named PROPHET. The purpose of the proposed technique is to predict, given two different classes of quality of service (QoS), the total amount of the demanded transmission requests per QoS class. PROPHET is constructed based on Hidden Markov Chains (HMC), modeled by an ergodic framework. The prediction objective is to reduce the amount of time spent in computing the transmission schedule by predicting traffic requests. The evaluation of the predictor is realized in a Wavelength Division Multiplexing (WDM) single-hop network with star topology. Furthermore, PROPHET is compared to a previous prediction-based scheme, called POSA. Simulation results indicate that the novel technique supports efficient predictable QoS, since it operates more accurately than POSA.**

Keywords-prediction; WDM networks; optical networks; scheduling.

I. INTRODUCTION

Network traffic prediction is an important and valuable issue in network management and configuration. Accurate predictions may lead to various benefits such as time benefit regarding transmission scheduling, utilization benefits in a medium exploitation manner, throughtput improvements in a network design etc [1]-[7]. Usually, most of the traffic prediction models do not take into account QoS matters, such as the differentiation of the packets' priority. In this work, a novel prediction technique is presented, which supports QoS. The introduced PRiOritized Prediction Hidden-chained Ergodic Technique (PROPHET) constitutes a novel way of introducing estimation procedure by applying a distributed Hidden Marcov Chains (HMCs) [8]-[9], in order to predict the upcoming nodes' requests per service class. The suggested scheme is evaluated into a single-hop WDM optical network with star topology. A reservation-based Multiple Access Control (MAC) protocol is applied in order to address the channel allocation process [10]-[11], by creating a transmission program, known as transmission schedule. PROPHET tries to predict the traffic requests that each node sends to all other in order to coordinate the transmissions without any collisions. The main purpose of the PROPHET framework is to provide the (predicted) data requests in advance, enabling the overlapping between the schedule computation phase and the data transmission phase. In this way, the schedule computed during the current data phase (let it be frame f) is used for the next data phase (frame f+1). The result of this overlapping is the reduction of the schedule computation time, earning less spent control time, solving a crucial problem for optical access networks. The advantage of PROPHET scheme is that supports QoS, by predicting separately the real- and non-real traffic requests. In this context the performance of the prediction modules is counted in terms of prediction accuracy. In other words, the prediction accuracy emerges from the portion of the number of successful predictions divided by the total predictions. Simulation results indicate that the novel QoS-based prediction framework supports more accurate estimations than the previous POSA scheme [12]. The rest of this paper is organized as follows. Section II describes the network background, Section III introduces the PROPHET framework, Section IV shows and makes remarks about the simulation results, and finally Section V concludes the letter.

II. NETWORK MODEL

In the adopted WDM star network, a set of N nodes are interconnecting to a passive star coupler via two-way optical fibers [13]-[15]. Each node is equipped with a tunable transmitter and a fixed receiver in order to send data packets on any available channel and to accept data on a specific channel, known as home channel. The network is supported by W data channels of the same bandwidth, while there is no separate control channel. For a more realistic scenario it is considered that the number of nodes is larger than the number of channels ($N > W$), hence N/W nodes share the same home channel.

The system is synchronous and so the network operates in a slotted mode with a timeslot equal to the transmission time of one fixed-length packet. Each node maintains W queues, one queue per channel, with upper limit equal to K (fixed-length packets) in order to simulate a deterministic prediction model. In this way, a request, consisting of a variable-length packet, is generated at a node and it is sorted in the corresponding queue according to the destination's home channel. A scheduling algorithm functions in a distributed manner, so each node broadcasts control packets to all channels in a TDM-based fashion. This phase is called reservation phase. Next, the scheduling algorithm calculates the transmission schedule and the data phase is followed. During the data phase nodes apply the common schedule and transmit their packets on the appropriate channel. At the end of the data phase, the current frame (transmission cycle) comes to an end, introducing the following frame.

As said before, the distributed algorithm accepts the requests of each node and stores them in a matrix $D = [d_{i,j}]$, called traffic demand matrix. The matrix has N rows and W columns, hence, the entry stored in i (i belongs to N) row and j (j belongs to W) column contains the amount of packets, which i node requests to transmit on the j channel, as j channel is the home channel of the destination node. Upon the beginning of each frame, all nodes run the same distributed scheduling algorithm, based on the same global information. Thus, the algorithm can be able to decide how transmissions and receptions for the next phase should be made in order to form a common schedule for all.

III. The Prediction Framework

The proposed prediction framework is designed to predict the real and non-real traffic demands of each node. In this work, two individual traffic streams are considered in order to crystallize the prediction model in a simple way. Eventually, the suggested framework can be easily expanded to cover more different priority traffic streams. Currently, two traffic matrices are defined, the D_h traffic matrix, which stores the real traffic load and the D_l traffic matrix, storing the non-real traffic stream. The goal of the overall scheme is to accurately predict both real D_h and non-real D_l traffic matrices. Therefore, the overall predictor consists of 2NW individual predictors. In order to model such a framework a set of K distinct states are considered. Each state denotes the amount of request traffic for each transmission frame. Fig. 1 shows an HMC model paradigm with four states. Hence, the k state, where $0 \le k \le K$, represents the number of requested packets. In our system a set of 2NW(K+1) distinct states are considered, which denote the number (z) of real (h) or non-real time (l) requests that node i demands on channel j at frame t, denoted as f_t as follow:

$${}^{h}S_z^{i,j} = \{{}^{h}S_0^{i,j}, {}^{h}S_1^{i,j}, \ldots, {}^{h}S_K^{i,j}\},$$

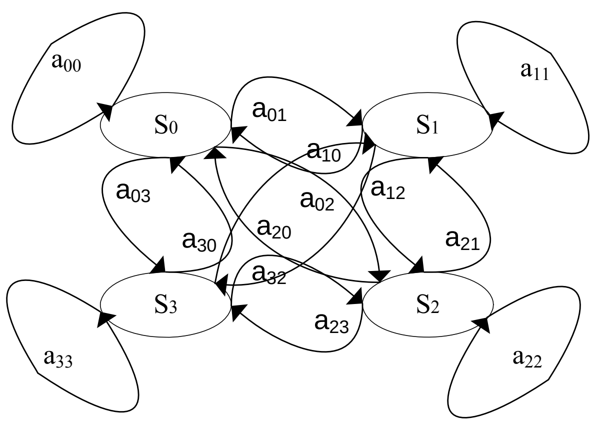

Figure 1. An indicative ergodic HMC model with four states.

and

$${}^{l}S_z^{i,j} = \{{}^{l}S_0^{i,j}, {}^{l}S_1^{i,j}, \ldots, {}^{l}S_K^{i,j}\},$$

where $1 \le i \le N$, $1 \le j \le W$, $0 \le z \le K$.

For example, at frame f_{120} state ${}^{h}S_4^{3,2}$ denotes that node 3 requests 4 real-time packets on channel 2 during the frame 120. Also, the state transition probability distribution is considered, as follows:

$${}^{h}A_{u,y}^{i,j} = P[\, f_t = {}^{h}S_y^{i,j} \mid f_{t-1} = {}^{h}S_u^{i,j}\,],$$

and

$${}^{l}A_{u,y}^{i,j} = P[\, f_t = {}^{l}S_y^{i,j} \mid f_{t-1} = {}^{l}S_u^{i,j}\,],$$

Where $1 \le i \le N$, $1 \le j \le W$, $0 \le u,y \le K$, while t denotes the current frame. For example, at frame f_{120} transition probability ${}^{h}A_{4,1}^{3,2}$ refers to the transition from state ${}^{h}S_4^{3,2}$ (frame 119) to state ${}^{h}S_1^{3,2}$ (frame 120). In particular, this transition concerns node 3 on channel 2, which requests 4 real-time packets, during frame 120, while the same node on the same channel requested 1 real-time packet during the previous frame. The target is to determine the transition probabilities for each node and for each channel in order to predict the traffic matrices D_h and D_l for the next frame. The output of each predictor is the most probable transition, i.e., the transition with the greatest probability. The most probable transition is decided according to the recorded transmission requests. For that purpose a set of history vectors are considered for each distinct state to store the past transmission requests with size equal to V entries:

$${}^{h}H_z^{i,j}(g) = \{{}^{h}H_z^{i,j}(1), {}^{h}H_z^{i,j}(2), \ldots, {}^{h}H_z^{i,j}(V)\},$$

and

$${}^{l}H_z^{i,j}(g) = \{{}^{l}H_z^{i,j}(1), {}^{l}H_z^{i,j}(2), \ldots, {}^{l}H_z^{i,j}(V)\},$$

where $1 \le i \le N$, $1 \le j \le W$, $0 \le u,y \le K$, $1 \le g \le V$.

For example, entry ${}^{h}H_4^{3,2}(9) = 3$ means that node 3 on channel 2 moved from state ${}^{h}S_4^{3,2}$ to state ${}^{h}S_3^{3,2}$, 9 entries before. History vectors operate like first in first out queues, i.e., the first entry that comes is stored in the first place, while the last entry is deleted. All other entries are shifted one place to the end of the vector. In this way, these vectors monitor the past pattern traffic of the network. History vectors in conjunction with the current actual request define the predicted request for the following frame. At frame f_{t-1}, let the actual request for node i on channel j be ph_{t-1} real-time packets and pl_{t-1} non real-time packets. Also, consider that the actual requests for node i on channel j for the next frame f_t are ph_t and pl_t respectively. Then it holds that the active current states of node i on channel j for frame t are ${}^{h}S_{ph_t}^{i,j}$ and ${}^{l}S_{pl_t}^{i,j}$. The next step is the history vector update. History vectors regarding node i and channel j are updated as follows:

$$ {}^{h}H^{i,j}_{ph_{t-1}}(g) = {}^{h}H^{i,j}_{ph_{t-1}}(g-1), $$

$$ {}^{l}H^{i,j}_{pl_{t-1}}(g) = {}^{l}H^{i,j}_{pl_{t-1}}(g-1), $$

for each g, where $2 \le g \le V$ and

$$ {}^{h}H^{i,j}_{ph_{t-1}}(1) = ph_t, $$

$$ {}^{l}H^{i,j}_{pl_{t-1}}(1) = pl_t, $$

Then the corresponding transition probabilities are updated, according to history vectors:

$$ {}^{h}A^{i,j}_{ph_{t-1},y} = \frac{number.of.recorded.entries.in.{}^{h}H^{i,j}_{ph_{t-1}}.vector.equal.to.y}{V} $$

$$ {}^{l}A^{i,j}_{pl_{t-1},y} = \frac{number.of.recorded.entries.in.{}^{l}H^{i,j}_{pl_{t-1}}.vector.equal.to.y}{V} $$

for each y, where $0 \le y \le K$.

For example, consider that at frame 120 the actual request for node 3 on channel 2 is 4 real-time packets and at frame 121 the same node on the same channel requests 1 real-time packet. Furthermore, let the size of the history vectors be equal to100 entries. History records are updated, hence ${}^{h}H^{3,2}_{4}(g) = {}^{h}H^{3,2}_{4}(g-1)$ for each g, where $2 \le g \le 100$ and ${}^{h}H^{3,2}_{4}(1) = 1$. Next, the transition probabilities are changed. Assuming that K = 4 and the number of recorded entries for state 0 is 12, for state 1 is 23, for state 2 is 33, for state 3 is 8, and for state 4 is 23, it holds that the corresponding transition probabilities are:

$$ {}^{h}A^{3,2}_{4,0} = 0{,}12, {}^{h}A^{3,2}_{4,1} = 0{,}23, {}^{h}A^{3,2}_{4,2} = 0{,}33, {}^{h}A^{3,2}_{4,3} = 0{,}08, and\ {}^{h}A^{3,2}_{4,4} = 0{,}23 $$

Finally, during frame f_t the predictor chooses the most probable transition state for frame f_{t+1}, as follows:

$$ D_h(i,j) = \arg\max_{0 \le x \le K}[{}^{h}A^{i,j}_{ph_t,x}] $$

$$ D_l(i,j) = \arg\max_{0 \le x \le K}[{}^{l}A^{i,j}_{pl_t,x}] $$

For instance, supposing that for node 3 on channel 2 the corresponding transition probabilities during frame 121 are: ${}^{h}A^{3,2}_{1,0} = 0{,}15, {}^{h}A^{3,2}_{1,1} = 0{,}24, {}^{h}A^{3,2}_{1,2} = 0{,}11, {}^{h}A^{3,2}_{1,3} = 0{,}30, and\ {}^{h}A^{3,2}_{1,4} = 0{,}20$, hence the predictor selects the transition ${}^{h}A^{3,2}_{1,3}$ and predicts 3 real-traffic packets for the next frame. Obviously, the current actual request for node 3 on channel 2 is one real-time packet.

IV. Simulation Results

In this section the performance of the proposed prediction scheme is evaluated, in terms of accuracy via a set of simulation experiments. The following results show the accuracy level of individual predictions for both POSA and PROPHET schemes. Four sets of simulated experiments were conducted on both predictor modules in order to evaluate the accuracy of the predictions. Regarding the simulation parameters the following assumptions hold:

a) The packet arrivals follow a Bernoulli distribution, which produces one packet for each node destined to each (destination) channel with success probability p. For example, if p = 0,95 then it holds that for each timeslot it is 95% possible for one packet to be generated for each node destined to each channel.

b) For each generated packet there is a constant probability that indicates the priority of the packet. More specifically, for each generated packet there is 25% possibility to be real traffic packet and 75% to be non-real traffic packet, since.

c) For each experiment a total of 100.000 frames were generated, where the initial 100 frames were used as training frames, filling the history vector of each node for each channel, without producing predictions. During the remaining frames the prediction takes place and the accuracy results were recorded.

d) The accuracy of each prediction is evaluated, based on the error level of the prediction. The error level for a given individual prediction is given by the following equation [12]:

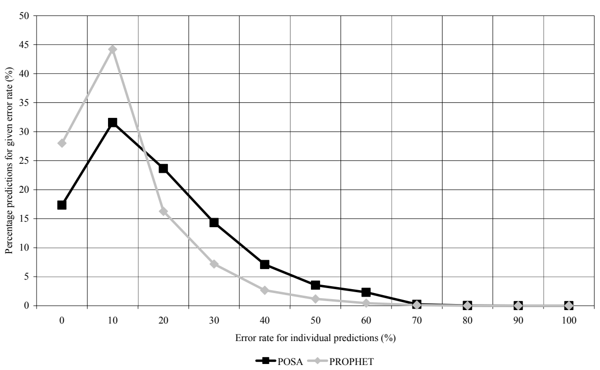

Figure 2. Prediction error rate for K = 10 and W = 10.

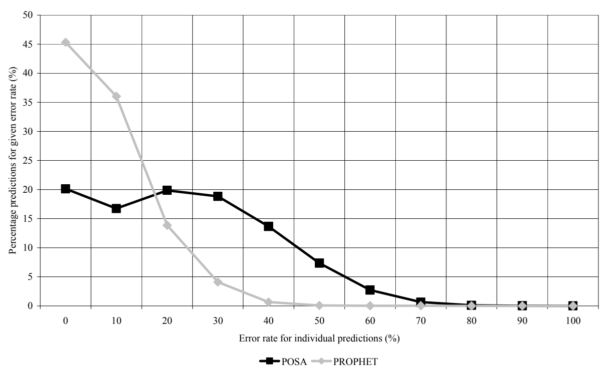

Figure 3. Prediction error rate for K = 20 and W = 10.

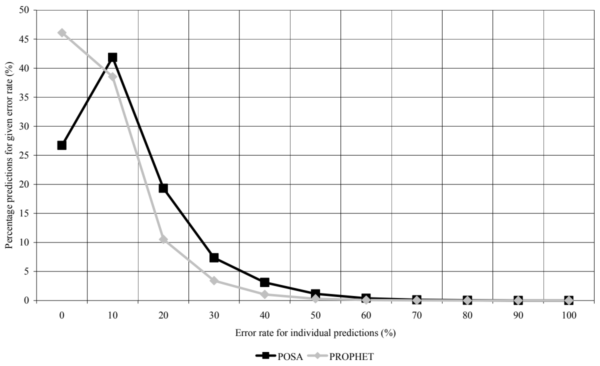

Figure 4. Prediction error rate for K = 10 and W = 5.

$$E_p = \frac{\left|\ell_p - \ell_a\right|}{\left|\ell_U\right|}$$

Where Ep is the error rate in prediction, ℓ_p is the state to which the individual predictor anticipates a change, ℓ_a is the actual state that the request of the traffic matrix changed to, and ℓ_U is the total number of states possible.

For each experiment the number of nodes varies from 10 to 100, while the channels are set 5 or 10. Also, the history vector has 100 entries for all experiments. Figure 2 shows the accuracy performance of the schemes with 10 channels, K = 10 and p = 0,973 or 97,3%. It is clear that PROPHET presents 28% predictions with 0% error, while POSA presents only 17% predictions with no error. In the same manner, PROPHET predicts 44% of predictions with a minor error of 10%, while for the same error level POSA ouputs 31% of predictions. In general, for this figure PROPHET's 72% predictions present an error level of less than 10%. In reverse, only 48% of POSA's predictions present an error level of less than 10%.

Figure 3 demonstrates the comparison of the two prediction schemes, in terms of accuracy for 10 channels, K = 20 and p = 0,973. Again, PROPHET predicts more accurately that POSA, since the rate of the 100% accurate predictions is 25% more than POSA and the rate of predictions that present an error level of less than 10% is 45% more than POSA.

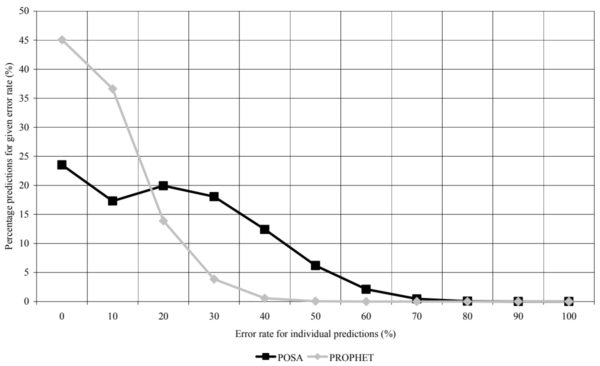

Figure 5. Prediction error rate for K = 20 and W = 5.

Next, the accuracy level is tested for less channels. In Fig. 4 the error rate is presented of both schemes with 5 channels, K = 10, and p = 0,97. Once more, PROPHET presents seriously more accurate predictions, since 10% more predictions are completely accurate than POSA. Also, 22% more predictions of PROPHET scheme have an error rate of less than 10% in comparison with POSA.

Finally, Fig. 5 depicts the accuracy level of POSA and PROPHET with 5 channels, K = 20 and p = 0,97. Obviously, PROPHET seems to be superior than POSA, in terms of accuracy, undependably of the number of channels or the traffic load. In this figure, the differences are more wide, since PROPHET presents 22% more 100% accurate predictions and 41% more predictions with error level of less than 10%, compared to POSA.

V. Conclusion

A dynamic prediction framework, called PROPHET, was presented in this paper, provisioning predicted QoS. The proposed scheme is modelled by an ergodic HMC framework. The logic is that the packet history tracks estimate the possible network behaviour. This dynamic scheme follows the traffic changes and tries to determine the traffic pattern. In this manner, predictions are provided for real and non real time packet requests in advance. The predictor is evaluated within a WDM optical access network, in which a scheduling algorithm calculates the transmission schedule for the next transmission frame. Simulation results are provided, which indicate that PROPHET improves a previous predictor, called POSA, and concurrently it supports QoS, operating more precisely, regarding the prediction error levels.

References

[1] J. A. McLinn, "Reliability predictions — more than the sum of the parts," *Reliability and Maintainability Symposium, 2008. RAMS 2008. Annual* , vol., no., pp.446-451, 28-31 Jan. 2008.

[2] H. Matsutani, M. Koibuchi, H. Amano, T. Yoshinaga, "Prediction router: Yet another low latency on-chip router architecture," *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on* , vol., no., pp.367-378, 14-18 Feb. 2009.

[3] Sang Aimin, Li San-qi Li, "A predictability analysis of network traffic," *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* , vol.1, no., pp.342-351 vol.1, 2000.

[4] Y. Sugaya, H. Tatsumi, M. Kobayashi, H. Aso, "Long-Term CPU Load Prediction System for Scheduling of Distributed Processes and its Implementation," *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on* , vol., no., pp.971-977, 25-28 March 2008.

[5] Peng Wang, Yuan Liu, "Network Traffic Prediction Based on Improved BP Wavelet Neural Network," *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on* , vol., no., pp.1-5, 12-14 Oct. 2008.

[6] Yu Xue-gang, Liu Yan-heng, Wei Da, Ting Min, "Hybrid Markov Models Used for Path Prediction," *Computer Communications and Networks, 2006. ICCCN 2006. Proceedings.15th International Conference on* , vol., no., pp.374-379, 9-11 Oct. 2006.

[7] C. Ghosh, C. Cordeiro, D. P. Agrawal, M. B. Rao, "Markov chain existence and Hidden Markov models in spectrum sensing," *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on* , vol., no., pp.1-6, 9-13 March 2009

[8] J. Dai, "Hybrid approach to speech recognition using hidden Markov models and Markov chains," *Vision, Image and Signal Processing, IEE Proceedings* - , vol.141, no.5, pp.273-279, Oct 1994.

[9] Liu Jia, Li Shunxiang, Jia Shusheng, "A prediction model based on neural network and fuzzy Markov chain," *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on* , vol., no., pp.790-793, 25-27 June 2008.

[10] B. Mukherjee, *Optical WDM Networks*. Springer, 2006.

[11] M. Maier, *Optical Switching Networks*. Cambridge, 2008.

[12] E. Johnson, M. Mishra, and K. M. Sivalingam, "Scheduling in optical WDM networks using hidden Markov chain based traffic prediction," Journal of Photonic Network Communications, vol. 3, no. 3, pp. 271-286, July 2001

[13] S. Petridou, P. Sarigiannidis, G. Papadimitriou, and A. Pomportsis, *" On the use of clustering algorithms for message scheduling in wdm star networks* ", IEEE Journal of Lightwave Technology, vol. 26, no. 17, pp. 2999–3010, 2008.

[14] P. Sarigiannidis, G. Papadimitriou, and A. Pomportsis, *" A high throughput scheduling technique, with idle timeslot elimination mechanism ",* IEEE Journal of Lightwave Technology, vol. 24, no. 12, pp. 4811–4827, 2006.

[15] P. Sarigiannidis, S. Petridou, G. Papadimitriou, M. Obaidat, and A. Pomportsis, " *Supporting quality-of-service scheduling in a tt-fr wdm system* ", IEEE Systems Journal, vol. 2, no. 4, pp. 525–535, 2008.