

MIT Open Access Articles

*Cognitive Management and Control of
Optical Networks in Dynamic Environments*

The MIT Faculty has made this article openly available. **Please share**
how this access benefits you. Your story matters.

Citation: Zheng, Anny Xijia and Vincent W. S. Chan. "Cognitive Management and Control of Optical Networks in Dynamic Environments." IEEE International Conference on Communications, June 2020, Dublin, Ireland, Institute of Electrical and Electronics Engineers, July 2020. © 2020 IEEE

As Published: <http://dx.doi.org/10.1109/icc40277.2020.9149338>

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: <https://hdl.handle.net/1721.1/131059>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



Cognitive Management and Control of Optical Networks in Dynamic Environments

Aanny Xijia Zheng, *Student Member*, Vincent W.S. Chan, *Life Fellow IEEE, Fellow OSA*

Claude E. Shannon Communication and Network Group, Research Laboratory of Electronics

Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge MA, USA

Emails: {xjzheng, chan}@mit.edu

Abstract—Emerging network traffic requires a more dynamic and more agile network management and control system than today's networks. We propose the use of cognitive techniques for the fast and adaptive management of future optical networks. As a first approximation, we model our traffic arrivals as a multi-state Markov process and categorize different network traffic environments by the length of the traffic coherence time. For the traffic with moderate and short coherence times, the stopping-trial estimator still responds to the traffic changes with a short detection time as long as the inter-arrival times of traffic transactions are independent, which is a very reasonable assumption. The algorithm provides no prejudice on the exact network traffic model avoiding having to sense and estimate detailed traffic statistics. The transition rates of the Markov model can change gently or abruptly due to the shift of the offered traffic's statistics, and a transient condition prevails until the system arrives at a new steady state. We model the transient behavior of such network traffic drifts towards convergence to a new steady state analytically and validate the feasibility and utility of the traffic prediction. When the network traffic rate changes quickly, our sequential maximum likelihood estimator will capture the traffic trend with a small number of arrivals and provide fast reconfiguration, which is very important for maintaining quality of service during large traffic shifts.

Index Terms—cognitive network, optical network, network management and control

I. INTRODUCTION

Emerging network applications will make the offered traffic in future optical networks more bursty with high granularity sessions from megabytes to terabytes. Thus, statistical multiplexing to smooth quiescent flows between major nodes will not occur most of the time. Such dynamic traffic will require a much smarter network management and control system to rapidly adapt to bursty application and their service needs, especially for the increasingly large-volumed "elephants". Meanwhile, the network control and management system should be kept simple and efficient enough to reduce the burdens of network operations. Cognitive techniques can provide fast, dynamic, and efficient network adaptations to meet the above requirements. The cognitive process senses current network conditions and reconfigure network resources in a timely manner with little or no human involvement.

This work is sponsored by NSF NeTS program, Grant No. 6936827, the HKUST/MIT programs, and Claude E. Shannon Research Assistantship.

Our goal is to develop the algorithms that can detect the traffic changes as quickly as possible and reconfigure accordingly in dynamic network traffic environments. Generally, traffic environments can be categorized into three regimes: 1. network traffic changes very slowly so adaption can be done while the traffic is in the same state; 2. network traffic changes at a moderate rate commensurate with the adaption times; 3. network traffic changes very rapidly so adaptations can only be limited to trends rather than the detail changes. A comprehensive analysis on such dynamic network environments is a must for the design of cognitive management and control of optical networks. Therefore, we focus on the design a traffic detection mechanism and the reconfiguration scheme for all three regimes in our work. Our previous work [1] has solved the first regime where the network traffic changes very slowly, showing the stopping-trial estimator has the shortest detection time when the session arrival statistics provide enough confidence for reconfiguration. In this paper, we further discuss the fast detection in the moderate and short network coherence time regimes with a general multi-state Markov traffic model. Our results show the stopping-trial estimator still reacts effectively to the traffic rate changes as long as the inter-arrival times of traffic sessions are independent. The beauty of the stopping-trial estimator is that it can trigger the reconfiguration without inferring the network traffic state statistics. To further deal with the fast-change traffic, we analyze the transient convergent behaviors of network traffic drift as a result of traffic transition rate changes. For a special case of fast-changing traffic where the traffic rate increases/decreases monotonically, we develop a sequential maximum likelihood estimator that can predict the traffic trend in the super-short network coherence time environment. It can sufficiently estimate the traffic trend with a reasonable number of arrivals and trigger reconfigurations.

II. TRAFFIC MODEL AND STOPPING-TRIAL ESTIMATOR

A. All-to-all Poisson Traffic

We will use a simple network topology (*e.g.* WAN) with a WDM tunnel architecture depicted in our previous work [1], where each source-destination pair is connected via a pre-configured set of wavelengths in a single path or over multiple

paths for traffic transmission. Every node pair is assumed to have all-to-all independent and identically distributed (I.I.D.) traffic. We assume the arrival traffic at the source forms a doubly stochastic Poisson point process with a time-dependent rate of $\lambda(t)$. Therefore, each inter-arrival time in $T_i, i \geq 1$ follows an exponential distribution with the parameter $\lambda(t)$. In our previous work [1], we assume $\lambda(t)$ switches between a non-surging state and a surging state as a two-state countable-state Markov process. In this work, We extend it into a general traffic rate model where the traffic arrival rate $\lambda(t)$ switches among different states $\lambda_1, \lambda_2, \dots, \lambda_l$ as the embedded Markov chain of a countable-state Markov process shown in Fig. 1, where $0 < \lambda_1 < \lambda_2 < \dots < \lambda_l$. In this model, the traffic arrival rate will not change dramatically in a short amount of time. We assume the transitions only include a state switches to its previous state, its next state, or itself. The transition from λ_i to λ_{i+1} ($0 < i < l - 1$) indicates a traffic surge and we want to detect it promptly to avoid potential traffic congestion and large queueing delays. The transition from λ_{i+1} to λ_i ($0 < i < l - 1$) indicates a traffic drop and we want to detect it promptly to avoid any waste of resources. When $\lambda(t)$ maintains at state λ_i , there is no traffic change and we assume the holding interval associated with the state λ_i follows an exponential distribution with a rate of v_i , which is named as the holding rate. Each state is associated with a transient rate $a_{i,j}$ from λ_i to λ_j .

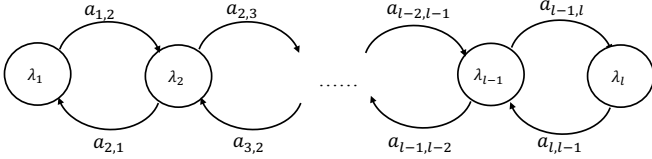


Fig. 1. Multi-state embedded Markov chain transition of the traffic arrival rate.

B. Network Coherence Time

The network coherence time is the duration that the network traffic pattern is not varying. In this work, we define it as the holding interval where the traffic arrival rate maintains at a certain state, and we denote the average coherence time as D . In our traffic arrival model, given the current state is λ_i , the average coherence time in state λ_i is $\frac{1}{v_i} = \frac{1}{a_{i,i-1} + a_{i,i+1}}$. It is because the network will remain in this state and the time to the next transition is the time until either a transition to the previous state or the next state. With the prior of state λ_i as π_i , the average coherence time for the whole system is $\sum_{i=1}^l \frac{\pi_i}{v_i}$.

We can categorize the network traffic environments using the coherence time and the time to make decisions into three regimes. With the average detect time of the estimator defined as τ_1 , the three regimes are: 1. long coherence time: $\tau_1 \ll D$, where the traffic changes very slowly; 2. moderate coherence time: $\tau_1 \sim D$, where the traffic changes at a moderate rate; 3. short coherence time: $\tau_1 \gg D$, where the traffic changes very quickly. Alternatively, we can quantize the coherence time with the holding rate v_i . We assume v_i

is changing within a range of $[v_{min}, v_{max}]$. The range of v_i can be learned and determined from the historical records of network traffic. The three regimes become: 1. long coherence time, where $v_i \sim v_{min}$; 2. moderate coherence time, where $v_i \sim \frac{v_{min} + v_{max}}{2}$; 3. short coherence time, where $v_i \sim v_{max}$. In the long coherence time regime, we could detect the traffic change with a random walk of I.I.D. T_i s as solved in [1]. In the moderate coherence time regime, the algorithm still works but the T_i s are no longer identically distributed. In the short coherence time regime, it is hard to detect every single traffic change. If the coherence time is extremely short, the algorithm will no longer converge, since the network arrival rate has changed again when the threshold is triggered and even before a single trigger. In this case, we can only try to predict the trend when the session arrival statistics provides enough confidence whether the fast traffic changes has an underlying trend.

C. The stopping-trial estimator

In this work, we keep exploring the efficacy of the stopping-trial estimator $\hat{\lambda}_{ST}(t)$ developed in [1] [2]. We have shown $\hat{\lambda}_{ST}(t)$ can make a decision at the shortest possible time when the session arrival statistics provide enough confidence for reconfiguration without a pre-determined observation time or count. We observe the inter-arrival times T_i of the traffic arrival process as a sequential test to trigger network reconfigurations. In our general multi-state traffic arrival model, define a random walk as the sum of the inter-arrival time minus an offset determined by the starting state, i.e., $S_J = \sum_{i=1}^J (T_i - \frac{1}{\lambda_j})$ if the process start from state λ_j , where J is the time that a threshold is crossed. Once S_J crosses a threshold, the corresponding reconfiguration will be made. S_J will then reset and the offset will be changed from the new state. Two sample random walks are shown in Fig. 2, where n is a discretized time index in the unit of arrivals. η_+ is the threshold to add a new wavelength, and η_- is the threshold to tear down an existing wavelength. Both η_+ and η_- are determined by the desired error probabilities as developed in [1].

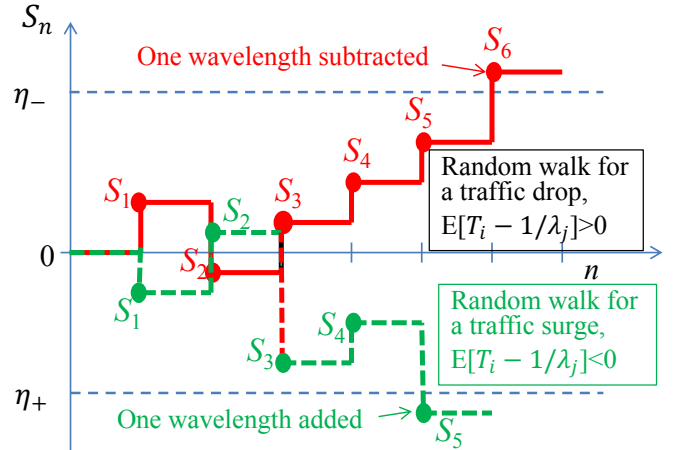


Fig. 2. Sample functions of random walks S_n with one for traffic surge and one for traffic drop. n is a discretized time index in the unit of arrivals. [1]

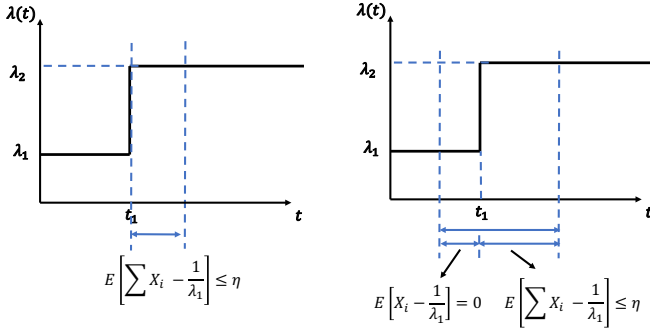


Fig. 3. The threshold trigger processes with different counting start points.

Our previous work assumes that the random walk counting starts from the point of the traffic change. Therefore, all T_i s are identical and S_n is a random walk. However, a more realistic case is the counting start point can start at any time, since we cannot predict when the traffic changes will occur. Also, the T_i s may not be identically distributed since it could be a mix of different λ_i s. Fortunately, we can prove the threshold trigger process with an earlier counting start point is the same as the case of counting starting from the point of traffic change. Without loss of generality, we demonstrate it in a two-state Markov process as shown in Fig. 3. When the traffic rate is at λ_1 , $E[T_i - \frac{1}{\lambda_1}] = 0$, which contributes nothing to the S_n for the threshold crossing on average. Therefore, only the inter-arrival time at rate λ_2 contributed to the threshold crossing, since $E[T_i - \frac{1}{\lambda_1}] = \frac{1}{\lambda_2} - \frac{1}{\lambda_1} < 0$. The statement also holds if later a traffic drop happens, since the inter-arrival time at λ_1 contributes nothing to the S_n on average. One tricky part is the inter-arrival where the surge occurs. To deal with it, we can first look forward starting from the epoch of the surge to the next arrival with λ_2 . Due to the memoryless property of the exponential distribution, this interval follows an exponential distribution with λ_2 . If we look backwards starting from the epoch of the surge to the previous arrival with λ_2 , this interval also follows an exponential distribution with λ_1 due to the reversibility of the Poisson process, and it has no impact on S_n on average. Therefore, the threshold trigger process with an earlier counting start point on average is the same with the case of counting starting from the point of traffic change.

III. FAST DETECTION IN MODERATE NETWORK COHERENCE TIME ENVIRONMENT

When the change frequency of the traffic rate is moderate, the network coherence time is similar to the average detection time of $\hat{\lambda}_{ST}(t)$. In this case, the random walk S_J can consist of independent but non-identically distributed T_i s as $\lambda(t)$ is a mixture of different states. Since T_i s are no longer identically distributed, we cannot apply Wald's Identity to get the average detection time. Fortunately, we can generalize Wald's Identity with non-identical but independent random variables as follows:

Statement (Generalized Wald's Identity) For a sequence of non-identically distributed but independent random variables z_1, z_2, \dots, z_N , denote $Z_N = \sum_{i=1}^N z_i$, where N is a nonnegative integer-valued random variable. Define $\bar{z} = \frac{1}{N} \sum_{i=1}^N E[z_i]$, and we have

$$E[Z_N] = E[N]E[\bar{z}]. \quad (1)$$

Proof $\frac{1}{N}$ is a random variable with the distribution of N 's inverse distribution, since N is a nonnegative integer-valued random variable. Then we have

$$E[Z_N] = E\left[\sum_{i=1}^N z_i\right] = E\left[N \cdot \frac{1}{N} \cdot \sum_{i=1}^N z_i\right] \quad (2)$$

$$= E[N]E\left[\frac{1}{N} \sum_{i=1}^N z_i\right] \quad (3)$$

$$= E[N]E[\bar{z}] \quad (4)$$

where Equation (3) results from that the expectation of the product of independent random variables equals to the product of individual expectations of random variables.

The generalized Wald's Identity validates the efficacy of $\hat{\lambda}_{ST}(t)$ in the moderate and even short coherence time environments. With the generalized Wald's Identity, we can find the average stopping time $\tau_{1_{ST}}$ of $\hat{\lambda}_{ST}(t)$ in a multi-state Markov process model as

$$\tau_{1_{ST}} = \frac{E[\bar{T}_i]E[S_J]}{E[T_i - \frac{1}{\lambda_o}]} \quad (5)$$

where $E[\bar{T}_i] = \frac{E[\frac{1}{J} \sum_{i=1}^J E[T_i]]}{J} = \frac{1}{J} \sum_{i=1}^J E[T_i]$, and $E[J] = E[S_J]/E[T_i - \frac{1}{\lambda_o}]$, where λ_o is the starting state of the arrival rate. $E[S_J] = \eta_+$ if a traffic surge happens, and $E[S_J] = \eta_-$ if a traffic drop happens. It is tricky to directly apply the generalized Wald's Identity to get the average detection time as in [1], since J is also a random variable used in both $E[\bar{T}_i]$ and $E[T_i - \frac{1}{\lambda_o}]$. To solve the problem, we can develop the distribution of J from $J\bar{T}_i = \sum_{i=1}^J E[T_i]$, but the analytical solution is not elegant. Instead, we provide an easily calculable analytical upper and lower bounds for $E[J]$ in this work. If we know the range $[\lambda_{min}, \lambda_{max}]$ that $\lambda(t)$ potentially move within, the bounds on $\tau_{1_{ST}}$ for a traffic surge are

$$\frac{\lambda_o \eta_+}{\lambda_o - \lambda_{max}} \leq \tau_{1_{ST}} \leq \frac{\lambda_o \eta_+}{\lambda_o - \lambda_{min}} \quad (7)$$

Figure 4 shows $\hat{\lambda}_{ST}(t)$ can provide desired wavelength assignment schemes in different network traffic environments. When the coherence time is long or moderate as shown in Fig. 4 (a) (b), $\hat{\lambda}_{ST}(t)$ can catch the traffic changes in a possible short time interval before the underlying traffic statistics changes, and the reconfiguration schemes match the requirements of the underlying traffic arrival patterns. Though $\hat{\lambda}_{ST}(t)$ cannot track well the traffic changes within a super-short coherence time as shown in Fig. 4 (c), it can still track

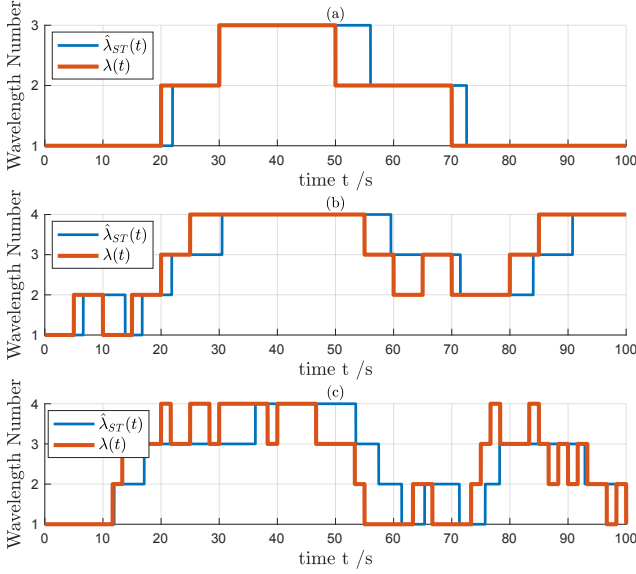


Fig. 4. The comparison of simulated wavelength assignment of $\hat{\lambda}_{ST}(t)$ and the desired wavelength assignment in different network traffic environments: (a) long network coherence time; (b) moderate network coherence time; (c) short network coherence time.

the trend of the traffic. $\hat{\lambda}_{ST}(t)$ provides less frequent reconfigurations as a result of each of the super-fast traffic changes only provide none, or one observable arrival. Only the overall trend affects reconfigurations, which is the desirable attribute of the algorithm. Coincidentally, this also reduces the network control efforts, since the random walk accumulation and the memory reset upon the detection of $\hat{\lambda}_{ST}(t)$ help to stabilize the reconfigurations avoiding highly frequent changes. Even if any false alarm happens, the error can be quickly corrected.

Figure 5 shows the potential range of the average detection time τ_1 of $\hat{\lambda}_{ST}(t)$ versus the probability of missed detection for three different arrival rate ranges. For each range, the solid line represents the upper bound of the average detection time given by λ_{min} , and the dash line represents the lower bound of the average detection time given by λ_{max} . When the traffic arrival rate changes into a higher state, the average detection time becomes shorter to make a reconfiguration faster, which reflects the adaptive detection time of $\hat{\lambda}_{ST}(t)$. A strict missed detection probability requires a long detection time on a traffic surge. It is important to pick a good error probability for the threshold crossing, since we want to keep the desirable short detection time as well as the high reconfiguration accuracy to reduce the burdens of network management and control.

The beauty of the stopping-trial estimator $\hat{\lambda}_{ST}(t)$ is its capability to provide the reconfiguration without assuming any detailed traffic model and inference on the network traffic statistics. It will only make a reconfiguration decision when it is necessary and do so at the shortest possible time. Compared to other commonly used estimators requiring the I.I.D. properties of traffic arrivals, $\hat{\lambda}_{ST}(t)$ can response to the traffic changes fast as long as the inter-arrival times of traffic transactions are independent. Also, the adaptable

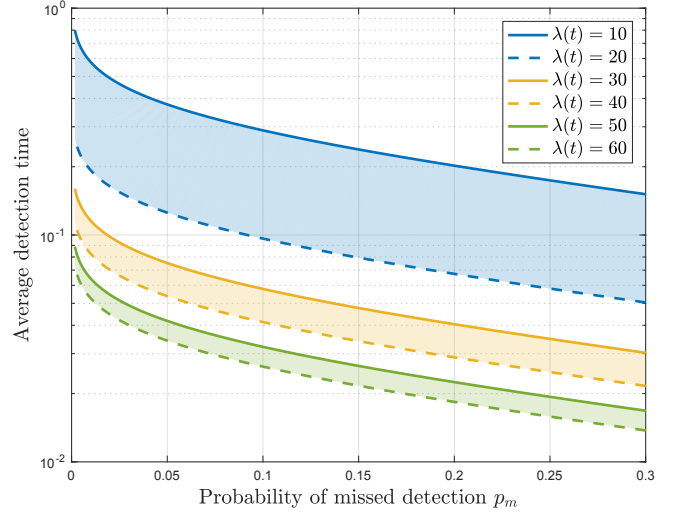


Fig. 5. The range of the average detection time τ_1 of $\hat{\lambda}_{ST}(t)$ versus probability of missed detection (crossing thresholds) for different arrival rate ranges.

detection time of $\hat{\lambda}_{ST}(t)$ enables the system to reconfigure at a fast time scale. A higher traffic arrival rate yields a shorter detection time as desired.

IV. FAST DETECTION IN SHORT NETWORK COHERENCE TIME ENVIRONMENT

When the changing frequency of the traffic rate is high, the network coherence time is much shorter than the detection time of the estimator. The simulation results of the previous section have showed it is hard to track all the super-fast changes. One way to approximate the real-time arrival rate is to use the average arrival rate over the time as $\lambda = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \lambda(t) dt$, where t_1 is the start time and t_2 is the end time. For our traffic arrival model, we define Λ_i as the average arrival rate over the corresponding inter-arrival time T_i , and $\Lambda_i = \frac{1}{X_i} \int_{X_i} \lambda(t) dt$. For n inter-arrival times T_1, T_2, \dots, T_n , we assume the corresponding arrival rates are $\Lambda_1, \Lambda_2, \dots, \Lambda_n$. It is because the real-time arrival rate changes too fast to be reflected in an inter-arrival time interval so that a quasi-statically constant intensity substitute Λ_i is a good approximation. As $\hat{\lambda}_{ST}(t)$ requires the sum of the inter-arrival times, we have

$$S_n = \sum_{i=1}^n (T_i - \frac{1}{\Lambda_o}) \Leftrightarrow \sum_{i=1}^n T_i = S_n + \frac{n}{\Lambda_o}, \quad (8)$$

where Λ_o is the starting arrival rate before the threshold is crossed. The distribution of S_n follows the convolution of n exponential distributions where each one has a unique rate as shown in Equation (8). Therefore, the distribution of S_n is

$$f_{S_n}(t) = \sum_{i=1}^n \frac{\Lambda_1 \dots \Lambda_n}{\prod_{j=1, j \neq i}^n (\Lambda_j - \Lambda_i)} e^{-\Lambda_i t}, t > 0 \quad (9)$$

It is extremely difficult to track a totally random process of the arrival rate transition in the short network coherence time

regime, since the network arrival rate changes again almost from one arrival to the next. It is also not necessary because the result is the short term average of these rates that will affect the length of the queues. However, if the fast changing traffic follows a certain pattern, we can try to predict the trend, which is the relevant parameter to track and affect the queues in the network. A simple nontrivial case is that the traffic changes so fast that the traffic arrival rate increases/decreases monotonically as a linear function. Intuitively, the traffic is drifting to a direction of the embedded Markov chain and we define the duration of the traffic's drift in one direction as the network drifting time. The network usually drifts in a direction when the embedded Markov chain is converging to a new steady state as the results of the transition rate changes. Though the changing frequency of the steady state is low practically, it can cause severe network congestions and bring burdens for the subsequent reconfigurations if it is not detected promptly. Therefore, it is necessary to analyze the transient behaviors of network drifts. Moreover, we should try to predict the traffic trend and develop the corresponding reconfiguration algorithm for network traffic drifts. Sometimes, a network traffic drifting can happen as the normal transition of the Markov process without the change of the steady state. However, such drifts are temporary and will not last for a long time, which will not be discussed in this work.

A. Network drifting time

When the network traffic drifts due to a shift of the Markov process transition rates, we want to know the transient behavior of the system on how the drift converges and how long it takes to the new steady state. Both metrics are determined by the new transition probability matrix P with each entry p_{ij} indicating the transition probability from state λ_i to state λ_j . The convergent rate to a new steady state is largely determined by the second largest eigenvalue in magnitude of P , since the largest eigenvalue is always 1 and correspond to the steady state [3]. A larger second largest eigenvalue of P in magnitude yields a lower convergent rate, which means the system takes a longer time to converge. Apart from the convergent rate, the total convergent time also greatly depends on the initial state probability distribution. If the final steady state probability distribution differs more from the initial distribution, it will take a longer time to converge.

We can find the settling time τ_2 to the new steady state with the eigenvalues of P , the eigenvectors of P , and the initial steady state probability distribution. To get the eigenvalues and the eigenvectors of P , we decompose P as $P = U\Gamma U^{-1}$, where Γ is the diagonal matrix with entries $\gamma_1, \dots, \gamma_l$ as all the eigenvalues and U is matrix whose columns are the corresponding eigenvectors of the eigenvalues on the diagonal of Γ . U^{-1} is the inverse of U . We assume P is invertible to facilitate the analysis, and P has l distinct eigenvalues. The transition probability matrix P must have the largest eigenvalue as 1 [3] and it is the only one since all l eigenvalues are assumed distinct. Without loss of generality, denote $\gamma_1 = 1$, $|\gamma_i| < 1$ for $i \neq 1$, and γ_2 as the second largest eigen-

value in magnitude. Denote $\Pi(0) = [\pi_1(0), \pi_2(0), \dots, \pi_l(0)]$ as the initial steady state probability distribution when the network drift begins. The new steady state is reached when time tends to infinity with the steady state distribution as $\Pi(\infty) = [\pi_1(\infty), \pi_2(\infty), \dots, \pi_l(\infty)]$ if P never changes. With an arbitrarily small positive quantity ϵ and a sample unit time δ , we now can represent $\Pi(0)$ approximately close to $\Pi(\infty)$ within a time length of τ_2 as

$$\|\Pi(0)P^{\frac{\tau_2}{\delta}} - \Pi(\infty)\| \leq \epsilon \quad (10)$$

$$\Leftrightarrow \|\Pi(0)(U\Gamma U^{-1})^{\frac{\tau_2}{\delta}} - \Pi(0)(U\Gamma U^{-1})^\infty\| \leq \epsilon \quad (11)$$

$$\Leftrightarrow \|\Pi(0)U\Gamma^{\frac{\tau_2}{\delta}}U^{-1} - \Pi(0)U\Gamma^\infty U^{-1}\| \leq \epsilon \quad (12)$$

$$\Leftrightarrow \|\Pi(0)U(\Gamma^{\frac{\tau_2}{\delta}} - \Gamma^\infty)U^{-1}\| \leq \epsilon \quad (13)$$

where

$$\Gamma^{\frac{\tau_2}{\delta}} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \gamma_2^{\frac{\tau_2}{\delta}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \gamma_l^{\frac{\tau_2}{\delta}} \end{bmatrix}, \Gamma^\infty = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}.$$

By solving the Inequality (13), we can get the settling time τ_2 . We can also get an elegant analytical upper bound on τ_2 with the second largest eigenvalue γ_2 by applying an upper bound on $\Gamma^{\frac{\tau_2}{\delta}} - \Gamma^\infty =$

$$\begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & \gamma_2^{\frac{\tau_2}{\delta}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \gamma_l^{\frac{\tau_2}{\delta}} \end{bmatrix} \leq \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & \gamma_2^{\frac{\tau_2}{\delta}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \gamma_2^{\frac{\tau_2}{\delta}} \end{bmatrix} \quad (14)$$

Denote $W = \Pi(0)U = [w_1, w_2, \dots, w_l]$ to facilitate the notation. We can get an upper bound on the left-hand side of Equation (13) as

$$\Pi(0)U(\Gamma^{\frac{\tau_2}{\delta}} - \Gamma^\infty)U^{-1} \leq [0, w_2\gamma_2^{\frac{\tau_2}{\delta}}, \dots, w_l\gamma_l^{\frac{\tau_2}{\delta}}]U^{-1} \quad (15)$$

$$= \gamma_2^{\frac{\tau_2}{\delta}} [0, w_2, \dots, w_l]U^{-1} \quad (16)$$

$$\triangleq \gamma_2^{\frac{\tau_2}{\delta}} W'U^{-1} \quad (17)$$

Then we have

$$\|\Pi(0)U(\Gamma^{\frac{\tau_2}{\delta}} - \Gamma^\infty)U^{-1}\| \leq \|\gamma_2^{\frac{\tau_2}{\delta}} W'U^{-1}\| \quad (18)$$

$$\leq \|\gamma_2^{\frac{\tau_2}{\delta}} W'\| \cdot \|U^{-1}\| \quad (19)$$

$$= \gamma_2^{\frac{\tau_2}{\delta}} \|W'\| \cdot \|U^{-1}\| \quad (20)$$

where $\|W'\|$ is the Euclidean norm (or 2-norm) of the vector W' , and $\|U^{-1}\|$ is the 2-norm of the matrix U^{-1} . Using the upper bound in (20), an upper bound on τ_2 is

$$\gamma_2^{\frac{\tau_2}{\delta}} \|W'\| \cdot \|U^{-1}\| \leq \epsilon \quad (21)$$

$$\Leftrightarrow \tau_2 \leq \log_{\gamma_2} \frac{\epsilon\delta}{\|W'\| \cdot \|U^{-1}\|} \quad (22)$$

$$\Leftrightarrow \tau_2 \leq \log_{\gamma_2} \frac{\epsilon\delta}{\|W'\| \sigma_{\max}(U^{-1})} \quad (23)$$

where $\sigma_{\max}(U^{-1})$ represents the largest singular value of U^{-1} .

To evaluate the convergent times of different initial steady states, we use the idea of Kullback-Leibler divergence [4] to represent the distance between the initial steady state probability distribution and the new steady state probability distribution after the network drift as

$$Dist.(\Pi(0)||\Pi(\infty)) = \sum_{x \in \mathcal{X}} \Pi(0) \log \left(\frac{\Pi(0)}{\Pi(\infty)} \right) \quad (24)$$

where both $\Pi(0)$ and $\Pi(\infty)$ are defined on the same probability space \mathcal{X} . $Dist.(\Pi(0)||\Pi(\infty))$ is a measure of how $\Pi(0)$ is different from $\Pi(\infty)$, where we set $\Pi(\infty)$ as the reference probability distribution. A larger $Dist.(\Pi(0)||\Pi(\infty))$ represents a larger difference.

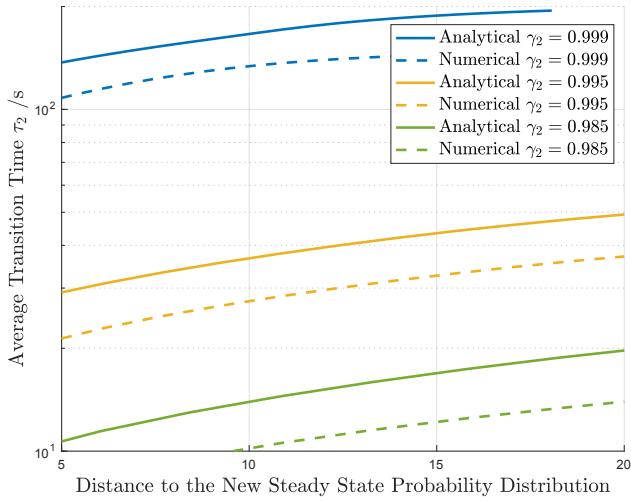


Fig. 6. Analytical upper bounds and numerical results on the settling time τ_2 versus different distances between the initial steady state probability distribution and the new steady state probability distribution after the network drift with different convergent rates (the second largest eigenvalue in magnitude of the probability transition matrix). $l = 100$, $a_{i,i+1} = 10$ ($0 < i < l-1$), $\delta = 0.01s$, $\epsilon = 10^{-5}$.

Figure 6 shows that our analytical upper bound on the settling time τ_2 can approximate τ_2 well. The settling time τ_2 increases with the increase of the distance between the initial steady state probability distribution and the new steady state probability distribution after the network traffic drift. Besides, a larger second largest eigenvalue in magnitude γ_2 of the probability transition matrix results in a longer settling time as expected. A small amount of increase in γ_2 can result in a noticeable increase in the settling time due to the iterative multiplication of the transition probability matrix. By modeling the transient behavior of the network drifts analytically, we can know whether the network traffic drift will maintain a long enough network settling time for us to predict the traffic trend. It will be good if we can predict the trend before the new steady state is reached and allocate resource in advance to avoid the foreseeable network congestion.

B. Prediction of the network traffic drifting trend

We depict the extreme case of the network traffic drifts into a linear model, where the network coherence time is so short that the arrival rate changes at every epoch. Without loss of generality, we only discuss the process of increasing network traffic drifts in the following sections as the decreasing process is its reverse. We assume the arrival rate increases in a constant slope of k as $\lambda(t) = \Lambda_o + kt$, where $k > 0$ as an increasing process and Λ_o is the starting arrival rate. Our goal is to determine k with a given time interval of arrivals.

We can estimate k by the maximum likelihood estimation based on the probability of the arrivals sequentially. To reduce the computation complexity, we limit k to a certain range, i.e., $k_{min} \leq k \leq k_{max}$. If k is too large such as tending to the infinity, the traffic arrival rate increases too sharply to be reconfigured. If k is too small such as tending to zero, there is almost no traffic arrival changes and no reconfiguration is needed. The probability density function of an inhomogeneous Poisson Process with arrivals at epoch t_1, t_2, \dots, t_n in a time interval with length of T_0 is

$$f(t_1, t_2, \dots, t_n | \lambda(t)) = \left[\prod_{i=1}^n \lambda(t_i) \right] \exp \left\{ - \int_0^{T_0} \lambda(\eta) d\eta \right\} \quad (25)$$

The probability that n arrivals happen in the time interval $(0, T_0)$ is

$$Pr(N = n) = \int_{0 < t_1 < \dots < t_n < T_0} f(t_1, \dots, t_n | \lambda(t)) dt_1 \dots dt_n \quad (26)$$

Therefore, we have the maximum likelihood estimator as

$$\hat{k} = \operatorname{argmax}_{k_{min} \leq k \leq k_{max}} f(t_1, t_2, \dots, t_n | \lambda(t)) \quad (27)$$

Given the linear arrival rate change model as $\lambda(t) = \Lambda_o + kt$, we get \hat{k} as

$$\frac{d}{dk} f(t_1, t_2, \dots, t_n | \lambda(t)) = 0 \quad (28)$$

$$\Leftrightarrow \frac{d}{dk} \prod_{i=1}^n (\Lambda_o + kt_i) \exp \left\{ - \int_0^{T_0} \lambda(\eta) d\eta \right\} = 0 \quad (29)$$

Obviously, the $\exp\{\cdot\}$ part is the area of density function during $[0, T_0]$. By taking log of the Equation (29), we have

$$\frac{d}{dk} \left[\sum_{i=1}^n \ln(\Lambda_o + kt_i) \right] - (\Lambda_o + \frac{kT_0}{2})T_0 = 0 \quad (30)$$

$$\Leftrightarrow \left[\sum_{i=1}^n \frac{t_i}{\Lambda_o + kt_i} \right] - \frac{T_0^2}{2} = 0 \quad (31)$$

It is hard to find the analytical results of k from Equation (31), and we can also prove that there is no closed-form solution for the roots of a fifth or higher degree polynomial equations by Abel-Ruffini Theorem [5]. Instead, we can use the numerical methods to find the only positive root. Another way is to find closed-form analytical bounds on \hat{k} . For the

lower bound, we apply Arithmetic Mean - Harmonic Mean (AM-HM) inequality as

$$\sum_{i=1}^n \frac{t_i}{\Lambda_o + kt_i} = \sum_{i=1}^n \frac{1}{\frac{\Lambda_o}{t_i} + k} \quad (32)$$

$$\geq \frac{n^2}{\sum_{i=1}^n (\frac{\Lambda_o}{t_i} + k)} \quad (33)$$

Combining with the Equation in (31), we have

$$\sum_{i=1}^n (\frac{\Lambda_o}{t_i} + k) \geq \frac{2n^2}{T_0^2} \quad (34)$$

$$\Leftrightarrow k \geq \frac{2n}{T_0^2} - \frac{\Lambda_o}{n} \sum_{i=1}^n \frac{1}{t_i} \quad (35)$$

To find an upper bound, let's first assume $k > \frac{2n}{T_0^2} - \frac{\Lambda_o}{t_n}$. Given the fact $0 < t_1 < t_2 < \dots < t_n$, we have $\frac{\Lambda_o}{t_1} > \frac{\Lambda_o}{t_2} > \dots > \frac{\Lambda_o}{t_n} > 0$. Therefore, we have

$$k + \frac{\Lambda_o}{t_i} \geq k + \frac{\Lambda_o}{t_n} \quad (36)$$

$$> \frac{2n}{T_0^2} \quad (37)$$

$$\Leftrightarrow \frac{1}{k + \frac{\Lambda_o}{t_i}} < \frac{T_0^2}{2n} \quad (38)$$

$$\Leftrightarrow \sum_{i=1}^n \frac{1}{k + \frac{\Lambda_o}{t_i}} < \frac{T_0^2}{2} \quad (39)$$

where the Inequality (39) contradicts to the Equation (31). Therefore, our assumption $k > \frac{2n}{T_0^2} - \frac{\Lambda_o}{t_n}$ is wrong, and it thus gives us an upper bound on \hat{k} as

$$k \leq \frac{2n}{T_0^2} - \frac{\Lambda_o}{t_n} \quad (40)$$

Figure 7 shows the slope estimation of our sequential maximum likelihood estimator \hat{k} in a time period of T_0 with different random numbers of arrivals. It can estimate the slope sufficiently well with a reasonable number of arrivals. Also, both the lower bound \hat{k}_{lower} and the upper bound \hat{k}_{upper} together give an good approximation to \hat{k} analytically. Since the upper bound is simpler and only depends on n and t_n (only the last arrival time is needed), it is a useful exact analytical approximation to the optimum estimate. The convex curve shape indicates that less samples make the results inaccurate and more samples can cause overfitting. With the prediction of the network traffic drift trend, we can design the proper reconfiguration algorithm and allocate more resources in advance to avoid potential traffic congestion. It enables the system to quickly and accurately adapt to the network traffic conditions to save overly jittery control efforts, minimize queueing delays due to late reconfigurations, and use network resources efficiently.

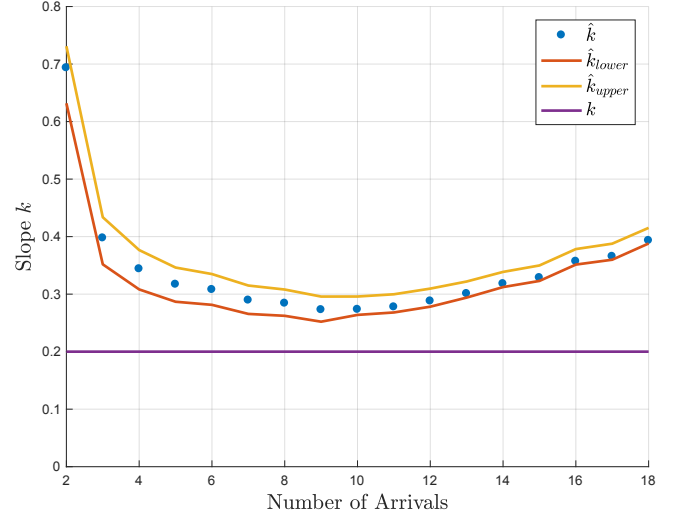


Fig. 7. Slope estimation of the network traffic drifting trend in a linear increasing model versus different number of arrivals over 200 runs. $\Lambda_o = 5$, $k = 0.2$.

V. CONCLUSION

In this paper, we generalize our traffic arrival model into a multi-state Markov process and show the stopping-trial estimator can provide the desired wavelength assignment scheme in dynamic network traffic environments with different network coherence times. The stopping-trial estimator can make reconfiguration decisions at the shortest possible time with independent but not necessarily identically distributed inter-arrival times of traffic sessions. We model the transient behavior of the network traffic drifts towards convergence to a new steady state analytically for the super-fast traffic rate changes to validate the feasibility of the traffic prediction. Given a specific network drift, we provide a traffic trend estimation technique for the super-fast traffic rate changes approximated by a linear model. With a reasonable number of arrivals, our sequential maximum likelihood estimator can estimate the network traffic drifting trend and enable reconfigurations in advance to minimize the build of congestion. Our algorithms make the network management and control efforts more efficient and thus more affordable to meet the requirement of future dynamic traffic.

REFERENCES

- [1] A. Zheng and V. Chan, "Cognitive management and control for wavelength assignment and reconfiguration in optical networks," in *International Conference on Communications(ICC)*, May 2019, pp. 1–7.
- [2] V. Chan, "Cognitive optical networks," in *International Conference on Communications(ICC)*, May 2017, pp. 1–6.
- [3] R. Gallager, *Stochastic Processes: Theory for Applications*. New York, NY: Cambridge University Press, 2014.
- [4] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley-Interscience, 2006.
- [5] J. Tignol, *Galois' Theory of Algebraic Equations: Second Edition*. World Scientific Publishing Company Pte Limited, 2015.