Cognitive Management and Control for Wavelength Assignment and Reconfiguration in Optical Networks

Anny 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*—We address the design of fast wavelength assignment and reconfiguration using cognitive management and control that can quickly and accurately adapt to the operating conditions for optical networks. The traffic detection performances of two Bayesian estimators and a stopping-trial (sequential) estimator are examined based on the transient behaviors of networks. The stopping-trial estimator has the fastest response time to the changes of traffic arrival statistics. We propose a wavelength reconfiguration algorithm with continuous assessment where the system reconfigures whenever it deems necessary. The reconfiguration can involve addition or subtraction of multiple wavelengths. Using the fastest detection and reconfiguration algorithm can reduce queueing delays during traffic surges without over provisioning and thus can reduce network capital expenditure and prevent waste of resources upon erroneous decision on occurrence of surges.**

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

I. Introduction

The bursty, unscheduled and large data transactions introduced by new technological applications can cause both high costs and extreme congestions in networks. The dynamic and bursty (unpredictable) nature of large traffic transactions either requires over-provisioning of the networks which is costly or a more agile network control and management system that adaptively allocate resources by reconfiguring the network in a timely manner in reaction to the offered traffic. The network management and control system should be able to sense traffic changes and reconfigure to use network resources efficiently. Specifically, reconfigurations should be done as fast as a sub-second time scale with no human involvement. To meet these demands, cognitive networking is proposed as a candidate architectural construct that can provide fast, dynamic, and efficient control using cognitive techniques.

In this paper, we present the design of fast-reconfigurable cognitive wavelength management and control algorithms that can accurately adapt by observing the operating conditions of the networks. Our previous work [1] provided a brief overview of cognitive optical networks and proposed two Bayesian estimators and a stopping-trial estimator to detect traffic changes. In this work, we further develop these estimators and examine their traffic detection and queueing delay performances based on the resulting transient behaviors of networks. A network cost model is proposed to capture the trade-off between reconfiguration performance (transient queueing delays) and the cost of the capacity plus the control resources used. We recommend a wavelength reconfiguration algorithm based on the stopping-trial estimator with continuous assessment where the system reconfigures whenever necessary. The reconfiguration can involve addition or subtraction of multiple wavelengths.

II. Traffic Model and Tunneled Architecture

A. All-to-all Poisson Traffic

As a simple illustrative example, we consider a network topology (MAN or WAN) with a WDM tunnel architecture. We assume all-to-all independent and identically distributed (I.I.D.) sessions between every node pair. The size of each transaction L is exponentially distributed with the expectation L_0. The arrival traffic at the source is assumed to form a doubly stochastic Poisson point process with a time-dependent rate of $\lambda(t)$. We assume $\lambda(t)$ switches between a non-surging state λ_0 and a surging state λ_1, where $\lambda_0 < \lambda_1$. When $\lambda(t)$ switches from λ_0 to λ_1, there is a traffic surge and we want to detect it promptly to avoid potential traffic congestion and large queueing delays. When $\lambda(t)$ switches from λ_1 to λ_0, there is a traffic drop and we want to detect it promptly to avoid any waste of resources.

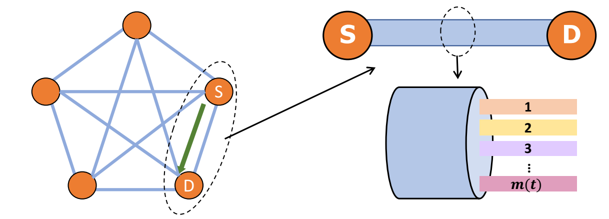
We can observe the Poisson arrival process in two ways. First, we observe the number of arrivals N in the observation interval $[t-T, t]$. N follows a Poisson distribution with the rate of $\lambda(t)T$. Notice that T should be less than the network coherence time for effective adaptions, and both time points $(t-T)$ and t are included to avoid ambiguity. Second, each inter-arrival time in $\{T_i, i \geq 1\}$ follows an exponential distribution with parameter $\lambda(t)$. We assume the 1^{st} arrival always happens at the starting time $(t-T)$, and T_i is the inter-arrival time between the i^{th} arrival and the $(i+1)^{th}$

This work is sponsored by NSF NeTS program, Grant No. #6936827 and the HKUST/MIT programs.

arrival. The sum of N inter-arrival times $\sum_{i=1}^{N} T_i$ follows an Erlang distribution with $\lambda(t)$.

B. Tunneled Optical Network Architecture

In this work, we assume a tunneled network architecture, where each node pair is connected via a pre-selected set of wavelengths within a single lightpath or multiple lightpaths for traffic transmission as shown in Fig. 1. Zhang showed in [2] that tunneled architecture can reduce control plane traffic and processing complexity significantly with little sacrifice in efficiency for heavy traffic volumes compared to meshed architecture that enables full switchability at all nodes. The capacity between each node pair is reconfigurable by adjusting the number of wavelengths used by the node pair based on the offered traffic. Assume $m(t)$ wavelengths are assigned between a node pair at time t, and each wavelength has a constant capacity R bits per second. Given the average size of a transaction L_0, the service rate of each wavelength per transaction is $\mu = \frac{R}{L_0}$. The queue between each node pair can be modeled as an $M/M/m(t)$ queue with the arrival rate $\lambda(t)$ and the service rate μ. Define the network load between a node pair as $\rho = \frac{\lambda(t)}{m(t)\mu}$. The system is in a stable state when $\rho < 1$. When $\rho \geq 1$, the network is overloaded, and network reconfigurations are needed to bring the system to a new steady state. In practice, the addition of capacities occurs as early as $\rho \sim \frac{1}{2}$ because users do not want excessive queueing delay.



All-to-all tunneled network topology Reconfigurable wavelength assignment

Fig. 1. An example of an all-to-all tunneled network connection between node pairs in the form of wavelengths.

III. Candidate Estimators

We consider two traditional Bayesian estimators and the stopping-trial sequential estimator proposed in [1] to detect the changes of traffic statistics. Given the binary nature of $\lambda(t)$ defined in section II-A, two possible hypotheses for the decision are: $H_0 : \lambda(t) = \lambda_0$; $H_1 : \lambda(t) = \lambda_1$. The false alarm probability P_f is the probability that we accept H_1 when H_0 is true. The missed detection probability P_m is the probability that we accept H_0 when H_1 is true. The probability of detection is $P_d = 1 - P_m$. If the *a priori* probabilities are known as π_{λ_0} for H_0 and $\pi_{\lambda_1} = 1 - \pi_{\lambda_0}$ for H_1, the total error probability is

$$Pr[e] = \pi_{\lambda_0} P_f + \pi_{\lambda_1} P_m \tag{1}$$

A. Fixed-time estimator $\hat{\lambda}_T(t)$

For the fixed-time estimator $\hat{\lambda}_T(t)$, we count the total number of arrivals in a fixed time interval $[t-T, t]$ denoted by $N(T)$ backwards in time to determine the validation of the hypotheses. We define

$$\hat{\lambda}_T(t) = \frac{N(T)}{T} \tag{2}$$

1) Bayesian Likelihood Ratio Test (Bayesian LRT): The Bayesian LRT for $N(T)$ with given *a priori* probabilities π_{λ_0} and π_{λ_1} is

$$n \underset{\lambda_0}{\overset{\lambda_1}{\gtrless}} \frac{(\lambda_1 - \lambda_0)T + \ln\left(\frac{\pi_{\lambda_0}}{\pi_{\lambda_1}}\right)}{\ln\left(\frac{\lambda_1}{\lambda_0}\right)} \triangleq \gamma_T \tag{3}$$

The error probability $Pr[e_T]$ for $\hat{\lambda}_T(t)$ is $Pr[e_T] =$

$$\pi_{\lambda_0} e^{-\lambda_0 T} \sum_{n=\lceil \gamma_T \rceil}^{\infty} \frac{(\lambda_0 T)^n}{n!} + \pi_{\lambda_1} e^{-\lambda_1 T} \sum_{n=0}^{\lceil \gamma_T \rceil - 1} \frac{(\lambda_1 T)^n}{n!} \tag{4}$$

2) Neyman-Pearson Test: In practice, *a priori* probabilities are usually unknown, which leads us to use the Neyman-Pearson test. Define a threshold η for the LRT, we have

$$n \underset{\lambda_0}{\overset{\lambda_1}{\gtrless}} \frac{(\lambda_1 - \lambda_0)T + \ln \eta}{\ln\left(\frac{\lambda_1}{\lambda_0}\right)} \triangleq \gamma_{T'} \tag{5}$$

Given n is an integer, the false alarm probability P_{f_T} is

$$P_{f_T} = \sum_{n=\lceil \gamma_{T'} \rceil}^{\infty} \frac{(\lambda_0 T)^n e^{-\lambda_0 T}}{n!} \tag{6}$$

where $\lceil \cdot \rceil$ is the ceiling function. The missed detection probability P_{m_T} is

$$P_{m_T} = \sum_{n=0}^{\lceil \gamma_{T'} \rceil - 1} \frac{(\lambda_1 T)^n e^{-\lambda_1 T}}{n!} \tag{7}$$

We can use Chernoff bounds to approximate the false alarm probability and the missed detection probability. However, the detection time T after a rate change needs to be as short as possible to achieve the fast response preventing the queue build-up. A short detection time will inevitably cause higher false alarm/missed detection rates, where the exponentially tight Chernoff bound is not a good approximation. Though we will not use the Chernoff bound approximation in this work, it does provide an easily calculable approximation when the requirement of the probability of false alarm/missed detection is strict, which will be discussed in our future work.

B. Fixed-count estimator $\hat{\lambda}_N(t)$

For the fixed-count estimator $\hat{\lambda}_N(t)$, we observe the duration $T(N)$ formed by the last N arrivals (including the one at $(t-T)$) backwards in time to determine the validation of the hypotheses. We define

$$\hat{\lambda}_N(t) = \frac{N}{T(N)} = \frac{N}{\sum_{i=1}^{N-1} T_i + Z(t)} \tag{8}$$

where $Z(t)$ is the age of the Poisson process of the observation interval ending at time t, which is defined as the interval from the most recent arrival (N^{th} arrival) before (not include) t until t. If the N^{th} arrival happens at time t, $Z(t) = 0$. $(N-1)$ inter-arrivals are included in the previous N arrivals, so that $Z(t) = T - \sum_{i=1}^{N-1} T_i$.

We can prove $Z(t)$ also follows an exponential distribution with rate $\lambda(t)$. If we look at the arrivals of the Poisson process in $[t-T, t]$ backward in time, it is still a Poisson process due to its time-reversibility [3], and $Z(t)$ becomes the interval between the starting time and the first arrival. Due to the memoryless property of the exponential distribution, $Z(t)$ follows an exponential distribution. Therefore, we have

$$\hat{\lambda}_N(t) = \frac{N}{T(N)} = \frac{N}{\sum_{i=1}^{N} T_i} \tag{9}$$

1) Bayesian LRT: The Bayesian LRT for $T(N)$ with *a priori* probabilities π_{λ_0} and π_{λ_1} is

$$\tau \underset{\lambda_1}{\overset{\lambda_0}{\gtrless}} \frac{\ln\left(\frac{\pi_{\lambda_1}}{\pi_{\lambda_0}}\right) + N\ln\left(\frac{\lambda_1}{\lambda_0}\right)}{(\lambda_1 - \lambda_0)} \triangleq \gamma_N \tag{10}$$

The error probability $Pr[e_N]$ for $\hat{\lambda}_N(t)$ is $Pr[e_N] =$

$$\frac{\pi_{\lambda_0}\lambda_0^N}{(N-1)!}\int_0^{\gamma_N} \tau^{N-1}e^{-\lambda_0\tau}d\tau + \frac{\pi_{\lambda_1}\lambda_1^N}{(N-1)!}\int_{\gamma_N}^{\infty} \tau^{N-1}e^{-\lambda_1\tau}d\tau \tag{11}$$

2) Neyman-Pearson Test: For the Neyman-Pearson test, define a threshold η for LRT and we have

$$\tau \underset{\lambda_1}{\overset{\lambda_0}{\gtrless}} \frac{N\ln\left(\frac{\lambda_1}{\lambda_0}\right) - \ln\eta}{(\lambda_1 - \lambda_0)} \triangleq \gamma_{N'} \tag{12}$$

The false alarm probability P_{f_T} is

$$P_{f_N} = \int_0^{\gamma_{N'}} \frac{\lambda_0^N \tau^{N-1} e^{-\lambda_0\tau}}{(N-1)!} d\tau \tag{13}$$

The missed detection probability P_{m_T} is

$$P_{m_N} = \int_{\gamma_{N'}}^{\infty} \frac{\lambda_1^N \tau^{N-1} e^{-\lambda_1\tau}}{(N-1)!} d\tau \tag{14}$$

C. Comparison of the two Bayesian estimators

The simulated rate change detection comparisons of both $\hat{\lambda}_T(t)$ and $\hat{\lambda}_N(t)$ in a single run and the average of 200 runs are shown in Fig. 2. The fixed count N in $\hat{\lambda}_N(t)$ is chosen such that $N = \lambda(t)T$ for the fixed time T in $\hat{\lambda}_T(t)$. $\hat{\lambda}_N(t)$ always responds faster to sudden rate changes than $\hat{\lambda}_T(t)$, though $\hat{\lambda}_N(t)$ may not be as accurate (and stable) as $\hat{\lambda}_T(t)$, which is also shown in the receiver operating characteristic (ROC) comparisons of the two estimators in Fig. 3. Compared to the fixed time T, the detection time of $\hat{\lambda}_N(t)$ can flexibly adjust to the underlying $\lambda(t)$. $\hat{\lambda}_N(t)$ can improve network efficiency by avoiding both a long detection time for fast changes and a high sampling frequency for low arrival rates.

Though both Bayesian estimators are simple to implement, their pre-determined fixed time or count limits their detection performance, which can result in inaccurate and even disruptive reconfigurations. Moreover, both Bayesian estimators require the *a priori* probability distributions, which are usually unknown and may be changing. Though the distribution can be estimated from prior traffic statistics by learning techniques, such techniques fail to perform well for extremely rare events (e.g. black swans) due to the lack of history. Hence there is the need to find an estimator that can detect rate changes in the shortest possible time so that the system can be reconfigured at a fast time scale. We will explore the efficacy of a sequential decision algorithm called "stopping trials" in the next section.

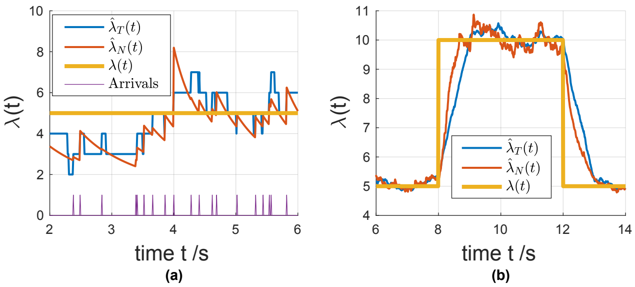


Fig. 2. (a) The comparison of detection processes of $\hat{\lambda}_T(t)$ and $\hat{\lambda}_N(t)$; (b) The comparison of the average detection results over 200 runs of $\hat{\lambda}_T(t)$ and $\hat{\lambda}_N(t)$. $\lambda_0 = 5, \lambda_1 = 10$. $T = 1, N = \lambda_0 T = 5$.

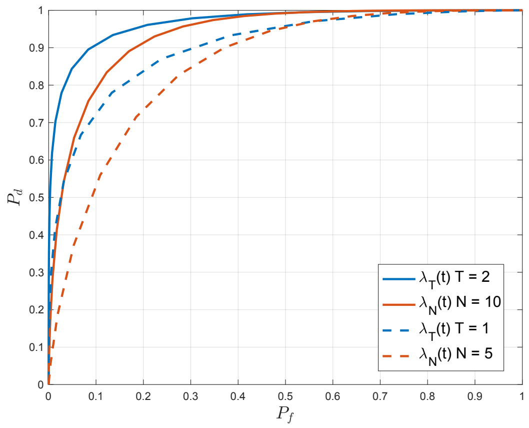


Fig. 3. The comparison of ROC curves of fixed-time estimator $\hat{\lambda}_T(t)$ and fixed-count estimator $\hat{\lambda}_N(t)$. $\lambda_0 = 5, \lambda_1 = 10$. The fixed count N in $\hat{\lambda}_N(t)$ is chosen such that $N = \lambda_0 T$ for the fixed time T in $\hat{\lambda}_T(t)$.

D. Stopping-trial estimator $\hat{\lambda}_{ST}(t)$

For the stopping-trial estimator $\hat{\lambda}_{ST}(t)$, we observe each inter-arrival time T_i of the doubly stochastic Poisson point process as a sequential test to trigger network reconfigurations. As opposed to $\hat{\lambda}_T(t)$ and $\hat{\lambda}_N(t)$, $\hat{\lambda}_{ST}(t)$ does not require a pre-determined observation time or count. It can make a decision at the shortest possible time when the session arrival statistics provide enough confidence for reconfiguration [3]. The process can be modeled as a random walk S_J based on $\{T_i, i \geq 1\}$, where J is the time that a threshold is crossed and a reconfiguration is made. Define $S_J = \sum_{i=1}^{J}(T_i - \frac{1}{\lambda_0})$ if the process start from a non-surging state; $S_J = \sum_{i=1}^{J}(T_i - \frac{1}{\lambda_1})$ if the process start from a surging state. To avoid bias from the previous decision, the algorithm resets S_J and starts from the new state once a traffic rate change is detected. Two sample random walks are shown in Fig. 4 [1], where n is a discretized time index in the unit of arrivals. Denote the threshold for adding a new wavelength as η_+ and the threshold for tearing down an existing wavelength as η_-. Both η_+ and η_- are determined by the desired error probabilities.

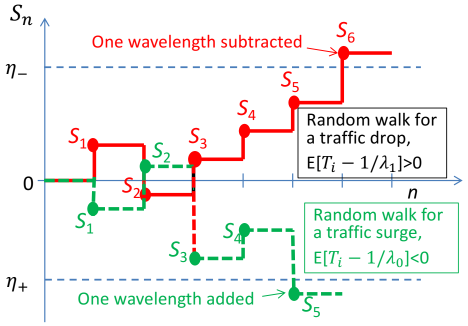

S_n
η_-
0
η_+
n
One wavelength subtracted
S_6
S_5
S_4
S_3
S_1
S_2
Random walk for a traffic drop, $E[T_i - 1/\lambda_1]$>0
Random walk for a traffic surge, $E[T_i - 1/\lambda_0]$<0
S_1
S_3
S_4
One wavelength added
S_5

Fig. 4. Sample functions of random walks S_n with one for traffic surge and one for traffic drop [1]. n is a discretized time index in the unit of arrivals.

An exponentially tight upper bound of the missed detection probability for a surge from Wald's identity in [3] is

$$P_{m_{ST}} \leq e^{-r^*\eta_-}. \tag{15}$$

where r^* is the positive root for $\ln(E[e^{r(T_i-\frac{1}{\lambda_0})}]) = 0$. An upper bound on the false alarm probability after κ arrivals given no surge happens from [1] is

$$P_{f_{ST}} \leq \frac{\kappa}{[\lambda_0\eta_+]^2} \tag{16}$$

Figure 5 shows the detection performances of traffic surges/drops for two Bayesian estimators and a stopping-trial estimator. The fixed time for $\hat{\lambda}_T(t)$, the fixed count for $\hat{\lambda}_N(t)$, and the thresholds for $\hat{\lambda}_{ST}(t)$ are picked so that all three estimators' probabilities of missed detection are 1%. From Fig. 5, $\hat{\lambda}_{ST}(t)$ has the shortest response time to rate changes. The memory reset upon the detection helps to stabilize $\hat{\lambda}_{ST}(t)$ to avoid highly frequent erroneous reconfigurations. Even if $\hat{\lambda}_{ST}(t)$ make any false alarm, it can be quickly corrected. $\hat{\lambda}_{ST}(t)$ requires no knowledge of *a priori* probabilities, and its detection time is shortest when the session arrival statistics provide enough confidence for reconfiguration. What is more, the algorithm is applicable beyond Poisson traffic arrival model. As long as the inter-arrival times of traffic transactions are independent, the algorithm still reacts fast as the traffic rate changes, which will be discussed in our future work.

IV. Network Transient behaviors

A good understanding of the transient behavior of the queue between each node pair is required for the design of the traffic rate change detection and network reconfiguration algorithms. When the traffic rate or the network configuration is changed, the network queueing delay also changes. Based on the $M/M/m(t)$ queueing model developed before, we focus on both the peak queueing delay and the total duration of a surge/drop from the time that a surge/drop happens to the time that the network reaches a new steady state. Since the processes of traffic surge and traffic drop are quite similar, without loss of generality, we only discuss the modeling of the traffic surge process, where $\lambda(t)$ switches from λ_0 to λ_1.

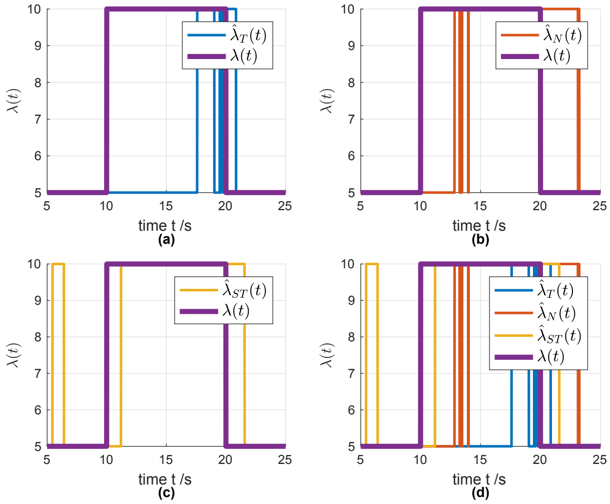


Fig. 5. The comparison of traffic surge/drop detections among different estimators with $p_m = 1\%$. (a) The fixed-time estimator $\hat{\lambda}_T(t)$; (b) The fixed-count estimator $\hat{\lambda}_N(t)$; (c) The stopping-trial estimator $\hat{\lambda}_{ST}(t)$; (d) All three estimators. $\pi_{\lambda_0} = \pi_{\lambda_1} = 0.5, \lambda_0 = 5, \lambda_1 = 10$.

A. *$M/M/1/M$ Queue Transient Behavior and $M/M/m(t)$ Queue Approximation*

Denote $p_n^k(t)$ as the probability that n transactions are in the system at the current time t given k transactions in the system at $t = 0$. Assume M as the maximum number of transactions in the system, where M is very large to approximate the system with the infinite buffer. From the analytical results of transient behavior of $M/M/1/M$ queues in [4], the corresponding time-dependent solution to $p_n^k(t)$ is

$$p_n^k(t) = \pi_n + \frac{2\rho^{\frac{1}{2}(n-k)}}{M+1}\sum_{i=1}^{M}\left(\frac{\mu}{\gamma_i}\right)\cdot \tag{17}$$

$$\cdot\left[\sin\frac{ik\pi}{M+1} - \sqrt{\rho}\sin\frac{i(k+1)\pi}{M+1}\right]\cdot \tag{18}$$

$$\cdot\left[\sin\frac{in\pi}{M+1} - \sqrt{\rho}\sin\frac{i(n+1)\pi}{M+1}\right]e^{-\gamma_i t} \tag{19}$$

where $\rho = \frac{\lambda(t)}{\mu}$, and π_n is the steady state probability for state n in $M/M/1/M$ queue as

$$\pi_n = p_n^k(\infty) = \frac{1-\rho}{1-\rho^{M+1}}\rho^n \qquad n = 0, 1, ..., M. \tag{20}$$

The expression for γ_i is

$$\gamma_i = \lambda(t) + \mu - 2\sqrt{\lambda(t)\mu}\cos\left(\frac{i\pi}{M+1}\right) \tag{21}$$

where $i = 1, 2, ..., M$, and $n = 0, 1, 2, ..., M$. The mean queue length at time t is

$$Q(t) = \sum_{n=1}^{M}(n-1)p_n^k(t) \tag{22}$$

Fig. 6(a) shows both the analytical and simulated transient behaviors of the queue for a traffic surge followed by a proper reconfiguration. t_1 is the time that the traffic arrival rate switches from λ_0 to λ_1. t_2 is the time that the change is detected and the network is reconfigured. We assume that a reconfiguration is completed instantaneously once a surge is detected. t_3 is the time that the network with the new wavelength assignment (*i.e.* the new service rate) reaches the steady state. The detection time is $\tau_1 = t_2 - t_1$ and the queue settling time is $\tau_2 = t_3 - t_2$. The duration of a surge is $\tau_{surge} = \tau_1 + \tau_2$, where we assume other delays are ignorable compared to τ_1 and τ_2. Since the average peak queue size Q_{peak} is reached at t_2, the average peak queueing delay Γ_{peak} is also reached at τ_2. Obviously, both Γ_{peak} and τ_2 depend on τ_1. Therefore, an estimator quickly responding to changes will lead to both a shorter peak queueing delay and a shorter queue settling time.

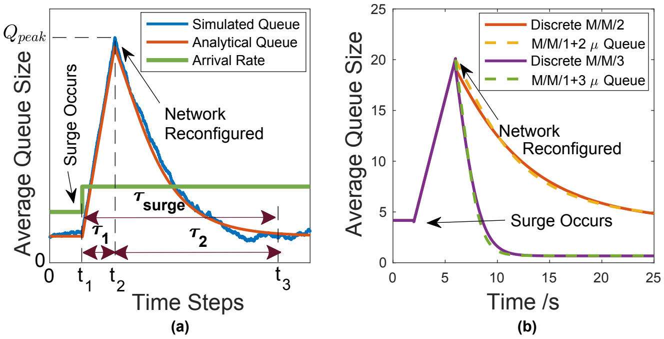


Fig. 6. (a) Simulated and analytical results of the evolution of queue size for a network surge followed by a proper reconfiguration for an $M/M/1/M$ queue; (b) The transient behavior of the average queue size. $\pi_{\lambda_0} = \pi_{\lambda_1} = 0.5, \lambda_0 = 5, \lambda_1 = 10, \mu = 6$.

The analytical results for the transient queueing delay for $M/M/m(t)$ queues are shown in [5] but are hard to use in practice. Instead, we can use the $M/M/1/M$ queue with the service rate $m(t)\mu$ and a large M to approximate the $M/M/m(t)$ queue, since their probability distributions are similar when the network is highly loaded or overloaded. We can also approximate the continuous-time $M/M/m(t)$ queue with a sampled-time $M/M/m(t)$ Markov chain with a very small sample unit time. Figure 6(b) shows the comparison between the $M/M/1$ queue with the service rate $m(t)\mu$ and a large M approximation and the sampled-time $M/M/m(t)$ Markov chain. The agreement of the results shows the analytical results of the $M/M/1$ queue with the service rate $m(t)\mu$ and a large M can approximate the transition behavior of the $M/M/m(t)$ queue well, and we will use the approximation in the following sections.

B. Detection time τ_1

The detection time τ_1 is crucial in the network delay transition as it affects τ_2 and Γ_{peak}. The value of τ_1 depends on both the base detection duration and the detection accuracy.

1) Fixed-time estimator $\hat{\lambda}_T(t)$: $\lambda_T(t)$ has a base detection duration T. If no rate change is detected within T, the algorithm keeps working as time evolves continuously until it catches a change. Let ΔT be the time it takes the algorithm to detect a range after the miss within T. We can model ΔT as a random walk where an arrival comes or leaves in a short unit time δ. Denote η as the threshold of determining the probability of detection p_{d_T} and missed detection $p_{m_T} = 1 - p_{d_T}$. When a surge happens, we have

$$\tau_{1_T} = p_{d_T} T + (1 - p_{d_T})(T + \Delta T) \tag{23}$$

where $\Delta T = \sum_{n=0}^{\lfloor \eta - 1 \rfloor} P(n) E[J] = \sum_{n=0}^{\lfloor \eta - 1 \rfloor} P(n) \frac{(\eta - n)^2}{2\lambda_1 \delta (1 - \lambda_1 \delta)}$ and $P(n)$ follows a Poisson distribution with λ_1.

In the event of false alarms, we have

$$\tau_{1_T} = (1 - p_{f_T}) T + p_{f_T} (T + \Delta T) \tag{24}$$

where $\Delta T = \sum_{n=\lfloor \eta \rfloor}^{\infty} P(n) E[J] = \sum_{n=\lfloor \eta \rfloor}^{\infty} P(n) \frac{(n - \eta)^2}{2\lambda_0 \delta (1 - \lambda_0 \delta)}$ and $P(n)$ follows a Poisson distribution with λ_0.

2) Fixed-count estimator $\hat{\lambda}_N(t)$: $\lambda_N(t)$ needs a duration of N arrivals as the detection time. The results will be updated once a new arrival comes. Then we can formulate the average detection time when a surge happens as

$$\tau_{1_N} = \sum_{n=1}^{\infty} p_{d_N} (1 - p_{d_N})^{n-1} \frac{N + n - 1}{\lambda_1} \tag{25}$$

In the event of false alarms, we have

$$\tau_{1_N} = \sum_{n=1}^{\infty} (1 - p_{f_N}) p_{f_N}^{n-1} \frac{N + n - 1}{\lambda_0} \tag{26}$$

3) Stopping-trial estimator $\hat{\lambda}_{ST}(t)$: $\lambda_{ST}(t)$ needs the average stopping time $E[T_i]E[J]$ for making the decision, in which $E[J]$ could be derived from Wald's equality as $E[S_J] = E[T_i - \frac{1}{\lambda_0}]E[J]$. With $E[T_i] = \frac{1}{\lambda_1}, E[S_J] = \eta_+$, the average detection time when a surge happens is

$$\tau_{1_{ST}} = \frac{E[T_i]E[S_J]}{E[T_i - \frac{1}{\lambda_0}]} = \frac{\lambda_0 \eta_+}{\lambda_0 - \lambda_1} \tag{27}$$

In the event of false alarms, $E[T_i - \frac{1}{\lambda_0}] = 0$, which makes Wald's equality inapplicable. In this case, we can use the second derivative of Wald's identity as $E[S_J^2] = E[J]\sigma^2_{(T_i - \lambda_0)}$. With $E[T_i] = \frac{1}{\lambda_0}, \sigma^2_{(T_i - \lambda_0)} = \frac{1}{\lambda_0^2}, E[S_J^2] = \eta_+^2$, we have

$$\tau_{1_{ST}} = \frac{E[T_i]E[S_J^2]}{\sigma^2_{(T_i - \lambda_0)}} = \lambda_0 \eta_+^2 \tag{28}$$

C. Peak queueing delay Γ_{peak}

For the $M/M/1$ queue, the queueing delay of the incoming transaction with $(n + 1)$ transactions already in the system (*i.e.* n in the queue) is the total transmission time of all $(n + 1)$ transactions. Given the average transmission delay per transaction as $\frac{1}{\mu}$ and the average queue length $Q(t)$, the average queueing delay for the incoming transaction is

$$\Gamma_q(t) = \frac{Q(t) + 1}{\mu} = \frac{[\sum_{n=1}^{N} (n - 1) p_n^m(t)] + 1}{\mu} \tag{29}$$

We can prove that the average queueing delay for the $M/M/m(t)$ queue with μ is the same as that of $M/M/1$ queue with the service rate $m(t)\mu$. The idea is that the queueing delay distribution for the $(n+1)^{th}$ transaction in both queueing systems are $i-$fold convolutions of $1 - e^{-m(t)\mu T}$. Therefore, the average queueing delays that are the weighted average of the distributions are the same.

The normalized peak delay increases with the increase of τ_1 are shown in Fig. 7 , where the normalized average peak delays is the peak queueing delay normalized by the average transmission delay $\tau_{trans} = \frac{L}{R}$. Therefore, a fast response time can help to avoid severe peak queueing delays.

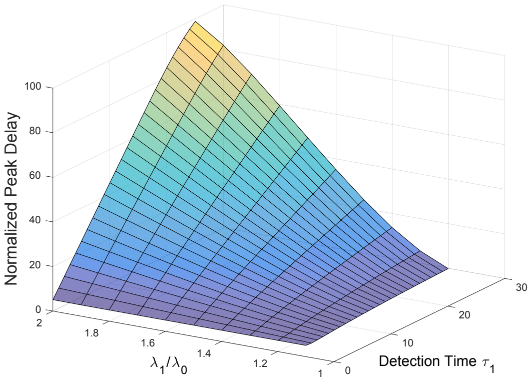


Fig. 7. Normalized average peak delays versus surge changes and detection times τ_1. $\lambda_0 = 5, \mu = 6$.

The transients of the normalized average queueing delay with one or more reconfigurations for different detection times τ_1 averaged over both detections and false alarms are shown in Fig. 8. The performances of different τ_1 diverge after the peak in Fig. 8(a) because p_m and p_f differ. Though a short detection time can lead to a low Γ_{peak}, it also suffers from a high missed detection probability so that the average queue size keeps increasing making the network unstable. A long detection time incurs a high Γ_{peak} in exchange for a low missed detection probability, and increases delays and degrades users' quality of service. Therefore, an optimized detection time is important in designing the cognitive control of wavelength assignment if the detection algorithms without continuous assessments are used. On the other hand, an algorithm with continuous assessments and reconfigurations will reduce the peak queueing delay as shown in Fig. 8(b), where the zig-zag shapes come from correcting errors previously made. Hence, continuous assessments and reconfigurations algorithms compensate for estimators' detection inaccuracy and should be used.

D. *Queue settling time* τ_2

τ_2 is the time that the network needs to serve the sessions accumulated in the queue up to τ_1 and settles to the steady state with the new service rate. We have

$$\tau_2 = t_3 - t_2 \tag{30}$$

where $t_3 = \min_{t > t_2} t, \; s.t. \; \Gamma_q(t) = \Gamma_{steady}$. Γ_{steady} is the new steady state queueing delay after a proper reconfiguration.

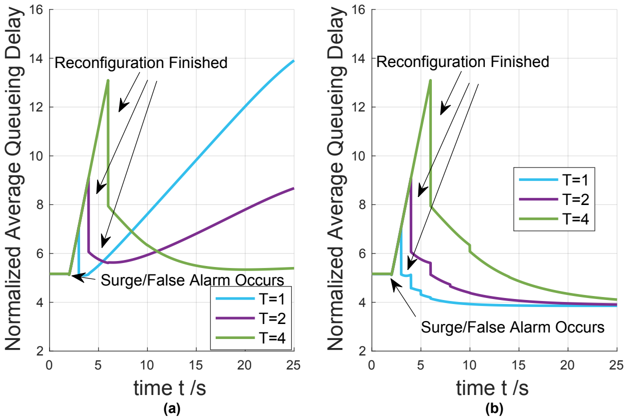


Fig. 8. (a) Average normalized queueing delay for different detection times τ_1 with one reconfiguration; (b) Average normalized queueing delay for different detection times τ_1 with multiple reconfigurations. The results are the average of both detections and false alarms. $\pi_{\lambda_0} = \pi_{\lambda_1} = 0.5, \lambda_0 = 5, \lambda_1 = 10, \mu = 6$.

We can find a bound on τ_2 using the convergence properties of the sampled-time $M/M/m(t)$ Markov chain, and it will be addressed in our future work.

V. Cost-driven network reconfiguration scheme

A. *Network Operating Cost Model*

There is an obvious trade-off between the queueing delay and the cost of a wavelength in determining the reconfiguration algorithm. More wavelengths can bring a lower queueing delay but comes at a higher total cost. The cost of a wavelength includes both the capital expenditure of fiber, switches and amplifiers, and the operating expenditure of setting up wavelengths. Sometimes, it is acceptable to make a wrong decision as long as the incurred total cost is low. On the other hand, we may not want to add a new wavelength for a transient traffic surge, since it may cost much more to reconfigure the network than to tolerate the transient delay increase. Denote the cost parameter C_d as the cost per unit of the normalized queueing delay, and denote the cost parameter C_w as the cost per wavelength. The total wavelength cost for an $M/M/m(t)$ queue is $m(t)C_w$.

Figure 9 shows the transient behaviors of the total costs of the different algorithms if one ore more reconfigurations are allowed averaging with detections and false alarms. $C_w = C_d = 100$, and the target probability of missed detection for all three estimators is 10%. Though the estimators respond differently to the surge, the single decision nature of this particular cases with no further correction upon erroneous actions leads all of them to higher costs eventually driven by the high queueing delay due to the missed detection errors. When continuous assessment with reconfigurations is enabled as shown in Fig.9(b), the system can effectively correct errors and bring down costs, even though missed detections/false alarms have occurred. Due to its fast-response to changes, the stopping-trial estimator requires the shortest time to reconfigure correctly, and yields the lowest total cost.

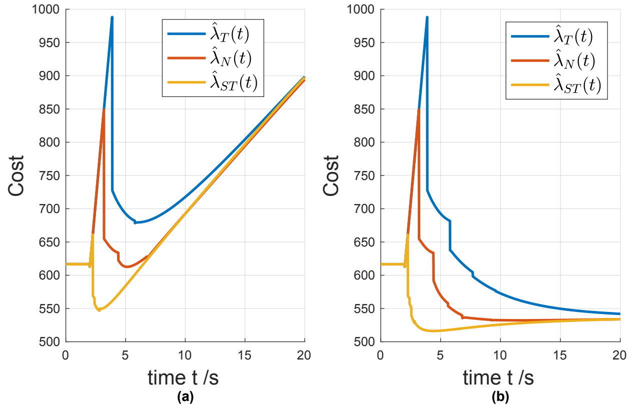


Fig. 9. (a) Cost transition versus time with (a) one reconfiguration; (b) Cost transition versus time with multiple reconfigurations. The results are the average of both detections and false alarms. $\pi_{\lambda_0} = \pi_{\lambda_1} = 0.5, \lambda_0 = 5, \lambda_1 = 10, \mu = 6$. $C_w = C_d = 100$, $p_m = 10\%$.

B. Multiple Wavelengths Addition and Subtraction

The addition and subtraction of multiple wavelengths can be used to deal with severe traffic surges, since more additional wavelengths assigned at once can better reduce the queueing delay. Considering the trade-off between the queueing delay and the cost of any additional wavelength, we need to find an optimal combination of both factors to achieve the optimal total cost. Figure 10 shows the cost comparisons of different estimators with different number of wavelengths assigned. The stopping-trial estimator requests a smaller number of wavelengths realizing higher cost efficiency than the other two estimators, since its fast response helps to avoid a high peak queueing delay.

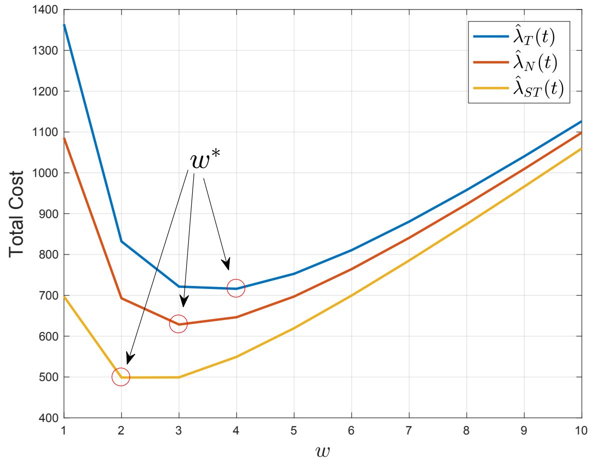


Fig. 10. Cost comparison of different estimators with $p_m = 10\%$. $\pi_{\lambda_0} = \pi_{\lambda_1} = 0.5, \lambda_0 = 5, \lambda_1 = 10, \mu = 6$. $C_w = C_d = 100$.

When the assignment of multiple wavelengths is allowed, it is possible find the optimal number of wavelengths to assign given the total cost constraints. The total cost is

$$C_{total} = C_w w + C_d \Gamma_{peak} \frac{w_0}{w} \tag{31}$$

where w_0 is the number of wavelength assigned before the surge occurs. Setting $dC_{total}/dw = 0$, the optimal number of wavelength w^* is

$$w^* = \left\lceil \sqrt{\frac{C_d \Gamma_{peak} w_0}{C_w}} \right\rceil \tag{32}$$

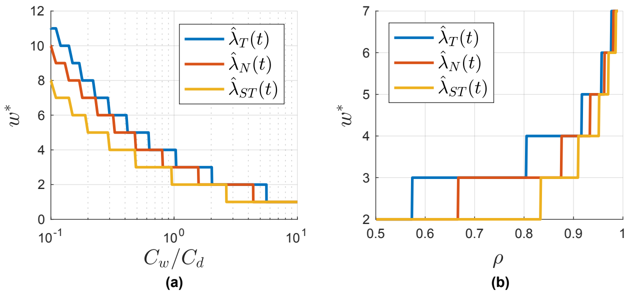


Fig. 11. (a) Optimal number of wavelengths comparison versus cost parameter ratio C_w/C_d of different estimators; (b) Optimal number of wavelengths comparison versus load ρ of different estimators. $\pi_{\lambda_0} = \pi_{\lambda_1} = 0.5, \lambda_0 = 5, \lambda_1 = 10, \mu = 6$. $C_w = C_d = 100$, $w_0 = 1$, $p_m = 10\%$.

We find the stopping-trial estimator uses the smallest number of wavelengths for the same quality of service among all three estimators for the different combinations of cost parameters or different network loads as shown in Fig. 11. The stopping-trial estimator is recommended as the algorithm for reconfigurations.

VI. Conclusion

In this paper, we address the design of a fast-response algorithm for wavelength reconfiguration. Two Bayesian estimators and a stopping-trial sequential estimator are developed to detect changes of traffic arrival statistics. Based on the network transient behaviors of the network, we have shown that the stopping-trial estimator has the shortest detection time for traffic rate changes, and it requires no knowledge of *a priori* probabilities. With continuous assessment, the system reconfigures only when it is necessary. Allowing for the possibility of the addition and subtraction of multiple wavelengths, the stopping-trial estimator (among all three estimators) requires the smallest number of wavelengths to be reconfigured due to its fastest response that helps to avoid a high peak queueing delay.

References

[1] V. Chan, “Cognitive optical networks,” in *International Conference on Communications(ICC)*, May 2018, pp. 1–6.

[2] L. Zhang, “Network management and control of flow-switched optical networks: Joint architecture design and analysis of control plane and data plane with physical-layer impairments,” Ph.D. dissertation, Massachusetts Institute of Technology, 2014.

[3] R. Gallager, *Stochastic Processes: Theory for Applications*. New York, NY: Cambridge University Press, 2014.

[4] P. Morse, *Queues, Inventories and Maintenance: The Analysis of Operational Systems with Variable Demand and Supply*, ser. Publications in operations research. John Wiley & Sons, 1965.

[5] T. Saaty, *Elements of queueing theory: with applications*. McGraw-Hill, 1961.