

Cognitive Optical Networks

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*

Abstract—Future optical networks will have 10^{3-4} increase in rates and highly granular traffic due to large transactions. Cognitive techniques will provide agile automated fast scheduling of resources, topology changes and agile adaptations for congestion control, load balancing and reconfiguration involving all layers.¹

Keywords—Optical network management and control

I. INTRODUCTION

New applications for big data, video and interactive games and future 5G wireless applications will require an increase of network data rates of 10^{3-4} and beyond. The cost per bit/second must be brought down by approximately the same order of magnitude. Overprovisioning for low network service delay is too costly and is not a viable option. Economic efficiency can be achieved by increasing resource sharing through agile network control and application specific network tuning. There are two major research forefronts that are currently being addressed:

1. Physical Layer: massive integration via silicon photonics and hybrids for reduction in footprints, lower costs, weight and power consumption.
2. Development of *agile, responsive and affordable on-demand* network services via a new architecture across ALL network layers.

Current optical networks, including today's software defined networks and orchestration, are operated with predominantly quasi-static connections. Present methods of setting up a wavelength path result in slow changes to the network (~ 10 min setup times), as each of the network element along the path is incrementally tuned to the final settings to avoid instabilities and transient impairments arising from rapidly introducing another optical channel into the network. Fast dynamics due to large flow traffic may use an entire wavelength for seconds to minutes with fast turn-ons and turn-offs, requiring adaptations of the order of ~ 10 ms. In today's optical network, the link quality of all wavelengths in a fiber is monitored as the lightpath is turned on in several steps. Optical Flow Switching [1] enables and coordinates per-session end-to-end all optical lightpath switching over an optical network. Its effectiveness is based on fast dynamic scheduling in a short time scale (< 100 ms) for transaction times of ≥ 1 s. The analysis on control traffic and control computation complexity to including impacts of such impairments showed that the management and control of such futuristic agile optical network is the major obstacle that needs to be addressed, [2]. Changes in offered traffic and

congestion within the network can occur in a second time scale and human cannot be in the loop of an efficient network management and control system. Cognitive techniques can provide fast network state assessments, scheduling and reconfigurations to maintain high network performance. While cognitive techniques can be applied to all networks in general, this paper will only address only ultra-fast optical networks of the future. There have not been many quantitative results pertaining to cognitive optical networks. To illustrate the usefulness of cognitive techniques, this paper analyzes a specific adaptation algorithm of optical network management and control and evaluates the performance of a particular case analytically.

II. COGNITIVE MANAGEMENT AND CONTROL OF AGILE DYNAMIC OPTICAL NETWORKS

A cognitive network management and control system senses current network state conditions such as traffic and flow patterns and uses this information to decide how to adapt the network to satisfy/improve overall performance and provide quick responses to transaction requests. The cognitive network module is part of the control plane that touches all layers of a network, Fig. 1, (it may reside at network nodes as well as at a centralised or distributed controller/s).

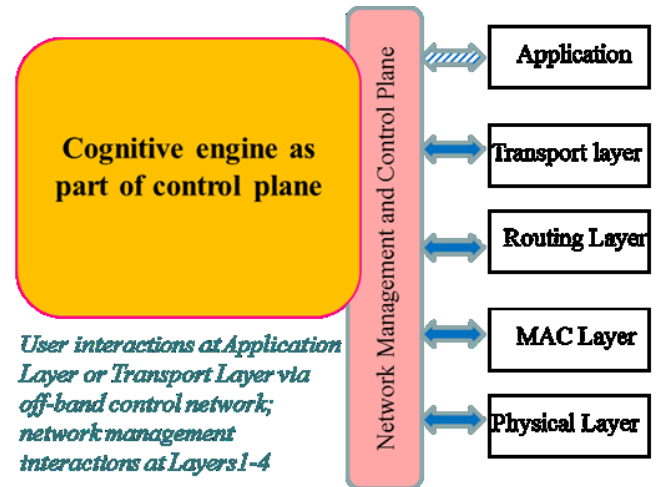


Figure 1. Cognitive engine as part of network control plane.

The design of a cognitive network should consider and answer the following questions (which is very far away from being completed):

1. Should the cognitive processes be centralized, fully distributed, or partially distributed?
2. How should network state information be filtered as the amount of global information grows?

¹ This work was partially supported by the NSF under the NETS program.

3. How should network changes be synchronized among subnets and controllers?
4. Can the network function work well with under-sampled (sparse) and stale link state data?

The number of network parameters and their short coherence time for the very dynamic future networks render acquiring the complete state information of the network impractical, [2]. The idea is to sense and control a small subset of the parameters with the “most” information contents and uses them to perform channel and traffic estimations and tune network performance. The network management and control system implements these objectives:

1. Infers network state based on traffic and active probing with sparsely sampled and stale data.
2. Make decisions on fast scheduling of bursty large transaction requests, load balancing, reconfiguration, and restoration.
3. Predict intention of users and take appropriate actions.

Research in this area has yet to yield many quantitative results and thus the promise of cognitive network is still quite illusive. In this paper, we will use a simple and well bounded problem; formulate and solve it analytically as an illustration of the efficacy of cognitive optical networks.

2.1. An example on fast reconfiguration for all-optical WAN

Consider the following fiber network that provides wide area tunnelled connections between major metropolitan areas in the form of data pipes of capacity C bits per second as depicted in Fig.2, [2]. The capacity assignment between the ingress and egress metro-WAN nodes will be in integral multiples of rate C (the rate per wavelength connection); i.e. $k > 0$, is an integer number of wavelengths each of capacity C . Since the offered traffic is random, the network management and control system needs to observe the input traffic and decide if the average traffic volume has changed and adaptively assigns new capacity or reduces the existing capacity (in number of wavelengths).

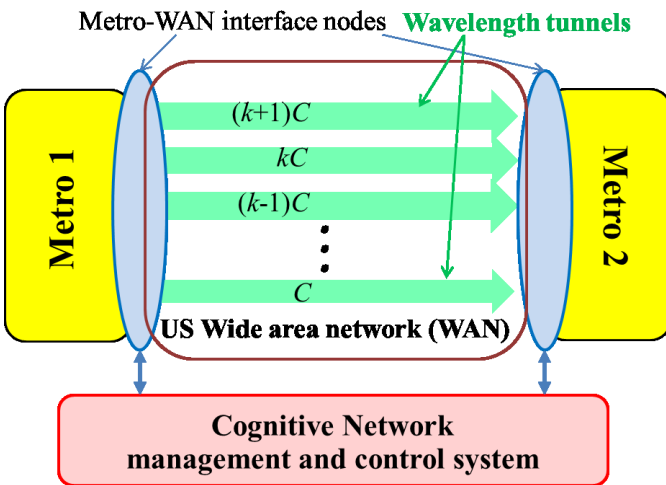


Figure 2. Cognitive network control system monitors traffic arrival at MAN-WAN interfaces triggering reconfiguration.

In this problem we assume flow sessions (typically sessions of lengths $> 1S$) arrive as a conditional Poisson Process with a

rate parameter λ that can change over time and usually is stochastic in nature. The arrival process in general is a doubly stochastic point process, [3]. If the tunnel has k wavelengths, it can transmit k sessions at the same time. When all k wavelengths are busy, sessions are scheduled for future transmissions. For the ingress MAN-WAN node the controller (can be distributed or centralised) maintains a virtual (virtual since the sessions are really held at the senders' terminals) input queue with assigned time and wavelength for each session to transmit. The service time of this system is the amount of time a single wavelength can transmit the session which can be modelled as deterministic (if the sessions are of constant size), exponential or of a general distribution (that can be acquired via learning on past data). If the service time is exponential (i.e. the sessions have exponentially distributed sizes) the queue can be modelled as an M/M/m queue characterised by the Markov Process as depicted in Fig.3. Note in this problem the rate parameter λ , can be time varying and stochastic. Thus Fig. 3 should be viewed as a short term model of the system over one coherence time of λ . It is a good approximation if the number of session arrivals per coherence time is fairly large. When the traffic pattern and thus λ or the number of wavelengths (k) assigned changes, the parameters in the model will change accordingly.

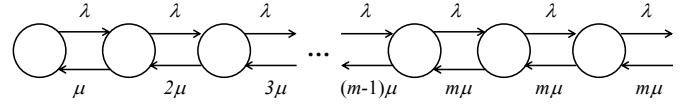


Figure 3. Markov Process Model for M/M/m queue.

If the system had been in steady state before the arrival rate changes, then the prior steady state probability distribution of the queue should be used as the initial condition of the next model and the transient until the next steady state is given by traditional queueing theory such as in “delayed renewal process,” [3]. The key to reconfiguration (in this case decision to increase or decrease the number of wavelengths assigned) can be done via estimates of the change in the arrival rate λ . The following are two simple estimators used in [4,6]. The first estimator simply counts for a fixed amount of time T backwards from the current time t and forms the estimate as shown below. The second estimator, also shown below, fixes the number of arrivals n and determines the amount of time it takes to find n arrivals.

$$\lambda_T = \frac{N(t) - N(t-T)}{T}, \text{ for fixed } T, \text{ and}$$

$$\lambda_n = \frac{N(t) - N(t-T) = n}{T}, \text{ for fixed } n; T = \text{time for } n \text{ arrivals.}$$

The first estimator is simple and easy to find analytical performance bounds. The second is much more responsive to abrupt arrival rate changes especially on fast uptakes or slow down-turns. It only requires n arrivals to make an estimate and the amount of time it takes to make the estimate will adjust to the underlying rate parameter albeit not optimally. Fig. 4 depicts the behavior of the two estimators. The performance is quite different even though n in the second case is picked nominally such that $n = \lambda T$. The value of T and n

must be chosen sufficiently large so the estimation errors are kept small. A small (normalized by the mean m) standard deviation σ/mean can be used with the Chebyshev Inequality to guarantee small false alarm rates and high probability of detection in a Neyman-Pearson Test. Tighter Chernoff Bounds can be used, though the bound will be more sensitive to the distribution of the arrival process, whereas the Chebychev Inequality has the advantage of only depending on the mean and standard deviation of the distribution and thus is more robust.

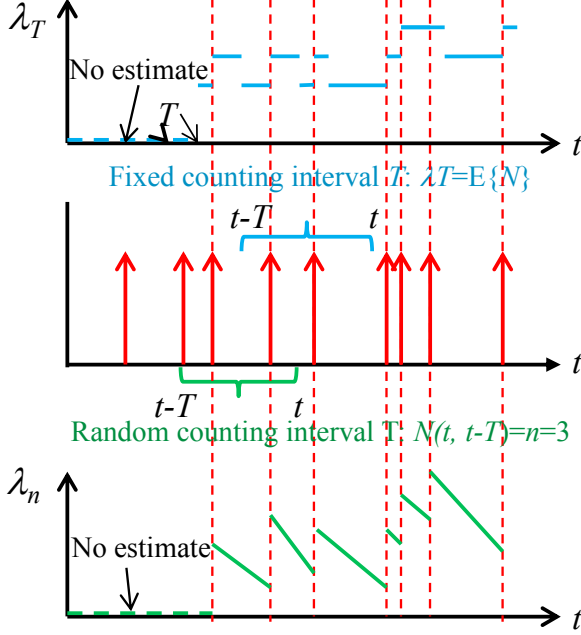


Figure 4. Sample function of Poisson arrival process and estimators behaviour; the arrows are session arrival times.

The optimum Bayesian Estimator is given by considering the arrival times of the flows: $\underline{t} = \{t_i, (-\infty, t)\}$. The estimator is the minimum mean square causal estimator of the arrival rate λ given $\{t_i\}$, using a doubly stochastic point process model for the arrival times of the sessions, [3],

$$\lambda_{MMSE} = \frac{\int_{-\infty}^{\infty} \lambda \exp\left(-\int_{t_0}^t \lambda_{\sigma} d\sigma\right) \prod_{i=1}^{N_t} \lambda_{t_i} p_{\lambda}(\lambda) d\lambda}{\int_{-\infty}^{\infty} \exp\left(-\int_{t_0}^t \lambda_{\sigma} d\sigma\right) \prod_{i=1}^{N_t} \lambda_{t_i} p_{\lambda}(\lambda) d\lambda}$$

Where $p(\lambda)$ is the a-priori probability distribution of the arrival rate parameter as a stochastic process, which are not usually known and also can be nonstationary. This can be estimated over the coherence time of the network from prior traffic statistics using a learning algorithm. The caution with such learning techniques is that it will not do well against an extremely rare or black-swan event. The assumption, that the coherence time of the arrival process is longer than the times that reconfigurations occur, is the regime where such algorithms are useful. In the event that the coherence time is shorter than the times needed for reconfiguration, adaptive techniques will not be effective. The queueing delay in

general is of the form depicted in Fig.5. f is a polynomial depending on the queueing model. The steady state delays for $k-1$, k and $k+1$ wavelengths assigned to the tunnel are shown.

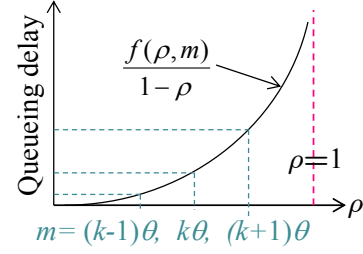


Figure 5. Queueing delay of general service time with $k-1$, k and $k+1$ wavelengths assigned to the tunnel; θ =target load.

Fig. 6 shows the delay in the system when there is a sudden surge of session arrival rates (doubled) at $t=0$ for the estimator λ_T . The estimator is accurate for large T 's but the reconfiguration response is slow whereas small T 's are more agile but exhibit high false alarm probabilities. Both the λ_n and the optimum estimator due to Clark are better but not perfect. In the case of the optimum estimator the algorithm is very hard to implement requiring the prior distribution of the λ process. The desire is to find an algorithm that runs continuously and adapts the system at any time scale and achieve the delay performance of the lower envelope of the various estimators. We will use a technique of sequential decision called "stopping trials" to explore its efficacy.

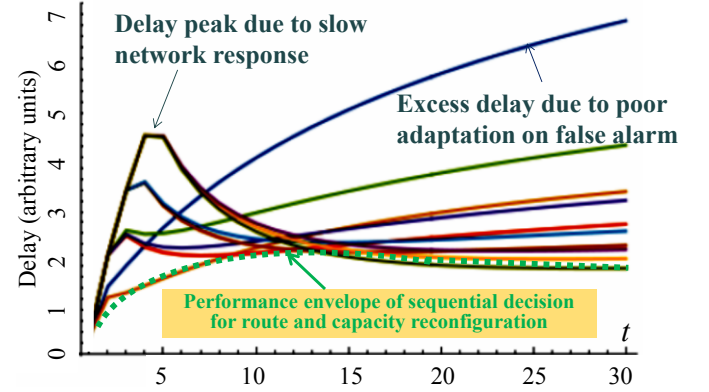


Figure 6. Delay performance of adaptation algorithm.

2.2 Stopping trial: optimum sequential maximum likelihood triggering of reconfiguration, [4].

The cognitive network management and control system monitors the traffic arrival at the metro-WAN (wide-area-network) interface node. We model the traffic arrival to the interface node as a Poisson Process with rate parameter λ arrivals per second, Fig.7, (any arrival process can be used).

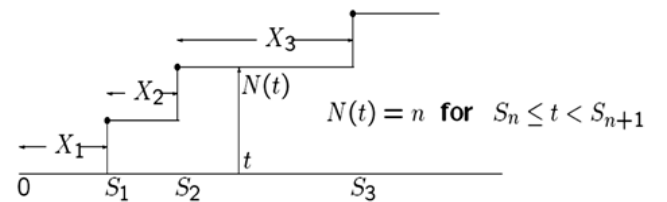


Figure 7. Poisson session arrival process, [5].

For simplicity (to get some simple analytical results to understand the dependency of the behavior of the system to key network parameters but the formulation is good for G/G/m systems) assume each traffic session is a deterministic binary sequence of length L ; hence a M/D/m queueing model. The service time is the transmission time, L/C second. The session inter-arrival times $\{X_i\}$ are independent of each other. The number of session arrivals $N(t)$ jumps by one at every event time $\{S_i\}$. Given λ , the relevant distribution, mean, variance and moment generating functions are:

$$f_X(x) = \lambda \exp(-\lambda x) \quad E[X] = 1/\lambda \quad \sigma_X^2 = 1/\lambda^2 \quad g_X(r) = \frac{\lambda}{\lambda - r}; \quad r < \lambda$$

The semi-invariant moment generating function $\gamma = \log(g)$ is convex and we can use the Chernoff to bound the error probability of a sequential test.

$$\gamma''(r) = \frac{1}{(\lambda - r)^2} \geq 0$$

For a threshold α , the Chernoff bound for the n^{th} session arrival time to exceed a threshold α is, $\Pr\{S_n \geq \alpha\}$ is:

$$\begin{aligned} \Pr\{S_n > \alpha\} &\leq \exp\left\{\alpha \left[\frac{\gamma_X(r_0)}{\gamma_X'(r_0)} - r_0 \right]\right\} \\ &= \exp\{n[\gamma_X(r_0) - r_0(\alpha/n)]\} \\ &= \exp\{-\alpha\lambda + n[\log(\alpha\lambda) - \log(n) + 1]\} \end{aligned}$$

$$\text{where } \gamma_X'(r_0) = \alpha/n$$

2.3 Sequential observations and stopping trials.

We consider the cognitive network management and control system that uses a sequential test (stopping trial) to trigger network reconfigurations. The objective is to find a good algorithm to make a decision at the fastest possible time when the session arrival statistics changes (increases or decreases) by observing the arrival process at the node sequentially. The performance metric is to make sure the queueing delay is not excessive or stay below a preset target value. Taking a long time to decide, after a change in arrival rates, will be unresponsive and increases queueing delays (exceeding threshold set by the performance metric on delay) on the uptake and waste resources on the downturn. Reacting too quickly may seem responsive but wrong (possibly frequent) decisions on noisy data (false alarms) will lead to wasted resources and deny resources for more needy lightpaths and trigger frequent adaptations. Simple decision algorithms such as λ_r and λ_n described above, have the disadvantage of taking too much time to react when there are more arrivals than expected and reacting to noisy data when there are not enough arrivals to give a good estimate of the underlying rate change, resulting in inappropriate and even disruptive reconfigurations. Here the sequential test makes decisions in a responsive manner yet with good accuracies. Thus, the algorithm will make a decision when it thinks there are enough arrivals to reliably make an estimate or when there are

not enough arrivals, it will decide to continue to make observations until it can make a clear choice with a small probability of making an error. The decision algorithm can be modeled as a two threshold crossing problem.

Assume the network has been running (since $t = -\infty$) with traffic session arrival rates of $\lambda = k\theta$ each with length L bits. The capacity used to serve this traffic is kC with average traffic $C\theta k$. The traffic undergoes a change at $t = 0$ to $(k+1)\theta$ or $(k-1)\theta$ as in Fig. 8. For respectable delays θ must be less than 1, possibly in the range 0.4 - 0.7. The idea is when the traffic changes enough to warrant a change in capacity assignment due to delay considerations the network management system will either add or subtract a wavelength of capacity C . We will explore the responsiveness of this system in the following analysis.

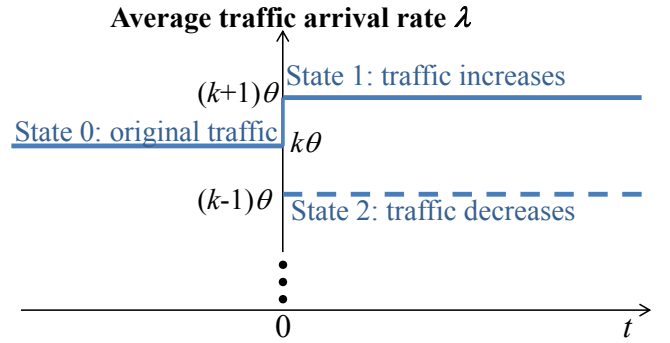


Figure 8. Average traffic Poisson arrival rate with possible change of rate at $t = 0$, from State 0 to State 1, $C=1$.

Define for the traffic in States 0, 1 and 2 respectively as, $Z_n = S_n - n/k\theta$. $E[Z_n] = 0$, $-n/\theta k(k+1)$ and $n/\theta k(k-1)$. We can define the new “inter-arrival times” sequence as: $Y_n \equiv X_n - 1/k\theta$, (C set to 1). Y_n is IID and is zero mean for the traffic in State 0 and $\bar{Y} < 0$ for the traffic in State 1 and $\bar{Y} > 0$ for the traffic in State 2 and the Y 's are IID except possibly across the boundary at $t = 0$. Thus,

$$Z_n = \sum_{i=1}^n Y_i \text{ is a random walk in intervals } t \in (-\infty, 0) \text{ and } (0, \infty).$$

Note that this shifted arrival process is no longer Poisson and the increment Y_n can be either positive or negative with non-zero probability and it has bounded expectation, as shown in Fig.9. It is still a random walk. Technically, Z_n is a zero mean Martingale given the network traffic is in State 0, a Submartingale given the traffic is in State 1 and a Supermartingale given the traffic is in State 2. The algorithm essentially is a sequential maximum-likelihood hypothesis test on the running sequence Z_n or equivalently the increments Y_n . We will assume the process starts at $t = 0$ and the system will eventually either make a transition to state 1 or state 2 from state 0 (the length of time may be unbounded). With no change in traffic load, Z_n will hover around 0, though since Z_n has non-zero variance the process eventually may wander off and trip the thresholds α or β causing an error in adaptation. If an error in adaptation occurs the algorithm is self-correcting in the sense that it will correct its errors after a while since the algorithm is always running. The probability of an erroneous reconfiguration can be set with arbitrary low probability or

long mean free time of occurrence by tuning α and β . Assume a state change occurred at some time say $t = 0$. The optimum hypothesis test is to calculate the likelihood ratio Λ , and compare it to the thresholds at every session arrival time.

$$\Lambda(\bar{Z}) = \frac{f_{\bar{Z}/H}(\bar{Z}|2)}{f_{\bar{Z}/H}(\bar{Z}|1)}$$

The Maximum A Posteriori MAP, rule is :

$$\Lambda(\bar{Z}) \begin{cases} > p_1 / p_2 \text{ select state 2} \\ \leq p_1 / p_2 \text{ select state 1} \end{cases}$$

Equivalently, we can use the log likelihood ratio,

$$\log \Lambda(\bar{Z}) = \sum_{n=1}^m \log \Lambda(y_n), \text{ since } y_n \text{ is IID}$$

Thus, given the state, the two random walks contain the same sequence $\{Z_n\}$ of sample values but with different probability measures, Fig.9.

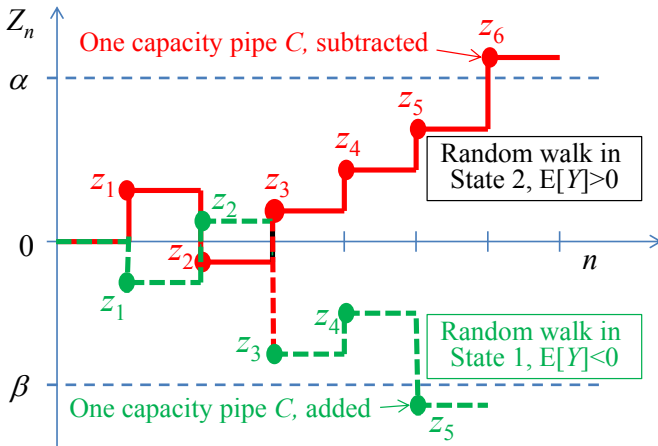


Figure 9. Sample functions of the random walk Z_n , one sample function for traffic in State 1 and one for State 2.

The network management system will run the sequential detection test with two thresholds α and β as shown in Fig.9. When the traffic increases (system in State 1), the random walk Z_n will drift towards crossing β (inter-arrival times are shorter and thus the negative drift) and result in the addition of a unit of capacity C . When the traffic decreases (system move to State 2), Z_n will drift towards crossing α (the inter-arrival times are longer) and result in a subtraction of capacity C . With no change in traffic load, Z_n will hover around 0, though since Z_n has non-zero variance the process eventually may wander off and trip the thresholds α or β causing an error in adaptation. Thus the magnitude of α and β must be set large enough so the probability of making such an error is small. However, since this algorithm is continuously running, any errors can be corrected eventually as the process evolves. If the traffic undergo a change in state at $t = 0$ to either State 1 or State 2 from State 0, it is informative to find out what is the mean time to tripping one of the thresholds. This can be found by using a technique for running sequential tests called stopping trials. Let J (the discrete stopping trial variable) be the number of arrivals after $t = 0$ that the network management system will add or subtract capacity. The

expected value of J given the traffic is in State 1 and 2 can be found using Wald's Identity and Equality, [5].

Wald's Identity: $E[\exp(rZ_J - J\gamma(r))] = 1$;

Wald's Equality: $E[S_J] = \bar{Y}E[J]$

$$\gamma(r) = -\frac{r}{k\theta} + \log \frac{\lambda}{\lambda - r}; \quad \gamma'(r) = -\frac{1}{k\theta} + \frac{1}{\lambda - r};$$

$$\gamma_1(0) = -\frac{1}{k\theta} + \frac{1}{(k+1)\theta} = -\frac{1}{\theta k(k+1)}$$

$$E[J|1] = \bar{J}(1) = \frac{\beta}{E[Z|1]} = \frac{\beta}{\gamma_1'(0)} = -\beta\theta k(k+1)$$

$$E[J|2] = \bar{J}(2) = \frac{\alpha}{E[Z|2]} = \frac{\alpha}{\gamma_2'(0)} = \alpha\theta k(k-1); \text{ for } k > 1$$

Thus, the mean number of arrivals that will trigger the addition of a wavelength is given by $E[J|1] = -\beta\theta k(k+1)$. Not surprisingly, the mean number of session arrivals until a threshold is tripped is proportional to the magnitude of β and the target load factor, θ . However, $E[J/H]$ for each hypothesis H , is proportional to the square of the current number of wavelengths k , assigned. The reason is with k wavelengths and at a nominal load away from the threshold the k wavelengths will share the arrivals evenly accounting for one factor of k and the other factor of k comes from the service rate of k wavelengths. An exponentially tight approximate expression for the error probability is:

$$\Pr\{e|1\} \sim \exp\{E[J|2]\gamma_2'(1)\} = e^{-\alpha}, \quad \Pr\{e|2\} \sim \exp\{E[J|1]\gamma_1'(1)\} = e^{-\beta}$$

Now we can treat the case when there is no state change (i.e. the network stays in State 0), and find the probability $\Pr\{e|0\}$, of making an erroneous reconfiguration as the network management system runs in real time. We also want to determine if the algorithm works when we do not know a traffic change has occurred. We can use the Chernoff Bound as above to estimate this probability but here we use the Chebychev Inequality instead since the bound only requires mean and variances and not knowledge of the detailed distributions. Hence the estimate is more robust to variations of the underlying statistic of session arrivals.

$$\begin{aligned} \Pr\{e|0\} &= \Pr\{Z_n \text{ crossing } \alpha \text{ or } \beta\} \\ &= \Pr\{Z_n \text{ crossing } \alpha\} + \Pr\{Z_n \text{ crossing } \beta\} \end{aligned}$$

Using Kolmogorov's random walk inequality on Z_n after m steps:

$$\Pr\left\{\max_{1 \leq n \leq m} |S_n - n\bar{X}| > m\varepsilon\right\} \leq \frac{\sigma^2}{m\varepsilon^2}$$

$$\Pr\{Z_n \text{ crossing } \alpha\} = \Pr\left\{\max_{1 \leq n \leq m} |Z_n| \geq m\varepsilon = \alpha\right\} \leq \frac{m\sigma^2}{\alpha^2} = \frac{m}{(\alpha k\theta)^2}$$

$$\sigma_y^2 = 1/\lambda^2 = 1/(k\theta)^2$$

$$\Pr\{Z_n \text{ crossing } \beta\} = \Pr\left\{\max_{1 \leq n \leq m} |Z_n| \geq m\varepsilon = |\beta|\right\} \leq \frac{m\sigma^2}{|\beta|^2} = \frac{m}{(\beta k\theta)^2}$$

$$\Pr\{e|0, \text{ after } m \text{ arrivals}\} \leq \frac{m}{(k\theta)^2} \left(\frac{1}{\alpha^2} + \frac{1}{\beta^2} \right)$$

Thus the error probability linearly increases with m and decreases as the square of α and β . Even if there has been an erroneous reconfiguration due to tripping of one of its threshold, the algorithm will self-correct after a while. It is possible to determine how quickly the algorithm can correct itself, by modeling the elapsed interval from erroneous adaptation to self-corrections as a renewal process. The expected time to tripping any threshold (including recovery via self-correction after making an error) can be obtained using first passage time techniques. The frequencies at which the errors occur can be obtained using renewal theory and Blackwell's Theorem. These quantities can be obtained analytically but we will save those for another time. Such information will be important and very useful for the network architecture to tune the algorithm and the value of the thresholds for good but not jittery response.

III. SUMMARY AND DISCUSSIONS

Future network management control systems must be agile and fast adaptive due to significant increase in large data volume and session granularities (a single transaction may occupy entire wavelength for a transmission period). These dynamics will demand network management and control of a time scale as fast as $\sim 10\text{ms}$ compared to minutes and hours of current networks. Sensing schemes will be employed where each node and link selectively (as in judiciously) senses the network state (in Layers 1, 2 and 3). This information can be combined across nodes by a centralized or distributed controller to perform scheduling, complex statistical inference and network reconfigurations/load-balancing and even more drastic operations such as isolating suspicious subnet upon detection of anomalies for information assurance purposes. Cognitive techniques can eliminate human in the loop for network management and rapidly (and statistically optimally or near optimally) estimate network state trajectories and optimize network configurations. Learning algorithms can be used to estimate from historical data the nominal traffic statistics. In this paper, we use a specific example that yields analytical results to illustrate the efficacy of cognitive optical networks but it is merely a hint of what a more extensive set of such algorithms can do.

The cognitive algorithm described here is Bayesian based, while other cognitive techniques may be used in other contexts to manage optical networks. The algorithm has four phases:

1. Observation of session arrivals
2. Estimation of arrival rate changes
3. React to increase or decrease of traffic and add or subtract a wavelength in the tunnel between two MANs.
4. Continue assessment for further changes or self-correct erroneous reconfigurations.

The first three traffic estimators all require some knowledge of the prior statistics of the session arrival processes. The first estimator, λ_T , uses λ to determine the amount of observation time T , to form an estimate. The second estimator, λ_n , predetermines the number of arrivals n , before the estimator makes a decision. This has disadvantages, among which is:

when the traffic suddenly reduces drastically, the algorithm does not react quickly. In fact if there are no arrivals the estimator will not output a decision. On the other hand, the stopping trial algorithm takes care of this case very gracefully. The third (optimum) estimator by Clark needs the prior statistical $p(\lambda)$, of the arrival process, which is hard to determine. Any distributions given by idealised theoretical models will not be totally accurate (though probably tolerable). Learning from past data will not yield reliable models for rare events especially when the network is close to a serious congestion breakdown or at the on-set of zero day events.

The stopping trial algorithm analysed here does not have the shortcomings of the first three estimators. Its key assumption is that the arrivals are independent and identically distributed if the arrival process is in a quasi-static state where λ does not change much. The distribution of the inter-arrival times do not have to be exponential as used here to get simple analytical results. As long as the arrival process can be approximately modelled as a renewal process (a special case of random walk), the algorithm will work. The algorithm performs well when the observation and decision process time duration is shorter than the coherence time of the session arrival process. When it takes longer than the arrival process coherence time to estimate, the algorithm is not expected to perform well. In those cases, it is not clear adaptive reconfigurations are beneficial. Perhaps over-provisioning is the only sensible solution then.

One superior characteristic of the stopping trial algorithm (it is the optimum maximum a-posteriori sequential test) is that it will make a decision whenever it determines it can do it reliably; not before or after. This 'just in time' behavior saves time and prevents unnecessary errors when there are insufficient observations. When the session arrival rate surges, the algorithm makes a decision as soon as the threshold, which is set with a target (low) probability of error, is tripped. On the downturn, it can be set to trip the threshold even if there are no arrivals, which is possible). The algorithm runs continuously and will self-correct reconfiguration errors. It is important for the network architect to use techniques such as 'first passage times' and 'renewal times' to estimate the frequency of errors and tune the thresholds away from unacceptable disruptive frequent reconfigurations. The choice will be a sensible balance of times to adaptation versus frequency of errors and coherence times of the network.

IV. REFERENCES

- [1] Vincent W. S. Chan, "Optical Flow Switching Networks," Special Issue, Proceeding of IEEE, March 2012.
- [2] Zhang Lei and Vincent W.S. Chan, "Joint Architecture of Data and Control Planes for Optical Flow Switched Networks," IEEE ICC June 2014 Sydney Australia
- [3] John Clark, "Estimation for Poisson processes with applications in optical communication," MIT PhD Thesis 1971.
- [4] Esther Jang, "Characterization and Performance Analysis of a Cognitive Routing Scheme for a Metropolitan-Area Sensor Network," MIT MEng Thesis, June, 2016.
- [5] Robert Gallager, "Stochastic Processes, Theory for Applications," Cambridge University Press 2013.
- [6] Vincent Chan and Esther Jang, "Cognitive All-Optical Fiber Network Architecture," ICTON 2017, Girona, Spain.