

Cognitive Network Management and Control with Significantly Reduced State Sensing

Arman Rezaee, *Student Member*, Vincent W.S. Chan, *Life Fellow IEEE, Fellow OSA*

Claude E. Shannon Communication and Network Group, RLE

Massachusetts Institute of Technology, Cambridge MA, USA

Emails: {armanr, chan}@mit.edu

Abstract—Future networks have to accommodate an increase of 3-4 orders of magnitude in data rates with very heterogeneous session sizes and sometimes with strict time deadline requirements. The dynamic nature of scheduling of large transactions and the need for rapid actions by the Network Management and Control (NMC) system, require timely collection of network state information. Rough estimates of the size of detailed network states suggest large volumes of data with refresh rates commensurate with the coherence time of the states (can be as fast as 100 ms), resulting in huge burden and cost for the network transport (~ 300 Gbps/link) and computation resources. Thus, judicious sampling of network states is necessary for a cost-effective network management system. In this paper, we consider a construct of an NMC system where sensing and routing decisions are made with cognitive understanding of the network states and short-term behavior of exogenous offered traffic. We have studied a small but realistic example of adaptive monitoring based on *significant sampling* techniques. This technique balances the need for accurate and updated state information against the updating cost and provides an algorithm that yields near optimum performance with significantly reduced burden of sampling, transport and computation. We show that our adaptive monitoring system can reduce the NMC overhead by a factor of 100 in one example. The essential spirit of the cognitive NMC is that it collects network states **ONLY** when they matter to the network performance.

I. INTRODUCTION

We are in the midst of a major technological storm that will change the landscape of networking for years to come. According to Cisco, IP video traffic will be 82 percent of all IP traffic by 2021, up from 73 percent in 2016 [1], [2]. The same reports forecast live video to grow 15-fold while virtual reality and augmented reality traffic will increase 20-fold in the same period. The bursty and dynamic nature of the traffic generated by these applications require quick (100 ms-1 s) network adaptation to maintain quality of service and experience [3]. Unfortunately, current Network Management and Control (NMC) systems are much too slow and their operational paradigm does not scale well with network size and traffic intensity.

To illustrate the efficacy of our cognitive management approach, we will use an example that primarily focuses on challenges involved with ‘shortest path’ routing in dynamic networks. In this context, an NMC system has to monitor the state of queues throughout the network to decide which path(s)

should be used to connect different origin-destination (OD) pairs. In practice, various shortest path algorithms (Bellman-Ford, Dijkstra, etc.) are used to identify the optimal shortest path. The correctness of these algorithms requires the assumption that after some period of time the system will settle into its steady state. We refer interested readers to [4] for a thorough explanation of these issues. With the dynamic nature of current and future networks, the steady state assumption is particularly not valid, nonetheless the basic functional unit of these algorithms can be adapted to address the new challenges.

Shortest path routing algorithms assign a length/weight to each link of the network; this length is usually a proxy for traffic congestion on the link but can also incorporate other factors. Depending on the specific implementation, the length may depend on the number of packets waiting in the queue, loading of the output line, or the average packet delay on the link during a pre-specified amount of time. Neighboring nodes exchange their estimated shortest distances to all other nodes periodically and new information will be disseminated throughout the network after a few rounds of exchanges.

Suppose the routing table for each node contains the available paths to each destination and the latest estimates of their lengths, as shown in Table I. Given this table, a node can identify the shortest path to all destinations, but more importantly it can use the relative difference between the length of the shortest path and that of other paths to determine the likelihood that the identity of the shortest path changes in the near future. As a result, as opposed to

Table I: Partial routing table maintained by node *A*

Destination	Path	Length
Y	$B \rightarrow C \rightarrow D \rightarrow Y$	2
	$E \rightarrow F \rightarrow G \rightarrow Y$	3
	$H \rightarrow I \rightarrow Y$	8
Z	$J \rightarrow K \rightarrow Z$	4
	$L \rightarrow M \rightarrow Z$	7
	$N \rightarrow O \rightarrow P \rightarrow Z$	9

updating the whole table at a minimum rate required for all dynamic situations, each entry of the routing table will be updated only when it can be of significant value to the optimal operation of the network. It is in this sense that we refer to our technique as *significant sampling*. To illustrate this principle, we will focus primarily on a simplified model where an OD pair is connected via exactly two independent paths with time varying lengths/weights. Within this framework, we

develop an adaptive monitoring system that determines the value of updating the length/weight of a given path as well the cost associated with the updating process. This allows the NMC system to optimally allocate its resources to collect and disseminate information regarding the state of network elements according to their importance.

The rest of the paper is organized as follows: Section II introduces the general stochastic model and establishes the framework through which the monitoring process is optimized. Section III-A assumes a Wiener process as the end-to-end delay process and evaluates the benefits and shortcomings of this model. Section III-B applies the framework to the much richer Ornstein-Uhlenbeck process. Section IV extends the results to general networks. A summary and concluding remarks are provided in Section V.

II. PROBLEM SETUP – GENERAL MODEL

Suppose that an OD pair is connected via two independent paths P_1 and P_2 as illustrated in Figure 1, and denote the stochastically evolving weight of P_i by $X_i(t)$.¹

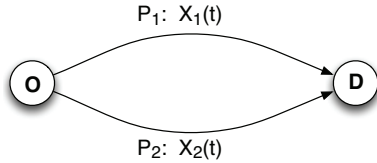


Figure 1: An OD pair with two independent paths.

Let us use $X(t)$ to denote the stochastic process that results from subtracting the weights of the two paths, i.e., $X(t) = X_2(t) - X_1(t)$. Clearly, continuous optimal routing can be achieved if we know whether or not $X_1(t) < X_2(t)$. This is equivalent to knowing the sign of $X(t)$ for all t . Since continuous monitoring of $X(t)$ is far from cost effective, our aim is to identify a strategy that specifies the best future updating times based on the last observed value of $X(t)$. More concretely, consider a sample function of $X(t)$ as shown in Figure 2. Given $X(t_0)$, (i.e., the state of both elements at time t_0), we would like to identify the next epoch t_1 for updating our routing tables. When the value of the function at t_1 , i.e. $X(t_1)$, is realized, we will use it to determine the following updating time t_2 and so forth. Notice that T_i 's and $X(t_i)$'s are the fundamental random variables defined recursively as

$$T_i = f\left(X(t_{i-1})\right) \quad \text{and} \quad t_i = t_{i-1} + T_i$$

Between two updating epochs t_{i-1} and t_i , the process $X(t)$ will evolve according to an underlying stochastic model. Recall that P_1 is the optimal route if $X(t) > 0$ and P_2 is the optimal route if $X(t) < 0$. So the communicating OD pair will experience an excess cost if the process $X(t)$ changes sign between two sampling epochs and the transmission route is not adapted. Let us use $C_{[t_{i-1}, t_i]}$ to denote the cost of such errors during $[t_{i-1}, t_i]$. Without loss of generality, assume

¹ $X_i(t)$ can be the rate of transmission (messages/s) times the expected delay/message on P_i , which would give it units of delay/s.

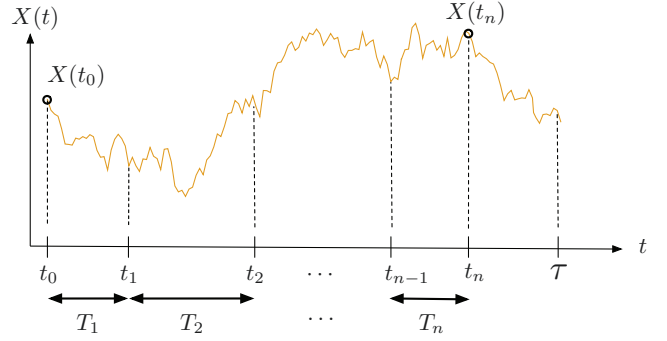


Figure 2: Illustration of n samples taken during a period τ .

$X(t_{i-1}) > 0$ and suppose that the OD pair uses P_1 as the route during $[t_{i-1}, t_i]$. Then the cost of error during this period is simply the integral of $X(t)$ after the process experienced a sign change

$$C_{[t_{i-1}, t_i]} = - \int_{t_{i-1}}^{t_i} X^-(t) dt$$

where $X^-(t)$ is the negative part of the function defined as $X^-(t) = (X(t) - |X(t)|)/2$. When the stochastic nature of the underlying process is known, a distribution is induced over possible sample paths of $X(t)$, and we can use this distribution to compute the expected cost of error during this period, denoted by $\mathbb{E}[C_{[t_{i-1}, t_i]}]$.

We shall now introduce the notion of *updating cost* to capture the efforts required to collect and disseminate state information throughout the network so that all routing tables are up-to-date. Google Maps offers a great example of various costs associated with such efforts, as the system gathers congestion information from individual drivers on the road. The GPS-enabled device will incur a data transmission cost by using the wireless services, and will drain its battery as a result of the required computation and communication. Furthermore, there is a cost associated with reporting the latest congestion information to all drivers. In general, the updating cost may vary over time and can differ for various network elements, but for simplicity we will use a single figure of merit, c , to account for the cost. This is a *catchall* quantity that should represent the costs associated with collection and dissemination of the routing information across the whole network. If we update the routing tables by sampling the process n times during $[0, \tau]$, as shown in Figure 2, then the expected total cost C_{Total} resulting from sampling and unintended errors is

$$\mathbb{E}[C_{Total}] = \sum_{i=1}^n (c + \mathbb{E}[C_{[t_{i-1}, t_i]}]) + \mathbb{E}[C_{[t_n, \tau]}]$$

For a given time horizon $[0, \tau]$, an optimal *offline* sampling strategy will identify the optimal number of samples and their corresponding epochs to minimize the expected total cost. Alternatively, we may choose to minimize the cost per unit time for an infinite time horizon. These possible approaches can be summarized as

$$\operatorname{argmin}_{n, \{t_1 < \dots < t_n \leq \tau\}} \mathbb{E}[C_{Total}]$$

$$\text{or} \quad \underset{n, \{t_1 < \dots < t_n \leq \tau\}}{\operatorname{argmin}} \lim_{\tau \rightarrow \infty} \frac{\mathbb{E}[C_{Total}]}{\tau}$$

The aforementioned formulations are not as useful in practice because routing and updating decisions should be made in real-time. In other words, an NMC is interested in making routing and sampling decisions for the immediate future and cannot afford to plan too far into the future. Let us use $C_{[t_{i-1}, t_i]}(x)$ to denote the cost of error during $[t_{i-1}, t_i]$ given that $X(t_{i-1}) = x$. Then the most informative formulation computes the optimal updating period T_i^* as

$$T_i^*(x) = \underset{T_i > 0}{\operatorname{argmin}} \frac{c + \mathbb{E}[C_{[t_{i-1}, t_i]}(x)]}{T_i}$$

Furthermore, if $X(t)$ is Markovian, then the distribution of its trajectory is only a function of its last observed value. Hence, for Markov processes $\mathbb{E}[C_{[t_i, t_i + \tau]}(x)] = \mathbb{E}[C_{[0, \tau]}(x)]$. As a result, for such processes we can treat every sampled value as if it had occurred at time zero. Using $C_T(x)$ as a shorthand for $C_{[0, T]}(x)$, we can rewrite our optimization as

$$T_1^*(x) = \underset{T > 0}{\operatorname{argmin}} \frac{c + \mathbb{E}[C_T(x)]}{T} \quad (1)$$

This concludes our discussion of the general setting of the problem and the relevant optimization formulations. The following section focuses on appropriate delay models.

III. DELAY MODELS

Queues constitute one of the basic building blocks of a communication network and have been extensively studied for decades [5]. While queuing theory has provided immense insight into the operation of data networks, it has struggled in providing tractable expressions that deal with transient behavior of queues (which is of immense importance to us!). Referring to the transient behavior of an $M/M/1$ queue, Kleinrock notes: “This last expression is disheartening. What it has to say is that an appropriate model for the simplest interesting queueing system [i.e. the $M/M/1$ queue] leads to an ugly expression for the time-dependent behavior of its state probabilities” [5]. Hence, we advocate an approach whereby operational insights can be provided through judiciously chosen alternative models which lend themselves to easier analysis and can be insightful as engineering guidelines for the design of future networks. Desirable transient models of delay in data networks should have the following characteristics:

- **Stability** - the stochastic model of delay should be *stable* so that linear combination of two or more such processes will remain in the same family of distributions.
- **Stochasticity** - capture the rise in uncertainty about state of the network as more time passes from last observation.
- **Simplicity** - provide a formulation that is amenable to analysis and/or numerical computation.

It is often beneficial to approximate the process with an appropriate diffusion process. The inherent structure of diffusion

processes makes it easier to avoid some of the combinatorial challenges involved in the original problem. The following two sections describe two such models of delay.

A. Wiener Process

Numerous successful attempts have been made at modeling queuing delay as a Wiener process. Most notably, it has been shown that the normalized queue length in heavy traffic (i.e., as $\rho \rightarrow 1$) can be approximated by a one-dimensional reflected Wiener process, also known as the reflected Brownian motion [6], [7]. Modeling the waiting time of customers as a Wiener process satisfies our modeling criteria because 1) Wiener process is stable, hence the waiting time of customers in cascaded series of independent links would itself be a Wiener process, 2) uncertainty in the realization of a sample path of a Wiener process grows with time.

We shall define and note a few properties of the Wiener process and refer the interested reader to [8] for a thorough investigation of general properties of the Wiener process.

Definition 1. A real valued stochastic process $\{W(t) : t \geq 0\}$ is a Wiener process with a start at $x \in \mathbb{R}$ if the following hold:

- $W(0) = x$
- the process has independent increments.
- for all $t \geq 0$ and $h > 0$, the increments $W(t+h) - W(t)$ are normally distributed with zero mean and variance h .
- the function $W(t)$ is continuous almost surely.

Lemma 1. A Wiener process $\{W(t) : t \geq 0\}$ with a start at 0, has following pdf and cdf functions

$$\begin{aligned} f_{W(t)}(x) &= \frac{1}{\sqrt{2\pi t}} \exp\left(-\frac{x^2}{2t}\right) \\ F_{W(t)}(x) &= \frac{1}{2} \left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2t}}\right)\right) \end{aligned}$$

where $\operatorname{erf}(\beta) = \frac{2}{\sqrt{\pi}} \int_0^\beta e^{-z^2} dz$.

Recall that we used $X(t)$ to denote the stochastic process that results from subtracting the weights of the two paths, i.e. $X(t) = X_2(t) - X_1(t)$. If each $X_i(t)$ is approximated as an independent Wiener process, we can see that $X(t)$ is another Wiener process with a start at $X(0) = X_2(0) - X_1(0)$ and a variance equal to the sum of the variances of the two processes. Without loss of generality, suppose that $X(t)$ has a variance of $\alpha^2 t$ for some $\alpha \in \mathbb{R}$, and assume that $X(0) = x > 0$. This is equivalent to assuming $X(t) = x + \alpha W(t)$. As before, we will assume that the OD pair uses P_1 as the shortest route until the next update time at t_1 . Noting that Wiener process is Markovian and using $C_T(x)$ as a shorthand for $C_{[0, T]}(x)$, we can see that routing through P_1 for T seconds will incur the following cost of error

$$\begin{aligned} C_T(x) &= -\int_0^T X^-(t) dx = -\int_0^T \min\{X(t), 0\} dt \\ &= -xT - \alpha \int_0^T \min\left\{W(t), \frac{-x}{\alpha}\right\} dt \end{aligned}$$

with an expected value of

$$\mathbb{E}[C_T(x)] = -xT - \alpha \int_0^T \mathbb{E} \left[\min \left\{ W(t), \frac{-x}{\alpha} \right\} \right] dt$$

Since x/α is a constant and is independent of $W(t)$, we can compute the distribution of the minimum as

$$\Pr \left\{ \min \left\{ W(t), \frac{-x}{\alpha} \right\} \leq x \right\} = \begin{cases} F_{W(t)}(x) & \text{for } x < \frac{-x}{\alpha} \\ 1 & \text{for } x \geq \frac{-x}{\alpha} \end{cases}$$

Hence,

$$\begin{aligned} \mathbb{E} \left[\min \left\{ W(t), \frac{-x}{\alpha} \right\} \right] &= -\frac{\sqrt{t}}{\sqrt{2\pi}} \exp \left(-\frac{x^2}{2t\alpha^2} \right) \\ &\quad - \frac{x}{2\alpha} \left[1 + \operatorname{erf} \left(\frac{x}{\alpha\sqrt{2t}} \right) \right] \end{aligned}$$

Putting it all together we have

$$\begin{aligned} \mathbb{E}[C_T(x)] &= \frac{\sqrt{T}(2T\alpha^2 + x^2)}{3\alpha\sqrt{2\pi}} \exp \left(-\frac{x^2}{2\alpha^2 T} \right) \\ &\quad - \operatorname{erfc} \left(\frac{x}{\alpha\sqrt{2T}} \right) \left[\frac{xT}{2} + \frac{x^3}{6\alpha^2} \right] \end{aligned}$$

which can be used with Eq. (1), as restated below, to get the optimal sampling period

$$T_1^*(x) = \operatorname{argmin}_{T>0} \frac{c + \mathbb{E}[C_T(x)]}{T}$$

Interestingly, the shortest possible sampling period can be computed by noting that $\mathbb{E}[C_T(x)]$ is strictly decreasing in x , and thus for any $T > 0$ the expected cost of error is minimized at $x = 0$. Solving the first order optimality condition gives us

$$\min_x T_1^*(x) = T_1^*(0) = \left(\frac{18\pi c^2}{\alpha^2} \right)^{\frac{1}{3}} \quad (2)$$

This is an important quantity as it constitutes a maximum updating frequency for all “fixed-period” (i.e. uniform) updating strategies. In other words, if the NMC updates occur any faster, the cost of updates will be larger than the potential gains from identifying the optimal route. Furthermore, Eq. (2) shows that increasing the updating cost c by a factor of γ reduces the frequency of updates (i.e. $1/T_1^*$) by a factor of $\gamma^{2/3}$; while increasing parameter α has the inverse of that effect.

Noting the complexity of the general optimization, let us use a simple first order method to approximate the optimal updating period T_1^* . Furthermore, to simplify our notation, we will assume that $\alpha = 1$ in the remainder of this section.

$$\begin{aligned} \mathbb{E}[C_T(x)|\alpha = 1] &= \frac{\sqrt{T}(2T + x^2) \exp \left(-\frac{x^2}{2T} \right)}{3\sqrt{2\pi}} \\ &\quad - \operatorname{erfc} \left(\frac{x}{\sqrt{2T}} \right) \left[\frac{xT}{2} + \frac{x^3}{6} \right] \end{aligned}$$

hence,

$$\begin{aligned} \frac{\partial}{\partial T} \left(\frac{c + \mathbb{E}[C_T(x)|\alpha = 1]}{T} \right) &= \\ \frac{1}{T^2} \left(-c + \frac{\sqrt{T}(T - x^2) e^{-\frac{x^2}{2T}}}{3\sqrt{2\pi}} + \frac{x^3}{6} \operatorname{erfc} \left(\frac{x}{\sqrt{2T}} \right) \right) &= 0 \quad (3) \end{aligned}$$

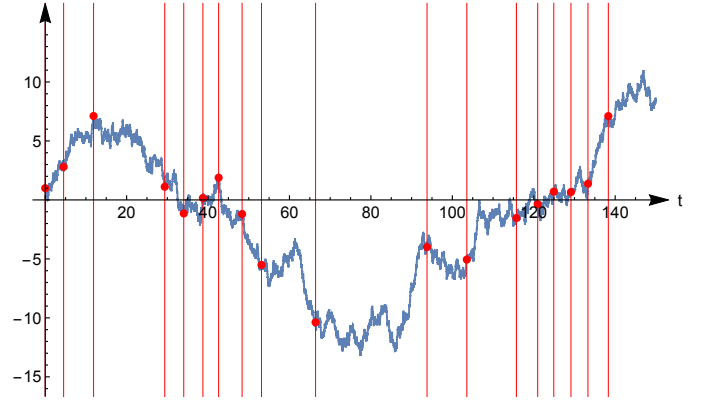


Figure 3: Sample path of a Wiener process, and associated samples for $c = 1$.

which can be solved numerically to obtain the optimal T_1^* for any given x . It can be shown that solving Eq. (3) is approximately equal to solving the following equation for T

$$x^2 = T \ln \left(\frac{T^3}{18\pi c^2} \right)$$

Figure 3 depicts a sample path of $X(t) = 1 + W(t)$ (i.e. $X(0) = 1$ and $\alpha = 1$). Red vertical lines are drawn to specify the updating epochs as computed numerically according to Eq. (3) for a sampling cost of $c = 1$. Notice that we should update more frequently when the process is close to zero. This is because when $X(t) \approx 0$ the delay on both paths are very similar and any small perturbation can change the identity of the shortest path. Much more insight can be gained from Figure 3, but for brevity and to avoid repetition we'll postpone this discussion to Section III-B.

B. Ornstein-Uhlenbeck Process

The Wiener process, $W(t)$, discussed in Section III-A lacks stationarity and rapidly wanders to infinity. The absence of stationarity makes it impossible to reason about long-term average behavior of our algorithm. To address this deficiency, we will leverage the heavy-traffic results of Halfin and Whitt [7], which show that when service times are exponentially distributed, the sequence of appropriately normalized queue lengths will converge to the Ornstein-Uhlenbeck (OU) process. It is well known that the OU process is the only non-trivial process that is simultaneously Gaussian, Markov, and stationary; all of which are ideal for our purposes. Ornstein-Uhlenbeck process is the continuous time analogue of the discrete time *auto-regressive* $AR(1)$ process and satisfies the following stochastic differential equation:

$$dX(t) = \theta(\mu - X(t)) dt + \sigma dW(t)$$

where $\sigma > 0$ is the standard deviation of the process and $W(t)$ denotes a standard Wiener process. Parameter μ is the long-term mean of the process, and $\theta > 0$ signifies the mean-reversion speed. Parameter θ may seem obscure at first glance, and the reader may wonder which network characteristic is captured by this parameter. To answer this question, note that

many mechanisms are employed to steer the network towards a desirable stable state. As an example, consider the role of congestion control in TCP, which regulates the rate of packet transmission to achieve a high level of link utilization while maintaining fairness and low delay. The magnitude of θ roughly captures the strength and speed of such mechanisms and the behavior of exogenous traffic. We should note that linear combination of independent OU processes with the same θ results in another OU process with the same θ , while variance and long-term mean parameters would be added in a linear fashion.

Let us suppose that weights of paths P_1 and P_2 can be approximated by independent OU processes with the same θ ; then $X(t) = X_2(t) - X_1(t)$ is another OU process. A full treatment of the general OU process is possible but due to length limitations and to simplify our notation we shall focus on the case where the difference process, $X(t)$, has a long-term mean of $\mu = 0$. This corresponds to the case where the long-term mean delays of both paths are similar/equal. Without loss of generality suppose $X(0) = x > 0$ and assume that the OD pair use P_1 as the optimal route until the first update epoch at T . Then the solution to the aforementioned stochastic differential equation can be written recursively as

$$X(t) = xe^{-\theta t} + \sigma \int_0^t e^{-\theta(t-s)} dW_s$$

which can be represented (conditioned on $X(0) = x$) via a time-scaled Wiener process as

$$X(t) = xe^{-\theta t} + \frac{\sigma e^{-\theta t}}{\sqrt{2\theta}} W(e^{2\theta t} - 1)$$

As a result, the cost of error associated with mis-identifying the shortest path during this period can be computed as

$$\begin{aligned} C_T(x) &= -\int_0^{t_1} X^-(t) dt = -\int_0^T \min\{X(t), 0\} dt \\ &= \frac{x(e^{-\theta T} - 1)}{\theta} \\ &\quad - \int_0^T \min\left\{\frac{\sigma e^{-\theta t}}{\sqrt{2\theta}} W(e^{2\theta t} - 1), -xe^{-\theta t}\right\} dt \end{aligned}$$

and

$$\begin{aligned} &- \int_0^T \min\left\{\frac{\sigma e^{-\theta t}}{\sqrt{2\theta}} W(e^{2\theta t} - 1), -xe^{-\theta t}\right\} dt \\ &= \frac{-\sigma}{(2\theta)^{\frac{3}{2}}} \int_0^{e^{2\theta T} - 1} \frac{\min\left\{W(y), \frac{-x\sqrt{2\theta}}{\sigma}\right\}}{(1+y)^{\frac{3}{2}}} dy \quad (4) \end{aligned}$$

$$= \frac{-\sigma}{(2\theta)^{\frac{3}{2}}} \int_0^{e^{2\theta T} - 1} \frac{\min\{W(y), h\}}{(1+y)^{\frac{3}{2}}} dy \quad (5)$$

Equation (4) is the result of a simple change of variable for $y = e^{2\theta t} - 1$, and (5) simplifies the notation by defining $h = -x\sqrt{2\theta}/\sigma$. Hence,

$$\begin{aligned} \mathbb{E}[C_T(x)] &= \frac{x(e^{-\theta T} - 1)}{\theta} \\ &\quad - \frac{\sigma}{(2\theta)^{\frac{3}{2}}} \int_0^{e^{2\theta T} - 1} \frac{\mathbb{E}[\min\{W(y), h\}]}{(1+y)^{\frac{3}{2}}} dy \end{aligned}$$

Using the same process used in Section III-A to compute the expectation, we obtain

$$\mathbb{E}[\min\{W(y), h\}] = -\frac{\sqrt{y}}{\sqrt{2\pi}} e^{\left(\frac{-h^2}{2y}\right)} + \frac{h}{2} \operatorname{erfc}\left(\frac{h}{\sqrt{2y}}\right)$$

which simplifies the expected cost of error to

$$\begin{aligned} \mathbb{E}[C_T(x)] &= \frac{x(e^{-\theta T} - 1)}{\theta} \\ &\quad + \frac{\sigma}{4\theta\sqrt{\theta\pi}} \int_0^{e^{2\theta T} - 1} \frac{\sqrt{y}}{(1+y)^{\frac{3}{2}}} \exp\left(-\frac{\theta x^2}{y\sigma^2}\right) dy \\ &\quad + \frac{x}{4\theta} \int_0^{e^{2\theta T} - 1} \frac{1}{(1+y)^{\frac{3}{2}}} \operatorname{erfc}\left(-\frac{\sqrt{\theta}x}{\sqrt{y}\sigma}\right) dy \end{aligned}$$

It comes as no surprise that $\mathbb{E}[C_T(x)]$ is the central quantity that dictates the behavior of the system, yet it presents a formidable challenge to intuition. Fortunately, $\mathbb{E}[C_T(x)]$ lends itself to a piecewise linear approximation that sheds light on its behavior. Let us start by evaluating $\mathbb{E}[C_T(x)]$ when $x = 0$,

$$\mathbb{E}[C_T(0)] = \frac{\sigma \left(\ln(e^{\theta T} + \sqrt{e^{2\theta T} - 1}) - \sqrt{1 - e^{-2\theta T}} \right)}{2\theta\sqrt{\theta\pi}}$$

despite the relative simplicity of this expression, we cannot find an analytic solution to the following objective function,

$$T_1^*(0) = \operatorname{argmin}_{T>0} \frac{c + \mathbb{E}[C_T(0)]}{T}$$

to obtain the first order optimality condition, let us differentiate the aforementioned expression and expand it as a Taylor series,

$$\frac{\partial}{\partial T} \left(\frac{c + \mathbb{E}[C_T(0)]}{T} \right) = -\frac{c}{T^2} + \frac{\sigma}{3\sqrt{2\pi}\sqrt{T}} + \sqrt{\mathcal{O}(T)}$$

setting the sum of the first two terms equal to zero gives us,

$$T_1^*(0) = \left(\frac{18\pi c^2}{\sigma^2} \right)^{\frac{1}{3}}$$

incidentally, this expression is identical to Eq. (2) which captured the shortest sampling period of a Wiener process. This should not be surprising because at $x = 0$, the OU process is not experiencing any mean reversion and is effectively indistinguishable from a Wiener process.

The aforementioned result can be used to create a piecewise linear approximation to $\mathbb{E}[C_T(0)]$. We can avoid the need for an excessive number of linear segments by noting that we are only interested in the behavior of the function near $T_1^*(0)$, and thus the simplest such approximation can be stated as

$$\mathbb{E}[C_T(0)] \approx \begin{cases} 0 & \text{for } T \in [0, T^*] \\ m(T - T^*) & \text{for } T > T^* \end{cases}$$

where $m = \left. \frac{\partial \mathbb{E}[C_T(0)]}{\partial T} \right|_{T=T^*} = \frac{\sigma\sqrt{1-e^{-2\theta T^*}}}{2\sqrt{\theta\pi}}$ and $T^* = T_1^*(0)$.

To generalize our results to non-zero values of x , note that

$$\mathbb{E}[C_{T^*}(x)] \approx \mathbb{E}[C_{T^*}(0)] + x \left(\left. \frac{\partial \mathbb{E}[C_{T^*}(x)]}{\partial x} \right|_{x=0} \right)$$

where

$$\left. \frac{\partial \mathbb{E}[C_{T^*}(x)]}{\partial x} \right|_{x=0} = \frac{e^{-\theta T^*} - 1}{2\theta}$$

Note that $\mathbb{E}[C_T(x)]$ is a non-negative quantity, and is decreasing in x . As a result, we only need to consider the effects of x on the non-zero portion of the approximation, and find its corresponding interception point with the horizontal axis. Putting it all together we have:

$$\mathbb{E}[C_T(x)] \approx \begin{cases} 0 & \text{for } T \in [0, f(x)] \\ m(T - f(x)) & \text{for } T > f(x) \end{cases}$$

$$\text{where } f(x) = T^* + \frac{1 - e^{-\theta T^*}}{\sqrt{1 - e^{-2\theta T^*}}} \frac{\sqrt{\pi}}{\sigma\theta} x$$

This approximation shows that the initial routing decision will remain correct for approximately $f(x)$ seconds; consequently the expected cost of error during this period is approximately 0. As we pass the threshold of $f(x)$ seconds, it becomes likely that the originally selected path is no longer optimal and routing erroneously through it will incur a cost of m units per second. Continuing with our approximation,

$$\frac{c + \mathbb{E}[C_T(x)]}{T} \approx \begin{cases} \frac{c}{T} & \text{for } T \in [0, f(x)] \\ m + \frac{c - mf(x)}{T} & \text{for } T > f(x) \end{cases}$$

The approximate cost rate function shows that if the update epoch T occurs at or before $f(x)$, we will only incur the updating cost c , amounting to a cost rate of c/T . However, if the updating epoch occurs after $f(x)$, then the cost includes the updating cost c , as well as the cost of error which accrues at a rate of m units per second. Mathematically speaking, when $T \in [0, f(x)]$, the objective function is decreasing with T and reaches its minimum at $T = f(x)$. On the other hand, when $T > f(x)$, an increase or decrease in the objective function depends on the sign of $c - mf(x)$. If this expression is greater than zero, it will reach its infimum at infinity; otherwise the minimum will occur at $T = f(x)$. Putting it all together we have

$$\operatorname{argmin}_{T>0} \frac{c + \mathbb{E}[C_T(x)]}{T} \approx \begin{cases} f(x) & \text{if } c < mf(x) \\ \infty & \text{else} \end{cases}$$

This gives us a clear description of the approximate algorithm: if the cost of updating routing information is small enough, i.e., if $c < mf(x)$, then we should update the routing tables by sampling the process at $T = f(x)$. Otherwise, the cost of updating is too large and we should continue our previous routing decisions without any new samples.

As a result, if we constrain ourselves to cases where the updating cost $c < mf(x)$, we get a simple expression for the time until the next updating epoch, T , as a function of the last observed value of the process, x , namely

$$T_1^*(x) = T^* + \frac{1 - e^{-\theta T^*}}{\sqrt{1 - e^{-2\theta T^*}}} \frac{\sqrt{\pi}}{\sigma\theta} x \quad (6)$$

Figure 4 depicts a sample path of an OU process with $\sigma = 0.5$, $\theta = 0.025$ and an initial value of $X(0) = 1$. Red vertical

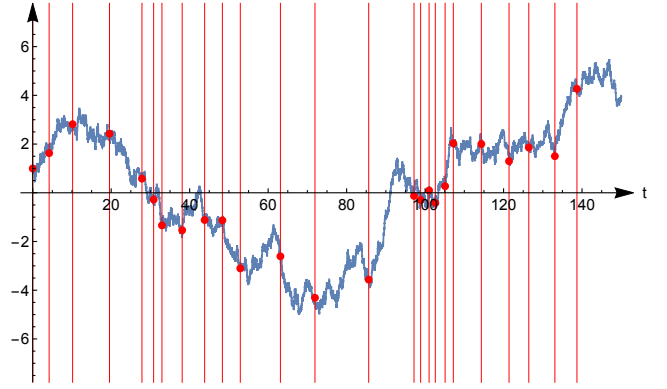


Figure 4: Sample path of an OU process with $\sigma = 0.5$, $\theta = 0.025$, $x_0 = 1$, $c = 0.1$.

lines are drawn to specify the updating epochs as suggested by our algorithm and the red dots denote the sampled values of the function. Notice that when the process is far from zero, indicating that the delay of one of the paths is significantly less than the other, we do not need to sample their weights frequently. This matches our intuition because in this scenario it is unlikely for the shorter/better path to get worse than the second path in a short period of time. On the other hand, when the process is close to zero, indicating that the weight of both paths are very similar, we require more frequent samples. The varying frequency of updates is revealed via the changing density of the vertical red lines in the figure.

We should additionally note that the shortest updating period occurs when the two paths have identical weights, i.e., $x = 0$. This corresponds to an updating period of $T_1^*(0) = (18\pi c^2 / \sigma^2)^{1/3}$. Comparing our adaptive updating method to a uniform one that samples at this rate, we see a gain of

$$G = \frac{\mathbb{E}[T_1^*(x)]}{T_1^*(0)} = \frac{\int_0^\infty T_1^*(x) f_{|X(t)|}(x) dx}{T_1^*(0)}$$

where $f_{|X(t)|}(x)$ denotes the pdf of the (one-sided) OU process. Recalling that the OU process is Gaussian we have

$$f_{|X(t)|}(x) = 2\sqrt{\frac{\theta}{\pi\sigma^2}} \exp\left(-\frac{\theta x^2}{\sigma^2}\right) \quad \text{for } x \geq 0$$

After some algebra we get

$$\begin{aligned} G &= 1 + \frac{1}{T^*} \frac{1 - e^{-\theta T^*}}{\sqrt{1 - e^{-2\theta T^*}}} \frac{1}{\theta\sqrt{\theta}} \\ &\approx 1 + \frac{1}{\theta} \left(\frac{\sigma}{c}\right)^{1/3} \left(\frac{1}{144\pi}\right)^{1/6} \end{aligned} \quad (7)$$

where the approximation is accurate when θT^* is small.

Note that the gain is inversely proportional to θ , and as such it improves unboundedly as θ goes to zero. This is due to the fact that θ represents the strength of mean-reversion for this process (and is inversely proportional to the coherence time of the process). As a result, when θ is small the process is not strongly attracted to its long-term mean, which significantly reduces the chances of crossing the horizontal

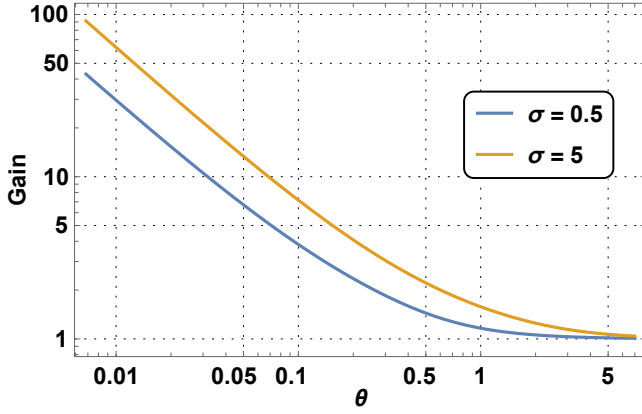


Figure 5: Gain of our adaptive sampling strategy ($c = 1$).

axis. For processes with small θ , our adaptive algorithm will automatically adopt a lower sampling rate resulting in a significant reduction in sampling and updating costs. This makes sense since processes with long coherence time are very predictable and do not require much sampling. The gain will also increase, though at a lower rate, when σ increases, which amplifies the wandering behavior of the process. Figure 5 depicts the gain as computed by Eq. (7).

IV. GENERALIZATIONS TO LARGE NETWORKS

Our analysis has so far focused on a single OD pair with two independent paths. In this section we will show that the basic operation can be extended to general networks. Recall that the shortest path can be identified through a sequence of pairwise comparisons of available paths. Furthermore, the length of each path can be updated at a time determined by our algorithm. To see the operation of such a system, let us augment the routing table of I with an additional column to track the variance of delay on each path, as shown in Table II.

Table II: Partial routing table maintained by node A

Destination	Path	Length	Variance
Y	$B \rightarrow C \rightarrow D \rightarrow Y$	2	4
	$E \rightarrow F \rightarrow G \rightarrow Y$	3	5
	$H \rightarrow I \rightarrow Y$	8	1
Z	$J \rightarrow K \rightarrow Z$	4	2
	$L \rightarrow M \rightarrow Z$	7	4
	$N \rightarrow O \rightarrow P \rightarrow Z$	9	7

Given the routing table in Table II, node A can query nodes B and E , which are the first hops on the two shortest paths to node Y , about their respective distances to node Y according to our sampling formula

$$T_1(x) = T^* + \frac{1 - e^{-\theta T^*}}{\sqrt{1 - e^{-2\theta T^*}}} \frac{\sqrt{\pi}}{\sigma \theta} x, \quad T^* = \left(\frac{18\pi c^2}{\sigma^2} \right)^{\frac{1}{3}}$$

where $x = 3 - 2 = 1$, $\sigma = \sqrt{4 + 5} = 3$, and θ is a global parameter that depends on network protocols and network size. The same procedure can be used to compute the updating time for all other paths. When a link is shared between multiple OD

pairs, the algorithm simply picks the smallest sampling time from those computed for all OD pairs.

Not only is the proposed algorithm simple to understand and implement, it also addresses an often-overlooked byproduct of traditional routing protocols. It was shown in [9] that routing updates can inadvertently become synchronized, causing instability as well as untimely and unmanageable bursts of traffic. To address such issues, a whole host of ad-hoc randomization procedures have been incorporated into commercial routers. In contrast, our algorithm requires each node to independently measure the variance of delay on each of its outgoing links. Given the unique geographical location of each node within the network and expected difference in their measurements, it is unlikely for the routing updates to synchronize, thus avoiding the need for explicit randomization procedures.

V. CONCLUSION

In this paper, we introduced a new algorithm that allows us to capture the tradeoff between monitoring and optimal operation of a network. We introduced the concept of sampling/updating cost to capture the cost associated with the collection and dissemination of routing information within a network. We further studied two stochastic models of delay, namely the Wiener process and the Ornstein-Uhlenbeck process and showed that we can dynamically adjust the sampling times of each link based on their instantaneous significance to network management. The gain (as reduction in number of samples) over the traditional uniform sampling was 100 in one example. We concluded our remarks by extending our notions to general networks and suggest that a network can be operated in a decentralized fashion at high performance using *significant sampling* and reporting of its network states, allowing the NMC system to be scalable. The essential spirit of the cognitive NMC is that it collects network states ONLY when they matter to the network performance.

REFERENCES

- [1] "The zettabyte era: Trends and analysis," June 2017. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html>
- [2] "Cisco vni: Forecast and methodology, 2016-2021," June 2017. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>
- [3] V. W. Chan and E. Jang, "Cognitive all-optical fiber network architecture," in *Transparent Optical Networks (ICTON), 2017 19th International Conference on*. IEEE, 2017, pp. 1-4.
- [4] D. Bertsekas and R. Gallager, *Data Networks (Second Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.
- [5] L. Kleinrock, *Queueing Systems Volume I: Theory*. New York: John Wiley & Sons, 1975, vol. 1.
- [6] J. F. C. Kingman, "The single server queue in heavy traffic," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 57, no. 4, pp. 902-904, 1961.
- [7] S. Halfin and W. Whitt, "Heavy-traffic limits for queues with many exponential servers," *Operations research*, vol. 29, no. 3, 1981.
- [8] P. Mörters and Y. Peres, *Brownian motion*. Cambridge University Press, 2010.
- [9] S. Floyd and V. Jacobson, "The synchronization of periodic routing messages," *IEEE/ACM transactions on networking*, vol. 2, no. 2, pp. 122-136, 1994.