Joint Minimization of Monitoring Cost and Delay in Overlay Networks: Optimal Policies with a Markovian Approach

Sandrine Vaton, Olivier Brun, Maxime Mouchet, Pablo Belzarena, Isabel Amigo, Balakrishna Prabhu, Thierry Chonavel

Noname manuscript No.
(will be inserted by the editor)

Joint minimization of monitoring cost and delay in overlay networks: optimal policies with a Markovian approach ⋆

Sandrine Vaton · Olivier Brun · Maxime Mouchet · Pablo Belzarena · Isabel Amigo · Balakrishna J. Prabhu · Thierry Chonavel

the date of receipt and acceptance should be inserted later

Abstract Continuous monitoring of network resources enables to make more-informed resource allocation decisions but incurs overheads. We investigate the trade-off between monitoring costs and benefits of accurate state information for a routing problem. In our approach link delays are modeled by Markov chains or hidden Markov models. The current delay information on a link can be obtained by actively monitoring this link at a fixed cost. At each time slot, the decision maker chooses to monitor a subset of links with the objective of minimizing a linear combination of long-run average delay and monitoring costs. This decision problem is modeled as a Markov Decision Process whose solution is computed numerically. In addition, in simple setttings we prove that immediate monitoring cost and delay minimization leads to a threshold policy on a filter which sums up information from past measurements. The lightweight method as well as the optimal policy are tested on several use-cases. We demonstrate on an overlay of 30 nodes of RIPE Atlas that we obtain delay values close to the performance of the always best path with an extremely low monitoring effort when delays between nodes are modeled with hierarchical Dirichlet process hidden Markov models.

Keywords active monitoring, routing overlays, Markov chains, hidden Markov models, HDP-HMM, Markov Decision Processes, sparse monitoring, Round Trip Times, RIPE Atlas

⋆ Neither the entire paper nor any part of its content has been published or has been accepted for publication elsewhere. It has not been submitted to any other journal.

S. Vaton · M. Mouchet · I. Amigo
IMT Atlantique, IRISA, UBL, Brest, France
E-mail: {sandrine.vaton, maxime.mouchet, isabel.amigo}@imt-atlantique.fr

P. Belzarena
Facultad de Ingeniería, Universidad de la República, Uruguay
E-mail: belza@fing.edu.uy

O. Brun · B.J. Prabhu
CNRS, LAAS-CNRS, Université de Toulouse, France
email{brun, balakrishna.prabhu}@laas.fr

T. Chonavel
IMT Atlantique, Lab-STICC, UBL, Brest, France
E-mail: thierry.chonavel@imt-atlantique.fr

1 Introduction

1.1 Context

Accurate and fine-grained network monitoring is essential to the overall operation of networks. Monitoring data is required to provide an accurate representation of the network state. It needs to be delivered in a timely manner in order to enable optimal resource allocation and quick decision-making in the face of adverse events. At the same time, the monitoring solution has to be able to handle a large number of resources and metrics without causing a significant impact on network resources. The increasing scale and complexity of communication networks therefore mandate autonomous and scalable monitoring schemes trading off the cost of monitoring against the potential benefits it brings for making more-informed management decisions.

This need for parsimonious yet accurate monitoring schemes is particularly true when it comes to routing overlays [1–4]. A routing overlay is formed by end-hosts, which are deployed in different spots over the Internet. The overlay nodes monitor the quality of the Internet routes between them and cooperate with each other to forward data on behalf of any pair of communicating nodes. By adding intermediate routing hops into the path taken by streams of packets, they influence the overall path, without modifying the underlying IP mechanism for computing routes. This approach has been used to create self-healing and self-optimizing routing overlays. These overlays are able to monitor the quality of Internet paths between their nodes and to adapt their routing schemes according to application-specific metrics and what is observed from the underlying network. However, most existing overlay technologies use all-pairs probing: they regularly monitor the quality of all overlay links in order to update their routing strategy. The advantage of this approach is that it guarantees optimal performance, but its main downside is that it results in a costly $O(n^2)$ probing overhead as the number of participating nodes n increases, since the topology of a routing overlay is that of a complete graph. Moreover, considering that measuring accurately the delay suffered by an application requires to inject a train of probe packets with a rate similar to that of the application [5] the cost will depend on the application and could be high at the overlay scale. Even more so, if other performance metrics were to be considered, such as bandwidth, the cost can rapidly become prohibitive.

The goal of our study is to design efficient monitoring and routing strategies to discover optimal routes with a scalable probing effort, in order to build a routing overlay that can be widely deployed over a sizable population of routers. More precisely, we seek for a method with a small monitoring effort but sufficiently accurate to enable near-optimal routing decisions. We want to demonstrate that it is enough to probe only from time to time a small number of alternate paths and thus obtain a considerable reduction in complexity of data processing and decision making. So, we look for a monitoring method that is sparse, both in time and in space. This method decides at each measurement epoch which of the paths should be monitored, if any, instead of systematically probing all paths. To start with, and because large scale measurement campaigns are publicly available for validations, we consider as path QoS metrics the round trip time (RTT) delay.

1.2 Contributions and structure of the article

As explained previously, there is a trade-off between the accuracy of delay estimates and the optimality of routing decisions. The accuracy can be improved by monitoring paths more often, but at an increased cost of monitoring. The main contribution of this article is to propose a theoretical framework for taking monitoring and routing decisions that offer a good trade-off between monitoring

costs and the gain in delay from appropriate routing decisions. This framework is based on Markov Decision Processes (MDPs) and some markovian model of paths delays.

The practical question answered in this paper is the following one. *Consider a routing overlay: at each time slot, which paths should one monitor and which path should be chosen for routing traffic, so as to minimize the delay and the monitoring effort?* As the paths performance is in practice stable over quite long periods of time it is possible to limit considerably the monitoring effort and still maintain a sufficiently accurate view of the network state to take close to optimal routing decisions (i.e. select the best path most of the time). A key element is to capture the stability of the path performance in an accurate markovian model of successive delay values on that path.

In more details here are the main contributions of the article:

- **HDP-HMM-based characterization of network delays:** Using RTT measurements from RIPE Atlas [6], we propose to model RTTs with Hidden Markov Models (HMM). More precisely we introduce the application of Hierachical Dirichlet Process HMM (HDP-HMM) as statistical models for RTT series. This model permits an accurate characterization of RTT. The number of states of the HMM is unknown a priori but calibrated from the measurement series.
- **Formulation as an MDP:** We define the cost as a linear combination of the monitoring cost and of the delay of the chosen path. We show that the problem of minimizing the discounted cumulative cost over an infinite horizon can be formulated as an MDP (Markov Decision Process). The optimal policy can be obtained by solving this MDP problem with a Value Iteration algorithm.
- **Optimal myopic policy:** We also prove a closed form solution in some simple settings when the one-step cost is minimized. We call this policy the "myopic" policy since future costs are not taken into account. Different scenarios are considered: first, a deterministic path and a stochastic path whose delay evolves randomly according to a MC or a HMM with two states, and second, two stochastic paths.
- **Monitoring cost vs. delay trade-off:** We provide extensive numerical results, including those obtained for an overlay of 30 RIPE Atlas anchors. For the latter example, we show that the MDP policy enables to reduce the monitoring load by 91% (on average) at the expense of an increase of network delay of only 0.07% (on average) with respect to an all-pair probing strategy.

The paper is organized as follows. Section 2 introduces Markov chains (MC), hidden Markov models (HMM) and HDP-HMM for statistical characterization of RTTs. We show that HDP-HMM is an accurate model of delays from the analysis of RIPE Atlas measurements. We also briefly explain how the parameters of the models can be trained from real datasets. Sections 3, 4 and 5 are devoted to the explicit characterization of the optimal myopic policy in some simple settings. We first consider the case of two different paths between a source and a destination. the myopic objective is to maximize the one-step reward (negative of the path delay and monitoring costs). In Section 4 where one of the paths is deterministic and the second one is a MC of a HMM with two states, we obtain a closed-form solution of threshold type for the optimal policy. In Section 5 we derive the myopic policy in the case of two stochastic paths in which each path can be modeled as either a MC or a HMM with any number of states. Then in Section 6, we generalize our problem to the case of several paths, each with a delay that is modeled as Markov Chain. The objective is to maximize the discounted cumulative reward, so that the benefits of monitoring now on future routing decisions is taken into account. We formulate this problem as a Markov Decision Process (MDP) whose solution is computed numerically with the value iteration algorithm. Section 7 discusses issues related to practical implementation of our approach using the Software Defined Network (SDN) paradigm. It also discusses computational complexity and scalability of the MDP approach. Some validation results are presented in Section 8 in different scenarios. We first consider some simple scenarios with two paths between one source and one destinations (and deterministic delays or two-states Markov or hidden Markov delays on each of

the paths). The next scenarios are more complex and based on delay data collected on RIPE Atlas. In the last scenario we consider an overlay of 30 RIPE Atlas anchors. The MDP approach is used and the benefits of our method in that case are highlighted. The article ends with a discussion on related works in Section 9, and a conclusion and presentation of future works in Section 10.

2 Markov modeling of delays

2.1 Analysis of RIPE Atlas measurements

In this section we analyze delay measurements from the RIPE Atlas infrastructure, and explain how RTT time series can be modeled as Hidden Markov Models.

RIPE Atlas is a global network of probes for the measurement of Internet connectivity, managed by the RIPE NCC [6, 7]. Probes are small devices deployed on various locations of the Internet, from broadband accesses in individual homes, to Internet Exchange Points. In high-availability environments, enhanced probes, called anchors, are deployed. Anchors can perform more simultaneous measurement than traditional probes, and are part of the anchoring mesh measurements system. That is, every anchor performs measurements with every other anchor, most notably pings every four minutes and traceroutes every fifteen minutes. A ping measurement consists of three ICMPv4 and ICMPv6 echo packets. As of 2017, there are 320 anchors, which means that more than 100,000 ping measurements are performed every four minutes. Combined with the global distribution of the anchors (Figure 1), this forms a comprehensive dataset of Internet delay measurements.

Because of their high-availability, and of full-mesh measurements, anchors are appropriate to study delay-based routing in global-scale overlays. In this section we'll consider every IPv6 anchoring measurements between the 6th and the 13th of November 2017, resulting in a dataset of 39903 RTT series of 2520 time slots each. In Section 8 we will consider a subset of 30 anchors for validating our approach. In that section, we shall also consider RTT measurements from a RIPE Atlas customized measurement campaign carried out on August 2016. RTT has been monitored for each pair of nodes, in this case during 7 days. The average RTT has been obtained for each time period of two minutes and for each origin-destination pair on the basis of five ICMP echo "ping" packets, resulting in 5036 values for each pair of nodes. Nodes (probes or anchors) have randomly been selected, taking into account their geographic distribution and reliability. In particular, we have chosen probes and anchors belonging to different countries and autonomous systems, and corresponding to the most recent software and hardware versions available in the measurement platform.

From the analysis of these datasets we affirm that RTT time series have a typical behavior that can be captured by a Markov chain (MC) or a Hidden Markov Model (HMM). The observed behavior is that RTT values switch among several probability distributions. Transitions from one distribution to another occur at random times, as it can be seen in Figure 2. We have noticed that, for a given Origin Destination pair, there exist few probabilistic laws according to which RTT take their values, and that the law of the RTT remains stable for some time. This behavior is in part explained by load-balancing and routing configuration changes in operator networks [8–10]. HMMs are generalization of mixture models that are used by several authors [11, 12] to characterize delays. But HMMs take into account time dependence between successive values of delays, a key property to enable sparse-in-time monitoring.

2.2 Model learning

Markov chains (MC) and Hidden Markov Models (HMM) are characterized by a set of parameters.

Let us first consider the case of a MC model. Let $L(t)$ denote the delay of a path (RTT) at time t. If we assume K possible values for $L(t)$, denoted by $l_1, l_2, \ldots, l_K$, the state space is $(l_i)_{i=1,K}$ and K is the order of the model. The model is parameterized by the transition probabilities $P_{ij} = \mathbb{P}(L(t+1) = l_j \mid L(t) = l_i)$. These parameters can be tuned from the analysis of a measurement dataset $L(t), t = 1, \ldots, T$. It turns out that the parameter values for which the likelihood of the dataset is maximum (Maximum Likelihood Estimators, MLE) are $\hat{P}_{ij} = \frac{\sum_{t=1}^{T-1} \mathbb{1}_{L(t)=l_i, L(t+1)=l_j}}{\sum_{t=1}^{T-1} \mathbb{1}_{L(t)=l_i}}$ where $\mathbb{1}$ denotes the indicator function, so, the frequency with which the MC switches to state l_j when it is in state l_i.

Now let us assume that $L(t)$ follows a HMM or, stated differently, a Markov chain with noise. The value of $L(t)$ relies on the value $S(t)$ of an underlying MC, denoted by S. Let $\{1, 2, \ldots, K\}$ denote the states of S and $(P_{ij})_{1 \leq i,j \leq K}$ its transition probabilities. $L(t)$ is a random function of $S(t)$, characterized by a probability density function $p_i(s)$, where $i = S(t)$. For example, in the Gaussian case, one can assume that if $S(t) = i$ then $L(t) \sim N(l_i, \sigma_i^2)$. It has a Gaussian distribution with mean l_i and variance σ_i^2: $p_i(l) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{1}{2\sigma_i^2}(l - l_i)^2)$. The parameters of the HMM are the transition probabilities

Fig. 1: Location of RIPE Atlas anchors.

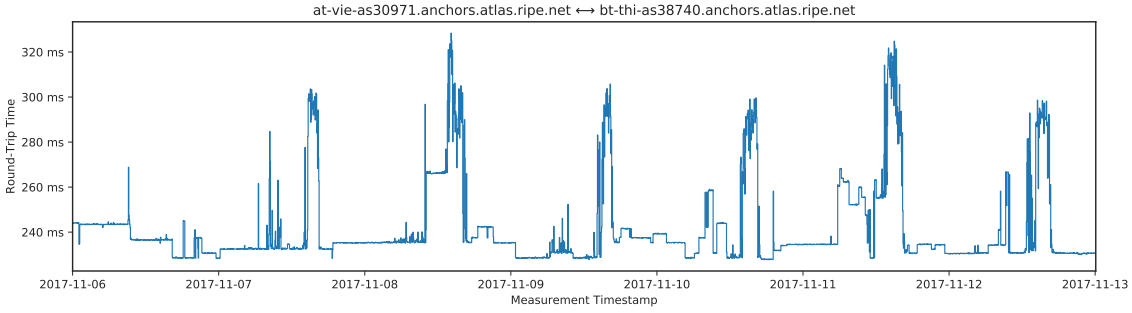

Fig. 2: Minimum IPv6 delay measured between two RIPE Atlas anchors over one week.

P_{ij} and the parameters of the conditional distributions of $L(t)$ knowing that $S(t) = i$, for example $(l_i)_{i=1,\dots,K}$ and $(\sigma_i^2)_{i=1,\dots,K}$ in the Gaussian case.

Tuning the parameters of a HMM is not straightforward since the values of $S(t)$ are not observed directly. In fact, this is a parameter estimation problem with unknown (also called missing) data. A classical approach to solve such a problem is the Expectation Maximization (EM) algorithm [13]. This algorithm makes it possible to perform ML estimation of the parameters of a statistical model when there are missing data. However, for RTT measurements, the number K of states is not known in advance and a standard HMM model trained with an EM algorithm cannot be applied directly to characterize them.

So far, several techniques have been proposed to cope with unknown HMM order. Recently efficient generic methods have been developed to estimate HMM parameters. These methods offer flexibility to capture the variability observed in different kinds of real datasets. This kind of methods, known as Hierarchical Dirichlet Process HMM (or, HDP-HMM [14]) builds on Bayesian inference. The number of states of the HMM as well as the parameters of the probability distribution of observations in each state of the Markov chain are considered as random variables.

In a Bayesian setting, a prior distribution is assumed for the parameters. The distribution of parameters conditioned on the observations (posterior likelihood) is computed to estimate the parameters, typically by looking at posterior modes or means. In HDP-HMM, Dirichlet Processes (DP) are used as priors on the transition matrix rows, which makes it possible to specify that the number of states is unknown. To ensure that these transition probabilities weight the same emission distributions, these DPs are parameterized by the same base distribution. This base distribution itself is modeled by a DP prior, hence the Hierarchical Dirichlet Process (HDP) structure of this modelling. This hierarchy of random dependences and vague priors introduce enough flexibility in the model to let it adapt to many different time series, such as RTT measurements.

In practice, the exact computation of the posterior distribution of parameters is intractable. However, it is possible to draw samples from this posterior distribution. This can be done by using MCMC (Markov Chain Monte Carlo) methods, that are classical stochastic simulation methods for Bayesian inference [15], and in particular Gibbs sampling.

A graphical representation of the HDP-HMM is given in Figure 3, where the arrows represent the dependencies. The HMM itself is represented by states and observations $\{(S(k), L(k))\}_{k\geq 1}$. Its parameters are $\{(\theta_k, \pi_k)\}_{k\geq 1}$: θ_k represents the parameters of the observation process in state k and π_k represents probability transitions vector from state k. α, γ, λ are the hyper-parameters. γ and λ are the parameters of the Dirichlet process that lies at the top of the HDP hierarchy: this Dirichlet process can be written in the form of a random distribution $G_0 = \sum_{k>0} \beta_k \delta_{\theta_k}$. The process $\beta = (\beta_k)_{k\geq 1}$ is a stick-breaking process: the β_ks can be seen as the lengths of the pieces of a unit length stick, the remaining part of which is broken infinitely many times [16]. Each π_k is modelled by a Dirichlet process that can be described via parameters α and β. See [14] for more details.

Note that in the literature Dirichlet Processes have already been considered to model RTT series: in [12], Dirichlet processes are used to estimate distributions of daily RTT series using a mixture model with unknown order. Then, dependence among successive daily mixtures is investigated and a heuristic approach is proposed. As it can be seen in Figure 2, there exists a strong dependence among successive RTT measurements and more accurate modelling can be achieved by accounting for this dependency.

We fitted the proposed HDP-HMM models using the pyhssm library [17]. Then, it was possible to estimate the underlying states of data sequences via a Viterbi algorithm. An example of the resulting data clustering is shown in Figure 4.

Since parameter estimation for HMM is not the main goal of this article, we shall not go into further details of this topic here. In what follows, we shall focus on the problem of optimal monitoring. The

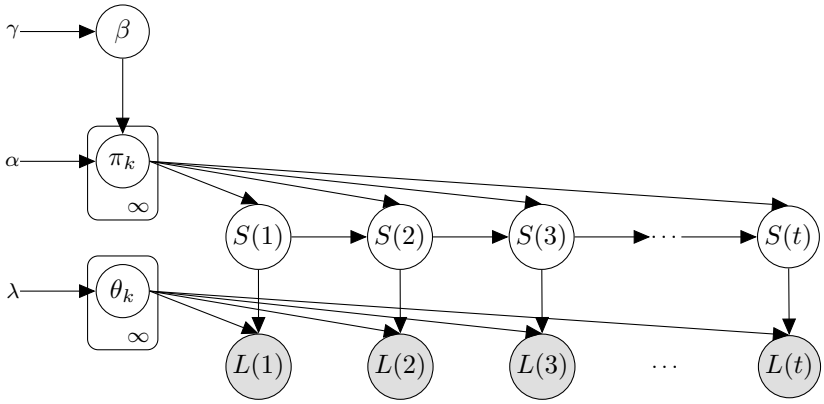

Fig. 3: The Hierarchical Dirichlet Process - Hidden Markov Model.

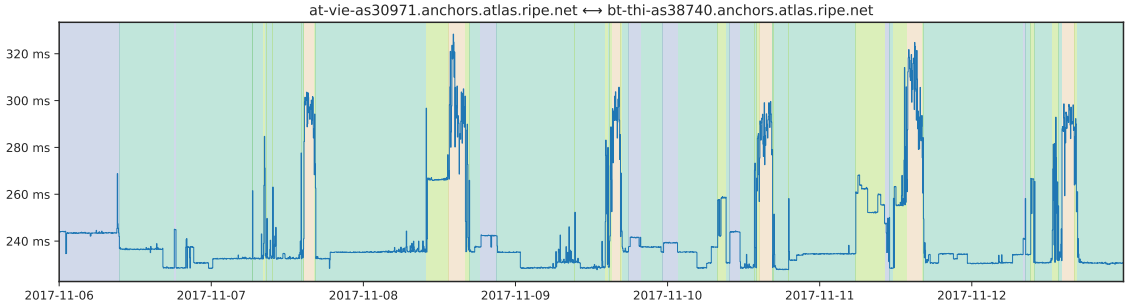

Fig. 4: States learned using HDP-HMM on a delay measurement between two RIPE Atlas anchors. Identical colors represent identical states.

goal is to reduce the frequency of measurements but be able to select the path in order and get a performance close to the delay of the fastest path. We will assume that an appropriate Markov or HMM model has been trained for the different paths between origin nodes and destination nodes. Our approach will be validated in Section 8 where we consider delay measurements in an overlay of 30 RIPE Atlas anchors.

3 Routing optimization

To start with we will consider some simple settings in order to obtain closed form solutions for the optimal monitoring policy. So, we first consider, in Sections 3, 4 and 5, the case of two paths. In Section 6 we introduce Markov Decision Processes (MDPs) to solve problems with more than two paths and any number of states per path and also to take into account the benefits of monitoring on future routing performance.

Let us now consider the case of two paths. These two paths have the same Origin node and the same Destination node but they have different delays. One of the paths has a deterministic delay l

whereas the other path has a random delay. To be more precise it is assumed that the delay of the random path is a Markov chain or a Hidden Markov Model.

At each time step two decisions must be taken: first of all the monitoring decision, and then the routing decision. At each time step one first decides to measure the random path or not, which is represented as $M(t) = 0$ (no measure at time t) or $M(t) = 1$ (measure). Then, based on past information, one decides which path to select: $C(t) = 1$ means that the random path is chosen whereas $C(t) = 0$ means that we opt for the deterministic path.

In what follows I_t represents the information available at time t (after $L(t)$ has possibly been measured). So, $I_t = \sigma\{L(u) \text{ s.t. } M(u) = 1; u \leq t\}$ where σ stands for sigma-algebra. It takes into account all past measurements, including the measure at time t if $M(t) = 1$.

3.1 Filtering and forecasting

3.1.1 Case of a Markov chain model

Assume that $L(t)$ follows a MC model. Because of the Markov property the information in I_t which is useful to forecast the future values of L is summed up by the filter $\gamma_{t,t}(i) = \mathbb{P}(L(t) = l_i \mid I_t)$. The first index t refers to I_t, that is to say that we assume information is available up to time t. The second index t refers to $L(t)$, that is to say that we infer the value of the random path delay at time t.

Obviously, if L has been monitored at time t then the value of $L(t)$ is known accurately and $\gamma_{t,t}(i)$ is 1 for the corresponding i value and 0 for the others. On the contrary, if $L(t)$ has not been monitored then the belief about its value can be computed from the belief about $L(t-1)$ taking into account the Markov dynamics. To sum up, it comes that:

$$\gamma_{t,t} = \mathbb{1}_{M(t)=1} \sum_i \mathbb{1}_{L(t)=l_i} e_i + \mathbb{1}_{M(t)=0} \gamma_{t-1,t-1} P \tag{1}$$

where e_i is an all-zero row vector except in position i which is equal to 1 and where $\gamma_{t-1,t-1}$ and $\gamma_{t,t}$ are row vectors. Equivalently, $\gamma_{t,t}(i) = \mathbb{1}_{M(t)=1}\mathbb{1}_{L(t)=l_i} + \mathbb{1}_{M(t)=0} \sum_j \gamma_{t-1,t-1}(j) P_{ji}$. Future values can be forecasted based on information available at time t. To do so, the predictor $\gamma_{t,t+u}(i) = \mathbb{P}(L(t+u) = l_i \mid I_t)$ is introduced. It can be computed as $\gamma_{t,t+u} = \gamma_{t,t} P^u$. In particular, the one-step predictor is $\gamma_{t,t+1} = \gamma_{t,t} P$ or equivalently, $\gamma_{t,t+1}(i) = \sum_j \gamma_{t,t}(j) P_{ji}$.

3.1.2 Case of a Hidden Markov Model

Let us consider now the HMM model. In that case the filter is defined as $\gamma_{t,t}(i) = \mathbb{P}(S(t) = i \mid I_t)$ and the predictor as $\gamma_{t,t+u}(i) = \mathbb{P}(S(t+u) = i \mid I_t)$. These quantities can be updated as follows.

First if $M(t) = 0$ then no new measurement is available and $\gamma_{t,t} = \gamma_{t-1,t-1} P$. Similarly, it is always true that $\gamma_{t,t+u} = \gamma_{t,t} P^u$. Now, if $M(t) = 1$ then knowing the exact value of $L(t)$ brings new information about $S(t)$. In that case,

$$\begin{aligned}
\gamma_{t,t}(i) &= \mathbb{P}(S(t) = i \mid I_{t-1}, L(t)) \\
&\propto p(S(t) = i, L(t) \mid I_{t-1}) \\
&= p(L(t) \mid S(t) = i) \mathbb{P}(S(t) = i \mid I_{t-1}) \\
&= p_i(L(t)) \gamma_{t-1,t}(i) \\
&= p_i(L(t)) \sum_j \gamma_{t-1,t-1}(j)\, P_{ji}.
\end{aligned}$$

In the equation above, $\propto$ means "proportional" and the proportionality factor can be computed by normalization: $\sum_i \gamma_{t,t}(i) = 1$.

So, if one considers both situations, $M(t) = 0$ and 1, it comes that

$$\gamma_{t,t} = \mathbb{1}_{M(t)=0}\gamma_{t-1,t-1}P + \mathbb{1}_{M(t)=1}\frac{\gamma_{t-1,t-1}P\text{diag}(p_i(L(t)),i=1,\ldots,K)}{\gamma_{t-1,t-1}P\text{diag}(p_i(L(t)),i=1,\ldots,K)e^T}, \tag{2}$$

where $e = (1, 1, \ldots, 1)$ is the all-one row vector and $\text{diag}(p_i(L(t)), i = 1, \ldots, K)$ is a diagonal matrix with elements $p_i(L(t))$ on the diagonal.

3.2 Path selection

Let us now consider the choice of a rational agent that has to select, at time t, either the path of deterministic delay l or the path of random delay $L(t)$. The decision is taken on the basis of the information I_t, that is to say all measurements available up to time t (including $L(t)$ if $M(t) = 1$).

We assume that the agent opts for the random path if its expected delay is lower than l, otherwise she prefers the deterministic path. The expected delay of $L(t)$ given I_t is either the observed value of $L(t)$ if $M(t) = 1$ or $\sum_i l_i\gamma_{t,t}(i)$ if $M(t) = 0$. Note that in the last equation l_i is one of the states of the Markov Chain in the MC model, or the average value $\mathbb{E}(L(t) \mid S(t) = i)$ of the conditional distribution in the HMM model.

So, if $C(t) = 1$ means that the path of random delay is selected whereas $C(t) = 0$ means that the path of deterministic delay is prefered, it comes that:

$$C(t) = \mathbb{1}_{M(t)=1}\mathbb{1}_{L(t)<l} + \mathbb{1}_{M(t)=0}\mathbb{1}_{\sum_i l_i\gamma_{t,t}(i)<l}. \tag{3}$$

4 QoS monitoring optimization: a first simple model

4.1 Reward

As stated previously our goal is to develop a theoretical framework to optimize monitoring decisions. One has to take into account the benefits of routing optimization and the cost of monitoring. We can consider that a fixed cost c has to be paid each time $M(t) = 1$. This represents that frequent measurements are not encouraged since, for example, active probing may load the network.

On the other hand the benefits of monitoring can be measured in terms of delay since a more accurate prediction of the path delays permits a better routing decision. We take as a reference value the delay l of the deterministic path and evaluate the gain as the difference between l and the delay of the selected path: $G(t) = \mathbb{1}_{C(t)=1}(l - L(t))$. This can be null if $C(t) = 0$ (deterministic path chosen), or positive or negative if the random path is chosen, depending on whether its delay is indeed smaller or greater than l. If the cost of monitoring is taken into account the penalized gain is then:

$$\tilde{G}(t) = \mathbb{1}_{C(t)=1}(l - L(t)) - c\mathbb{1}_{M(t)=1} \tag{4}$$

When the monitoring decision has to be taken the available information is $I_{t-1} = \sigma\{L(u) \text{ s.t. } M(u) = 1; u \leq t - 1\}$. We thus compare $\mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t) = 1)$ and $\mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t) = 0)$ and opt for the choice $M(t)$ which provides the greatest reward. Note that conditioning by I_{t-1} and $M(t) = 0$ or 1 means that we assume what the monitoring decision is but we do not take into account the monitoring result $L(t)$, since this value is unknown when the monitoring decision is taken.

$M(t)=0$	$\mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t)=0) = \mathbb{1}_{\overline{L}(t)<l}(l-\overline{L}(t))$, where $\overline{L}(t) = \sum_i \gamma_{t-1,t}(i) l_i$
$M(t)=1$	**MC model:** $\mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t)=1)$ $= l \sum_{i/l_i<l} \gamma_{t-1,t}(i) - \sum_{i/l_i<l} l_i \gamma_{t-1,t}(i) - c$ **HMM model:** $\mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t)=1)$ $= l \sum_i \gamma_{t-1,t}(i) F_i(l) - \sum_i \gamma_{t-1,t}(i) \int_0^l u p_i(u) du - c$ **Gaussian HMM:** $\mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t)=1)$ $= l \sum_i \gamma_{t-1,t}(i) \Phi(\frac{l-l_i}{\sigma_i})$ $- \sum_i \gamma_{t-1,t}(i) (l_i \Phi(\frac{l-l_i}{\sigma_i}) - \sigma_i \phi(\frac{l-l_i}{\sigma_i})) - c$

Table 1: Reward for $M(t)=0$ and $M(t)=1$. MC, HMM and Gaussian HMM models.

Case $M(t)=1$***:*** We opt for the non deterministic road if $L(t)<l$. So,

$$\mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t)=1) = l\,\mathbb{P}(L(t)<l \mid I_{t-1}) - \mathbb{E}(L(t)\mathbb{1}_{L(t)<l} \mid I_{t-1}) - c. \tag{5}$$

In the MC model $\mathbb{P}(L(t)<l \mid I_{t-1}) = \sum_{i/l_i<l} \gamma_{t-1,t}(i)$ and $\mathbb{E}(L(t)\mathbb{1}_{L(t)<l} \mid I_{t-1}) = \sum_{i/l_i<l} l_i \gamma_{t-1,t}(i)$. In the HMM model $\mathbb{P}(L(t)<l \mid I_{t-1}) = \sum_i \gamma_{t-1,t}(i) F_i(l)$ and $\mathbb{E}(L(t)\mathbb{1}_{L(t)<l} \mid I_{t-1}) = \sum_i \gamma_{t-1,t}(i) \int_0^l u p_i(u) du$, where $F_i(x) = \int_0^x p_i(u)du$ is the cumulative distribution function of the probability distribution of $[L(t) \mid S(t)=i]$ and $p_i(u)$ is the probability density function of $[L(t) \mid S(t)=i]$. In particular, in the Gaussian case $[L(t) \mid S(t)=i] \sim N(l_i, \sigma_i^2)$ it comes out that $F_i(l) = \Phi(\frac{l-l_i}{\sigma_i})$ and $\mathbb{E}(L(t)\mathbb{1}_{L(t)<l} \mid S(t)=i) = \int_0^l u p_i(u)du = l_i \Phi(\frac{l-l_i}{\sigma_i}) - \sigma_i \phi(\frac{l-l_i}{\sigma_i})$ where $\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$ and $\Phi(x) = \int_{-\infty}^x \phi(u)du$ are the p.d.f and c.d.f. of $N(0,1)$.

Case $M(t)=0$***:*** Let $\overline{L}(t) = \mathbb{E}(L(t) \mid I_{t-1}) = \sum_i \gamma_{t-1,t}(i) l_i$. We opt for the non deterministic road if $\overline{L}(t)<l$. We can note that this choice is not random with respect to I_{t-1}. If we know that $M(t)=0$ then the routing decision at time t, $C(t)$, is already known at time $t-1$: $C(t) = \mathbb{1}_{\overline{L}(t)<l}$. And the reward is $\mathbb{1}_{C(t)=1}(l-\overline{L}(t))$, that is to say:

$$R = \mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t)=0) = \mathbb{1}_{\overline{L}(t)<l}(l-\overline{L}(t)). \tag{6}$$

These results are summarized in Table 1.

4.2 Monitoring decision optimization

The rational agent takes at time t the monitoring decision with maximum reward, that is to say $M^*(t) = \mathrm{Arg}\max_{M\in\{0,1\}} \mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M)$. One should remark that the decision depends on the one-step predictor $\gamma_{t-1,t}$ only.

In what follows we are going to consider the particular case of a MC or HMM model with $K=2$ states and show that we arrive to a simple threshold policy. To do so we observe that because $\gamma_{t-1,t}(0) + \gamma_{t-1,t}(1) = 1$ the reward can be stated as a function of $x = \gamma_{t-1,t}(0)$ only.

MC model with 2 states: in the MC model we can assume without loss of generality that $l_0 < l < l_1$. Otherwise the optimal routing decision would always be to opt for the deterministic path ($l < l_0, l_1$) or for the random path ($l_0, l_1 < l$) and there would be no need to monitor. Using the definition $x = \gamma_{t-1,t}(0)$ and $1 - x = \gamma_{t-1,t}(1)$ the reward can be expressed as:

$$\begin{array}{l} \mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t) = 0) = ((l_1 - l_0)x - (l_1 - l))\mathbb{1}_{(l_1-l_0)x-(l_1-l)>0} \\ \mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t) = 1) = (l - l_0)x - c \end{array} \tag{7}$$

and the difference between rewards with and without monitoring is $\Delta = \mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t) = 1) - \mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M(t) = 0) = ((l - l_0)x - c) - ((l_1 - l_0)x - (l_1 - l))\mathbb{1}_{(l_1-l_0)x-(l_1-l)>0}$.

Let us first consider the case $x > \frac{l_1-l}{l_1-l_0}$. In that case, $\Delta = (l_1 - l)(1 - x) - c$ so that $\Delta > 0$ if $x < 1 - \frac{c}{l_1-l}$. On the contrary if $x < \frac{l_1-l}{l_1-l_0}$ then $\Delta = (l-l_0)x - c$ so that $\Delta > 0$ if $x > \frac{c}{l-l_0}$. The optimal monitoring decision is therefore:

$$\begin{array}{rl} x > \max\{\frac{l_1-l}{l_1-l_0}, 1 - \frac{c}{l_1-l}\} & \Rightarrow M^*(t) = 0 \\ \frac{l_1-l}{l_1-l_0} < x < 1 - \frac{c}{l_1-l} & \Rightarrow M^*(t) = 1 \\ \frac{c}{l-l_0} < x < \frac{l_1-l}{l_1-l_0} & \Rightarrow M^*(t) = 1 \\ x < \min\{\frac{l_1-l}{l_1-l_0}, \frac{c}{l-l_0}\} & \Rightarrow M^*(t) = 0 \end{array} \tag{8}$$

Let us now check if the two intervals in the middle in (8) exist. This is true if $\frac{l_1-l}{l_1-l_0} < 1 - \frac{c}{l_1-l}$ and if $\frac{c}{l-l_0} < \frac{l_1-l}{l_1-l_0}$. Both conditions are in fact equivalent to $c < \frac{(l-l_0)(l_1-l)}{l_1-l_0}$.

So, if the cost of monitoring is too large, more precisely if $c > \frac{(l-l_0)(l_1-l)}{l_1-l_0}$, then it is never recommended to probe the random path since the benefit of routing optimization would be in any case lower than the monitoring cost. In other words, $M^*(t) = 0$ for all x.

On the other hand, if $c < \frac{(l-l_0)(l_1-l)}{l_1-l_0}$ then it may be advisable to monitor the random path. More precisely one should do so if $\frac{c}{l-l_0} < x < 1 - \frac{c}{l_1-l}$; in that case $M^*(t) = 1$. That is, if the uncertainty about the state of the non deterministic path is large then it is recommended to monitor it once again in order to update conveniently the filter γ and take better routing decision. On the contrary if x is close to 0 or 1, more precisely if $x > 1 - \frac{c}{l_1-l}$ or $x < \frac{c}{l-l_0}$, we are "pretty sure" of the state of $L(t)$ and it is not worth monitoring the path. These results are summed up in Table 2. Another formulation of this optimal policy, in terms of number of time steps after which a new measurement must be performed, is also provided in Appendix A.

$c > \frac{(l_1-l)(l-l_0)}{l_1-l_0}$	$M^*(t) = 0$
$c < \frac{(l_1-l)(l-l_0)}{l_1-l_0}$	$x > 1 - \frac{c}{l_1-l} \Rightarrow M^*(t) = 0$ $\frac{c}{l-l_0} < x < 1 - \frac{c}{l_1-l} \Rightarrow M^*(t) = 1$ $x < \frac{c}{l-l_0} \Rightarrow M^*(t) = 0$

Table 2: MC with 2 states vs. deterministic path. Optimal monitoring decision as a function of $x = \gamma_{t-1,t}(0)$. Simple threshold policy.

5 A simple model in the case of two random paths

In the previous section we have considered the case of two paths: one path had a deterministic delay and the second path was a two-states MC or HMM. We have proven that the optimal monitoring strategy is in that case a simple threshold strategy. If the cost of monitoring is large one should never monitor, and if not, one should monitor the random path only if the uncertainty about its state is sufficiently large.

In this section we still pursue a closed-form solution but we consider the case of two paths, each of them with random delays (more precisely, a MC or HMM). We will prove that the monitoring decision can be formalized as a partitioning of the space of beliefs (over the states of the paths).

So, let us now consider that one has to take a decision between two random paths and that each of them can be modeled as a Markov Chain (with any number of states). Let us denote as path 0 and path 1 these two random paths. $L^i(t)$ is the delay of path i at time t. It is assumed that L^i is a MC with K^i states, with state space $\{l_0^i, l_1^i, \ldots, l_{K^i-1}^i\}$ and with transition matrix $P^{(i)}$.

The problem is now to decide which path should be taken at time t, and also if the delay of path 0 or path 1 (or both) should be monitored at time t. $M^i(t) = 1$ indicates that the decision is taken to monitor the delay of path i at time t (and conversely $M^i(t) = 0$ means that path i is not monitored). $C(t) = 1$ if path 1 is selected for routing whereas $C(t) = 0$ indicates that one opts for path 0.

5.1 Filtering

The filters $\gamma_{t,t}^{(i)}$ are defined as follows: $\gamma_{t,t}^{(i)}(j) = \mathbb{P}(L^i(t) = l_j^i \mid I_t)$ where I_t denotes the information (measurements) available up to time t. Similarly to Equation (1) it is easily proven that the filters $\gamma_{t,t}^{(i)}$, written as row vectors, are updated as follows:

$$\gamma_{t,t}^{(i)} = \mathbb{1}_{M^i(t)=1} \sum_{j=0}^{K^i-1} (\mathbb{1}_{L^i(t)=l_j^i} e_j) + \mathbb{1}_{M^i(t)=0} \gamma_{t-1,t}^{(i)} \tag{9}$$

The predictors $\gamma_{t-1,t}^i$ are defined as previously: $\gamma_{t-1,t}^i(j) = \mathbb{P}(L^i(t) = l_j^i \mid I_{t-1})$. They can be computed from the filters $\gamma_{t-1,t-1}^i$ taking into account the Markov dynamics:

$$\gamma_{t-1,t}^{(i)} = \gamma_{t-1,t-1}^{(i)} P^{(i)}. \tag{10}$$

5.2 Calculating rewards

The optimal routing choice is to opt for the path whose expected delay is the smallest conditionally to I_t, so:

$$C(t) = \mathbb{1}_{\mathbb{E}(L^1(t)|I_t) \leq \mathbb{E}(L^0(t)|I_t)} \tag{11}$$

The penalized gain, in the case of two random paths, can be defined as:

$$\tilde{G}(t) = -\mathbb{1}_{C(t)=1} L^1(t) - \mathbb{1}_{C(t)=0} L^0(t) - c^1 \mathbb{1}_{M^1(t)=1} - c^0 \mathbb{1}_{M^0(t)=1} \tag{12}$$

where c^0 and c^1 are the costs of monitoring path 0 or path 1. Similarly to the case of a random path and a deterministic path, the reward is here defined as the expected value of the penalized gain given

I_{t-1}:

$$\begin{aligned} R &= \mathbb{E}(\tilde{G}(t) \mid I_{t-1}; M^0(t), M^1(t)), \\ &= -\mathbb{E}(\mathbb{1}_{C(t)=1} L^1(t) \mid I_{t-1}; M^0(t), M^1(t)) \\ &\quad -\mathbb{E}(\mathbb{1}_{C(t)=0} L^0(t) \mid I_{t-1}; M^0(t), M^1(t)) - c^0 \mathbb{1}_{M^0(t)=1} - c^1 \mathbb{1}_{M^1(t)=1}. \end{aligned} \tag{13}$$

R depends implicitly on $M^0(t)$, $M^1(t)$ and I_{t-1}, or, equivalently, on $M^0(t)$, $M^1(t)$, $\gamma^{(0)}_{t-1,t}$, and $\gamma^{(1)}_{t-1,t}$. Indeed all the information I_{t-1} available from past measurements that is useful to predict future values of L^i is summed up in the predictor $\gamma^{(i)}_{t-1,t}$ at time $t-1$. To make this dependance explicit, we shall sometimes write $R_{M^0,M^1}(\gamma^{(0)}_{t-1,t}, \gamma^{(1)}_{t-1,t})$ instead of just R.

In Appendix B, we analyse the four possible cases ($M^0 = M^1 = 0$; $M^0 = M^1 = 1$; $M^0 = 0, M^1 = 1$; and $M^0 = 1, M^1 = 0$) and establish the rational routing decision and the reward for each of them. The results are summed up in Table 3 below, in which we use the notation $\overline{L^i}(t) = \mathbb{E}(L^i(t) \mid I_{t-1}) = \sum_j l^i_j \gamma^{(i)}_{t-1,t}(j)$. Comparing the four values of rewards, we can decide which monitoring decision is the optimal one, as described in what follows.

$M^0(t) = 0$ $M^1(t) = 0$	$C(t) = \mathbb{1}_{\overline{L^1}(t) \le \overline{L^0}(t)}$ $R = -C(t)\overline{L^1}(t) - (1 - C(t))\overline{L^0}(t)$
$M^0(t) = 0$ $M^1(t) = 1$	$C(t) = \mathbb{1}_{L^1(t) \le \overline{L^0}(t)}$ $R = -\sum_{j: l^1_j \le \overline{L^0}(t)} l^1_j \gamma^{(1)}_{t-1,t}(j) - \overline{L^0}(t) \sum_{j: l^1_j > \overline{L^0}(t)} \gamma^{(1)}_{t-1,t}(j) - c^1$
$M^0(t) = 1$ $M^1(t) = 0$	$C(t) = \mathbb{1}_{\overline{L^1}(t) \le L^0(t)}$ $R = -\sum_{i: l^0_i < \overline{L^1}(t)} l^0_i \gamma^{(0)}_{t-1,t}(i) - \overline{L^1}(t) \sum_{i: l^0_i \ge \overline{L^1}(t)} \gamma^{(0)}_{t-1,t}(i) - c^0$
$M^0(t) = 1$ $M^1(t) = 1$	$C(t) = \mathbb{1}_{L^1(t) \le L^0(t)}$ $R = -\sum_{i,j} \gamma^{(0)}_{t-1,t}(i) \gamma^{(1)}_{t-1,t}(j) \min(l^0_i, l^1_j) - c^0 - c^1$

Table 3: Rational routing decisions and rewards for two Markovian paths.

5.3 Monitoring optimisation

In the case of two MCs the goal is to decide if the delays of path 0 and path 1 should be measured, which sums up to selecting the appropriate value for the pair $(M^0(t), M^1(t))$. In that case the optimal monitoring decision is to maximize the reward $R_{M^0,M^1}(\gamma^{(0)}_{t-1,t}, \gamma^{(1)}_{t-1,t})$ as given in Table 3, so that:

$$(M^0, M^1)^* = Arg \max_{M^0, M^1} R_{M^0,M^1}(\gamma^{(0)}_{t-1,t}, \gamma^{(1)}_{t-1,t}) \tag{14}$$

Again, the decision depends only on the values of the one-step predictors $\gamma^{(0)}_{t-1,t}$ and $\gamma^{(1)}_{t-1,t}$ for paths 0 and 1.

In the case of the monitoring decision-making problem for a deterministic path and a MC with two states we have shown that the optimal policy is a threshold policy over $x = \gamma_{t-1,t}(0)$. This simple threshold policy has been summed up in Table 2.

Such a threshold policy exists for the case of two MCs of two states each but we were not able to find a closed form solution for the border between the four decision domains $(M^0, M^1)^* = (0,0), (0,1), (1,0)$ and $(1,1)$ as a function of $x^0 = \gamma^0_{t-1,t}(0)$ and $x^1 = \gamma^1_{t-1,t}(0)$. The solution is consequently in that case numerical.

On the other hand, we have checked that the results we obtain for the case of two MCs of two states each generalize the results obtained for the case of a deterministic path and a MC with two states. To do so, we have examined what happens if one of the predictor values, $\gamma^{(i)}_{t-1,t}(0)$ is equal to 0 or 1, which comes up to assuming that the path i has a constant delay (l^i_0 or l^i_1). In that case the optimal monitoring decision on the other path is the same that one would obtain with the simple threshold policy of Table 2.

5.4 Case of two hidden Markov models

Let us assume now that each of the random paths is not modeled as a Markov Chain but as a Hidden Markov Model. In what follows we are going to calculate the four reward functions R_{M^0,M^1} in that case.

So, for each path i (where $i = 0$ or 1), $L^i(t)$ is a random function of a MC $S^i(t)$. More precisely, $S^i(t)$ is a MC with transition probability matrix $P^{(i)}$ and the path delay $L^i(t)$ depends on the value of $S^i(t)$ only. The probability density function (pdf) of $L^i(t)$ given that $S^i(t) = j$ is the function $p^{(i)}_j(\bullet)$.

The filters $\gamma^{(i)}(t,t)$ and predictors $\gamma^{(i)}_{t-1,t}$ for both paths $i = 0, 1$ can be computed as stated in Equation (2). Given I_{t-1} the path delay $L^i(t)$ is distributed as a mixture distribution with pdf $\sum_j \gamma^{(i)}_{t-1,t}(j) p^{(i)}_j(l)$. As we assume that the two paths are independent the pdf of the couple $(L^0(t), L^1(t))$ is $\sum_{i,j} \gamma^{(0)}_{t-1,t}(i) \gamma^{(1)}_{t-1,t}(j) p^{(0)}_i(l^0) p^{(1)}_j(l^1)$.

Proceeding as in Section 5.2, we can establish which is the rational routing decision and what is the reward for each of the four possible cases. All results are given in closed-form in Appendix C. Similarly to Section 5.3 the optimal monitoring decision can be obtained by maximizing $R_{M^0,M^1}(\gamma^0_{t-1,t}, \gamma^1_{t-1,t})$ over the pair (M^0, M^1) as stated in (14).

6 Taking into account future rewards with Markov Decision Processes

Up to now we did not take into account that monitoring path delays not only permits a better routing decision at time t but also improves future decisions. Indeed if we gain some knowledge about a path delay by monitoring it at time t then it will increase the accuracy of our belief about the state of the path, not only at time t but also in the future, thus permitting better future routing decisions.

From now on, we are thus going to take into account future rewards. We will also consider any number of paths whereas in the previous sections we restricted the study to the case of two paths to obtain closed-form solutions for the optimal monitoring strategy. In this section we use the theory of Markov Decision Processes (MDP, see [18]) to solve the monitoring optimization problem.

We assume that we have $\mathcal{P}$ different paths between a same origin and destination. Each path is indexed by i, where $i \in \{1 \dots \mathcal{P}\}$ and each path i's delay is modeled as a Markov chain with K^i states.

Let $L^i(t)$ denote path i's delay at time t. For each path i we can take at any time the decision of monitoring that path or not, this is represented by M^i, where $M^i = 1$ means that path i is measured. We assume that monitoring path i incurs a cost of c^i.

All information I_t provided by past measurements up to time t and useful to predict future values $L^i(t+k), k \geq 0$ is summarized in the filter $\gamma_{t,t}^{(i)}$. It can be noticed that the filter takes on continuous values. However, it can also be observed that the information I_t can equivalently be summed up, for each path i, as the pair $s^i = (\tau^i, L^i_{\text{last}})$, where τ^i is the number of time steps since the last measurement and L^i_{last} is the last measured value. For example, if we are at time t and if we have just measured the value of $L^i(t)$ then $\tau^i = 0$ and L^i_{last} is the measured value of $L^i(t)$. If we are at time t, a value $\tau^i \geq 1$ means that L^i was monitored for the last time at $t - \tau^i$ and the corresponding value was L^i_{last}. With these notations it comes that $\gamma_{t,t}^{(i)} = e_{L^i_{\text{last}}}(P^{(i)})^{\tau^i}$, where $e_{L^i_{\text{last}}}$ is the canonical vector with a 1 in position L^i_{last}.

We can thus define the state of a controlled Markov chain as $s = \{s^i\}_{i=\{1\ldots\mathcal{P}\}} = \{(\tau^i, L^i_{\text{last}})\}_{i=\{1\ldots\mathcal{P}\}}$. Please note that with this definition of states we have a discrete state space and if τ^i is limited up to a certain value, say τ^i_{max}, the state space is as well finite. In addition, the vector of actions is denoted as M where $M = \{M^i\}_{i\in\{1\ldots\mathcal{P}\}}$. With these definitions s is a discrete-time-discrete state Markov Chain controlled by the monitoring decision M.

In order to properly define the MDP problem we need to define the instantaneous reward. First let us define the minimum expected delay over the $\mathcal{P}$ paths:

$$D(s) = \min_i \mathbb{E}(L^i(t) \mid I_t) = \min(\min_{i:\tau^i\geq 1} \mathbb{E}(L^i \mid \tau^i, L^i_{\text{last}}), \min_{i:\tau^i=0} L^i_{last}). \tag{15}$$

Note that if $\tau^i \geq 1$ then $\mathbb{E}(L^i \mid \tau^i, L^i_{\text{last}}) = \sum_j l^i_j \gamma^{(i)}_{t,t}(j)$ with $\gamma^{(i)}_{t,t} = e_{L^i_{\text{last}}}(P^{(i)})^{\tau^i}$.

With this definition we can define the instantaneous reward as:

$$R(s) = -D(s) - \sum_i c_i \mathbb{1}_{\tau^i=0}. \tag{16}$$

Let now $\mathcal{M}$ be the set of all possible actions and $\mathcal{S}$ the set of states. Let Π be the set of all possible policies π, where $\pi = \{\mu_0, \mu_1, \mu_2, \ldots, \mu_t, \ldots\}$ and μ_t is a function mapping, at time t, $\mathcal{S}$ into $\mathcal{M}$. $\mu_t(s)$ represents for the considered policy the monitoring decision $M(t)$ that is taken at time t if the system is in state s.

We now introduce the infinite horizon discounted cumulative reward J, which is defined as:

$$J_\pi(s_0) = \mathbb{E}_\pi(\sum_{t=1}^{\infty} \rho^t R(s_t) \mid s_0) \tag{17}$$

where $0 < \rho < 1$ is a discount factor that gives more importance to rewards in the close future (and also makes sure that the sum is finite).

The problem is therefore to maximize J_π over Π. It is well known that the MDP problem has an optimal policy that is time stationary [18], that is to say that at the optimum $\mu_0 = \mu_1 = \mu_2 = \ldots = \mu$. So, from now on, we assume that $\mu_t(s)$ depends on s but not on t. Consequently the infinite horizon discounted cumulated reward J is from now on parameterized by μ and denoted as $J_\mu(s)$.

The transition probabilities of the MC s controlled by M can be computed as follows.

$$\mathbb{P}(s' \mid s, M) = \prod_i \mathbb{P}(s'_i \mid s_i, M^i). \tag{18}$$

If $M^i = 0$ then with probability 1 the transition is of the form $(\tau^i, L^i_{\text{last}}) \rightarrow (\tau^i + 1, L^i_{\text{last}})$. On the contrary, if $M^i = 1$ then the possible transitions are $(\tau^i, L^i_{\text{last}}) \rightarrow (0, l^i_j)$, $j = 1 \dots K^i$ and the corresponding probabilities are $e_{L^i_{\text{last}}} P_i^{\tau^i+1} e_j^T$ where $\bullet^T$ is the transposition operator.

The optimal value function and the optimal policy are defined as:

$$V(s) = \max_\mu J_\mu(s) \quad \mu^* = \text{Arg}\max_\mu J_\mu(s) \tag{19}$$

It is well known that the optimal value function is the unique solution of Bellman's equation, which in our case is given by the following equation:

$$V(s) = \max_{M \in \mathcal{M}} \sum_{s'} \mathbb{P}(s' \mid s, M)(R(s') + \rho V(s')). \tag{20}$$

The function V can be approximated as closely as desired thanks to the value iteration algorithm [19]. The value iteration algorithm is a successive approximation algorithm. Iteration n of the algorithm updates function $V_n(s)$ as follows:

$$V_{n+1}(s) = \max_{M \in \mathcal{M}} \sum_{s'} \mathbb{P}(s' \mid s, M)(R(s') + \rho V_n(s')) \tag{21}$$

The optimal policy $\mu^*(s)$ is then obtained at convergence as the argument of the maximum in the above equation:

$$\mu^*(s) = Arg \max_{M \in \mathcal{M}} \sum_{s'} \mathbb{P}(s' \mid s, M)(R(s') + \rho V_n(s')) \tag{22}$$

The complexity of the Value Iteration algorithm depends upon the number of states which is $\prod_i K^i \tau^i_{max}$, and thus grows fast with the number of paths and their possible delays.

7 Implementation considerations

7.1 Implementation framework

The focus of this paper is not to provide a detailed implementation framework, however, a discussion regarding implementation guidelines is still pertinent. Traditional overlay routing solutions have existed since several decades in a completely distributed fashion, as will be detailed in Section 9. In such solutions, typically a software agent runs at each node. This agent performs measurements against all the other nodes, and potentially shares this information with the other nodes. Alternatively, other solutions propose to use adaptive learning techniques to avoid measuring all links in the overlay. With link performance information, each node can compute the optimal path to a destination node. Regarding signaling, a message is defined to be exchanged among agents. This message typically encapsulates the original packet and specifies in its header the path the packet must follow across the overlay. Each agent on the path unpacks the original packet, and repacks it in a new encapsulated message, containing information about the subsequent hops and addressed directly to the next hop. Examples of such an approach are [20] for an all-test solution and [21, 22] for a learning approach.

Such distributed approaches present scalability issues, as mentioned in Section 1. Our approach is substantially different and relies on the SDN paradigm. Recent proposals have also adopted SDN principles as a solution at the Internet scale or the WAN scale, rather than the initial SDN intra-datacenter solutions. Examples are [23], which responds to the many BGP problems by adding an Internet-scale controller, or the well known B4 network from Google [24], which showcases a SDN implementation at a worldwide scale. Even more recently, inter-datacenter connection under the SDN

paradigm was studied in [25], which seeks for resiliency, in [26] which focuses on a cognitive approach for routing decisions, as well as in [27] which discusses an architecture for inter-datacenter QoS-aware routing.

Adopting an SDN paradigm allows us to have a solution with three main features. First, a logically centralized controller allows the communication with each overlay site (node) as well as programming each node. Second a centralized application collects measurements and decides about our two controlling decisions: when to monitor each path, how to route each flow. And third, programmable switches are notified of forwarding rules according to the paths determined by the centralized application. In particular, the standardized OpenFlow protocol can fulfill such programming task. This architecture has been presented in more details in [27].

On the other hand, path delay between two switches is more difficult to measure in a SDN architecture. OpenFlow switches do not timestamp packets, therefore passive measurement of path delay in OpenFlow is unfeasible. In the last years, two interesting works [28] [29] have focused on the problem of active measurements using OpenFlow messages. OpenFlow has a PacketOut message, which is sent by the controller to the switch and allows injecting a packet into the network. Both proposals use a PacketOut message, which carries a timestamped raw packet to be injected into the switch. However, this approach has some inaccuracies. Another approach is to allow the centralized application to ask a probe packets generator to inject measurements packets to the SDN network and program the SDN switches to properly forward these packets and collect the measurements [27].

7.2 Scalability of the MDP approach

In this section we discuss considerations regarding the implementation of the MDP approach for our problem. The Markov Decision Processes framework allows for a simple formulation of the joint monitoring and path selection problem, for an arbitrary number of paths, taking into account rewards on the long-term. However this comes at the cost of an important numerical complexity. Indeed the state and action spaces of the MDP problem grow exponentially with the number of paths.

Let A denote the action space of an MDP and S its state space. The value iteration algorithm requires to store the value function and the policy vectors of dimension $|S|$, the reward matrix of size $|A| \times |S|$, and the transition matrices of size $|S|^2$ for each action. At each iteration $|A| \times |S|^2$ products are computed to update the value function as in Equation 20. Following the notations of Section 6, we have: $\mathcal{P}$ is the number of paths of the MDP problem, K^i is the number of Markov states of path i, and $\tau_{\max}$ is the number of time steps after which the stationary distribution is considered to be attained. With these notations we have $|A| = 2^{\mathcal{P}}$ and $|S| = \prod_{i=0}^{\mathcal{P}} K^i \tau_{\max}$. For example if we consider two paths of two states each, with $\tau_{\max} = 200$, the resulting MDP consists of $(2 \times 200)^2 = 160,000$ states. Even for such a small problem, storing one transition matrix would be difficult as it would contain $(1.6 \times 10^5)^2 = 2.56 \times 10^{10}$ coefficients, approximately 200GB of memory considering 64 bits floating-point numbers.

However these transition matrices are very sparse and this can be used to considerably reduce the required memory, as well as the number of products in the update of the value function. For a given monitoring action M, the number of non-zero coefficients is given by $|S| \times \prod_{i/M^i=1} K^i$ if we measure at least one path, and $|S|$ if we measure zero path. Indeed if we don't measure a path i, with probability 1 the transition is of the form $(\tau^i, L^i_{\text{last}}) \to (\tau^i + 1, L^i_{\text{last}})$. For our two paths problem example, the largest transition matrix ($M^i = 1, \forall i$) contains only $640,000$ non-zero coefficients, so approximately 5MB of memory.

Another observation is that for each origin-destination pair in the overlay, the MDP problems can be solved in parallel. There is no need to consider all the overlay paths jointly since they are optimized independently.

8 Validation results

In this section we will show different validation results. We start with two simple examples with simulated data. Then we consider more realistic contexts with RIPE Atlas measurements: the case of two paths between one origin and one destination, and to end an overlay of 30 RIPE Atlas nodes.

8.1 A first simple example

Let us first consider a simple example of the choice between a path of deterministic delay $l = 8$ and a path which delay is modeled as a two-states discrete-time Markov Chain. The state space of this MC is $\{l_0 = 5, l_1 = 10\}$ and the transition matrix is $P = \begin{bmatrix} 0.99 & 0.01 \\ 0.02 & 0.98 \end{bmatrix}$. The cost of monitoring is assumed to be $c = 0.65$ which is lower than the threshold $(l_1 - l)(l - l_0)/(l_1 - l_0) = 1.2$ (see Table 2). The steady-state vector of P is $[\pi_0 = \frac{2}{3},\ \pi_1 = \frac{1}{3}]$ and its second eigenvalue is $\lambda_2 = 0.97$ which is positive.

Assume that the simple policy of Section 4 is applied. In that case, for values of the one step predictor $\gamma_{t-1,t}(0)$ in the interval $[x_{\min}, x_{\max}]$ with $x_{\min} = c/(l - l_0) = 0.22$ and $x_{\max} = 1 - c/(l_1 - l) = 0.675$ the delay of the stochastic path is monitored. For larger or smaller values the rational choice is not to monitor.

The following simulations were performed. A trajectory of the Markov chain was simulated for $T = 3000$ consecutive time steps and the simple monitoring and routing policy was applied. Results are displayed on Figure 5. The simulated trajectory is shown on the upper part of the figure, and the values of the one step predictor $x = \gamma_{t-1,t}(0)$ are in the middle part of the figure. The monitoring instants are displayed as stars on the upper part. Each time a new measure is performed the filter is updated to $\gamma_{t,t}(0) = 0$ or 1 depending on the result of the measurement ($L(t) = l_1$ or $L(t) = l_0$), so that the one-step predictor $\gamma_{t,t+1}$ is $e_i P$ with $i = 0$ or 1. As one can see from the figure, between two measurements, the predictor converges geometrically towards the steady state distribution of the Markov Chain, $[\pi_0, \pi_1]$. As soon as a bound $x_{\min}$ or $x_{\max}$ is attained a new measure is performed. Note that once the filter $\gamma_{t,t}(0)$ has been updated to 0 (or 1) the threshold $x_{\min}$ (or $x_{\max}$) is attained in a fixed number of steps, $\min\{\tau \text{ s.t. } e_1 P^\tau [\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}] = x_{\min}\}$ (or $\min\{\tau \text{ s.t. } e_0 P^\tau [\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}] = x_{\max}\}$), which can also be pre-computed rather than maintaining the up-to-date value of γ. This precomputation can be done using Eq. (25) of Appendix A. The values obtained for these thresholds are $\theta_0 = 122$ and $\theta_1 = 13$. Further, since $\lambda_2 > 0$, the threshold $\eta_i = \theta_i - 1$, $i = 0, 1$. Thus, the simple policy for this setting will be of the form given in Table 4.

Table 4: Structure of the simple policy for this example

State measured at instant t	Next monitoring instant	Optimal path
0	$t + 122$	stochastic path until $t + 121$
1	$t + 13$	deterministic path until $t + 12$

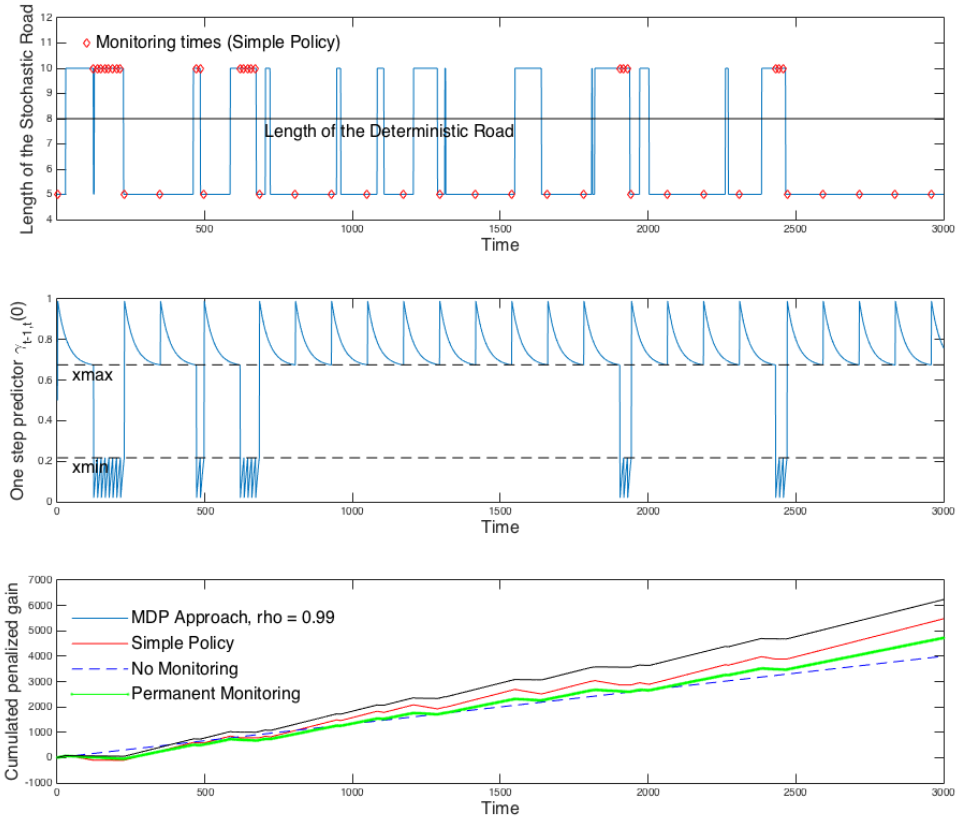

Fig. 5: Choice between a deterministic path and a path modeled as a 2-state MC. Simple threshold policy.

The cumulated value of the penalized gain $\sum_{t=1}^{T} \tilde{G}(t) = \sum_{t=1}^{T} [\mathbb{1}_{C(t)=1}(l - L(t)) - c\mathbb{1}_{M(t)=1}]$ has been computed for $T = 1$ to $T = 3000$. On the lower part of Figure 5 the evolution of the cumulated penalized gain is compared in the case of three policies: the simple policy $M^*(t)$ of Section 4, the constant monitoring policy ($M(t) = 1$ for all t) and the no monitoring policy ($M(t) = 0$ for all t). The cumulated penalized gain evolves randomly. But on the average one can observe a linear tendency with a slope $\mathbb{E}(\tilde{G}(t); M^*)$ which is greater for the monitoring policy M^* than for the permanent or no monitoring policies as we were expecting.

We have then compared the simple policy of Section 4 to the optimal policy obtained with the MDP approach (Section 6). Different values of the discount factor ρ have been considered: $\rho = 0.01$, 0.50 and 0.99. We have computed the total number of measurements performed during the $T = 3000$ time slots as well as the average penalized gain $\frac{1}{T}\sum_{t=1}^{T} \tilde{G}(t)$ for the different policies: no monitoring,

permanent monitoring, simple policy (Section 4), and MDP optimal policies (Section 6) with $\rho = 0.01$, 0.50 and 0.99. Each of these figures has been averaged over the 100 runs. The results are displayed in Table 5.

	No Monitoring	Permanent Monitoring	Simple policy	MDP $\rho = 0.01$	MDP $\rho = 0.50$	MDP $\rho = 0.99$
Average number of measures	0	3000	54.15	56.56	206.29	418.68
Average Penalized Gain	1.33	1.35	1.526	1.532	1.77	1.82

Table 5: Comparison of the performance of MDP optimal policies and simple policy

We can get analytical values for the average penalized gain of the no monitoring policy, the permanent monitoring policy and the simple policy. For the no monitoring policy it is $l-(\pi_0 l_0+\pi_1 l_1) = 1.33$. For permanent monitoring, we get $\pi_0(l - l_0) - c = 1.35$, and for the simple policy, the average penalized gain is given by (27) which for this example evaluates to 1.52. The average number of measures in 3000 time slots can be computed using the formula $\frac{3000}{\xi_0\theta_0+\xi_1\theta_1}$ (see (28)) which yields 52.8. The values in Table 5 correspond well to these analytical values.

As one can observe, when ρ is close to 0 then the performance of the simple policy and of the optimal MDP policy are pretty close, which is what we were expecting. As ρ increases from 0.01 to 0.99 the average penalized gain but also the number of measures increase, respectively from 1.532 to 1.82 and from 54 to 419. It can be remarked that in that case, for $\rho = 0.5$ the performance in terms of average penalized gain is close to the performance that would be obtained with $\rho = 0.99$ (1.77 versus 1.82) but the number of measurements is divided by a factor of 2 (206 instead of 418).

8.2 Two Markov Chains of two states each

Next, we consider the case of two stochastic paths, each of them being represented as a Markov Chain with two levels of delay. The chosen parameters are the following ones. For path 0 the two values of delays are $l_0^0 = 0.5$ and $l_1^0 = 2$ and the probability transition matrix is $P^{(0)} = [[0.7, 0.3], [0.3, 0.7]]$. The cost of monitoring path 0 is set to $c^0 = 0.05$. For path 1 the two values of delays are $l_0^1 = 1$ and $l_1^1 = 3$ and the probability transition matrix is $P^{(1)} = [[0.9, 0.1], [0.1, 0.9]]$. The cost of monitoring path 1 is assumed to be $c^1 = 0.15$.

Let us compute the optimal policy in that case. It can be noticed that in the case of two-state Markov Chains, i.e. $K^0 = K^1 = 2$ the rewards R_{M^0,M^1} of (30), (32), (34) and (36) can be stated as functions of $x^0 = \gamma^0_{t-1,t}(0)$ and $x^1 = \gamma^1_{t-1,t}(0)$ since $\gamma^{(i)}_{t-1,t}(0) + \gamma^{(i)}_{t-1,t}(1) = 1$.

We proceed with a numerical approach, discretizing the space of possible values for x^0 and x^1, and computing the reward $R_{M^0,M^1}(x^0, x^1)$ for each pair (x^0, x^1) of predictors values, and each possible monitoring action (M^0, M^1). Finally, we choose the optimal monitoring policy as the action providing the largest reward, for each pair of predictors values. The results obtained for this set of parameters are shown in Figure 6, where the four decision regions $(M^0 = 1, M^1 = 1)$, $(M^0 = 0, M^1 = 0)$, $(M^0 = 0, M^1 = 1)$ and $(M^0 = 1, M^1 = 0)$ are displayed in different colors.

It can also be checked that, when there is no uncertainty about the state of one of the paths, then the optimal monitoring decision on the other path is consistent with the results obtained in the case of a deterministic path and a 2-state Markov chain path (Table 2).

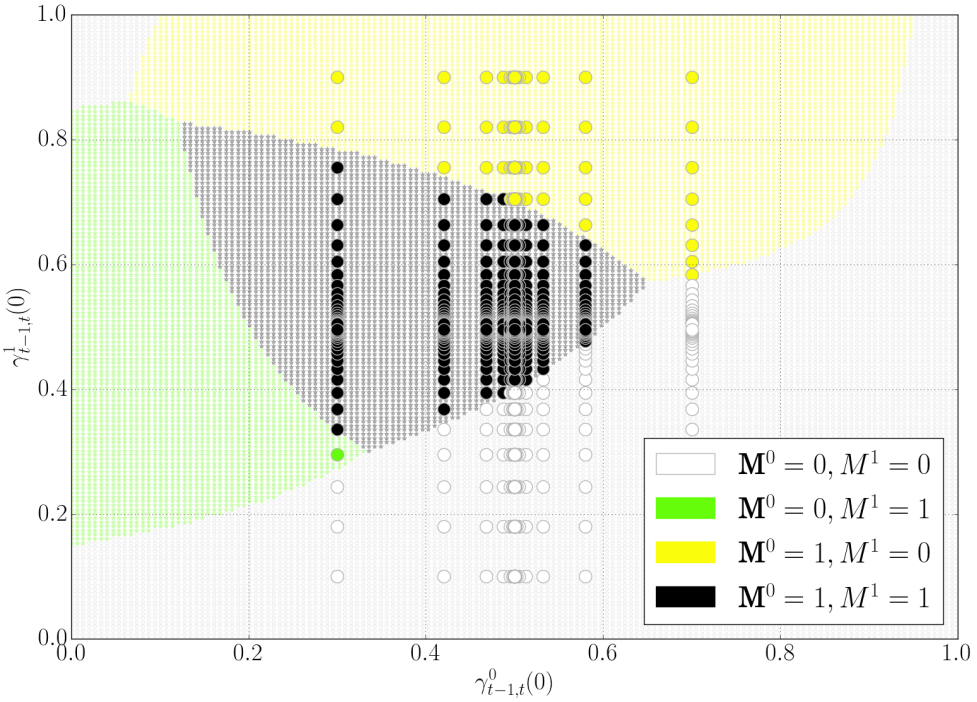

Fig. 6: Choice between two paths, each of them being modeled as a 2-state Markov Chain. Simple monitoring policy and MDP with $\rho = 0.01$.

Assume that there is no uncertainty on path i (with $i = 0$ or 1), that is to say that $x^i = 0$ or $x^i = 1$. The delay of path i is then known to be l, with $l = l^i_0$ if $x^i = 1$ and $l = l^i_1$ if $x^i = 1$. Let us denote as path j the other path. If the cost of monitoring c^j is greater than $c_{\max} = (l^j_1 - l)(l - l^j_0)/(l^j_1 - l^j_0)$ then the delay on this second path should never be monitored. On the contrary, if $c^j < c_{\max}$ then the optimal monitoring decision depends on the value of the predictor $x^j = \gamma^j_{t-1,t}(0)$. More precisely, the path delay should be monitored only if the value of x^i is in the interval $[x_{\min}, x_{\max}]$ with $x_{\min} = c^j/(l - l^j_0)$ and $x_{\max} = 1 - c^j/(l^j_1 - l)$.

We have considered four cases: $x^0 = 0$ or 1, and $x^1 = 0$ or 1. For each of those cases we have computed the thresholds $c_{\max}$, $x_{\min}$ and $x_{\max}$ that should be used to take a monitoring decision on the other path. The results are summarized in Table 6.

In particular, it can be observed that, as $l^0_0 = 0.5 < l^1_0$ and $l^0_0 < l^1_1$ then if $x^0 = 1$ the value of $c^1_{\max}$ is negative, so that the monitoring decision on path 1 is $M^1 = 0$ whatever the cost c^1 of monitoring this path. Similarly, as $l^1_1 = 3 > l^0_0$ and $l^1_1 > l^0_1$ we get when $x^1 = 0$ a negative value for $c^0_{\max}$ and the monitoring decision on path 0 is $M^0 = 0$ whatever the cost c^0.

In the other two limiting cases, namely $x^0 = 0$ or $x^1 = 1$, the bounds $x_{\min}$ and $x_{\max}$ have been computed. It can be checked that these bounds are consistent with what can be oberved for the corresponding limiting cases on Figure 6.

	Path 0	Path 1
$x^0 = 0$	Deterministic delay $l = l_1^0 = 2$	$c_{\max} = \frac{(l_1^1 - l)(l - l_0^1}{l_1^1 - l_0^1} = \frac{(3-2)(2-1)}{3-1} = 1$ $c^1 = 0.15 < c_{\max}$ $x_{\min} = \frac{c^1}{l - l_0^1} = \frac{0.15}{2-1} = 0.15$ $x_{\max} = 1 - \frac{c^1}{l_1^1 - l} = 1 - \frac{0.15}{3-2} = 0.85$ $M^1 = 1$ if $x^1 \in [x_{\min}, x_{\max}]$
$x^0 = 1$	Deterministic delay $l = l_0^0 = 0.5$	$c_{\max} = \frac{(l_1^1 - l)(l - l_0^1)}{l_1^1 - l_0^1} = \frac{(3-0.5)(0.5-1)}{3-1} = -0.625$ $c^1 = 0.15 > c_{\max}$ No monitoring: $M^1 = 0$
$x^1 = 0$	$c_{\max} = \frac{(l_1^0 - l)(l - l_0^0)}{l_1^0 - l_0^0} = \frac{(2-3)(3-0.5)}{2-0.5} = -1.67$ $c^0 = 0.05 > c_{\max}$ No monitoring: $M^0 = 0$	Deterministic delay $l = l_1^1 = 3$
$x^1 = 1$	$c_{\max} = \frac{(l_1^0 - l)(l - l_0^0)}{l_1^0 - l_0^0} = \frac{(2-1)(1-0.5)}{2-0.5} = 0.33$ $c^0 = 0.05 < c_{\max}$ $x_{\min} = \frac{c^0}{l - l_0^0} = \frac{0.05}{1-0.5} = 0.1$ $x_{\max} = 1 - \frac{c^0}{l_1^0 - l} = 1 - \frac{0.15}{2-1} = 0.95$ $M^0 = 1$ if $x^0 \in [x_{\min}, x_{\max}]$	Deterministic delay $l = l_0^1 = 1$

Table 6: Border conditions: optimal monitoring decisions when $x^0 = 0$ or 1 and when $x^1 = 0$ or 1.

Finally we have compared the simple policy to the MDP policy obtained with $\rho = 0.01$, a rather low value of ρ. The MDP decisions are displayed as dots and the same colors are used for the simple and MDP policies to permit a comparison. As it can be observed on Figure 6 the decisions are the same since ρ is very close to 0. It must be remarked that in the simulations the predictors $\gamma_{t-1,t}^{0,1}$ take on only a finite number of values. These values are in the form of $e_{L_{\text{last}}^i}(P^{(i)})^{\tau^i}$. This is why MDP decisions are displayed as colored dots on the $[0,1] \times [0,1]$ space.

8.3 Two paths between two anchors on RIPE Atlas

Let us now consider real datasets. As a matter of example we consider RTT we have obtained in the framework of the RIPE Atlas measurement campaign we have mentioned in Section 2. We consider three particular probes located in Hong Kong (HK), Latvia (LV) and Khazakstan (KZ) and two different paths with the same origin (HK) and destination (KZ): the direct Internet path $HK \to KZ$, which will be referred to as path 1 in the following, and the path $HK \to LV \to KZ$ (in which the LV node is used as a routing proxy), which will be referred to as path 0.

	Path 0 ($HK \to LV \to KZ$)	Path 1 ($HK \to KZ$)
Number of states	$K^0 = 2$	$K^1 = 2$
Mean and variance of delay in each state	$l_0^0 = 350$, $(\sigma_0^0)^2 = 30$ $l_1^0 = 440$, $(\sigma_1^0)^2 = 100$	$l_0^1 = 320$, $(\sigma_0^1)^2 = 30$ $l_1^1 = 370$, $(\sigma_1^1)^2 = 30$
Transition probability matrix	$P^{(0)} = \begin{pmatrix} 0.995 & 0.005 \\ 0.05 & 0.95 \end{pmatrix}$	$P^{(1)} = \begin{pmatrix} 0.995 & 0.005 \\ 0.005 & 0.995 \end{pmatrix}$

Table 7: Parameters of the HMM models for the two paths between HK and KZ.

The RTTs between each pair of nodes have been monitored during 7 days. The performance of both paths is displayed on the upper part of Figure 7. As one can see from this figure the shortest path is sometimes $HK \to KZ$ and sometimes $HK \to LV \to KZ$. A Gaussian HMM model is used in order to characterize both paths. The parameters of these HMMs are shown in Table 7.

The optimal MDP policy has been computed for $\rho = 0.9$ with the value iteration algorithm (cf. Section 6) , assuming that the cost of monitoring either of the two paths is $c^0 = c^1 = 4$. To do so we have simplified the HMM models of the delays to MC models, by neglecting the Gaussian noise of the HMM model and assuming that the delay of path k when in state i is its mean value l_i^k. Once the optimal monitoring policy μ^* computed, it was applied by processing the time series of delays sequentially from $t = 0$ to $t = T = 1008$. At each time t, the values of the filters $\gamma_{t,t}^{(i)}$ were updated for both paths according to Equation (2). The state of the controlled Markov chain was also updated. Note that since we do not directly observe the state of path i, this state was estimated to be l_j^i, where j is the index of the most likely component of the HMM given past observations, that is to say the $Arg\max_j \gamma_{t,t}^{(i)}(j)$.

In our dataset around 2% of the measurements were missing. This corresponds to cases when the RIPE Atlas probe did not answer. Missing data are simply dealt with by upgrading $\gamma_{t,t}^{(i)}$ according to $\gamma_{t,t}^{(i)} = \gamma_{t-1,t-1}^{(i)} P^{(i)}$ for the corresponding time indexes t. Equivalently, the pdf $p_j^{(i)}(L^i(t)), j = 1, \ldots, K^i$ are all set to an equal probability $1/K^i$ in Equation (2) if $L^i(t)$ is missing. We also have to deal with a few outliers, that is to say values $L^i(t)$ for which pdf $p_j^i(L^i(t))$ are extremely low for all components j of the HMM model (in fact numerically equal to 0). In order to avoid numerical problems in the update of the filter $\gamma_{t,t}^{(i)}$ we assume a minimum value of $p_j^i(L^i(t))$ that has been set to 10^{-4}.

Results are displayed on Figure 7. On the upper part the delays of both paths are displayed (in blue and red). Below one can observe at which instants each path is monitored (dots along the timeline). One can also see the evolution of the values of the filters $\gamma_{t,t}^{(i)}, i \in \{0, 1\}$ along the timeline. Then we show which path is chosen at each instant ($C = 0$ or $C = 1$) and the delay of the chosen path (black stars) along time (compared to the delays of the two possible paths, in blue and red dotted lines).

We observe that in these simulations the average frequency of monitoring is 11% for $HK \to KZ$ and 12.5% for $HK \to LV \to KZ$. This depends of course on the value of the monitoring costs c^i as well as on the value of ρ. The larger ρ is the more frequent measurements are, and the larger the costs c^i are the less frequent monitoring is. Table 8 sums up some performance indicators. Whereas the average delay on paths HK $\to$ KZ and HK $\to$ LV $\to$ KZ are 396 and 397 msec, the average delay on the optimized path is 345 msec. The optimized path is on the average around 50 msec shorter than the two other paths. Now, if both the delay and the cost of monitoring are taken into account, the average cost is 346 for the optimized path, compared to 396 and 397 for the direct Internet path and for the two hops path.

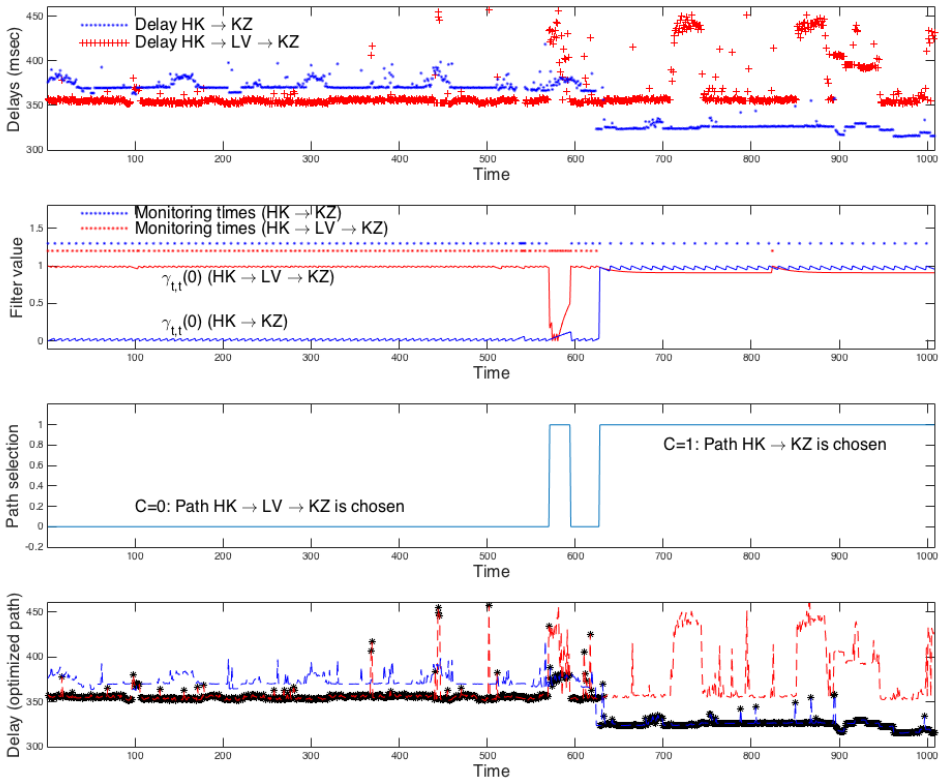

Fig. 7: RIPE Atlas dataset. Choice between two paths. MDP approach.

	Optimized Path	Internet Path HK → KZ	Alternative Path HK → LV → KZ
Average Delay	345 msec	396 msec	397 msec
Average Cost (Delay + Monitoring)	346	396	397

Table 8: RIPE Atlas dataset: path HK → LV → KZ versus Internet path HK → KZ. Monitoring optimized with a MDP approach ($\rho = 0.9$).

8.4 Overlay of 30 RIPE Atlas anchors

Following the methodology of the previous section, we give quantitative results of the monitoring optimization on a larger topology. We simulated a 30-node topology by choosing a subset of anchors in the RIPE Atlas dataset described in Section 2. Five anchors were randomly selected on each of

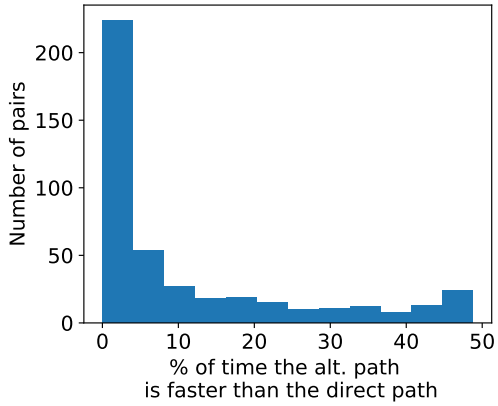

Fig. 8: Distribution of the percentage of time the alternative path is faster than direct path.

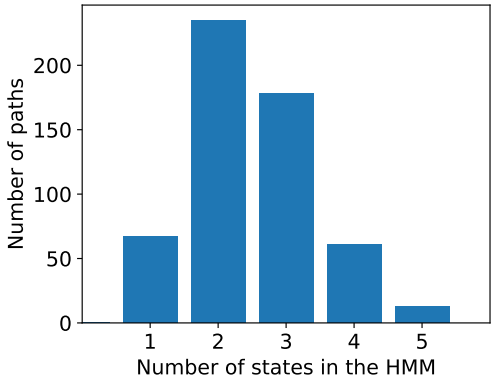

Fig. 9: Distribution of the number of states learned.

the six continents. For each origin-destination pair we kept the two paths which were the shortest ones most of the time. We refer to the path that is the shortest most of the time as the *direct path*, and to the other path as the *alternative path*. Figure 8 displays the distribution of the percentage of time the alternative path is faster than the direct path. We kept the origin-destination pairs for which the alternative path is the fastest at least 10% of the time, and learned the paths models using the HDP-HMM method described in Section 2. The distribution of the number of states learned per HMM is given on Figure 9.

The *optimal path* is the path that would result from an optimal routing decision at each time step if the exact values of the paths' delays were known. So, the optimal path is not always the same path: it is sometimes the direct path and sometimes the alternative path, but on the optimal path the delay is always the smallest one. Figure 10 displays the distribution of the average delay gain of the optimal path, versus the average delay of the direct path and of the alternative path for the retained origin-destination pairs of the considered topology. Box bounds correspond to the 25th and 75th percentile, middle bar is the median, and whiskers indicates min. and max. values. As can be seen from the two upper box plots, choosing dynamically the path in an overlay at each time step obviously allows to reduce the average delay from source to destination.

For each origin-destination pair we have then simulated three monitoring policies, *always measure*, *never measure*, and *MDP*. We have set the measurement cost to $c = 1$ for every path, and we have solved the MDP problems with $\tau_{\max} = 500$ and $\rho = 0.9$. The *always measure* policy can be used to benchmark our Markovian models. In the *always measure* policy the routing decision is taken on the basis of the expected value of the paths' delays under the HDP-HMM model. Imperfections in the models may result in expected delay values slightly different from the true delays.

Figure 10 displays the additional average delay observed for each policy, compared to the optimal path. Two observations can be made. We first observe that the average delay with an *always measure* policy is very close to the delay of the optimal path, meaning that HMMs provides sufficiently correct delay estimation for an optimal path selection. Next we observe that the average delay with the MDP policy is very close to the delay of the *always measure* policy but with a really small measurement cost. Indeed the average measurement cost with the MDP policy is 0.17 (max 0.65), so 91% lower than the constant cost of 2 for the *always measure policy*. This means that on average, with the parameters that we have chosen, each of the two paths is monitored 8.5% of the time only and yet the routing

performance in terms of delay is very close to the performance that one could get if both paths were constantly monitored.

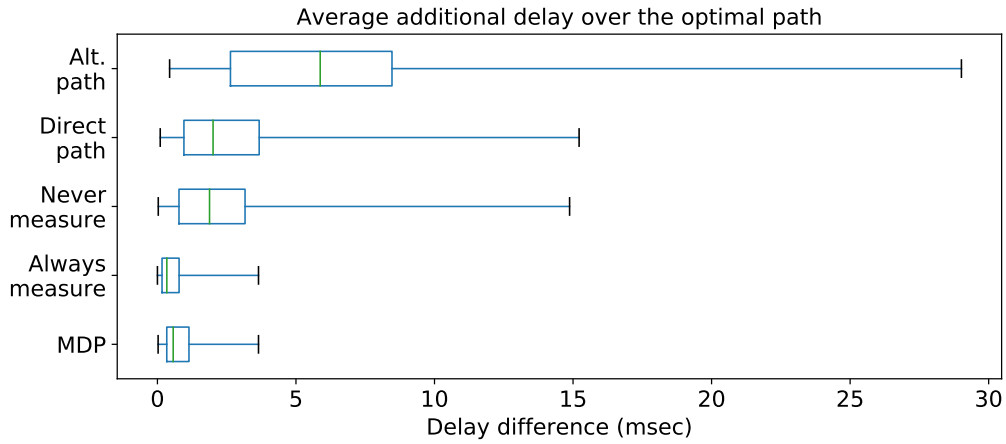

Fig. 10: Comparison of delay for constant and optimized path selection versus the ideal path.

These results demonstrate that the MDP approach succeeds in providing a parsimonious monitoring policy, where a small number of measurements in particular states allow to maintain a sufficient knowledge about the state of the network to correctly choose the shortest path most of the time.

Using a single-threaded Julia [30] implementation of the value iteration algorithm, it took approximately 4 minutes and 600MB of memory to solve one problem with two paths of three states each and $\tau_{\max} = 500$, on an Intel Core i7-7600U CPU @ 2.80GHz. Using a Python implementation of HDP-HMM [17] it took approximately 6 seconds to learn the model for a single path (2520 data points).

To end, Figures 11 and 12 display the effect of the measurement cost c on the frequency of measurements and on the delay. We have compared the average delay of the *MDP policy* for different values of c with the average delay of an *always measure* policy. For large values of c we tend to a *never measure* policy, while for $c = 0$ we obtain a delay equal to an *always measure* policy. Obviously the number of measures tends to 2 when $c \to 0$. But for very small values of c the average number of measures is significantly lower than 2 (it rapidly drops down to 0.5). Indeed the paths' performance is very stable, and for most paths the probability to remain in the current state is extremely high. The HDP-HMM model and the MDP policy capture the fact that, even if the measurement cost is very low, it is not always necessary to measure to take an optimal routing decision. On Figure 11 the curve displays the median while the color fill shows the 25th and 75th percentiles of the average number of measurements per time slot. Figure 12 displays the value of the additional delay of the *MDP policy* (w.r.t. the *always measure policy* path) as a function of the average number of measures per time slot. Each dot corresponds to a particular value of c and a particular origin-destination pair.

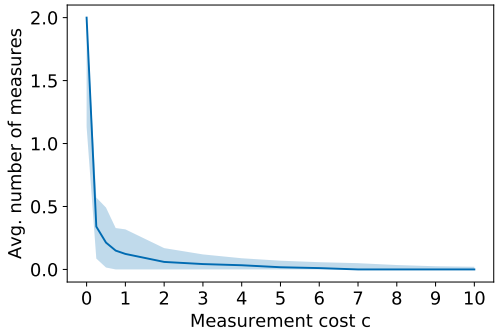

Fig. 11: Evolution of the average number of measures per time slot for different values of c.

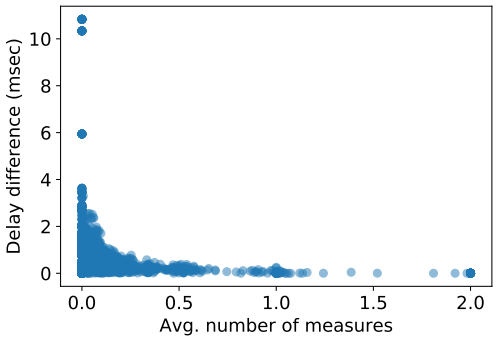

Fig. 12: Additional delay over an always measure policy vs. the observed avg. number of measures.

9 Related Work

Even if overlay networks are mainly used nowadays for overlaying virtualized Layer-2 networks over Layer-3 networks [31, 32] or for dynamic content delivery [33–37], they have been used in the past for a variety of purposes, including self-organization in peer-to-peer networks [38–40], application-layer multicast in IP networks [41–45], and protection against DDoS attacks [46, 47].

The first works to demonstrate the potential of overlay networks for route optimization were Detour [48] and RON (Resilient Overlay Network) [20], inspiring many other approaches [49–53]. All these routing overlay technologies use the all-pair probing approach, that is, they regularly monitor the health of all overlay links in order to dynamically select the best source-destination paths in the overlay according to application-specific metrics. Although this approach guarantees optimal performance, its main drawback is that it does not scale very well: as the number of participating routers n increases, the $O(n^2)$ probing overhead becomes a limiting factor. Evidences indicate that in practice it can support only about 50 routers before the probing overhead becomes overwhelming [20]. As shown by Chen et al. [54] this probing overhead can be slightly reduced by selectively monitoring $k \leq O(nlog(n))$ linearly independent paths that can fully describe all the $O(n^2)$ paths (see also [55]).

The design of parsimonious monitoring strategies allowing to deploy routing overlays over a sizable population of routers without compromising route quality has been addressed only recently. The use of adaptive learning techniques inspired from Cognitive Packet Networks (CPN) [21, 22] was investigated in [56], providing evidences that near-optimal routing can be achieved with a modest monitoring effort (see also [57] for the application of CPN techniques to peer-to-peer overlay networks). Another approach based on the adversarial learning algorithm EXP3 was investigated in [58], where significant improvements over native IP routing were obtained, both in terms of latency and throughput. Whereas the above results were obtained with routing overlays using legacy IP technologies, an SDN-based architecture for distant points interconnection through a resilient and high-performance overlay network was proposed in [27] (see also [59]) . Note that in [56, 58], it is assumed that a small, but given number of paths are probed at each measurement epoch, and the only question is therefore how to select these paths for minimizing the routing cost. In contrast to these works, the present paper seeks for an optimal trade-off between the monitoring and routing costs, without any constraint on the monitoring budget.

Similar questions have been addressed in the context of network tomography [60–63]. Tomographic techniques infer the state of network elements from end-to-end probe exchanged between a few probing nodes (usually referred to as beacons or monitors). These techniques are known to provide a cost-efficient alternative to placing and operating sophisticated monitors at all nodes in a network. Some authors have designed solutions for minimizing the cost of the monitoring infrastructure, that is, for placing the minimum number of monitors to identify all the links, a problem which is known to be NP-hard [64–68]. Besides, since active probing can generate tremendous traffic and degrade the overall network performance, the minimization of the online probing cost has been addressed in [69–72]. The issue here is that not all the probing paths contain useful information for identifying the link metrics of interest, implying that monitoring paths have to be selected so as to optimize a trade-off between identifiability and probing cost. Recently, it was shown in [71] that by carefully selecting the probing paths, the amount of probing traffic can be significantly reduced while achieving the same monitoring performance as would be obtained with all-path probing. Despite some similarities, the above works are different from our work in that they focus on identifying the performance of each individual link with as few end-to-end probes as possible, whereas in our case the monitoring decisions are driven by the perspective of lower routing costs. In addition, the above works consider a static setting, whereas the problem addressed in the present paper is fundamentally a sequential decision-making problem.

A more-closely related work is [73]. In the context of wireless networks, the authors use the approach of multi-armed bandits to obtain heuristic policies to determine which links to monitor. The approach consists of relaxing a strict constraint on the maximum number of links that can be monitored simultaneously to a weaker constraint on the long-term average number of links. They show that optimal monitoring policy for the relaxed problem lies in the set of threshold policies under some assumptions on the structure of the transition probabilities of the Markov chains as well as on the reward structure. One of these assumptions requires that the reward of choosing a link be higher with newer information. The problem we consider and the approach we take differs [73]. First of all, we have a monitoring cost that is included in the objective but there is no constraint on the number of links monitored. Second, the reward for routing on a link needs not to be restricted by the assumption of a better reward for newer information.

Another contribution of this article is to introduce a new statistical model for Internet delays. Previously finite mixture models [11] and more recently infinite mixture models [12] have been used to characterize delays. In our article we introduce a new model for delays, the Hierarchical Dirichlet Process - Hidden Markov Model (HDP-HMM). HMMs are generalization of mixture models. With HMMs the dependence of delay values over previous values is taken into account, whereas a mixture model assumes that they are independent. The HMM captures the stability of the performance of paths over time. This is essential to limit the number of measurements that are required to maintain an accurate information on the expected values of the paths delays and take close to optimal routing decisions. The work in [12] uses Dirichlet process priors to cope with unknown orders in mixture models, while HDP-HMMs use Dirichlet process priors for flexible description of dependency among successive observations via HMM of unknown order. Optimal monitoring of HMM has been addressed in [74, 75] but with a cost function dependent on the state estimation error, in contrast with the reward dependent on the path selection in our article.

10 Conclusions and future work

A Markov Decision Process formulation of a decision making problem in which one decides which paths to monitor in a network was presented. The objective of the decision maker was to minimize a linear combination of long-run average delay and monitoring cost.

We first introduced the application of HDP-HMM for estimating HMMs parameters from delay observations where the number of states is not known a-priori. We then established the structure of the optimal myopic policy for a simple scenario, in which there is a deterministic path and a stochastic path whose delay evolves randomly according to a MC or a HMM with two states. It turns out that the optimal myopic policy amounts to monitor the stochastic path only when the uncertainty on the state of the stochastic path is sufficiently high. In other words, this path should be monitored when the belief on this path being in "good state" is in between two thresholds. Similarly, an explicit characterization of the optimal myopic policy was obtained for the case of two stochastic paths. For the general case of more than two paths and more than two states, we have shown that the problem of maximizing the cumulated discounted reward can be cast as a MDP, whose optimal policy can be obtained with the Value Iteration algorithm. We have demonstrated the validity of our approach in different settings. In particular, using a 30 nodes topology and a dataset of RTT measurements between RIPE Atlas anchors, we have shown that with a reduction of monitoring load by 91% it is possible to obtain a routing performance which is the same that one would obtain if the performance of paths was monitored permanently.

Our ongoing work focuses on several directions. We would like to improve the scalability of the decision-making problem in case of large number of paths and large number of states. A possible approach could be to exploit the insights obtained in this paper in simple scenarios in order to design efficient policies for that general case. We may design policies in which the monitoring decision depends on the uncertainty on the state of a few paths that may have the minimum delay.

As another research direction, we would like to use the theory of Partially Observable Markov Decision Processes in order to study the case where the delays of the paths are modeled as HMMs. We would also like to investigate how online learning methods, for example Q-learning, could be used in order to learn the optimal policy. We also think that the properties of the value function of the MDP problem could be studied and lead to faster resolution algorithms, in particular we would like to investigate modified policy iteration methods.

Concerning the statistical characterization of RTT series with an unknown number of states, we would like to investigate further HDP-HMMs. We would also like to consider other metrics, such as the number of hops available from traceroute measurements, or BGP announcements and the AS paths.

With respect to implementation of the proposed overlay framework, we are currently developing an SDN emulated testbed so as to challenge our proposal with a real implementation.

Finally, we believe that the same methodology could be applied to other network metrics and in other contexts. Regarding the first point, we would be interested in extending our work to a context in which the routing metrics is the available bandwidth, although this metrics is notably difficult to accurately estimate. Regarding the second point, we would like to study contexts in which control decisions are based on the system state, but where this state cannot be measured constantly, so that a parsimonious monitoring strategy is relevant.An example of such a context is the allocation of tasks to servers in large data-centers.

Acknowledgments

The authors would like to thank the STIC AmSud program which financially supports their collaboration through the PROVE project (2016-2017). P. Belzarena was partially supported by CSIC, UDELAR (GRUPOS I+D, ARTES).

References

1. L. Peterson, S. Shenker, and J. Turner. Overcoming the internet impasse through virtualization. In *in Proceedings of the 3rd ACM Workshop on Hot Topics in Networks (HotNets-III)*, November 2004.
2. J. Touch, Y. Wang, L. Eggert, and G. Finn. A virtual Internet architecture. Technical Report ISI-TR-2003-570, ISI, March 2003.
3. N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe. The case for separating routing from routers. In ACM Press, editor, *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, 2004.
4. M. Beck, T. Moore, and J.S. Plank. An end-to-end approach to globally scalable programmable networking. In ACM Press, editor, *in Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, 2003.
5. P. Belzarena and L. Aspirot. End-to-end quality of service seen by applications: A statistical learning approach. *Computer Networks*, 54(17):3123 – 3143, 2010.
6. RIPE NCC Staff. RIPE Atlas: A Global Internet Measurement Network. *Internet Protocol Journal*, 18(3), 2015.
7. RIPE Atlas. `https://atlas.ripe.net/`. Accessed: 2017-01-01.
8. H. Pucha, Y. Zhang, Z. M. Mao, and Y. C. Hu. Understanding network delay changes caused by routing events. *SIGMETRICS Perform. Eval. Rev.*, 35(1):73–84, June 2007.
9. M. Rimondini and C. Squarcella. From BGP to RTT and beyond: Matching BGP routing changes and network delay variations with an eye on traceroute paths, 2013.
10. Y. Schwartz, Y. Shavitt, and U. Weinsberg. A measurement study of the origins of end-to-end delay variations. In *Passive and Active Measurement (PAM)*, 2010.
11. M-F. Shih and A. O. Hero. Unicast-based inference of network link delay distributions with finite mixture models. *IEEE Transactions on Signal Processing*, pages 2219–2228, 2003.
12. R. Fontugne, J. Mazel, and K. Fukuda. An empirical mixture model for large-scale RTT measurements. In *IEEE Conference on Computer Communications (INFOCOM)*, 2015.
13. A.P. Dempster, N.M. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B (Methodological)*, 39(1):1–38, 1977.
14. Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, December 2006.
15. C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 1998.
16. J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.
17. Bayesian inference in HSMMs and HMMs. `https://github.com/mattjj/pyhsmm`. Accessed: 2017-11-01.
18. D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2000.
19. R. Bellman. A Markov Decision Process. *Journal of Mathematics and Mechanics*, (6), 1957.
20. D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, SOSP '01, pages 131–145, New York, NY, USA, 2001. ACM.
21. E. Gelenbe, R. Lent, A. Montuori, and Z. Xu. Towards networks with cognitive packets. In *Proc. 8th Int. Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems (IEEE MASCOTS), San Francisco, CA, USA*, pages pp 3–12, August 29-September 1 2000.
22. E. Gelenbe and Z. Kazhmaganbetova. Cognitive packet network for bilateral asymmetric connections. *IEEE Trans. Industrial Informatics*, 10(3):1717–1725, 2014.
23. V. Kotronis, X. Dimitropoulos, and B. Ager. Outsourcing the routing control logic: Better internet routing based on sdn principles. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, pages 55–60, New York, NY, USA, 2012. ACM.
24. S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat. B4: Experience with a globally-deployed software defined WAN. *SIGCOMM Comput. Commun. Rev.*, 43(4):3–14, August 2013.
25. A. Fressancourt and M. Gagnaire. A SDN-based network architecture for cloud resiliency. In *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*, Jan 2015.
26. F. Francois and E. Gelenbe. Optimizing secure sdn-enabled inter-data centre overlay networks through cognitive routing. In *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 283–288, Sept 2016.
27. P. Belzarena, G. Gomez, I. Amigo, and S. Vaton. SDN-based overlay networks for QoS-aware routing. In *ACM SIGCOMM Workshop on Fostering Latin-American Research in Data Communication Networks*, 2016.
28. N. L. M. van Adrichem, C. Doerr, and F. A. Kuipers. OpenNetMon: Network monitoring in OpenFlow software-defined networks. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–8, May 2014.

29. C. Yu, C. Lumezanu, A. Sharma, Q. Xu, G. Jiang, and H. V. Madhyastha. *Software-Defined Latency Monitoring in Data Center Networks*, pages 360–372. Springer International Publishing, Cham, 2015.
30. J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.
31. J. Moy. RFC 7348: Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. Technical report, 2014.
32. R. Moats. Open DOVE. `https://wiki.opendaylight.org/view/Open_DOVE:Main`, 2013.
33. K. Andreev, B. M. Maggs, A. Meyerson, and R. Sitaraman. Designing overlay multicast networks for streaming. In *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), San Diego, CA, USA*, June 2003.
34. H. Rahul, M. Kasbekar, R. Sitaraman, and A. Berger. Towards realizing the performance and availability benefits of a global overlay network. In *Passive and Active Measurement Conference, Adelaide, Australia*, March 2006.
35. T. Leighton. Improving performance on the internet. *Communications of the ACM*, 52(2), February 2009.
36. E. Nygren, R. K. Sitaraman, and J. Sun. The Akamai network: A platform for high-performance internet applications. *ACM SIGOPS Operating Systems Review*, 44(3), July 2010.
37. R. K. Sitaraman, M. Kasbekar, W. Lichtenstein, and M. Jain. *Overlay Networks: An Akamai Perspective*. In Advanced Content Delivery, Streaming, and Cloud Services. John Wiley & Sons, 2014.
38. I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM'01*, San Diego, California, USA., August 27-31 2001.
39. A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *In the Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, 2001.
40. B. Y. Zhao, L. Huang, J. Stribling, S.C. Rhea, A.D. Joseph, and J.D. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 2003.
41. Y.H. Chu, S.G. Rao, and H. Zhang. A case for end system multicast. In ACM, editor, *ACM SIGMETRICS 2000*, pages 1–12, Santa Clara, CA, June 2000.
42. S. Banerjee, B. Bhattacharjee, C. Kommareddy, and G. Varghese. Scalable application layer multicast. In *Proc. of the ACM SIGCOMM*, New York, USA, 2002.
43. D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. Almi: An application level multicast infrastructure. In *Proc of the 3rd USNIX Symposium on Internet Technologies and Systems (USITS)*, San Francisco, CA, USA, March 2001.
44. J. Liebeherr and T. K. Beam. Hypercast: A protocol for maintaining multicast group members in a logical hypercube topology. In *Proceedings of the First International COST264 Workshop on Networked Group Communication*, pages 72–89. Springer-Verlag, 1999.
45. A. Babay, C. Danilov, J. Lane, M. Miskin-Amir, D. Obenshain, J. Schultz, J. Stanton, T. Tantillo, and Y. Amir. Structured overlay networks for a new generation of internet services. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1771–1779, June 2017.
46. R. Stone. Centertrack: An IP overlay network for tracking DoS floods. In *in Proc. USENIX Security Symposium '00*, August 2000.
47. J. Wang, L. Lu, and A.A. Chien. Tolerating denial-of-service attacks using overlay networks - impact of overlay network topology. In *in Proc. First ACM Workshop on Survivable and Self-Regenerative Systems*, 2003.
48. A. Collins. The Detour framework for packet rerouting. Technical report, 1998.
49. K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall. Improving the reliability of Internet paths with one-hop source routing. In *In Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, 2004.
50. S.-Y. Hu and G.-M. Liao. Scalable peer-to-peer networked virtual environment. In *In NetGames'04: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*, pages 129–133, New York, NY, USA, 2004. ACM Press.
51. A. Nakao, L. Peterson, and A. Bavier. Scalable routing overlay networks. *SIGOPS Oper. Syst. Rev.*, 40(1):49–61, 2006.
52. P. Medagliani, S. Paris, J. Leguay, L. Maggi, C. Xue, and H. Zhou. Overlay routing for fast video transfers in CDN. *CoRR*, abs/1701.09011, 2017.
53. A. Rai, R. Singh, and E. Modiano. A distributed algorithm for throughput optimal routing in overlay networks. *CoRR*, abs/1612.05537, 2016.
54. Y. Chen, D. Bindel, H. Song, and R. H. Katz. An algebraic approach to practical and scalable overlay network monitoring. *ACM SIGCOMM Computer Communication Review*, 34(4):55–66, Oct 2004.
55. F. Li and M. Thottan. End-to-end service quality measurement using source-routed probes. In *INFOCOM*, 2006.
56. O. Brun, L. Wang, and E. Gelenbe. Big data for autonomic intercontinental overlays. *IEEE Jour. Selected Areas in Communications (special Issue on Emerging Technologies in Communications - Big data)*, 34:575–584, 2016.

57. M. Gellman. *QoS Routing for Real-time Traffic*. PhD thesis, Imperial College London, 2007.
58. O.Brun, H. Hassan, and J. Vallet. Scalable, self-healing, and self-optimizing routing overlays. In *IFIP Networking 2016*, Vienna, Austria, May 17-19 2016.
59. S. Sahhaf, W. Tavernier, D. Colle, and M. Pickavet. Adaptive and reliable multipath provisioning for media transfer in SDN-based overlay networks. *Computer Communications*, 106:107 – 116, 2017.
60. Y. Vardi. Network Tomography: estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association. American Statistical Association*, 91(433):365377, 1996.
61. A. Coates, A. O. Hero III, R. Nowak, and Bin Yu. Internet tomography. *IEEE Signal Processing Magazine*, 19(3):47–65, May 2002.
62. D. Rubenstein, J. Kurose, and D. Towsley. Detecting shared congestion of flows via end-to-end measurement. *IEEE/ACM Transactions on Networking*, 10(3):381395, 2002.
63. N. Etemadi Rad, Y. Ephraim, and B. L. Mark. Delay Network Tomography Using a Partially Observable Bivariate Markov Chain. *IEEE/ACM Transactions on Networking*, 25(1):126–138, 2017.
64. J.D. Horton and A. Lopez-Ortiz. On the number of distributed measurement points for network tomography. In *Proceedings of the 2003 ACM SIGCOMM conference on Internet measurement*, page 204209, 2003.
65. Y. Bejerano and R. Rastogi. Robust monitoring of link delays and faults in IP networks. *IEEE/ACM Transactions on Networking (TON)*, 14(5):1092 – 1103, Oct 2006.
66. R. Kumar and J. Kaur. Practical beacon placement for link monitoring using network tomography. *IEEE Journal on Selected Areas in Communications*, 24(12):1092 – 1103, Dec 2006.
67. Y. A. Pignolet, S. Schmid, and G. Trédan. Tomographic Node Placement Strategies and the Impact of the Routing Model. *Proc. ACM on Measurement and Analysis of Computing Systems*, 1(2):42:1–42:23, 2017.
68. T. He, L. Ma, A. Gkelias, K. K Leung, A. Swami, and D. Towsley. Robust Monitor Placement for Network Tomography in Dynamic Networks. In *IEEE INFOCOM*, Apr. 2016.
69. A. Gopalan and S. Ramasubramanian. On identifying additive link metrics using linearly independent cycles and paths. *IEEE/ACM Transactions on Networking (TON)*, 20(3):906 – 916, Jun 2012.
70. L. Ma, T. He, K. K. Leung, D. Towsley, and A. Swami. Efficient identification of additive link metrics via network tomography. In *IEEE ICDCS*, 2013.
71. D. Z. Tootaghaj, T. He, and T. La Porta. Parsimonious tomography: Optimizing cost-identifiability trade-off for probing-based network monitoring. In *IFIP Performance 2017*, 2017.
72. T. He. Distributed Link Anomaly Detection via Partial Network Tomography. In *IFIP Performance*, Nov. 2017.
73. M. Larranaga, M. Assaad, A. Destounis, and G. S. Paschos. Asymptotically optimal pilot allocation over Markovian fading channels. *ArXiv e-prints*, August 2016.
74. V. Krishnamurthy. Algorithms for optimal scheduling and management of hidden markov model sensors. *IEEE Trans. Signal Processing*, 50:1382–1397, 2002.
75. V. Krishnamurthy. *Partially Observed Markov Decision Processes: From Filtering to Controlled Sensing*. Cambridge University Press, 2016.

A Another formulation of the optimal myopic policy of Section 4

The transition matrix of a two-state MC has certain properties that can also be used to compute the thresholds and the long-term average reward. Here, the long-term average reward is defined as

$$\bar{\mathsf{G}} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \tilde{G}(t).$$

This quantity does not depend on the initial conditions. Hence, on the right-hand side, we have omitted to specify them.

Let the transition matrix of the stochastic path be

$$P = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix} \end{array} \tag{23}$$

where the labels indicate the index of the stochastic path. It is known that P^u can be expressed as

$$P^u = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} \pi_0 + \pi_1\lambda_2^u & \pi_1 - \pi_1\lambda_2^u \\ \pi_0 - \pi_0\lambda_2^u & \pi_1 + \pi_0\lambda_2^u \end{bmatrix} \end{array} \tag{24}$$

where λ_2 is the second eigenvalue of P (the first one being 1) and

$$\pi := [\pi_0, \pi_1] = \left[\frac{1-q}{1-p+1-q}, \frac{1-p}{1-p+1-q}\right]$$

is the steady state probability vector of P. Thus, $\gamma_{t,t+u}$ can be computed explicitly given $\gamma_{t,t}$.

Suppose that at time instant 0, we measure state $i = 0$ or 1. Define

$$\theta_i = \min\{u > 0 : M(u) = 1 | M(0) = 1, L(0) = i\}.$$

Then, θ_i is the first instant after 0 at which we decide to measure, given that we measured state i at time 0. From Table 2 for $t > 0$, $M(t) = 1$ if

$$\frac{c}{l - l_0} < x < 1 - \frac{c}{l_1 - l},$$

where $x = \mathbb{P}(L(u) = 0 | L(0) = i)$. Thus,

$$\theta_i = \min\left\{u > 0 : [P]^u_{i,0} \in \left(\frac{c}{l-l_0}, 1 - \frac{c}{l_1 - l}\right)\right\}. \tag{25}$$

where $[P]^t_{i,0}$ can be computed from (24).

Note that θ_i can be infinite. In this case, once we measure state i, there will be no further measurements. A sufficient condition for θ_i to be finite is

$$\pi_0 \in \left(\frac{c}{l-l_0}, 1 - \frac{c}{l_1 - l}\right).$$

For the threshold policy we can also compute $\bar{\mathsf{G}}$. For this, we shall embed $L(t)$ at measurement epochs. Let us call this embedded chain $E(t)$. Then, the transition matrix of $E(t)$ is given by

$$Q = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} [P]^{\theta_0}_{0,0} & [P]^{\theta_0}_{0,1} \\ [P]^{\theta_1}_{1,0} & [P]^{\theta_1}_{1,1} \end{bmatrix} \end{array}. \tag{26}$$

Note that $[P]^t_{i,j}$ can be computed using (24). Let $\xi = [\xi_0, \xi_1]$ be the steady-state vector of Q. Then,

$$\xi_0 = \frac{[P]^{\theta_1}_{1,0}}{[P]^{\theta_0}_{0,1} + [P]^{\theta_1}_{1,0}}.$$

Define G_i to be the average cumulated reward between successive measurement epochs given that the measured state was i. Then,

$$G_i = -c + \max(0, l - l_i) + \sum_{t=1}^{\theta_i - 1} \max(0, \mathbb{E}(l - L(t) | L(0) = i)).$$

The first two terms are for the reward at the instant we measure and the last term is for the total expected reward during the interval between two measurements.

The average long-term reward of the threshold policy can then be computed using the formula

$$\bar{\mathsf{G}} = \frac{\sum_i \xi_i G_i}{\sum_i \xi_i \theta_i}. \tag{27}$$

To see this, let $W_i(T)$ be the number of visits to state i of the chain E in the interval {1,T}. The starting state is not important and can be arbitrary. Then,

$$\begin{aligned} \bar{\mathsf{G}} &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \tilde{G}(t) \\ &= \lim_{T\to\infty} \frac{1}{T} (W_0(T) G_0 + W_1(T) G_1) \\ &= \lim_{T\to\infty} \frac{W_0(T) G_0 + W_1(T) G_1}{W_0(T)\theta_0 + W_1(T)\theta_1} \\ &= \frac{\sum_i \xi_i G_i}{\sum_i \xi_i \theta_i}. \end{aligned}$$

A similar reasoning can be employed to compute the probability of measuring in a time slot. This probability evaluates to

$$\frac{1}{\xi_0\theta_0 + \xi_1\theta_1}. \tag{28}$$

Special case of $\lambda_2 > 0$: When $\lambda_2 > 0$, $\mathbb{E}(L(t)|L(0) = i)$ is a monotone function of t which is increasing for $i = 0$ and decreasing for $i = 1$. Thus, the optimal policy will have the following property: in state 0, for the first η_0 steps the message will be sent on the stochastic path and then for the remaining $\theta_0 - \eta_0$, it will be sent on the deterministic path. On the other hand, if the measured state is 1, then for the first η_1 steps, the message will be sent on the deterministic path and for the remaining $\theta_1 - \eta_1$ it will be sent on the stochastic path. The thresholds η_0 and η_1 can be computed using the relation

$$\eta_0 = \max\left\{u \in \{0, \theta_0 - 1\} : \lambda_2^u > \frac{\mathbb{E}(L_\infty) - l}{\pi_1(l_1 - l_0)}\right\}, \tag{29a}$$

$$\eta_1 = \max\left\{u \in \{0, \theta_1 - 1\} : \lambda_2^u > \frac{l - \mathbb{E}(L_\infty)}{\pi_0(l_1 - l_0)}\right\}, \tag{29b}$$

where $\mathbb{E}(L_\infty) = \pi_0 l_0 + \pi_1 l_1$ is the stationary mean delay on the stochastic path.

Since it is assumed that $\frac{l_1 - l}{l_1 - l_0} < 1 - \frac{c}{l_1 = l}$, for this special case, after some algebra, it can be seen that $\eta_i = \theta_i - 1$. Table 9 summarizes the optimal policy for $\lambda_2 > 0$.

Table 9: Structure of the optimal policy for $\lambda_2 > 0$

State measured at instant t	Next monitoring instant	Optimal path
0	$t + \theta_0$	stochastic for $t \in \{0, \theta_0 - 1\}$
1	$t + \theta_1$	deterministic for $t \in \{0, \theta_1 - 1\}$.

B Proof of Results in Section 5.2

In what follows, we are going to consider four cases: $M^0 = M^1 = 0$; $M^0 = M^1 = 1$; $M^0 = 0, M^1 = 1$; and $M^0 = 1, M^1 = 0$. In each case we establish which is the rational routing decision and what is the reward. To simplify notations, we let $\overline{L^k}(t) = \mathbb{E}(L^k(t) \mid I_{t-1}) = \sum_j l_j^k \gamma_{t-1,t}^{(k)}(j)$ denote the expected delay of path k when $M^k(t) = 0$.

First case: $M^0(t) = M^1(t) = 1$

In that case the routing decision $C(t)$ is not independent from the path delays $L^0(t)$ and $L^1(t)$ (knowing I_{t-1}). Indeed the values $L^0(t)$ and $L^1(t)$ are measured and the routing decision is $C(t) = \mathbb{1}_{L^1(t) \leq L^0(t)}$. As a consequence, the reward can be rewritten as:

$$\begin{aligned} R &= -\mathbb{E}(\mathbb{1}_{L^1(t) \leq L^0(t)} L^1(t) \mid I_{t-1}) - \mathbb{E}(\mathbb{1}_{L^0(t) < L^1(t)} L^0(t) \mid I_{t-1}) - c^0 - c^1, \\ &= -\sum_i \sum_j \gamma_{t-1,t}^{(0)}(i) \gamma_{t-1,t}^{(1)}(j) \min(l_i^0, l_j^1) - c^0 - c^1. \end{aligned} \tag{30}$$

Second case: $M^0(t) = M^1(t) = 0$

In that case the routing decision $C(t)$ is independent from the path delays $L^0(t)$ and $L^1(t)$ (conditionnally to I_{t-1}) since $L^0(t)$ and $L^1(t)$ are not measured. $C(t)$ is deterministic given I_{t-1} and equals:

$$C(t) = \mathbb{1}_{\overline{L^1}(t) \leq \overline{L^0}(t)}. \tag{31}$$

The reward is thus equal to:

$$\begin{aligned} R &= -C(t)\mathbb{E}(L^1(t) \mid I_{t-1}) - (1 - C(t))\mathbb{E}(L^0(t) \mid I_{t-1}), \\ &= -C(t)\overline{L^1}(t) - (1 - C(t))\overline{L^0}(t). \end{aligned} \tag{32}$$

Third case: $M^0(t) = 0, M^1(t) = 1$

In that case the delay of path 1, $L^1(t)$, is monitored whereas the delay of path 0 is forecasted based on past measurements. Consequently the routing decision is:

$$C(t) = \mathbb{1}_{L^1(t) \leq \overline{L^0}(t)}. \tag{33}$$

And the reward is equal to:

$$\begin{aligned} R &= -\mathbb{E}(L^1(t)\mathbb{1}_{L^1(t) \leq \overline{L^0}(t)} \mid I_{t-1}) - \mathbb{E}(L^0(t)\mathbb{1}_{L^1(t) > \overline{L^0}(t)} \mid I_{t-1}) - c_1, \\ &= -\sum_{j : l_j^1 \leq \overline{L^0}(t)} l_j^1 \gamma_{t-1,t}^{(1)}(j) - \mathbb{P}\left(L^1(t) > \overline{L^0}(t)\right) \overline{L^0}(t) - c_1, \\ &= -\sum_{j : l_j^1 \leq \overline{L^0}(t)} l_j^1 \gamma_{t-1,t}^{(1)}(j) - \overline{L^0}(t) \sum_{j : l_j^1 > \overline{L^0}(t)} \gamma_{t-1,t}^{(1)}(j) - c_1. \end{aligned} \tag{34}$$

Fourth case: $M^0(t) = 1, M^1(t) = 0$

Similarly to the previous case, when $M^0 = 1, M^1 = 0$ the optimal routing decision is:

$$C(t) = \mathbb{1}_{\overline{L^1}(t) \leq L^0(t)}. \tag{35}$$

And the reward has the following expression:

$$\begin{aligned} R &= -\mathbb{E}(L^0(t)\mathbb{1}_{\overline{L^1}(t) > L^0(t)}) - \mathbb{E}(L^1(t)\mathbb{1}_{\overline{L^1}(t) \leq L^0(t)}) - c_0, \\ &= -\sum_{i : l_i^0 < \overline{L^1}(t)} l_i^0 \gamma_{t-1,t}^{(0)}(i) - \mathbb{P}(L^0(t) \geq \overline{L^1}(t) \mid I_{t-1})\overline{L^1}(t) - c_0, \\ &= -\sum_{i : l_i^0 < \overline{L^1}(t)} l_i^0 \gamma_{t-1,t}^{(0)}(i) - \overline{L^1}(t) \sum_{i : l_i^0 \geq \overline{L^1}(t)} \gamma_{t-1,t}^{(0)}(i) - c_0. \end{aligned} \tag{36}$$

C Proof of Results in Section 5.4

We derive below the expression of the rewards R_{M^0,M^1} for the four combinations of M^0 and M^1 in the HMM case. In why follows, it is assumed that l_i^0 and l_i^1 denote the mean values of the paths 0 and 1 when in state i, that is to say $l_i^0 = \mathbb{E}(L^0(t) \mid S^0(t) = i)$ and $l_i^1 = \mathbb{E}(L^1(t) \mid S^1(t) = i)$. As in Section 5.2, we let $\overline{L^k}(t) = \mathbb{E}(L^k(t) \mid I_{t-1}) = \sum_j l_j^k \gamma_{t-1,t}^{(k)}(j)$.

First case: $M^0(t) = M^1(t) = 1$

In that case it holds according to Equation (30) that:

$$R = -\mathbb{E}(\mathbb{1}_{L^1(t)\leq L^0(t)} L^1(t) \mid I_{t-1}) - \mathbb{E}(\mathbb{1}_{L^0(t)<L^1(t)} L^0(t) \mid I_{t-1}) - c^0 - c^1,$$

where

$$\mathbb{E}(\mathbb{1}_{L^n(t)\leq L^m(t)} L^n(t) \mid I_{t-1}) = \sum_{i,j} \gamma^{(m)}_{t-1,t}(i)\gamma^{(n)}_{t-1,t}(j) \int\int \mathbb{1}_{l^n \leq l^m} l^n p_i^{(m)}(l^m) p_j^{(n)}(l^n)\, dl^m dl^n,$$

for $m, n \in \{0, 1\}$, $n \neq m$. Consequently,

$$R = -\sum_{i,j} \gamma^{(0)}_{t-1,t}(i)\gamma^{(1)}_{t-1,t}(j) \int\int \min(l^0, l^1)\, p_i^{(0)}(l^0)\, p_j^{(1)}(l^1)\, dl^0 dl^1 - c^0 - c^1. \tag{37}$$

Second case: $M^0(t) = M^1(t) = 0$

With l_i^0, l_i^1 and $\overline{L^k}(t)$ as defined in the introduction of this section, when $M^0(t) = M^1(t) = 0$ Equations (31) and (32) are still true, that is to say:

$$C(t) = \mathbb{1}_{\overline{L^1}(t)\leq \overline{L^0}(t)},$$

and

$$R = -C(t)\overline{L^1}(t) - (1 - C(t))\overline{L^0}(t).$$

Third case: $M^0(t) = 0, M^1(t) = 1$

Following the same reasoning as in Equations (33) and (34) it holds that:

$$R = -\mathbb{E}(L^1(t)\mathbb{1}_{L^1(t)\leq \overline{L^0}(t)} \mid I_{t-1}) - \overline{L^0}(t)\, \mathbb{P}(L^1(t) > \overline{L^0}(t)) - c_1. \tag{38}$$

If F_j^1 is the cumulative distribution function of $L^1(t)$ when $S^1(t) = j$, then

$$\mathbb{P}(L^1(t) > \overline{L^0}(t)) = 1 - \sum_j \gamma^{(1)}_{t-1,t}(j) F_j^1(\overline{L^0}(t)), \tag{39}$$

and

$$\mathbb{E}(L^1(t)\mathbb{1}_{L^1(t)\leq \overline{L^0}(t)} \mid I_{t-1}) = \sum_j \gamma^{(1)}_{t-1,t}(j) \int_0^{\overline{L^0}(t)} y\, p_j^{(1)}(y)\, dy. \tag{40}$$

Let us consider in particular the Gaussian case, that is to say $[L^1(t) \mid S^1(t) = j] \sim N(l_j^1, (\sigma_j^1)^2)$. Then, as we have observed in Section 4.1, the following closed forms can be used:

$$F_j^1(l) = \Phi(\tfrac{l-l_j^1}{\sigma_j^1}) \text{ and } \textstyle\int_0^l y\, p_j^{(1)}(y)dy = l_j^1 \Phi(\tfrac{l-l_j^1}{\sigma_j^1}) - \sigma_j^1 \phi(\tfrac{l-l_j^1}{\sigma_j^1}) \tag{41}$$

where $\phi(x)$ and $\Phi(x)$ are the p.d.f and c.d.f. of $N(0, 1)$.

Fourth case: $M^0(t) = 1, M^1(t) = 0$

Reciprocally, we consider the case $M^0(t) = 1, M^1(t) = 0$. Then,

$$R = -\mathbb{E}(L^0(t)\mathbb{1}_{L^0(t)\leq\overline{L^1}(t)} \mid I_{t-1}) - \overline{L^1}(t)\,\mathbb{P}(L^0(t) > \overline{L^1}(t)) - c_0,$$

where

$$\mathbb{P}(L^0(t) > \overline{L^1}(t)) = 1 - \sum_i \gamma^{(0)}_{t-1,t}(i) F^0_i(\overline{L^1}(t)), \tag{42}$$

and

$$\mathbb{E}(L^0(t)\mathbb{1}_{L^0(t)\leq\overline{L^1}(t)} \mid I_{t-1}) = \sum_i \gamma^{(0)}_{t-1,t}(i) \int_0^{\overline{L^1}(t)} y\, p^{(0)}_i(y)\, dy. \tag{43}$$

Again, in the Gaussian case, a closed form expression can be found for $F^0_i(l)$ and for $\int_0^l y\, p^{(0)}_i(y)dy$.

Authors biography

Sandrine Vaton is Full Professor at IMT Atlantique (Brest, France). She has received an Engineering Degree and a PhD in signal processing from Télécom Paris, a master's degree in Probabilities and Finance from Université Pierre et Marie Curie (UPMC, Paris, France) and an accreditation to supervise research (HDR) in Computer Science from Université Rennes 1 (France). Her research interests are statistical modelling of telecommunications network and traffic, performance evaluation, network monitoring and anomaly detection.

Olivier Brun is a CNRS research staff member at LAAS, Toulouse, France. He graduated from the Institut National des Télécommunications and he was awarded his PhD degree from Université Toulouse III. Before joining LAAS, he spent one year working for Delta Partners company as a R&D engineer and was in charge of the NEMOS project for British Telecom. His research interests lie in queueing and game theories as well as network optimization.

Maxime Mouchet is currently a PhD student at IMT Atlantique, from which he also obtained an engineering degree in telecommunications. His work concerns the optimization of active monitoring in computer networks through statistical modelling and prediction of the QoS.

Pablo Belzarena received his Electrical Engineering degree and his M.S. and PhD degrees in Electrical Engineering from the Universidad de la Republica, Uruguay. He is Full Professor of the Telecommunications Department at Universidad de la Republica, Uruguay. His research interests are performance evaluation and economic models for networks and software defined radio.

Isabel Amigo is an associate professor at IMT Atlantique. She obtained a PhD (2013) in computer science from Telecom Bretagne, France, and Universidad de la República, Uruguay, and an electrical engineering diploma (2007) from Universidad de la República. Her main research interests are traffic engineering, interdomain QoS, network economics, game theory, and network performance and protocols.

Balakrishna J. Prabhu is a CNRS researcher at LAAS-CNRS, Toulouse, France. His research interests are in performance analysis of communication systems using stochastic modelling and game theory. He obtained his PhD from INRIA Sophia Antipolis (France) in 2005 and M.Sc (Eng.) from the IISc (India). Before joining LAAS-CNRS, he was postdoctoral student at VTT (Finland), CWI, Eurandom and TU/e (The Netherlands).

Thierry Chonavel obtained a PhD from Télécom Paris in 1992. Since 1993 he has been Professor at IMT Atlantique (formerly Télécom Bretagne). His research is related to statistical signal processing methods with applications to several fields (transmissions, speech, sonar, radar, networks). In the area of hidden Markov modelling, he contributed to techniques for tracking states with unknown and varying dimension in dynamical systems observed from sensor arrays.