



A Markovian Approach for Improving End-to-End Data Rates in the Internet

Marine Ségneré-Yter, Olivier Brun^(✉), and Balakrishna Prabhu

LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
brun@laas.fr

Abstract. We model the tradeoff between the monitoring costs and gain in throughput for overlay-based routing in the Internet. A Markovian model is shown to fit the real throughput traces quite well. The tradeoff problem is formulated as Markov decision process and it is observed that the myopic policy that maximizes the immediate utility is close to optimal on the real traces.

1 Introduction

More than two decades ago, it was observed that the performance of network flows could be improved by choosing other paths than those computed by IP routing protocols (see, e.g., [7]). Routing overlay networks were then proposed as a solution for achieving spectacular performance improvements, without the need to re-engineer the Internet (see [1] and references therein). An overlay network is composed of Internet end-hosts which can monitor the quality of Internet paths between themselves by sending probe packets. Since all pairs of nodes are connected, the default topology of a routing overlay is that of a complete graph. Although the monitoring cost is highly variable depending on the metric to be probed, it is usually not possible to discover an optimal path by probing all links in large overlay networks (see [2] for a graph-theoretic analysis of this issue). An alternative approach is to devise a parsimonious monitoring approach making the trade-off between the quality of routing decisions and the monitoring cost. Given a source and a destination node in the overlay, the idea is to probe only a small number of overlay paths between the two nodes at each measurement epoch, but to choose those paths so as to make the best routing decision.

Assuming known Markovian models for path delays, this trade-off problem was formulated as a Markov Decision Process (MDP) in [8]. Using delay data collected over the Internet, it was shown that the optimal monitoring policy enables to obtain a routing performance almost as good as the one obtained when all paths are always monitored, but with a modest monitoring effort.

In this paper, we adopt the theoretical framework introduced in [8], but focus on data throughput rather than RTT. We note that efficient parsimonious monitoring strategies are even more important for the throughput metric. Indeed, although lightweight methods for estimating the available bandwidth between

two Internet end-hosts were proposed in [4,5], in practice the only accurate method is to transfer a large file between the two endpoints. It turns out that the MDP formulation for maximizing the data throughput is equivalent to the MDP formulation for minimizing the RTT. The contribution of the present paper is therefore not on the theoretical side, but rather to investigate the applicability of the approach proposed in [8] for optimizing throughput in overlay networks. To this end, we use we use throughput measurements that were made between 9 AWS (Amazon Web Services) data centres.

2 MDP Formulation

The problem formulation in this section is essentially the same as in [6,8] except that the quantity of interest is bandwidth instead of delay. Consider a single origin-destination pair and $\{1, 2, \dots, P\}$ a set of P paths between the origin and the destination. The network topology is thus that of parallel links. At time step t , path i is assumed to have a bandwidth $X_i(t)$, where $X_i(t)$ is a discrete-time Markov chain taking values in a finite set. The transition matrix for path i will be denoted by M_i .

At each time step, the routing agent has to decide on which path it should send data. For this, the agent has at its disposal the last observed bandwidth for each path. Further, it can choose to measure the bandwidth on one or more paths and update its state information before taking the routing decision. The agent incurs a cost of c_i for probing path i independently of time step. The decision-maker must find a compromise between paying to retrieve information from the system to get a higher bandwidth and not retrieving information leading to a lower bandwidth.

Let $\mathbf{u}(t) \in \{0, 1\}^P$ whose i th component indicates whether path i is monitored in time step t or not. The total cost paid for action $\mathbf{u}(t)$ is $\sum_{i|u_i(t)=1} -c_i = -\mathbf{c} \cdot \mathbf{u}(t)$ with $\mathbf{c} = (c_1, \dots, c_P)$. Let $r(t)$ be the path chosen in time step t . A policy θ can be defined by the sequence $\{(\mathbf{u}(t), r(t))\}_{t \geq 0}$. Just as in [8], it can be seen that knowing only the last observed bandwidth for a link is not enough to determine the distribution the bandwidth that will be obtained in a given step. The state can be made Markovian by incorporating the age of the last observed bandwidth as well. That is, the pair $(y_i(t), \tau_i(t))$, where $y_i(t)$ is the last observed bandwidth of link i at time t and $\tau_i(t)$ is the age of the last observation is sufficient as the state variable for a Markovian representation of path i . All this information is summarized in a vector $\mathbf{s}(t) = (s_1(t), s_2(t), \dots, s_P(t))$ where $s_i(t) = (y_i(t), \tau_i(t))$.

Since the state is now Markovian, the problem can be formulated as a Markov Decision Process (MDP). This MDP can be further simplified by noting that in the model, the routing decision does not have any impact on the evolution of the state. Thus, a locally greedy routing decision conditioned on $\mathbf{u}(t)$ and the current state will be optimal. In other words, for a given $\mathbf{u}(t)$, it will be optimal to choose the path that maximizes the expected bandwidth. With this in mind, the decision problem can be reduced to determining which paths to monitor in

each time step. For a given state $s \equiv (y, \tau)$ of path i , define the belief on the bandwidth being z of this path as follows: $b_i(z|s) := \mathbb{P}(X_i(\tau) = z | X_i(0) = y)$, which is just the probability of path i transitioning from y to z in τ steps, and can be computed by choosing the corresponding element of M_i^τ .

If path i is measured, then its actual bandwidth, $X_i(t)$, will be known and can be used in the routing decision. Otherwise, it is its expected conditional bandwidth $\mathbb{E}[X_i|s_i] = \sum_{x \in \mathcal{X}_i} x \cdot b_i(x|s_i)$ that will be used. The locally greedy routing decision will be to choose $r(t)$ that maximizes $(u_i X_i + (1 - u_i) \mathbb{E}[X_i|s_i])$. Note that this decision is taken after performing the action of monitoring the subset of selected links. This leads to maximum bandwidth conditioned on \mathbf{s} and \mathbf{u} of $B(\mathbf{X}|\mathbf{s}, \mathbf{u}) = \max_i (u_i X_i + (1 - u_i) \mathbb{E}[X_i|s_i])$, and an expected maximum bandwidth of:

$$\bar{B}(\mathbf{s}; \mathbf{u}) = \sum_{\mathbf{x}} \left(\prod_{i=1}^P b_i(x_i|s_i) \right) B(\mathbf{x}|\mathbf{s}; \mathbf{u}). \quad (1)$$

Here the product measure is used because X_i s evolve independently.

Now that the routing decision is known, the final MDP takes the form:

$$\max_{\theta} \mathbb{E}_{\mathbf{s}_0}^{\theta} \left\{ \sum_{t=0}^{\infty} \rho^t [\bar{B}(\mathbf{s}(t); \mathbf{u}(t)) - \mathbf{c} \cdot \mathbf{u}(t)] \right\}. \quad (2)$$

where $\theta \equiv \mathbf{u}(t)_{t \geq 0}$, is limited to monitoring decisions only.

We remark that the above problem formulation resembles the multi-armed bandit (MAB) framework. However, unlike standard MABs in which the cost function is decomposable in the individual costs of the bandits, in our problem the overall cost is not decomposable.

3 Numerical Results

In order to validate our approach on real data, for which the Markovian assumption is not perfectly met, we use throughput measurements that were made between 9 AWS (Amazon Web Services) data centres located in several places around the world. In summer 2015, we measured the available throughput between all pairs of data centres every five minutes, by transferring a 10 MB file through the Internet, for a period of four days. We thus collected some 8.3×10^4 measurement data over the 4 days period. Assuming that the available throughput over a path is the minimum of the throughputs of its constituent links, the analysis of these data revealed that the IP route is the maximum throughput route only in 23% of the cases, and that most of the time, the maximum throughput overlay route passes through 1 or 2 intermediate nodes (see [1] for details).

We selected three origin-destination (OD) pairs: Virginia/Ireland, Virginia/Frankfurt and Frankfurt/Tokyo. For the first two pairs, in addition to the IP path, we selected two alternative paths which were sometimes better than the IP path, whereas for the last example there was one alternative path.

For each path, we fitted a Markov model using a clustering method called, Hierarchical Agglomerative Clustering [3]. This method creates a hierarchy between clusters, like a tree. At the beginning, each value of bandwidth is a cluster. The algorithm agglomerate one by one the closest data (in term of a distance metric chosen) together in a new cluster, until it creates one big cluster. On our data, we use the Euclidean distance between the bandwidth values. After that, we decided where to *cut* the tree and obtain a certain number of clusters.

Now that we have our different states, we have to determine the transition probability matrices, P_i . We elaborate this matrix by counting the number of transition between each pair of states. Finally, we search the minimum value τ_{\max_i} which satisfy $\max(P_i^{\text{lim}} - P_i^{\tau_{\max_i}}) < 10^{-2}$. It appears that, on real data, the τ_{\max} per link is lower than 10 and the number of states per link is between 2 and 12.

We evaluate the average utility (see (2)) for four policies: optimal, myopic policy that optimizes the immediate cost only, a receding horizon policy (with a horizon of 3) and a decomposition based heuristic. For a description of the last two policies, we refer the reader to [6].

First, we check that the Markov models we fitted are representative of the real traces. For this, for each OD pair, using the transition matrices, we generate a sample path of throughputs on each of the paths. On these sample paths, we apply the three heuristics (but not the optimal) and compute the average utility for each policy. We then apply the policies on the real traces and compute the average utilities. Table 1 shows the percentage relative error between the average utility computed on a sample path and that on the corresponding real trace. The relative error is less than 2% which indicates a good match. Finally, Table 2 shows the utilities of the four policies for varying monitoring costs. One surprising observation from these examples is that the myopic policy is almost optimal.

Table 1. Percentage relative error between utility computed using Markov model and on real trace.

(a) Virginia/Ireland.

c	1	2	3	4	5
Myopic	1.38	1.14	0.52	0.95	0.01
RH	1.38	0.98	0.50	0.62	0.01
H2	1.73	1.81	1.84	1.90	1.97

(b) Virginia/Frankfurt.

c	1	2	3	4	5
Myopic	1.36	0.91	0.6	1.07	0.03
RH	1.34	1.046	0.62	1.067	0.03
H2	1.63	1.90	1.92	1.97	2.06

Table 2. Utilities for different policies as a function of the monitoring cost.

(a) Virginia/Ireland.					(b) Virginia/Frankfurt.					(c) Frankfurt/Tokyo.				
c	OPT	Myopic	RH	H2	c	OPT	Myopic	RH	H2	c	OPT	Myopic	RH	H2
1.0	41.98	41.98	41.98	40.45	1.0	41.91	41.91	41.91	40.77	1.0	58.44	58.44	58.44	57.83
2.0	39.42	39.40	39.43	38.72	2.0	39.45	39.47	39.45	38.84	2.0	57.71	57.67	57.71	56.59
3.0	37.90	37.89	37.90	36.98	3.0	38.02	38.03	38.03	36.97	3.0	57.20	57.20	57.20	55.35
4.0	36.99	36.92	36.97	35.25	4.0	37.11	37.11	37.11	35.13	4.0	57.20	57.20	57.20	54.11
5.0	36.43	36.43	36.43	33.52	5.0	36.38	36.36	36.36	33.30	5.0	57.20	57.20	57.20	53.09

4 Conclusion and Future Work

The results indicate that Markovian models are a good fit for throughput on paths in the Internet. Further, a myopic policy is nearly optimal for minimizing a linear combination of the throughput and monitoring costs.

As future work, we would first like to understand why the myopic policy works well on these examples. It would be interesting to obtain conditions under which this is true. Next, we would like to generalize these models to multi-agent settings in which each node of the overlay can be seen as an agents. These agents can be either cooperative or be non-cooperative. Another possible improvement of the setting would be to allow the routing decision to influence the future evolution of the bandwidth of the path and to get state information from the current routing decision.

Acknowledgment. The work of Marine Ségneré was partially funded by a contract with Direction Générale de l’Armement (DGA), France.

References

1. Brun, O., Hassan, H., Vallet, J.: Scalable, self-healing, and self-optimizing routing overlays. In: IFIP Networking, Vienna, pp. 64–72 (2016)
2. Thraves Caro, C., Doncel, J., Brun, O.: Optimal path discovery problem with homogeneous knowledge. *Theory Comput. Syst.* **64**(2), 227–250 (2020)
3. James, G., Witten, D., Hastie, T., Tibshirani, R.: *An Introduction to Statistical Learning: With Applications in R*. Springer, New York (2014). <https://doi.org/10.1007/978-1-4614-7138-7>
4. Johnsson, A., Melander, B., Björkman, M.: On the analysis of packet-train probing schemes. In: Proceedings of the International Conference on Communication in Computing, Las Vegas, USA, June 2004
5. Melander, B., Bjorkman, M., Gunningberg, P.: A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. In: Global Telecommunications Conference, GLOBECOM 2000, vol. 1, pp. 415–420. IEEE (2000)
6. Mouchet, M.: Scalable monitoring heuristics for improving network latency. In: IEEE/IFIP Network Operations and Management Symposium (IEEE/IFIP NOMS 2020), Budapest, Hungary, April 2020
7. Savage, S., Collins, A., Hoffman, E., Snell, J., Anderson, T.: The end-to-end effects of internet path selection. *SIGCOMM Comput. Commun. Rev.* **29**(4), 289–299 (1999)
8. Vaton, S., et al.: Joint minimization of monitoring cost and delay in overlay networks: optimal policies with a Markovian approach. *J. Netw. Syst. Manag. (JONS)* **27**(1), 188–232 (2019)