Learning Multiagent Congestion Control Schemes

AmirMahdi Ahmadinejad, Saied Mehdian {ahmadi,smehdian}@stanford.edu

Stanford University

Abstract

Human designed protocols (e.g. TCP) do not adopt well to new technologies and circumstances. In this work, we devise a method to auto-generate congestion control protocols using Machine Learning. Our method is much faster and more scalable as compared to previous works.

Introduction

Human designed protocols (e.g. TCP) do not adopt well to new technologies and circumstances. In this work, we devise a method to auto-generate congestion control protocols using Reinforcement Learning and Unsupervised Learning. Previously, [1, 2] proposed a learning method for this purpose but their method is computationally intensive. In our work, we develop a method to improve the run time.

Problem Setup

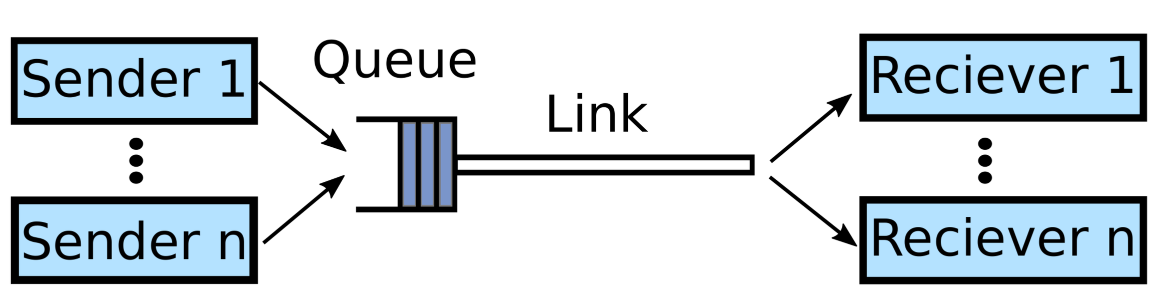


Figure: Dumbell Network

The problem is reformulated as a decision process:

- states: (s_ewma,r_ewma,RTT)
  - s_ewma: packet inter-transmit time
  - r_ewma: ACK inter-transmit time
  - RTT: round travel time
- action: (m,b,τ) (each action is a policy for a sender)
  - m: multiplier to current window
  - b: increment to current window
  - τ: minimum inter-send time
- objective: Finding a mapping μ of states to actions in order to maximize the utility function:

$$\max \sum_{user\ i} \log(th_i(\mu)) - \beta \log(d_i(\mu)) \qquad (1)$$

  - $th_i(\mu)$: average throughput of user i
  - $d_i(\mu)$: average delay of user i

Previous Methodology (Remy)

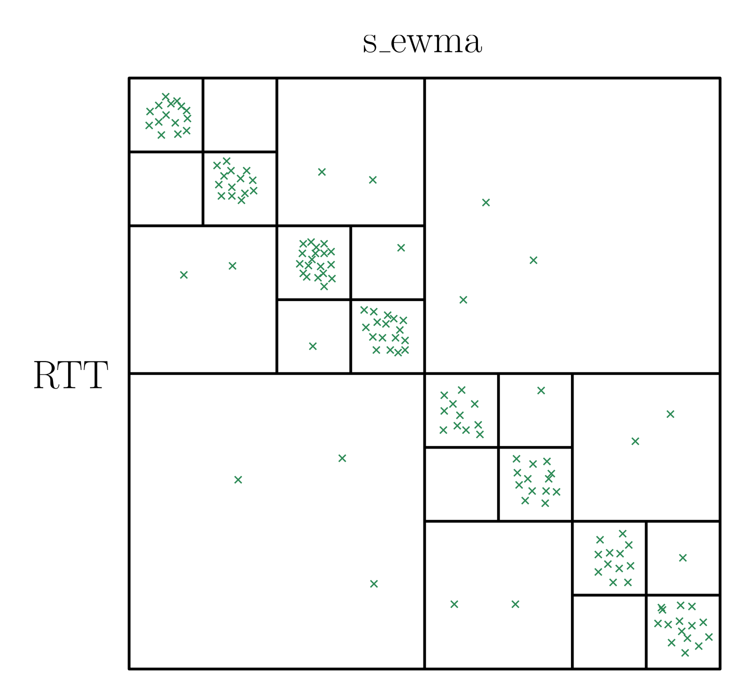
The mapping assigns a policy to regions of the state space. The method is run in iterations. In each iteration:

- The network is simulated to collect statistics on states occurrences.
- The most likely state space region is sliced based on median query of marginal distribution of state parameters. The region is divided into $2^{\#features}$ subregions.
- The optimal mapping is searched using coordinate descent.

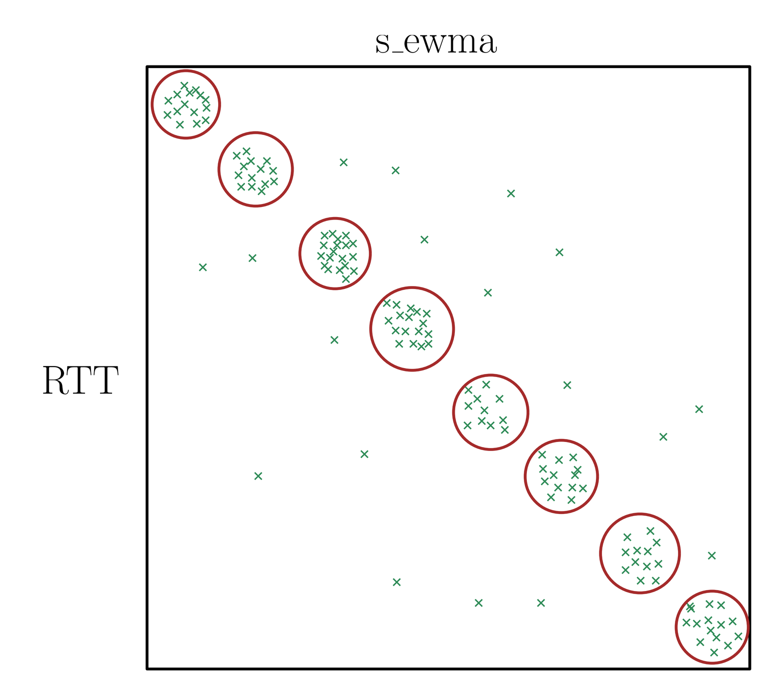
Our Methodology

The mapping comprises of a set of center points that have assigned policies. For each state value, the policy of the closest center point is assigned. The method is run in iterations and in each run a new center point is added. In the m-th iteration:

- Simulate the network with the mapping from the previous iteration and collect statistics on state occurrences.
- Perform k-means clustering with m+1 centers.
- The optimal mapping is searched using coordinate descent. As the initial mapping, use the assignment from the mapping of the previous iteration.

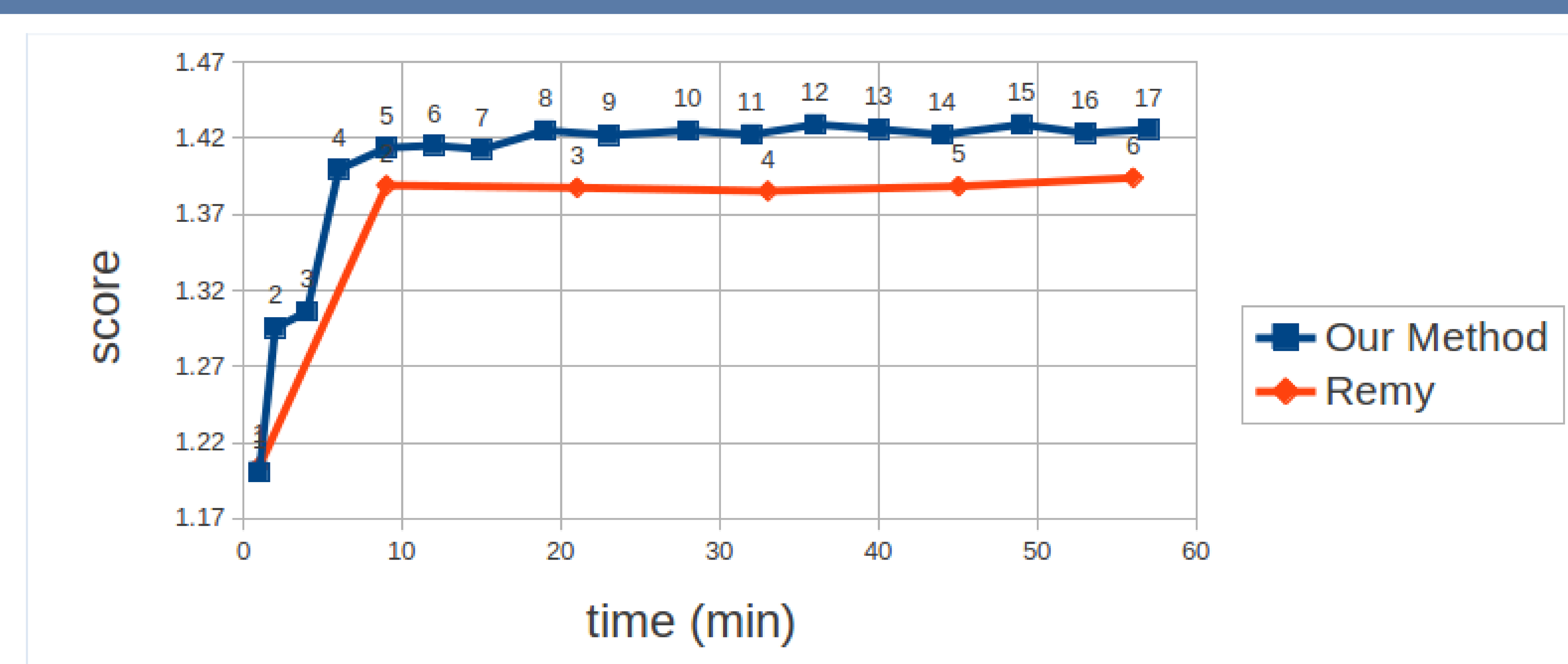


(a) Remy



(b) Our Method

Result



Discussion

Benefits over previous method (Remy):

- The number of rules added per iteration is one and independent of features whereas in the previous method it grows exponentially with the number of features.
- It is able to exploit the correlation between the features while the previous method only considers the marginal probabilities of each feature.

Conclusion

- Proposed and implemented a learning method for congestion control using k-means clustering
- Through simulation, showed our method outperforms Remy both in score and running time.

References

[1] K. Winstein and H. Balakrishnan, "Tcp ex machina: computer-generated congestion control," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 123–134.

[2] A. Sivaraman, K. Winstein, P. Thaker, and H. Balakrishnan, "An experimental study of the learnability of congestion control," in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4. ACM, 2014, pp. 479–490.

Stanford University