Learning to Optimize a Network Overlay Router

Karen Zita Haigh
Raytheon BBN Technologies
10 Moulton St, Cambridge, MA 02138
karen.haigh@raytheon.com

We present our Strategy Optimizer that learns how to configure a network node to optimize performance. The Strategy Optimizer selects a configuration in real-time, and learns on-the-fly during a mission. The Strategy Optimizer has a rapid decision-making module that selects the configurations, and a slower learning loop that updates the models as it encounters new environmental conditions.

1 Introduction

Mobile ad hoc networks (MANETs) operate in highly dynamic, potentially hostile environments. Current approaches to network configuration tend to be static, and therefore perform poorly. It is instead desirable to adaptively configure the radio and network stack to maintain consistent communications. Our goal is to automatically recognize conditions that affect communications quality, and select a configuration that improves performance, even in highly-dynamic missions.

This domain requires the ability for a decision maker to *select a configuration in real-time*, within the decision-making loop of the radio and IP stack. A human is unable to perform this dynamic configuration because of the rapid timescales and the exponential number of configurations.

This domain also requires the ability to *learn on the fly during a mission*, for example to recognize changes to the RF environment, or to recognize when components have failed. The system must learn new models at runtime that accurately describe new communications environments. A real-time decision maker then uses these newly learned models to make decisions.

This paper describes how we use Machine Learning (ML) to configure nodes in AORTA, the Adaptive Overlay and Routing Technologies for Airborne Networks. The ML-based Strategy Optimizer builds on our prior efforts in cognitive RF and networking [2, 4, 7]. The Strategy Optimizer has a rapid decision-making module that selects the configurations, and a slower learning loop that updates the models as it encounters new environmental conditions. It uses Support Vector Machines (SVMs) as the learning approach [8, 9].

2 Communications Domain

Our target domain is a communications controller that automatically learns the relationships among configuration parameters of a MANET to maintain near-optimal configurations automatically in highly dynamic environments. Consider a MANET with N nodes; each node has

- a set of observable parameters o that describe the environment, including RF observations such as signal-to-noise ratio and gaussianness of detected emitters, and waveform statistics such as error rates and retransmission statistics;
- a set of controllable parameters c that can use to change its behavior, including those in the RF hardware, the FPGAs, and the IP stack; and
- a set of metrics m that provide feedback on how well it is doing, including measures of effectiveness and cost, and an approach on combining metrics, e.g. weighted sum.

Each control parameter has a known set of discrete values. We denote a *Strategy* as a combination of control parameters (CPs). The maximum number of strategies is $\Pi_{\forall c} v_c$, where v_c is the number of possible values for the controllable parameter c; if all n CPs are binary on/off, then there are 2^n strategies, well beyond the ability of a human to manage. The goal is to have each node choose its strategy s, as a combination

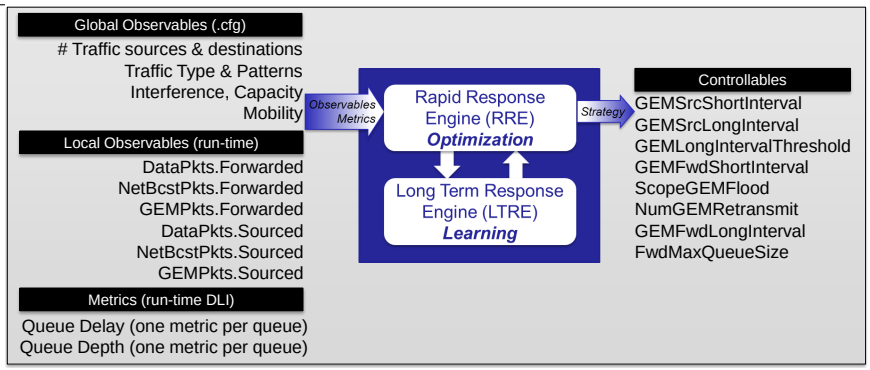

Figure 1. AORTA's Strategy Optimizer focuses on learning to control the overlay routing layer, adjusting timers and flooding parameters.

of controllables c, to maximize performance of each metric m, by learning SVM models f that predict performance of each metric from the observables and strategy: $m = f(o, s)$. The mathematics of this domain is described in more detail elsewhere [3, 4].

In AORTA, we focus on learning within the overlay routing layer, adjusting timers and flooding parameters. Our prior efforts have demonstrated effectiveness at adjusting RF front end, PHY and MAC parameters [2] through routing and application layer paremeters [4, 7]; in these efforts, control parameters were local to a single node or required coordination with only a single neighbour. AORTA will extend these efforts by adding an explicit coordination step among nodes across the network. Figure 1 shows the current set of observables, metrics and controllables within the context of the Strategy Optimizer's architecture.

3 Architecture

The Strategy Optimizer comprises two main modules, the *Rapid Response Engine (RRE)* and *Long Term Response Engine (LTRE)*, as illustrated in Figure 1. The RRE selects the best strategy based on SVM predictions, and if prediction error has exceeded a configured error threshold, instructs the LTRE to retrain on all collected data. To select the best strategy, the RRE first uses the previously-learned SVM models to predict the performance of each candidate strategy against each metric in the current environment: $\forall m \; in \mathcal{M}, \forall s \in \mathcal{S}, m_s = \text{SVM}(o, s)$. The RRE then combines the metrics using a weighted sum, and chooses the strategy s that optimizes performance.

The LTRE updates the learned models on the fly during a mission. This capability enables the Strategy Optimizer to handle new conditions such as new interference sources, or when a technique is no longer operating effectively (e.g., a component has failed). The LTRE manages the dataset, and builds the SVM models from previous observations to send to the RRE. The LTRE limits redundancy in the dataset and removes old instances if memory is limited (e.g. on an embedded system).

AORTA operates within the Extendable Mobile Ad-hoc Network Emulator (EMANE) [1]; each node operates its own Strategy Optimizer in a separate executable that communicates to the AORTA node through UDP sockets. Figure 2 illustrates the sequence of events on each node, for every observation made in the AORTA. Because the Strategy Optimizer was developed and demonstrated in an embedded system, a *Bridge* reshapes the incoming data to `int8` format to emulate the data from an FGPA, and then transforms the decision into a `.cfg` file that the AORTA node will reread.

4 Results

The results in Figure 3 highlight the effectiveness of the Strategy Optimizer. In this scenario, a set of mobile sensor nodes collect data over a Link16 waveform with contention. The forward node transmits data to the other nodes, and intermediate nodes forward information to trailing nodes.

To evaluate performance, we examine whether the Strategy Optimizer chose the best strategy for each operating environment. We use Gaussian-Mixture models to cluster the observables into environments [6]; Figure 3 shows two of these environments.

Cluster 1 (row 1) represents nodes that generate original traffic, and the combination of delay and queue depth is (a) highly variable and likely somewhat unpredictable, and (b) relatively independent of the chosen

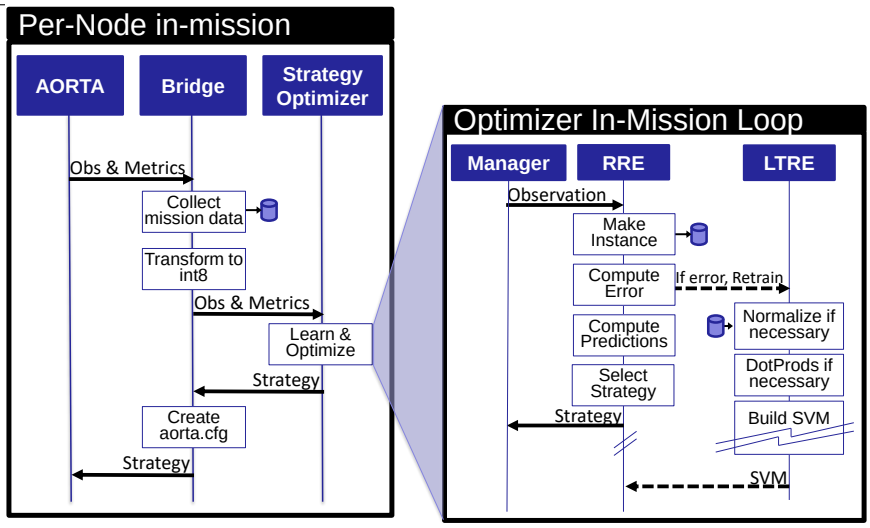

Figure 2. Operating within EMANE, AORTA's nodes invoke an executable for the bridge and the Strategy Optimizer, and then communicate through UDP sockets. The Strategy Optimizer uses threads for each of its internal modules.

strategy. Boxplots with yellow background indicate those that are similar to the strategy with the best mean with 95% confidence. Despite the prediction challenge, the Strategy Optimizer chooses strategies that are very similar to the best values (p=0.26 in a paired-ttest [5]), while extremely different from the broader dataset (p=6.0 x 10^{-9}).

Cluster 2 represents nodes that forward traffic, and several strategies are clearly better choices than other strategies. The Strategy Optimizer chooses clearly more similar to the best strategies (p=0.18), and dissimilar to the broader possibilities (p=4.9 x 10^{-13}).

5 Conclusion

This paper described our effort to use Machine Learning to optimally configure a MANET. Our Strategy Optimizer has a rapid decision-making loop that selects a configuration in real-time to optimize performance of the network as conditions change. The Strategy Optimizer also has a slower learning loop that updates the prediction models as the system encounters novel conditions.

In prior work [2], we show that the Strategy Optimizer can effectively learn to configure nodes even with no prior training data, and only small numbers of examples (< 5) are sufficient to model a given environment. The RRE of the embedded system operates within 1ms, and the LTRE can generate a new SVM model in under 2ms.

Acknowledgments

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] The extendable mobile ad-hoc network emulator (EMANE). https://www.nrl.navy.mil/itd/ncs/products/emane.

[2] K. Z. Haigh, A. M. Mackay, M. R. Cook, and L. G. Lin. Parallel learning and decision making for a smart embedded communications platform. Technical Report BBN REPORT 8579, BBN Technologies, August 2015. http://www.cs.cmu.edu/~khaigh/papers/2015-HaighTechReport-SO.sm.pdf.

[3] K. Z. Haigh, O. Olofinboba, and C. Y. Tang. Designing an implementable user-oriented objective function for MANETs. In *IEEE International Conference On Networking, Sensing and Control*, pages 693–698, London, U.K., April 2007. (New York, NY: IEEE Press). http://www.cs.cmu.edu/~khaigh/papers/Haigh07-ICNSC.pdf.

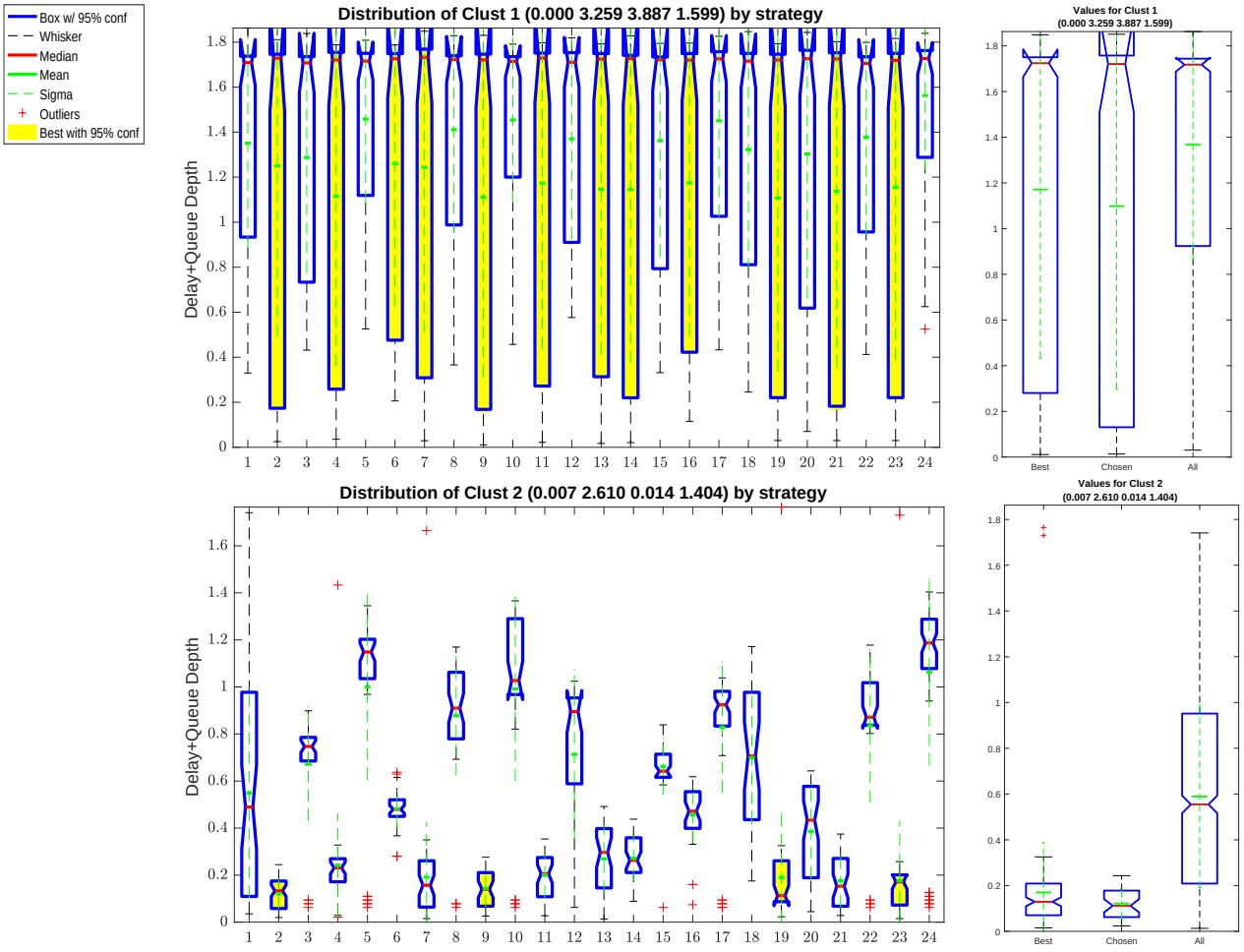

Figure 3. The Strategy Optimizer learns how the different strategies perform for each operating environment, and chooses strategies that optimize performance. *Rows:* Operating environments; *Left column:* Expected performance per strategy, yellow indicates the best performing strategies; *Right column:* Performance of the the best strategies, the strategies chosen by the learner, and all possible strategies.

[4] K. Z. Haigh, S. Varadarajan, and C. Y. Tang. Automatic learning-based MANET cross-layer parameter configuration. In *Workshop on Wireless Ad hoc and Sensor Networks (WWASN2006)*, Lisbon, Portugal, 2006. (New York, NY: ACM Press). http://www.cs.cmu.edu/∼khaigh/papers/haigh06a-configuration.pdf.

[5] Mathworks. Two-sample t-test. https://www.mathworks.com/help/stats/ttest2.html.

[6] Mathworks. Gaussian Mixture Models, 2013. http://www.mathworks.com/help/stats/gmdistribution class.html and http://www.mathworks.com/help/stats/gaussian-mixture-models.html.

[7] G. D. Troxel, A. Caro, I. Castineyra, N. Goffee, K. Z. Haigh, T. Hussain, V. Kawadia, P. G. Rubel, and D. Wiggins. Cognitive adaptation for teams in ADROIT. In *IEEE Global Communications Conference*, pages 4868–4872, Washington, DC, November 2007. (New York, NY: IEEE Press). Invited. http://www.cs.cmu.edu/∼khaigh/papers/troxel07-globecom.pdf.

[8] B. Üstün, W. J. Melssen, and L. M. C. Buydens. Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel. *Chemometrics and Intelligent Laboratory Systems*, 81(1):29–40, March 2006. http://www.sciencedirect.com/science/article/pii/S0169743905001474.

[9] V. N. Vapnik. *The Nature of Statistical Learning Theory*. (New York, NY: Springer), 1995.