CS557: Final Project Proposal

For Dr Joseph Gersch
By Vignesh M. Pagadala
Vignesh.Pagadala@ColoState.Edu

Title: An Investigation into the use of Reinforcement Learning for Enhanced Queue Stability in RED Gateways

Abstract:

Despite being a well-established congestion-avoidance scheme in packet-switched networks, there happens to be a general reluctance amongst network administrators in implementing Random Early Detection (RED) in their network gateways. This is due to the presence of several adjustable parameters associated with RED, whose values are to be determined by the network administrators themselves. There happens to be no known methods which espouse on how to manually set these values to achieve peak network performance. To circumvent this problem, several studies have relied on automating this procedure, using feedback about the performance over discretized time intervals, and information regarding the state of the network to arrive at an optimum solution. In this work we shall adopt a reinforcement-learning-based approach to arrive at the optimal values of the control parameters. We shall describe in detail the state-description of the network, the features against which we reinforce, and then use simulations to examine the results.

Objectives:

The first important step would obviously be to perform thorough background analysis, through literature review of existing RED implementations which are adaptive in nature. After this is accomplished, the following modular goals will have to achieved.

1. Control Parameters: The first step is to determine the control parameters, whose values we desire to get the optimum performance. Through initial survey, I have identified the following parameters:

- p(x, y, z, ...) - Probability function which determines the random drop probability of a given packet. x, y, z etc. are the parameters upon which the probability should may depend on.
- MaxP – Maximum drop probability possible.
- Qmax – Maximum queue length threshold.
- Qmin – Minimum queue length threshold
- w – Weight associated with the queue (for calculating average queue length)

These are the parameters I have identified. It is possible that more parameters might exist, which when tweaked, positively influence the network performance.

2. Queue Length Calculation Method: This is one of the most important aspects which might affect the RED algorithm. There are multiple ways by which we could calculate the queue length. In this paper, I intend to investigate several different methods proposed in existing literature, and determine which of them appears best through experimental analysis. A few methodologies which I will investigate are listed below:

- Instantaneous queue value
- Exponential Weighted Moving Average
- Discreet-Time Feedback

3. Performance Measure: The next important question to be answered is, what would be a good performance measure which would accurately reflect network performance. Through background analysis, I have determined

that the stability of average queue length would ideally serve this purpose. So, we would essentially be solving the optimization problem, where we choose to minimize Var(Qavg), subject to the control parameter variables mentioned above, where Var(Qavg) represents the variance of several measures of average queue length calculated over several, fixed, discreet time-intervals. This optimization problem shall be delegated to reinforcement learning.

4. Presentation of Results: The results will be presented through simulations. For performing simulations, I shall be making use of the network simulation package put together by Grotto Networking (Source: https://www.grotto-networking.com/DiscreteEventPython.html), based on the Python package SimPy. I also intend to use MatPlotLib to plot various curves, which would demonstrate the obtained results to the reader in a neat, understandable manner.

Project Milestones:

Week 1 (Nov 2nd to 9th): Finish and solidify background analysis. Read existing literature, and gain understanding of the methodologies used, and the significance of the final results. Play around with SimPy, and gain an understanding of how the network package works.

Week 2 (Nov 9th to 16th): Formulate a simple simulation – implement RED with tail-drop. After this, try mirroring experiments done in previous papers. Compare results. Perform this incrementally.

Week 3 (Nov 16th to 23rd): Experiment with different methodologies, and finalize the method to be used. This would entail answers to each of the objectives listed above. Get experimental results, plots, graphs, figures etc.

Week 4 (Nov 23rd to 30th): Prepare the write-up.

Week 5 (Dec 1st to 4th): Refine the paper.

December 4th : Submit.

References:

1. Floyd, Sally, and Van Jacobson. ”Random early detection gateways for congestion avoidance.” IEEE/ACM Transactions on networking 1.4 (1993): 397-413.
2. Jacobson, Van, K. Nichols, and K. Poduri. ”RED in a Different Light.” Draft, Cisco Systems (1999).
3. Lin, Dong, and Robert Morris. ”Dynamics of random early detection.” ACM SIGCOMM Computer Communication Review. Vol. 27. No. 4. ACM, 1997.
4. Lyon, Norman A., and Todd D. Morris. ”Method and apparatus for red (random early detection) and enhancements.” U.S. Patent No. 6,333,917. 25 Dec. 2001.
5. Misra, Sudip, et al. ”Random early detection for congestion avoidance in wired networks: a discretized pursuit learning-automata-like solution.” (2010).
6. Freed, Michael, and Satish Kumar Amara. ”Policy-based weighted random early detection method for avoiding congestion in internet traffic.” U.S. Patent No. 6,996,062. 7 Feb. 2006.
7. Masoumzadeh, Seyed Saeid, et al. ”FQLRED: an adaptive scalable schema for active queue management.” International Journal of Network Management 21.2 (2011): 147-167.
8. Zhani, Mohamed Faten, Halima Elbiaze, and Farouk Kamoun. ”SNFAQM: an active queue management mechanism using neurofuzzy prediction.” Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on. IEEE, 2007.