On Designing Improved Controllers for AQM Routers Supporting TCP flows

By C.V Hollot, Vishal Mishra, Don Towsley and Wei-Bo Gong

Presented by
Pushkaraj Chitre
Meganne Atkins
Outline

• Introduction
• Background
• The Proportional Controller
  • Experiments
  • Limitation
• PI Controller
  • Experiments
  • The Delay Utilization Trade-Off
• Conclusion and Future Work
Introduction

• Uses Classical Control System Techniques for developing controllers.

• Proposes 2 designs
  – Proportional Controller
  – Proportional Integrator Control

Uses NS-2 Simulations
Performed control theoretic analysis of RED
• 2 limitations of RED:-
  – Compromise speed for stability and vice versa
  – Direct coupling between queue length and loss probability
Background

- Linearized the TCP model
P(s) = P_{TCP}(s)P_{Queue}(s)

\[ P_{tcp}(s) = \frac{\frac{R_0C^2}{2N^2}}{s + \frac{2N}{R_0^2C}}; \]

\[ P_{queue}(s) = \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}}. \]

R_0 = Round Trip Time at the operating point

C = Link Capacity (packets/sec)

N = Load Factor (No of Connections)
\[ C'(s) = C_{red}(s) = \frac{L_{red}}{s/K + 1} \]

packet-marking profile

averaging filter

Fig. 3. RED as a cascade of low-pass filter and nonlinear gain element.
The Proportional Controller

\[ \omega_g = 0.1 \min(p_{tcp}, p_{queue}) \]

- Lag in the low pass filter responsible for the sluggishness of the RED controller
- Not replacing the low pass filter by proportional controller, the authors suggest designing of the stabilizing controller.
• **Design:-**
  
  • \( K = \infty \)
  
  \[
  L_{\text{red}} = \left| \frac{(\frac{j\omega_g}{P_{\text{rep}}} + 1)}{(\frac{j\omega_g}{P_{\text{queue}}} + 1)} \right| \frac{(r+C)^3}{(2N-C)^3}
  \]

  \[
  \omega_g = \sqrt{\frac{P_{\text{rep}} P_{\text{queue}}}{r+C}} = \sqrt{\frac{2N-C}{r+3C}}
  \]

  • \( \omega_g \approx 1.5 \text{ rad/sec} \)
  
  • Note: the values are calculated in the Control Theoretic analysis of RED”
Experiments with proportional controller

- X-axis->time(sec)
- Y-axis->Queue Size(packets)

Experiment 1:-
- 60 FTP flows
- 180 http sessions
- Link bandwidth=15Mb/s
- Added time-varying dynamics
- Buffer size=800 packets
Comparison of RED and Proportional controller

Fig. 5. Comparison of RED and the proportional controller

Settling time

Sluggish response Of RED
• Experiment 2
  • Repeat the previous experiment by doubling Round Trip Times.

Overshoots on RED
Proportional controller with high gain
Limitations of Proportional Controller

- For stable operation, a relatively shallow slope in the loss profile required.
- Reason-coupling between queue size and marking probability
- Solution – decouple by using integral control
- Steady state error
Solution to limitations

- Use of proportional Integrator Controller
  - Steady state error=0
  - Can clamp queue size ro reference value “$q_{ref}$”
  - Much higher loop bandwidth=faster response
The Proportional Integrator (PI) Controller

Higher loop bandwidth = faster response time
C(s) = K_{PI} \frac{(s/z + 1)}{s}
Digital Implementation

Difference Equation:

\[ p(kT) = a\delta q(kT) - b\delta q((k - 1)T) + p((k - 1)T) \]

Pseudo Code:

\[
\begin{align*}
p &:= a*(q - q_{\text{ref}}) - b*(q_{\text{old}} - q_{\text{ref}}) + p_{\text{old}} \\
p_{\text{old}} &:= p \\
q_{\text{old}} &:= q
\end{align*}
\]

q_{\text{ref}} = desired queue length
Experiment Tools & Parameters

- Used *ns* simulator
- Sampling frequency of 160 Hz
- PI coefficients
  - $a = 1.822 \times 10^{-5}$
  - $b = 1.816 \times 10^{-5}$
- $q_{\text{ref}} = 200$ packets
- Buffer = 800 packets

\[ p(kT) = a \delta q(kT) - b \delta q((k - 1)T) + p((k - 1)T) \]
Experiment 3

- Faster response time
- Regulation of output
- PI Controller insensitive to load level variations

PI Controller regulates the queue length to 200 packets
Experiment 4

- Faster response time

Fig. 11. Experiment 4
Experiment 5

- PI controller settles at ~10 milliseconds
- RED settles at ~ 115 milliseconds

PI more robust at higher work loads
Experiment 6

- RED experiences oscillations

**PI still stable at lower work loads**
Experiment 7

- PI controller is still at acceptable performance
- Response time has slowed (~ 40 milliseconds)
- RED and Proportional Controller “hit the roof”

AQM system (with finite buffer) needs integral control

10/2/2007
The RED controller's steady state error has increased due to:

- Shorter RTT
- Operating Point Queue Length Higher
The Delay Utilization Tradeoff

- Large buffers lead to:
  - Higher utilization of the link
  - Larger queueing delays

- In RED:
  - $\min_{th}$
  - $\max_{th}$
  - $p_{max}$

- $q_0$ in the PI Controller controls the delay

Larger values of $q_0$ = larger delays and utilization
Delay Utilization Tradeoff

For (nearly) full utilization:
- Small $q_0$ for FTP ONLY
- Large $q_0$ for Mix (FTP/http)

Nearly linear relationship between $q_0$ and delays

Fig. 16. Utilization versus operating point $q_0$. PI controller.

Fig. 17. Queuing delay versus operating point $q_0$. PI controller.

Fig. 18. Utilization versus queueing delay. PI controller.
Delay in RED controlled by $min_{th}$

To dynamic ranges $(max_{th} - min_{th})$ used for RED:
- Fig. 19 used 550
- Fig. 20 used 55

Mixed flows were used

PI Controller capable of handling low delay and high utilization

10/2/2007
The Importance of ECN

- PI Controller can regulate the queue to a low level
  - Lower Delay
  - Less efficient performance
- Dropping packets leads to higher transmission completion time

AQM used with ECN produces an almost lossless system
Conclusions

• Two controllers:
  – Proportional
    • Simple to implement
    • AQM response time better than REDs
  – Proportional Integrator
    • Improves network performance
    • AQM response time better than REDs
    • Able to handle and regulate queue level

• Objectives:
  – Queue Usage
  – Latency Reduction

• PI Controller outperformed RED

10/2/2007
Limitations and Future Work

• Limitations
  – Used linear models
  – Focused on classical control methods
  – Did not look at global or optimal results

• Future Work
  – More complex controllers
THANK YOU!