

# IETF-87 AQM BoF

<http://www.ietf.org/mail-archive/web/aqm/current/maillist.html>

Wesley Eddy <[wes@mti-systems.com](mailto:wes@mti-systems.com)>  
Richard Scheffenegger <[rs@netapp.com](mailto:rs@netapp.com)>

Tue., 30. July 2013  
17:00, Potsdam 1 Room

# Introduction

- **The Active Queue Management and Packet Scheduling working group (AQM) works on algorithms for managing queues in order to minimize standing queues, help control the sending rates without undue losses, minimize delays for interactive apps, and protect flows from misbehaving flows.**

# Background

- There is a desire to update the RED manifesto based on “lessons learned”:
  - <http://tools.ietf.org/html/draft-baker-aqm-recommendation>
- There are new AQM algorithms being defined, which should improve on RED both in operation (improved performance) and operability (reduced tuning):
  - <http://tools.ietf.org/html/draft-pan-tsvwg-pie>
  - <http://tools.ietf.org/html/draft-nichols-tsvwg-codel>

# Problem Statement

- Bufferbloat exists in routers, lower-layer switches, end-hosts, and other middleboxes (in hardware, drivers, and software)
  - Problem mainly shows in access and edge equipment, but you can find it almost everywhere
- Absorbing bursts is good; causing undue delay and jitter is bad

# Benefits of AQM

- AQM and traffic isolation can:
  1. help flow sources control their sending rates before the onset of necessary losses, e.g. through ECN
  2. help minimize delays for interactive applications
  3. help protect flows from negative impacts of other more aggressive or misbehaving flows
  4. counteract global synchronization between competing flows

# Desired Outputs

- Informational and Best Current Practices documents that cover the design, use, and configuration of algorithms for managing queues in Internet devices and software.
- Algorithm specifications that are found to be broadly applicable and beneficial

# IETF-87 AQM BoF Agenda

Topics	Speaker	Time
Introduction & Background	Chairs	17:00
Recommendations	Fred Baker (Cisco)	17:05
PIE (Proportional Integral Controller Enhanced)	Rong Pan (Cisco)	17:15
[FQ-]CoDel	Andrew McGregor (Google)	17:30
Algorithm discussion	Group	17:45
BoF Questions	Chairs	18:00
Adjourn BoF	Chairs	<18:30

# Algorithm discussion

## Similarities

- Use delay rather than occupancy
- Minimal tunable parameters
- Permit high link utilization
- Intend to permit efficient implementations
- Both are promising
  - Better than drop tail, RED
- Incremental deployment

## Differences

- PIE
  - Drop before enqueue
  - Compute drop rate from departure rate and queue length
  - Decoupled from FQ/CBQ implementation
- FQ-CoDel
  - Drop at Dequeue
  - Drop based on inferring a „bad“ standing queue
  - Recent CoDel work includes emphasis on integrating FQ/SFQ aspects with the AQM



# Impact to Vendors

- RED is implemented today
  - May not be used often; only performs as well as it's been tuned
  - Not viewed as effective general solution to bufferbloat due to necessity of configuration tuning
  - New algorithms aim to involve less tuning; vendors will need to implement one or more new algorithms to benefit from them
- In designing the new algorithms, implementability is a major goal
  - Should take existing architectures into account, though may involve updates to hardware/firmware
  - Where the queues are (ingress or egress) and where the computation is done or the drops/ECN are performed is important
    - Fred Baker example: Cisco GSR did not implement ECN because queue was on ingress but RED implementation was on the egress