Workgroup: Transport Working Group
Internet-Draft:
draft-morton-tsvwg-codel-approx-fair-01
Published: 9 March 2020
Intended Status: Informational
Expires: 10 September 2020
Authors: J. Morton   P. Heist

Controlled Delay Approximate Fairness AQM

Abstract

This note presents CodelAF, or Controlled Delay Approximate Fairness in full, as an alternative to single-queue AQM or Fair Queue implementations in the low-cost or high-speed network hardware spaces. It builds on the seminal work in Codel [RFC8289], and guides multiple competing flows towards similar throughputs by differential congestion signalling, whilst requiring only a single FIFO queue. It may also be combined with CNQ [I-D.morton-tsvwg-cheap-nasty-queueing] to provide a latency optimisation for sparse flows.

Status of This Memo

This Internet-Draft will expire on 10 September 2020.

Copyright Notice

Table of Contents

1. Introduction

For some years, the solution of choice for improving network performance as been the combination of Fair Queuing (FQ) with Active Queue Management (AQM) as demonstrated in FQ-Codel [RFC8290]. However, concerns are legitimately raised over the difficulty of implementing FQ in hardware, making it a weak proposition for very low-cost and very high-speed network devices alike. There is some evidence to suggest that implementing multiple AQM instances is not very difficult in hardware, but implementing multiple FIFOs can be prohibitive.

CodelAF addresses this design space with a straightforward extension to the Codel AQM, allowing its target to be biased according to relative queue occupancy of a particular flow, and its signals applied only to that flow. An arbitrary number of independent flows can then be signalled to more independently than a single AQM can, allowing convergence towards a fair-throughput state.

This approach also successfully addresses the problem of allowing flows responding to dissimilar congestion signals to share the same FIFO queue without excessive bias. In particular, it applies to Some Congestion Experienced [I-D.morton-tsvwg-sce] flows sharing a queue with conventional ECN [RFC3168] and Not-ECT flows.

It is likely that a similar AF technique can also be applied to other AQMs that employ a target queue sojourn time, such as PIE and BLUE.

Building on the basic CodelAF algorithm, this memo also shows how to provide a low-latency PHB through a twinned CodelAF configuration, requiring configuration of only a second set of AQM parameters and retaining approximate flow-fairness between the low-latency and best-effort traffic classes.

2. Background

A brief summary of the basic Codel algorithm follows. For full details, see [RFC8289].

Codel is parameterised by a target setpoint, indicating the amount of tolerable standing queue (default 5 ms) and an initial signalling interval which is set to an estimate of the typical path latency (default 100 ms). The principle dynamic state elements are a flag indicating whether Codel is in the "marking" state or not, a timer indicating when the next mark is due, and a counter indicating how many marks have been set since entering the marking state.

Since Codel was designed at a time when ECN was not commonly used, the "marking" state is often described as the "dropping" state, including by the original authors. Here the term "marking" state is used to match the increased deployment of ECN today.

Codel enters the "marking" state when the sojourn time of a packet within its queue first exceeds its target setpoint. At this time, the counter is initialised to 1 and the timer is set for interval/sqrt(counter) time in the future. This first packet, therefore, is not marked, as it may be an outlier belonging to an isolated and temporary burst of traffic. Only if the sojourn times of all subsequent packets (until the timer expires) also exceed the target will ECN marking (or dropping of Not-ECT packets) begin. Marking is always performed at the head of the queue, where the sojourn time of individual packets is precisely known.

After each mark (or drop), the counter is incremented and the timer advanced, again, by interval/sqrt(counter). This causes a linear increase of marking frequency over time, until the queue is brought under control. This is signified by the sojourn times of packets dropping below the target, at which time marking immediately stops and Codel exits the marking state.

When Codel exits the marking state, the counter is not immediately reset, as further control of an aggressive flow may still be needed. The reference implementation pauses for some multiple of the interval and then resets the counter. The COBALT variant instead decrements the counter and resets the timer on the same linear frequency ramp, run in reverse, the benefit of which can be seen in [COBALT].

The reference CodelAF implementation is built around a combination of COBALT with CNQ [I-D.morton-tsvwg-cheap-nasty-queueing], to which only small code changes were required.

3. The Codel Approximate Fairness Algorithm

In CodelAF, a separate instance of the Codel state variables (marking flag, timer, and counter) are kept for each flow. In addition, an account of the instantaneous queue occupancy of each flow is maintained, as well as the total queue occupancy, and the number of "active flows" which have traffic in the queue.

Flows may be distinguished by whichever means is convenient, for example a hash function over the traditional 5-tuple of protocol number, source/destination addresses and port numbers. Some deployments may prefer to use a smaller set of packet header information, or to distinguish based on subscriber ID metadata. The result in any case is an index into a flow table containing the queue occupancy data and AQM state mentioned above.

The Codel parameters (interval, target) are common to all flows. However, when evaluating the AQM state for a packet, the target parameter is locally adjusted based on the actual queue occupancy by that packet's flow, compared to the fair-share queue occupancy based on dividing the total occupancy between all active flows. Hence a sparser flow, with lower than average occupancy, will receive more leniency from the AQM.

The basic Codel criterion:

```
if(sojourn > target):
        enter_dropping_state;
```

becomes:

```
if(sojourn * flow occupancy * active flows > target * total occupancy):
        enter_dropping_state;
```

This is sufficient to guide flows that are responsive to AQM signals towards throughput fairness.

4. Extending CodelAF to Provide a Low Latency PHB

The Internet is a highly heterogeneous environment, with path lengths as short as single-digit milliseconds on some paths, and approaching a full second on others. An AQM is thus set for a reasonable compromise corresponding to a "typical" path length; in the case of Codel and CodelAF, this is 100ms RTT, which works well on transcontinental and inter-continental paths, and also has acceptable behaviour on shorter paths for many applications.

However, better control of latency may be desired for traffic known to be on such a short path, eg. between an end-user and a Content Distribution Network (CDN) or gaming service local to that end-user. This requires that a second queue and AQM is selectable by some classifier, such as a Diffserv codepoint (DSCP) [RFC2475][RFC7657] [RFC8100], and tuned for the shorter path length.

A perennial concern with Diffserv deployment is ensuring that traffic originators are not incentivised to mis-mark their traffic by, for example, obtaining an unreasonable throughput increase at the expense of traffic legitimately marked one way or the other. The configuration described here addresses this concern by ensuring that throughput is controlled in a flow-fair manner between the classes, as well as within them. Hence there is no unfair throughput benefit from selecting the low-latency class, while the more severe AQM action will encourage long-path flows to select the more appropriate default class. Hence marking incentives are properly aligned with the intent of the PHB.

Two complete CodelAF instances are provided, the ensemble being referred to as Twin-CodelAF. Packets are simply enqueued into one of the two instances, depending on whether the classifier matches the configured value(s) or not. Admission control of any kind is not necessary. The "low latency" instance is configured for the expected path RTT of suitably marked traffic, while the "default" instance remains configured for a general Internet path RTT.

Because CodelAF keeps track of the number of active flows, it is then straightforward to perform Weighted Round Robin (WRR) between the two instances on dequeue, with the weight of each instance corresponding to the number of active flows in each. This is the mechanism which enforces flow-fairness between the classes.

```
if(only one queue contains packets):
        deliver from that queue;
else
        deliver from queue with lowest deficit;

deficit of delivered queue += active flows of other queue;
deficit of both queues -= min(deficits);
```

5. Security Considerations

No particular security concerns are anticipated.

6. IANA Considerations

There are no IANA considerations.

7. Informative References

[RFC8100]
Geib, R., Ed. and D. Black, "Diffserv-Interconnection Classes and Practice", RFC 8100, DOI 10.17487/RFC8100, March 2017, <https://www.rfc-editor.org/info/rfc8100>.

[I-D.morton-tsvwg-cheap-nasty-queueing]
Morton, J. and P. Heist, "Cheap Nasty Queueing", Work in Progress, Internet-Draft, draft-morton-tsvwg-cheap-nasty-queueing-01, 4 November 2019, <https://tools.ietf.org/html/draft-morton-tsvwg-cheap-nasty-queueing-01>.

[RFC8290] Hoeiland-Joergensen, T., McKenney, P., Taht, D., Gettys, J., and E. Dumazet, "The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm", RFC 8290, DOI 10.17487/RFC8290, January 2018, <https://www.rfc-editor.org/info/rfc8290>.

[RFC7657] Black, D., Ed. and P. Jones, "Differentiated Services (Diffserv) and Real-Time Communication", RFC 7657, DOI 10.17487/RFC7657, November 2015, <https://www.rfc-editor.org/info/rfc7657>.

[COBALT] Palmei, J., Gupta, S., Imputato, P., Morton, J., Tahiliani, M.P., Avallone, S., and D. Taht, "Design and Evaluation of COBALT Queue Discipline", September 2019, <https://ieeexplore.ieee.org/abstract/document/8847054>.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <https://www.rfc-editor.org/info/rfc2475>.

[RFC8289] Nichols, K., Jacobson, V., McGregor, A., Ed., and J. Iyengar, Ed., "Controlled Delay Active Queue Management", RFC 8289, DOI 10.17487/RFC8289, January 2018, <https://www.rfc-editor.org/info/rfc8289>.

[I-D.morton-tsvwg-sce] Morton, J. and R. Grimes, "The Some Congestion Experienced ECN Codepoint", Work in Progress, Internet-Draft, draft-morton-tsvwg-sce-01, 4 November 2019, <https://tools.ietf.org/html/draft-morton-tsvwg-sce-01>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

Authors' Addresses

Jonathan Morton
Kokkonranta 21
31520 Pitkajarvi
Finland

Phone: +358 44 927 2377
Email: chromatix99@gmail.com

Peter G. Heist
Redacted
463 11 Liberec 30
Czech Republic

Email: pete@heistp.net