

Content-Type: text/html

< draft-ietf-aqm-fq-implementation-02.txt draft-ietf-aqm-fq-implementation-04.txt >

Active Queue Management F. Baker
Internet-Draft R. Pan
Intended status: Informational Cisco Systems
Expires: November 2, 2015 May 1, 2015

Active Queue Management F. Baker
Internet-Draft R. Pan
Intended status: Informational Cisco Systems
Expires: April 24, 2016 October 22, 2015

On Queuing, Marking, and Dropping
draft-ietf-aqm-fq-implementation-02

On Queuing, Marking, and Dropping
draft-ietf-aqm-fq-implementation-04

Abstract
This note discusses implementation strategies for coupled queuing and mark/drop algorithms.

Abstract
This note discusses queuing and marking/dropping algorithms. While these algorithms may be implemented in a coupled manner, this note argues that specifications, measurements, and comparisons should decouple the different algorithms and their contributions to system behavior.

Status of This Memo
This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
This Internet-Draft will expire on November 2, 2015.

Status of This Memo
This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
This Internet-Draft will expire on April 24, 2016.

Copyright Notice
Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.
This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents

Copyright Notice
Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.
This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents

skipping to change at page 2, line 26
2.2.3. Calendar Queue Models 7
2.2.4. Work Conserving Models and Stochastic Fairness Queuing 9
2.2.5. Non Work Conserving Models and Virtual Clock 9
3. Queuing, Marking, and Dropping 10
3.1. Queuing with Tail Mark/Drop 10
3.2. Queuing with CoDel Mark/Drop 11
3.3. Queuing with RED or PIE Mark/Drop 11
4. Conclusion 12
5. IANA Considerations 12
6. Security Considerations 13
7. Acknowledgements 13
8. References 13
8.1. Normative References 13
8.2. Informative References 13
Appendix A. Change Log 15
Authors' Addresses 15
1. Introduction
In the discussion of Active Queue Management, there has been

skipping to change at page 2, line 26
2.2.3. Calendar Queue Models 7
2.2.4. Work Conserving Models and Stochastic Fairness Queuing 9
2.2.5. Non Work Conserving Models and Virtual Clock 9
3. Queuing, Marking, and Dropping 10
3.1. Queuing with Tail Mark/Drop 10
3.2. Queuing with CoDel Mark/Drop 11
3.3. Queuing with RED or PIE Mark/Drop 11
4. Conclusion 12
5. IANA Considerations 12
6. Security Considerations 12
7. Acknowledgements 13
8. References 13
8.1. Normative References 13
8.2. Informative References 13
Appendix A. Change Log 15
Authors' Addresses 15
1. Introduction
In the discussion of Active Queue Management, there has been

skipping to change at page 5, line 26
2.1.3. GPS Comparisons: unit of measurement
And finally, there is the question of what is measured for rate. If the only objective is to force packet streams to not dominate each other, it is sufficient to count packets. However, if the issue is the bit rate of an SLA, one must consider the sizes of the packets (the aggregate throughput of a flow, measured in bits or bytes). And if predictable unfairness is a consideration, the value must be weighted accordingly.
Briscoe discusses measurement in his paper on Byte and Packet Congestion Notification [RFC7141].
2.2. GPS Approximations
Carrying the matter further, a queuing algorithm may also be termed "Work Conserving" or "Non Work Conserving". A "work conserving" algorithm, by definition, is either empty, in which case no attempt is being made to dequeue data from it, or contains something, in which case it continuously tries to empty the queue. A work conserving queue that contains queued data, at an interface with a given rate, will deliver data at that rate until it empties. A non-work-conserving queue might stop delivering even though it still contains data. A common reason for doing this is to impose an artificial upper bound on a class of traffic that is lower than the rate of the underlying physical interface.

skipping to change at page 5, line 26
2.1.3. GPS Comparisons: unit of measurement
And finally, there is the question of what is measured for rate. If the only objective is to force packet streams to not dominate each other, it is sufficient to count packets. However, if the issue is the bit rate of an SLA, one must consider the sizes of the packets (the aggregate throughput of a flow, measured in bits or bytes). And if predictable unfairness is a consideration, the value must be weighted accordingly.
[RFC7141] discusses measurement.
2.2. GPS Approximations
Carrying the matter further, a queuing algorithm may also be termed "Work Conserving" or "Non Work Conserving". A queue in a "work conserving" algorithm, by definition, is either empty, in which case no attempt is being made to dequeue data from it, or contains something, in which case the algorithm continuously tries to empty the queue. A work conserving queue that contains queued data, at an interface with a given rate, will deliver data at that rate until it empties. A non-work-conserving queue might stop delivering even though it still contains data. A common reason for doing this is to impose an artificial upper bound on a class of traffic that is lower than the rate of the underlying physical interface.

classes of queuing algorithms were explored. Queuing algorithms such as SFQ, Virtual Clock, and FlowQueue-Codel [I-D.ietf-aqm-fq-codel] have value in the network, in that they delay packets to enforce a rate upper bound or to permit competing flows to compete more effectively. ECN Marking and loss are also useful signals if used in a manner that enhances TCP/SCTP operation or restrains unmanaged UDP data flows.

Conceptually, queuing algorithms and a mark/drop algorithms operate in series, as discussed in Section 3, not as a single algorithm. The observed effects differ: defensive loss protects the intermediate system and provides a signal, AQM mark/drop works to reduce mean latency, and the scheduling of flows works to modify flow interleave and acknowledgement pacing. Certain features like flow isolation are provided by fair queuing related designs, but are not the effect of the mark/drop algorithm.

There is value in implementing and coupling the operation of both queuing algorithms and queue management algorithms, and there is definitely interesting research in this area, but specifications,

classes of queuing algorithms were explored. Queuing algorithms such as SFQ, Virtual Clock, and FlowQueue-Codel [I-D.ietf-aqm-fq-codel] have value in the network, in that they delay packets to enforce a rate upper bound or to permit competing flows to compete more effectively. ECN Marking and loss are also useful signals if used in a manner that enhances TCP/SCTP operation or restrains unmanaged UDP data flows.

Conceptually, queuing algorithms and mark/drop algorithms operate in series, as discussed in Section 3, not as a single algorithm. The observed effects differ: defensive loss protects the intermediate system and provides a signal, AQM mark/drop works to reduce mean latency, and the scheduling of flows works to modify flow interleave and acknowledgement pacing. Certain features like flow isolation are provided by fair queuing related designs, but are not the effect of the mark/drop algorithm.

There is value in implementing and coupling the operation of both queuing algorithms and queue management algorithms, and there is definitely interesting research in this area, but specifications,

skipping to change at page 13, line 23

discussions in AQM, in which some have pushed an algorithm the compare to AQM marking and dropping algorithms, but which includes Flow Queuing.

8. References

8.1. Normative References

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

8.2. Informative References

[DRR] Microsoft Corporation and Washington University in St. Louis, "Efficient fair queueing using deficit round robin", ACM SIGCOMM 1995, October 1995, <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=502236>>.

[GPS] Xerox PARC, University of California, Berkeley, and Xerox PARC, "Analysis and simulation of a fair queueing algorithm", ACM SIGCOMM 1989, September 1989, <<http://blizzard.cs.uwaterloo.ca/keslav/home/Papers/data/89/fq.pdf>>.

[I-D.ietf-aqm-codel] Nichols, K., Jacobson, V., McGregor, A., and J. Jana, "Controlled Delay Active Queue Management", draft-ietf-aqm-codel-01 (work in progress), April 2015.

[I-D.ietf-aqm-fq-codel] Hoeiland-Joergensen, T., McKenney, P., Taht, D., Gettys, J., and E. Dumazet, "FlowQueue-Codel", draft-ietf-aqm-fq-codel-00 (work in progress), January 2015.

[I-D.ietf-aqm-pie] Pan, R., Natarajan, P., Baker, F., and G. White, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", draft-ietf-aqm-pie-01 (work in progress), March 2015.

[NoFair] British Telecom, "Flow rate fairness: dismantling a religion", ACM SIGCOMM 2007, April 2007, <<http://dl.acm.org/citation.cfm?id=1232926>>.

[RFC0970] Nagle, J., "On packet switches with infinite storage", RFC 970, December 1985.

[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

[RFC2990] Huston, G., "Next Steps for the IP QoS Architecture", RFC 2990, November 2000.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC3390] Allman, M., Floyd, S., and C. Partridge, "Increasing TCP's

skipping to change at page 13, line 18

discussions in AQM, in which some have pushed an algorithm the compare to AQM marking and dropping algorithms, but which includes Flow Queuing.

8. References

8.1. Normative References

[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, <<http://www.rfc-editor.org/info/rfc2475>>.

8.2. Informative References

[CalendarQueue] "Calendar queues: a fast O(1) priority queue implementation for the simulation event set problem", Communications of the ACM 1988, October 1988, <<http://www.cs.ucla.edu/~lixia/papers/90sigcomm.pdf>>.

[DRR] Microsoft Corporation and Washington University in St. Louis, "Efficient fair queueing using deficit round robin", ACM SIGCOMM 1995, October 1995, <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=502236>>.

[GPS] Xerox PARC, University of California, Berkeley, and Xerox PARC, "Analysis and simulation of a fair queueing algorithm", ACM SIGCOMM 1989, September 1989, <<http://blizzard.cs.uwaterloo.ca/keslav/home/Papers/data/89/fq.pdf>>.

[I-D.ietf-aqm-codel] Nichols, K., Jacobson, V., McGregor, A., and J. Jana, "Controlled Delay Active Queue Management", draft-ietf-aqm-codel-01 (work in progress), April 2015.

[I-D.ietf-aqm-fq-codel] Hoeiland-Joergensen, T., McKenney, P., dave.taht@gmail.com, d., Gettys, J., and E. Dumazet, "FlowQueue-Codel", draft-ietf-aqm-fq-codel-02 (work in progress), October 2015.

[I-D.ietf-aqm-pie] Pan, R., Natarajan, P., and F. Baker, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", draft-ietf-aqm-pie-02 (work in progress), August 2015.

[NoFair] British Telecom, "Flow rate fairness: dismantling a religion", ACM SIGCOMM 2007, April 2007, <<http://dl.acm.org/citation.cfm?id=1232926>>.

[RFC0970] Nagle, J., "On Packet Switches With Infinite Storage", RFC 970, DOI 10.17487/RFC0970, December 1985, <<http://www.rfc-editor.org/info/rfc970>>.

[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, DOI 10.17487/RFC2309, April 1998, <<http://www.rfc-editor.org/info/rfc2309>>.

[RFC2990] Huston, G., "Next Steps for the IP QoS Architecture", RFC 2990, DOI 10.17487/RFC2990, November 2000, <<http://www.rfc-editor.org/info/rfc2990>>.

[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, <<http://www.rfc-editor.org/info/rfc3168>>.

[RFC3390] Allman, M., Floyd, S., and C. Partridge, "Increasing TCP's

Initial Window", RFC 3390, October 2002.	Initial Window", RFC 3390, DOI 10.17487/RFC3390, October 2002, < http://www.rfc-editor.org/info/rfc3390 >.
[RFC5690] Floyd, S., Arcia, A., Ros, D., and J. Iyengar, "Adding Acknowledgement Congestion Control to TCP", RFC 5690, February 2010.	[RFC5690] Floyd, S., Arcia, A., Ros, D., and J. Iyengar, "Adding Acknowledgement Congestion Control to TCP", RFC 5690, DOI 10.17487/RFC5690, February 2010, < http://www.rfc-editor.org/info/rfc5690 >.
[RFC6057] Bastian, C., Klieber, T., Livingood, J., Mills, J., and R. Woundy, "Comcast's Protocol-Agnostic Congestion Management System", RFC 6057, December 2010.	[RFC6057] Bastian, C., Klieber, T., Livingood, J., Mills, J., and R. Woundy, "Comcast's Protocol-Agnostic Congestion Management System", RFC 6057, DOI 10.17487/RFC6057, December 2010, < http://www.rfc-editor.org/info/rfc6057 >.
[RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, August 2012.	[RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August 2012, < http://www.rfc-editor.org/info/rfc6679 >.
[RFC6928] Chu, J., Dukkupati, N., Cheng, Y., and M. Mathis, "Increasing TCP's Initial Window", RFC 6928, April 2013.	[RFC6928] Chu, J., Dukkupati, N., Cheng, Y., and M. Mathis, "Increasing TCP's Initial Window", RFC 6928, DOI 10.17487/RFC6928, April 2013, < http://www.rfc-editor.org/info/rfc6928 >.
[RFC7141] Briscoe, B. and J. Manner, "Byte and Packet Congestion Notification", BCP 41, RFC 7141, February 2014.	[RFC7141] Briscoe, B. and J. Manner, "Byte and Packet Congestion Notification", BCP 41, RFC 7141, DOI 10.17487/RFC7141, February 2014, < http://www.rfc-editor.org/info/rfc7141 >.
[SFQ] SRI International, "Stochastic Fairness Queuing", IEEE Infocom 1990, June 1990, < http://www2.rdrop.com/~paulmck/scalability/paper/sfq.2002.06.04.pdf >.	[SFQ] SRI International, "Stochastic Fairness Queuing", IEEE Infocom 1990, June 1990, < http://www2.rdrop.com/~paulmck/scalability/paper/sfq.2002.06.04.pdf >.
[VirtualClock] Xerox PARC, "Virtual Clock", ACM SIGCOMM 1990, September 1990, < http://www.cs.ucla.edu/~lixia/papers/90sigcomm.pdf >.	[VirtualClock] Xerox PARC, "Virtual Clock", ACM SIGCOMM 1990, September 1990, < http://www.cs.ucla.edu/~lixia/papers/90sigcomm.pdf >.

End of changes. 36 change blocks.

123 lines changed or deleted

144 lines changed or added

This html diff was produced by rfcdiff 1.42. The latest version is available from <http://tools.ietf.org/tools/rfcdiff/>