Transport Area Working Group
Internet-Draft
Intended status: Informational
Expires: 2 September 2026

J. Yang
W. Cheng
Y. Tian
China Mobile
J. Wang
G. Zhang
Centec

1 March 2026

Coupling ECN Marking Thresholds with Dynamic Buffer Allocation
draft-yang-dynamic-ecn-threshold-00

Abstract

Explicit Congestion Notification (ECN) marking thresholds are typically configured statically. In modern network devices that employ dynamic buffer allocation -- where the maximum buffer available to a queue fluctuates dynamically based on the number of active queues and the remaining shared buffer pool -- a static ECN threshold can frequently become misaligned with the actual instantaneous buffering capacity.

This misalignment can lead to pathological behaviors: either premature marking (which underutilizes available buffers and throttles throughput) or late marking (which provides no advance warning before tail drop occurs). This document describes an operational framework and a deterministic reference algorithm for dynamically coupling the ECN marking threshold with the dynamic buffer allocation limit. By maintaining an adaptive relationship through configurable parameters, this mechanism ensures robust congestion signaling across varying load conditions without requiring complex external machine-learning models or per-flow tracking.

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 https://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 2 September 2026.

Copyright Notice

Copyright (c) 2026 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 (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Explicit Congestion Notification (ECN) [RFC3168] enables network devices to signal incipient congestion to endpoints without resorting to packet drops. A device marks a packet's IP header with the Congestion Experienced (CE) codepoint when a specific queue metric exceeds a configured Active Queue Management (AQM) threshold. The sender, upon learning of the CE mark through transport-layer feedback, proactively reduces its sending rate.

Conventionally, the ECN marking threshold is established as a static value chosen by the network operator. This static approach functions adequately when the maximum buffer available to a given queue is also static and predictable. However, the architecture of modern data center switches heavily relies on dynamic buffer allocation. In such architectures, the maximum buffer a queue is permitted to consume (Buf_Thrd) fluctuates significantly based on the total available shared buffer and the instantaneous number of active queues drawing from it. Dynamic buffer allocation schemes, such as those utilizing the alpha parameter model, are widely deployed in commodity switching silicon to maximize memory utilization.

When Buf_Thrd shrinks (e.g., due to an incast event activating many queues), a static ECN threshold originally positioned well below the nominal buffer limit may suddenly be equal to or greater than the current Buf_Thrd. In this scenario, the device is forced into tail drop before the queue occupancy ever reaches the ECN threshold. The ECN mechanism effectively fails, yielding severe packet loss and higher tail latency rather than graceful rate reduction.

Conversely, when the network load decreases and Buf_Thrd expands, the static threshold may sit far below the actual buffer capacity. This drastically underutilizes available buffering, generating premature congestion signals that trigger unnecessary rate reduction and diminish overall link utilization.

Unlike sojourn-time based AQM algorithms (such as CoDel [RFC8289] or PIE [RFC8033]), which inherently adapt to buffer size variations by measuring delay rather than bytes, queue-depth based marking mechanisms (e.g., standard step-marking in DCTCP [RFC8257] or RoCEv2 environments) are highly vulnerable to dynamic buffer fluctuations.

This document specifies an operational mechanism that continually derives the ECN marking threshold (ECN_Thrd) from the instantaneous value of Buf_Thrd. The computation introduces two operator-configurable parameters to maintain predictable headroom. The approach offers a deterministic, hardware-friendly solution to maintain a consistent relationship between ECN marking and buffer availability.

2. Terminology

In the context of this document, a "queue" typically refers to a per-port, per-traffic-class transmission queue within a forwarding device.

Buf_Thrd (Buffer Threshold): The dynamic buffer allocation limit for

a specific queue. This represents the maximum amount of shared buffer memory that the queue is currently authorized to occupy. Buf_Thrd is periodically or event-driven recomputed by the device’s buffer management subsystem.

ECN_Thrd (ECN Threshold): The active ECN marking threshold for a queue. When the instantaneous or averaged queue occupancy meets or exceeds ECN_Thrd, the device applies the CE codepoint to arriving ECN-capable packets.

Offset: A configurable parameter dictating the desired buffer headroom (typically measured in bytes or cells) maintained between Buf_Thrd and ECN_Thrd. The Offset acts as a shock absorber for packets already in-flight during the control loop feedback delay.

ECN_Floor: A configurable parameter establishing the minimum permissible boundary for ECN_Thrd. It acts as a safeguard against ECN_Thrd collapsing to excessively low values (e.g., below a single MTU), which would cause catastrophic throughput degradation via aggressive continuous marking.

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Applicability Statement

This document is explicitly applicable to network forwarding devices utilizing queue-depth based ECN marking mechanisms in conjunction with a dynamic buffer allocation scheme. It is primarily targeted at Data Center Networks (DCN) and high-speed interconnects where instantaneous queue length or average queue length is evaluated against a byte-based or cell-based threshold.

This specification does NOT target devices employing sojourn-time based AQMs (e.g., [RFC8289], [RFC8033]), as time-based algorithms naturally abstract away the physical buffer size and are generally immune to the dynamic shared buffer problem described herein.

The operational logic defined here is strictly internal to the network device. It does not alter the ECN wire protocol, IP-layer ECN codepoint semantics, or the transport-layer negotiation standardized in [RFC3168].

The method is compatible with Classic ECN marking as well as modern scalable congestion controls (e.g., the L4S architecture [RFC9330] and its ECN protocol [RFC9331]). In a DualQ Coupled AQM [I-D.ietf-tsvwg-aqm-dualq-coupled] architecture, the dynamically computed ECN_Thrd may serve as the target threshold for the Classic queue, leaving the L4S queue's specialized marking behavior independent.

4. Problem Statement

To formalize the problem context, consider a Top-of-Rack (ToR) switch equipped with a 12 MB shared buffer pool and 48 egress ports. Under light traffic conditions with only 4 queues active, the dynamic buffer management may assign a Buf_Thrd of 3 MB to each active queue. Assuming a network operator statically configures an ECN threshold of 200 KB, the system operates with 2.8 MB of effective headroom, providing ample shock absorption.

However, during a coordinated incast event where all 48 ports become heavily congested, the shared buffer is fractured, and the dynamic Buf_Thrd for each queue plummets to 250 KB. The statically configured 200 KB ECN threshold now yields a mere 50 KB of headroom. In high-speed environments (e.g., 100Gbps+), 50 KB is significantly smaller than the Bandwidth-Delay Product (BDP) of the control loop. Consequently, the queue will hit the tail drop limit (Buf_Thrd) before the transport sender has time to react to the CE marks, inducing severe retransmission timeouts and latency spikes.

Conversely, if the operator statically configures the ECN threshold to 2 MB to optimize for high throughput under light load, the ECN mechanism will completely fail during the incast event because the static ECN threshold (2 MB) heavily exceeds the active Buf_Thrd (250 KB).

A deterministic, dynamic coupling between Buf_Thrd and ECN_Thrd is necessary to resolve these dual failure modes without relying on static compromises.

5. Dynamic Coupling Architecture

5.1. Prerequisite: Buffer State Awareness

The foundation of this architecture requires the device's forwarding plane to expose the current Buf_Thrd value to the AQM/ECN marking engine. The specific memory management algorithm (e.g., alpha-based proportional allocation) calculating Buf_Thrd is outside the scope of this document. The sole prerequisite is that Buf_Thrd is continuously updated and accessible with low latency.

5.2. Reference Algorithm for ECN Threshold

Network devices SHOULD compute ECN_Thrd continuously based on Buf_Thrd, Offset, and ECN_Floor. To ensure stability across all load extremes, the logic is segmented into three distinct operational regions:

Region A -- Sufficient Buffer (Nominal State):

Condition: (Buf_Thrd - Offset) > ECN_Floor. The buffer allocation is generous enough to accommodate the full requested headroom (Offset). Here, ECN_Thrd = Buf_Thrd - Offset. The ECN threshold securely tracks the dynamic buffer limit, guaranteeing precisely the configured absorption capacity.

Region B -- Constrained Buffer (Congested State):

Condition: (Buf_Thrd - Offset) <= ECN_Floor AND Buf_Thrd > ECN_Floor. The shared buffer is highly constrained. Enforcing the full Offset would depress ECN_Thrd below the critical ECN_Floor, risking excessive marking and severe throughput collapse. To mitigate this, the threshold is clamped: ECN_Thrd = ECN_Floor. The available headroom compresses to (Buf_Thrd - ECN_Floor), prioritizing reasonable throughput over optimal packet absorption.

Region C -- Critical Buffer (Exhaustion State):

Condition: Buf_Thrd <= ECN_Floor. The queue’s buffer allocation has collapsed to or below the minimum floor. In this critical state, clamping ECN_Thrd to ECN_Floor would result in ECN_Thrd >= Buf_Thrd, rendering ECN useless (tail drops would occur silently). Thus, ECN_Thrd = Buf_Thrd. While zero headroom remains, the device marks packets exactly at the tail drop boundary, ensuring the network still transmits explicit congestion signals.

The reference logic is expressed as follows:

```
function compute_ecn_threshold(Buf_Thrd, Offset, ECN_Floor):
    IF (Buf_Thrd - Offset) > ECN_Floor:
        RETURN Buf_Thrd - Offset          // Region A: Optimal tracking
    ELSE IF Buf_Thrd > ECN_Floor:
        RETURN ECN_Floor                  // Region B: Floor clamped
    ELSE:
        RETURN Buf_Thrd                   // Region C: Drop boundary
```

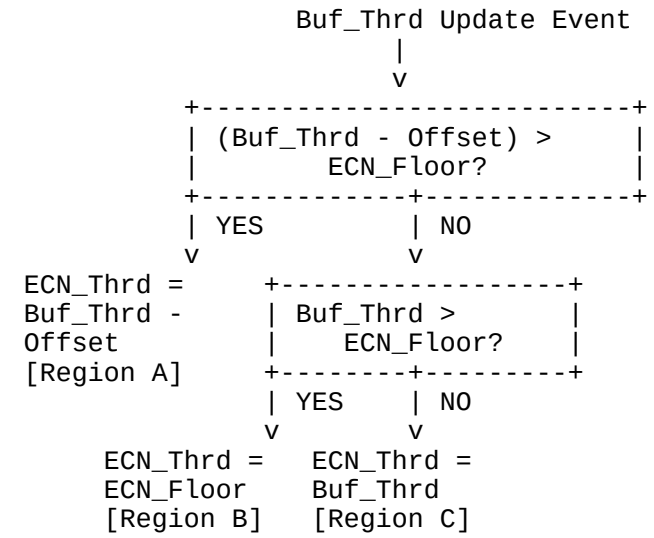

Figure 1: State Transition of Dynamic ECN Threshold

This algorithm requires minimal logic gates (two comparators and one subtractor), ensuring it can be evaluated in standard Application-Specific Integrated Circuit (ASIC) pipelines with nominal nanosecond latency.

5.3. Architectural Invariants

Implementations conforming to this framework SHOULD validate the following invariants to prevent anomalous traffic handling:

1. ECN_Thrd MUST NOT exceed Buf_Thrd (ECN_Thrd <= Buf_Thrd). This mathematically guarantees ECN marking is always attempted prior to or simultaneously with queue tail drop.

2. ECN_Thrd MUST NOT fall below ECN_Floor, UNLESS the maximum physical buffer limit (Buf_Thrd) has itself fallen below ECN_Floor.

6. Operational Considerations

6.1. Update Synchronization

ECN_Thrd MUST be inherently recomputed concurrently with any transition in Buf_Thrd. Event-driven synchronization is highly RECOMMENDED over periodic polling. Polling introduces phase-delay, leaving the ECN_Thrd stale during the most critical microsecond inflection points of transient congestion. If atomic hardware updates are impossible, implementations SHOULD bias the asynchronous

race condition to temporarily favor a lower ECN_Thrd (causing a premature mark) over a higher ECN_Thrd (causing an unnotified drop).

6.2. Tuning the Offset Parameter

The Offset represents the network's required "shock absorber." Operators SHOULD calibrate the Offset to slightly exceed the expected Bandwidth-Delay Product (BDP) of the typical congestion control feedback loop:

Offset Link_Rate * RTT

In contemporary intra-data-center fabrics (RTT ~20-50 microseconds, 400 Gbps links), Offset values ranging from 1 MB to 2.5 MB are operationally appropriate. Oversizing the Offset prematurely throttles flows; undersizing it invites high tail-drop rates despite ECN capability.

6.3. Tuning the ECN_Floor Parameter

ECN_Floor establishes the maximum throttling severity. It MUST NOT be configured smaller than the Maximum Transmission Unit (MTU) of the link (e.g., 9000 bytes). For environments executing Data Center TCP (DCTCP) [RFC8257], ECN_Floor SHOULD typically mirror the static thresholds recommended for shallow buffering (e.g., 30 KB to 100 KB), preventing the queue from emptying completely while maintaining ultra-low queuing delay.

7. Implementation Status

[RFC Editor: Please remove this section before publication.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs.

The dynamic ECN threshold coupling mechanism described in this document has been implemented and validated in the data plane of Centec Networks' switching silicon, specifically designed to mitigate micro-bursts and incast congestion in large-scale RDMA over Converged Ethernet (RoCEv2) deployments by China Mobile.

8. Related Work

AQM recommendations generalized in [RFC7567] outline the complexities of parameter tuning. While this document aligns with the intent of [RFC7567], it specifically isolates and resolves the intersection of AQM and dynamic shared buffering, a domain not fully explored in legacy AQM guidelines.

The AI-based ECN approach proposed in [I-D.zhuang-tsvwg-ai-ecn-for-dcn] targets similar parameter adaptation via machine learning. The framework in this document, conversely, advocates for a mathematically deterministic data-path calculation, demanding no training data, no external control-plane telemetry loop, and zero inference latency.

TCP Alternative Backoff with ECN (ABE) [RFC8511] optimizes how endpoints react to CE marks. ABE is strictly complementary; it refines the sender response, whereas this architecture ensures the network device generates those marks at structurally correct moments.

9. Security Considerations

This specification introduces an automated internal parameter coupling within the network forwarding plane. It does not exchange new protocol messages across the wire, thus introducing no new cryptographic or protocol-level attack surfaces.

Operational Degradation via Misconfiguration: Invalid configuration of Offset or ECN_Floor can initiate self-inflicted Denial of Service (DoS) behaviors. For instance, an immensely inflated Offset might universally push the system into Region C, effectively disabling early congestion warning. Implementations SHOULD validate parameter inputs through management interfaces and emit warnings if Offset exceeds typical physical buffer allocations.

Internal Signaling Integrity: The architectural dependency between the memory management unit (MMU) and the ECN marking engine requires deterministic internal signaling. If the internal update of Buf_Thrd is delayed or corrupted under heavy system load, the ECN_Thrd calculation will be based on stale memory constraints, leading to temporary periods of over-marking or under-marking. Hardware designs SHOULD prioritize this internal signaling path.

Buffer Exhaustion Vectors: Malicious, non-responsive flows could intentionally occupy massive allocations of the shared buffer pool. In dynamic buffer architectures, this action compresses the Buf_Thrd for all other benign queues, plunging them into Region B or Region C. This is an inherent vulnerability of shared memory switches, not

generated by this ECN algorithm. Operators MUST utilize per-queue maximum caps, port-level QoS scheduling, and admission control to insulate queues from cross-traffic buffer starvation.

10. IANA Considerations

This document has no IANA actions.

11. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

12. Informative References

[RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <https://www.rfc-editor.org/info/rfc7567>.

[RFC8033] Pan, R., Natarajan, P., Baker, F., and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem", RFC 8033, DOI 10.17487/RFC8033, February 2017, <https://www.rfc-editor.org/info/rfc8033>.

[RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257, October 2017, <https://www.rfc-editor.org/info/rfc8257>.

[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>.

[RFC8511] Khademi, N., Welzl, M., Armitage, G., and G. Fairhurst, "TCP Alternative Backoff with ECN (ABE)", RFC 8511, DOI 10.17487/RFC8511, December 2018, <https://www.rfc-editor.org/info/rfc8511>.

[RFC9330] Briscoe, B., Ed., De Schepper, K., Bagnulo, M., and G. White, "Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture", RFC 9330, DOI 10.17487/RFC9330, January 2023, <https://www.rfc-editor.org/info/rfc9330>.

[RFC9331] De Schepper, K. and B. Briscoe, Ed., "The Explicit Congestion Notification (ECN) Protocol for Low Latency, Low Loss, and Scalable Throughput (L4S)", RFC 9331, DOI 10.17487/RFC9331, January 2023, <https://www.rfc-editor.org/info/rfc9331>.

[I-D.zhuang-tsvwg-ai-ecn-for-dcn] Zhuang, Y., Zhang, B., and H. Pan, "Artificial Intelligence (AI) based ECN adaptive reconfiguration for datacenter networks", Work in Progress, Internet-Draft, draft-zhuang-tsvwg-ai-ecn-for-dcn-00, October 2019, <https://datatracker.ietf.org/doc/draft-zhuang-tsvwg-ai-ecn-for-dcn/>.

[I-D.ietf-tsvwg-aqm-dualq-coupled] De Schepper, K. and B. Briscoe, Ed., "DualQ Coupled AQMs for Low Latency, Low Loss and Scalable Throughput (L4S)", Work in Progress, Internet-Draft, draft-ietf-tsvwg-aqm-dualq-coupled-24, 2024, <https://datatracker.ietf.org/doc/draft-ietf-tsvwg-aqm-dualq-coupled/>.

Authors' Addresses

Jin Yang
China Mobile
Beijing
100053
China
Email: yangjinwl@chinamobile.com

Weiqiang Cheng
China Mobile
Beijing
100053
China
Email: chengweiqiang@chinamobile.com

Yuchi Tian
China Mobile
Beijing
100053
China
Email: tianyuchi@chinamobile.com

Junjie Wang
Centec
Suzhou
215000
China
Email: wangjj@centec.com

Guoying Zhang
Centec
Suzhou
215000
China
Email: zhanggy@centec.com