

FANTEL  
Internet-Draft  
Intended status: Standards Track  
Expires: 8 January 2026

X. Geng  
Huawei  
P. Huo  
ByteDance  
W. Cheng  
China Mobile  
D. Li  
Tsinghua University  
Y. Zhu  
China Telecom  
Z. Han  
China Unicom  
7 July 2025

Gap Analysis of Fast Notification for Traffic Engineering and Load  
Balancing  
draft-geng-fantel-fantel-gap-analysis-01

Abstract

Modern networks require fast, adaptive Traffic Engineering (TE) to support demanding applications like AI training and real-time services. Existing mechanisms for load balancing, protection, and flow control often lack responsiveness and scalability. This document analyzes key gaps in current TE solutions and proposes fast notification as a low-latency, event-driven enhancement. Fast notification enables real-time network awareness and quicker reactions to dynamic conditions, improving overall network efficiency and reliability.

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 8 January 2026.

## Copyright Notice

Copyright (c) 2025 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. Fast Notification for Traffic Engineering and Load Balancing:	
Gap Analysis . . . . .	2
1.1. Introduction . . . . .	2
1.2. Requirements Language . . . . .	3
1.3. Gap Analysis for Load Balancing . . . . .	3
1.3.1. IOAM Telemetry Limitations . . . . .	3
1.3.2. Role of Fast Notification . . . . .	4
1.4. Gap Analysis for Protection . . . . .	4
1.4.1. Bidirectional Forwarding Detection (BFD) . . . . .	4
1.4.2. Fast Reroute (FRR) . . . . .	5
1.4.3. Routing Convergence . . . . .	5
1.4.4. Multi-Path Routing (e.g., ECMP) . . . . .	5
1.5. Gap Analysis for Flow Control . . . . .	6
1.5.1. Sender-Based Congestion Control . . . . .	6
1.5.2. Receiver Based TCP Congestion Control . . . . .	6
1.5.3. Explicit Congestion Notification (ECN) . . . . .	7
1.5.4. Inband Network Telemetry (INT) . . . . .	7
1.6. Conclusion . . . . .	8
2. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Fast Notification for Traffic Engineering and Load Balancing: Gap Analysis

### 1.1. Introduction

In use cases such as AI training, a lossless and adaptive network is required to ensure reliable and congestion-free data transfer. These workloads demand high throughput, low latency, and zero packet loss across dynamically shifting traffic patterns.

To meet these demands, networks rely on Traffic Engineering (TE) mechanisms, including load balancing, protection, and flow control. However, existing solutions face limitations in responsiveness, coverage, and operational overhead, especially in high-speed, large-scale environments.

This document provides a gap analysis focused on three key TE areas:

- \* Load Balancing
- \* Protection
- \* Flow Control

For each area, we analyze current limitations and explore how fast notification mechanisms can help fill these gaps.

It is important to clarify that the mechanisms discussed in this document, such as BFD, IOAM, and traditional transport-layer flow control, are not necessarily alternatives to fast notification. Instead, these mechanisms can complement notification-based approaches. For instance, measurement results from BFD or IOAM can serve as triggers for fast notifications. Similarly, existing flow control mechanisms at the transport layer can work in coordination with network-layer flow control enabled by notifications. Therefore, the "gaps" identified in this document reflect potential enhancements when relying solely on these mechanisms without fast notification, rather than suggesting they should be replaced.

## 1.2. Requirements Language

TBD

## 1.3. Gap Analysis for Load Balancing

Load balancing ensures efficient utilization of available bandwidth and reduces congestion. In modern networks, dynamic load balancing is essential but often lacks real-time responsiveness.

### 1.3.1. IOAM Telemetry Limitations

In-situ OAM (IOAM) provides visibility into traffic by embedding telemetry data directly in packets. It enables measurement of path latency, loss, and performance metrics.

However, IOAM has notable drawbacks:

- \* **Telemetry Export Delays:** IOAM data is extracted and reported by the device CPU to a controller. This adds latency and limits responsiveness.
- \* **Controller Reaction Time:** Centralized controllers typically process telemetry in software, resulting in delayed decision-making.

Gap: These factors reduce the effectiveness of real-time load balancing.

#### 1.3.2. Role of Fast Notification

To address the above:

- \* **Proactive Signaling:** Fast notification can signal network conditions (e.g., congestion) before service degradation occurs.
- \* **Event-Driven Control:** Control loops can dynamically adjust traffic distribution without relying on polling or telemetry aggregation.
- \* **Lightweight Signaling:** Avoids the overhead of traditional telemetry processing.

#### 1.4. Gap Analysis for Protection

Protection mechanisms ensure service continuity in case of failures. While existing tools like BFD and FRR are widely deployed, they have inherent limitations in speed and scope.

##### 1.4.1. Bidirectional Forwarding Detection (BFD)

BFD is designed for rapid fault detection by sending frequent control packets between peers. While widely used, it presents the following limitations:

- \* **Overhead vs. Frequency Tradeoff:** Higher probe frequency improves detection time but increases CPU and bandwidth usage.
- \* **Scalability Issues:** Maintaining many BFD sessions in large-scale networks strains the control plane.
- \* **Path Detection Limitations:** In scenarios with multiple ECMP paths, BFD struggles to detect the status of specific paths, making it difficult to identify partial failures or asymmetrical degradations.

Gap: BFD struggles to balance detection speed with system overhead.

#### 1.4.1.1. Fast Notification Enhancement

- \* **Targeted Notifications:** Fast notification provides event-driven alerts rather than continuous probing.
- \* **Improved Scalability:** Reduces resource usage while preserving rapid failure detection.

#### 1.4.2. Fast Reroute (FRR)

FRR reroutes traffic upon link or node failures. However:

- \* **Local-Only Protection:** Typically protects against only adjacent failures.

Gap: FRR lacks flexibility and responsiveness in complex topologies.

#### 1.4.2.1. Fast Notification Enhancement

- \* **Instant Failure Alerts:** Enables immediate detection and rerouting across the network.
- \* **Minimized Packet Loss:** Reduces the time between failure detection and redirection.

#### 1.4.3. Routing Convergence

Routing Convergence mechanisms depend on routing protocol convergence, which may take hundreds of milliseconds.

Gap: Delay-sensitive services cannot tolerate slow failover.

#### 1.4.3.1. Fast Notification Enhancement

- \* **Real-Time Failover:** Triggers immediate switching to standby paths.
- \* **Service Continuity:** Ensures uninterrupted performance for critical applications.

#### 1.4.4. Multi-Path Routing (e.g., ECMP)

Equal-Cost Multi-Path (ECMP) routing uses multiple paths for load sharing. However:

Gap: It lacks fast detection of path degradation or failure, making real-time traffic rebalancing difficult.

#### 1.4.4.1. Fast Notification Enhancement

- \* On-the-Fly Path Reallocation: Shifts traffic to healthy paths based on real-time failure or degradation alerts.
- \* Improved Reliability: Maintains availability during partial failures.

### 1.5. Gap Analysis for Flow Control

Flow control ensures congestion-free transmission and optimal throughput. Current mechanisms either react too slowly or lack granular, real-time information.

#### 1.5.1. Sender-Based Congestion Control

Congestion control is based on end-to-end feedback such as packet loss or RTT increases.

- \* End-to-End Delay Sensitivity: Sender-driven control relies on detecting congestion from end-to-end signals, often after at least one RTT. In bursty traffic scenarios such as data centers, this delay may result in buffer bloat or packet loss.
- \* Ambiguity in Signal Source: It's also hard to distinguish between congestion and transient fluctuations, leading to overreaction or misjudgment in rate adaptation.

Gap: These signals are slow and reactive, especially in high-latency or long-RTT environments.

##### 1.5.1.1. Fast Notification Enhancement

- \* Mid-Path Feedback: Intermediate nodes can issue real-time congestion alerts.
- \* Faster Rate Adjustment: Prevents packet loss and improves flow responsiveness.

#### 1.5.2. Receiver Based TCP Congestion Control

Receiver driven congestion control uses feedback signals from the receiver to adjust transmission rate of the sender.

- \* Control Loop Latency: These signals still traverse the network and are subject to RTT delays, especially problematic in high-speed dynamic environments.

- \* **Bandwidth Overhead:** In large-scale or short-flow-intensive environments like data centers, signaling from massive numbers of receivers can impose significant bandwidth and processing overhead.

#### 1.5.2.1. Fast Notification Enhancement

- \* **Direct Congestion Signals:** Reduces RTT-related lag by injecting congestion indicators directly into the network fabric.
- \* **Efficient Scaling:** Enables scalable control even in environments with many short-lived flows.

#### 1.5.3. Explicit Congestion Notification (ECN)

ECN marks packets to indicate congestion, avoiding drops. However:

Gap: ECN still relies on end-to-end signaling and lacks precise real-time feedback.

##### 1.5.3.1. Fast Notification Enhancement

- \* **Granular Congestion Updates:** Real-time alerts from within the network augment ECN markings.
- \* **Proactive Shaping:** Faster congestion mitigation before queue buildup.

#### 1.5.4. Inband Network Telemetry (INT)

INT provides path-level telemetry by inserting metadata at each hop, which is returned to the sender via the ACK. Some congestion control algorithms, such as HPCC, utilize INT for precise load-awareness.

However:

- \* **RTT Dependency:** INT-based telemetry still incurs a one-RTT delay before feedback is received by the sender.
- \* **Feedback Loop Latency:** This delay limits responsiveness, especially in dynamic high-speed environments.

##### 1.5.4.1. Fast Notification Enhancement

- \* **Immediate Inline Feedback:** Enables mid-network nodes to send congestion indicators directly, bypassing RTT delays.

- \* Enhanced Responsiveness: Combines the accuracy of INT with faster notification paths for congestion control.

## 1.6. Conclusion

This document highlights the following gaps in Traffic Engineering mechanisms and how fast notification can enhance each area:

Area	Key Gap	Fast Notification Enhancement
Load Balancing	Slow telemetry export and software control delays	Event-driven signaling for immediate adjustment
Protection	BFD/FRR trade off speed for overhead; slow convergence	Lightweight, fast fault alerts across the network topology
Flow Control	TCP/ECN feedback too slow for real-time adaptation	Real-time congestion feedback from network infrastructure

Table 1

Fast notification mechanisms provide a low-latency, low-overhead method for improving responsiveness across load balancing, protection, and flow control. These capabilities are increasingly vital to support demanding applications like distributed AI training and real-time cloud services.

## 2. Informative References

- [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/rfc/rfc3168>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/rfc/rfc5880>>.
- [RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <<https://www.rfc-editor.org/rfc/rfc7490>>.



Authors' Addresses

Xuesong Geng  
Huawei  
Email: gengxuesong@huawei.com

PengFei Huo  
ByteDance  
Email: huopengfei@bytedance.com

Weiqiang Cheng  
China Mobile  
Email: chengweiqiang@chinamobile.com

Dan Li  
Tsinghua University  
Email: toolidan@tsinghua.edu.cn

Yongqing Zhu  
China Telecom  
Email: zhuyq8@chinatelecom.cn

Zhengxin Han  
China Unicom  
Email: hanzx21@chinaunicom.cn