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

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

Requirements of Fast Notification for Traffic Engineering and Load Balancing
draft-geng-fantel-fantel-requirements-02

Abstract

This document defines the requirements for Fast Notification for Traffic Engineering and Load Balancing (FaNTEL), a mechanism designed to deliver timely network status updates directly from the network device with a change to the device expected to react to it. FaNTEL supports fast failure and congestion notifications, enabling rapid protection switching and dynamic load balancing. By providing low-latency alerts, it helps networks respond quickly to link failures and congestion events, enhancing service reliability and performance.

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. Introduction to Fast Notification

1.1. Background and Motivation

In today’s increasingly dynamic and complex network environments, efficient traffic management and rapid adaptation to network changes are critical. Traditional network management systems are often limited in their ability to react quickly to sudden traffic shifts, failures, or congestion. As a result, these networks may experience performance degradation, prolonged service disruptions, or inefficient resource utilization.

The demand for faster, more responsive network management has intensified significantly with the evolution of AI training and reasoning traffic. This new scenario presents unique characteristics, including larger packets (e.g., 4KB), increased overall traffic volume, and a shift towards fewer but larger flows.

These changes introduce distinct network challenges. Maintaining high performance and availability necessitates high-speed interconnects supporting 200-400 Gbps for GPUs. Furthermore, effective load balancing and congestion control mechanisms are crucial to ensure that these massive, critical data flows are managed efficiently and without interruption. The ability to meet these demands is paramount for optimizing AI workloads and ensuring continuous, high-performance operations.

Fast Notification for Traffic Engineering and Load Balancing is a mechanism that delivers timely notification of network events (e.g., link failures, congestion, traffic shifts, or load imbalances) to the relevant network nodes. By enabling rapid communication between devices, fast notification facilitates quicker decision-making and faster adjustments to network routing and traffic management strategies.

The core principle of Fast Notification is to reduce the time it takes for a network node to become aware of a change in its environment and to adjust accordingly. This is achieved through the use of high-priority, low-latency signaling mechanisms that notify nodes of changes in traffic patterns or network conditions almost immediately.

1.2. Notification Procedure

* Fast Notification Messages: Lightweight messages that convey state changes (such as traffic or network failure events) from one node to others.

* Notification Propagation Mechanism: A reliable and efficient way to disseminate notifications quickly throughout the network.

* Triggering Mechanism for Message Sending(out of scope of FaNTEL): A mechanism that detects significant network changes (e.g., link utilization thresholds, delay spikes, packet loss) and initiates the sending of fast notification messages.

* Action after Receiving the Message(out of scope of FaNTEL): An action (such as rerouting traffic or applying flow control) once the notification is received.

The requirements of FaNTEL focus on the definition of notifications and their corresponding propagation mechanisms. The methods for triggering notifications and the actions taken upon receiving these notifications, whether through existing solutions or new protocol extensions, are out of scope for this document.

The mechanisms that are out of the scope of current notification requirements could be implemented using existing solutions.

Note: The detailed mechanisms and implementations (such as message format, propagation protocols) are out of scope of this document and will be specified in separate documents.

2. Fast Notification for Load Balancing

2.1. Background: Challenges in Load Balancing

Load balancing is a critical function in AI networks, ensuring that network resources are efficiently allocated and that no single node or link becomes overwhelmed with excessive traffic. Proper load balancing improves network performance, prevents bottlenecks, and ensures that network services remain responsive and reliable.

However, current load balancing techniques face significant challenges in highly dynamic environments. One of the core issues is the lack of timely awareness and adaptive response to network state changes. Traditional mechanisms often rely on periodic global state synchronization or static policies, which results in delayed and inaccurate decision-making. These delays make it difficult to capture instantaneous changes such as link congestion, node failures, or traffic bursts.

Moreover, load balancing decisions based on local views cannot perceive downstream contention or routing fluctuations, potentially leading to persistent traffic injection into congested paths and increased queuing and packet loss.

Fast Notification is supposed to support load balancing by providing fast, efficient notification of changes in traffic patterns, network failures, and congestion. By using high-priority, low-latency messages, Fast Notification allows network nodes to immediately adjust their load balancing decisions in response to these changes, ensuring optimal resource utilization and performance.

2.2. Requirements for Fast Notification in Load Balancing

1. Traffic State Detection: Monitoring of traffic patterns, link utilization, and node load to trigger notifications on significant deviations.

2. Notification Propagation: Propagation from congestion node with event details (e.g., congestion, traffic shift) to relevant devices.

3. Action Adjustments: Nodes can reroute or redistribute traffic immediately upon receiving a notification.

Once a fast notification message is received, the load balancing mechanism is supposed to immediately reassess the routing and traffic allocation strategy. This may involve:

* Shifting flows to underutilized paths

* Splitting traffic across multiple paths

* Throttling traffic destined for congested regions

In addition, nodes may update their local state or forward the notification upstream to further optimize the network reaction. Timely and coordinated response across the network significantly enhances load balancing effectiveness.

2.3. Design Gaols

* Traffic Information in time: Fast notification provides up-to-date information about the current state of the network, including traffic volume, node utilization, and link load.

* Precise Load Rebalancing: Enables immediate notifications to the affected nodes for quick traffic redistribution.

* Optimized Resource Utilization: Supports fine-grained traffic distribution on a per-packet or per-flow basis.

3. Fast Notification for Protection

3.1. Background: Challenges in Network Protection

Network protection ensures service availability and minimizes disruptions due to failures like link outages or device malfunctions. However, traditional protection mechanisms face several limitations:

* Slow Detection and Recovery: Traditional protection often relies on periodic failure detection and centralized rerouting, resulting in recovery times that are not fast enough for modern service expectations.

* Inefficient Failover: Without fast notification, failover paths may not be activated or optimized in time, leading to service interruption.

In high-reliability scenarios, network protection must be capable of rapid detection and notification of failures to meet performance goals such as sub-50ms recovery.

Fast Notification enables rapid notification of failures, allowing near-instantaneous and dynamic protection responses, minimizing user impact.

3.2. Requirements for Fast Notification in Protection

1. Failure Detection and Notification: Notifications are generated when failures occur and propagated directly from failed node to the relevant respond node.

2. Precise Notification Propagation: Notifications must reach relevant nodes quickly, such as upstream routers.

3. Optimization of Backup Paths: Failure notifications can trigger optimized rerouting or pre-established backup path activation.

Upon receiving a notification of failure, protection mechanisms may immediately switch to backup paths, reroute traffic, or suppress affected routes. This ensures minimal disruption and quick recovery. Coordinated response strategies may include upstream node notification, service-aware failover, and path re-optimization based on updated network topology.

3.3. Design Gaols

* Rapid Failure Response: Enables sub-second (or even sub-50ms) reaction to failures.

* Improved Service Continuity: Minimizes traffic loss and recovery time.

* Efficient Resource Utilization: Ensures backup resources are used only when needed, and in the most optimal way.

3.4. Integration Requirements with Existing Protection Mechanisms

Fast Notification can be integrated with various existing protection schemes to improve their responsiveness and efficiency:

* Fast Reroute (FRR): Fast notification enhances FRR by delivering failure notifications almost instantaneously, allowing for faster and more efficient rerouting of traffic. This helps maintain high availability and minimizes service disruption.

* Hot Stand-by: Fast notification complements Routing Protocol Convergence protocols by providing fast failure notifications, ensuring that devices can quickly switch to backup paths and maintain service continuity.

* Multi-Path Routing: In networks using ECMP or other multi-path routing protocols, fast notification enables the immediate re-adjustment of traffic flows when a failure is detected, ensuring optimal use of available paths.

4. Fast Notification for Flow Control

4.1. Background: Challenges in Flow Control

Fast Notification enhances flow control by providing a fast, low-latency notification system that can detect and alert network devices to congestion events in time. With Fast Notification, congestion can be identified and communicated to relevant network nodes almost instantaneously, allowing for rapid mitigation actions such as traffic rerouting, rate limiting, or queuing adjustments.

Note: Unlike traditional host-to-host (end-to-end) flow control mechanisms at the transport layer (e.g., TCP), this document focuses on Layer 3 (network layer) flow control. Specifically, it targets congestion control and buffering actions between adjacent network nodes, enabling upstream nodes to slow down or buffer traffic in response to downstream congestion.

A key challenge in flow control is the timely detection and dissemination of congestion events to avoid packet loss and throughput degradation. Traditional flow control mechanisms often rely on delayed feedback or reactive responses, which can lead to suboptimal network performance in highly dynamic environments.

4.2. Requirements for Fast Notification in Flow Control

The integration of Fast Notification into flow control mechanisms involves several key processes:

1. Congestion Detection: Network devices continuously monitor traffic flows and link usage to identify potential congestion points. When congestion is detected, a notification is generated and sent through the Fast Notification system. These notifications include critical information, such as the affected link or device, the severity of the congestion, and the current traffic load.

2. Notification Propagation: Once the congestion event is detected, the Fast Notification system quickly propagates this information upstream to adjacent nodes that may contribute to the congestion. This enables upstream nodes to take appropriate actions, such as rate limiting or buffering.

3. Backpressure and Buffering: Instead of relying solely on rerouting or end-to-end rate control, this approach allows upstream network nodes to apply backpressure by slowing down traffic forwarding or buffering packets locally. This helps to absorb traffic bursts and prevent packet loss downstream.

4.3. Design Gaols

* Congestion Detection: Fast Notification delivers updates about network conditions, enabling relevant network nodes to know the congestion as soon as it occurs. This ensures that corrective actions can be taken promptly before the congestion worsens.

* Adaptive Node-to-Node Congestion Management: Fast Notification enables adaptive congestion management at the network node level by allowing nodes to dynamically adjust forwarding behavior and buffer usage in response to congestion notifications from downstream nodes.

* Minimized Packet Loss: By enabling fast congestion alerts within the network, Fast Notification helps avoid packet loss by triggering corrective actions such as backpressure and flow rate adjustments upstream, before congestion reaches critical levels.

4.4. Integration with Existing Flow Control Mechanisms

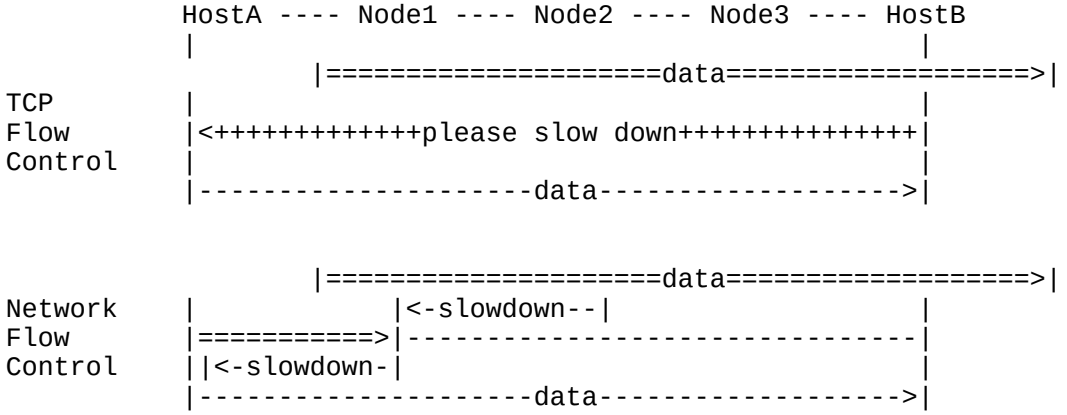
Fast Notification can be integrated with existing flow control strategies to improve their responsiveness and efficiency:

* Transport Layer Flow Control (for example: TCP): Fast Notification is distinct from traditional TCP flow control, which operates end-to-end between hosts. TCP reacts to congestion signals that are often delayed due to network round-trip times.

* Layer 3 Node-to-Node Flow Control: The mechanism proposed here focuses on adjacent network nodes cooperating via Fast Notification to perform rapid congestion signaling and buffering. This reduces reaction time and improves network stability in dynamic environments.

* Explicit Congestion Notification (ECN): Fast Notification can complement ECN by providing more granular, rapid updates on congestion status within the network fabric, allowing quicker local reactions beyond the transport layer.

4.5. Illustration: Host-to-Host vs Node-to-Node Flow Control



5. Fast Notification for Capability Announcement

In addition to conveying failure or performance-related events, there is a potential need for a lightweight mechanism to announce certain network capabilities that are not directly related to routing.

Specifically:

* Some types of network capability information (e.g., processing features, service functions, queuing models, etc.) may need to be announced among network nodes;

* These types of information are not suitable for distribution via existing IGP or BGP mechanisms, either due to scope, frequency, or protocol constraints;

While this document does not define specific mechanisms, it highlights the potential requirement for fast, low-overhead notifications to convey such capability announcements across devices.

Further analysis and definition of this use case is TBD.

6. Scope of Notification Mechanism Definition

To support fast and reliable notification in network systems, it is important to clearly define the boundary of what needs to be standardized or further specified. This section identifies components that fall within the scope of the notification mechanism and those that are explicitly out of scope, in order to guide future work and maintain modularity.

6.1. Out-of-Scope Elements

The following components are considered outside the scope of the notification mechanism definition. These elements are assumed to be supported by existing technologies, protocols, or implementation practices:

* Trigger Event Mechanism: The process of detecting network events (such as link failure, persistent congestion, or threshold violations in delay/loss) and deciding when to send a notification. This function is typically handled by existing telemetry systems, performance monitoring tools, or alarm-based threshold mechanisms.

* Action Mechanism: The logic that determines and executes the response after a notification is received (e.g., traffic rerouting, congestion control, or flow prioritization). As this is deployment-specific and closely tied to the control or management plane behavior, it is outside the scope of this document.

6.2. In-Scope Aspects and Potential Work

The following aspects are considered within the scope of the Fantel notification mechanism and may require further specification:

6.2.1. Notification Format

The encoding format for notifications must be compact, extensible, and interoperable to ensure efficiency across diverse implementations. Candidate approaches include:

* TLV-Based Notification: A Type-Length-Value structure allowing flexible expression of notification content while supporting forward compatibility.

* OPAQUE-Based Notification Structur: Notification encoding structures may draw inspiration from mechanisms defined in [RFC 5250] or similar OPAQUE models used for flexible and structured signaling. Reuse or adaptation of such formats may enhance compatibility and extensibility.

6.2.2. Notification Content

Notification messages must provide enough information to convey relevant network conditions, which may include:

* Network State Information: Metrics such as interface status, delay measurements, packet loss ratios, queue depth, or congestion indicators. The applicable granularity may depend on whether the information is interface-, path-, or flow-specific.

* Optional Capability Advertisement: Nodes may include information about their supported notification handling capabilities or processing constraints to allow the receiver to make more informed decisions.

6.2.3. Notification Propagation and Scope

The delivery scope and propagation mechanism of notifications must strike a balance between speed and scalability:

* Point-to-Point (P2P): Delivery to a directly connected neighbor or designated next-hop.

* Point-to-Multipoint (P2MP): Dissemination to a selected set of nodes, for example along a service or forwarding path.

* Scoped Flooding or Domain-wide Broadcast: Delivery to all nodes in a defined region or domain. Suitable for critical events, though special attention must be paid to control overhead and duplication.

These in-scope aspects form the foundation for standardizing a modular and interoperable notification mechanism within the Fantel framework. By focusing on propagation procedures and message format while leveraging existing technologies for detection and reaction, the architecture supports fast and reliable awareness of performance-impacting events across the network.

7. Summary

This document defines the requirements for Fast Notification for Traffic Engineering and Load Balancing (FaNTEL), focusing on the role of fast notification.

It outlines how fast notification can be applied in four key areas:

* Load Balancing: Enables rapid dissemination of network state to assist in balancing traffic across multiple paths, improving utilization and responsiveness.

* Protection: Facilitates fast awareness of link or node failures, supporting quicker protection switching and reduced traffic loss.

* Flow Control: Helps inform upstream nodes of downstream congestion or performance degradation, enabling timely traffic shaping or rate adjustment.

* Capability Announcement: Supports lightweight and flexible notification of node or service capabilities (e.g., processing features or queue models), which may not be efficiently handled by existing routing protocols.

The document emphasizes core requirements such as notification message structure, delivery scope, and interoperability, which could be defined in following work, while keeping trigger detection and action logic out of scope.

8. 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

Yongqing Zhu
China Telecom
Email: zhuyq8@chinatelecom.cn

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

Weiqiang Cheng
China Mobile
Email: chengweiqiang@chinamobile.com

Chang Liu
China Unicom
Email: liuc131@chinaunicom.cn