Network Working Group
Internet-Draft
Intended status: Informational
Expires: 27 July 2026

J. Dong, Ed.
Huawei Technologies
M. McBride, Ed.
Futurewei
F. Clad, Ed.
Cisco Systems
Z. Zhang
Juniper Networks
Y. Zhu
China Telecom
X. Xu
R. Zhuang
China Mobile
R. Pang
China Unicom
H. Lu
Y. Liu
Tencent
L. Contreras
Telefonica
M. Durmus
Turkcell
R. Rahman
Equinix

23 January 2026

Fast Network Notifications Problem Statement
draft-dong-fantel-problem-statement-04

Abstract

Modern network applications, ranging from Artificial Intelligence (AI) /Machine Learning (ML) training to large-scale cloud services, require adaptive networks to ensure reliable and congestion-free data transfer within or across multiple data centers. A good and timely understanding of network operational status can help to enable faster response to critical events, so as to enable the selection of paths with reduced latency and improve network utilization. This document describes the existing problems and the need of fast network notification solutions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 27 July 2026.

Copyright Notice

Table of Contents

1. Introduction

Modern network applications, ranging from AI/ML training to large-scale cloud services, require adaptive networks to ensure reliable and congestion-free data transfer within or across multiple data centers. These workloads demand high throughput, low latency, and minimal packet loss across dynamically shifting traffic patterns so that the service continuity and performance can be maintained. To meet these requirements, networks employ mechanisms such as traffic engineering (TE), load balancing, flow control, and protection. However, existing solutions often face limitations in responsiveness, coverage, and operational complexity, particularly in high-speed, large-scale environments.

Modern forwarding silicon is capable of detecting congestion, microbursts, queue buildup and other localized impairments at fine-grained time scales, ranging from microseconds to sub-millisecond, depending on hardware capabilities and deployment requirements. These detection capabilities substantially outpace the time required for such information to be disseminated to other relevant nodes for their actions, creating a gap between what the detecting node can observe and when recipients can react. Fast network notification identifies the need for complementary mechanisms that enable low-latency notification of network conditions, allowing actions taken in the data plane, control plane or management plane to more closely align with the capabilities of contemporary forwarding hardware.

This document summarizes the limitations of existing mechanisms that prevent them being used for rapid notification of critical network events, including link or node failures and congestion. It also identifies the need for fast network notification which is critical for enabling fast reaction. In the context of this document, fast does not imply a single, rigid numerical time threshold. Instead, it characterizes a class of mechanisms to minimize the delivery time so that the latency of the notification is in the order of sub-milliseconds or milliseconds, depending on the operational objective and the range of the network domain, and can be substantially shorter than the Round-Trip-Time (RTT) of the network traffic involved.

[I-D.geng-fantel-fantel-gap-analysis] provides a gap analysis of existing solutions and where they are deficient in supporting high demand services. This document describes the set of problems which the a network notification solution needs to address.

2. Glossary

BFD: Bidirectional Forwarding Detection [RFC5880]

ECN: Explicit Congestion Notification [RFC3168]

FRR: Fast Re-Route [RFC4090] [RFC5714]

IOAM: In-situ Operations, Administration, and Maintenance [RFC9197]

3. Why Fast Network Notification is Needed

Current network mechanisms were not designed for the responsiveness and scale required by todays' dynamic environments. Techniques such as load balancing, protection switching, and flow control rely on feedback loops that are often too slow, too coarse, or too resource-intensive. This results in performance bottlenecks, delayed recovery, and inefficiencies in large-scale AI, cloud, and WAN deployments. A fast network notification mechanism could help to address these gaps by providing lightweight, real-time, actionable alerts that complement existing tools and enable faster, more accurate traffic manipulation decisions.

In particular, the detection and propagation of network events (e.g., failure, congestion or state change) must occur within a timeframe short enough to meaningfully influence traffic engineering and load-balancing decisions before congestion or micro-loops occur or develop. In backbone or datacenter networks, this typically implies a target of notification delivery in the order of milliseconds, with some environments requiring sub-millisecond performance. The precise requirement is driven by:

* the speed at which traffic shifts can induce overload

* the granularity of TE tuning (fine-grained vs. coarse-grained)

* the propagation diameter of the network notification

* the responsiveness of the control-plane and forwarding-plane components

Therefore, this document focuses on notification mechanisms capable of operating within these millisecond/sub-millisecond ranges, rather than mechanisms whose latency spans tens or hundreds of milliseconds, which are insufficient for preventing transient overload under rapid traffic transitions.

4. The Problem with Existing Notification Mechanisms

Current network traffic manipulation mechanisms such as TE, load balancing, flow control, and protection, have deficiencies in providing the low-latency, high-granularity responsiveness needed in modern, dynamic networks, at least in part due to the lack of dynamic network state information. This results in suboptimal performance, low reliability and delayed recovery. Fast network notification is a set of solutions to address this by enabling real-time, lightweight notifications that enhance the responsiveness for traffic engineering, congestion mitigation, and failure protection. There is a demonstrable need for a standardized framework to define these fast network notification mechanisms, requirements and integration strategies.

There follows a summary of the limitations of existing notification mechanisms:

* Slow Dissemination: Existing control protocols (e.g., routing protocol, etc.) may be used for dissemination of dynamic network state information, while they usually rely on control plane based hop-by-hop distribution, which causes delay when the recipient is multiple hops away. With modern high-throughput environments (AI/ML clusters, multi-DC WANs), this delay is often prohibitive. Explicit Congestion Notification (ECN) [RFC3168] needs congestion signals to be sent back to the sender, which introduces Round-Trip-Time (RTT) delay and can be slow if the source node is far away, and it relies on the source node to take action in the transport layer. What is needed is a lightweight signaling method that can provide real-time alerts (e.g., at the sub-milliseconds level or in the order of a few milliseconds) on failures, congestion, or threshold breaches, enabling prompt actions (e.g., in the range of one millisecond to tens of milliseconds) in the network layer.

* Coarse-Grained Signals: Classic ECN [RFC3168] uses a 2-bit field in packet header to convey the ECN capability and congestion indication, which inherenetly limits the information it can report to the receiving nodes. What would be useful is a set of notifications that aren’t just "on-off" state reports, but can also convey more information like congestion level/utilization information, latency spikes, queue buildup or flow characteristics, so that it can trigger immediate and precise responses like rerouting, rate adjustment, or protection switching for specific flows.

* Local-Only Decision Making: Current load-balancing, flow-control, and FRR techniques often act on local information and fail to capture downstream or cross-domain network conditions, limiting their effectiveness and leading to suboptimal decisions. For example, the Point of Local Repair (PLR) executing FRR makes its decision based on its local view of the topology and network status. It may switch traffic to a backup path and cause cascading congestion on that path, as it lacks visibility into the state of the entire backup path. Similarly, traditional load-balancing is based on local link utilization information, which may cause some paths overloaded while others remain underutilized. This local view of network status prevents precise and optimized decisions and adjustments. It would be helpful to send fast network notifications to upstream nodes so that they can perform action based on a wider view of network conditions.

* Overhead and Scalability Challenges: The distribution of high-volume network operational status information or frequent signaling introduces bandwidth and processing overhead. At scale, this becomes a bottleneck rather than a solution. IOAM [RFC9197] and similar tools provide detailed telemetry information, but the collection and feedback loops are controller-centric. They cannot be used to deliver lightweight, real-time alerts for immediate action on specific network nodes. Carrying dynamic network state information in control protocols (e.g., routing protocols) also increases the overhead and churn of the control plane, which may have negative impact to the core functionality of the protocol. It would be useful to have solutions designed to avoid the overhead and churn introduced by telemetry flooding or route distribution, so it can adapt to large-scale networks and dynamic traffic patterns (e.g., AI workloads, cloud WAN bursts).

4.1. Example: AI Training Cluster with Fiber Link Failure

Consider a large-scale AI training job distributed across multiple data centers. These clusters exchange terabits of data per second between Graphics Processing Unit (GPU) nodes, requiring ultra-low latency and high throughput to maintain synchronization.

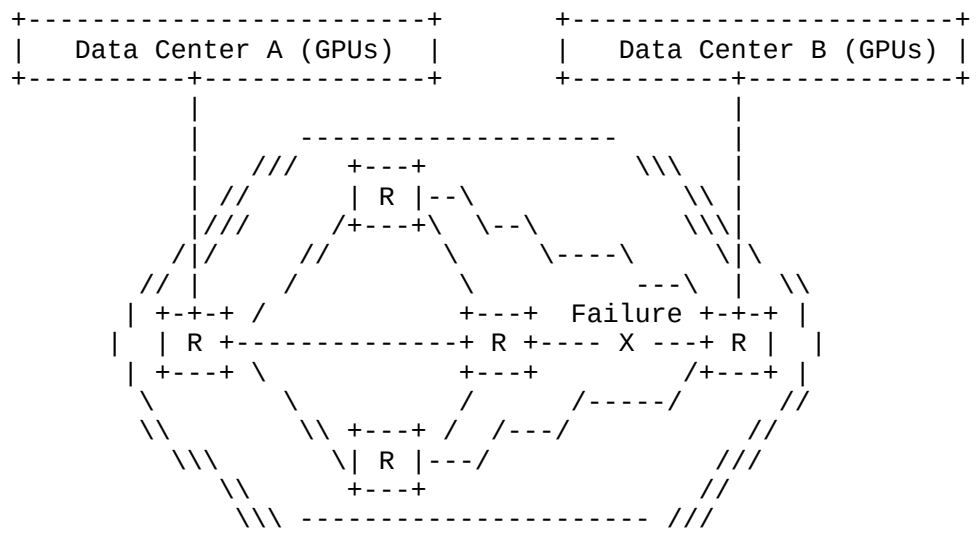

Figure 1: Distributed AI Training Clusters with Fiber Link Failure

As depicted in the above figure, a single fiber link failure event can disrupt the entire training run, leading to:

* Delays in job completion (hours to days for large models)

* Massive energy and compute cost waste due to resynchronization

* Degraded convergence accuracy if synchronization windows are missed

4.1.1. Limitations of Existing Mechanisms

Today’s mechanisms provide partial solutions but are not fast or precise enough for these scenarios:

* BFD [RFC5880]: Provides fast faults detection in the bidirectional path between two forwarding engines. BFD can be one of the detection mechanisms for link or path failures, while it is not used to notify the failure to nodes other than the BFD endpoints in the network. BFD is preconfigured with periodic message exchange, while fast network notifications needs to be event-driven.

* FRR [RFC4090][RFC5714] /Route convergence: Without fast notification, the failure detection can take tens of milliseconds, followed by either local repair (FRR) or route convergence. The former lacks visibility of the global network situation and thus

may cause congestion on the backup paths, while the latter may breach strict synchronization requirements of the AI/ML application.

In practice, this means that by the time a fiber link failure is detected and recovery mechanisms are invoked, critical GPU synchronization barriers may already have been missed, forcing rollbacks or restarts of the training process.

4.1.2. How Fast Network Notifications Help

Fast network notification mechanisms could improve the response to fiber link failures and congestion in distributed AI/ML clusters:

* Real-Time Alerts: Nodes adjacent to the failure or congestion could immediately (e.g., in the order of sub-milliseconds or milliseconds) send lightweight notifications to nodes whose fowarding paths might be affected.

* Action-Oriented Response: Upon receiving the notification, routing and load balancing mechanisms could instantly shift traffic to backup paths or alternative DC interconnects.

* Granularity: Notifications could carry more detailed information than "link failure/congestion," e.g., indicating specific link utilization, queue buildup or microburst congestion, allowing differentiated responses to different traffic flows.

* Complementary: The fast notification solutions are complementary to BFD, FRR or Telemetry, it would bridge the time gap between event onset and slower control plane or telemetry-driven responses, and enable network-wide optimization.

By deploying fast notifications, large AI/ML workloads can maintain synchronization across data centers even during transient failures or congestion, protecting job completion time and resource utilization.

Existing Approach:

* BFD detects failure after tens of ms

* FRR may cause congestion on backup paths

* Reroute/convergence delays impact GPU sync

* Result: Training stalls, compute resources wasted, job completion delayed

Fast Notifications Approach:

* Forwarding plane detects failure at the level of sub-millisecond

* Fast network notification alerts upstream nodes of failure or congestion in real time

* Regional or global TE steers traffic quickly to alternate link/ path without causing new congestion

* Result: Training continues with minimal disruption

5. Fast Network Notifications Problem Statement

5.1. Information of Fast Network Notifications

The information carried in the fast network notifications, by the originating node, can be one or multiple of the following:

* Event Type: This can be used to indicate the type of events (e.g. failure, congestion, performance degradation, etc.).

* Location of Event: This can be used to indicate the location where the event occurred in the network (e.g. the identifier of the link, the node, or the queue, etc.).

* Fine-grained Network Status information: This can include quantifiable network metrics like link utilization, queue length, level of congestion, link or node delay, jitter, packet loss, etc.

* Path Identification information: This can be used to indicate the path which is affected by the event.

* Flow Identification information: This can include the identification or the 5-tuple of a flow which is affected by the event.

Other information related to the network status change and need to be actioned in a timely manner may also be carried in the fast network notifications. Thus there is a need to work on the information model of fast network notifications to better understand what needs to be carried in the notifications.

5.2. Recipients of Fast Network Notifications

Fast network notifications may be consumed by two broad forms of recipient: (1) recipient nodes that participate directly in forwarding or signaling, and (2) functions and applications that consume notifications in order to optimize, monitor, or adapt behaviors as depicted in the following two tables. Separating these categories clarifies which entities are physical/logical nodes versus which are higher-level functional consumers.

Node Type	Role	Example Benefit
Adjacent Routers / Switches	Data-plane neighbors that forward packets	Enable local repair (e.g., FRR, ECMP adjustments)
Non-Adjacent Routers / Switches	Remote upstream forwarding elements	Accelerated awareness of failure/congestions on specific nodes
Ingress Routers / Switches	Traffic entry points of a network domain	Re-map affected flows before forwarding into failed regions
End Hosts / Edge Nodes	Optional subscribers, policy-driven	Adapt sending rate, select alternate uplinks
Network Controller	Optional subscribers, policy-driven	Accelerated awareness of failure/congestion for global TE/LB

Table 1: Recipient Nodes

Function / Application	Role	Example Benefit
Routing Protocols (OSPF, IS-IS, BGP)	Control-plane convergence	Accelerated path re-computation after failure
Traffic Engineering Element (PCE)	Centralized optimization	Pre-compute new paths before congestion propagates
Network Operators (NMS/OSS)	Operational visibility	Faster troubleshooting, earlier alerting
Telemetry / Analytics Systems	Monitoring and prediction	Predictive analytics, ML-based congestion forecasting
Applications / Services	Critical app consumers	AI workloads, financial apps adapt to degraded links

Table 2: Recipient Functions and Applications

The tables have three columns. The fist column lists the type or node or type of application/function. The second shows the example of the role that the node or application/function is responsible for within the network that could benefit from fast network notifications. The third column indicates examples of how fast notification could benefit the node/application/function in filling its role.

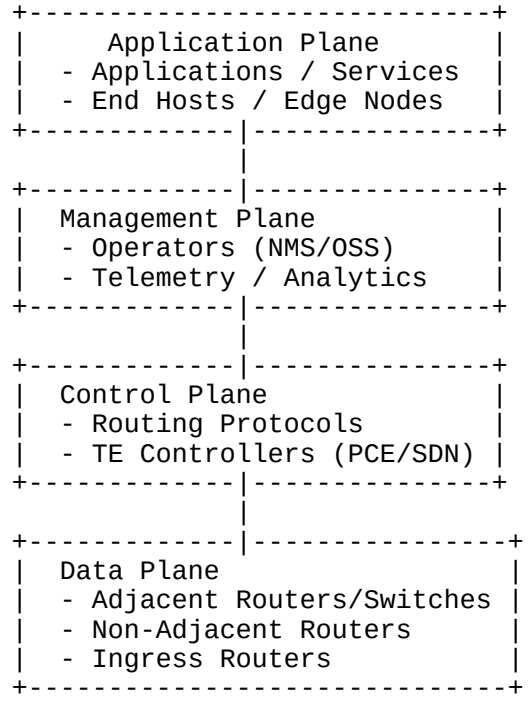

Figure 2: Notification Recipients Across Network Planes

As illustrated in Figure 2, the latency sensitivity of recipients decreases as one moves from the data plane to the application plane. Recipient nodes (e.g., adjacent forwarding elements, ingress routers, etc.) often require very quick notification, while functions and applications (e.g., routing protocols, analytics systems, NMS, etc.) may tolerate slightly longer timescales but still benefit from rapid awareness compared to existing mechanisms. The range of recipients of the notification depends on the type of recipients, it also depends on what type of action is required. The mechanism to determine the type and range of the recipients is something that needs further consideration.

5.3. Delivery of Fast Network Notifications

Depending on the position and number of the recipient nodes, fast network notifications may be sent via one of the following delivery modes:

* Unicast directly to the recipient node

* Multicast to a group of recipient nodes

* Hop-by-hop to a series of receipt nodes along a specified path

* Flooding in a specified range of the network

Additionally, recipient nodes or functions may subscribe to specific types of notifications based on their roles or interests. A subscription-based approach enables selective delivery, reduces unnecessary signaling overhead, and ensures that each recipient receives only the information relevant to its function. Mechanisms supporting both delivery and subscription must guarantee timely, reliable, and secure propagation of notifications. Examples:

* Adjacent routers subscribing to all local failure notifications

* Centralized controllers subscribing only to congestion alerts exceeding defined thresholds

* Applications or analytics systems subscribing to performance degradation events affecting specific flows or services

The mechanisms to support the above delivery mode needs to make sure the notification is always sent to the targeted recipient nodes in a timely manner. It could be based on existing messaging and transport mechanisms, or a new protocol may be introduced.

5.4. Actions to Fast Network Notifications

Once a fast network notification is received, the recipient needs to take appropriate actions to help mitigating the event reported in the fast network notification. The action can be based on the information carried in the fast network notification, or it can be based on both the information in the notification and the information obtained by the recipient in other ways. The action to be performed by the recipient may be explicitly carried in the notification, or it may be implicitly determined by the type of information carried in the notification. Some actions are mandatory, while some actions can be optional. The possible actions in response to the notification can be, but not limited, to one or multiple of the following:

* Switches all traffic from a path to other available paths

* Steers specific traffic flows to alternate links or paths

* Modifies the load balancing ratio among a group of paths

* Sends the notification further to other recipients

Whether the actions need to be explicitly indicated in the notification, and if so, which ones, requires further consideration. It is noted that in some of the cases as described in Section 5.2, multiple recipients may receive the same notification, then some action may be taken by multiple recipients. The sender of the fast network notification needs to take this into consideration if some coordination in the actions is needed. The mechanism for action coordination is for further study and is out of the scope of this document.

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

Fast network notifications, if not properly authenticated and rate-limited, could be exploited as a vector for Denial-of-Service (DoS) attacks. An attacker able to inject or flood spurious notifications may trigger unnecessary re-convergence, path changes or repeated state updates, overwhelming both recipient nodes and higher-level applications. An attacker may cause the sender of fast network notifications overwhelmed by making some network state flapping, so that the node is busy with sending notifications. Fast network notifications may reveal sensitive information about the network, in some scenarios such information may be made visible to external entities, either by inspecting the notifications, or by registering as a consumer of the notifications. Implementations must therefore ensure integrity protection, origin authentication, and appropriate rate controls on sending and receiving fast network notification messages.

8. Acknowledgement

The authors would like to thank Alia Atlas, David Black, Jeffrey Haas, Tony Li, Carlos J. Bernardos, Fan Zhang and Adrian Farrel for their valuable comments and discussion.

9. Contributors

The following people contributed substantially to the content of this document.

Zafar Ali
Cisco
zali@cisco.com

Tianran Zhou
Huawei
zhoutianran@huawei.com

Xuesong Geng
Huawei
gengxuesong@huawei.com

10. References

10.1. 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>.

[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>.

10.2. Informative References

[I-D.geng-fantel-fantel-gap-analysis] Geng, X., Huo, P., Cheng, W., Li, D., Zhu, Y., and H. Zhengxin, "Gap Analysis of Fast Notification for Traffic Engineering and Load Balancing", Work in Progress, Internet-Draft, draft-geng-fantel-fantel-gap-analysis-01, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-geng-fantel-fantel-gap-analysis-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>.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <https://www.rfc-editor.org/info/rfc4090>.

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <https://www.rfc-editor.org/info/rfc5714>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi, Ed., "Data Fields for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197, May 2022, <https://www.rfc-editor.org/info/rfc9197>.

Authors' Addresses

Jie Dong (editor)
Huawei Technologies
Email: jie.dong@huawei.com

Mike McBride (editor)
Futurewei
Email: mmcbride7@gmail.com

Francois Clad (editor)
Cisco Systems
Email: fclad@cisco.com

Jeffrey Zhang
Juniper Networks
Email: zzhang@juniper.net

Yongqing Zhu
China Telecom
Email: zhuyq8@chinatelecom.cn

Xiaohu Xu
China Mobile
Email: xuxiaohu_ietf@hotmail.com

Rui Zhuang
China Mobile
Email: zhuangruiyjy@chinamobile.com

Ran Pang
China Unicom

Email: pangran@chinaunicom.cn

Hao Lu
Tencent
Email: vickkylu@tencent.com

Yadong Liu
Tencent
Email: zeepliu@tencent.com

Luis M. Contreras
Telefonica
Email: luismiguel.contrerasmurillo@telefonica.com

Mehmet Durmus
Turkcell
Email: mehmet.durmus@turkcell.com.tr

Reshad Rahman
Equinix
Email: reshad@yahoo.com