Computing-Aware Traffic Steering   Q. Li
Internet-Draft   H. Wang
Intended status: Informational   Pengcheng Laboratory
Expires: 3 September 2026   Y. Jiang
  Tsinghua Shenzhen International Graduate School & Pengcheng Laboratory
                                                               M. Xu
                                                  Tsinghua University
                                                        2 March 2026

A Framework of Intelligence Delivery Network (IDN) for Deep Learning Inference
draft-li-cats-idn-00

Abstract

The rapid growth of deep learning inference workloads is placing increasing pressure on existing Internet and computing infrastructures. To support latency-aware, privacy-enhanced, and scalable inference services, this document introduces the concept of Intelligence Delivery Network (IDN), in which models with different inference capabilities are deployed across geographically distributed servers and selected to serve inference requests. This document describes the challenges motivating such networks, presents an architectural framework, and defines a common vocabulary for discussing the systems. This document does not specify protocol details, which are left to future documents.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://kongyanye.github.io/draft-li-cats-idn/draft-li-cats-idn.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-li-cats-idn/.

Discussion of this document takes place on the Computing-Aware Traffic Steering Working Group mailing list (mailto:cats@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/cats/. Subscribe at https://www.ietf.org/mailman/listinfo/cats/.

Source for this draft and an issue tracker can be found at https://github.com/kongyanye/draft-li-cats-idn.

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 3 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 3
2. Background and Challenges 4
2.1. Characteristics of Deep Learning Inference Workloads . . 4
2.2. Limitations of Existing Computing Infrastructures 4
2.3. Emerging Capabilities in Model Deployment 5
2.4. Architectural Challenges 5
3. Intelligence Delivery Network Framework 6
3.1. Design Principles 6
3.2. High-Level Architecture 7
3.3. Inference Capability Representation 8
3.4. Capability Placement and Distribution 8
3.5. Inference Request Handling 9
3.6. Caching and Capability Reuse 9
3.7. Model Evolution and Lifecycle 10
4. Terminology 10

5. Security, Privacy, and Trust Considerations 12
5.1. Data Privacy and Locality 12
5.2. Model Integrity and Authenticity 13
5.3. Trust Across Administrative Domains 13
5.4. Inference Result Reuse and Isolation 13
5.5. Availability and Abuse Considerations 14
6. Acknowledgments 14
7. Informative References 14
Authors' Addresses 14

1. Introduction

The increasing deployment of deep learning models has led to rapid growth in inference workloads across the Internet. Unlike model training, which is typically performed in centralized data centers using batch-oriented processing, inference workloads are often latency-sensitive, geographically distributed, and closely coupled with user data. These characteristics introduce new requirements that were not primary design considerations in earlier Internet and computing architectures.

Current approaches to deep learning inference largely rely on centralized or regionally centralized cloud infrastructures. In such systems, user data is transmitted to a limited number of locations where inference is performed. While effective for certain applications, this model can introduce challenges related to end-to-end latency, scalability, and privacy [RFC9556]. As inference demand continues to grow and applications increasingly require real-time responses, these limitations become more significant.

At the same time, advances in model compression, quantization, distillation, and specialization have enabled inference capabilities to be represented in models of varying size and complexity. As a result, it has become feasible to deploy different models at different locations in the network, ranging from large, general-purpose models in cloud data centers to smaller, task-oriented models on edge devices. These developments motivate a shift from data-centric inference processing toward a model-centric approach to inference delivery.

Inspired by the architectural principles of Content Delivery Networks (CDNs), this document introduces the concept of Intelligence Delivery Network (IDN). An IDN is a network architecture in which inference capabilities, encoded in trained deep learning models, are deployed across a set of interconnected nodes. Analogous to how CDNs cache content closer to users to improve delivery performance, IDNs place model-encoded intelligence closer to inference request sources in order to reduce latency and improve scalability. In this framework,

inference requests are served by appropriate model instances based on factors such as task requirements, locality, and system conditions, rather than being uniformly directed to a single centralized location.

The remainder of this document is organized as follows. Section 2 provides background and discusses the challenges motivating IDNs. Section 3 presents the IDN architectural framework. Section 4 defines the terminology used in IDNs and explains the relationships among these terms. Section 5 discusses security, privacy, and trust considerations.

2. Background and Challenges

This section provides background on deep learning inference deployment and identifies challenges that motivate the Intelligence Delivery Network (IDN).

2.1. Characteristics of Deep Learning Inference Workloads

Deep learning inference workloads differ from traditional Internet services and deep learning training workloads in several important aspects. First, inference requests are often generated by interactive applications (e.g., conversational interfaces and programming assistants) and therefore tend to be latency-sensitive. Second, they are geographically distributed, reflecting the locations of end users and data sources. Third, inference usually operates directly on user-generated or user-specific data, increasing sensitivity to data locality and privacy requirements.

Inference workloads are also heterogeneous [Elf]. Different applications impose different requirements in terms of model accuracy, response time, resource consumption, and availability. A single inference service may involve a mix of simple tasks that can be handled by lightweight models and more complex tasks that require larger or more capable models. This diversity complicates uniform deployment and execution strategies.

2.2. Limitations of Existing Computing Infrastructures

Today, large-scale deep learning models, exemplified by large language models (LLMs), have become dominant solutions across a wide range of application domains. Due to their substantial computational and memory requirements, these models are typically deployed in centralized or regionally centralized cloud infrastructures. In this deployment paradigm, inference requests and associated data are transmitted from clients to a limited number of data centers where the models are hosted and executed. This approach benefits from

operational simplicity and centralized management, and it aligns well with existing cloud computing practices.

However, centralized inference deployment also exhibits limitations. Routing all inference requests to a limited set of locations can increase end-to-end latency, particularly for users far from data center locations. Centralized processing may create scalability bottlenecks during demand spikes, and it can increase network traffic as raw data has to be transmitted over wide-area networks. In addition, transferring user data to centralized locations may raise privacy, regulatory, or policy concerns in certain environments.

2.3. Emerging Capabilities in Model Deployment

Recent advances in deep learning have enabled inference capabilities to be packaged in models of varying size, complexity, and specialization. Techniques such as model compression, quantization, distillation, and task-specific fine-tuning allow smaller models to approximate the behavior of larger models for specific tasks or domains. These developments make it possible to deploy inference models on a broader range of platforms and locations, including regional servers and edge devices.

As a result, inference capability is no longer inherently tied to a small number of large data centers. Instead, it can be distributed across multiple layers of the network, with different models providing different levels of capability and performance. This flexibility creates opportunities for architectures that place inference closer to data sources or users while retaining access to more capable models when needed.

2.4. Architectural Challenges

Despite these advances, several architectural challenges remain unaddressed by existing deployment paradigm:

* Capability Placement: Determining where different inference capabilities should be deployed to balance latency, accuracy, resource usage, and operational cost.

* Request Selection and Steering: Selecting appropriate model instances for individual inference requests based on task requirements, locality, and system conditions.

* Scalability: Supporting large and dynamic inference demand without introducing centralized bottlenecks.

* Data Locality and Privacy: Limiting unnecessary data movement while meeting privacy and regulatory requirements.

* Operational Heterogeneity: Managing inference services across diverse hardware platforms, network conditions, and administrative domains.

These challenges motivate the need for a new architectural framework that treats inference capability as a distributable, cacheable, and selectable entity within the network, rather than binding inference execution to a small number of centralized locations.

3. Intelligence Delivery Network Framework

This section presents the architectural framework for Intelligence Delivery Network (IDN). The framework describes how inference capabilities are organized, deployed, and selected across interconnected nodes, and how these components relate to one another at a conceptual level.

3.1. Design Principles

The IDN framework is guided by the following design principles:

* Capability-Centric Delivery: Inference capability, encoded in trained models, is treated as a first-class entity that can be deployed, selected, and reused independently of specific compute resources.

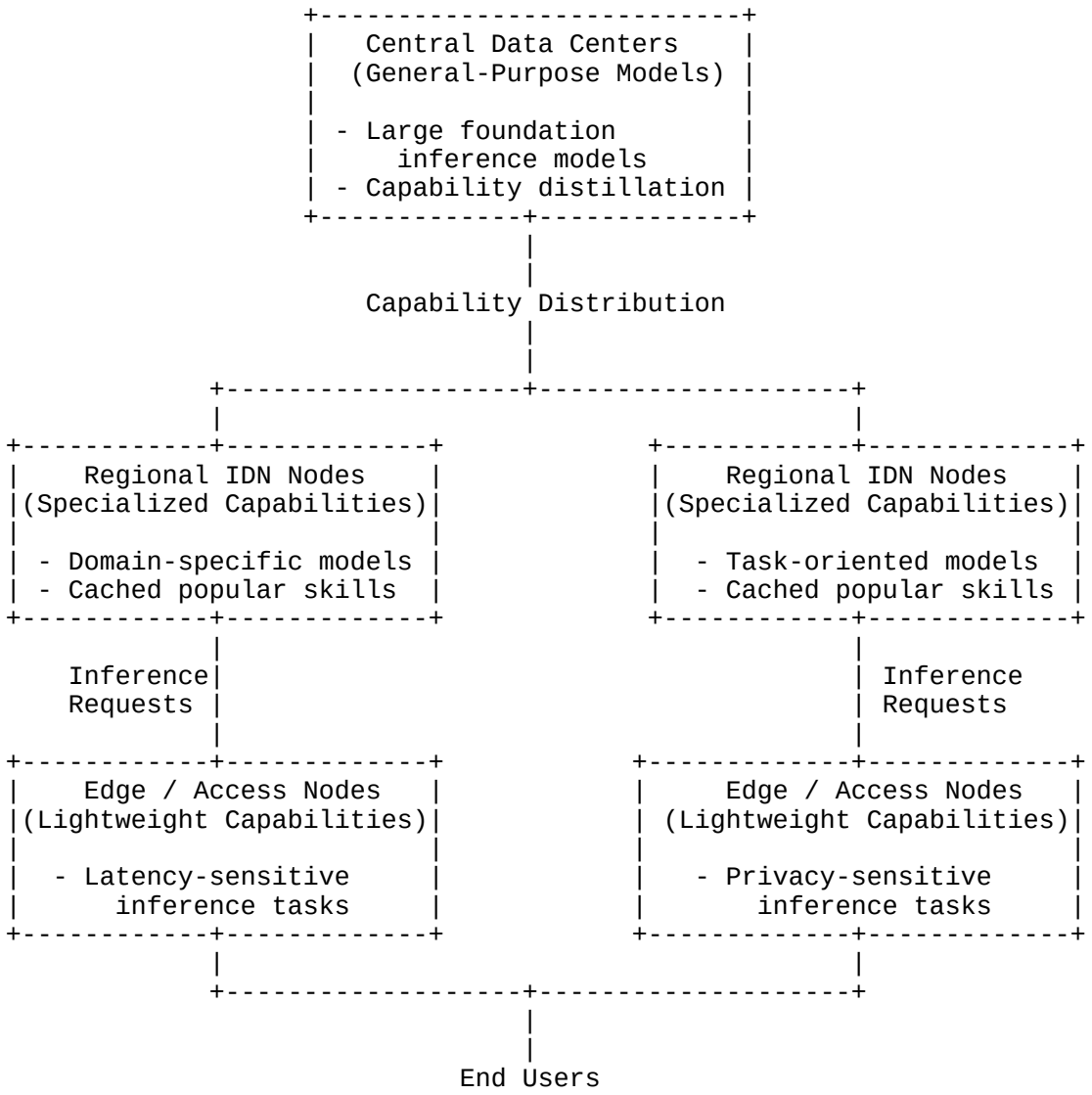
* Hierarchical and Distributed Deployment: Inference capabilities are deployed across multiple layers of the network, including centralized data centers, regional infrastructure, and edge devices, to balance performance, cost, and scalability.

* Locality Awareness: Placement and selection of inference capabilities should account for network proximity, data locality, and latency sensitivity.

* Reuse and Efficiency: The framework encourages reuse of inference results, intermediate representations, and distilled capabilities to reduce redundant computation and unnecessary data movement.

* Incremental Evolution: The framework accommodates continuous evolution of inference capabilities, including updates, specialization, and replacement, without requiring global redeployment.

3.2. High-Level Architecture



At a high level, an IDN consists of a set of interconnected nodes capable of hosting and executing inference models. These nodes may belong to different administrative domains and may operate under diverse hardware and network conditions. Each node may host one or more inference capabilities, represented by trained models.

Inference requests enter the IDN from clients or upstream systems and are directed to appropriate nodes based on capability requirements and operational considerations. Rather than uniformly forwarding requests to a centralized location, the IDN enables selection among multiple candidate nodes that offer suitable inference capabilities.

The architecture allows inference capabilities to be replicated, cached, or specialized at different locations. Popular or frequently invoked capabilities may be deployed closer to request sources, while less frequently used or more complex capabilities may remain centralized.

3.3. Inference Capability Representation

In the IDN framework, inference capability refers to the ability of a model to perform a particular class of inference tasks with defined accuracy, latency, and resource characteristics. Capabilities may differ in scope and complexity, ranging from general-purpose models to highly specialized or task-specific models.

Inference capabilities may be derived from larger models through techniques such as distillation, compression, fine-tuning, or specialization. In addition, analysis of usage patterns across users or applications may identify frequently invoked tasks or “hot” capabilities. Such capabilities can be decomposed or extracted from more general models and represented as smaller, specialized models that retain task-level accuracy while reducing computational and memory requirements.

Each inference capability is associated with a set of descriptive attributes, such as supported tasks, expected resource usage, accuracy characteristics, performance profiles, and update frequency. These attributes provide an abstract description of the capability independent of its deployment location and serve as inputs to placement, distribution, and request selection decisions within the IDN.

3.4. Capability Placement and Distribution

The IDN framework separates the concept of capability placement from inference execution. Placement refers to decisions about where inference capabilities are deployed within the network, while execution refers to the processing of individual inference requests by deployed capability instances.

Based on capability attributes and observed demand, capabilities may be proactively deployed in anticipation of expected usage or dynamically replicated in response to changing access patterns.

Frequently used or latency-sensitive capabilities may be placed closer to inference request sources, such as regional or edge nodes, while less frequently used or more resource-intensive capabilities may remain at centralized locations.

The ability to represent popular or task-specific capabilities as smaller models enables selective distribution of such capabilities to locations closer to users. This approach allows the framework to reduce inference latency and resource consumption while preserving access to larger, more general models when higher capability is required. The placement and distribution of inference capabilities are conceptually analogous to the distribution of popular content in Content Delivery Networks (CDNs).

The framework allows multiple capability instances providing the same task to exist at different locations, potentially with different operational characteristics, enabling flexible trade-offs among performance, cost, and scalability.

3.5. Inference Request Handling

When an inference request is issued, the IDN framework enables selection of an appropriate capability instance based on factors such as task requirements, locality, availability, and current system conditions. This selection may involve an explicit or implicit request resolution step, in which a client or intermediary is directed to an appropriate IDN node that hosts a suitable inference capability. Such resolution may be performed using existing Internet mechanisms or application-layer services.

The framework does not prescribe specific routing or selection mechanisms. Instead, it defines the architectural context in which such mechanisms operate, enabling flexible request steering across distributed capability instances.

3.6. Caching and Capability Reuse

To improve efficiency and scalability, the IDN framework supports caching and reuse at multiple levels. Inference results or intermediate representations may be reused across similar requests, including requests from different users, when appropriate [KVShare]. This reuse can reduce repeated computation for common or popular inference tasks.

Caching and reuse decisions are influenced by factors such as workload characteristics, resource constraints, and consistency requirements.

3.7. Model Evolution and Lifecycle

Inference capabilities within an IDN are expected to evolve over time. Models may be updated, replaced, refined, or specialized as new data becomes available, as usage patterns change, or as application requirements evolve. The framework supports incremental updates and the coexistence of multiple capability versions, enabling gradual transitions rather than requiring global or disruptive replacements.

Model evolution may be driven by multiple sources. Updates can be produced centrally, for example through cloud-side retraining, refinement, or distillation of models, and subsequently distributed to appropriate locations within the IDN. In addition, where permitted by policy and regulatory constraints, inference capabilities deployed at edge or near-user locations may be locally adapted using user-provided or locally observed data. Such local adaptation may follow federated or privacy-preserving learning approaches, in which locally derived updates contribute to global model improvement without requiring raw data to leave the local environment.

Lifecycle management of inference capabilities includes deployment, update, versioning, deprecation, and removal. Different versions of a capability may coexist at the same or different locations, allowing the framework to balance stability, performance, and innovation. While this document does not specify how lifecycle management processes are implemented, it assumes that mechanisms for controlled rollout, compatibility management, and rollback are necessary to ensure operational stability and consistency within an IDN.

4. Terminology

This section defines the terminology used throughout this document. The terms defined here are intended to provide a common vocabulary for discussing IDNs.

Phrases in upper-case refer to other defined terms.

CACHE AND REUSE

CACHE AND REUSE refers to mechanisms that reduce redundant inference execution by reusing inference results, intermediate representations, or derived capabilities across multiple requests or users. Caching may apply to complete inference outputs or to partial results, depending on workload characteristics and policy constraints.

CAPABILITY DISTRIBUTION

CAPABILITY DISTRIBUTION refers to the dissemination of INFERENCE CAPABILITIES or their derived forms across IDN nodes. Distribution may involve replication, specialization, or relocation of capabilities to support demand and performance objectives.

CAPABILITY PLACEMENT

CAPABILITY PLACEMENT refers to the process of determining where INFERENCE CAPABILITIES or MODEL INSTANCES are deployed within an IDN. Placement decisions may consider factors such as demand, locality, resource availability, and operational cost.

IDN

Intelligence Delivery Network. IDN is a network architecture in which INTELLIGENCE, encoded in deep learning models, is deployed, distributed, and selected across interconnected nodes to serve INFERENCE REQUESTS. An IDN enables INFERENCE CAPABILITIES to be placed at different locations in the network and selected based on task requirements, locality, and operational conditions.

IDN NODE

A network-accessible entity that hosts one or more MODEL INSTANCES and is capable of executing INFERENCE REQUESTS. IDN nodes may be located in cloud data centers, regional infrastructure, or edge environments and may operate under different administrative domains.

INFERENCE CAPABILITY

The ability of a model to perform a defined class of inference tasks with specified accuracy, performance, and resource characteristics. INFERENCE CAPABILITY may be provided by a single model or by a set of related model artifacts derived from a common source.

INFERENCE REQUEST

A request to perform inference using a specified or implied INFERENCE CAPABILITY on provided input data. An INFERENCE REQUEST may include task-specific parameters, quality requirements, or constraints relevant to capability selection.

INFERENCE REQUEST STEERING

The process of selecting and directing an INFERENCE REQUEST to an appropriate MODEL INSTANCE within an IDN. Steering decisions may take into account task requirements, proximity, system load, and policy constraints.

INTELLIGENCE

INTELLIGENCE refers to INFERENCE CAPABILITY encoded in a trained deep learning model. This includes the ability to perform specific tasks, reasoning functions, or predictions based on input data. In this document, INTELLIGENCE is treated as a distributable and reusable capability.

MODEL EVOLUTION

MODEL EVOLUTION describes the process by which INFERENCE CAPABILITIES change over time, including updates, specialization, replacement, or deprecation of models. Model evolution may result in multiple versions of a capability coexisting within an IDN.

MODEL INSTANCE

A deployed realization of an INFERENCE CAPABILITY at a specific network node. Multiple MODEL INSTANCE may provide the same INFERENCE CAPABILITY and may differ in operational characteristics such as latency, capacity, or availability.

5. Security, Privacy, and Trust Considerations

IDNs introduce new security, privacy, and trust considerations by distributing inference capabilities and processing across multiple network locations and administrative domains. This section discusses these considerations at an architectural level.

5.1. Data Privacy and Locality

Inference workloads frequently operate on user-generated or user-specific data, which may be sensitive in nature. Transmitting such data across the network to centralized locations can raise privacy, regulatory, or policy concerns. IDNs may mitigate some of these concerns by enabling inference to be performed closer to data sources, thereby reducing unnecessary data movement.

However, distributing inference capabilities across multiple nodes also increases the number of locations where data may be processed. This expansion of the processing surface requires careful consideration of data handling policies, access controls, and compliance with applicable regulations. The IDN framework does not mandate specific privacy mechanisms but assumes that privacy requirements influence capability placement and request steering decisions.

5.2. Model Integrity and Authenticity

Inference capabilities in an IDN are encoded in trained models that may be distributed, replicated, or derived across the network. Ensuring the integrity and authenticity of these models is critical, as tampered or malicious models could produce incorrect or harmful inference results.

Architectural considerations include the ability to verify that a capability instance corresponds to an authorized and unmodified model, and to ensure that model updates or derived capabilities originate from trusted sources. The framework assumes the existence of mechanisms to support model provenance and integrity, but does not specify how such mechanisms are implemented.

5.3. Trust Across Administrative Domains

IDNs may span multiple administrative domains, particularly when inference capabilities are deployed across cloud, edge, and user-managed environments. In such scenarios, each administrative domain may operate under distinct operational policies, trust assumptions, and security controls. As a result, inference requests and inference capabilities may traverse trust boundaries, requiring explicit consideration of trust relationships among participating entities.

Trust considerations include determining which nodes are permitted to host or execute specific inference capabilities, which entities are authorized to distribute or update models, and under what conditions inference requests may be forwarded across domains. The IDN framework assumes that trust relationships influence capability distribution and request steering, but does not define trust establishment or enforcement mechanisms.

5.4. Inference Result Reuse and Isolation

Caching and reuse of inference results or intermediate representations can improve efficiency and scalability, but may introduce security and privacy risks if not properly managed. Reuse across different users or contexts may lead to unintended information disclosure or inference of sensitive attributes.

Architectural considerations include ensuring appropriate isolation between inference contexts, defining conditions under which reuse is permissible, and preventing cross-user data leakage. The framework treats reuse as an optional optimization whose applicability depends on workload characteristics and policy constraints.

5.5. Availability and Abuse Considerations

As inference capabilities become more widely distributed, IDN nodes may be exposed to abuse, misuse, or denial-of-service attacks. Concentration of popular capabilities at specific nodes may create attractive targets for attacks that aim to degrade service availability.

The IDN framework recognizes the need to consider resilience and robustness in the presence of such threats, including the ability to distribute load, replicate capabilities, or redirect requests in response to failures or attacks.

6. Acknowledgments

The authors would like to thank colleagues and reviewers in the community who provided feedback on the early version of this draft.

7. Informative References

[Elf] Zhang, W., He, Z., Liu, L., Jia, Z., Liu, Y., Gruteser, M., Raychaudhuri, D., and Y. Zhang, "Elf: accelerate high-resolution mobile deep vision with content-aware parallel offloading", Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, DOI 10.1145/3447993.3448628, September 2021, <https://dl.acm.org/doi/abs/10.1145/3447993.3448628>.

[KVShare] Yang, H., Zhang, R., Huang, M., Wang, W., Tang, Y., Li, Y., Liu, Y., and D. Zhang, "KVShare: An LLM Service System with Efficient and Effective Multi-Tenant KV Cache Reuse", arXiv preprint arXiv:2503.16525, May 2025, <https://arxiv.org/abs/2503.16525>.

[RFC9556] Hong, J., Hong, Y., de Foy, X., Kovatsch, M., Schooler, E., and D. Kutscher, "Internet of Things (IoT) Edge Challenges and Functions", RFC 9556, DOI 10.17487/RFC9556, April 2024, <https://www.rfc-editor.org/rfc/rfc9556>.

Authors' Addresses

Qing Li
Pengcheng Laboratory
Email: liq@pcl.ac.cn

Hanling Wang
Pengcheng Laboratory

Email: wanghl03@pcl.ac.cn

Yong Jiang
Tsinghua Shenzhen International Graduate School & Pengcheng Laboratory
Email: jiangy@sz.tsinghua.edu.cn

Mingwei Xu
Tsinghua University
Email: xumw@tsinghua.edu.cn