

ccamp  
Internet-Draft  
Intended status: Informational  
Expires: 3 September 2026

Q. Hu, Ed.  
Z. Han  
W. Wang  
J. Zhang  
Y. Zhao  
Beijing University of Posts and Telecommunications  
Y. Tan, Ed.  
Y. Zheng  
China Unicom  
2 March 2026

A Control Framework for Unified Optical Networks and AI Computing  
Orchestration (UONACO)  
draft-hu-ccamp-uonaco-control-framework-01

## Abstract

This document presents the control framework for Unified Optical Networks and AI Computing Orchestration (UONACO). Specifically, it defines the AI computing service model over wide-area networks, outlines the UONACO control architecture, identifies a set of UONACO components and interfaces, and describes their interactions.

## 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 . . . . .	2
1.1. Requirements Language . . . . .	3
2. Service Model for AI Computing over Optical Network . . . . .	3
2.1. Customer . . . . .	4
2.2. Service Provider . . . . .	4
2.3. Network Provider . . . . .	4
2.4. Computing Power Provider . . . . .	4
3. UONACO Control and Management Architecture . . . . .	5
3.1. Overview . . . . .	5
3.2. Service Orchestrator . . . . .	6
3.3. Unified Compute-Optical Orchestrator . . . . .	6
3.4. Transport Network Controller . . . . .	6
3.5. Computing Power Scheduler . . . . .	7
3.6. UONACO Interfaces . . . . .	7
4. IANA Considerations . . . . .	8
5. Security Considerations . . . . .	8
6. References . . . . .	8
6.1. Normative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

Distributed AI computing has become a dominant paradigm for delivering large-scale AI services, enabling providers to meet stringent performance and scalability requirements by leveraging geographically dispersed AI data centers (AIDCs). In such environments, the efficiency of distributed training, inference, and remote service access depends critically on tight coordination between optical transport networks and compute orchestration systems.

However, today's infrastructure operates with fundamentally isolated control planes: the optical transport layer, despite providing the high-bandwidth, low-latency, and deterministic backbone for wide-area AI collaboration, remains blind to the dynamic, heterogeneous demands of AI workloads. It cannot discern whether a traffic flow stems from a bandwidth-intensive distributed training job requiring synchronized all-reduce operations across thousands of GPUs, or from a latency-

critical inference request demanding sub-10ms end-to-end response. Consequently, optical networks provision static or best-effort lightpaths without adapting to the real-time compute intent, leading to underutilized spectral resources or, worse, congestion-induced stalls during critical gradient synchronization phases.

Conversely, AI compute schedulers (e.g., Kubernetes-based orchestrators in AIDCs) make placement decisions based solely on local GPU/CPU availability and memory capacity, with no awareness of the underlying optical fabric's state, such as available wavelength continuity, end-to-end propagation delay, per-link bandwidth headroom, or even the presence of OXC-based reconfigurable paths. As a result, a training job may be split across geographically distant AIDCs with abundant but poorly interconnected GPU pools, causing prolonged communication phases and severe "compute efficiency loss." Similarly, a low-latency inference service might be deployed in a remote AIDC simply because it has idle GPUs, even though the optical path violates the application's SLA due to high round-trip delay or lack of dedicated wavelength isolation.

To address these challenges, this document introduces the Unified Optical Networks and AI Computing Orchestration (UONACO) framework. UONACO establishes a unified control architecture that enables bidirectional signaling, joint resource modeling, and synchronized orchestration between the compute and optical domains. The framework supports three representative service models: AI training, AI inference, and accessing remote AI inference services. By aligning network provisioning with compute intent—and vice versa—UONACO aims to improve the efficiency of wide-area collaborative AI computing.

### 1.1. 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.

## 2. Service Model for AI Computing over Optical Network

The deployment of wide-area AI services over optical infrastructure involves multiple stakeholders, each playing a distinct role in the end-to-end service delivery chain. To clarify responsibilities and interactions, this document defines a service model comprising the Customer, Service Provider, Network Provider, and Computing Power Provider.

### 2.1. Customer

The Customer is the end user or enterprise that consumes AI capabilities. Three primary service patterns are observed:

In AI training, the customer delegates the training of large-scale AI models to service providers, typically specifying performance, scale, and data privacy requirements.

In AI inference, the customer leases computing resources to deploy and operate inference models, often serving downstream internet users with real-time or batch inference services.

In accessing remote AI inference service, the customer invokes pre-deployed inference APIs offered by third parties, expecting deterministic latency, reliability, and quality of service without managing underlying infrastructure.

### 2.2. Service Provider

The Service Provider acts as the business orchestrator, interfacing directly with the Customer to translate high-level service intents—such as SLAs, geographic constraints, or performance targets—into concrete resource demands. It coordinates with both the Network Provider and the Computing Power Provider to fulfill these demands, and is responsible for service lifecycle management, billing, and customer support.

### 2.3. Network Provider

The Network Provider operates and manages the underlying optical transport infrastructure. It delivers high-bandwidth, low-latency, and deterministic connectivity services, including inter-AIDC backbone links and user-to-AIDC dedicated access circuits. The Network Provider exposes network capabilities—such as available bandwidth, path latency, and reliability—through standardized control interfaces to enable coordinated service provisioning.

### 2.4. Computing Power Provider

The Computing Power Provider owns and operates one or more Artificial Intelligence Data Centers (AIDCs). It offers compute, memory, and accelerator resources (e.g., GPUs, TPUs) for AI training and inference workloads. The Computing Power Provider reports real-time resource availability and performance metrics to the Service Provider and supports dynamic task placement and scaling based on orchestration instructions.

### 3. UONACO Control and Management Architecture

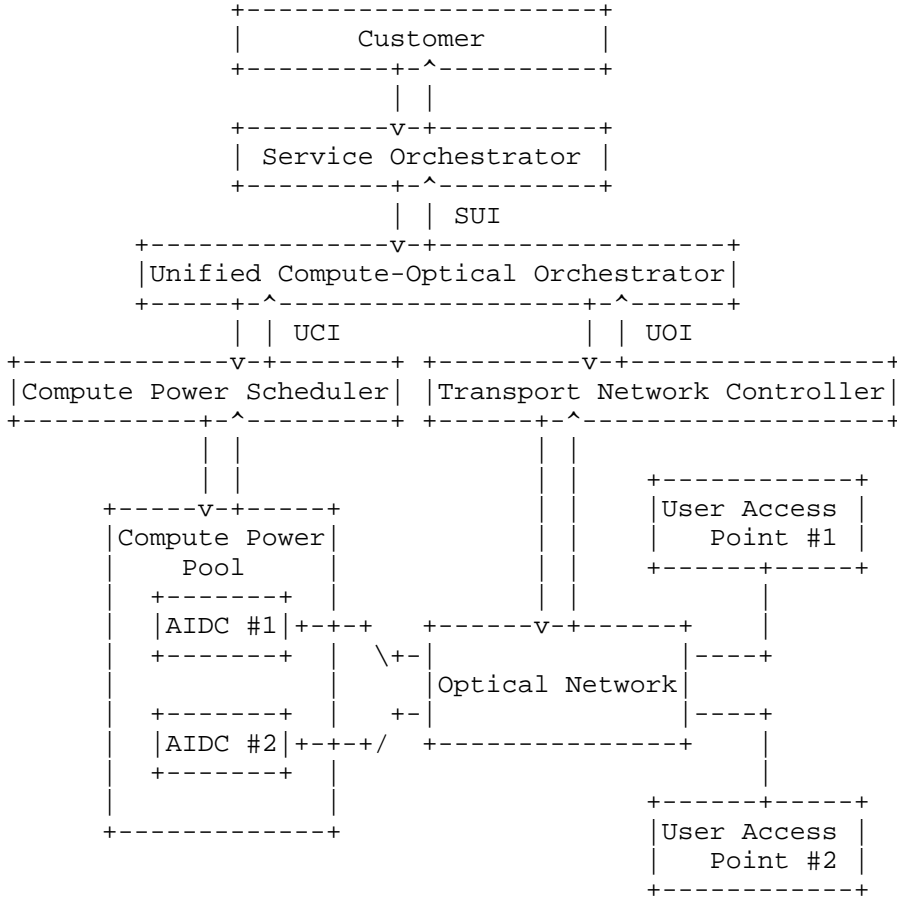


Figure 1: UONACO Control and Management Architecture

#### 3.1. Overview

As shown in Figure 1, the UONACO framework establishes a layered control architecture that enables end-to-end coordination between service intent, compute resources, and optical transport infrastructure. This architecture comprises five core functional components—Customer, Service Orchestrator (SO), Unified Compute-Optical Orchestrator (UCOO), Transport Network Controller (TNC), and Computing Power Scheduler (CPS)—interconnected through three standardized interfaces.

### 3.2. Service Orchestrator

The SO serves as the business-facing interface of the UONACO framework. It is responsible for accepting AI service requests from customers—such as “deploy a distributed training job across multiple AIDCs with end-to-end latency under X ms” or “provision an inference service with Y GPU instances and guaranteed bandwidth”—and translating these intent-based specifications into structured resource requirements. The SO also handles service lifecycle management, including billing, SLA enforcement, and user authentication. It does not manage physical resources directly but instead communicates abstracted demands to the UCOO via the SUI interface.

### 3.3. Unified Compute-Optical Orchestrator

The UCOO is the central coordination engine of the UONACO architecture. It receives service intents from the SO and continuously collects real-time telemetry from both the optical network (via TNC) and compute infrastructure (via CPS). Based on this global view, the UCOO executes joint optimization algorithms that consider both compute capabilities (e.g., GPU availability, memory) and network conditions (e.g., path latency, available bandwidth, congestion). The output of this decision process is a pair of synchronized instructions: one for optical path provisioning and another for compute task placement. The UCOO thus bridges the semantic and operational gap between the service layer and the infrastructure layer.

### 3.4. Transport Network Controller

The TNC represents the control plane of the underlying optical transport infrastructure. It may encompass a hierarchy of controllers, including intra-domain optical controllers and inter-domain coordinators (e.g., multi-domain WSON or OXC orchestrators). The TNC is responsible for managing physical and virtual optical resources—such as wavelengths, time slots, fgOTN/OSU slices, and OXC cross-connects—and for executing path computation, signaling, and protection mechanisms. In the UONACO framework, the TNC exposes network topology, available capacity, and performance metrics to the UCOO through the UOI interface, and applies provisioning commands issued by the UCOO to establish, adjust, or release optical connections in response to compute workload dynamics.

### 3.5. Computing Power Scheduler

The CPS acts as the controller for the AI compute pool, typically spanning one or more Artificial Intelligence Data Centers (AIDCs). It manages heterogeneous compute resources—including CPUs, GPUs, TPUs, memory, and storage—and reports their real-time availability, utilization, and performance characteristics (e.g., FLOPS, VRAM usage) to the UCOO. Upon receiving placement instructions from the UCOO via the UCI interface, the CPS schedules AI workloads (e.g., training jobs or inference containers) onto appropriate nodes, configures runtime environments, and ensures that compute tasks are aligned with the concurrently provisioned optical connectivity.

### 3.6. UONACO Interfaces

The UONACO framework defines three key interfaces which have been shown in Figure 1, to enable interoperability and decoupled evolution of its components.

**SUI (SO-UCOO Interface):** SUI connects SO and UCOO. Through this northbound interface, the SO conveys high-level service intent, including abstracted SLA requirements (e.g., maximum end-to-end latency, minimum bandwidth, geographic constraints), service type (e.g., AI training, inference, or remote access), and lifecycle events (e.g., service activation, modification, or termination). The UCOO interprets these intents as concrete resource demands and initiates joint optimization. The SUI thus serves as the bridge between business-oriented service definitions and infrastructure-aware orchestration.

**UOI (UCOO-TNC Interface):** UOI links UCOO with TNC. This interface enables bidirectional communication: the UCOO sends optical resource requests specifying required connectivity attributes such as bandwidth, end-to-end latency bounds, path isolation level, and resilience requirements; in return, the TNC provides real-time network state updates, including topology, available wavelengths or time slots, link utilization, propagation delay, and fault status. By exposing network capabilities and constraints to the orchestration layer, the UOI allows the UCOO to make network-feasible decisions and enables the TNC to provision optical paths that are aligned with compute workload dynamics.

**UCI (UCOO-CPS Interface):** UCI connects UCOO and CPS. Through this interface, the UCOO issues compute resource demands and task placement directives—such as the number and type of accelerators required, memory footprint, and preferred deployment topology—based on the outcome of joint compute-optical optimization. Conversely, the CPS reports real-time compute resource availability, node load,

energy efficiency metrics, and task execution status (e.g., job progress, failure alerts). This feedback loop ensures that compute allocation respects both application requirements and the quality of the concurrently provisioned optical connectivity, thereby avoiding placements that would violate network SLAs.

These interfaces are designed to be protocol-agnostic but are expected to leverage standardized, model-driven approaches (e.g., YANG/NETCONF or RESTCONF) to ensure vendor neutrality and scalability.

#### 4. IANA Considerations

TBD

#### 5. Security Considerations

TBD

#### 6. References

##### 6.1. 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>>.

#### Authors' Addresses

Qiaojun Hu (editor)  
Beijing University of Posts and Telecommunications  
Email: [qiaoj475@bupt.edu.cn](mailto:qiaoj475@bupt.edu.cn)

Zheng Han  
Beijing University of Posts and Telecommunications  
Email: [2025010255@bupt.cn](mailto:2025010255@bupt.cn)

Wei Wang  
Beijing University of Posts and Telecommunications  
Email: [weiw@bupt.edu.cn](mailto:weiw@bupt.edu.cn)

Jie Zhang  
Beijing University of Posts and Telecommunications  
Email: jie.zhang@bupt.edu.cn

Yongli Zhao  
Beijing University of Posts and Telecommunications  
Email: yonglizhao@bupt.edu.cn

Yanxia Tan (editor)  
China Unicom  
Email: tanyx11@chinaunicom.cn

Yanlei Zheng  
China Unicom  
Email: zhengyanlei@chinaunicom.cn