Optical Networks and AI Computing Orchestration (ONCO)

CCAMP WG, IETF 126

draft-tan-ccamp-onco-problem-statement-00
draft-tan-ccamp-onco-control-framework-00

Authors:

Yanxia Tan (China Unicom)

Zheng Han (BUPT)

Xiao Li (Huawei)

Sanwei Yang (ZTE)

Xing Zhao (CAICT)

Main changes after IETF 125

Name	UONACO	→	**ONCO**	A scope tied directly to optical networks
Use cases	3 scenarios	→	**Distributed Training + Inference**	Focus on compute-to-compute workloads
Architecture	New architecture	→	**ACTN-based**	Reuse and extension
Workflow	Not explicit	→	**Centralized + distributed lifecycle workflows**	Makes provisioning, adjustment, and release reviewable

Identified gap: Cross-domain lifecycle coordination

Compute scheduler
GPU availability
Job placement

No shared view of readiness, reservation, or release

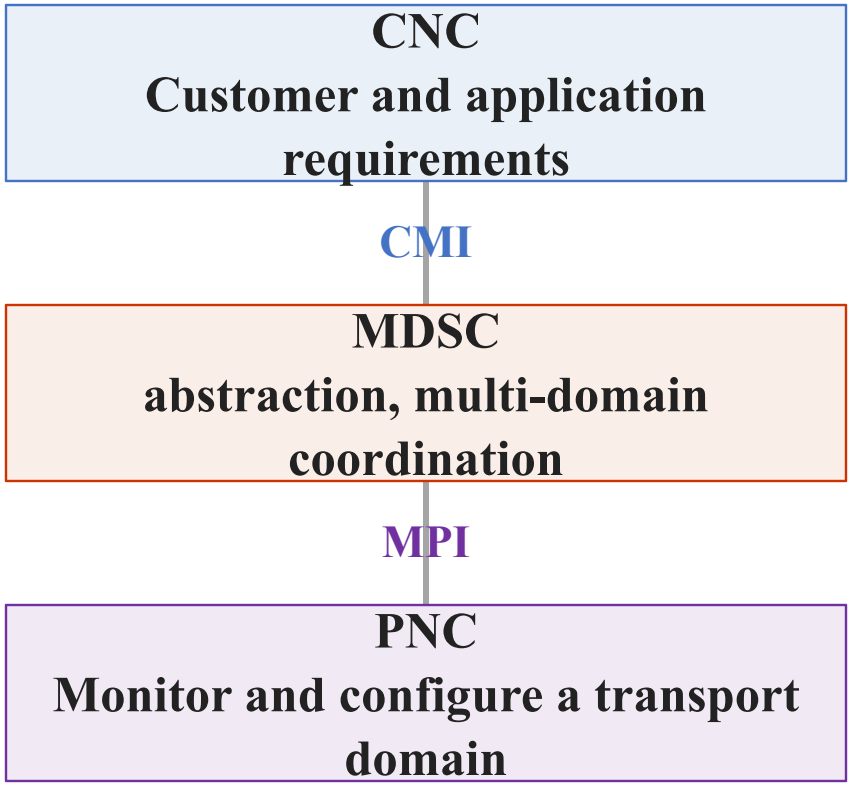
Optical controller
Topology and capacity
Path provisioning

ONCO coordinates the full lifecycle before traffic starts

Select → Reserve + Provision → Execute → Release

CATS selects among available service instances and steers traffic;
ONCO reserves compute and provisions deterministic optical connectivity.

ACTN provides the transport-control foundation



Reuse the ACTN foundation

1. Service expression (CNC / CMI)

2. Abstraction and multi-domain coordination (MDSC)

3. Transport monitoring and provisioning (PNC / MPI)

ONCO retains the ACTN control structure and reuses the optical MPI and PNC as-is.

What ONCO adds beyond ACTN

1

Abstract compute state

Expose feasible capacity and constraints—not internal topology.

2

Reserve and lock compute resources

Turn awareness into committed capacity for long-running AI jobs.

3

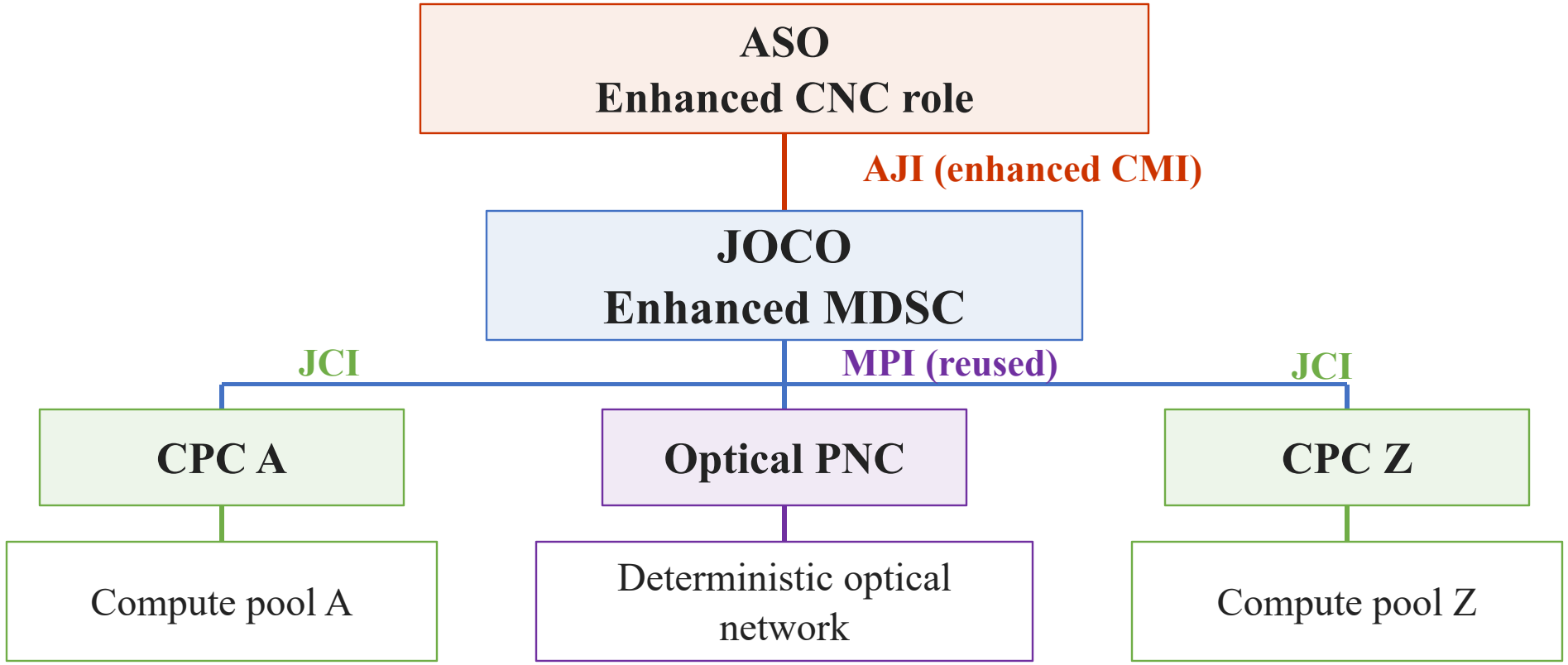
Make one joint decision

Select compute placement and deterministic connectivity together.

ONCO mapping to ACTN and minimum additions

ACTN function	ONCO function	Treatment	
CNC	**ASO**	ENHANCED	Compute + connectivity intent
CMI	**AJI**	ENHANCED	Combined intent
MDSC	**JOCO**	ENHANCED	Compute–optical coordination
MPI + optical PNC	**MPI + optical PNC**	REUSED	Optical control unchanged
—	**JCI + CPC**	NEW	State, reserve, lock, release

Centralized ONCO Architecture



JOCO coordinates abstract views; domains stay authoritative.

Centralized workflow: synchronized provisioning

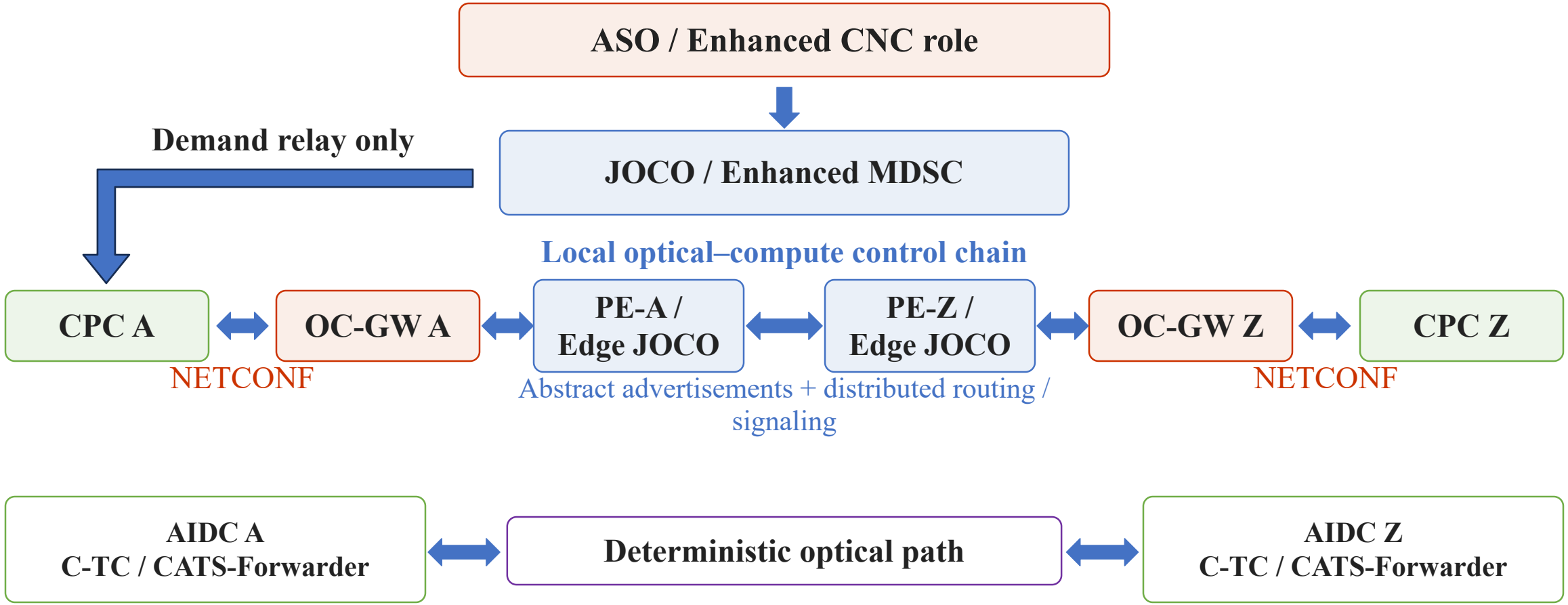
1	2	3	4	5	6
AI intent	**Abstract state**	**Joint selection**	**Synchronized setup**	**Execute**	**Release**
ASO → JOCO	JCI + MPI	Compute + path	Lock + provision	Start when ready	Return resources

JCI: compute state, reservation, locking, and release

MPI: optical state, path setup, modification, and teardown

Start only after compute and optical resources are ready.

Distributed ONCO Architecture



Edge JOCO selects the remote endpoint using advertised compute and optical state.

Distributed workflow: edge selection and delivery

1	2	3	4	5
Advertise state	**Relay demand**	**Select and connect**	**Start service**	**Adjust / recover / release**
CPC → OC-GW → Edge JOCO	JOCO → CPC Local check	Select endpoint then establish path	After remote readiness ACK	Reselect, restore, or teardown

Abstract compute + optical state

Endpoint selection + deterministic path

Start service only after remote readiness is confirmed.

The same chain handles adjustment, recovery, and release.

Feedback requested from the CCAMP WG

1. **Is the ACTN-to-ONCO mapping clear and technically accurate?**

2. **Is the CPC/JCI branch the minimum necessary compute-domain addition?**

3. **Should centralized and distributed deployments remain in one framework draft?**

Comments on both drafts are welcome on the CCAMP mailing list.