北京航空航天大學 BEIHANG UNIVERSITY

HiCom: A Hyper-ICN Architecture for Computing Power Network in Edge

Xiang Li, Shan Zhang, Zhiyuan Wang, Yang Li,

Tianchun Gan, Guanhan Peng, Huchao Xiong, Hongbin Luo

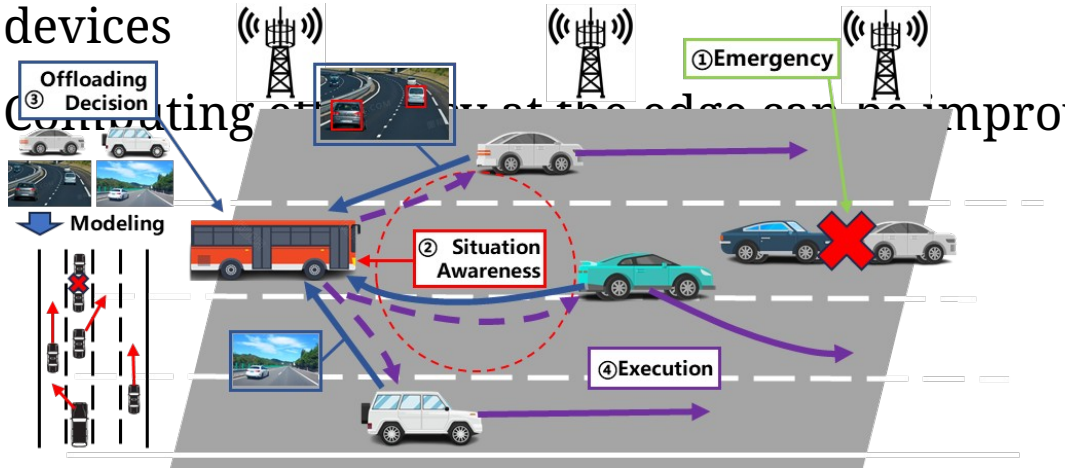
Beihang University

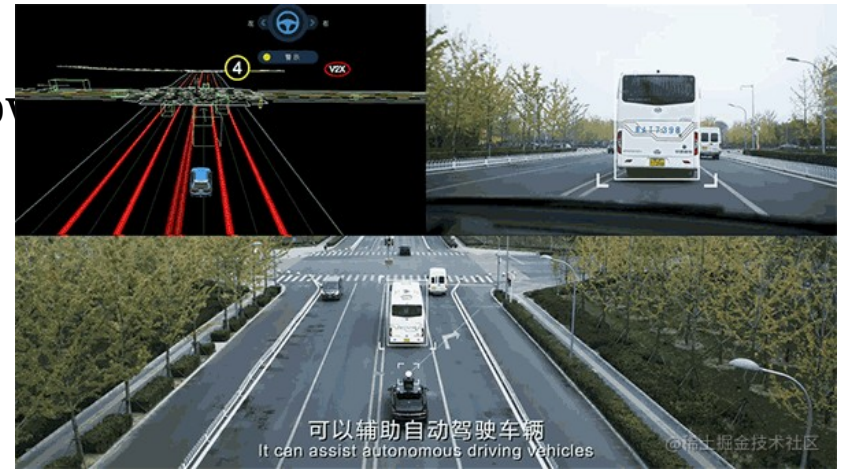
March, 2026

1. Background

Towards Computing Power Network (CPN) in Edge

- **80% of data will be generated and processed at the network edge**[Huawei]
  - Resource limited devices: camaras, sensors, IoT devices and machines
  - Resource rich devices: 2000 TOPS (NVIDIA Drive Thor) vs. 320 TOPS (L4 autonomous driving)
- **CPN unlocking the under-utilized computing resources**
  - Interconnecting and coordinating the ubiquitous computing power resources across diverse devices
  - Computing efficiency at the edge can be improv

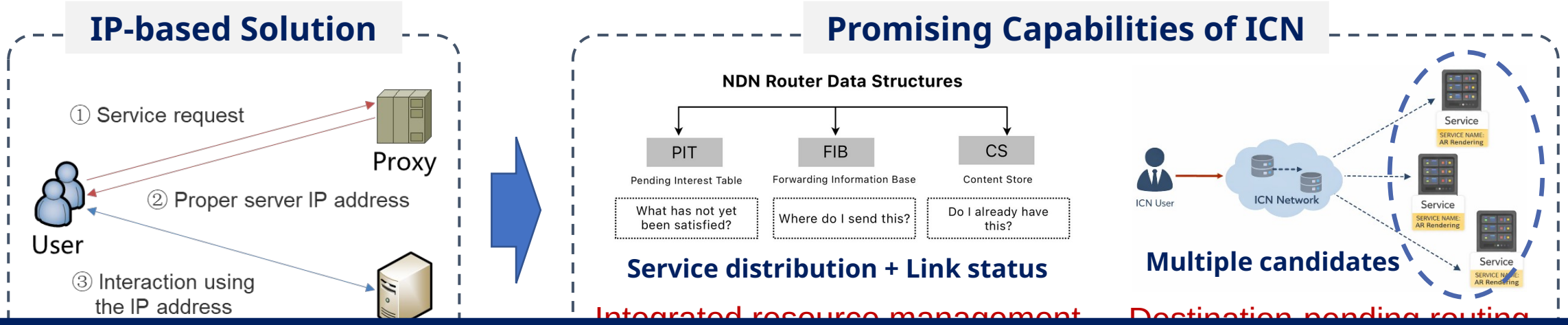




What kind of network architecture is required to support CPN?

From IP Networks to ICN

- **Conventional IP-based CPN solutions**
  - **IP-based solution: Separated networking and computing**
    - **Application-layer proxies**: manage computing resources and select servers for task execution via application layer
    - **Network routers**: build an end-to-end path between user and the selected server
    - **Resulting in weakly connected servers**: proxies lack real-time and comprehensive network awareness
  - **ICN-based solution: Coupled routing and addressing**
    - Promising to find a server with sufficient computing power and acceptable connection

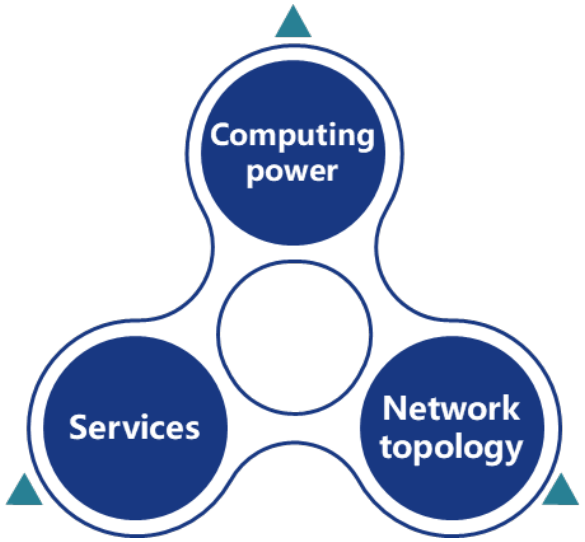


We explore a hyper-ICN architecture for edge CPN

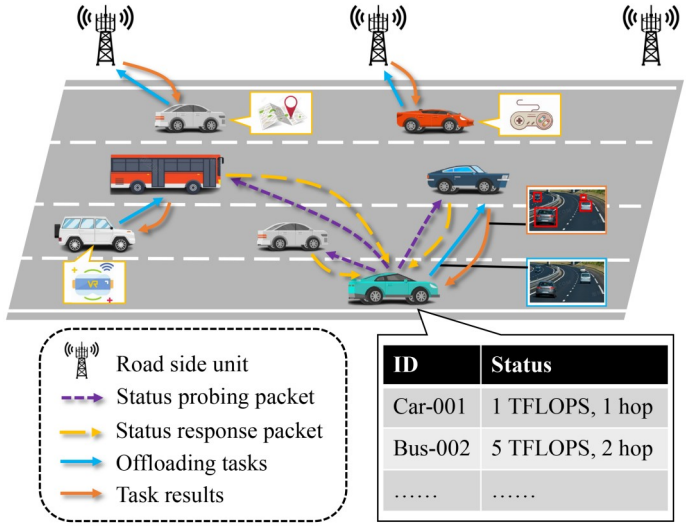
2. Three Fundamental Problems and Related Works

Three Fundamental Problems-1

- **1. Management of heterogeneous resources**
  - **Content/service/device is unique**: routing table should be global
  - **Computing power is reusable**: local routing table is enough
  - **How maintain the time-varying status of different resources at lower costs?**



Key resources for CPN tasks

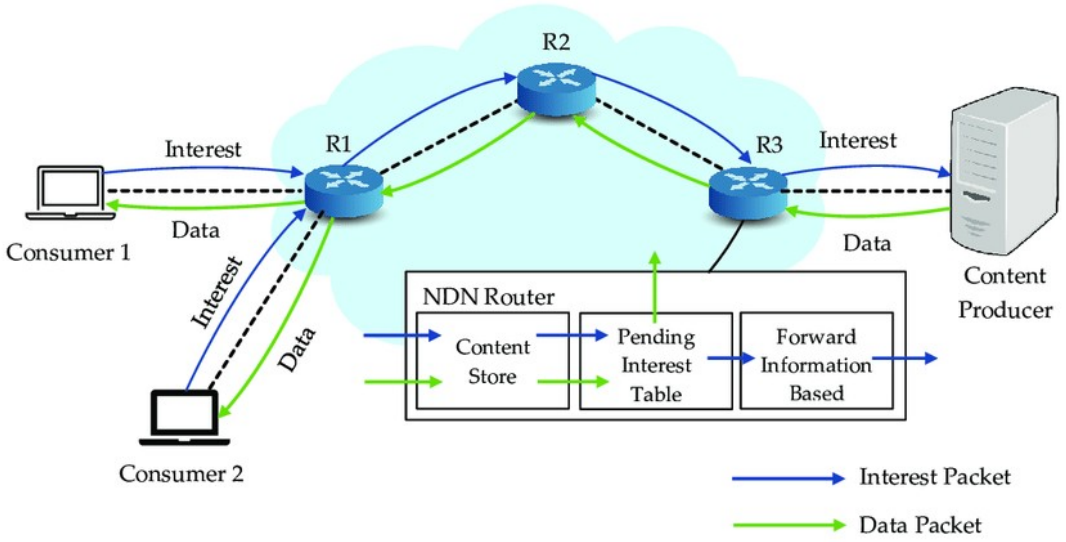


A typical use-case: Task offloading

This vehicle only cares about the computing status of nearby servers

Three Fundamental Problems-2

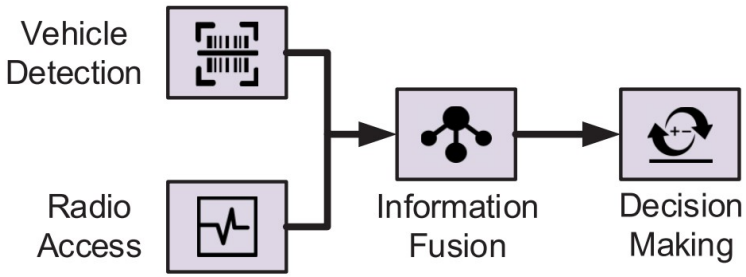
- **2. Supporting both server finding and results returning**
  - **Finding the server with ICN**: Interest packet with computing requirement, ACK to build the connection
  - **Result returning with ICN: invalid reverse path** after a long and uncertain time to complete the computing task at the server side
  - **How to re-bu[illegible]n dynamic environments?**



Stateful reverse path in ICN

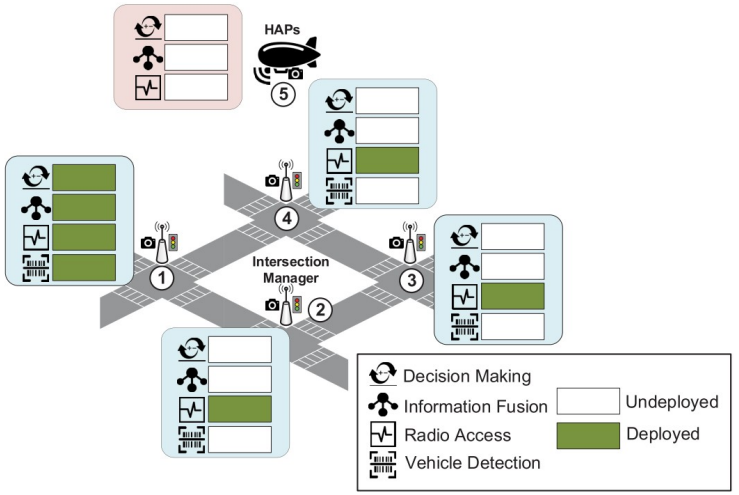
Three Fundamental Problems-3

- **3. Supporting dependent computing tasks**
  - Constrained computing capability of end/edge devices
  - Multiple servers to complete one task in a collaborative manner
  - Only single data/service retrieval supported by ICN
  - How to decompose the task and how to support the dependent sub-tasks?



A simple dependent computing task

Executed by multiple devices



3. HiCom Architecture

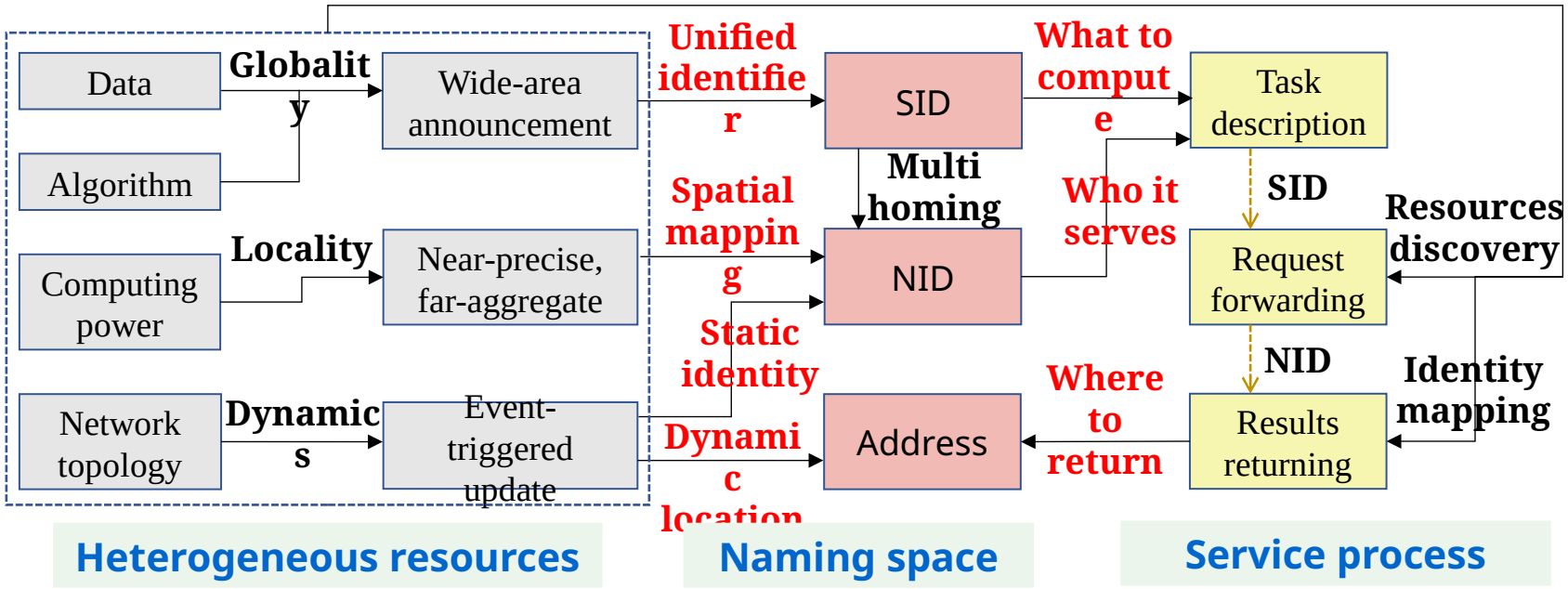
Design 1: Integrated yet Differentiated Control Plane

- **A multi-dimensional naming space (for heterogeneous resources mapping)**
  - **Service identifier (SID):** Identifies all data and services
  - **Node identifier (NID):** identifies networked nodes, with computing power attached
  - **Addresses**: IPv4/v6
  - **Cons:** (1) a global view of service and node distribution; (2) truncated status space of computing power based on SID and IP; (3) support destination-determined routing for result returning

Objects:
- Network topology
- Service distribution
- Computing resources

Mapping

Naming space:
How to identify different resources?



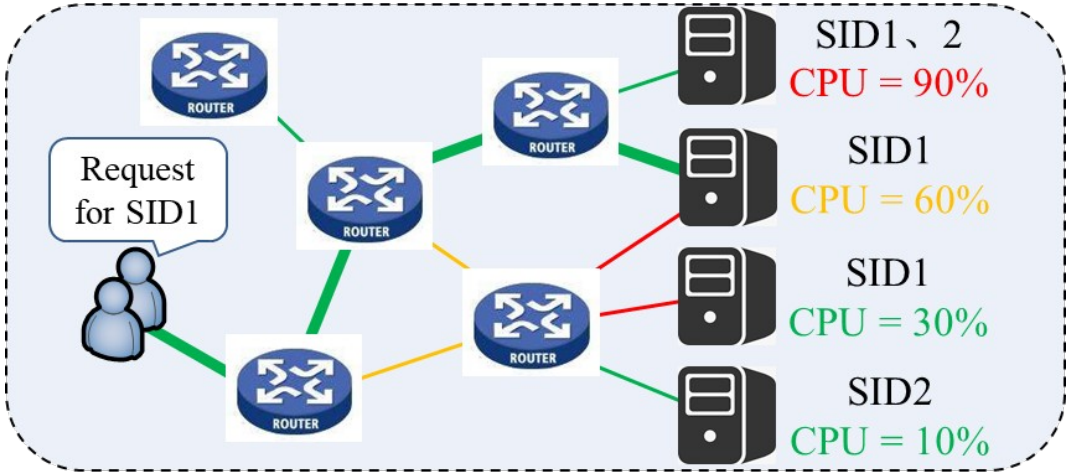
Design 2: ICN-IP Integrated Service Mechanism

Request forwarding

ICN's destination-pending routing for joint server and path optimization

SID

Pending server



Service Registration Table (SRT)

SID	Provider NID

Computing Power Table (CPT)

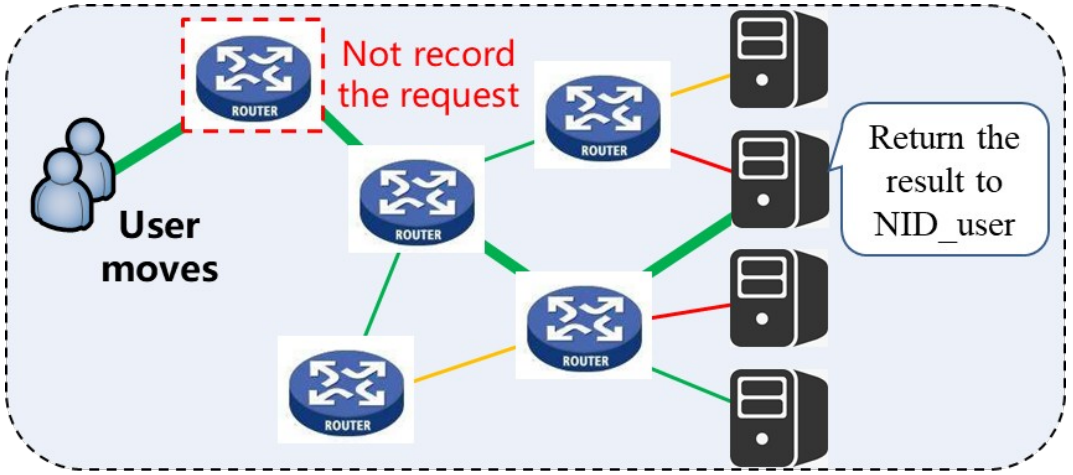
Server NID	Computing status

Result returning

IP's destination-determined routing to ensure timely and reliable transmission

NID

The requesting user



Address Mapping Table (AMT)

NID	IP address

IP Routing Table

Flow tables used for routing

Design 3: DAG-based Dependent Task Scheduling

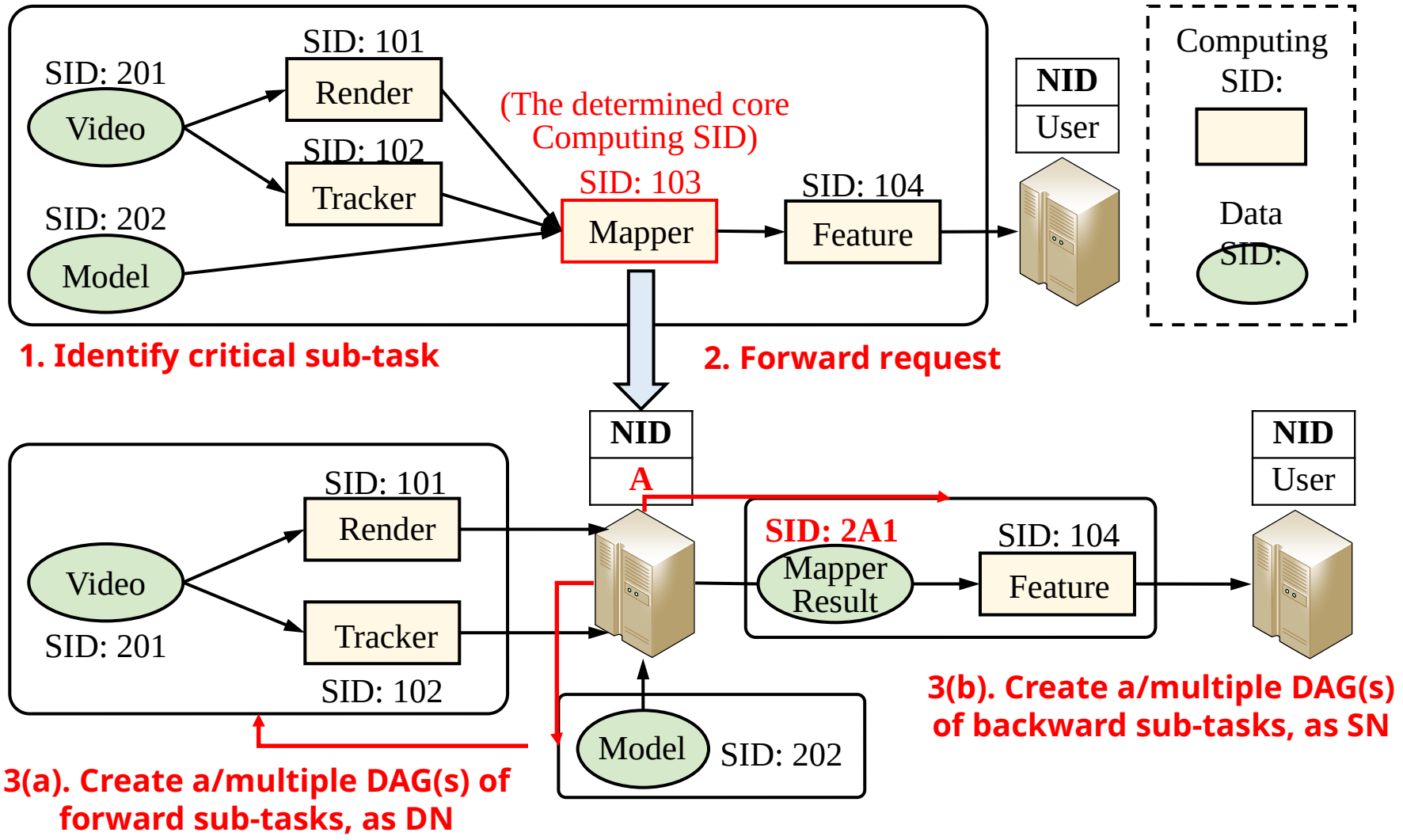
- **Recursive procedures to schedule sub-tasks through DAG disassembly**

DAG-based dependent task description

ICN-based request forwarding

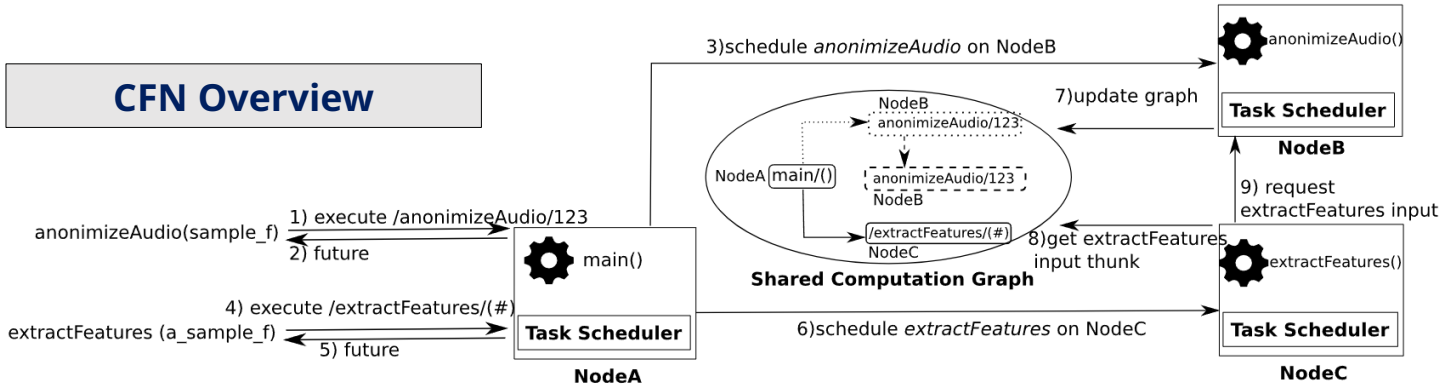
Local NID to acquire preceding inputs

New data SID to indicate processed result



Compare to Computing First Networking (CFN)[Krol19]

- **Similarities in task scheduling**
  - **ICN name-based routing:** enables network-assisted service discovery and execution
  - **Computation graph-based scheduling:** exposes task dependencies for flexible distributed scheduling

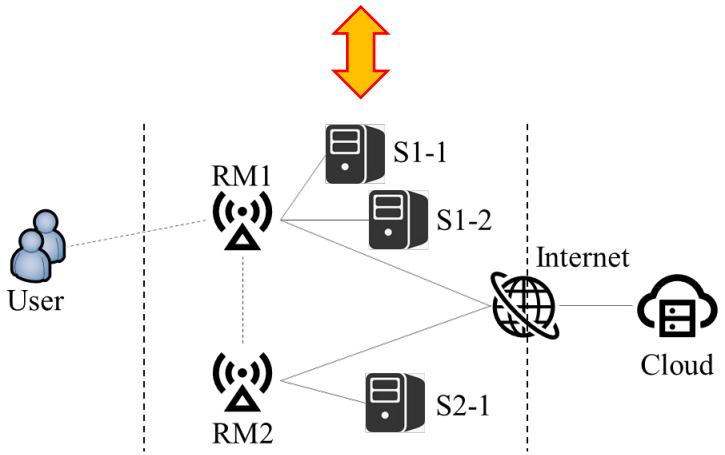


- **Differences**
  - **Computing power awareness**
    - Application-layer management (task schedulers) → Network-layer maintenance via multi-dimensional naming
  - **Result returning:** RICE-enabled result retrieval → IP-based destination-determined routing
  - **Status synchronization of task processing:** Graph-base task state synchronization among all involved servers → sub-task status synchronization with predecessor and successor nodes

[Krol19] Król M, Mastorakis S, Oran D, et al. Compute first networking: Distributed computing meets ICN[C]//Proceedings of the 6th ACM Conference on Information-Centric Networking. 2019: 67-77.

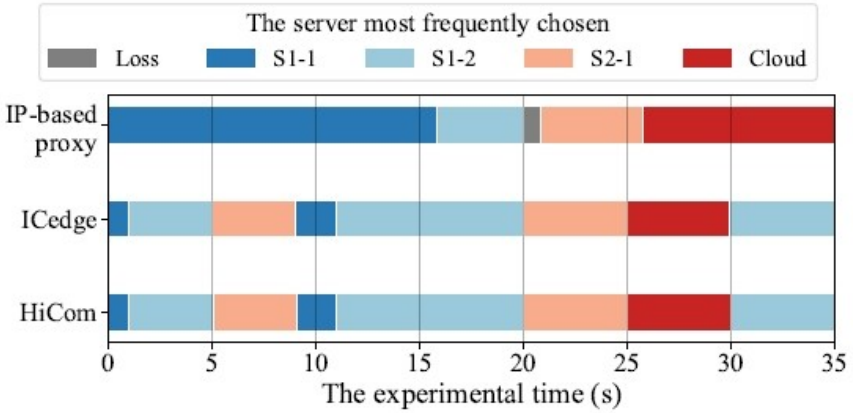
HiCom Prototype

- **Cloud-edge collaborative scenario**
  - **Functional hosts**: developed by DPDK
  - **Hierarchical computing power configuration**
    - End user: Intel Core i7-6700 CPU
    - Edge servers: Nvida GTX 1660s GPU
    - Simulated cloud server: Nvida RTX 4070Ti GPU
  - **Network configuration**
    - Sampling based on real-trace data from the Internet
    - Inter-domain latency & link bandwidth
  - **Service deployment**
    - Each server pre-deployed with the image detection service (YOLOv5)

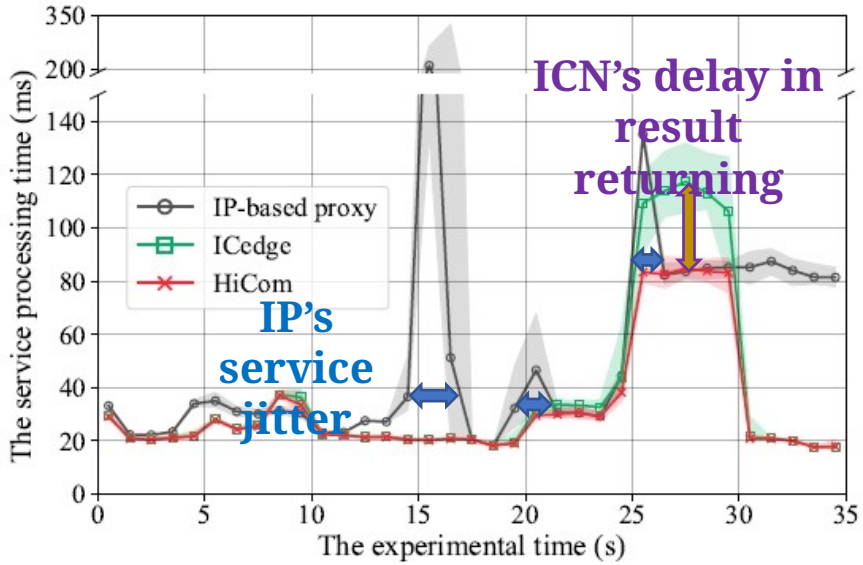


HiCom Prototype

- **Experimental results**
  - User requests the service at 25 FPS
  - Dynamic test environment
    - 5s: User switches edge access to **RM2**
    - 10s: User switches back to **RM1**
    - 15s: Load increases on local edge server **S1-1**
    - 20s: Local edge server **S1-2** disconnects
    - 25s: Bandwidth drops on neighboring server **S2-1**
    - 30s: Edge server **S1-2** reconnects
  - Results Analysis
    - Compared with **IP-based proxy** scheme:
      - Supports real-time, context-aware task scheduling
    - Compared with **ICN-based** scheme (**ICedge**):
      - Reduces task completion time by at least 8.6%



Server selection trends



Variation in service times

Summary

- **HiCom: A Hyper-ICN Architecture for Computing Power Network in Edge**
- **Three fundamental problems**
  - Management of heterogeneous resources
  - Supporting both server finding and results returning
  - Supporting dependent computing tasks
- **HiCom design**
  - An integrated yet differentiated control plane
  - An ICN-IP integrated service mechanism
  - A DAG-based dependent task scheduling mechanism
- **Open issues**
  - Computing Power Modeling and Management
  - Fault Tolerance and Resilience in Dynamic Environments
  - NL-described task support with AI agents

北京航空航天大學 BEIHANG UNIVERSITY

Thank You!

zhangshan18@buaa.edu.cn

X. Li *et al*., "HiCom: A Hyper-ICN Architecture for Computing Power Network in Edge," in *IEEE Network*, vol. 40, no. 1, pp. 173-182, Jan. 2026. https://ieeexplore.ieee.org/document/10933996