SANNet: A Semantic-Aware Agentic AI Networking Framework for Multi-Agent Cross-Layer Coordination

Yong Xiao*†‡, Haoran Zhou*, **Xubo Li***, Yayu Gao*, Guangming Shi†§‡, Ping Zhang¶
* School of Elect. Inform. & Commun., Huazhong Univ. of Science & Technology, China
† Peng Cheng Laboratory, Shenzhen, China
‡ Pazhou Laboratory (Huangpu), Guangzhou, China
§ School of Artificial Intelligence, Xidian University, Xi'an, China
¶ State Key Lab. of Networking & Switching Tech., Beijing Univ. of Posts & Telecom., Beijing, China

IEEE GLOBECOM®

Outline

- ■Background and Motivation
- ■System Model
- ■SANNet Architecture
- ■Prototype and Experimental Results
- ■Conclusion

Background and Motivation

Evolution toward Agentic AI Networking

The Trend:

- 6G will be dominated by diverse AI agents coexisting and interacting.
- Shift from Data-Transportation-Centric to Agentic AI Networking (AgentNet).

The Challenge:

- **Complex Coordination:** Existing architectures lack frameworks for automatic goal discovery and multi-agent self-orchestration.
- **Conflicting Objectives:** Agents at different layers (Application, Network, Physical) often have conflicting optimization goals.
- **Generalization:** Local data is limited; agents struggle with unseen environments.

Background and Motivation

Challenges in Current Agentic Architectures

Limitations of Current Research:

- Most existing studies focus heavily on multi-agent task planning and adaptation logic, often ignoring the fundamental role of the communication networking architecture.

The Semantic Gap:

- There is a critical lack of comprehensive frameworks capable of:
  - Automatically detecting a user's high-level semantic goal
  - Self-orchestrating diverse agents to fulfill these detected goals

The Cross-Layer Necessity:

- Effective AgentNet systems require seamless interaction not just between agents, but across distinct system layers:
  - Application Layer (User intent/Semantics)
  - Network Layer (Routing/Resources)
  - Physical Layer (Spectrum/Channel State)

Key Problem

How to coordinate specialized agents across different layers to fulfill high-level semantic user goals while resolving objective conflicts?

Outline

- ■Background and Motivation
- ■System Model
- ■SANNet Architecture
- ■Prototype and Experimental Results
- ■Conclusion

System Model

Cross-Layer Multi-Agent System Model Architecture

We consider a mobile networking system comprising three distinct agent types:

1. **Application-layer Agent (aAgent):**
   - Interacts with users/environment
   - define $\mathcal{G}^a = \langle \mathcal{A}^a, \mathcal{S}^a, \mathcal{L}^a, \mathcal{D}^a \rangle$
2. **Physical-layer Agent (pAgent):**
   - Estimates spectrum, channel state (CSI)
   - define $\mathcal{G}^p = \langle \mathcal{A}^p, \mathcal{S}^p, \mathcal{L}^p, \mathcal{D}^p \rangle$
3. **Network-layer Agent (nAgent):**
   - Manages routing and bandwidth resources
   - define $\mathcal{G}^n = \langle \mathcal{A}^n, \mathcal{S}^n, \mathcal{L}^n, \mathcal{D}^n \rangle$

$\mathcal{A}$ is the set of acts
$\mathcal{L}$ is the set of environmental state
$\mathcal{S}$ is the set of task-related loss functions
$\mathcal{D}$ is the set of data samples for training

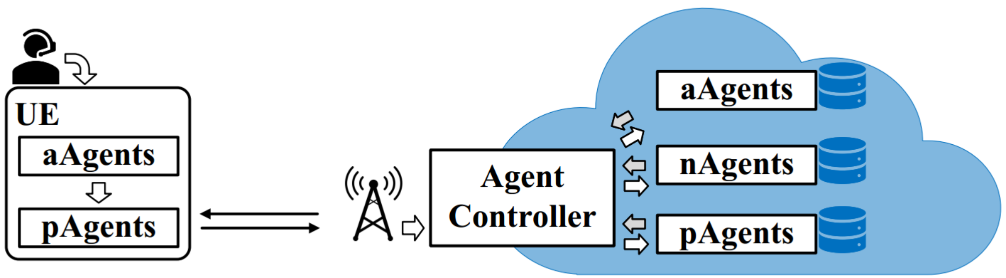


Figure: System model of cross-layer mobile networking system

System Model

Problem Formulation

model parameter

Optimization Goal: Minimize the vector of loss functions from all participating agents.

$$\min_{\Omega} L_m(\Omega) := \left(l_m^a(\Omega_m, \mathcal{D}^a), l_m^p(\Omega_m, \mathcal{D}^p), l_m^n(\Omega_m, \mathcal{D}^n)\right)$$

Key Challenges:

the optimal weight

- **Multi-Agent Conflict Error (C-error):** Agents have divergent gradients.

$$\mathcal{E}_C = \left\| \sum_{i \in \{a,p,n\}} (\gamma^i - \gamma^{i*}) \nabla l_m^i(\Omega_m, \mathcal{D}^i) \right\|$$

- **Generalization Error (G-error):** Discrepancy between training performance and real-world deployment.

$$\mathcal{E}_G = \left\| \sum_{i \in \{a,p,n\}} \gamma^i \left(\nabla l_t^i(\Omega_t, \mathcal{D}^i) - \nabla \tilde{l}_t^i(\Omega_t) \right) \right\|$$

the loss gradient of the test dataset

Outline

- Background and Motivation
- System Model
- SANNet Architecture
- Prototype and Experimental Results
- Conclusion

SANNet Architecture

A New Functional Entity: Agent Controller

The Agent Controller manages the lifecycle of a semantic request:

- **Semantic Cognition:** Infers user intent (e.g., via LLMs).
- **Task Separation:** Translates intent into subtasks for aAgent, pAgent, nAgent.
- **Agent Selection:** Matches subtasks to agents based on "Agent Cards".
- **Conflict Resolving:** Dynamic weighting mechanism to handle divergent goals.

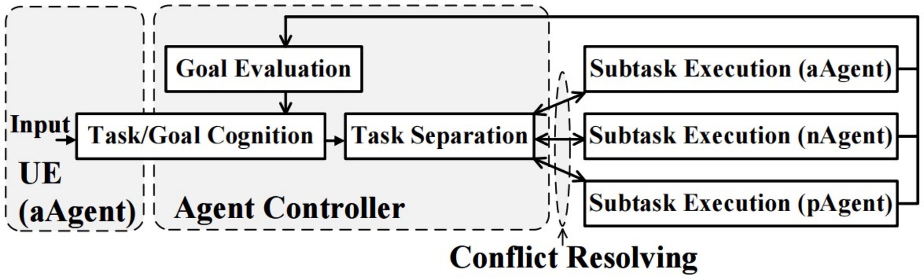


Figure: Coordination procedures in SANNet

SANNet Architecture

Conflict Resolution Mechanism

Dynamic Weighting Algorithm

Since a global optimum is impossible with conflicting losses, we seek a **Pareto stationary solution**. We dynamically update weights γ^i to minimize the **Conflict Error (C-error)**.

Update Rule (Algorithm 1)

For each iteration t:

step sizes of the weighting update

data samples

- Update weights based on gradient direction alignment:

$$\gamma_{t+1}^{i} = \gamma_{t}^{i} - \eta_t \nabla l^i(\Omega_t, \gamma_t^i, d_{t,1}^i)^{\top} \nabla l^i(\Omega_t, \gamma_t^i, d_{t,2}^i)$$

- Update model parameters:

$$\Omega_{t+1} = \Omega_t - \beta_t \nabla l^i(\Omega_t, \gamma_{t+1}^i, d_{t,3}^i)$$

step sizes of agents' model parameter update

SANNet Architecture

Theoretical Guarantees

We provide **theoretical bounds** for both Conflict and Generalization errors.

Bound on C-error

Suppose the following assumptions holds：

(i) $\nabla l^i(\Omega,\gamma)$ is l'_f - Lipschitz continuous for any data sample;

(ii) $l^i(\Omega,\gamma)$ is l_f - Lipschitz continuous for any data sample.

The following upper bound holds:

$$\mathcal{E}_C \leqslant \frac{4}{\eta T} + 6\sqrt{3 l'_f l_f^2 \frac{\beta}{\eta}} + 3\eta l_f^4$$

By setting specific step sizes, $\beta = \Theta(T^{-3/4}), \eta = \Theta(T^{-1/4})$, the C-error converges to the conflict-resolving direction at rate $\mathcal{O}(T^{-1/4})$:

SANNet Architecture

Theoretical Guarantees

We provide theoretical bounds for both Conflict and Generalization errors.

Bound on G-error

Suppose the Frobenius norm of the summation of gradients of all agents is upper bounded by constant, i.e, $\mathbb{E}\left[|| \sum \gamma^{i} \nabla l_m^i (\Omega_{\boldsymbol{m}}, \mathcal{D}^{i}) ||_F^2 \right] \leqslant U^2$ for U is a constant.

The generalization error is upper bounded by:

$$\mathcal{E}_G \leqslant \mathcal{O}\left(T^{\frac{1}{2}} D^{-\frac{1}{2}} \right)$$

the size of the training dataset

There is a trade-off, excessive iterations reduce C-error but may slightly increase G-error due to overfitting.

Outline

- **Background and Motivation**
- **System Model**
- **SANNet Architecture**
- **Prototype and Experimental Results**
- **Conclusion**

Prototype and Experimental Results

Prototype Implementation

We developed a hardware prototype based on **Open RAN** and **Open 5GS.**

Hardware Stack:

- **gNB:** NI USRP 2944R + Intel i9 Workstation + RTX 4090
- **UE:** NI USRP 2944R + Intel i7 Desktop.

Software Stack:

- srsRAN (Physical/MAC)
- Open5GS (Core Network)
- Agent Controller: OpenManus + Qwen-7B LLM.

Figure: SANNet Prototype: UE, 5G Core, and Agents

Prototype and Experimental Results

Experimental Results

Case Study (Immersive communication): User requests “Increase video resolution”

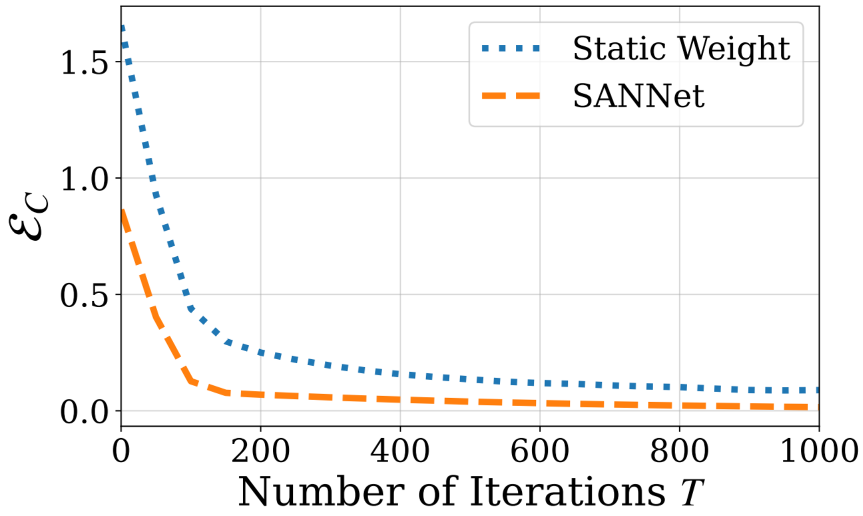


Figure: C-error Comparison

SANNet reduces C-error by up to **63%** compared to static weighting.

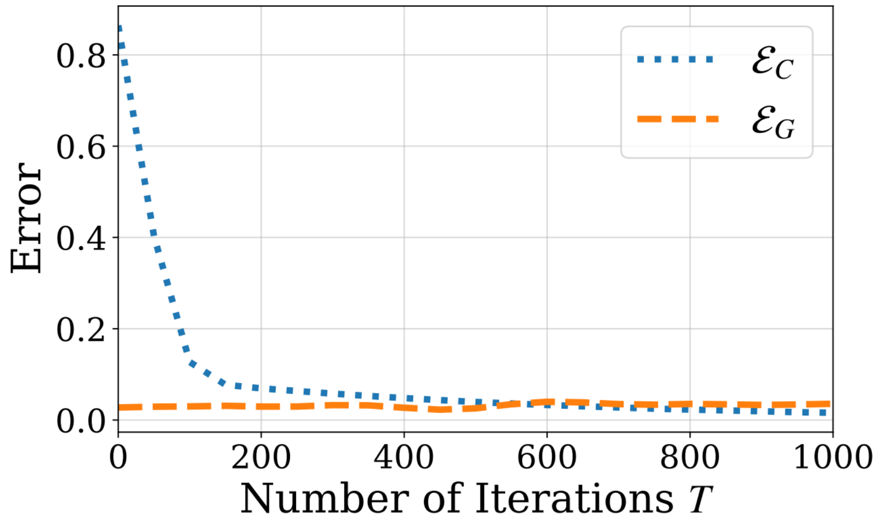
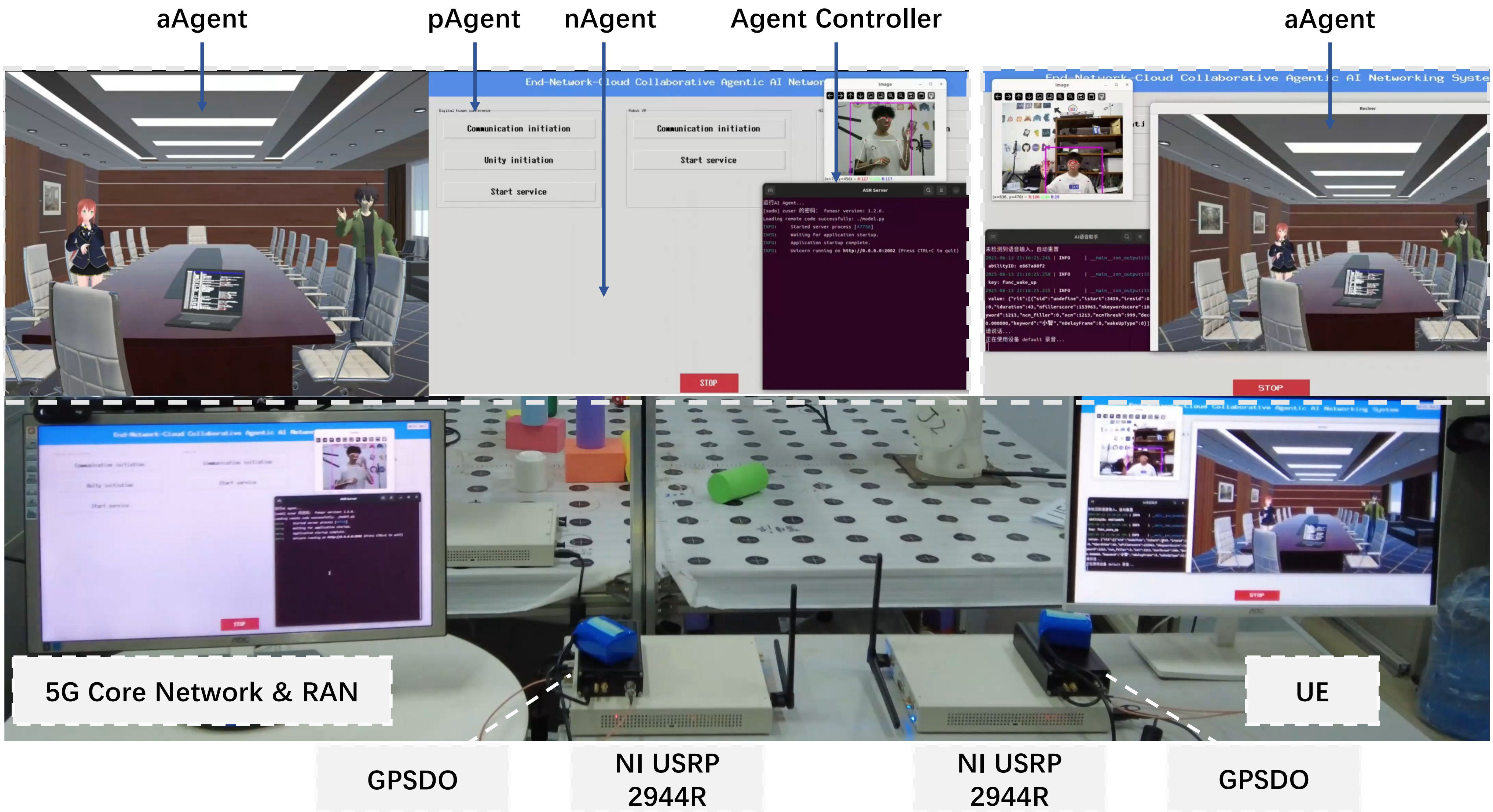


Figure: C-error vs. G-error

Shows convergence of Conflict Error while maintaining stable Generalization Error.


aAgent
pAgent
nAgent
Agent Controller
aAgent
End-Network-Cloud Collaborative Agentic AI Networking System
Communication initiation
Unity initiation
Start service
Communication initiation
Start service
STOP
5G Core Network & RAN
UE
GPSDO
NI USRP 2944R
NI USRP 2944R
GPSDO

Outline

- ■Background and Motivation
- ■System Model
- ■SANNet Architecture
- ■Prototype and Experimental Results
- ■Conclusion

Conclusion

SANNet Architecture: A novel semantic-aware framework enabling autonomous multi-agent coordination across layers.

Agent Controller: Facilitates semantic cognition, task separation, and agent selection.

Conflict Resolution: Proposed a dynamic weighting mechanism that theoretically guarantees convergence for conflicting objectives.

Real-World Viability: Validated on a hardware prototype (Open RAN + 5GS), demonstrating significant performance improvements over static baselines.

Future Work: Extending to massive multi-agent systems and more complex 6G scenarios.

Thank You!

For more information, please contact:

Xubo Li
(xuboli@hust.edu.cn)
Yong Xiao
(yongxiao@hust.edu.cn)