

Internet Research Task Force  
Internet-Draft  
Intended status: Informational  
Expires: 8 January 2026

S-B. Oh  
KSA  
Y-G. Hong  
Daejeon University  
J-S. Youn  
DONG-EUI University  
HJ. Lee  
S-W. Hong  
ETRI  
H-K. Kahng  
Korea University  
7 July 2025

AI-Based Distributed Processing Automation in Network Digital Twin  
draft-oh-nmrg-ai-adp-03

Abstract

This document discusses the use of AI technology and digital twin technology to automate the management of computer network resources distributed across different locations. Digital twin technology involves creating a virtual model of real-world physical objects or processes, which is utilized to analyze and optimize complex systems. In a network digital twin, AI-based network management by automating distributed processing involves utilizing deep learning algorithms to analyze network traffic, identify potential issues, and take proactive measures to prevent or mitigate those issues. Network administrators can efficiently manage and optimize their networks, thereby improving network performance and reliability. AI-based network management, utilizing network digital twin technology, also aids in optimizing network performance by identifying bottlenecks in the network and automatically adjusting network settings to enhance throughput and reduce latency. By implementing AI-based network management through automated distributed processing, organizations can improve network performance, and reduce the need for manual network management tasks.

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 8 January 2026.

## Copyright Notice

Copyright (c) 2025 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. Conventional Task Distributed Processing Techniques and Problems . . . . .	3
2.1. Challenges and Alternatives in Task Distributed Processing . . . . .	3
2.2. Considerations for Resource Allocation in Task Distributed Processing . . . . .	7
3. Requirements of Conventional Task Distributed Processing . . . . .	8
4. Automating Distributed Processing with Digital Twin and AI . . . . .	8
5. Technologies for AI-Based Distributed Processing Automation in Network Digital Twin . . . . .	9
5.1. Configuration of Network Digital Twin . . . . .	9
5.2. Data Collection and Processing . . . . .	10
5.3. AI Model Training and Deployment . . . . .	10
5.4. AI-based Distributed Processing . . . . .	10
5.5. AI-based network operation . . . . .	10
6. Security Considerations . . . . .	11
6.1. Data Validation and Bias Mitigation . . . . .	11
6.2. AI Model Vulnerability Detection . . . . .	11
7. IANA Considerations . . . . .	12
8. Acknowledgements . . . . .	12
9. Informative References . . . . .	12
Authors' Addresses . . . . .	12

## 1. Introduction

Due to industrial digitalization, the number of devices connected to the network is increasing rapidly. As the number of devices increases, the amount of data that needs to be processed in the network is increasing due to the interconnection between various devices.

Existing network management was managed manually by administrators/operators, but network management becomes complicated, and the possibility of network malfunction increases, which can cause serious damage.

Digital twin is a digital representation of an object of interest and may require different capabilities (e.g., synchronization, real-time support) according to the specific domain of application [Y.4600]. Digital twin systems help organizations improve important functional objectives including real-time control, off-line analytics, predictive maintenance by modelling and simulating of objects in the real world. Therefore, it is important for a digital twin system to represent as much real-world information about the object as possible when digitally representing the object.

Therefore, this document considers the configuration of systems using both digital twin technology and artificial intelligence (AI) technology for network management and operation, in order to adapt to the dynamically changing network environment. In this regard, AI technologies play a key role by maximizing the utilization of network resources. They achieve this by providing resource access control and optimal task distribution processing based on the characteristics of nodes that offer network functions for network management automation and operation[I-D.irtf-nmrg-ai-challenges].

## 2. Conventional Task Distributed Processing Techniques and Problems

### 2.1. Challenges and Alternatives in Task Distributed Processing

Conventional Task Distributed Processing Techniques refer to methods and approaches used to distribute computational tasks among multiple nodes in a network. These techniques are typically used in distributed computing environments to improve the efficiency and speed of processing large volumes of data.

Some common conventional techniques used in task distributed processing include load balancing, parallel processing, and pipelining. Load balancing involves distributing tasks across multiple nodes in a way that minimizes the overall workload of each node, while parallel processing involves dividing a single task into

multiple sub-tasks that can be processed simultaneously. Pipelining involves breaking a task into smaller stages, with each stage being processed by a different node.

However, conventional task distributed processing techniques also face several challenges and problems. One of the main challenges is ensuring that tasks are distributed evenly among nodes, so that no single node is overburdened while others remain idle. Another challenge is managing the communication between nodes, as this can often be a bottleneck that slows down overall processing speed. Additionally, fault tolerance and reliability can be problematic, as a single node failure can disrupt the entire processing workflow.

To address these challenges, new techniques such as edge computing, and distributed deep learning are being developed and used in modern distributed computing environments. The optimal resource must be allocated according to the characteristics of the node that provides the network function. Cloud servers generally have more powerful performance. However, to transfer data from the local machine to the cloud, it is necessary to move across multiple access networks, and it takes high latency and energy consumption because it processes and delivers a large number of packets. The MEC server is less powerful and less efficient than the cloud server, but it can be more efficient considering the overall delay and energy consumption because it is placed closer to the local machine[MEC.IEG006]. These architectures combine computing energy, telecommunications, storage, and energy resources flexibly, requiring service requests to be handled in consideration of various performance trade-offs.

The existing distributed processing technique can divide the case according to the subject performing the service request as follows.

- (1) All tasks are performed on the local machine.

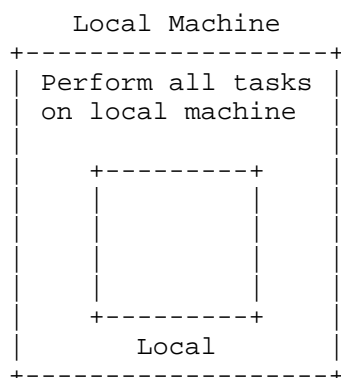


Figure 1: All tasks on local machine

(2) Some of the tasks are performed on the local machine and some are performed on the MEC server.

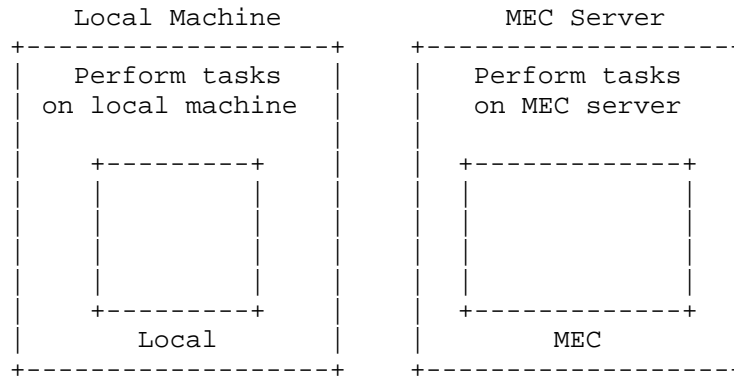


Figure 2: Some tasks on local machine and MEC server

(3) Some of the tasks are performed on local machine and some are performed on cloud server

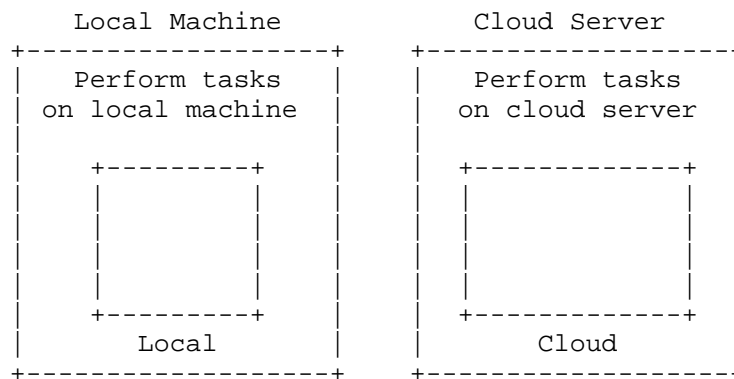


Figure 3: Some tasks on local machine and cloud server

(4) Some of the tasks are performed on local machine, some on MEC servers, some on cloud servers

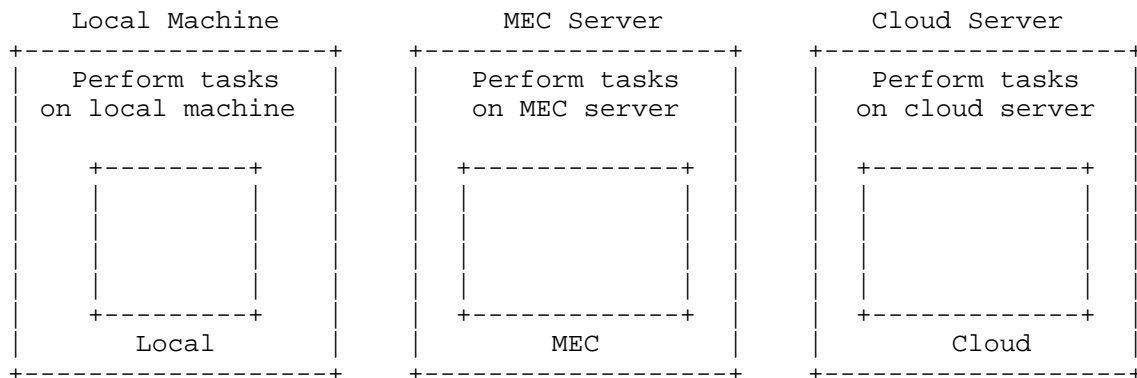


Figure 4: Some tasks on local machine, MEC server, and cloud server

(5) Some of the tasks are performed on the MEC server and some are performed on the cloud server

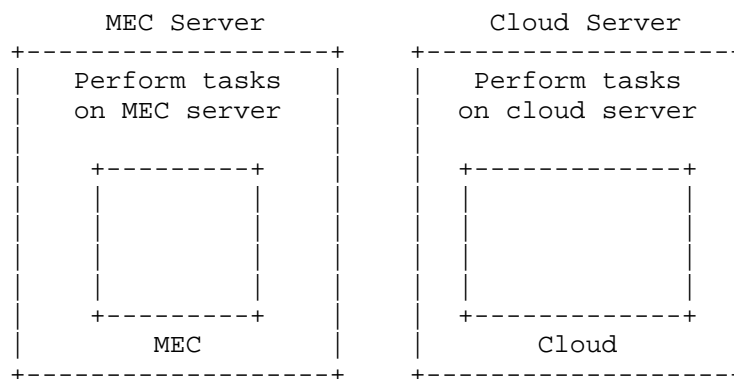


Figure 5: Some tasks on MEC server and cloud server

(6) All tasks are performed on the MEC server

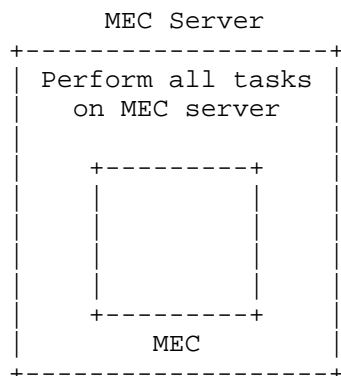


Figure 6: All tasks on MEC server

(7) All tasks are performed on cloud servers

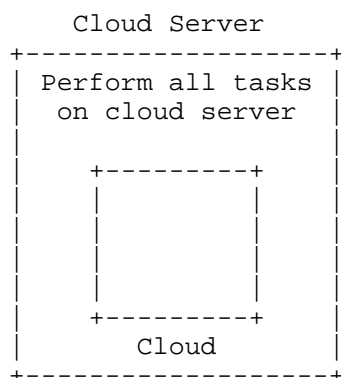


Figure 7: All tasks on cloud server

## 2.2. Considerations for Resource Allocation in Task Distributed Processing

In addition, it is necessary to consider various environments depending on the delay time and the importance of energy consumption to determine which source is appropriate to handle requests for resource use. The importance of delay time and energy consumption depends on the service requirements for resource use. There is a need to adjust the traffic flow according to service requirements.

### 3. Requirements of Conventional Task Distributed Processing

The requirements of task distributed processing refer to the key elements that must be considered and met to effectively distribute computing tasks across multiple nodes in a network. These requirements include:

- \* **Scalability:** The ability to add or remove nodes from the network and distribute tasks efficiently and effectively, without compromising performance or functionality.
- \* **Fault tolerance:** The ability to handle node failures and network outages without disrupting overall system performance or task completion.
- \* **Load balancing:** The ability to distribute tasks evenly across all nodes, ensuring that no single node becomes overwhelmed or underutilized.
- \* **Task coordination:** The ability to manage task dependencies and ensure that tasks are completed in the correct order and on time.
- \* **Resource management:** The ability to manage system resources such as memory, storage, and processing power effectively, to optimize task completion and minimize delays or errors.
- \* **Security:** The ability to ensure the integrity and confidentiality of data and tasks, and protect against unauthorized access or tampering.

Meeting these requirements is essential to the successful implementation and operation of task distributed processing systems. The effective distribution of tasks across multiple nodes in a network can improve overall system performance and efficiency, while also increasing fault tolerance and scalability.

### 4. Automating Distributed Processing with Digital Twin and AI

Automating distributed processing utilizing digital twin technology involves digitally modeling physical objects and processes from the real world, enabling real-time tracking and manipulation. This technology enables real-time monitoring and manipulation, revolutionizing how we understand and manage complex networks.

When combined with AI technology, these digital twins form a robust automated distributed processing system. For instance, digital twins can project all nodes and devices within a network digitally. The AI model can utilize various types of information, such as:



- \* Network data: Network-related data such as network traffic, packet loss, latency, bandwidth usage, etc., can be valuable for distributed processing automation. This data helps in understanding the current state and trends of the network, optimizing task distribution, and processing.
- \* Task and task characteristic data: Data that describes the characteristics and requirements of the tasks processed in the distributed processing system is also important. This can include the size, complexity, priority, dependencies, and other attributes of the tasks. Such data allows the AI technology to distribute tasks appropriately and allocate them to the optimal nodes.
- \* Performance and resource data: Data related to the performance and resource usage of the distributed processing system is crucial. For example, data representing the processing capabilities of nodes, memory usage, bandwidth, etc., can be utilized to efficiently distribute tasks and optimize task processing.
- \* Network configuration and device data: External environmental factors should also be considered. Data such as network topology, connectivity between nodes, energy consumption, temperature, etc., can be useful for optimizing task distribution and processing.

AI algorithms, based on this digital twin data, can automatically optimize network operations. For example, if overload is detected on a specific node, AI can redistribute tasks to other nodes, minimizing congestion. The real-time updates from digital twins enable continuous, optimal task distribution, allowing the network to adapt swiftly to changes.

By integrating digital twins and AI, the automated distributed processing system maximizes network performance while minimizing bottlenecks. This technology reduces the burden on network administrators, eliminating the need for manual adjustments and enhancing network flexibility and responsiveness.

## 5. Technologies for AI-Based Distributed Processing Automation in Network Digital Twin

### 5.1. Configuration of Network Digital Twin

In a network environment, digital twins are used to monitor the performance of the network infrastructure in real-time, optimize network traffic through AI-based distributed processing, predict issues, and automatically resolve them. To this end, it is important to select physical objects to be represented as digital twins in order to collect the various data described in Section 4.

## 5.2. Data Collection and Processing

Monitoring agents installed on network devices collect real-time data. This data includes traffic volume, latency, packet loss rates, CPU and memory usage, etc. Edge computing devices perform initial data processing before transmitting the data to the central management system.

## 5.3. AI Model Training and Deployment

The central system trains models for traffic prediction, fault prediction, and optimization based on the collected data. The trained models are deployed to network devices to perform real-time traffic analysis and optimization tasks.

## 5.4. AI-based Distributed Processing

Each network device or edge computing device analyzes data in real-time and dynamically adjusts traffic routes. The overall network status is monitored, and in case of a fault, traffic is automatically rerouted or devices are reset. Distributed edge devices communicate with each other to share network status and collaborate with the central system to optimize the entire network.

## 5.5. AI-based network operation

A Network Digital Twin (NDT) is a virtual replica of a real-world network that mirrors its behavior in real time. When combined with artificial intelligence (AI), it enables intelligent and proactive network operations. NDT continuously monitors traffic flow, device status, and potential failure signs by synchronizing with real network data. AI models trained on this data can predict traffic congestion or possible failures in advance and suggest appropriate countermeasures. Before applying any policy or structural changes to the actual network, these changes can be simulated within the digital twin to ensure safety and stability. This approach significantly improves network availability, reliability, and operational efficiency. A major advantage is self-diagnosis and self-healing, where AI autonomously detects issues and executes corrective actions without human intervention. Technologies like SDN (Software-Defined Networking), telemetry, and machine learning are integrated to support a wide range of use cases. In addition, NDT-based AI operations represent a future-oriented approach to managing complex networks more securely and intelligently.

## 6. Security Considerations

When providing AI services, it is essential to consider security measures to protect sensitive data such as network configurations, user information, and traffic patterns. Robust privacy measures must be in place to prevent unauthorized access and data breaches.

Implementing effective access control mechanisms is essential to ensure that only authorized personnel or systems can access and modify the network management infrastructure. This involves managing user privileges, using authentication mechanisms, and enforcing strong password policies.

### 6.1. Data Validation and Bias Mitigation

Ensuring the quality and integrity of the training data is critical for AI model performance. This involves several key steps:

- \* **Data Validation Procedures:** Implement rigorous validation processes, including data cleaning to remove noise and irrelevant data, consistency checks to ensure uniformity across datasets, and anomaly detection to address outliers that could skew model training.
- \* **Bias Detection and Mitigation:** Ensure fairness and accuracy by using diverse data sources, applying fairness metrics, and performing adversarial testing to identify and mitigate biases.

### 6.2. AI Model Vulnerability Detection

Regularly auditing and evaluating the AI model is essential to detect and address vulnerabilities:

- \* **Performance Monitoring:** Continuously monitor the AI model's performance to identify any degradation or unexpected behavior.
- \* **Security Testing:** Conduct security tests such as penetration testing and adversarial attacks to evaluate the model's robustness.
- \* **Update and Patch Management:** Keep the AI model and its underlying systems updated with the latest security patches and improvements.

Enhancing the explainability and transparency of AI models is also important:

- \* Model Interpretability Tools: Use tools and techniques to interpret the AI model's decisions and understand the factors influencing its predictions.
- \* Transparent Reporting: Provide clear and transparent reports on the AI model's performance, biases, and decision-making processes to stakeholders.

## 7. IANA Considerations

There are no IANA considerations related to this document.

## 8. Acknowledgements

TBA

## 9. Informative References

- [Y.4600] Union, I. T., "Recommendation ITU-T Y.4600 (2022), Requirements and capabilities of a digital twin system for smart cities.", August 2022.
- [I-D.irtf-nmrg-ai-challenges]  
François, J., Clemm, A., Papadimitriou, D., Fernandes, S., and S. Schneider, "Research Challenges in Coupling Artificial Intelligence and Network Management", Work in Progress, Internet-Draft, draft-irtf-nmrg-ai-challenges-05, 18 March 2025, <<https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-ai-challenges-05>>.
- [MEC.IEG006]  
ETSI, "Mobile Edge Computing; Market Acceleration; MEC Metrics Best Practice and Guidelines", Group Specification ETSI GS MEC-IEG 006 V1.1.1 (2017-01), January 2017.

## Authors' Addresses

SeokBeom Oh  
KSA  
Digital Transformation Center, 5  
Teheran-ro 69-gil, Gangnamgu  
Seoul  
06160  
South Korea  
Phone: +82 2 1670 6009  
Email: [isb6655@korea.ac.kr](mailto:isb6655@korea.ac.kr)

Yong-Geun Hong  
Daejeon University  
62 Daehak-ro, Dong-gu  
Daejeon  
34520  
South Korea  
Phone: +82 42 280 4841  
Email: yonggeun.hong@gmail.com

Joo-Sang Youn  
DONG-EUI University  
176 Eomgwangno Busan\_jin\_gu  
Busan  
614-714  
South Korea  
Phone: +82 51 890 1993  
Email: joosang.youn@gmail.com

Hyunjeong Lee  
Electronics and Telecommunications Research Institute  
218 Gajeong-ro, Yuseong-gu  
Daejeon  
34129  
South Korea  
Phone: +82 42 860 1213  
Email: hjlee294@etri.re.kr

Seung-Woo Hong  
ETRI  
218 Gajeong-ro Yuseong-gu  
Daejeon  
34129  
South Korea  
Phone: +82 42 860 1041  
Email: swhong@etri.re.kr

Hyun-Kook Kahng  
Korea University  
2511 Sejong-ro  
Sejong City  
Email: kahng@korea.ac.kr