Network Management
Internet-Draft
Intended status: Informational
Expires: 8 January 2026

P. Cassara'
A. Gotta
CNR-ISTI
G. Fioccola
A. Artigiani
Huawei Technologies
R. Burrai
E. Kolaj
SIRIUS Technology

7 July 2025

Generative AI for Intent-Based Networking
draft-cgfabk-nmrg-ibn-generative-ai-00

Abstract

This document describes how to specialize AI models in order to offer a scalable, efficient path to creating highly targeted generative models for Intent-Based Networking.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://giuseppefioccola.github.io/draft-cgfabk-nmrg-ibn-generative-ai/draft-cgfabk-nmrg-ibn-generative-ai.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-cgfabk-nmrg-ibn-generative-ai/.

Discussion of this document takes place on the Network Management Research Group mailing list (mailto:nmrg@irtf.org), which is archived at https://mailarchive.ietf.org/arch/browse/nmrg. Subscribe at https://www.ietf.org/mailman/listinfo/nmrg/.

Source for this draft and an issue tracker can be found at https://github.com/giuseppefioccola/draft-cgfabk-nmrg-ibn-generative-ai.

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
  1.1. Overview of IBN 4
  1.2. Role of Generative AI in IBN 4
  1.3. Motivation for AI Model Specialization 5
  1.4. Transfer Learning 5
2. Conventions and Definitions 6
3. Background Concepts 6
  3.1. Generative AI and LLMs in Networking 6
  3.2. Intent Parsing and Translation via Generative Models . . 6
  3.3. Fundamentals of LoRA 7
4. Specializing Generative AI with LoRA 7
  4.1. The Need for Model Specialization in IBN 7
  4.2. LoRA for Lightweight Fine-Tuning 7
  4.3. Comparison with Traditional Fine-Tuning and Pruning . . . 7
  4.4. Efficiency and Performance Gains in Network Models . . . 7
5. Hub: Repository of Specialized Models 7
  5.1. Concept and Architecture of a Hub 7
  5.2. Organizing and Indexing Adapters for Networking Tasks . . 8
  5.3. Examples of IBN-Targeted LoRA Adapters 8
  5.4. Benefits of Modular and Shareable Specializations 8

6. Flow: Model Fusion and Evolution 8
6.1. Concept of Flow for Adapter Composition 8
6.2. Fusion of Multiple Adapters to Generate New Models . . . 8
6.3. Workflow for Dynamic Specialization Based on Intent . . . 9
6.4. Case Study: Multi-Domain Network Management 9
7. Lifecycle of Specialized Models in IBN 9
7.1. Model Generation, Evaluation, and Deployment 9
7.2. Feedback Loops and Continuous Adaptation 10
7.3. Benchmarking Specialized Models: Accuracy, Latency, and Resource Consumption 10
8. Practical Frameworks and Tools 10
8.1. LoRA Fine-Tuning Libraries 10
8.2. LoRA Hub Implementations 10
8.3. Toolchains for LoRA Flow and Adapter Fusion 10
9. Logical Architecture for Model Management 10
9.1. Overview 11
9.2. Logical Architecture Diagram 11
9.3. Detailed Workflow 12
9.4. Models Adaptation, and Orchestration 13
10. Challenges and Research Directions 13
10.1. Model Interoperability in LoRA Flow 13
10.2. Ensuring Security and Robustness in Specialized Adapters 13
10.3. Governance of LoRA Repositories 13
10.4. Towards Autonomous Model Lifecycle Management in IBN . . 13
11. Security Considerations 13
12. IANA Considerations 13
13. References 14
13.1. Normative References 14
13.2. Informative References 14
Acknowledgments 14
Authors' Addresses 14

1. Introduction

This document describes how transfer learning techniques can be adopted to design generative AI specialized models for Intent-Based Networking (IBN). It also describes tools, such as Low-rank Adaptation (LoRA), for achieving efficient and scalable transfer learning in data networks for designing targeted generative models for Intent-Based Networking (IBN).

The objective of this document is define a framework for the rapid adaptation and composition of specialized knowledge, addressing the challenges identified in [AI-Challenges] and enabling the dynamic, multi-domain use cases of [IBN-UseCases].

Future work should focus on interoperability, governance, and autonomous management to fully realize the potential of this approach.

1.1. Overview of IBN

IBN represents a paradigm shift in network management, aiming to bridge the gap between business objectives and network configurations. Unlike traditional networking, which requires manual, device-level configurations, IBN allows operators to specify high-level intents (e.g., "ensure low latency for video traffic"), which the system then automatically translates, enforces, and continuously validates. Key references include [IBN-UseCases], which outlines IBN use cases across enterprise, data center, and service provider environments.

IBN fundamentally changes the role of network administrators by focusing on desired outcomes instead of manual command-line configurations. The system must be capable of not only translating intents into actionable policies but also of continuously validating whether those intents are met. This requires advanced mechanisms for policy enforcement, telemetry collection, and feedback loops that ensure alignment between high-level business goals and actual network performance. IBN can reduce operational complexity, improve network agility, and significantly lower the time required to deploy new services.

1.2. Role of Generative AI in IBN

Generative AI, particularly Large Language Models (LLMs), can enhance IBN by automating intent parsing, policy generation, and network troubleshooting. LLMs can understand natural language intents, generate low-level configurations, and adapt policies in real-time. This aligns with the challenges presented in [AI-Challenges], which stresses the need for models that can dynamically adapt to context and scale efficiently.

Generative AI can extend its role to include auto-remediation, dynamic configuration adjustments, and the creation of context-aware policies based on real-time network conditions. These capabilities are critical in environments with rapidly shifting traffic patterns, such as 5G networks, multi-cloud environments, and large-scale SDN deployments. By leveraging fine-tuned generative models, IBN systems can become self-configuring, self-optimizing, and self-healing, moving closer to the vision of fully autonomous networks.

1.3. Motivation for AI Model Specialization

Generic LLMs often lack the precision required for domain-specific applications like IBN. Specialization is critical to improve accuracy, reduce inference time, and minimize resource consumption. LoRA provides an efficient method to fine-tune large models using minimal computational resources, enabling targeted model specialization for distinct network domains or intent categories.

Specialization ensures that the AI system correctly interprets and applies complex domain-specific policies, such as those for Quality of Service (QoS), security compliance, and multi-vendor interoperability. Without specialization, generic models risk generating suboptimal or even invalid configurations. LoRA-based specialization enables the rapid creation and deployment of these focused models, supporting agile, intent-driven network operations while maintaining the scalability and cost-effectiveness required for wide deployment across heterogeneous infrastructures.

1.4. Transfer Learning

Transfer learning enables pretrained models to adapt to specific tasks with significantly less data and computational resources. In the context of IBN, this approach offers a dual advantage: enhancing the efficiency of model training and improving the reliability of intent recognition and execution. Indeed, it avoids the need for building models from scratch, which is not only resource-intensive but also prone to overfitting due to limited labeled network-specific data. By fine-tuning models on domain-specific datasets, developers can quickly create robust models that interpret network intents with high accuracy. Furthermore, it facilitates cross-domain knowledge integration, allowing the model to generalize better across various networking environments and topologies. Additionally, continual learning mechanisms can be integrated into the LLM-based IBN framework to refine performance over time, incorporating feedback loops that learn from real-world interactions and user corrections.

In conclusion, transfer learning significantly boosts the viability of LLM-based solutions in IBN by offering a cost-effective, accurate, and scalable method for intent interpretation and execution. This fusion of LLMs with transfer learning not only propels the automation of network operations but also ensures that the AI systems remain adaptive, interpretable, and aligned with organizational objectives in dynamic digital infrastructures.

2. Conventions and Definitions

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.

3. Background Concepts

3.1. Generative AI and LLMs in Networking

LLMs like GPT and BERT can process complex, unstructured inputs and generate contextually relevant outputs. In IBN, LLMs are used to translate high-level intents into vendor-agnostic policies and device-specific configurations. The integration of these models into network management systems is emerging as a critical research area, as outlined in [AI-Challenges].

3.2. Intent Parsing and Translation via Generative Models

An example flow in Figure 1 illustrates how an intent like "minimize latency for video traffic" is parsed by a generative model and translated into specific QoS policies.

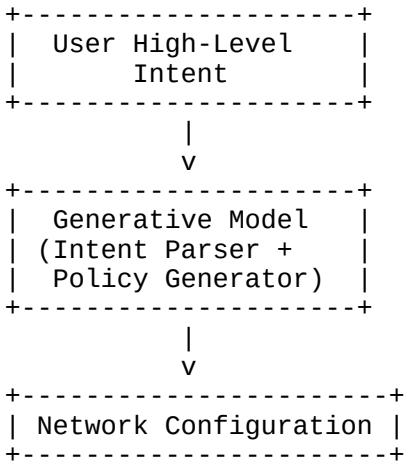

Figure 1: Example flow of intent parsing and translation

3.3. Fundamentals of LoRA

LoRA [LoRA] introduces trainable low-rank matrices into the layers of pre-trained models, allowing fine-tuning with significantly fewer parameters. This approach reduces the storage and computational footprint, enabling deployment of specialized models even on resource-constrained systems.

4. Specializing Generative AI with LoRA

4.1. The Need for Model Specialization in IBN

IBN requires high domain accuracy. For example, a generic model may misinterpret a telecommunications-specific intent. Specializing models using LoRA ensures they are finely tuned to the nuances of networking language, policies, and protocols.

4.2. LoRA for Lightweight Fine-Tuning

LoRA fine-tunes only small, low-rank matrices, leaving the rest of the model unchanged. For instance, adapting a generic LLM to focus on BGP policy generation could be achieved by training a small adapter on BGP-related datasets, reducing the compute requirements compared to full fine-tuning.

4.3. Comparison with Traditional Fine-Tuning and Pruning

Unlike traditional fine-tuning, which adjusts the entire model, or pruning, which removes less important weights, LoRA introduces new parameters without overwriting existing knowledge. This allows multiple specialized LoRA adapters to coexist and be swapped or combined as needed.

4.4. Efficiency and Performance Gains in Network Models

Empirical studies show that LoRA adapters can reduce memory consumption by over 70% compared to full fine-tuning. For example, fine-tuning a model for IPv6 routing policy generation using LoRA reduces the required GPU VRAM from 16 GB to approximately 5 GB while maintaining performance.

5. Hub: Repository of Specialized Models

5.1. Concept and Architecture of a Hub

A Hub is a structured repository where specialized adapters are stored, indexed, and shared. It enables efficient reuse and modularity. The architecture typically includes:

* Adapter storage

* Metadata index (domain, use case, version)

* Dependency tracking for composite models

5.2. Organizing and Indexing Adapters for Networking Tasks

Adapters can be categorized by networking domain (e.g., security, QoS, routing) and intent granularity (e.g., traffic type, SLA constraint). Cross-indexing improves discoverability and composability.

5.3. Examples of IBN-Targeted LoRA Adapters

* Adapter A: IPv6 Segment Routing configuration

* Adapter B: QoS policy for video prioritization

* Adapter C: Security baseline for zero-trust architectures

5.4. Benefits of Modular and Shareable Specializations

Operators can quickly compose or replace adapters based on current intents, improving agility and reducing the cost of specialization. LoRA Hub supports collaboration and community-driven model evolution.

6. Flow: Model Fusion and Evolution

6.1. Concept of Flow for Adapter Composition

Flow refers to the systematic combination of multiple adapters to form a new, composite model capable of handling complex, multi-domain intents.

6.2. Fusion of Multiple Adapters to Generate New Models

Example: Combining an adapter for SLA enforcement with another for security compliance to address intents like "guarantee low latency for critical applications within zero-trust boundaries."

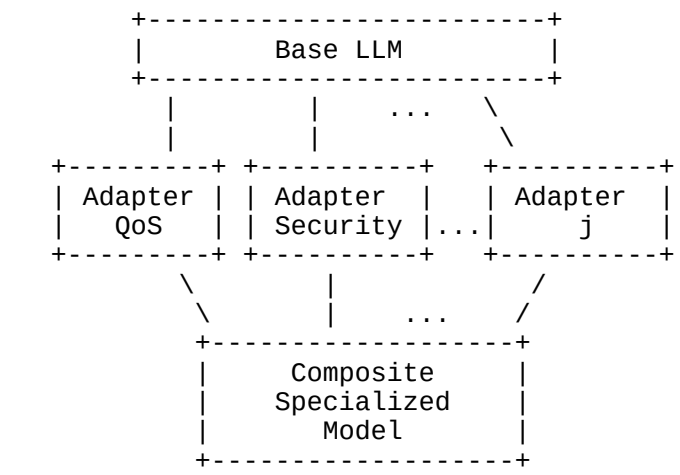

Figure 2: Fusion of multiple adapters to generate new models

6.3. Workflow for Dynamic Specialization Based on Intent

1. Parse user intent
2. Query LoRA Hub for relevant adapters
3. Dynamically load and fuse adapters
4. Generate configuration

6.4. Case Study: Multi-Domain Network Management

A service provider managing both MPLS and SDN domains can fuse LoRA adapters for each to generate cross-domain policies automatically in response to high-level intents.

7. Lifecycle of Specialized Models in IBN

7.1. Model Generation, Evaluation, and Deployment

* Generation: Fine-tuning on domain datasets
* Evaluation: Accuracy, latency, resource profiling
* Deployment: Containerized adapters for on-demand loading

7.2. Feedback Loops and Continuous Adaptation

In-network telemetry provides real-time feedback on policy effectiveness, enabling further fine-tuning or adapter updates, aligning with the adaptive model requirements discussed in [AI-Challenges].

7.3. Benchmarking Specialized Models: Accuracy, Latency, and Resource Consumption

Key benchmarks include:

* Policy generation latency (< 50 ms target)
* Memory footprint reduction (70% over full models)
* Domain-specific intent accuracy (> 90%)

8. Practical Frameworks and Tools

8.1. LoRA Fine-Tuning Libraries

* PEFT (Parameter-Efficient Fine-Tuning) [HF-PEFT]
* Hugging Face LoRA integration

8.2. LoRA Hub Implementations

* Model repositories using Hugging Face Hub structure
* Metadata indexing via JSON-LD for semantic search

8.3. Toolchains for LoRA Flow and Adapter Fusion

* Adapter composition APIs
* Intent parsing frontends
* Deployment via Kubernetes with sidecar loading of adapters

9. Logical Architecture for Model Management

9.1. Overview

Drawing inspiration from the Analytics Logical Function (AnLF) and Model Training Logical Function (MTLF) architectures defined in 3GPP NWDAF, this section proposes a logical architecture to manage, store, and orchestrate AI models for Intent-Based Networking. The architecture introduces the following logical components:

* Model Repository Function (MRF): Stores adapters and base models.

* Model Training and Specialization Function (MTSF): Handles fine-tuning, evaluation, and versioning.

* Model Fusion and Composition Function (MFCF): Manages flow processes to create composite models.

* Intent Processing and Adapter Orchestration Function (IPOF): Dynamically selects, composes, and deploys adapters based on parsed intents.

* Telemetry Feedback Function (TFF): Continuously monitors performance and triggers model re-specialization when required.

9.2. Logical Architecture Diagram

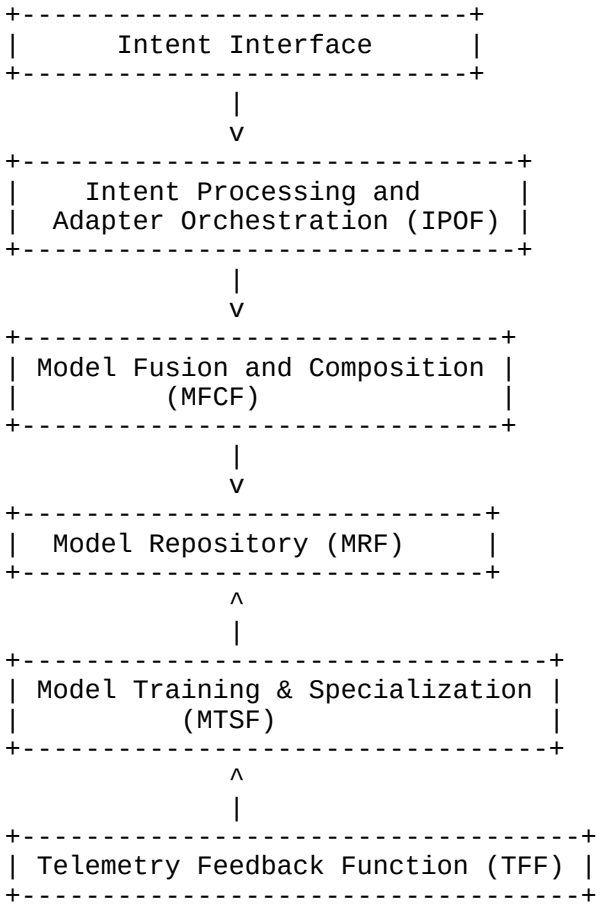

Figure 3: Logical Architecture Diagram

9.3. Detailed Workflow

1. Intent Reception: User or application submits a high-level intent through the Intent Interface.

2. Adapter Orchestration: IPOF parses the intent, queries the MRF for relevant adapters, and orchestrates dynamic fusion via MFCF.

3. Model Fusion: MFCF composes adapters into a composite model tailored to the intent.

4. Deployment: The composite model is deployed into the IBN system to generate specific configurations.

5. Telemetry Feedback: The TFF monitors the applied configurations' impact on network KPIs and feeds insights back to IPOF and MTSF.

6. Re-Specialization: If performance deviates from expected thresholds, MTSF initiates LoRA re-training or fine-tuning using new data collected via TFF.

9.4. Models Adaptation, and Orchestration

This architecture closely follows the modularity principles of NWDAF's AnLF for real-time analytics and MTLF for model training and distribution, enabling scalability, continuous learning, and rapid specialization across distributed edge and core network nodes. Each function can be independently scaled and containerized, ensuring fault tolerance and efficient resource utilization. This logical separation allows efficient model lifecycle management, supports federated learning, and can integrate with standard IBN orchestrators via REST APIs or 3GPP-compatible interfaces.

10. Challenges and Research Directions

10.1. Model Interoperability in LoRA Flow

Ensuring consistent parameterization and preventing interference when fusing adapters.

10.2. Ensuring Security and Robustness in Specialized Adapters

Preventing malicious adapter injection and ensuring adapters respect security constraints.

10.3. Governance of LoRA Repositories

Managing trust, versioning, and deprecation of adapters.

10.4. Towards Autonomous Model Lifecycle Management in IBN

Developing self-optimizing pipelines for adapter generation, validation, and retirement.

11. Security Considerations

TODO

12. IANA Considerations

This document has no IANA actions.

13. References

13.1. 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/rfc/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/rfc/rfc8174>.

13.2. Informative References

[AI-Challenges] Fran巽ois, 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>.

[HF-PEFT] "Hugging Face PEFT", n.d., <https://github.com/huggingface/peft>.

[IBN-UseCases] Yao, K., Chen, D., Jeong, J. P., Wu, Q., Yang, C., Contreras, L. M., and G. Fioccola, "Use Cases and Practices for Intent-Based Networking", Work in Progress, Internet-Draft, draft-irtf-nmrg-ibn-usecases-00, 28 March 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-ibn-usecases-00>.

[LoRA] Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models", arXiv, DOI 10.48550/ARXIV.2106.09685, 2021, <https://doi.org/10.48550/ARXIV.2106.09685>.

Acknowledgments

TODO

Authors' Addresses

Pietro Cassara’
CNR-ISTI
Pisa
Italy
Email: pietro.cassara@isti.cnr.it

Alberto Gotta
CNR-ISTI
Pisa
Italy
Email: alberto.gotta@isti.cnr.it

Giuseppe Fioccola
Huawei Technologies
Vimodrone (Milan)
Italy
Email: giuseppe.fioccola@huawei.com

Aldo Artigiani
Huawei Technologies
Turin
Italy
Email: aldo.artigiani@huawei.com

Riccardo Burrai
SIRIUS Technology
Prato
Italy
Email: riccardo.burrai@siriustec.it

Emiljan Kolaj
SIRIUS Technology
Prato
Italy
Email: emiljan.kolaj@siriustec.it