Network Acceleration for AI Workloads in Telco Networks

Filippo Cugini, Ahmed Salah Tawfik Ibrahim
Rana Abu Bakar, Francesco Paolucci
CNIT
Pisa, Italy

Ashley Degl'Innocenti, Layal Ismail, Emilio Paolini
Andrea Sgambelluri, Piero Castoldi
Scuola Superiore Sant'Anna
Pisa, Italy

Oscar Gonzalez de Dios
Juan Pedro Fernandez-Palacios
Telefonica
Madrid, Spain

Benedetta Picano
University of Florence
Florence, Italy

Juan Jose Vegas Olmos
Nvidia
Denmark

***Abstract*—This paper explores how Telco networks can enable distributed Artificial Intelligence (AI) by equipping Central Offices with GPUs and leveraging network acceleration. We discuss Remote Direct Memory Access (RDMA) over Ethernet and its evolution towards Ultra-Ethernet, Data Processing Units (DPU), and in-network computing as key enablers to transform geographically distributed infrastructure into a cost-effective programmable AI platform.**

***Index Terms*—Federated Learning, P4, Telemetry-Aware, NVFlare**

I. Introduction

The rapid growth of Artificial Intelligence (AI) workloads is driving an unprecedented demand for large-scale computing infrastructure. Among the various AI workloads, the training of Large Language Models (LLMs) is particularly demanding. However, in several regions — most notably in Europe — the deployment of hyperscale AI data centers is increasingly constrained by the high cost of building new infrastructure (land, power delivery, cooling systems, and grid upgrades, environmental considerations, etc.) [1]. In contrast, telecommunication operators already own and operate a dense network of Central Offices (CO), equipped with power, cooling, and high-capacity connectivity, offering a unique opportunity to repurpose existing infrastructure for AI workloads. Moreover, the evolution toward 6G, which envisions AI-native and compute-integrated networks delivering advanced services close to the end user, is accelerating the deployment of GPU resources within COs [2]. When interconnected through high-performance, low-latency networks, these GPU-enabled sites can be federated into a single distributed AI infrastructure.

However, the software and infrastructure currently used to deploy and operate AI workloads have been primarily designed for centralized data centers. These solutions assume tightly coupled compute clusters, short physical distances, and homogeneous networking characterized by a uniform, high-bandwidth, low-latency fabric in which links, switches, congestion control mechanisms, and failure domains are largely consistent and optimized for east–west traffic patterns [3].

Managing distributed AI workloads across multiple sites introduces new challenges in terms of latency, bandwidth efficiency, synchronization, resource orchestration, and cost-effectiveness [4]. When existing AI frameworks and networking solutions are directly applied to Telco networks, the resulting systems exhibit significant performance degradation and poor resource utilization [5].

Addressing this gap requires a comprehensive approach encompassing enhanced networking and system-level capabilities specifically tailored for distributed AI deployments.

First, the underlying networking layer must support traffic dynamicity, prioritization, and deterministic performance, while ensuring coexistence with other 6G services through mechanisms such as network slicing. This includes highly dynamic optical transport networks, hardware acceleration and built-in security to offload data movement, isolation, and protection functions from host CPUs and GPUs. Second, the infrastructure must provide network awareness, enabling continuous monitoring of latency, congestion, and available bandwidth across geographically distributed sites. Beyond the networking layer, distributed AI also requires tools and execution frameworks (such as federated learning, data-parallel processing, and map-reduce–style abstractions) that are explicitly designed to operate across multiple heterogenous Telco resources. Finally, effective utilization of a distributed Telco AI infrastructure calls for scheduling and orchestration algorithms that jointly consider computing and networking resources, enabling intelligent placement of AI workloads based on both compute availability and network conditions.

In this work, we discuss these challenges and show how a comprehensive approach spanning optical networks, accelerated networking, network awareness, AI frameworks, and joint compute–network scheduling can enable efficient AI execution over geographically distributed Telco infrastructures.

This work was partly funded by the SMARTY Project (G.A. 101140087)

II. Challenges of LLM Training over Telco Infrastructures

The training of LLMs typically follows a synchronous iterative process in which multiple workers exchange model updates with an aggregator. As a result, the overall training time is determined by the slowest participating node, making training performance strongly dependent on tail latency.

Federated learning frameworks such as Flower [6] or Nvidia Flare (NvFlare) [7] are primarily designed for data center environments, where network performance is relatively uniform, predictable and provisioned. Therefore, these frameworks focus on compute- and GPU-centric scheduling and workload distribution, while abstracting away network dynamics.

When such network-unaware AI frameworks are deployed over geographically distributed Telco infrastructures, the resulting mismatch between communication patterns and heterogeneous network conditions leads to significant performance degradation and inefficient utilization of both compute and network resources.

A testbed consisting of one parameter server (H) and two sets of clients (L_1 and L_2) was set up to implement a federated LLM training scenario. Clients in L_1 are connected to H through a fast path including high-quality links at 100 Gbps and negligible congestion. In contrast, clients in L_2 experience degraded network conditions, including a slow path with persistent congestion and traversal of lower-rate links. Training orchestration was performed using the NvFlare framework, which coordinates synchronous federated learning rounds across all participating clients.

Fig. 1 shows the network traffic observed during federated LLM training under uncongested conditions. The figure highlights the asymmetric behavior of the fast and slow paths during model synchronization. In the initial phase, H distributes the model parameters to all participating clients in both L_1 and L_2. Each client performs a local training task on its assigned data partition using the GPU resources available at the CO, and subsequently sends the updated model weights back to H. While the communication pattern of this distributed workload is predictable, it is inherently bursty. Network traffic remains negligible during local computation phases; however, upon completion of local training, all clients transmit their learned weights to the parameter server, generating intense traffic bursts. As shown in Fig. 1, traffic exchanged over the fast path associated with clients in L_1 is delivered within short, well-defined bursts that complete quickly. In contrast, traffic on the slow path associated with clients in L_2 is significantly stretched over time due to limited bandwidth and congestion. While synchronization over the fast path completes within a relatively short time window (in the order of several hundred seconds), synchronization involving L_2 clients persists for a much longer duration, effectively dominating the overall training time.

This behavior highlights the tail-latency domination effect in federated LLM training: the overall training time is dominated by the slowest clients. In this scenario, executing two training

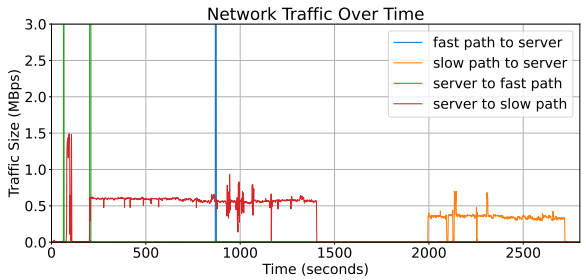

Fig. 1. Network traffic over time during federated LLM training using FLARE, highlighting the different behavior of fast and slow paths during model synchronization.

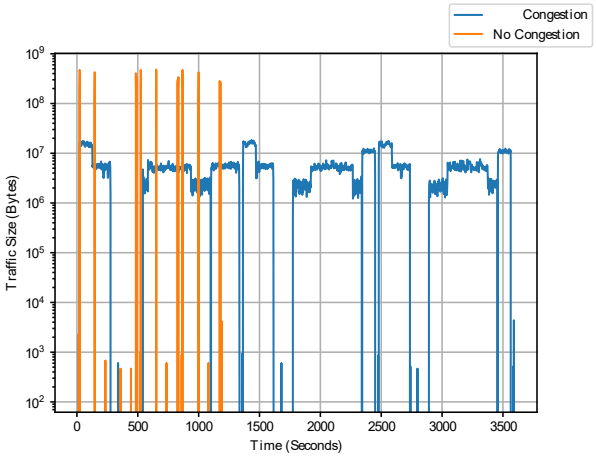

Fig. 2. Effect of congestion on training time using Flower.

rounds using only L_1 would have completed faster than a single round distributed across both L_1 and L_2, due to the tail latency introduced by L_2. Although not shown in the figure, subsequent training rounds exhibit the same behavior, confirming that NvFlare currently adopts a compute-centric client selection and scheduling model, leaving network-aware optimization as an opportunity for future enhancement.

We repeated a similar experiment using the Flower framework and observed the same qualitative behavior, as shown in Fig. 2. Despite differences in the experimental setup, training performance remains dominated by congested clients, confirming that Flower, like NvFlare, does not incorporate network awareness in client selection or scheduling.

An additional challenge in Telco-based AI infrastructures stems from the diversity of AI execution models that must be supported concurrently. Unlike centralized cloud environments, which typically optimize for a single dominant training paradigm, Telco networks are expected to host heterogeneous AI workflows with fundamentally different communication and computation patterns. For instance, in some scenarios a Telco operator may only provide connectivity for AI workloads executed at the edge or at customer premises, such as hospitals or enterprises, where local resources perform training or inference and the network must efficiently handle bursty

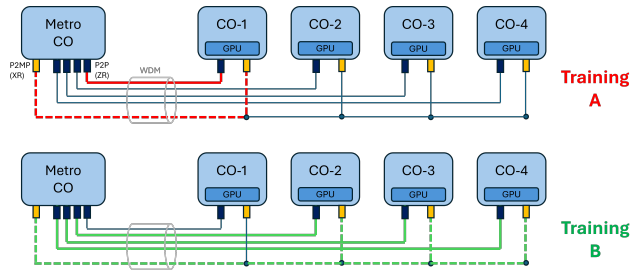

Fig. 3. Telco scenario for distributed LLM training using P2MP optical connectivity based on DSCM, where digital subcarriers are dynamically allocated to distinct sets of GPU-enabled Central Offices to support multiple training jobs over a shared optical infrastructure.

synchronization traffic. In other cases, the Telco infrastructure may actively participate in data distribution and processing, receiving datasets from external entities and orchestrating their execution across multiple COs. Such scenarios naturally require paradigms such as map-reduce, split learning, or hybrid distributed training models. These heterogeneous workflows may need to be executed simultaneously over the same physical infrastructure, each with distinct requirements in terms of bandwidth, latency, synchronization, and resource allocation. As a result, AI frameworks designed for homogeneous cloud environments (such as Flower, NvFlare, Horovod, or DeepSpeed), cannot be directly applied as-is. Telco infrastructures require more general and flexible mechanisms capable of jointly managing multiple AI paradigms, provided with network awareness, dynamically adapting to workload characteristics, and coordinating compute and network resources across distributed sites.

III. Network Acceleration Technologies

A. Highly dynamic optical transport infrastructure

Supporting distributed AI workloads in Telco infrastructures requires a comprehensive network acceleration approach that extends beyond packet-based mechanisms, requiring optical transport networks to evolve in order to efficiently handle the dynamic communication patterns of AI workloads. Indeed, traditional optical transport designs, optimized for relatively stable traffic matrices and long-lived flows, are less effective to support the highly dynamic, bursty, and synchronized traffic generated by distributed AI workloads [8].

However, achieving such levels of flexibility and reactivity in optical transport networks is non-trivial, as key components such as lasers, modulators, and optical amplifiers (e.g., EDFAs) exhibit inherently analog behavior and are traditionally operated under relatively static configurations.

An interesting direction to address this challenge is offered by Digital Subcarrier Multiplexing (DSCM) technology, particularly in point-to-multipoint (P2MP) configurations. Although currently proposed mainly for metro and x-haul scenarios [9]–[11], DSCM-based P2MP architectures have the potential to provide adequate level of flexibility since they avoid mechanical reconfigurations.

Fig. 3 shows a representative scenario for distributed LLM training in a Telco infrastructure. In this scenario, each CO is connected through traditional packet-optical connectivity, which is augmented by an additional P2MP optical connection based on DSCM, depicted as dotted lines in the figure. Specifically, the P2MP optical hub is physically connected to all the COs considered in the scenario, providing an additional shared high-capacity optical substrate.

Two independent Telco clients request the execution of distinct LLM training jobs, denoted as A and B in the figure, and the Telco operator assigns each client a distinct set of COs equipped with GPU resources. Let L_1 denote the set of COs participating in the first client's training job, namely CO-1 assigned to job A, and L_2 the distinct set assigned to the second client, namely CO-2, CO-3, and CO-4 assigned to job B.

In this setting, an LLM parameter server located at the Metro CO coordinates the execution of federated training tasks across the COs assigned to each client. Since the communication pattern of federated LLM training is largely predictable and characterized by alternating computation and synchronization phases, the optical connectivity can be proactively configured to match the active training phase. Digital subcarriers are dynamically assigned in a time-multiplexed manner. During the execution of training job A, subcarriers are allocated to the COs in L_1, which in this scenario corresponds to CO-1, providing additional bandwidth during synchronization phases. Once the first training job completes or transitions to a compute-dominated phase, the same subcarriers are reallocated to the distinct set L_2 to support the second client's training job B. In this way, the predictability of the workload enables efficient, on-demand allocation of optical capacity, allowing multiple training jobs to share the same physical infrastructure while mitigating congestion and reducing tail latency during critical synchronization periods.

We experimentally validated this concept using current DSCM-based equipment by successfully activating and deactivating digital subcarriers on demand at leaf endpoints. Although current commercial implementations do not yet expose full subcarrier flexibility, this reflects the scope of functionality presently available rather than an inherent limitation of DSCM, which natively supports rapid, software-controlled reconfiguration.

B. Programmable packet networks: programmable switches and DPU

Programmable packet networks are a key enabler for network-aware distributed AI in Telco infrastructures, providing fine-grained control and visibility directly in the data plane. Unlike fixed-function switches, P4-programmable devices support custom, line-rate packet processing, allowing the network fabric to actively participate in AI workload orchestration [12]. Such switches enable application-driven packet classification, in-band telemetry, and decentralized feature extraction, exposing per-flow and per-hop metrics including queue occupancy, link utilization, timestamps, and ECN

markings. Beyond telemetry, they support advanced traffic steering and load balancing beyond hash-based ECMP, leveraging per-packet or per-flow state to implement adaptive path selection and priority-aware forwarding. These capabilities help mitigate incast and synchronized bursts during collective communication, particularly in Telco networks with non-ideal Clos or partial-mesh topologies.

While programmable switches provide in-fabric visibility and forwarding control, Data Processing Units (DPUs) extend programmability to the network edge by offering a dedicated execution environment for transport-layer and endpoint-adjacent functions. By offloading network stacks, RDMA engines, and control logic, DPUs decouple network processing from application execution, enabling fine-grained packet handling without impacting AI workloads on host CPUs or GPUs. In distributed AI deployments, DPUs can terminate RDMA connections and implement transport-layer offloads such as reliable delivery, congestion response, and flow control. Local processing of telemetry signals, including RTT variations, ECN rates, and queue occupancy, enables closed-loop congestion control with faster reaction times than centralized SDN approaches. DPUs also serve as effective enforcement points for traffic isolation, prioritization, and security in multi-tenant Telco infrastructures, supporting deterministic AI performance while enabling coexistence with latency-sensitive 6G services through mechanisms such as network slicing.

C. RDMA with Data Path Acceleration (DPA)

Conventional RDMA deployments rely on host-side processing for control-plane tasks such as completion handling and congestion response, which can introduce latency and limit scalability at high link rates. To overcome this limitation, we leverage the Data Path Acceleration (DPA) capabilities of SmartNIC to offload RDMA control and event processing directly onto the NIC. DPA provides a programmable execution environment within the network interface, allowing event-driven kernels to execute in response to RDMA completions and network feedback without host CPU intervention.

In our design, the RDMA application remains unmodified on the host and uses the DOCA RDMA API, while a DPA application is loaded during initialization and attached to hardware completion contexts. After setup, steady-state control decisions, such as rate adaptation and pacing updates, are computed by DPA threads and written to hardware-visible state, enabling low-latency enforcement at line rate. As shown in Fig. 4, this separation between host control and DPA execution moves performance-critical logic closer to the hardware, improving responsiveness and scalability under high-traffic conditions.

1) Programmable Congestion Control (PCC) for DPA-Based RDMA: Traditional hardware congestion-control mechanisms are largely fixed-function and lack the flexibility to adapt to workload. To overcome this limitation, our system leverages DOCA PCC to implement custom congestion control algorithms that execute directly on the BlueField-3 DPA, enabling in-NIC control decisions without host intervention.

The PCC framework is organized around two roles: the Reaction Point (RP) and the Notification Point (NP). The RP monitors congestion-related events and dynamically adjusts RDMA transmission rates, while the NP processes probe responses and explicit congestion notifications. Both execute on the DPA and are managed by the DOCA PCC runtime.

Each RDMA flow maintains a PCC context with per-flow state, including transmission rate, baseline RTT, EWMA RTT estimate, and credit-based pacing parameters. Congestion feedback is obtained via probe packets, primarily through Congestion Control Message After Drop (CCMAD), providing RTT visibility without modifying RDMA payloads. Based on this feedback, the PCC algorithm updates rate and credit parameters by writing to hardware-managed result structures for timely enforcement.

The congestion logic is implemented using user-defined PCC callbacks executed on the DPA. These callbacks initialize the flow state, process congestion events, and update rate and pacing parameters at runtime, enabling policy customization while preserving hardware-acceleration benefits. For robustness, DOCA PCC supports high-availability operation with multiple registered PCC processes, where one instance is active and others remain in standby mode for automatic failover.

2) A path to Ultra-Ethernet: Ultra-Ethernet aims to address these challenges by promoting a more programmable and telemetry-aware Ethernet fabric, with explicit support for low-latency transport, fine-grained congestion signaling, and in-network control. A key principle of Ultra-Ethernet is the ability to react to congestion events close to the data path, thereby reducing control-loop latency and improving network stability under load.

The proposed PCC-based design aligns with this vision by moving congestion control logic into the NIC data plane. By executing rate adaptation and feedback processing directly on the DPA, the system enables fast, per-flow congestion response without host CPU intervention. The use of probe-based telemetry, such as CCMAD, provides accurate and timely visibility into network conditions, which is essential for scalable RDMA transport, as shown by the RDMA read and write results in Fig 5. From this perspective, DOCA PCC can be viewed as an incremental and practical step toward Ultra-Ethernet principles. It demonstrates how programmable congestion control, combined with hardware acceleration and in-network telemetry, can improve adaptability and performance in modern Ethernet-based RDMA networks, while remaining compatible with existing Ethernet infrastructure.

D. RDMA and Ultra-Ethernet

RDMA over Ethernet (RoCEv2) enables high-performance distributed AI communication through zero-copy transfers, kernel bypass, and direct GPU-to-GPU data movement, reducing CPU overhead and improving latency and throughput for collective operations such as all-reduce and parameter synchronization in LLM training. Conventional RoCEv2 deployments primarily rely on Priority Flow Control (PFC) and

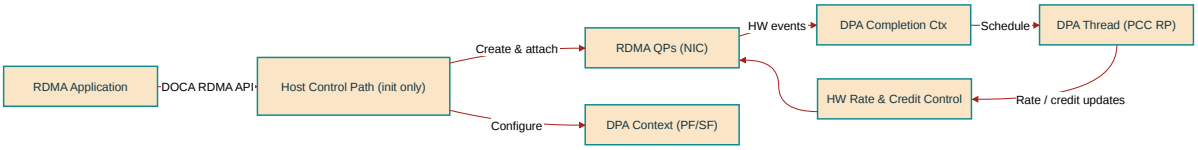

Fig. 4. DPA-based RDMA architecture on BlueField-3.

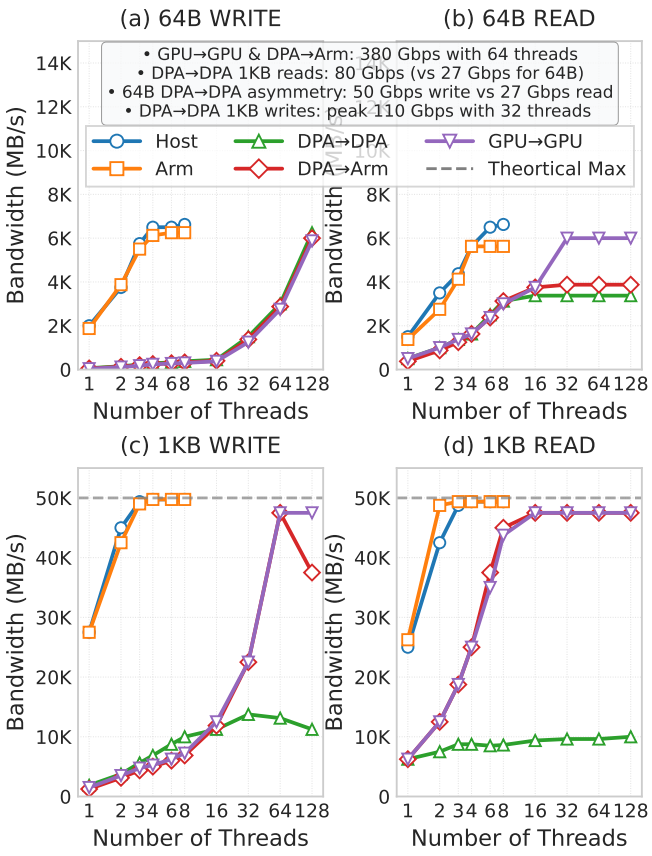

Fig. 5. BlueField-3 RDMA bandwidth scaling on 400GbE.

Explicit Congestion Notification (ECN) for congestion management and assume tightly coupled, homogeneous data center fabrics. These assumptions do not hold in Telco environments, where longer RTTs, heterogeneous link capacities, multi-domain routing (e.g., ECMP), and dynamic congestion can amplify tail latency and degrade synchronization efficiency. Ultra-Ethernet extends the RDMA model with AI-centric enhancements, including ECN-aware and telemetry-informed congestion control, adaptive multipath forwarding such as packet spraying, and both sender-based rate adaptation and receiver-based credit mechanisms to mitigate incast and last-hop queue buildup. Additional transport-level optimizations, including selective acknowledgments and out-of-order packet handling, further reduce retransmission overhead and tail latency. While Ultra-Ethernet is currently defined at the specification level, its practical deployment in Telco infrastructure requires careful system-level integration across heterogeneous, multi-domain networks.

IV. Pervasive Telemetry and Network Awareness

As shown in the previous sections, the lack of network awareness in current federated learning frameworks can lead to severe tail-latency domination, where congested paths or slow clients dictate the overall training time. Addressing this limitation requires pervasive telemetry mechanisms capable of exposing real-time network state at flow-level granularity, rather than relying on coarse, aggregated monitoring information.

As anticipated, recent advances in programmable data planes enable the extraction of detailed traffic features directly within the network, leveraging technologies such as P4-based in-band network telemetry and decentralized feature extraction [13]–[15]. In this paradigm, programmable switches perform application-driven feature extraction at line rate, selectively collecting packet- and flow-level metrics, such as per-hop latency, queue occupancy, burstiness indicators, packet timestamps, and protocol-specific metadata. At the same time, it is critical to avoid overwhelming telemetry controllers and AI frameworks with excessive data volumes. Decentralized feature extraction (DFE) addresses this challenge by exporting only relevant, pre-processed features, rather than raw traffic traces, significantly reducing telemetry overhead while preserving actionable information [16].

In the proposed DFE Telemetry (DFET), with the full P4 pipeline shown in Fig. 6, flow traffic is identified through match-action rules over 5-tuples associated with training sessions. Upon classification, selected packets are cloned and processed in the egress pipeline, where DFET performs programmable layer-level and field-level extraction according to control-plane configuration. Extracted fields include L3/L4 identifiers, ingress/egress timestamps, hop latency, enqueue/dequeue queue depth, and dequeue time delta. Telemetry information is encapsulated into compact UDP reports and streamed toward external collectors. The two-phase configuration mechanism enables activation of only the required layers and fields on a per-flow basis, minimizing report size and processing overhead. For training tasks acceleration, DFET can expose congestion indicators (queue depth, latency, timestamp deltas) to detect synchronization-induced bottlenecks during gradient uploads, per-flow load metrics (session identifiers and counters) to assess bandwidth utilization and fairness, and transport-level dynamics (e.g., TCP flags and RTT inference via timestamp correlation) to distinguish compute-bound from network-bound events. Because extraction is flow-specific and

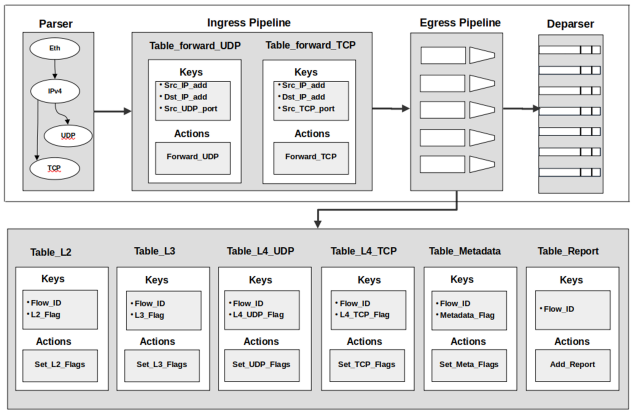

Fig. 6. P4 pipeline of Decentralized Feature Extraction Telemetry function.

dynamically configurable, DFET granularity can be adapted across training phases, enabling fine monitoring during aggregation and reduced profiling during local computation.

The resulting real-time, per-flow metadata supports a closed control loop between AI workloads logic and network management. Congestion signals can trigger different training clients activation/involvement, adaptive aggregation strategies, partial update selection, gradient compression, or local-epoch scaling, while throughput imbalance can drive SDN-based path reconfiguration or slice prioritization for model-update traffic. Experimental results obtained with software switch implementation show moderate latency overhead due to additional parsing, table lookups, and report generation. Hardware P4 targets are expected to introduce low resource overhead and negligible latency (in the order of few microseconds) while sustaining line-rate operation.

V. AI Workload Control and Network-Aware Orchestration

Although this work focuses on network acceleration mechanisms and experimental validation of network-induced bottlenecks, a key open challenge is the design of network-aware AI workload control frameworks. Current AI execution systems are predominantly compute-centric, making scheduling decisions based on GPU and memory utilization while treating the network as a black box. This model is insufficient in geographically distributed Telco infrastructures, where heterogeneous and time-varying network conditions can dominate end-to-end performance and tail latency.

A future direction is the development of a generic control layer that integrates real-time network telemetry, such as congestion signals, RTT variability, bandwidth availability, and path asymmetry, with application-level information, including training phase and synchronization patterns. Based on this joint visibility, the controller could dynamically adapt workload placement, synchronization timing, and resource allocation across distributed AI paradigms (e.g., federated learning, data-parallel training, distributed inference). Key challenges include scalability, closed-loop control stability, abstraction across heterogeneous network domains, and seamless integration with existing AI frameworks.

VI. Conclusions

This paper investigated how Telco networks can support distributed AI workloads by equipping COs with GPUs and leveraging network acceleration. We showed that AI frameworks designed for centralized data centers suffer from severe performance degradation when deployed over heterogeneous Telco infrastructures, where network conditions strongly impact tail latency. Then, we discussed a comprehensive set of enabling technologies, including dynamic optical transport, programmable packet networks, RDMA and Ultra-Ethernet, and pervasive telemetry, highlighting the need for tighter integration between AI workloads and network infrastructures.

References

[1] X. Chen, X. Wang, A. Colacelli, M. Lee, and L. Xie, "Electricity demand and grid impacts of ai data centers: Challenges and prospects," 2025.

[2] O. T. Basaran and F. Dressler, "Xai-on-ran: Explainable, ai-native, and gpu-accelerated ran towards 6g," 2025.

[3] A. Barceló *et al.*, "Offloading artificial intelligence workloads across the computing continuum by means of active storage systems," *Future Generation Computer Systems*, vol. 178, p. 108271, May 2026.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2023.

[5] K. Le, N. Luong-Ha, M. Nguyen-Duc, D. Le-Phuoc, C. Do, and K.-S. Wong, "Exploring the practicality of federated learning: A survey towards the communication perspective," 2024. [Online]. Available: https://arxiv.org/abs/2405.20431

[6] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. De Gusmão *et al.*, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[7] H. R. Roth, Y. Cheng, Y. Wen, I. Yang, Z. Xu, Y.-T. Hsieh, K. Kersten, A. Harouni, C. Zhao, K. Lu *et al.*, "Nvidia flare: Federated learning from simulation to real-world," *arXiv preprint arXiv:2210.13291*, 2022.

[8] R. Morro *et al.*, "Disaggregated and transparent multi-band optical continuum across access, horseshoe aggregation, and metro IPoWDM networks," *Journal of Optical Communications and Networking*, vol. 17, no. 10, pp. C170–C181, 2025.

[9] D. Welch *et al.*, "Digital subcarrier multiplexing: Enabling software-configurable optical networks," *Journal of Lightwave Technology*, vol. 41, no. 4, pp. 1175–1191, 2023.

[10] P. Soumplis *et al.*, "Techno-economic and feasibility study of point-to-multipoint communications in the metro-core," in *CSNDSP Conf.*, 2024.

[11] M. Radovic *et al.*, "Sdn-controlled open RAN x-haul with point-to-multipoint transceivers on a horseshoe network," *Journal of Optical Communications and Networking*, vol. 17, no. 11, pp. E109–E116, 2025.

[12] D. Scano, A. Giorgetti, F. Paolucci, A. Sgambelluri, J. Chammanara, J. Rothman, M. Al-Bado, E. Marx, S. Ahearne, and F. Cugini, "Enabling p4 network telemetry in edge micro data centers with kubernetes orchestration," *IEEE Access*, vol. 11, pp. 22 637–22 653, 2023.

[13] Z. Xu, Z. Lu, and Z. Zhu, "Information-sensitive in-band network telemetry in p4-based programmable data plane," *IEEE/ACM Transactions on Networking*, vol. 32, no. 6, pp. 5081–5096, 2024.

[14] F. Cugini *et al.*, "Telemetry and ai-based security p4 applications for optical networks [invited]," *Journal of Optical Communications and Networking*, vol. 15, no. 1, pp. A1–A10, 2023.

[15] J. Geng, J. Yan, Y. Ren, and Y. Zhang, "Design and implementation of network monitoring and scheduling architecture based on p4," in *Conf. on Computer Science and Application Engineering*, 2018.

[16] L. Ismail, A. Sgambelluri, F. Cugini, and F. Paolucci, "Programmable decentralized feature extraction telemetry," in *International Conference on Transparent Optical Networks (ICTON)*, 2025.