

ACCEPTED MANUSCRIPT

Bringing Cloud Computing to the Tactical Edge: Requirements, Technical Approach, and Findings

A. Amato, G. Bartolomeo, H. Bastiaansen, M. Fogli, T. Kudla and N. Suri

IEEE Communications Magazine, 2026

Cite this as

A. Amato, G. Bartolomeo, H. Bastiaansen, M. Fogli, T. Kudla and N. Suri, “Bringing Cloud Computing to the Tactical Edge: Requirements, Technical Approach, and Findings,” accepted for publication in *IEEE Communications Magazine*.

Notice

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Bringing Cloud Computing to the Tactical Edge: Requirements, Technical Approach, and Findings

Alessandro Amato, Giovanni Bartolomeo, Harrie Bastiaansen, Mattia Fogli, Thomas Kudla, Niranjan Suri

Abstract—Tactical units generate vast amounts of data that must be processed locally, as transferring them over denied, disrupted, intermittent, and limited (D-DIL) networks is often infeasible. However, edge computing in tactical environments faces severe challenges: resources are heterogeneous and transient, network links are unreliable, and trust among partners varies over time. Moreover, pooling resources across coalition partners is not straightforward, as each partner retains sovereignty over its own assets and enforces distinct resource-sharing policies. At the same time, individual partners often lack the compute capacity to run all mission-critical services independently. This tutorial-style article describes how these challenges were addressed by experimentally designing, implementing, and evaluating a federated, adaptive, and trust-aware system that brings cloud computing to the tactical edge. Results show that edge data processing reduces service response times, adaptive re-orchestration restores service continuity at negligible overhead, and heuristic schedulers achieve good schedule quality at a fraction of the computational cost of brute-force solvers.

Index Terms—adaptive orchestration, cloud computing, edge computing, federated service management, tactical networks, trust awareness

I. INTRODUCTION

Modern military operations generate unprecedented volumes of data from ubiquitous sensing assets on the battlefield. Artificial intelligence (AI) services can convert these data into actionable information, yet they typically execute either in enterprise clouds or in data centers at headquarters. Tactical units operating at the edge, however, cannot always reach headquarters, and even when they can, transferring large volumes of data over denied, disrupted, intermittent, and limited (D-DIL) network links is prohibitively slow—or altogether infeasible. Processing data closer to their source is therefore the natural alternative [1], [2], but the tactical edge introduces its own challenges.

No single partner controls all resources at the tactical edge—each nation retains sovereignty over its own assets. A centralized solution managed by one partner is therefore not a viable option. At the same time, operating in isolation is equally problematic: individual partners often lack the resources to execute all mission-critical services on their own. A federated approach—pooling resources across coalition partners while preserving each partner’s sovereignty—offers a way to bridge this gap. Yet the volatile nature of the tactical edge, where resources may go offline unpredictably, be destroyed by adversarial attacks, or not be trusted anymore, demands adaptive orchestration that continuously monitors conditions and adjusts deployment plans accordingly. Moreover, sharing resources across federates with different security postures

requires trust-aware mechanisms that account for the evolving trustworthiness of each partner. These challenges make current civilian cloud solutions largely impractical, as they rest on assumptions that rarely hold here: strong centralization, resource-rich environments, and good connectivity.

Prior work covers only parts of the problem. Lewis et al. [1] proposed tactical cloudlets to support computation offload at the tactical edge, though cloudlets operate independently, without federation across partners. Brannsten et al. [3] examined how federated mission networking extends into the tactical domain, addressing messaging, collaboration, service discovery, and security rather than the sharing of compute resources. Bastiaansen et al. [4] defined the interfaces to federate services and resources across sovereign partners in coalition tactical operations, leaving orchestration and trust to be built and evaluated. Surveys of civilian research report federation across providers as largely unstudied in fog orchestration [5], service placement as predominantly centralized [6], and trust between federated clouds as an open problem [7].

The North Atlantic Treaty Organization (NATO) Information Systems Technology (IST)-193 research task group (RTG) on *Edge Computing at the Tactical Edge* addressed these challenges by experimentally designing, implementing, and evaluating a federated, adaptive, and trust-aware system that brings cloud computing to the tactical edge. The system organizes tactical units into Tactical Edge Clouds (TECs)—clouds composed of one or more compute nodes deployed on vehicles, small aircraft, or carried by dismounted soldiers—and federates them across coalition partners (see Fig. 1).

We evaluated the system on an emulated tactical testbed to answer three questions. Can open-source Kubernetes distributions operate under D-DIL conditions (Q1)? Do the benefits of adaptive orchestration at the tactical edge outweigh the costs (Q2)? Can scheduling algorithms for adaptive orchestration in tactical networks (TNs) scale effectively (Q3)? The evaluation yielded the following findings. First, existing open-source container orchestration platforms can operate under D-DIL conditions, but with significant scalability constraints that make federation essential [8]. Second, adaptive orchestration substantially reduces service response times compared to centralized processing and rapidly restores service continuity when a TEC becomes unavailable, while the overhead introduced by the system services remains negligible [9]. Third, heuristic scheduling algorithms achieve good schedule quality at a fraction of the computational cost of brute-force solvers, making them practical for the continuous re-orchestration demands of TNs [10].

This tutorial-style article distills the work of the NATO IST-

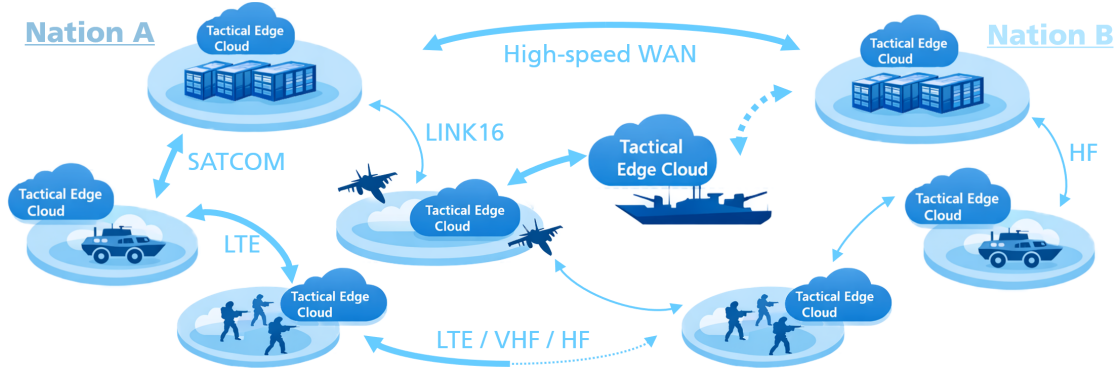


Fig. 1. High-level view of a federation of TECs.

193 RTG into a comprehensive view of how to bring cloud computing principles and technology solutions to the tactical edge.

II. CHALLENGES

TNs interconnect tactical units with highly *heterogeneous resources* [4]. At headquarters, abundant compute resources are available, while at the edge, tactical units—including vehicles, small aircraft, and dismounted soldiers—have resource-constrained compute capabilities. TECs, which may consist of one or more compute nodes, depending on various design decisions (e.g., operational domain, mission proximity), are interconnected using diverse communication technologies, such as satellite communications (SATCOM), Link 16, Long-Term Evolution (LTE), and various radio frequencies, such as very high frequency (VHF), UHF, and HF.

Severe network constraints create the first major challenge [11]. TNs operate as self-organizing wireless multi-hop infrastructures without dependence on fixed facilities. They suffer from *limited bandwidth*, *intermittent connectivity*, and *variable latency*. Network capacity typically emerges as the primary performance bottleneck.

The small size of individual TECs means they lack the compute capacity and network reach to support all mission-critical services independently, creating a need for *coalition resource pooling*. Without resource pooling across coalition partners, services might not achieve the geographic coverage or computational power required for effective execution.

The dynamic battlefield introduces *resource transiency* as another challenge. Tactical units may become unavailable due to network partitioning, adversarial action, or energy depletion. Fire-and-forget deployment decisions fail as conditions evolve, as the assumptions made at deployment time may no longer hold throughout the operation.

Service heterogeneity complicates service placement and orchestration at the edge [6]. Services range from low-latency, high-throughput tasks to periodic tasks with modest resource demands, and are described with high-level service level agreements (SLAs) that hide their internal behavior and resource usage patterns. This makes resource contention hard to predict. Services may also depend on one another, requiring co-location or specific communication patterns that further constrain placement.

Sovereignty concerns across coalition domains introduce *complex resource-sharing policies* [4]. Partners restrict which services may execute, impose CPU and memory limits, and permit only certain network operations. These policies vary across administrative domains and often conflict, leaving no feasible placement or forcing placements that degrade performance.

Trust variability among coalition partners presents the final challenge [7]. Partners have different trust levels that evolve throughout coalition operations. Partners may misuse shared resources through repeated service preemption, causing deployment delays and operational friction. Reputation and monitoring systems enable trust estimation but consume network and compute resources, requiring careful trade-offs between security assurance and operational overhead.

III. REQUIREMENTS

These challenges translate into the following requirements. Any system that brings cloud computing principles and technology solutions to the tactical edge must meet them, whatever design realizes it.

A. Federation

A federation of TECs consists of loosely coupled clouds, where each federated cloud retains sovereignty over its resources. Partners maintain control over their compute and network assets while participating in collective mission objectives. Unlike centralized solutions, which violate sovereignty requirements by placing control outside partner domains, federation allows each partner to enforce its own resource-sharing policies.

Network limitations (i.e., limited bandwidth, intermittent connectivity, and variable latency) among coalition partners require a distributed approach. Smaller TECs may not be able to support all mission-critical services independently. Federation enables resource pooling across coalition partners, extending both geographic coverage and computational power beyond what any single partner could achieve alone.

TECs must be able to locate and access services across partner clouds without comprehensive knowledge of the entire federation topology. This becomes critical when network partitions fragment the battlefield—tactical units must discover

available services within their reachable network segment. When deployment constraints conflict across administrative domains, the system requires protocols for identifying feasible compromises or, when no viable solution exists, for clearly indicating deployment infeasibility. The system must maintain awareness of resource and service availability through discovery mechanisms.

B. Adaptiveness

The challenge of resource transiency requires TECs to adapt continuously. Adaptiveness involves recognizing either a failure affecting a deployed workflow or a change in circumstances that results in suboptimal performance. The system must detect when the desired state does not match the actual state or when opportunities exist to improve performance. Adaptive orchestration is a continuous process ensuring that requested workflows—a set of services that interact with each other to accomplish a task—remain deployed optimally over time, where the definition of optimality varies with specific partner objectives and mission requirements.

Given limited and transient resources, adaptive orchestration must accurately estimate deployment costs before initiating redeployment. An efficient deployment considers factors often overlooked in traditional cloud environments. Deployment cost estimation must account for: the time the scheduling algorithm takes to determine a solution; the time required to transfer data (e.g., container images or workflow state) to enable deployment; the expected runtime of the workflow once deployed relative to platform availability; the time needed to configure partner clouds hosting the workflow services; and the activation time required by selected partner clouds to initiate scheduled workflows. These factors become critical when deciding whether redeployment benefits outweigh costs, particularly when bandwidth limitations make data transfer expensive.

C. Trust-awareness

Trust variability among coalition partners demands that orchestration incorporates trust considerations into resource sharing and workflow placement decisions. Without trust-aware mechanisms, the federation risks misusing shared resources, causing deployment failures and operational friction. For example, a compromised TEC may advertise resources it does not have, thus attracting deployment requests from the federation. These requests remain pending until re-orchestration places them elsewhere, delaying services that may be mission-critical. External circumstances shift trust too—tactical units entering contested territory face elevated adversarial compromise risk, threatening data confidentiality and service integrity. Trust-awareness therefore prevents sensitive services from executing on compromised or unreliable TECs.

This requires trust assessment mechanisms that estimate partner reliability from historical behavior, security posture, and current operational status. The system must aggregate these factors into trust metrics that guide placement decisions

without requiring complete information sharing. Comprehensive trust assessment provides better security guarantees but requires frequent communication between partner TECs to share behavior observations and security assessments. In bandwidth-constrained tactical environments, this communication competes with mission-critical traffic. The system must support configurable trust assessment granularity, allowing partners to adjust the trade-off between security assurance and resource consumption based on current operational requirements and threat levels.

D. Uniform Abstractions

A federation of TECs that is trust-aware and adaptive requires high-level abstractions that hide the complexity of the underlying infrastructure from both service developers and operators while letting each partner decide, through policies, which of its resources and services to expose and to which partners. Accordingly, the orchestration layer must provide standard interfaces that guarantee interoperability and connectivity transparently across the federation. Indeed, mission-critical services must be developed to ensure rapid response to emerging operational needs and must therefore focus on core logic while remaining agnostic to the specific configurations and capabilities of the infrastructure. Network configuration details, resource failures, service migrations, trust evaluations, and policy enforcement must be handled by the orchestration layer based on the high-level SLAs provided by developers and operators.

IV. TECHNICAL APPROACH

The system described here is one realization of these requirements, showing how each can be met in practice. Design and implementation are based on prior work by the NATO IST-168 and IST-193 RTGs [4], [8], [9], [10].

A. System Overview

We relied on Kubernetes, the de facto industry standard for cloud container orchestration, to manage services within a TEC. Kubernetes is available in many distributions, each with its own target use case. For example, Kubernetes Native Edge Computing Framework (KubeEdge) is a Kubernetes distribution for edge environments [12].

As the Kubernetes application programming interface (API) supports Custom Resource Definitions (CRDs), which enable developers to extend Kubernetes with custom abstractions, we defined our own to represent resources, services, and workflows at the federation level. For instance, we implemented the abstraction of a federated workflow—a set of services that may be deployed across the federation along with their constraints (e.g., compute, networking, and trust requirements) and dependencies (e.g., how services communicate with each other). This set of abstractions can also work with any other underlying Kubernetes distribution (e.g., KubeEdge) as long as it is compatible with the official Kubernetes API. This gives coalition partners freedom to opt for different Kubernetes distributions depending on their requirements or to develop custom distributions from existing projects.

In cloud environments, service images are typically pulled from remote registries over the Internet. However, TECs operating on the battlefield cannot reach the Internet or often even headquarters. Each TEC therefore maintains a local service registry that stores service images and metadata. This ensures that locally available services are always executable, and lets them be transferred peer-to-peer between TECs, subject to policy and resource constraints. Once services are deployed, secure inter-TEC communication is provided through an application-layer service interconnect. This interconnect creates secure service networks across environments without requiring virtual private networks (VPNs) or public service exposure.

Lastly, we designed and developed four system services to create a federation of TECs, enable adaptive orchestration of workflows across the federation, and ensure trust during system operation. These services must run in each TEC to address the identified requirements. However, each coalition partner can use its own implementations as long as they adhere to the defined CRDs and APIs, thus guaranteeing sovereignty over infrastructure management, data sharing, and service execution. These system services are (see Fig. 2):

- The Federated Resource Discovery (FRD), which exchanges information across the federation and stores it locally as Custom Resources (CRs).
- The Federated Service Deployment (FSD), which deploys services either locally or on other TECs and appropriately interconnects such services once deployed.
- The Federated Adaptive Orchestrator (FAO), which creates deployment plans across the federation and adaptively adjusts these plans as conditions change based on the policies in place.
- The Trust Resource Management (TRM), which computes trust metrics based on monitored activities and updates policies for the other system services accordingly.

Within a TEC, the system services do not communicate directly but coordinate by creating, reading, updating, and deleting CRs through the Kubernetes API; across the federation, FRD and FSD instances communicate peer-to-peer.

B. System Operation

A federation requires each federate to know the others and how much each contributes in resources and services. The FRD enables this. Specifically, FRD instances communicate with each other to gather, disseminate, and exchange up-to-date information about: services currently running (deployed as containerized applications through Kubernetes) and services available for deployment (stored as container images in the local registry); and compute (CPUs, graphics processing units (GPUs), memory, storage, etc.) and networking resources (pairs of links connecting federated TECs). Another piece of information required for the federation is the extent to which a federate trusts the others. Each TEC runs its own TRM, which independently computes the trust toward the other federates.

Through our abstractions, a coalition partner can request a federated workflow from their own local TEC via the extended Kubernetes API. The federated workflow description consists

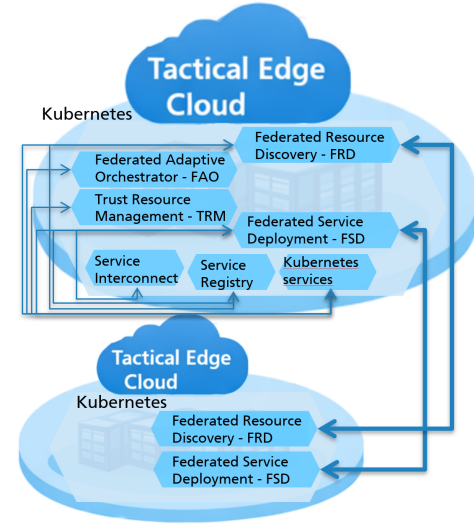


Fig. 2. Components of an individual TEC and APIs for federated interactions. Thin and thick arrows denote intra-TEC and inter-TEC interactions, respectively.

of key-value pairs that specify which services are required, how these services communicate with each other, what the requirements for these services are, and any preferences, such as running a service on a specific partner or using a service instance already running. Based on this description, the FAO creates a deployment plan that indicates where to schedule each service and in which TEC, considering the available resources, service locations, and the policies in place. The deployment plan captures the desired state of the workflow, and is marked infeasible when no assignment satisfies all constraints. The FAO autonomously updates the plan as conditions evolve, without manual intervention, while the operator may amend the workflow description to facilitate a solution. Each update is either a first deployment—when the plan was just created or a previous attempt was infeasible—or a re-orchestration—when deployed services no longer meet their requirements or the optimization objective has changed. Among the feasible plans, the FAO selects an optimal one according to a configurable criterion, such as maximizing trust. The FSD executes this deployment plan by deploying services locally or delegating deployment to a remote FSD instance. If a required service is missing on the target TEC, the local FSD transfers it to the remote instance before deployment. If services are deployed across different TECs, the FSD configures the service interconnect accordingly.

The FRD continuously updates information about resources and services across the federation. This information enables the FAO to react promptly when required and update its deployment plans based on current circumstances. For example, suppose a service was scheduled on a TEC that is now operating in enemy territory. This may cause the TRM to lower the trust level of that TEC, which is no longer considered trustworthy for hosting services. This information update triggers the FAO to change the deployment plan and, in turn, the FSD to redeploy the service elsewhere. Overall, adaptivity is achieved through a continuous loop of monitoring, replanning,

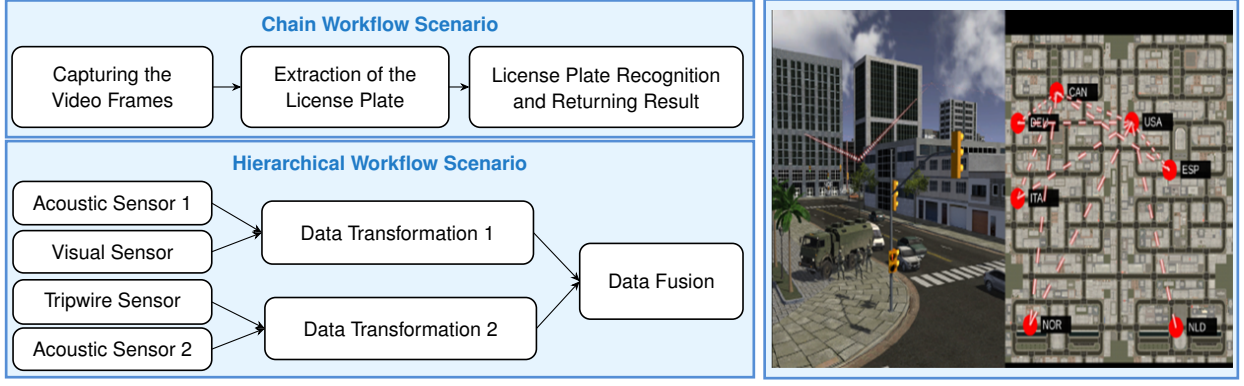


Fig. 3. Chain workflow (top left), hierarchical workflow (bottom left), and a federation of TECs in an urban environment simulated in Unity (right).

and execution across the federation.

V. FINDINGS

The system was experimentally evaluated to answer the three questions raised in Section I. These questions operate at two different levels. Q1 asks whether Kubernetes can be used to build a TEC and, if so, at what scale, that is, how many nodes a single TEC can host under D-DIL conditions. Q2 and Q3 instead concern the federation level: once a federation of TECs is available, whether adaptive orchestration is effective (Q2) and whether it scales (Q3). This section describes the representative scenarios to evaluate the proposed technical approach, the testbed used for the evaluation, and the performance tests to answer such questions.

A. Representative Scenarios

We developed two representative scenarios emulating realistic joint NATO military operations in an urban area. These scenarios showcase the system capabilities under D-DIL conditions for two different workflows (see Fig. 3).

The first scenario is a *chain workflow*. A dismounted soldier captures video of a suspicious vehicle, and the local TEC lacks the resources to recognize its license plate. The workflow chains three tasks: splitting the video into frames, extracting the plate region, and recognizing the plate.

The second scenario is a *hierarchical workflow* for surveillance. Acoustic, visual, and tripwire sensors detect and localize drones across federated TECs. Sensor transformation services first standardize the heterogeneous data, which a fusion service then integrates into a common operational picture.

B. Testbed

We built a testbed consisting of multiple TECs interconnected into a federation through the system services using tactical radio links emulated by the Extendable Mobile Ad-hoc Network Emulator (EMANE) [13]. The testbed is integrated with a simulation environment based on the Unity game engine, which serves two purposes: generating camera and sensor feeds and controlling D-DIL network conditions. Changes in the Unity simulation are automatically reflected in the testbed by reconfiguring EMANE to match the simulated environment,

thereby provoking measurable system responses. Combining physical hardware and emulated components, the testbed enables controlled and repeatable experiments that mirror the D-DIL conditions of the tactical edge. In the experiments reported here, network conditions were set directly.

C. Kubernetes Performance in TNs under D-DIL Conditions

Experimental evidence [8] demonstrated that Kubernetes can operate under D-DIL conditions, but only under specific circumstances. Feasibility depends on the combination of cloud size (number of participating nodes within one cloud), cloud workload (number of running applications), network parameters (bandwidth, packet loss, and latency), and the distribution itself. The experiments evaluated Kubernetes, Lightweight Kubernetes (K3s), and KubeEdge in clouds of up to 24 nodes, with each worker node running up to 20 application replicas. At 4 Mb/s shared among nodes, 200 ms latency, and 5% packet loss per link, Kubernetes and K3s sustained TECs of up to 12 nodes and KubeEdge up to 18. At 2 Mb/s, 400 ms, and 10%, these limits dropped to 6 and 12 nodes, respectively. At 0.65 Mb/s, 3 s, and 15%, all distributions failed.

KubeEdge tolerated D-DIL conditions best because of its cloud-edge architecture: a standard Kubernetes cluster acts as the cloud side, while edge nodes connect to it through a lightweight, User Datagram Protocol (UDP)-based QUIC channel that tolerates disconnections and synchronizes state when connectivity permits. In a TEC, the Kubernetes cluster serves as the cloud kernel, clustering co-located nodes that share adequate connectivity—resources within a single vehicle, aircraft, or ship. This kernel maintains service and resource availability over time through scaling, transparent migration, and load balancing. Dynamic tactical elements, such as mobile dismounted units, then join the kernel as edge nodes. Although this architecture is promising for individual TECs, it cannot scale for battlefield-wide coverage.

Q1. Can open-source Kubernetes distributions operate under D-DIL conditions? Existing Kubernetes distributions may work in TNs but with severe scalability limitations due to the D-DIL conditions. In practice, Kubernetes is suitable as the orchestration layer only within a single TEC, whose nodes should be chosen strategically, for example, nodes likely to

share sufficient connectivity throughout a mission. Achieving adequate operational coverage therefore requires a federation of TECs.

D. Adaptive Orchestration at the Tactical Edge

We analyzed the benefits and costs of adaptive orchestration at the tactical edge in the chain workflow scenario (see Fig. 3). As the soldier's own TEC does not have sufficient resources to run the license plate recognition service, the system services coordinate to deploy the workflow on a sufficiently trusted federated TEC with available resources. Then, the car is flagged as suspicious and requested to be tracked city-wide using cameras mounted on other partners' vehicles, generating multiple concurrent requests that stress the federation and require adaptive orchestration when TECs become unavailable due to adversarial activity or loss of connectivity.

Experimental results [9] demonstrate that processing data at the tactical edge (three federated TECs interconnected via 1 Mb/s links, with 256 Kb/s links to headquarters) rather than at a centralized headquarters reduced service response times by up to 58%, due to the higher bandwidth available on edge-to-edge links compared to edge-to-headquarters satellite links. The gain depends on how much data must cross the link: three concurrent requests on a 13 MB video took 1072 s at headquarters and 443 s at the edge, whereas the same requests on a 0.5 MB video improved by 7%. When a TEC became unavailable, the FAO re-allocated the affected service to an alternative TEC, updating its deployment plan in 661 ms on average (99.9th percentile: 2.46 s) and restoring continuity without manual intervention. The overhead to sustain this adaptivity was low: inter-cloud bandwidth consumed by FRD remained in the order of bytes per second (~ 247 bytes/s in and ~ 22 bytes/s out per instance), with network sensing requiring approximately 1.5 KB/s per neighbor link; and computation and memory overhead were modest (combined: ~ 280 milliCPU and ~ 467 MB across FRD, FSD, and FAO). Inter-cloud bandwidth scales linearly with the number of TECs and the resource discovery update frequency, and can be further reduced through delta-based update strategies.

Q2. Do the benefits of adaptive orchestration at the tactical edge outweigh the costs? *Adaptive orchestration significantly improves service response times and ensures continuity when TECs become unavailable, while the bandwidth, computation, and memory overhead introduced by the federation services remains negligible relative to the operational gains.*

E. Workflow Scheduling Algorithms for TNs

Adaptive orchestration requires continuously making decisions on where to deploy services across the federation of TECs. This problem is NP-hard. Brute-force approaches based on constraint solvers can find near-optimal solutions, but their computational cost grows significantly with problem size, becoming impractical as the problem scales. The FAO is algorithm-agnostic, so it can be configured to work with different types of scheduling algorithms. We integrated Scheduling Algorithms Gathered (SAGA) [14], a framework

providing a family of heuristic schedulers, and conducted a systematic evaluation of 120 configurations—composed from different priority functions, comparison functions, insertion strategies, critical path reservation, and sufferage options—across the two representative workflows. The evaluation used a network model of seven fully connected TECs with links of 4, 2, and 0.65 Mb/s, scheduling one to five concurrent workflows under a one-minute timeout.

Results [10] demonstrated that the brute-force solver runtime grew rapidly with problem size, timing out at the one-minute threshold when scheduling as few as three hierarchical workflows for throughput optimization, whereas heuristic schedulers, such as Heterogeneous Earliest Finish Time (HEFT) [15], completed in negligible time regardless of the scale. Among the algorithmic components, the comparison function—which determines the node assignment for each task—proved to be the dominant factor influencing schedule quality. For makespan minimization (total time required to complete a set of tasks from start to finish), the Earliest Finish Time (EFT) and Earliest Start Time (EST) comparison functions consistently outperformed alternatives across both workflows, particularly when combined with no critical path reservation. For throughput maximization, the Bottleneck comparison function emerged as the most effective, again benefiting from disabling critical path reservation. Other components, including the priority function, insertion-versus-append strategy, and sufferage, had minimal impact on schedule quality.

Q3. Can scheduling algorithms for adaptive orchestration in TNs scale effectively? *Brute-force solvers produce near-optimal plans but fail to scale beyond a handful of concurrent workflows. Heuristic-based schedulers, particularly those leveraging EFT and EST for makespan minimization and Bottleneck for throughput maximization, achieve good schedule quality with negligible runtime, making them practical for the continuous re-orchestration demands of TNs.*

F. Lessons Learned

Section V-C shows that a single large cloud is not viable at the tactical edge. Under D-DIL conditions, node counts are sharply limited: at 2 Mb/s shared among nodes, 400 ms latency, and 10% packet loss, KubeEdge sustained 12 nodes and Kubernetes and K3s only 6. Coverage therefore comes from federation, not from enlarging a single cloud. Section V-D shows when the tactical edge pays off. Processing data locally reduces service response times when edge-to-edge links offer better connectivity than the backhaul to headquarters, by up to 58% when tracking a vehicle city-wide, where video must cross the network. Adaptive orchestration improves continuity when failures occur, re-planning in under a second on average and sustained by about 1.5 KB/s per neighbor link. Section V-E shows that heuristic schedulers keep orchestration decisions practical as the federation scales. Brute-force solvers time out at three concurrent hierarchical workflows across seven TECs when maximizing throughput.

These lessons rest on what we evaluated: two workflow types (chain and hierarchical), clouds of 6 to 24 nodes, federations of three to seven TECs, and, for the cloud-size

limits, network conditions ranging from 4 Mb/s, 200 ms, and 5% packet loss to 0.65 Mb/s, 3 s, and 15% packet loss. We chose these parameters to be as realistic as we could. They also rest on the technology available at the time, such as container runtimes and orchestrators. Better technology would move the thresholds, though not the trade-offs behind them.

VI. CONCLUSION AND FUTURE WORK

This work shows an approach to bringing cloud computing principles and technology solutions to the tactical edge: a federation of adaptive, trust-aware TECs that operates under D-DIL conditions, reducing service response times and restoring service continuity when TECs become unavailable. Although designed for TNs, this approach can be transferred to civilian scenarios with similar requirements and challenges, such as humanitarian assistance and disaster relief, smart cities, and industrial environments.

Future work will be conducted by the IST-240 RTG, *Tactical Adaptivity Concept Convergence for Meshed Edge Service and Data Handling*, and will pursue four directions. First, migrating service execution state. Services may be stateful, or stateless with long-running invocations, so restarting them elsewhere discards context that cannot be rebuilt, such as state bound to a live sensor stream, or costs more time than the mission can afford. Second, integrating tactical data meshes to optimize data availability across TECs for local processing. Service and data orchestration are mutually dependent, as a placement decision determines which container images, execution state, and consumed data must be moved, and the cost of moving them constrains the decision itself. Third, optimizing the network, data, and service orchestration layers jointly rather than in isolation. Fourth, coordinating orchestration at the federation level to balance mission objectives with individual partners' requests.

ACKNOWLEDGMENT

The work presented in this article was conducted as part of the NATO IST-193 RTG *Edge Computing at the Tactical Edge*, which built and extended upon its predecessor NATO IST-168 RTG on *Adaptive Information Processing and Distribution to Support Command and Control*. We thank the NATO IST-Panel for the opportunity to conduct this research and the participating partners for their valuable contributions in a stimulating and cooperative environment.

REFERENCES

- [1] G. Lewis, S. Echeverría, S. Simanta, B. Bradshaw, and J. Root, "Tactical cloudlets: Moving cloud computing to the edge," in *2014 IEEE Military Communications Conference*, 2014, pp. 1440–1446.
- [2] M. Satyanarayanan, R. Schuster, M. Ebling, G. Fettweis, H. Flinck, K. Joshi, and K. Sabnani, "An open ecosystem for mobile-cloud convergence," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 63–70, 2015.
- [3] M. R. Brannsten, F. T. Johnsen, T. H. Bloebaum, and K. Lund, "Toward federated mission networking in the tactical domain," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 52–58, 2015.
- [4] H. Bastiaansen, J. van der Geest, C. van den Broek, T. Kudla, A. Isenor, S. Webb, N. Suri, M. Fogli, B. Canessa, A. Masini, R. Goniacz, and J. Sliwa, "Federated control of distributed multi-partner cloud resources for adaptive C2 in disadvantaged networks," *IEEE Communications Magazine*, vol. 58, no. 8, pp. 21–27, 2020.

- [5] B. Costa, J. Bachiega, L. R. de Carvalho, and A. P. F. Araujo, "Orchestration in fog computing: A comprehensive survey," *ACM Comput. Surv.*, vol. 55, no. 2, Art. no. 29, Jan. 2022.
- [6] F. A. Salaht, F. Desprez, and A. Lebre, "An overview of service placement problem in fog and edge computing," *ACM Comput. Surv.*, vol. 53, no. 3, Art. no. 65, Jun. 2020.
- [7] U. Ahmed, I. Raza, and S. A. Hussain, "Trust evaluation in cross-cloud federation: Survey and requirement analysis," *ACM Comput. Surv.*, vol. 52, no. 1, Art. no. 19, Feb. 2019.
- [8] T. Kudla, M. Fogli, S. Webb, G. Pinget, N. Suri, and H. Bastiaansen, "Quantifying the performance of cloud-oriented container orchestrators on emulated tactical networks," *IEEE Communications Magazine*, vol. 60, no. 5, pp. 74–80, 2022.
- [9] A. Amato, H. Bastiaansen, W. Datema, M. Fogli, R. Fronteddu, R. Galliera, J. van der Geest, T. Kudla, P. Sanchez, N. Suri, and S. Watson, "A performance cost/benefit analysis of adaptive computing in the tactical edge," in *2024 International Conference on Military Communication and Information Systems (ICMCIS)*, 2024, pp. 1–8.
- [10] A. Amato, H. Bastiaansen, J. Coleman, W. Datema, M. Fogli, J. van der Geest, B. Krishnamachari, T. Kudla, P. Sanchez, and N. Suri, "Evaluating scheduling algorithms for adaptive orchestration in federated tactical edge cloud environments," in *2025 International Conference on Military Communication and Information Systems (ICMCIS)*, 2025, pp. 1–10.
- [11] R. Amin, D. Ripplinger, D. Mehta, and B.-N. Cheng, "Design considerations in applying disruption tolerant networking to tactical edge networks," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 32–38, 2015.
- [12] Y. Xiong, Y. Sun, L. Xing, and Y. Huang, "Extend cloud to edge with KubeEdge," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018, pp. 373–377.
- [13] EMANE: Distributed wireless network emulation framework. Accessed: 2026-02-07. [Online]. Available: <https://emane.io>
- [14] SAGA: Scheduling Algorithms Gathered - collection of task graph scheduling algorithms for dispersed / distributed computing. Accessed: 2026-02-07. [Online]. Available: <https://github.com/ANRGUSC/saga>
- [15] H. Topcuoglu, S. Hariri, and M.-Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, 2002.

BIOGRAPHIES

Alessandro Amato (aamato@ihmc.org) is currently pursuing a Ph.D. degree at the Florida Institute for Human and Machine Cognition (IHMC), Pensacola, FL, USA.

Giovanni Bartolomeo (giovanni.bartolomeo@tum.de) is currently pursuing a Ph.D. degree at the Technical University of Munich, Munich, Germany.

Harrie Bastiaansen (harrie.bastiaansen@tno.nl) is a senior scientist at the Netherlands Organisation for Applied Scientific Research (TNO) and an endowed professor at the University of Groningen, Groningen, The Netherlands.

Mattia Fogli (mattia.fogli@unife.it) is an assistant professor at the Department of Mathematics and Computer Science, University of Ferrara, Ferrara, Italy.

Thomas Kudla (thomas.kudla@fkie.fraunhofer.de) is a researcher and consultant at the Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE), Wachtberg, Germany.

Niranjani Suri (niranjani.suri.civ@army.mil, nsuri@ihmc.org) is the Associate for Research in the Military Information Sciences Division at the US Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL), Adelphi, MD, USA, and a senior research scientist at the Florida Institute for Human and Machine Cognition (IHMC), Pensacola, FL, USA.