Towards a COTS-Enabled Federated Cloud Architecture for Adaptive C2 in Coalition Tactical Operations: A Performance Analysis of Kubernetes

M. Fogli, G. Pingen, T. Kudla, S. Webb, N. Suri and H. Bastiaansen

MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM), 2021

Cite this as

M. Fogli, G. Pingen, T. Kudla, S. Webb, N. Suri and H. Bastiaansen, "Towards a COTS-Enabled Federated Cloud Architecture for Adaptive C2 in Coalition Tactical Operations: A Performance Analysis of Kubernetes," *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, San Diego, CA, USA, 2021, pp. 225-230, doi: 10.1109/MILCOM52596.2021.9653050.

Towards a COTS-Enabled Federated Cloud Architecture for Adaptive C2 in Coalition Tactical Operations: A Performance Analysis of Kubernetes

Mattia Fogli*, Geert Pingen†, Thomas Kudla‡, Sean Webb§, Niranjan Suri¶‖, Harrie Bastiaansen‡
* *University of Ferrara*, Ferrara, Italy
mattia.fogli@unife.it
† *The Netherlands Organisation for Applied Scientific Research (TNO)*, The Hague, The Netherlands
{geert.pingen, harrie.bastiaansen}@tno.nl
‡ *Fraunhofer Institute for Communication, Information Processing and Ergonomics*, Wachtberg, Germany
thomas.kudla@fkie.fraunhofer.de
§ *Defence Research and Development Canada*, Dartmouth, Canada
sean.webb3@ecn.forces.gc.ca
¶ *Florida Institute for Human and Machine Cognition (IHMC)*, Pensacola, FL, USA
nsuri@ihmc.org
‖ *US Army Research Laboratory (ARL)*, Adelphi, MD, USA
niranjan.suri.civ@army.mil

***Abstract*—Nowadays, ever-increasing processing and storage resources are available at all echelons, from operations centers to tactical units. However, tactical-edge communications still suffer from scarce network resources such as limited bandwidth, intermittent connectivity, and variable latency. In addition, modern military missions typically involve coalition operations, where heterogeneous mission partners (even belonging to different nations) cooperate in the field. As a result, the distribution of mission-critical information is more complicated than ever. On the one hand, the dynamic nature of the tactical environment frequently disrupts communications. On the other hand, individual resource-sharing policies prevent mission partners from taking full advantage of the available resources in situ. The NATO IST-168 RTG has been exploring commercial-off-the-shelf orchestration technologies for implementing a federated cloud architecture that enables adaptive information processing and dissemination while living within the constraints of the tactical domain. This paper is a follow-up study that assesses the behaviour of Kubernetes under the disadvantaged network conditions characterizing tactical edge networks.**

***Index Terms*—Adaptive C2, Tactical Edge Networks, Coalition Tactical Operations, Federated Cloud Architecture, Commercial-Off-The-Shelf, Kubernetes**

I. Introduction

Coalition tactical operations typically involve several operational domains, potentially commanded by distinct Tactical Operations Centers (TOCs) belonging to different nations [1]. As a result, the battlefield comprises a plethora of heterogeneous assets needing to communicate mission-critical information to each other in a real-time fashion in order to enable Situation Awareness (SA) dissemination [2], which is crucial to enhance decision-making. Such information distribution should adhere to the Need-To-Know (NTK) principle [3] that allows sharing of sensitive information directly at the tactical edge, without having to resort to the traditional path of going up one command hierarchy before coming down a second hierarchy.

Another crucial aspect of coalition tactical operations concerns the dissimilarity of computational capabilities (i.e., network, storage, and processing [4]) between the decision-making level (i.e., TOCs) and the tactical-operational level (i.e., mobile dismounted units deployed on the tactical field). In fact, the former benefits from abundant resources, whereas the latter cannot take resources for granted. At the tactical-operational level, Tactical Edge Networks (TENs) interconnect mobile units in self-organizing wireless multi-hop networks, enabling communications on the battlefield without relying on pre-existing network infrastructures. Such interconnected units differ highly in terms of computational capabilities, ranging from resource-rich aircraft, battleships, and vehicles to resource-constrained dismounted soldiers and Unmanned Aerial Vehicles (UAVs) [5]. In addition, the dynamic nature of the tactical environment heavily affects communications, which suffer from limited bandwidth, variable latency, and intermittent connectivity.

The challenge is to provide SA (i.e., mission-critical information to all echelons); information integrity and confidentiality (i.e., sensitive-information protection from being compromised by adversarial attacks); and optimal resource utilization [6]. A federated cloud architecture, where a cloud consists of processing and storage resources made available on-demand, can enable adaptive information processing and dissemination while coping with the ever-changing network conditions [7]. Such a federated strategy ensures sovereignty over resources to mission partners, which maintain fine-grained control over the computational resources shared with others. Moreover, this

cloud-based approach allows mission partners to cluster such resources according to the optimal tradeoff between ownership and network availability, where clouds mutually expose internal resources to each other through well-defined Application Programming Interfaces (APIs). As a result, tactical units can adaptively schedule information and processing tasks on partner clouds based on network, processing, and storage available in situ. Furthermore, a federated cloud architecture can adhere to the NTK principle and still improve the overall distribution of information at all echelons by processing raw data with coalition partner analysis tools locally and just providing the results. Such an approach enables compliance with desired access control policies, security measures, and data protection mechanisms.

This approach to using a federated cloud architecture to move processing to data (i.e., deploying data processing services close to the source of raw data) also provides significant benefits for TENs. Raw data does not need to be transmitted over already congested communications links. Further, processed data can be locally disseminated over communications links that are likely to be more robust than reachback links to TOCs.

A potential pillar for enabling intra-cloud capabilities is Kubernetes (K8s) [8], an open-source platform that provides orchestration mechanisms for deploying, maintaining, and scaling containerized applications across multiple hosts. Currently, K8s is a de facto industry standard. However, cloud orchestration technologies usually target enterprise environments, taking for granted stability and resources that might not be available in TENs. Nowadays, various K8s-based solutions are available, even some specifically designed for resource-constrained scenarios, such as Lightweight Kubernetes (K3s) [9] and Kubernetes Native Edge Computing Framework (KubeEdge) [10]. In our previous work [11], we assessed KubeEdge's superiority while performing in disadvantaged network conditions over K8s and K3s. However, KubeEdge needs a K8s-based cluster already up and running as a prerequisite. We believe that the synergy between K8s and KubeEdge could lay a solid foundation for enabling the intra-cloud capabilities of the individual clouds jointly constituting a federated cloud architecture for adaptive Command and Control (C2) in coalition tactical operations. Given the dependence of the potential solution on K8s, this follow-up study extends upon our previous work by presenting an in-depth analysis of the performance and constraints of K8s in TENs.

The remainder of the paper has the following structure. First, Section II outlines ambitions, objectives, and previous work of North Atlantic Treaty Organization (NATO) IST-168 Research Task Group (RTG). Subsequently, Section III details testbed setup and test framework. Next, Section IV describes the experiments for performance assessment. Then, Section V presents the experimental evaluation results. Finally, Section VI provides the conclusion and future work.

II. The Work of NATO IST-168 RTG

The NATO IST-168 RTG's overarching ambition is to develop a federated cloud architecture based on state-of-the-art Commercial-Off-The-Shelf (COTS) standards and technologies to enable adaptive C2 in coalition tactical operations.

A. Land and Maritime Scenarios

In [12], we describe illustrative and representative scenarios showcasing the benefits of such an architecture in accomplishing real-world mission-oriented operational tasks within both land and maritime environments. The land scenario describes an urban operation [13] that takes place in the fictitious country of Anglova [14], where military and civilian actors participate and cooperate by carrying out activities that range from enemy engagement to medical evacuation to Improvised Explosive Device (IED) neutralization. The maritime scenario involves a coordinated Anti-Submarine Warfare (ASW) activity, which takes place in a narrow strait (connecting the open ocean and an inland sea). Specifically, it consists of multi-nation surface platforms supported by a shore site to detect and classify submarines passing through the strait. In addition, fishing activities, civilian ferries, and commercial air traffic occur during the operation.

The land and maritime scenarios depict coalition tactical operations, where multiple mission partners need to cooperate, although they belong to different operational domains. It is worth remarking that mission partners might belong to different nations and include both military (e.g., Army, Navy, and Air Force) and civilian (e.g., local police, firefighters, medical rescuers, and city officials) entities, which potentially deploy terrestrial as well as aerial assets.

B. Architectural Principles and Artifacts

In [7], we define architectural principles and artifacts to implement a federated cloud architecture for adaptive C2 in coalition tactical operations. As previously described, coalition tactical operations involve various mission partners, which might highly differ in access control policies, security measures, and data protection mechanisms. On the one hand, each mission partner demands sovereignty over its resources. On the other hand, resource- and communication-constrained nodes at the tactical edge (e.g., dismounted soldiers and underwater platforms) might lack adequate resources during the operation. Accordingly, the primary objective of such an architecture is to enable adaptive information processing and dissemination based on a federation strategy, ensuring fine-grained resource control to mission partners while promoting resource sharing. Specifically, mission partners cluster resources in clouds, consisting of network, processing, and storage resources made available on-demand.

Such clouds mutually expose internal resources to each other through well-defined APIs. Specifically, the Service Discovery API provides information about invocable and deployable services (i.e., software capabilities of a cloud). Next, the Resource Discovery API provides information about allocable resources (i.e., hardware capabilities of a cloud),

which consist of processing, storage, and network availability towards partner clouds. Lastly, the Service Management API serves incoming requests concerning service deployment and resource allocation. The inter-cloud communications' standardization allows interoperability between partner clouds without affecting sovereignty. In fact, each mission partner can inject its own policies regarding what partners can see (information about services and resources) and do (ability to invoke/deploy services and take resources).

C. COTS-based Implementation

Given that each mission partner might adopt heterogeneous hardware/software equipment, vendor-agnostic approaches and virtualization techniques are essential in enabling intra-cloud capabilities, which range from resource orchestration to application scheduling and management. In this regard, COTS technologies (i.e., hardware/software ready-made and available for the general public) offer several advantages, such as the benefits of large-scale economies: lower development costs and the usage of widely deployed solutions. However, COTS technologies targeting enterprises typically take for granted resources that might not be available in TENs. In [11], we explore open source orchestration COTS technologies for deploying, maintaining, and scaling containerized applications in a cloud environment. Specifically, we assess the performance of K8s, K3s, and KubeEdge while performing in disadvantaged network conditions. Such a study reveals that KubeEdge, which relies on the QUIC protocol for communications, is more resilient to disadvantaged network conditions than K8s and K3s, which make use of the Transport Control Protocol (TCP). However, KubeEdge is not a solution per se but is ancillary to K8s. In fact, KubeEdge works on top of K8s, demanding as a prerequisite a K8s-based cluster up and running (even only comprising a single node acting as the master). Therefore, it is crucial to further assess the performance and clearly detail the operational working range (i.e., worst conditions allowing to work successfully) of K8s, which is a fundamental building block of the aforementioned federated and adaptive cloud architecture.

TABLE I: Network scenarios

	Bandwidth (Mbps)	Latency (s)	Packet Loss (%)
LAN	No constraints	No delay	No loss
BEST	4	0.2	5
AVG	2	0.4	10
WORST	0.65	3	15

TABLE II: Maximum available bandwidth per CN

	BEST (Kbps)	AVG (Kbps)	WORST (Kbps)
6 (CNs)	667	333	108
12	333	167	54
18	222	111	36
24	167	83	27

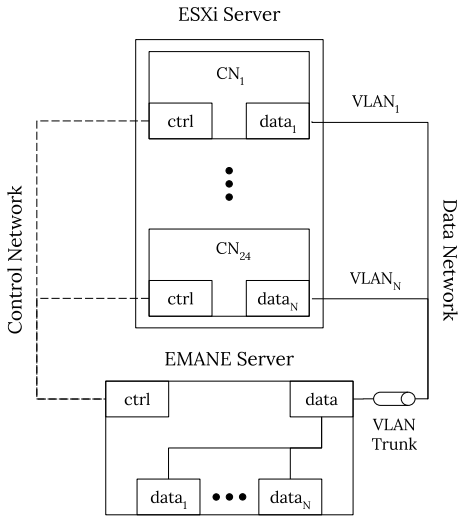


Fig. 1: Testbed setup.

III. Experimental Setup

A. Testbed Configuration

As shown in Fig. 1, the experimentation testbed comprises two physical servers and two networks. Above, a Dell PowerEdge R630 equipped with two Intel(R) Xeon(R) E5-2680 v3 CPUs and 192 GB of RAM runs VMWare vSphere Hypervisor (ESXi) 6.7. The ESXi server hosts twenty-four Virtual Machines (VMs) overall, each equipped with two CPUs, four GB of RAM, and two network interfaces. Henceforth, we refer to such VMs as Cluster Nodes (CNs). Below, a Dell PowerEdge R620 runs Extendable Mobile Ad-hoc Network Emulator (EMANE) [15], i.e., an open-source framework allowing to model link and physical layer connectivity in a real-time fashion. On the right side, the data network (continuous line) conveys the whole traffic exchanged by CNs to accomplish K8s-related tasks. In particular, the data network consists of a virtual switch within the ESXi server to which CNs connect their data interfaces. The EMANE server also connects its data interface to that switch through a VLAN trunk (IEEE 802.1Q). Note that each CN's data interface has a counterpart within the EMANE server. As a CN sends out network packets via its data interface, such packets are transparently transferred to the corresponding interface within the EMANE server. In turn, EMANE takes those packets, puts communication effects into action, and then delivers them to the appropriate interface, according to the actual destination. On the left side, the control network (dashed line) allows the test framework (see Section III-B) to manage the testbed bypassing the data network, whose traffic flows through EMANE as described before.

B. Test Framework

A test framework consisting of a collection of software modules has been developed to automate the execution of the experiments and perform analysis of raw network data in a consistent and reproducible manner. In this regard, we

make use of Ansible [16], which is a well-known configuration management tool that allows a control node to inject configurations into a set of target nodes over SSH. In general, we take advantage of Ansible for accomplishing all the tasks concerning the experiments, such as un/installing packages, providing system configurations, initiating cluster initialization and application deployment, setting up communication effects (EMANE), capturing network traffic (TCPDUMP), and fetching PCAP files from CNs. Then, we make use of shell scripts to process such files and R [17] scripts to extract, manipulate, and plot the data from different perspectives.

IV. Experiments for Performance Assessment

The performance assessment of K8s-based clusters comprises two classes of experiments, i.e. Cluster Initialization (CI) (see Section IV-A) and Application Deployment (AD) (see Section IV-B).

Each experiment consists of a setup phase, where the test framework configures the testbed to meet the preconditions of the target experiment; a bootstrap phase, where the test framework declares the cluster's desired state; and a steady-state phase, where the cluster's current state matches the desired state. It is worth mentioning that once a cluster's desired state is declared, K8s will continuously try to meet it. Therefore, we define an arbitrary (but reasonable) timeout threshold to distinguish between failed and succeeded experiments. Specifically, if the bootstrap phase takes longer than five minutes, we mark that experiment as a failure; otherwise, it is marked as a success. Given that particular simulation settings might result in experiments failing or succeeding inconsistently, we repeat each experiment three times to better support our findings.

A. Cluster Initialization (CI)

CI experiments assume that software packages and system configurations for enabling K8s are already on CNs (setup phase). Such a precondition is logical, especially considering mission military environments, where it would be unreasonable (or even unfeasible) to download those packages at the tactical edge. Then, the test framework initializes one CN as the master and the others as workers, which try to join the master simultaneously (bootstrap phase). The cluster's current state matches the desired state when the joining phase completes successfully, i.e., the master marks all the workers as "ready" (steady-state phase).

TABLE III: Experimentation Settings

Experimentation Setting	Value(s) [Step]
Operating system	Ubuntu 18.04 LTS
K8s (kubelet, kubectl, kubeadm)	1.21.1
CRI-O	1.20
Flannel	0.14.0
No. of CNs	6
No. of replicas per worker	10, 20
Network scenario	LAN, AVG
Bandwidth (Mbps)	0.5 - 4 [0.5]
Latency (s)	0 - 8 [0.4]
Packet loss (%)	0 - 40 [2]

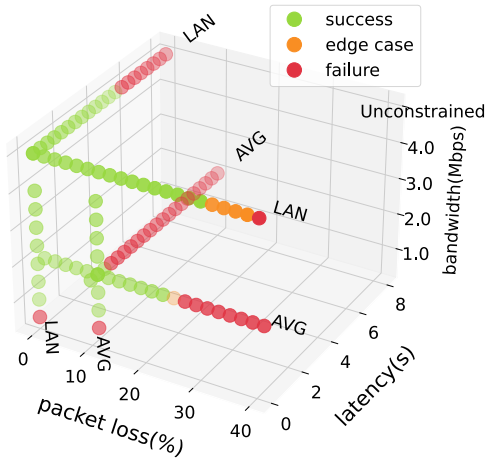


Fig. 2: Successes, edge cases, and failures in CI experiments. Labels AVG/LAN indicate the baseline for each data series.

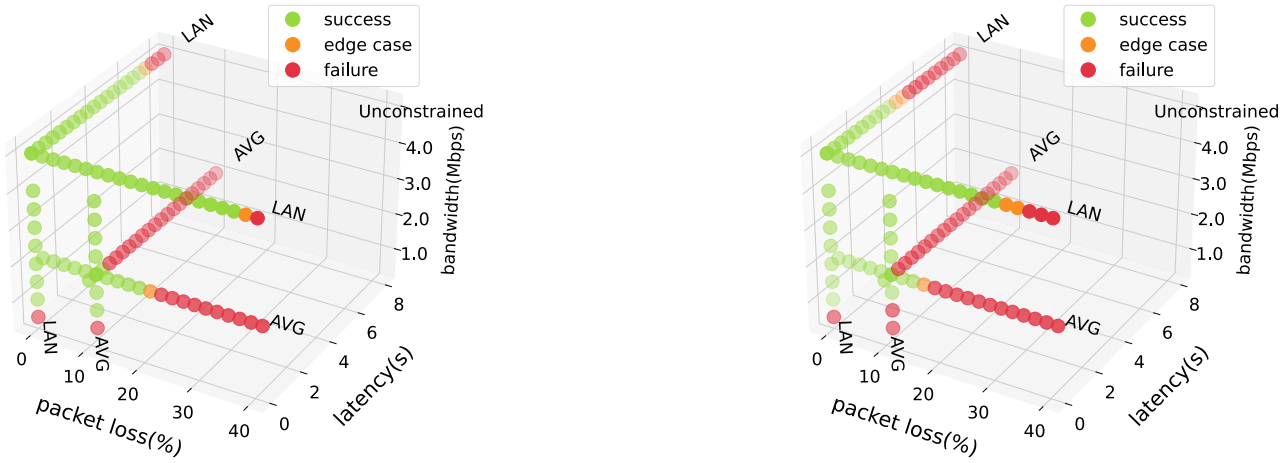
B. Application Deployment (AD)

AD experiments assume a K8s-based cluster already initialized and container images pre-downloaded on CNs (setup phase). Such a precondition makes the experiments independent from image sizes (ranging from MB to GB, potentially affecting the results) by only taking into account the network overhead introduced by K8s itself. Then, the test framework instructs the master to deploy a certain number of application replicas per worker (bootstrap phase). The cluster's current state matches the desired state when the master marks every replica as "ready" (steady-state phase).

C. Experimentation Settings

In [11], we provide experimental evaluations concerning the performance of K8s, K3s, and KubeEdge in four different network scenarios (LAN, BEST, AVG, WORST) affected by incrementally degraded network conditions. As shown in Table I, these scenarios vary along three dimensions (i.e., bandwidth, latency, and packet loss). It is worth mentioning that the available bandwidth in each scenario is distributed among CNs equally, which means that each CN consumes at most a fraction of the configured bandwidth. More specifically, Table II indicates the maximum available bandwidth per CN while considering the network scenarios and different cluster sizes. For example, in experiments involving six CNs and the AVG scenario, each CN can take up to 333 Kbps. Note that despite CNs being fully connected, control-plane communications follow a star topology, where workers only interact with the master (central point) and never with each other. Therefore, workers typically underutilize their slices of bandwidth while the master easily becomes a bottleneck. A High-Availability (HA) setup, i.e., multiple masters, might alleviate that bottleneck but imposes additional communication overhead in synchronization and distributed consensus mechanisms. Therefore, it is not considered in this work.

We tune the experimentation settings of this follow-up study based on our previous work [11], which shows that K8s-based



(a) 10 replicas per CN.

(b) 20 replicas per CN.

Fig. 3: Successes, edge cases, and failures in AD experiments. Labels AVG/LAN indicate the baseline for each data series.

clusters work at worst in the AVG scenario with at most six CNs. Such evidence is valid both for CI and AD experiments. Therefore, we extend the performance analysis using the AVG scenario as the baseline and vary only one network parameter at a time. Moreover, we also conduct a similar study but using the LAN scenario as the baseline, disclosing patterns about how network parameters influence each other. A full parameter sweep is performed for bandwidth (ranging from 0.5 Mbps to 4 Mbps with a step of 0.5 Mbps); packet loss (ranging from 0% to 40% loss with a step of 2%); and latency (ranging from 0 s to 8 s with a step of 0.4 s), totaling 295 performed experiments each repeated three times, as described in Section IV. Table III indicates the full spectrum of simulation settings. Note that such settings refer both to CI and AD experiments, except for the number of replicas per worker that is only valid for AD experiments. In contrast to the previous work, we use CRI-O [18] as the container runtime.

V. Performance Evaluation Results

Fig. 2 shows the results for CI experiments for every experiment setting, where each data point reflects a 3-times repeated experiment. If all iterations succeed, we mark the experiment as a success (denoted in green); otherwise, if at least one iteration succeeds and at least one fails, we label the experiment as an edge case (denoted in orange), whereas, if all iterations fail, we label the experiment as a failure (denoted in red). For clarity, we also label data series according to the network scenario (i.e., LAN and AVG) used as the baseline.

In the LAN scenario, where no communication effects are in place by default, we observe K8s to be most sensitive to increased latency, with failures occurring at an added latency of 5.2 s. Every CI experiment at least partially succeeds when only constraining the available bandwidth: the only failed iteration in that study is observed when capping the total available to 0.5 Mbps, indicating that CI is relatively robust to low bandwidth environments in the chosen range. In CI experiments varying just packet loss, K8s clusters are able to initialize successfully up to an added packet loss of 30%, after which partial failures are observed until finally full failures are seen at an added packet loss of 40%.

This effect is compounded in the AVG scenario. Here, CI struggles to succeed at just 1.2 s of added latency in this relatively small 6-CN cluster. This effect is not as strong in the other two studies where bandwidth is varied (fixing latency at 0.4 s, we only see a single failed experiment at 0.5 Mbps), or packet loss is varied (partial failures at 24% and full failures after 26% of added packet loss).

In Fig. 3a and 3b results concerning the AD experiments are presented, for deployments of 10 and 20 replicas per worker respectively. When compared with the CI experiments, these figures show similar results, where we observe robustness to low bandwidth environments, but we see increased sensitivity to packet loss.

We further investigate the relative sensitivity to each network parameter in Table IV and V, which present the parameter estimates of logistic regression (success vs. failure) on normalized results regarding CI and AD experiments, respectively. Specifically, Table IV shows that the coefficients and the related odds ratio are highest for latency ($B = 32.019$), lower for packet loss ($B = 19.650$), and lowest for bandwidth ($B = -13.079$, note the reverse in sign). This indicates that our data shows that a unit increase in latency will increase the odds of a failed experiment compared to success with a higher percentage contrasted with packet loss and bandwidth. Table V shows a similar relation, albeit that we observe variation in latency to have a much greater impact on AD success in the 20-replica setting compared to the 10-replica setting, whereas it is relatively less when contrasting AD to CI. This can be explained by the amount of network traffic required for successful deployment in AD experiments compared to that of CI, where workers communicate with the master for retrieving cluster information and certificate issuing during TLS bootstrapping.

TABLE IV: Parameter estimates of logistic regression (success vs. failure) on CI experiments

	B	SE	exp(B)	p-value
Intercept	-7.788	2.420	4.145e-4	0.001
Packet Loss	19.650	5.088	3.419e8	<0.001
Latency	32.019	8.124	8.048e13	<0.001
Bandwidth	-13.079	3.809	2.089e-6	<0.001

TABLE V: Parameter estimates of logistic regression (success vs. failure) on AD experiments

	B	SE	exp(B)	p-value
Intercept	-2.584	0.820	7.548e-2	0.002
Packet Loss	14.394	2.648	1.784e6	<0.001
Latency	19.536	3.389	3.049e8	<0.001
Bandwidth	-12.115	2.198	5.478e-6	<0.001

VI. Conclusion and Future Work

This work focuses on the applicability of K8s-based solutions in enabling an adaptive and federated architecture for supporting coalition tactical operations, which is ongoing research within the IST-168 RTG. Building on our previous work [11], we conducted extensive new experiments along the operational working range of K8s in circumstances of disadvantaged network conditions, showing that both CI and AD are most sensitive to increased network latency. In particular, the CI experiments fail at 1.2 s of added latency with the AVG scenario (i.e., 2 Mbps bandwidth, 10% packet loss) as the baseline, whilst remaining relatively robust to low bandwidth. We found these effects to be similar in the AD experiments.

In future work, we plan a similar follow-up study about KubeEdge, which is a promising solution specifically designed for resource-constrained scenarios. It is worth noting that KubeEdge is ancillary to K8s, expecting a K8s-based cluster (at least a master node) to be up and running beforehand. We believe that such a study will allow us to achieve the optimal synergy between K8s and KubeEdge, potentially enabling unprecedented orchestration capabilities in military mission contexts. In addition, we plan to further validate our findings by enriching our experimentation with the Anglova scenario, which emulates real-world tactical networking environments. We are also working to open source the test framework described in Section III-B and benchmark results to serve as an open experimentation platform for cloud research in disadvantaged environments. Lastly, we plan to investigate state-of-the-art strategies to adapt TCP-based COTS solutions (e.g., K8s) to TENs. Specifically, we will consider both alternative TCP congestion control schemes (e.g., BBR [19]) and proxy-based technologies (e.g., NetProxy [20]).

Acknowledgment

The work as presented in this paper is being done as part of the NATO IST-168 RTG on *Adaptive information processing and distribution to support Command and Control*. We would like to thank the NATO IST-Panel for providing us the opportunity to do this highly relevant and interesting research and the individual participating partners for providing valuable inputs within a simulating and cooperative setting.

References

[1] 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.

[2] J. Nobre, D. Rosario, C. Both, E. Cerqueira, and M. Gerla, "Toward software-defined battlefield networking," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 152–157, 2016.

[3] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Sdn-enabled tactical ad hoc networks: Extending programmable control to the edge," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 132–138, 2018.

[4] Q. Chen, H. Wang, and N. Liu, "Integrating networking, storage, and computing for resilient battlefield networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 56–63, 2019.

[5] I. Zacarias, L. P. Gaspary, A. Kohl, R. Q. A. Fernandes, J. M. Stocchero, and E. P. de Freitas, "Combining software-defined and delay-tolerant approaches in last-mile tactical edge networking," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 22–29, 2017.

[6] V. Sridharan, M. Motani, B. Jalaian, and N. Suri, "Exploring performance trade-offs in tactical edge networks," *IEEE Communications Magazine*, vol. 58, no. 8, pp. 28–33, 2020.

[7] H. Bastiaansen, J. v. d. Geest, C. v. d. 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.

[8] "Kubernetes (k8s): Production-grade container scheduling and management," https://github.com/kubernetes/kubernetes, [Online; accessed 23-July-2021].

[9] "Lightweight kubernetes (k3s)," https://github.com/k3s-io/k3s, [Online; accessed 23-July-2021].

[10] "Kubernetes native edge computing framework (kubeedge)," https://github.com/kubeedge/kubeedge, [Online; accessed 23-July-2021].

[11] M. Fogli, T. Kudla, B. Musters, G. Pingen, C. Van den Broek, H. Bastiaansen, N. Suri, and S. Webb, "Performance evaluation of kubernetes distributions (k8s, k3s, kubeedge) in an adaptive and federated cloud infrastructure for disadvantaged tactical networks," in *2021 International Conference on Military Communication and Information Systems (ICMCIS)*, 2021, pp. 1–7.

[12] H. Bastiaansen, C. van den Broek, T. Kudla, A. Isenor, S. Webb, N. Suri, A. Masini, C. Bilir, and M. Cocelli, "Adaptive information processing and distribution to support command and control in situations of disadvantaged battlefield network connectivity," in *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, 2019, pp. 1–7.

[13] N. Suri, K. M. Marcus, C. van den Broek, H. Bastiaansen, P. Lubkowski, and M. Hauge, "Extending the anglova scenario for urban operations," in *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, 2019, pp. 1–7.

[14] N. Suri, J. Nilsson, A. Hansson, U. Sterner, K. Marcus, L. Misirlioğlu, M. Hauge, M. Peuhkuri, B. Buchin, R. in't Velt, and M. Breedy, "The anglova tactical military scenario and experimentation environment," in *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, 2018, pp. 1–8.

[15] "Extendable mobile ad-hoc network emulator (emane)," https://github.com/adjacentlink/emane, [Online; accessed 23-July-2021].

[16] "Ansible: a radically simple it automation platform," https://github.com/ansible/ansible, [Online; accessed 23-July-2021].

[17] "The r project for statistical computing," https://www.r-project.org/, [Online; accessed 23-July-2021].

[18] "Cri-o: Lightweight container runtime for kubernetes," https://github.com/cri-o/cri-o, [Online; accessed 23-July-2021].

[19] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control," *Commun. ACM*, vol. 60, no. 2, p. 58–66, Jan. 2017. [Online]. Available: https://doi.org/10.1145/3009824

[20] M. Tortonesi, A. Morelli, C. Stefanelli, R. Kohler, N. Suri, and S. Watson, "Enabling the deployment of cots applications in tactical edge networks," *IEEE Communications Magazine*, vol. 51, no. 10, pp. 66–73, 2013.