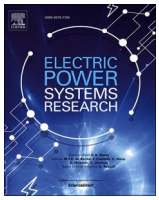
Electric Power Systems Research 201 (2021) 107547

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr



A distributed communication framework for smart Grid control applications based on data distribution service

Shivam Saxena [b,*], Hany E.Z. Farag [a,b], Nader El-Taweel [b]

[a] King Fahd University, Academic Belt Road, Dhahran 31261, Saudi Arabia
[b] Department of Electrical Engineering and Computer Science, York University, 4700 Keele Street, Toronto, ON, Canada M3J 1P3

ARTICLE INFO

Keywords:
Data distribution service
Voltage regulation
Software-in-the-loop
Internet of Things (IoT)
Multi-agent
Smart grid

ABSTRACT

The successful implementation of multi-agent control strategies (MAS) within smart grid applications is heavily reliant on communication frameworks that enable robust inter-agent messaging. In this regard, a promising communication framework is Data Distribution Service (DDS), which is fully distributed, enables low-latency messaging, and provides configurable Quality of Service (QoS) profiles that customize messaging parameters. As such, this paper proposes a DDS-based, fully distributed communication framework (DCF) for smart grid applications. The proposed DCF is configured with QoS profiles that discard outdated messages when agents experience intermittent connection issues, thereby preventing the misaligned execution of distributed control strategies. The performance of the proposed DCF is validated by developing a Software-In-The-Loop platform, where agents utilize the DCF to execute the Asynchronous Weak Commitment (AWC) control strategy within the application of distributed voltage regulation. Experimental results show that the proposed DCF delivers messages at latencies between 4-36 milliseconds, which is sufficient for smart grid applications. The proposed DCF also enhances the efficacy of the AWC strategy when performing voltage regulation by discarding outdated messages and preventing unnecessary control actions in resolving voltage violations. The proposed DCF is validated at a real-world Canadian microgrid, where overvoltage violations are resolved in a timely manner.

1. Introduction

Under the smart grid paradigm, centralized legacy power systems are slowly being decentralized into modular subsystems that rely heavily on distributed energy resources (DERs) to satisfy local operational objectives [1]. However, the massive growth of DERs places a great deal of computational burden on conventionally *centralized* power system control strategies that are vulnerable to communication bottlenecks and scalability issues [2]. Thus, researchers are turning towards *distributed*, multi-agent control strategies [3], where autonomous agents assume responsibility of independent subsystems and execute control strategies by exchanging information only with neighboring agents. As such, multi-agent systems (MAS) have recently been used in many different smart grid applications that seek to utilize the controllability of distributed generation units in achieving a particular control objective [4,5], such as voltage regulation and service restoration [6–8].

However, the emphasis of previous works within MAS smart grid applications have focused on the formulation of the control strategy, while the underlying communication framework has often been neglected [9]. Given that distributed control strategies are highly dependent on inter-agent communication, there is significant motivation to develop communication frameworks that mitigate communication issues and improve agent performance. Thus, the works in [6–8] develop communication frameworks that enable agents to provide robust performance despite the presence of time delays and communication link failures. Yet, the previous works do not clearly describe how agent messages are generated, stored, and routed to the correct agent. Specifically, the issue of outdated and improperly ordered messages being sent to agents when they reconnect to a network is not discussed. This issue is significant because it results in the misaligned execution of agents participating in a control strategy [10]. Furthermore, much of the previous work employs the popular Java Agent Development Environment (JADE) platform as the communication framework [7,11–14]. However, JADE utilizes a singular, centralized message transport system (MTS) to facilitate inter-agent communication, which is not robust since it exposes a single point of failure [15].

As such, there is a need to develop a new, distributed communication framework for MAS to alleviate the aforementioned shortcomings. In this regard, a promising communication framework is Data Distribution Service (DDS), which exists within the paradigm of the Internet of

* Corresponding author.

https://doi.org/10.1016/j.epsr.2021.107547
Received 31 March 2021; Received in revised form 4 August 2021; Accepted 20 August 2021
Available online 3 September 2021

Things (IoT) [16]. DDS is a *fully distributed* communication middleware that provides low-latency, high-throughput communication between agents. DDS supports the feature of dynamic discovery, which enables agents to discover other agents based on common sets of data they have interest in, thereby obviating the need for a centralized MTS to coordinate agent interaction. DDS also provides Quality of Service (QoS) profiles, which are configurable settings for data flow that are based on the data's ordering and priority, thereby offering great flexibility in optimizing the performance of distributed control strategies.

The use of DDS has been proposed in recent smart grid applications [17–24]. The work in [17] utilizes DDS to test the feasibility of information exchanges between system operators and metering devices, with several case studies exploring the effect of different QoS profiles on latency and jitter. The work in [18] leverages DDS as a secure message framework to facilitate distributed state estimation across networked microgrids. In [19], DDS is used to synchronize three-phase measurements with varying priority levels. On the other hand, the work in [20], a DDS-based protection scheme is proposed to island microgrids when under a cyber-attack. In [21], DDS is used to manage data flow in a demand response market for microgrids. In [22], an energy management system is proposed, which uses DDS to facilitate dynamic pricing signals to smart meters. In [23], DDS dynamic discovery is used to synchronize the connection of new microgrids to the main grid. Lastly, in [24], a DDS-based hardware-in-the-loop (HIL) testbed is developed, which is validated on smart grid applications such as microgrid islanding and frequency regulation.

The aforementioned works have made strong contributions in exploring the usage of DDS within smart grid applications, however, improvements can be made in the following aspects, which are summarized in Table 1 and expanded on below. First, the scalability of inter-agent DDS messaging has not been appropriately evaluated. The works in [17,19,22,24] benchmark the transmission latency of DDS messages on a single computer, which does not adequately reflect the entire agent communication process that occurs between multiple agents on multiple platforms. Second, the aforementioned works do not discuss how the QoS profiles of DDS can be tailored to mitigate the issue of agents acting on outdated messages. Third, the previous works do not enforce the compliance of DDS with MAS communication standards, which are specified by the Foundation of Intelligent Physical Agents [25]. Fourth, the previous works do not validate the efficacy of DDS in real-world settings, where lessons learned from practical implementation can be derived.

Motivated by the above discussion, this paper proposes a distributed communication framework (DCF) for MAS in smart grid applications that is based on DDS. The DCF is implemented as a *fully distributed* entity that provides robust message transmission since it is not vulnerable to a single point of failure. Further, the DCF is designed with QoS profiles that automatically discard outdated messages, thereby preventing agents from taking incorrect decisions that could result in the misaligned execution of their control strategy. The DCF is also designed to be compliant with FIPA standards. The performance of the DCF is validated by developing a Software-In-The-Loop (SIL) platform that deploys multiple agents on multiple real-time micro-controllers, where the message latency is benchmarked as a function of message rate and message size, and compared against minimum latency requirements for smart grid applications [26]. The developed SIL platform is tested within the smart grid application of distributed voltage regulation in active distribution networks (ADNs), where agents formulate the problem of voltage regulation as a Distributed Constraint Satisfaction Problem (DisCSP), and solve it by exchanging messages in accordance with the distributed Asynchronous Weak Commitment (AWC) control strategy [27]. Simulation experiments are carried out on a modified 39 bus test feeder in order to validate the superiority of the proposed DCF in enhancing the efficacy of the AWC control strategy by automatically discarding outdated messages when agents rejoin the network, thereby preventing the misaligned execution of the AWC. Lastly, the DCF is validated by deploying it to a real-world Canadian microgrid to demonstrate its ability to mitigate overvoltage violations using the AWC control strategy.

Table 1
Comparison table of related work with proposed work.

Reference	FIPA	Bench-mark	Outdated Messages	Deployment
[17,19,22] [18,20,21,23]		✓		Simulated HIL
[24]	✓	✓		HIL
Proposed paper	✓	✓	✓	Real-world

The main contributions of this paper are summarized as follows: (1) proposal of a new, DDS-based DCF as a suitable means for inter-agent communication within smart grid applications, (2) development of a SIL platform that benchmarks the performance of the proposed DCF against minimum latency requirements for smart grid applications, (3) utilization of the proposed DCF to enhance the conventional AWC control strategy to prevent agents from acting on outdated messages, and (4) validation of the DCF in resolving voltage violations at a Canadian microgrid. As seen in Table 1, the proposed paper is the only paper that satisfies all comparable criteria.

The remainder of the paper is organized as follows: Section 2 provides an overview of DDS, while section 3 introduces the architecture of the proposed DCF. Section 4 introduces the problem of distributed voltage regulation as a DisCSP and highlights how the proposed DCF is used to enhance the efficacy of the AWC control strategy in solving the DisCSP. Section 5 presents experimental results related to DCF benchmarking, as well as the mitigation of voltage violations in ADNs in both simulated and real-world scenarios. Finally, Section 6 concludes the paper.

2. Overview of DDS middleware

In communication frameworks, middleware is the component that is responsible for the transport, buffering, and routing of all messages [28]. IoT-based middlewares often utilize a topic-based, publish-subscribe messaging paradigm that defines a topic as a named channel of data that is comprised of a stream of messages [29]. An example of this paradigm can be seen in Fig. 1, where Agent 1 generates a request message and publishes it to *Topic Y*, and subsequently, the message is received by Agent 2 because it is also subscribed to *Topic Y*. In turn, Agent 2 publishes a response message to *Topic Y*, which is received by Agent 1 because it is also subscribed to *Topic Y*. The message buffering is handled by bi-directional message buffers, which store incoming published messages and release the messages to agents when they are online. It is worth noting that each topic can be configured with a QoS profile, which alters how the middleware transports the message [28]. Examples of QoS profiles include the ability to prioritize certain messages to improve their latency, defining a maximum acceptable duration latency, and specifying the exact order that an agent should receive messages in. Thus, it can be seen that the execution of distributed control strategies can benefit from utilizing the flexible range of QoS profiles to increase their operational efficacy.

DDS is a fully distributed middleware that has upwards of 20 unique QoS profiles. As shown in Fig. 2, DDS defines a distributed global data space (GDS) that is comprised of all the topics and their associated QoS profiles. Agents merely need to declare their publishing/subscribing interest in a topic, and the GDS utilizes distributed multi-cast techniques to ensure that data flows directly from the publishing agent to the subscribing agent [30]. Additionally, DDS provides a feature known as *content filtering*, where agents can specify conditional expressions that define a subset of messages they are interested in receiving from the GDS [31]. An example of this is shown in Fig. 2, where Agent 1 and Agent 2

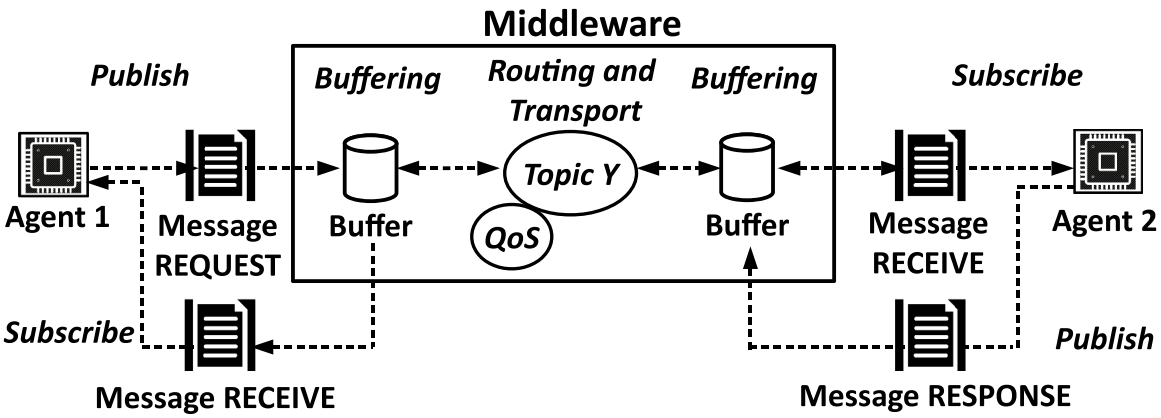


Fig. 1. Message flow in topic-based, publish-subscribe middleware.

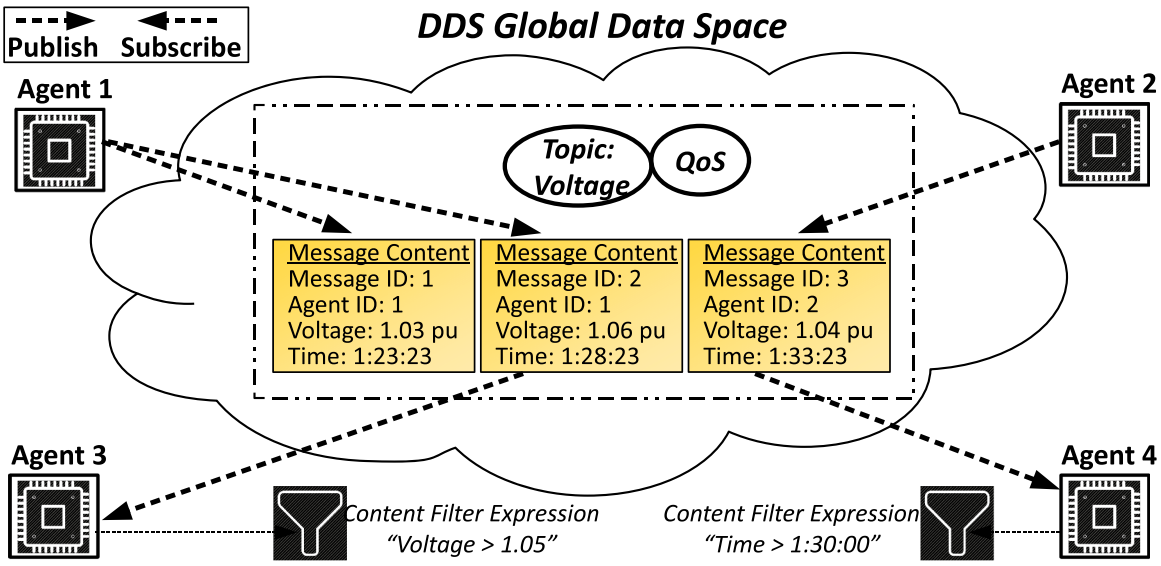


Fig. 2. The DDS Global Data Space that enables dynamic routing of messages depending on content filtering expressions.

publish messages to the *Voltage* topic in the form of voltage measurements, and Agent 3 and Agent 4 receive messages from the *Voltage* topic because they are subscribed to it. However, Agent 3 specifies a content filter that indexes all messages within the *Voltage* topic by the "Voltage" field, and accepts only the messages where the voltage exceeds 1.05 p.u., while similarly, Agent 4 specifies a content filter that only accepts messages when the "Time" field exceeds 1:30:00. Since the GDS has the ability to understand and evaluate the data inside of a message, the GDS can execute content filters automatically, thus relieving Agents 3 and 4 of computational burden.

DDS is designed for real-time, mission critical applications and can send over one million messages per second to multiple subscribers [28]. Considering the fact that DDS provides deterministic bounds for acceptable message latency, and uses completely a distributed communication architecture, DDS is suitable for smart grid applications that often require message latency ranging from 15 - 200 ms [26]. For these reasons, the proposed DCF will utilize DDS as its communication middleware.

3. Design of proposed DCF

This section introduces the design of the proposed DCF architecture, where technical details are given with respect to how the proposed DCF complies with the FIPA standard, as well as how QoS profiles are customized to enhance message latency and robustness. Subsequently, the messaging workflow of the agents is discussed, which is followed by a description of the SIL platform that is developed to host the proposed DCF. The section is concluded with a discussion that explains how the proposed DCF enhances the efficacy of distributed control strategies by preventing misaligned execution when agents frequently disconnect and rejoin a network.

3.1. DCF architecture

The proposed architecture of the DCF is designed to leverage the strengths of DDS while also including all of the essential components of MAS communication architecture as specified by FIPA. These components include an agent management system (AMS), a directory facilitator (DF), as well an MTS, where the MTS was discussed in Section 1 [32]. The AMS is used for registering the identities of all agents within a network, while the DF is used for registering a list of services that agents provide. The MTS is responsible for facilitating all messages between agents.

To mitigate the vulnerability of using a centralized MTS implementation, as observed in JADE, the architecture of the proposed DCF is designed using a distributed approach. The AMS, DF, and MTS are all moved into the distributed GDS that is shared by all DCF agents, and mapped to separate topics that all agents subscribe to. Accordingly, the

three main topics defined in the GDS are *Services, AgentID*, and *MsgBus*, which are mapped to the DF, AMS, and MTS, respectively, and are shown in Fig. 3. The *Services* topic allows an abstract definition of services to be defined for each agent, where in the context of smart grid applications, services may comprise of ancillary services such as voltage regulation and demand response. The *AgentID* topic provides a lookup table of agent identities, which are indexed by the specific subsystem, or spatial zone of operation that they govern. The *MsgBus* topic is used by all agents to exchange messages in real-time. Fig. 3 shows the underlying data structure of the *MsgBus* topic, which conforms to the agent communication language (ACL) specifications as per FIPA, where the sending (*Sender_Agent_ID*) and receiving (*Receiver_Agent_ID*) IDs of the agents, as well as the performative and content of the message are explicitly defined.

The *MsgBus* topic is also configured with three unique QoS profiles. The TRANSPORT_PRIORITY profile prioritizes the delivery of messages within the *MsgBus* topic over all other topics, while the LATENCY_BUDGET defines the maximum allowable latency for each message associated with the *MsgBus* topic. The PRESENTATION profile ensures that messages are received by an agent in the exact order they were sent, thus preventing agents from acting on misrepresented data.

The mapping of FIPA-specified components to topics in the DCF has four major advantages. First, since the GDS is inherently distributed, the *MsgBus* topic does not expose a single point of failure for agent communication, unlike JADE. Second, by abstracting the *Services* and *AgentID* topics away from the *MsgBus* topic, agents can receive the latest updates to agent IDs and services by subscribing to these topics, thus leaving the *MsgBus* topic to facilitate high-priority messages when executing distributed control strategies. Third, to route messages correctly and efficiently, the DCF specifies content filters on the *Receiver_Agent_ID* field and the *Time* field within the *MsgBus*. These content filters enable the *MsgBus* to automatically filter out messages that are not intended for the correct agent, as well as messages that exceed a configurable period of time and are outdated. Fourth, the unique QoS configuration of the *MsgBus* topic with the PRESENTATION profile enables agents to receive messages in order, which prevents the misaligned execution of distributed control strategies.

A potential security concern with the *MsgBus* topic may be realized if a malicious agent sends fraudulent messages by incorrectly specifying the *Sender_Agent_ID* field within a message, or eavesdrops on unauthorized messages by specifying an incorrect *Receiver_Agent_ID* field. To mitigate this, the DCF sets access control privileges for each field inside the *MsgBus* topic by specifying read/write access levels for all agents. The DCF architecture imposes read-only access to the

Receiver_Agent_ID field when establishing a subscription to the *MsgBus* topic, and similarly imposes read-only access for the *Sender_Agent_ID* field when publishing to the *MsgBus* topic. The DCF presets the correct ID for the agent as soon as the agent joins the network, and as such, agents using the *MsgBus* topic can only access messages they are authorized for. It is worthwhile to mention that the DCF relies on encryption features provided by DDS to validate the authenticity of messages during transport, which mitigates, to some extent, false messages generated by malicious agents [22]. However, malicious agents that have access control privileges and access to public/private key pairs used for encryption purposes may still inject false messages. Strategies to mitigate false data injections, however, are out of scope for this paper,

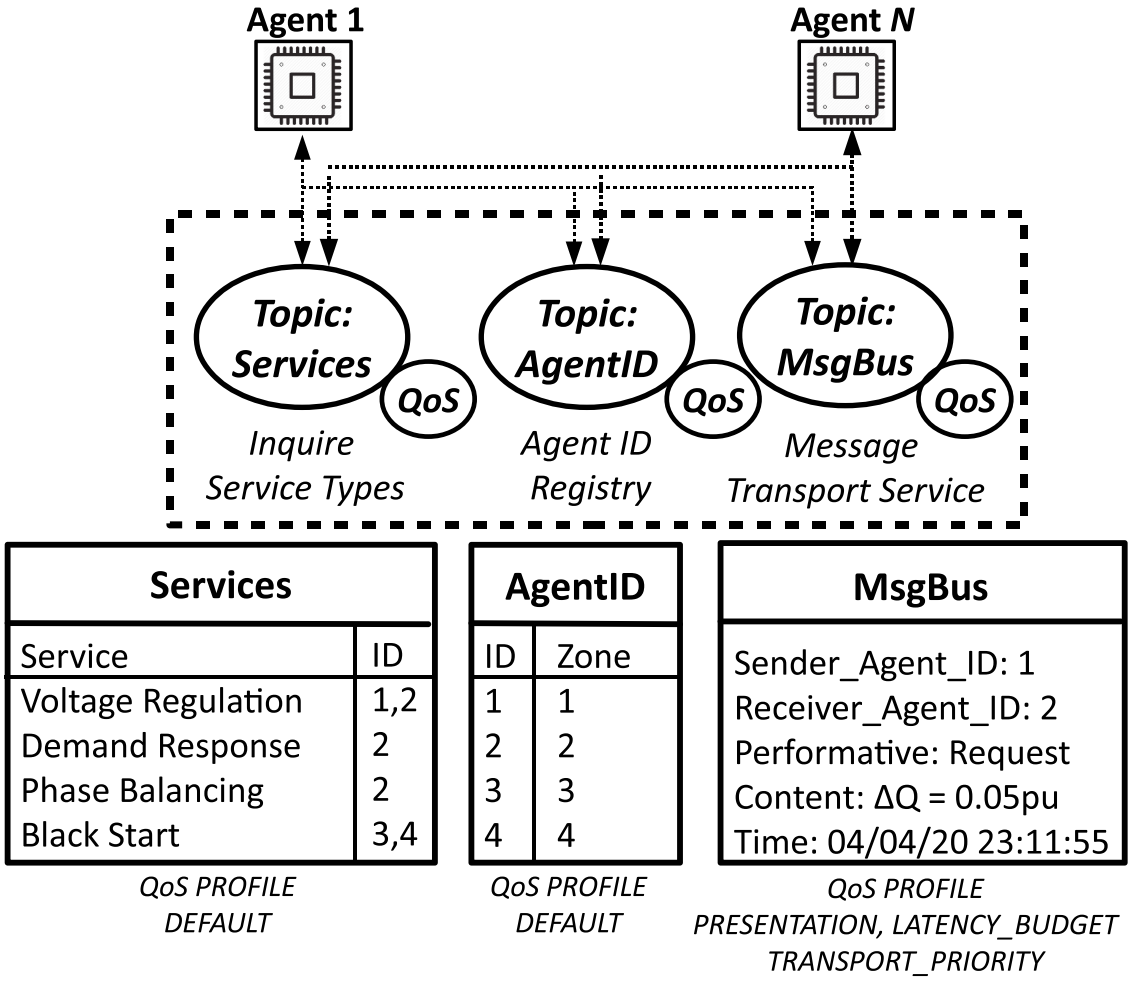


Fig. 3. Depiction of the topics and underlying data structures of the DCF Architecture.

but can be found in [33]

3.2. Messaging workflow of DCF agents

An example of the messaging workflow of agents using the DCF is given in Fig. 4. When launched, all agents must initialize their settings by establishing publishing/subscribing access to the mandatory DCF topics, which are *Services, AgentID*, and *MsgBus*. Agents may then engage in request/response messaging as shown in Fig. 4, where the format of the message would be identical to the data structure of the MsgBus depicted in Fig. 3. Namely, Agent 1 would send a request message to Agent 2 by specifying the *Sender_Agent_ID* field as *1*, the *Receiver_Agent_ID* field as *2*. the *Performative* field as *Request*, the *Content* field with the appropriate content, and the *Time* field with the current timestamp. The DCF would filter the message for Agent 2 by the *Receiver_Agent_ID* field, and send the message to Agent 2, which would send back an appropriate message response depending on the control strategy that is being executed.

3.3. Proposed SIL platform for DCF implementation

To validate the ability of the DCF to execute distributed control strategies for smart grid applications, a SIL platform is developed. The SIL platform is composed of networked real-time microcontrollers that host multiple agents, all of which use the *MsgBus* topic to exchange messages. The mathematical system model of any power system can be inserted directly into the simulation loop as a special agent, referred to as the model agent, as seen in the block diagram in Fig. 5.

The model agent begins the simulation process by generating a state vector that represents the power system state at a discrete time step k, denoted as $X(k)$. The state vector is published to the SimData topic, which each agent is subscribed to, and each agent content-filters a subset of the overall state vector that pertains to their zone of operation, which is denoted as $X(k) = \{X_1(k)...X_n(k)\}$. The agents then use the *MsgBus* to exchange messages and coordinate their control actions according to a specified control strategy, where the control actions are represented by $\alpha(k) = \{\alpha_1(k)...\alpha_n(k)\}$. The finalized control actions are then published to the *ControlActions* topic, which is subscribed to by the model agent. The model agent gathers all control actions, updates the overall state vector, and executes the system model to generate a new state vector at time $k+1$. The process repeats at the next time step, and is represented as follows

$$X(k+1) = \boldsymbol{F}(X(k) + \alpha(k)) \tag{1}$$

where $\boldsymbol{F}$ is a symbolic function for the system model. In this paper, the system model will be representative of a physical ADN, and agents will utilize the *MsgBus* to exchange messages in accordance with the distributed AWC control strategy for the application of distributed voltage regulation. It is worthwhile to note that the proposed DCF is representative of heterogeneously communicating MAS, where agents have different control actions and sensory inputs within an environment, but use a semantically consistent agent communication language to communicate and cooperate with each other [34]. As mentioned earlier, the agent communication language is consistent with FIPA performatives, and implemented directly within the *MsgBus* topic itself. This paper discusses the application of voltage regulation, where agents control BESS and/or DG units, which leads to different available control actions and heterogeneous behavior.

3.4. Application of DCF in improving distributed control strategies

Many conventional distributed control strategies, such as the AWC strategy utilized in this paper, define two assumptions that limit their efficacy [27]. First, it is assumed that the agents have prior and continuous knowledge of the physical network address of each agent, such that messages can be directed to the intended agent correctly. Second, it is assumed that the agents receive messages exactly in the order that they were sent. These assumptions are unrealistic because they imply that the agents remain connected to the network at all times. Yet, agents frequently disconnect and rejoin the network when communicating over unreliable physical interfaces, and upon rejoining the network, they are assigned to new network addresses, and begin receiving an influx of messages that are out of order and may be outdated [10].

The design of the DCF removes the need for these assumptions. First, agents do not need to know the physical address of other agents because they exchange messages by explicitly defining the *Receiver_Agent_ID* within the message via the *MsgBus* topic, and content filters are utilized to ensure that the message is routed to the correct agent. All agents are also subscribed to the *Services* and *AgentID* topics within the DCF, and are instantaneously updated if there are changes to an agent ID, which obviates the need for agents to poll for the latest physical network addresses. Second, the QoS profile of PRESENTATION for the *MsgBus* topic ensures that messages are received by the agent in the exact order in which they were sent. Third, a content filter is placed on the *Time* field of the *MsgBus* topic, and thus, the DCF is able to filter out any outdated messages based on a reference point of time, thus ensuring that agents do not act on outdated messages.

As such, the DCF enables the execution of distributed control strategies to support scenarios where some agents may frequently disconnect and rejoin a network, but will still be able to execute the control strategy in alignment with the rest of the agents, which increases its robustness. Examples of these scenarios are when the host platform of the agent experiences a device failure or a network disconnection, where the latter scenario will be presented in the experimental results in Section 5.

4. The application of DCF within distributed voltage regulation

This section seeks to illustrate how the DCF can be applied to the smart grid application of distributed voltage regulation within ADNs. As discussed earlier, the application of distributed voltage regulation is formulated as a DisCSP, which is solved by agents using the AWC control strategy. This can be seen by the three layers shown in Fig. 6, where Layer 1 depicts the physical infrastructure of the ADN clustered into virtual zones of operation, Layer 2 depicts how agents formulate the DisCSP, and Layer 3 shows how agents execute the AWC control strategy to regulate the voltage of each zone, where the message exchange is

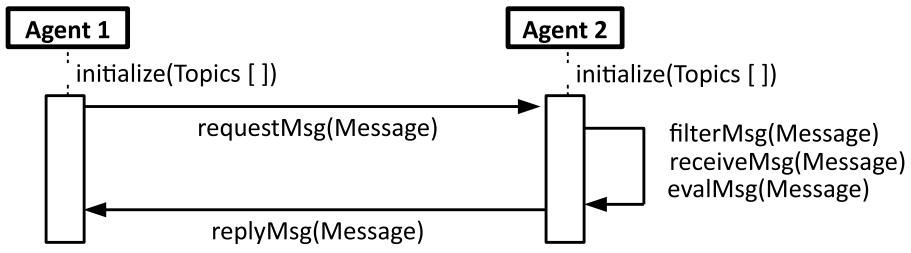


Fig. 4. Typical agent workflow for message exchange.

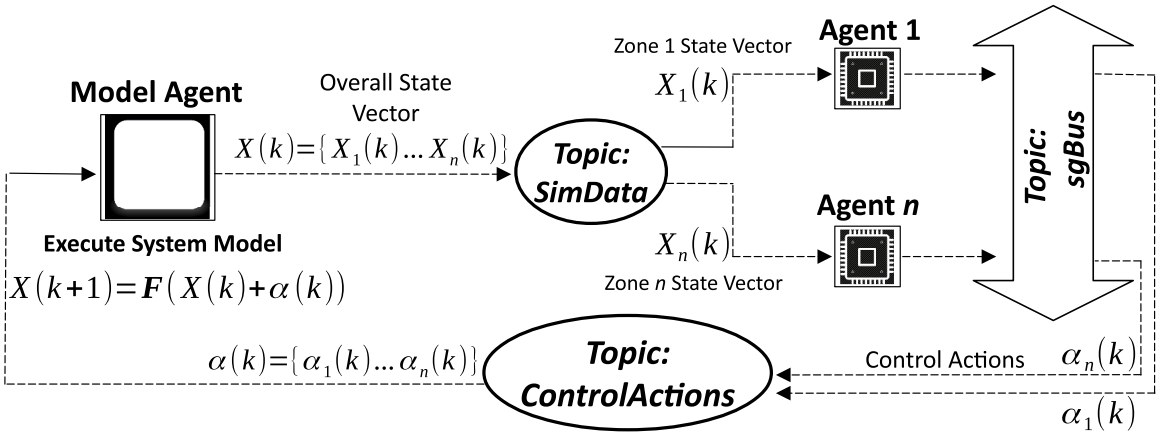


Fig. 5. Block diagram of the proposed SIL platform.

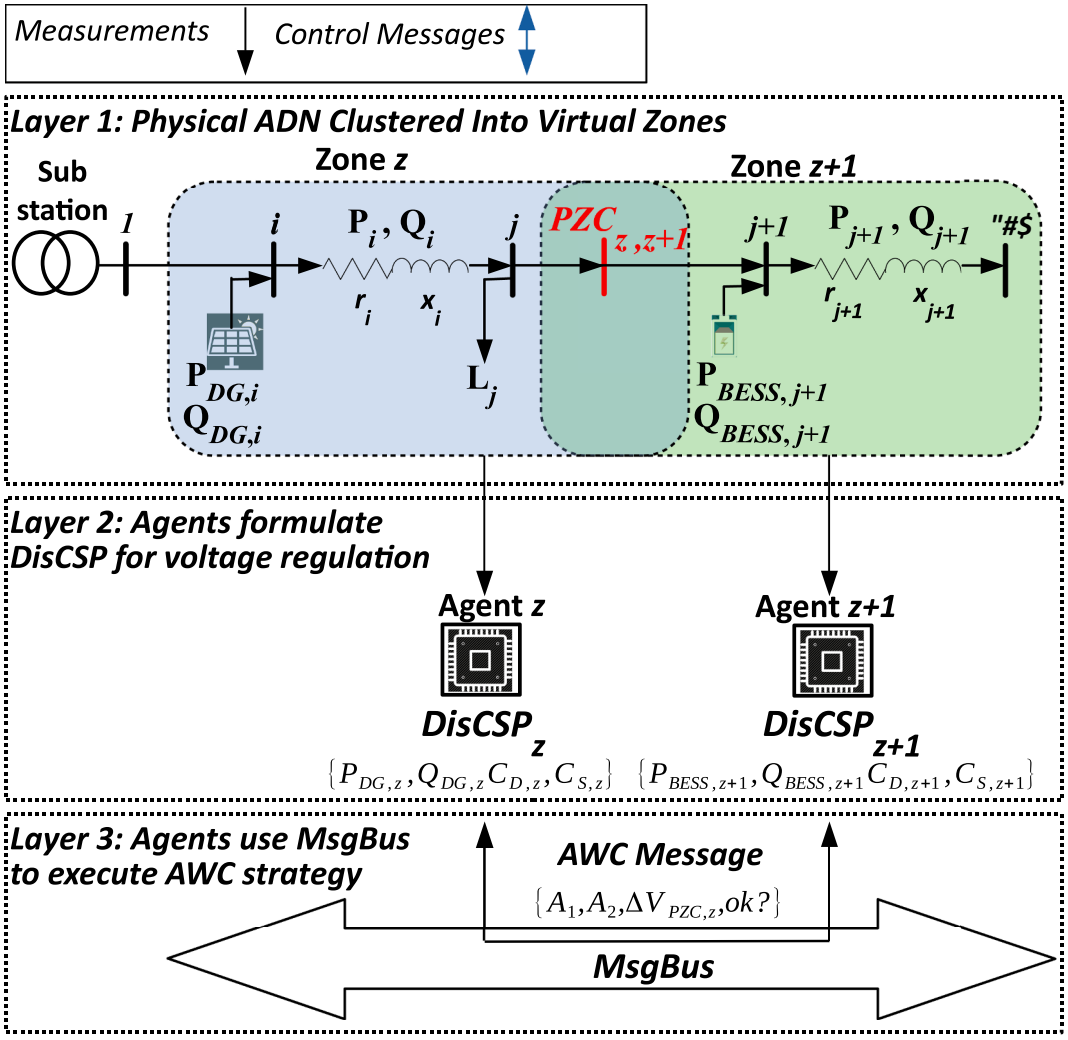


Fig. 6. Illustrative example of agents using the DCF to resolve voltage violation via the AWC control strategy.

handled by the *MsgBus* of the DCF.

4.1. Layer 1: Modeling and virtual zoning of ADNs

In Fig. 6, two types of DERs are connected to the ADN feeder, which are renewable distributed generation units (DGs), and battery energy storage systems (BESSs). BESSs are assumed to be capable of operating in the four quadrants, i.e., being able to modulate their active and reactive power output, while renewable DGs are assumed to operate at the maximum power point tracking setting [35] when the system voltage is at normal levels [36]. In general, at timestep t, the active and reactive power flow in branch i, and the voltage at bus i, can be represented with respect to bus j as follows [37]:

$$P_i(t) = P_j(t) + \sum_{n=i}^{n=j-1}\left(P_{L,n}(t) - P_{BESS,n}(t) - P_{DG,n}(t)\right) \tag{2}$$

$$Q_i(t) = Q_j(t) + \sum_{n=i}^{n=j-1}\left(Q_{L,n}(t) - Q_{BESS,n}(t) - Q_{DG,n}(t)\right) \tag{3}$$

$$V_i^2(t) = V_j^2(t) - 2\sum_{n=i}^{n=j-1}\left(r_n P_n(t) + x_n Q_n(t)\right) \tag{4}$$

where n is the branch iterator, $\{Pn, Qn\}$ are the output active/reactive powers, $\{P_{L,n}, Q_{L,n}\}$ are the active/reactive loads, $\{P_{BESS,n}, Q_{BESS,n}\}$ are the injected active/reactive powers from the BESS, $\{P_{DG,n}, Q_{DG,n}\}$ are the injected active/reactive powers from the DG, V_i is the voltage at each bus, and $\{r_n, x_n\}$ is the resistance/reactance of each feeder. It can be observed from (2)-(4) that the voltage profile across the feeder depends on the change of the branches active and reactive power flow along the feeder, which may cause voltage violations due to the intermittent changes of DGs. Simplifying (4) by linearization and neglecting the change in load demand during these events, the change in voltage can be expressed as:

$$\Delta V_i(t) = \frac{1}{V_i(t-1)}\left[V_1(t-1)\Delta V_1(t) + \Delta P_{DG,i}(t)\sum_{n=i}^{i-1} r_n + \Delta Q_{DG,i}(t)\sum_{n=i}^{i-1} x_n\right] \tag{5}$$

Since the substation V_1 holds its own voltage steady, $\Delta V1$ can be set to zero, resulting in:

$$\Delta V_i(t) = \Delta P_{DG,i}(t)\frac{\sum_{n=i}^{i-1} r_n}{V_i(t-1)} + \Delta Q_{DG,i}(t)\frac{\sum_{n=i}^{i-1} x_n}{V_i(t-1)} \tag{6}$$

Hence, the sensitivity at each bus due to active/reactive power injections can be calculated as follows:

$$SP_i = \frac{\partial V_i}{\partial P_i} = \frac{\sum_{n=i}^{i-1} r_n}{V_i(t-1)}, SQ_i = \frac{\partial V_i}{\partial Q_i} = \frac{\sum_{n=i}^{i-1} x_n}{V_i(t-1)} \tag{7}$$

A BESS deployed to a bus within the feeder can be used dynamically to regulate the voltage via active/reactive power modulation. Thus, any change in the power output of a BESS located at bus j will impose a change upon node i voltage as follows:

$$\Delta V_i(t) = \frac{1}{V_i(t-1)}\left[V_j(t-1)\Delta V_j(t) + \Delta P_{BESS,i}(t)\sum_{n=i}^{i-1} r_n + \Delta Q_{BESS,i}(t)\sum_{n=i}^{i-1} x_n\right] \tag{8}$$

The physical ADN can then be clustered into z zones of operation and assigned to an agent that uses the control actions of the BESS and/or DG of its zone to regulate the zonal voltage. Each zone shares a bus with its adjacent zone, which is referred to as the point of zone coupling (PZC). Thus, the agents are jointly responsible for regulating the voltage at the PZC (V_{PZC}), and must communicate with each other when executing control actions that may affect V_{PZC}. However, the control actions are subject to certain constraints, which will be exemplified in the next subsection that discusses the DisCSP.

4.2. Formulation of distributed voltage regulation as DisCSP

The formulation of a DisCSP involves a set of agents that execute control actions to arrive at a state where all system constraints are satisfied [27]. In this paper, the agents seek to adjust the control agents of their BESS/DG units to regulate the voltage of their zone, subject to constraints of device ratings and feeder capacity. If the constraints do not permit the local mitigation of a voltage violation, the agent may exchange messages with neighboring agents, who, in turn, adjust the control actions of their BESS/DG units to help mitigate the violation. It is worth noting that the problem formulation is meant for *constraint satisfaction* and not necessarily *optimality*, where the control actions may be influenced by other related objectives, such as power loss minimization, as found in [38]. Thus, the application of distributed voltage regulation as a DisCSP can be formulated as follows:

$$DisCSP_z = \boldsymbol{F}\left(\Delta P_{BESS,z}, \Delta Q_{BESS,z}, \Delta P_{DG,z}, C_{D,z}, C_{S,z}\right) \tag{9}$$

where $\{\Delta P_{BESS,z}, \Delta Q_{BESS,z}, \Delta P_{DG,z}\}$ represent the control actions available to each agent in zone z when resolving a voltage violation, which are constrained by both device constraints, $C_{D,z}$, as well as system constraints, $C_{S,z}$. The limits of these constraints form a domain from which the control actions can be determined. The device constraints include the manufacturer's limits for the range of operation for the BESS capacity, state of charge (SoC), the maximum apparent inverter capacity, as well as the minimum power factor, as shown below.

$$C_{D,z} = \begin{cases} P^{Pre}_{BESS,z}(t+1)\Delta t \le B^{cap}_{BESS,z} \\ SoC^{min}_{BESS,z} \le SoC_{BESS,z}(t) \le SoC^{max}_{BESS,z} \\ \sqrt{P^2_{BESS,z}(t+1) + Q^2_{BESS,z}(t+1)} \le S^{cap}_{BESS,z} \\ PF_{BESS,z}(t) \le PF_{min} \end{cases} \tag{10}$$

where $P^{Pre}_{BESS,z}$ is the predicted active power at the next timestep, $B^{cap}_{BESS,z}$ is the BESS capacity in kWh, $SoC^{min}_{BESS,z}$, $SoC^{max}_{BESS,z}$, $SoC_{BESS,z}$ are the minimum, maximum, and current SoC of the BESS, $S^{cap}_{BESS,z}$ is the BESS inverter capacity in MVA, and PF_{min} is the minimum power factor. On the other hand, the system constraints describe the limits of the voltage at every bus within the zone as well as the current flowing through the zonal feeder.

$$C_{S,z} = \begin{cases} I_i \le I^{cap}_{f,z} & \forall\, i \in F^z \\ V_{lb} \le V_{i,z} \le V_{ub} & \forall\, i \in N^z \end{cases} \tag{11}$$

where $I^{cap}_{f,z}$ is the carrying capacity of a zonal feeder, F^z is the set of all feeders within zone z, $\{V_{lb}, V_{ub}\}$ are the lower and upper voltage limits for $V_{i,z}$, and N^z is the set of all nodes within zone z. Note that $V_{i,z}$ includes the PZC, which is a common constraint to adjacent zones and requires message exchanges between agents to ensure that any control action does not violate PZC constraints.

4.3. Solution of DisCSP: Asynchronous weak commitment

The AWC is a distributed control strategy to solve DisCSPs [27]. As shown in Fig. 7, every agent executing the AWC maintains a local agent-view that stores requests from other agents, where the agents must evaluate the requests against local constraints before determining a control action. Thus, there are five main message types within the AWC paradigm, which include: i) an *ok?* message that represents an agent's request to another agent, ii) a *good* message that a responding agent sends when the *ok?* request satisfies all constraints, iii) a *no-good* message that is the opposite of a *good* message, iv) an *execute* message that a requesting agent sends to confirm execution of a control action, and v) an *executed* message that a responding agent sends when the control action is executed. Since all agents operate concurrently, it is possible that they may encounter two separate agent requests at the same timestep. To avoid deadlock, the agent prioritizes the agent request that violates the least number of constraints, which is referred to as the *min-conflict heuristic* [27]. Thus, mapping the AWC technique to the proposed application, an agent that detects a voltage violation initiates a

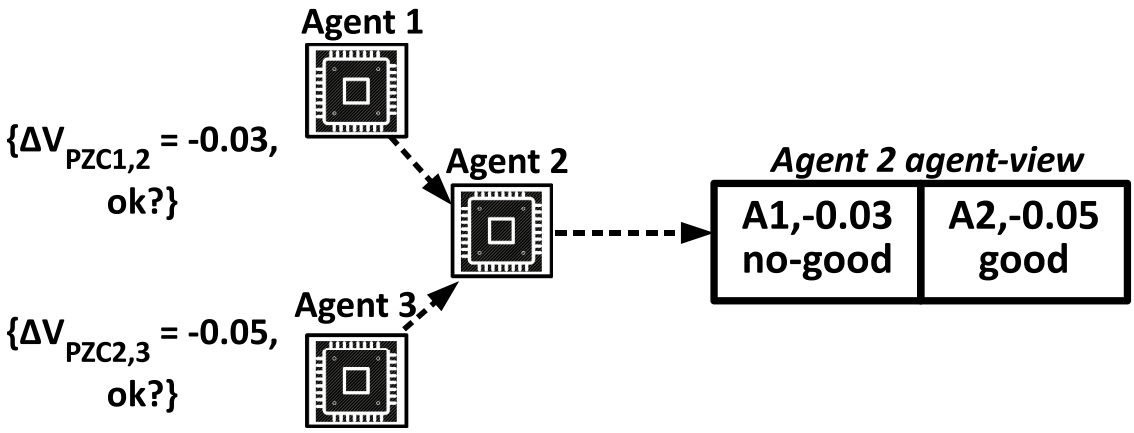


Fig. 7. Message exchange during AWC and an agent's agent-view.

DisCSP by specifying the change in voltage needed at its PZC, $V_{PZC,z}$, and encoding it within an *ok?* message to its neighbors. The neighboring agents evaluate the message and determine if it violates any device or system constraints, and send back an appropriate *good* or *no-good* message. If an agent has more than one request in its agent-view, as shown in Fig. 7, it uses the min-conflict heuristic to prioritize the violation with the highest $V_{PZC,z}$. If a *good* message is sent back, a subsequent *execute* message will actuate the agent's BESS/DG units to resolve the voltage violation.

It is worth noting that all message exchanges for the AWC are facilitated by the *MsgBus*. A generic message can be formulated as follows

$$\left\{A_{ID}^{FROM}, A_{ID}^{TO}, \Delta V_{PZC,z}, timeRef, msg\right\} \tag{12}$$

where A_{ID}^{FROM}, A_{ID}^{TO} represent the sending and receiving IDs for the agents, respectively, *timeRef* is a time reference, and *msg* is the message type (*ok?,good,etc.*). The *timeRef* parameter is analogous to the *Time* field as discussed in Section 3, and is crucial in ensuring that the *MsgBus* can filter out any outdated messages based on the reference point of time to prevent misaligned execution of the AWC. With this in mind, the remainder of the subsection will formally discuss the AWC implementation.

First: Each agent performs state estimation for its zone to determine the zonal voltage profile. Further discussion of the state estimation techniques are outside the scope of the paper, but can be found in [39]. The determination of the change in voltage needed to mitigate a voltage violation at bus i in zone z can be expressed as

$$\Delta V_{z,i} = \begin{cases} max\left(V_{z,i}(t) - V_{ub}\right), & \forall\, i \in N^z \,\&\, V_i > V_{ub} \\ min\left(V_{z,i}(t) - V_{lb}\right), & \forall\, i \in N^z \,\&\, V_i < V_{lb} \\ 0, & \forall\, i \in N^z \,\&\, V_{ub} \geq V_i \geq V_{lb} \end{cases} \tag{13}$$

Second: The agent predicts the magnitude of the voltage at the violated bus ($V_{z,i}^{pre}$) at the next timestep, determines a target for the required voltage adjustment needed ($\Delta V_{z,i}(t+1)$), and defines a tolerance for the target voltage deviation (ϵ) to account for any estimation error as follows:

$$V_{z,i}^{pre} = V_{z,i}(t) + \left(V_{z,i}(t) - V_{z,i}(t-1)\right) \tag{14}$$

$$\Delta V_{z,i}(t+1) = \begin{cases} max\left(V_{z,i}^{pre}(t+1) - V_{ub}\right) - \epsilon, & \forall\, i \in N^z \,\&\, V_i > V_{ub} \\ min\left(V_{z,i}^{pre}(t+1) - V_{lb}\right) + \epsilon, & \forall\, i \in N^z \,\&\, V_i < V_{lb} \end{cases} \tag{15}$$

Third: The agent calculates the assignment for its local control actions of $\Delta P_{BESS,z}$ and $\Delta Q_{BESS,z}$, considering the constraints given in (10)-(11) and the predicted active and reactive powers ($P_{BESS,z}^{Pre}$) and ($Q_{BESS,z}^{Pre}$) in the next step, respectively, as follows:

$$\begin{aligned} \Delta Q_{BESS,z}(t+1) \;=\; & \left(\left(P_{BESS,z}^{pre}(t+1) + \Delta P_{BESS,z,j}(t+1)\right)\right. \\ & \left.\times tan(cos^{-1}(PF_{min}))\right) - Q_{BESS,z}^{pre}(t+1) \end{aligned} \tag{16}$$

Fourth, If the local control action assignment violates any constraints, it can request assistance from neighboring agents. The violated agent calculates the required change in voltage at its PZC ($V_{PZC,z}$) with the neighboring agents that will resolve the voltage violation [40]:

$$V_{PZC,z}^{limit}(t+1) = V_{PZC,z}^{pre}(t+1) + \Delta V_{PZC,z}(t) \tag{17}$$

$$\Delta V_{PZC,z}(t) = \begin{cases} \dfrac{\partial V_{PZC,z}}{\partial V_{z,i}}\left(V_{ub} - V_{z,i}^{pre}(t+1)\right) - \epsilon, \forall\, V_{z,i} > V_{ub} \\ \dfrac{\partial V_{PZC,z}}{\partial V_{z,i}}\left(V_{z,i}^{pre}(t+1) - V_{lb}\right) + \epsilon, \forall\, V_{z,i} < V_{lb} \end{cases} \tag{18}$$

Subsequently, the violated agent sends its neighboring agent an *ok?* message that specifies the $\Delta V_{PZC,z}$ required to maintain the PZC voltage as a common constraint. Any agent request that cannot be satisfied by the neighboring agent will be passed on to *its* neighboring agent, based on the current voltage at their PZC.

Fifth, if the violated agent receives a *no-good* message from the neighboring agents, its last resort is to curtail the power from the DGs in its own zone, as follows:

$$\Delta P_{DG,z}(t+1) = \begin{cases} \dfrac{\partial P_{i,z}}{\partial V_{z,i}}\left(V_{ub} - V_{z,i}^{pre}(t+1)\right) \forall\, i \in N^z \,\&\, V_{z,i} > V_{ub} \\ \dfrac{\partial P_{i,z}}{\partial V_{z,i}}\left(V_{z,i}^{pre}(t+1) - (V_{lb})\right) \forall\, i \in N^z \,\&\, V_{z,i} < V_{lb} \end{cases} \tag{19}$$

Thus, the execution of the AWC control strategy is completed when all agents have processed all requests in their agent-view in accordance to all constraints, meaning that all voltage violations have been resolved. It is worth noting that if a particular DG or BESS within an agent's zone experiences failure, the sensitivity factors in (7) must be recalculated according to the new size and location of the unit, and the agent must take this into consideration as it evaluates the new constraints posed by the unit during the execution of the AWC control strategy.

5. Experiments and simulation results

This section presents the results of three case studies that evaluate the efficacy of the proposed DCF and its implementation as a SIL platform, including i) benchmarking the SIL platform in terms of average message latency against the recommended latency requirements of smart-grid applications [26], ii) prototyping the modified AWC strategy on a 39 bus test feeder, and iii) deployment of the DCF at a Canadian

microgrid to mitigate overvoltage violations.

5.1. Case study 1: Validation of DCF using SIL platform

The experimental methodology for benchmarking the SIL platform consists of three experiments that measure the average latency of messages exchanged using the *MsgBus* topic as a function of message rate (messages/second), message size (bytes/message), and the number of agents. At the start of each experiment, all agents are launched simultaneously, and publish messages of a configurable size to the *MsgBus* topic by specifying a random agent ID in the *Receiver_Agent_ID* field. Subsequently, the agents begin to receive messages from the *MsgBus* topic, which are timestamped and logged to disk. Since the messages are timestamped at both the sending/receiving ends, the message latency is calculating by subtracting the two measurements from each other. Monte Carlo experiments of 1000 trials are repeated for each experiment, with the average message latency being taken as the final result.

For comparison, the *MsgBus* topic is also implemented on two other popular IoT middlewares, including Advanced Message Queuing Protocol (AMQP) and Message Queuing Telemetry Transport (MQTT). As a result, two different agent platforms are used within the benchmarking process. The first agent platform consists of four National Instruments myRIO-1900 Linux-based real-time microcontrollers with a processing speed of 667 MHz and 256 MB RAM, while the second platform utilizes laptops running Windows 10 with an i7-2.60 GHz processor and 8 GB RAM. Both agent platforms are connected on a local area network by way of a 2.4 GHz wireless-N router (WiFi). The DCF is benchmarked on both agent platforms, denoted as DCF-RIO and DCF-PC, respectively, but both AMQP and MQTT could not be benchmarked on the myRIO platform because it does not support their native client libraries. Thus, the DCF-RIO benchmarks are included in the results to demonstrate how the DCF performs on resource constrained devices, while the PC results serve as a consistent platform to benchmark all three middlewares. Table 2 provides additional implementation details of the benchmarking procedure.

Experimental results are depicted in Figs. 8-10. Fig. 8 shows the result for message latency as a function of message rate, where the number of agents is fixed at 4 and the message size is fixed arbitrarily at 8192 bytes. The average message latency increases with the increase of the message rate for all three middlewares,

except for the DCF-PC implementation, where the recorded latency is consistently below 2 ms. Within the DCF-RIO implementation, the average message latency exceeds 10 ms only after reaching messaging rates of 1000 messages/second, which may also be attributable to the fact that the myRIO processor gives unstable performance at process rates above 1 kHz [41]. Nonetheless, the DCF-RIO implementation significantly outperforms the PC implementation of MQTT (faster by a factor of 5.2), and is at par with AMQP-PC (slower by a factor of 1.26) despite using a platform that has significantly less computational power. The DCF-PC implementation is faster than the AMQP-PC and MQTT-PC by a factor of 47.8 and 112.3, respectively. It is worthwhile to note that the benchmarks for all three frameworks satisfy the minimum latency requirements for smart grid applications, where the lowest specified latency recommendation is in the range of 15-200 ms [26]. However, the trend shown in Fig. 8 shows that the implementation of the DCF is consistently more scalable than AMQP or MQTT.

Fig. 9 shows the result of the second experiment, where the message rate is fixed at 10 messages/second and the number of agents is fixed at 4. The DCF-PC implementation shows superior performance over AMQP-PC and MQTT-PC by a factor of 1.18 and 5.67, respectively, and all three implementations show consistent performance in holding the average message latency to below 10 ms. The DCF-RIO delivers message latency below 10 ms until the message size reaches 16384 bytes, with the maximum latency observed at 27.9 ms for a message size of 32768 bytes. Given that distributed control strategies tend to limit the message size to reduce communication costs [35], a maximum message size of 8192 should suffice to ensure compliance with latency requirements in [26].

Fig. 10 shows the result of the third experiment, where the number of agents is increased while fixing the message size and rate at 256 bytes and 10 messages/second, respectively. Fig. 10 shows consistent performance of all middlewares on all platforms, with DDS-PC, DDS-RIO, and AMQP-PC recording average message latencies below 5 ms when 32 agents are deployed. In summary, the results of this case study show that the DCF-PC implementation is superior to both MQTT-PC and AMQP-PC implementations, and that the DCF-RIO implementation is still within the tolerable requirements for smart grid communication standards. It is worthwhile mentioning that the results obtained in this case study are closely aligned with another benchmarking evaluation done in [42], which also compares DDS, AMQP, and MQTT. In [42], benchmarks conducted on a computer with 1 GB RAM indicate that DDS has superior message latency at 1000 messages/second (20 ms) as compared to AMQP (40 ms), and MQTT (100 ms), with MQTT performing significantly worse when moving from 50 messages/second (10 ms) to 1000 messages/second (100 ms), which was also seen in this case study. When comparing the results in this case study to [42], it is noteworthy that the DCF-PC significantly outperforms the DDS benchmark by a factor of 10 (2 ms to 20 ms), while the DCF-RIO is slower by a factor of 1.8 (36ms to 20ms), even though the DCF-RIO has considerably less RAM than the computer used in [42] (256MB to 1GB).

Table 2
Software versions used in the benchmarking procedure.

	Client Library	Broker	Platform
AMQP	Python AMQP 2.5.2	RabbitMQ	Windows 10 PC
MQTT	Python Paho 1.5.0	Mosquitto	Windows 10 PC
DCF-PC	RTI DDS LabVIEW	-	Windows 10 PC
DCF-RIO	RTI DDS LabVIEW	-	Linux myRIO-1900

5.2. Case study 2: SIL simulation of voltage regulation scheme

In this case study, the SIL platform is used to validate the implementation of the modified AWC strategy in regulating the voltage of an ADN. A modified 39-bus ADN feeder is used as the test system [43], which is divided into 4 zones as seen in Fig. 11, where PV DG units of 1.0 MW capacity (PV2 and PV4) are deployed at buses 38 and 36, respectively, and BESS1-BESS4 are deployed at buses 34, 37, 35, and 39, respectively. BESS1, BESS2, and BESS4 have identical sizing, with a maximum capacity of 300 kWh, a minimum power factor of 0.89, and maximum apparent power of 0.3 MVA, while BESS3 has a slightly larger apparent power rating of 0.4 MVA. It is worth noting that the PV units are deployed at the end of the feeder, as is common in low/medium voltage distribution systems, to boost voltage levels at feeder end [44]. The location for the BESS units, on the other hand, were chosen arbitrarily. The upper bound for the voltage limit is set at 1.05 p.u. [43]. The simulation is carried out on the SIL platform as per Fig. 5, with the model agent being implemented in MATLAB to execute the ADN system model, while the agents are deployed on the myRIOs and implemented in LabVIEW.

In this case study, the PV generation results in overvoltage violations most prominently in Zone 4, and as such, Fig. 12 shows a plot of the voltage magnitude at the bus of PV4 under different scenarios, including the conventional (old) AWC, the modified AWC as implemented by the DCF (mod), as well as with no AWC. To support this figure, Figs. 13-15 show the resultant control actions from the agents in terms of their modulation of BESS active/reactive power and PV curtailment in order to resolve the overvoltage violations. Additionally, the issue of the DCF handling outdated messages when executing the AWC is highlighted in Fig. 16, which prevents misaligned execution and unnecessary energy loss.

As shown in Fig. 12, an overvoltage violation of 1.0534 p.u. is

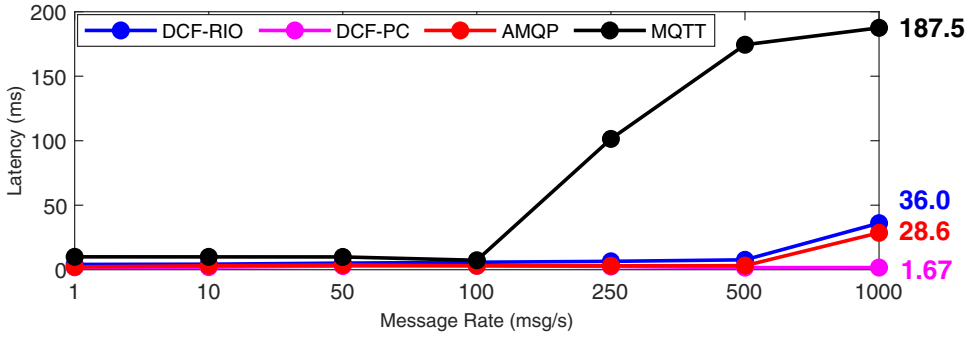


Fig. 8. Benchmark of message latency vs message rate.

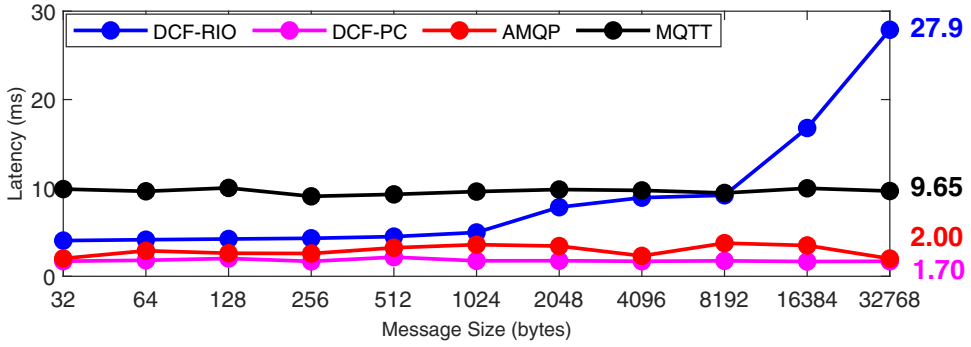


Fig. 9. Benchmark of message latency vs message size.

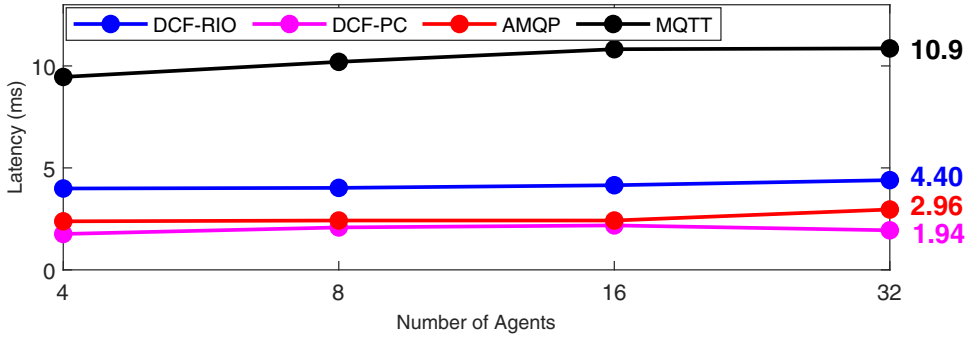


Fig. 10. Benchmark of message latency vs number of agents.

detected at the bus of PV4 by Agent 4 at timestep 2. Agent 4 reacts by setting a voltage target of 1.042 p.u. at the next timestep and uses (16) to find a solution that does not violate any of the agent constraints, which is $\Delta P_{BESS,4}$=0.171 MW and $\Delta Q_{BESS,4}$=0.0876 MVAR. A similar event occurs at timestep 4 in Zone 2, where Agent 2 detects the maximum voltage to be 1.506 p.u., and resolves the violation by adjusting $\Delta P_{BESS,2}$=0.13 MW and $\Delta Q_{BESS,2}$=0.066 MVAR. At timestep 6, both Agents 2 and 4 detect voltage violations of 1.0526 p.u. and 1.0586 p.u., respectively, but their BESS inverter capacities have been reached, thus requiring agents to utilize the AWC technique.

As shown in Fig. 16, Agents 2 and 4 calculate the ΔV_{PZC} required to mitigate the voltage violations using (18) at timestep 7, and send *ok?* messages to Agents 1 and 4, respectively, using the syntax defined in (12). Both agents store the requests in their agent-view, where Agent 1 determines a solution set of $\Delta P_{BESS,1}$=0.176 MW and $\Delta Q_{BESS,1}$=0.0902 MVAR that does not violate any constraints, and sends back a *good* message to Agent 2. Agent 3 evaluates both requests from Agent 2 and Agent 4, and according to the min-conflict heuristic, gives priority to Agent 4's request by sending back a *good* message after determining a solution set of $\Delta P_{BESS,3}$=0.184 MW and $\Delta Q_{BESS,3}$=0.0943 MVAR. In response, Agents 2 and 4 send back *execute* messages, to which Agent 1 responds, but Agent 3 does not respond since it suffers a disconnection from the network. Thus, at timestep 8, the violation still exists in Zone 4 as seen in Fig. 12, and Agent 4 sends another message to Agent 3 with a new target V_{PZC}, which again remains unanswered. With no answer from Agent 3, Agent 4 curtails its PV at timestep 8 by -0.068 MW to resolve the violation.

It as this point where the difference between the conventional AWC and the modified AWC can begun to be seen. As shown in Fig. 16, Agent 3 rejoins the network at timestep 9 and erroneously executes the outdated *execute* message sent by Agent 4 at timestep 7, even though there is no voltage violation. This results in unnecessary, misaligned execution of BESS3 as seen in Figs. 13-14. At timestep 10, both Agents 2 and 4 again detect overvoltage violations, and *ok?* messages are sent in the same fashion as timestep 7. Agent 3 again erroneously processes the *ok?* message from timestep 7, resulting in an inconsistent agent-view. Due to the unnecessary, misaligned execution of the control action of BESS3 at timestep 9, BESS3 has reached its maximum capacity and sends

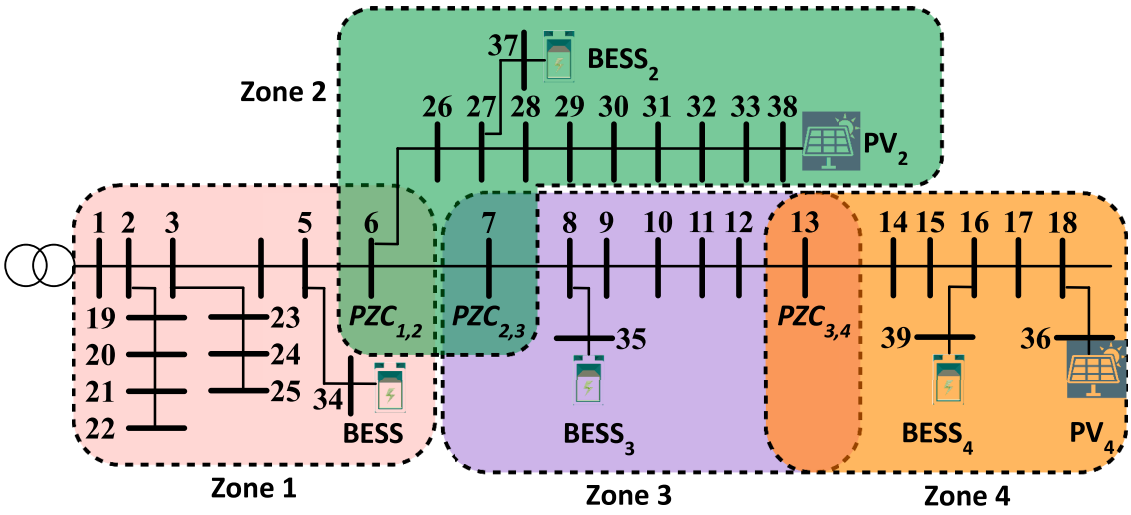


Fig. 11. The 39-bus test feeder divided into 4 zones.

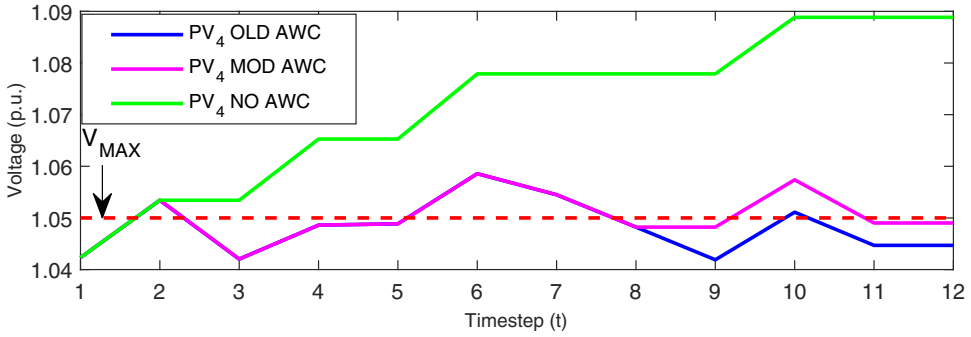


Fig. 12. PV_4 voltage under different AWC implementations.

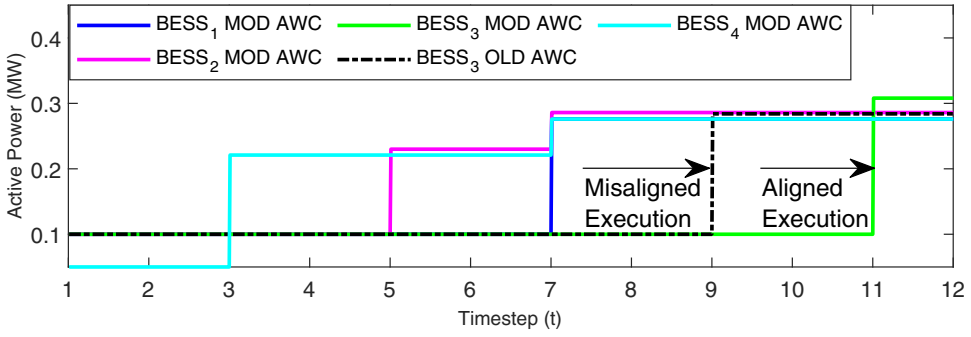


Fig. 13. BESS active power charging at each timestep.

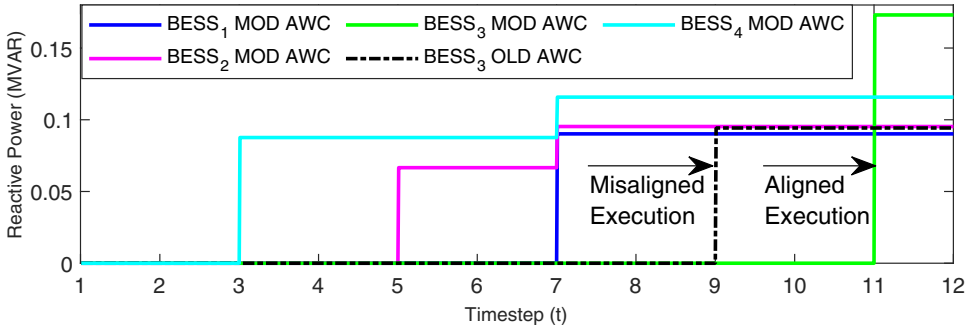


Fig. 14. BESS reactive power absorption at each timestep.

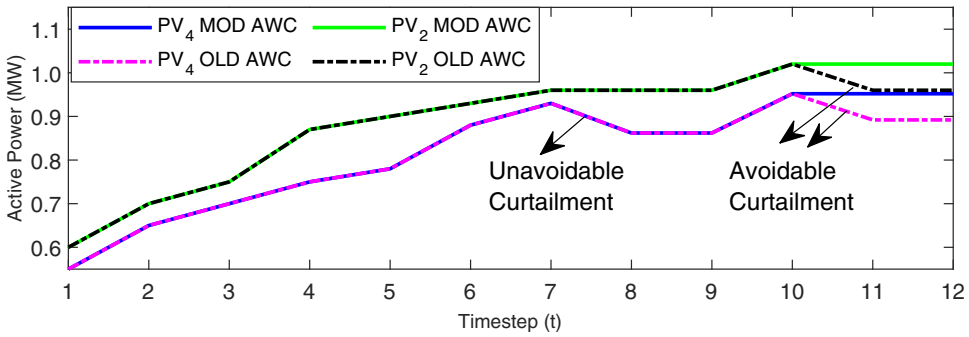


Fig. 15. PV power generation at each timestep.

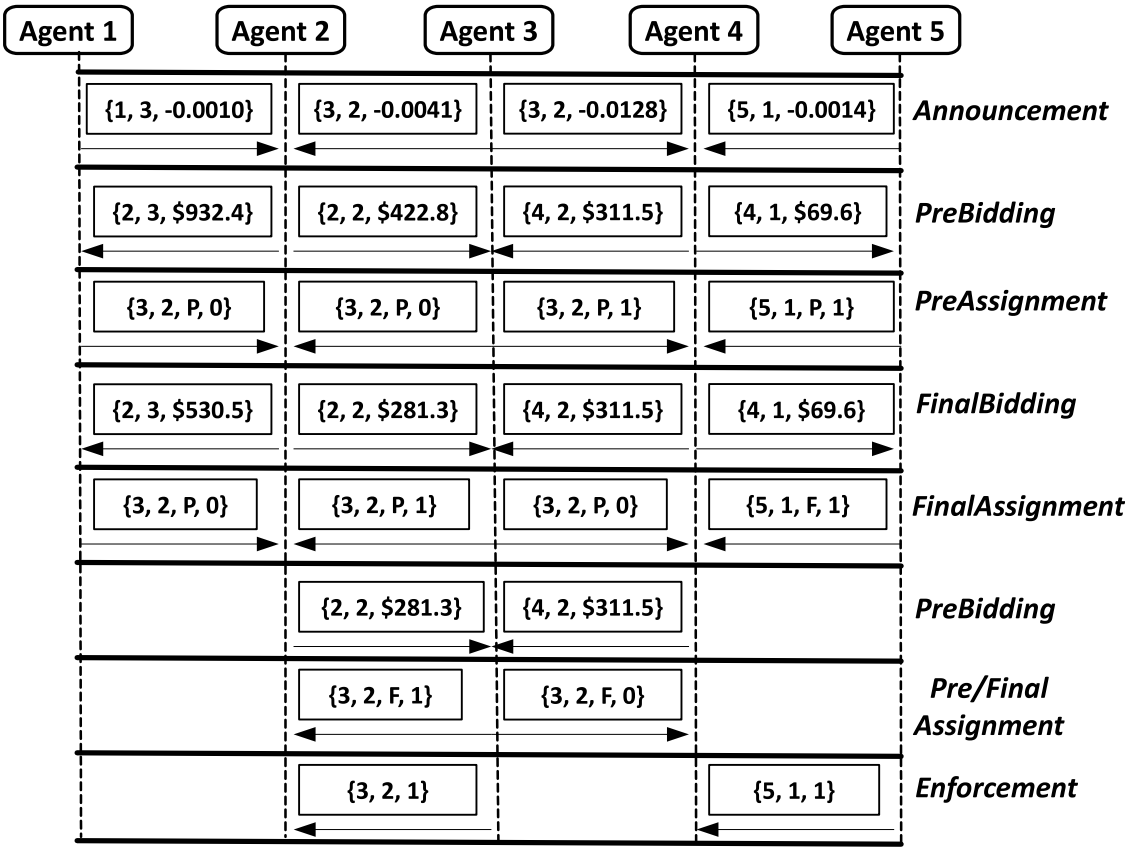


Fig. 16. Erroneous message exchange using conventional/old AWC.

back a *no-good* message to Agent 4. At this point, all other BESSs have also reached their maximum inverter capacity, resulting in the curtailment of both PV2 and PV4 to resolve the violation and unnecessary energy loss.

Within the modified AWC strategy implemented by the DCF, which has the capacity to automatically filter the outdated messages, Agent 3 *does not* process the *execute* message from Agent 4 at timestep 9, and therefore, does not unnecessarily execute any control actions. At timestep 11, Agent 3 receives the correct *ok?* message from Agent 4, and has sufficient capacity to resolve the voltage violations for both Agent 2 and Agent 4, thereby obviating the need to curtail PV power, as shown in Fig. 15.

5.3. Case study 3: Real world implementation

The DCF agents are implemented at the Kortright Centre Microgrid (KCM), located in Vaughan, Ontario, which has over 50 kW renewable power production capability and 75 kWh energy storage capacity. A simplified single line diagram for the facility is shown in Fig. 17, where Agent 1 controls the 20 kW SolarEdge 5000 smart inverter at bus 3, while Agent 2 controls the Schneider Xantrax 6848 XW+ BESS (6.8 kW/75 kWh) at bus 4. All control devices are controlled by the agents via the MODBUS protocol and the agents are deployed to a National Instruments PXI controller. The KCM experiences severe overvoltage violations due to a combination of very light loading and high renewable power production on its main feeder, and as such, the proposed DCF is deployed to mitigate these voltage issues.

The experimental result can be seen in Fig. 18, where the agents execute the modified AWC strategy to keep the voltage magnitude at bus 3 within the specified limits. Fig. 18 shows two major instances of overvoltage violations at times 13:03:30 and 13:03:50, where Agent 1 observes voltages of 1.052 p.u. and 1.053 p.u., respectively. In each instance, Agent 1 sends an *ok?* message to Agent 2, which receives the messages within the next second, evaluates the requests according to

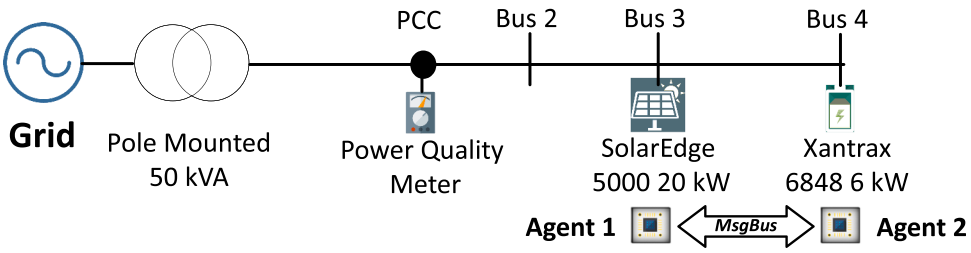


Fig. 17. Single line diagram of the Kortright Centre Microgrid.

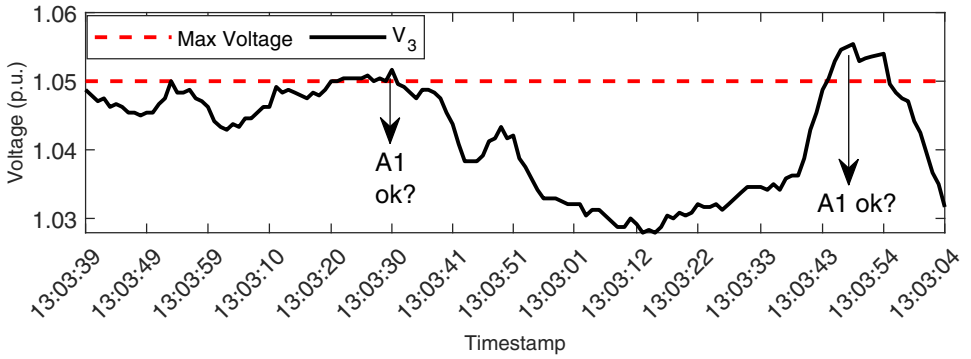


Fig. 18. Voltage profile at bus 3 during coordinated control.

(10), and sets a target of $\Delta P = +5$ kW (or 0.05 p.u.). After confirmation that the constraints have been satisfied, Agent 2 sends back an *good* message and begins to charge the BESS. The impact of these control actions can be seen in Fig. 19, which displays measurements of power demand at the PCC. After the *good* messages are sent back to Agent 1, the power demand of the microgrid increases by approximately 0.05 p.u., which equals the setpoint calculated by Agent 2. Fig. 19 also shows steady active power generation at bus 3, confirming that maximum active power generation is harvested during the experiment while regulating the voltage within permissible limits. The end-to-end mitigation time for both instances of voltage regulation is 5 seconds.

6. Conclusion and future work

This paper proposes a DDS-based DCF for MAS-based smart grid applications that provides real-time and robust communication between agents, thereby increasing the efficacy of distributed control strategies. A SIL platform is developed to prototype the DCF, and benchmarks indicate that the DCF provides significantly reduced message latency when compared to other middlewares, including AMQP and MQTT, and satisfies minimal latency requirements for smart grid applications. The DCF is subsequently applied to the smart grid application of distributed voltage regulation, which is formulated as a DisCSP and solved using the AWC technique. The DCF is used to increase the the efficacy of the conventional AWC technique by providing support for discarding outdated messages, thereby enabling agents to rejoin the network after a disconnection and align their control actions to prevent energy loss. Both simulation and real-world tests show the effectiveness of the DCF under different operational parameters, including the ability to resolve voltage violations in a timely manner.

In terms of future work, there is more scope to improve the robustness of the DCF by designing attack prevention strategies that mitigate attacks such as man-in-the-middle and denial of service [45]. In particular, the use of encryption is needed to protect the integrity of control messages sent between agents to avoid malicious tampering [46]. Thus, the intention to pursue this research direction will involve implementing baseline scenarios, wherein malicious agents intentionally tamper messages to change control actions, potentially discharging a BESS in an overvoltage violation when the original control message was to charge. Subsequently, security and encryption techniques will be implemented within the DCF to demonstrate how the tampering is obviated. Real-world testing will also be executed to determine if the added level of security impacted the message latency of the DCF in a negative manner.

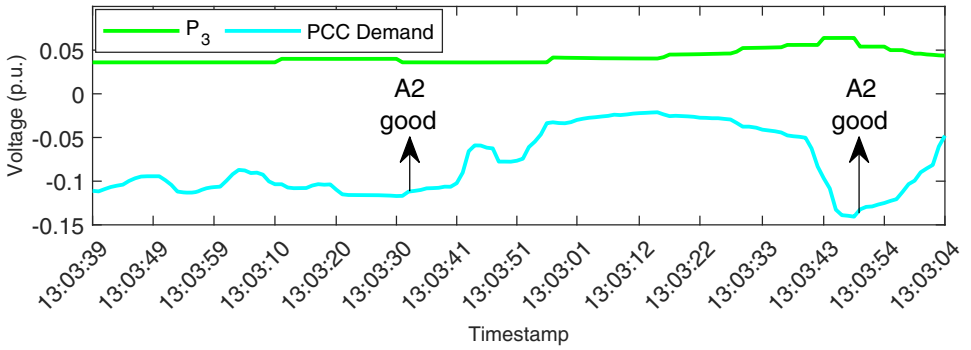


Fig. 19. BESS charging mitigates voltage violation.

CRediT authorship contribution statement

Shivam Saxena: Conceptualization, Software, Writing – original draft, Data curation. **Hany E.Z. Farag:** Writing – review & editing, Validation, Visualization. **Nader El-Taweel:** Data curation, Formal analysis, Methodology.

Declaration of Competing Interest

1. We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.
2. We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.
3. We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.
4. We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author. Signed by the corresponding author, as follows:

References

[1] D. Li, S.K. Jayaweera, Machine-learning aided optimal customer decisions for an interactive smart grid, IEEE Syst. J. 9 (4) (2015) 1529–1540.

[2] A.N. Samudrala, M.H. Amini, S. Kar, R.S. Blum, Distributed outage detection in power distribution networks, IEEE Trans. Smart Grid 11 (6) (2020) 5124–5137.

[3] M. Hasanuzzaman Shawon, S.M. Muyeen, A. Ghosh, S.M. Islam, M.S. Baptista, Multi-agent systems in ict enabled smart grid: a status update on technology framework and applications, IEEE Access 7 (2019) 959–973.

[4] M. Nazari-Heris, B. Mohammadi-Ivatloo, G.B. Gharehpetian, M. Shahidehpour, Robust short-term scheduling of integrated heat and power microgrids, IEEE Syst. J. 13 (3) (2019) 3295–3303, https://doi.org/10.1109/JSYST.2018.2837224.

[5] M. Nazari-Heris, S. Abapour, B. Mohammadi-Ivatloo, Optimal economic dispatch of fc-chp based heat and power micro-grids, Appl. Thermal Eng. 114 (2017) 756–769, https://doi.org/10.1016/j.applthermaleng.2016.12.016.

[6] C. Dou, D. Yue, J.M. Guerrero, X. Xie, S. Hu, Multiagent system-based distributed coordinated control for radial dc microgrid considering transmission time delays, IEEE Trans. Smart Grid 8 (5) (2017) 2370–2381, https://doi.org/10.1109/TSG.2016.2524688.

[7] F. Perkonigg, D. Brujic, M. Ristic, Platform for multiagent application development incorporating accurate communications modeling, IEEE Trans. Ind. Inf. 11 (3) (2015) 728–736, https://doi.org/10.1109/TII.2015.2428633.

[8] V.P. Singh, N. Kishor, P. Samuel, Distributed multi-agent system-based load frequency control for multi-area power system in smart grid, IEEE Trans. Ind. Electron. 64 (6) (2017) 5151–5160, https://doi.org/10.1109/TIE.2017.2668983.

[9] C.P. Nguyen, A.J. Flueck, Modeling of communication latency in smart grid. 2011 IEEE Power and Energy Society General Meeting, 2011, pp. 1–7, https://doi.org/10.1109/PES.2011.6039815.

[10] M.P. Singh, A.K. Chopra, The internet of things and multiagent systems: decentralized intelligence in distributed computing. 2017 IEEE Conference on Distributed Computing Systems, 2017, pp. 1738–1747, https://doi.org/10.1109/ICDCS.2017.304.

[11] H.S.V.S.K. Nunna, S. Doolla, Demand response in smart distribution system with multiple microgrids, IEEE Trans. Smart Grid 3 (4) (2012) 1641–1649, https://doi.org/10.1109/TSG.2012.2208658.

[12] A. Elmitwally, M. Elsaid, M. Elgamal, Z. Chen, A fuzzy-multiagent self-healing scheme for a distribution system with distributed generations, IEEE Trans. Power Syst. 30 (5) (2015) 2612–2622, https://doi.org/10.1109/TPWRS.2014.2366072.

[13] W. Li, Y. Li, C. Chen, Y. Tan, Y. Cao, M. Zhang, Y. Peng, S. Chen, A full decentralized multi-agent service restoration for distribution network with DGs, IEEE Trans. Smart Grid 11 (2) (2019) 1100–1111, https://doi.org/10.1109/TSG.2019.2932009.

[14] A. Abel Hafez, W.A. Omran, Y.G. Hegazy, A decentralized technique for autonomous service restoration in active radial distribution networks, IEEE Trans. Smart Grid 9 (3) (2018) 1911–1919, https://doi.org/10.1109/TSG.2016.2602541.

[15] S. Bashir, M. Rehman, H.F. Ahmad, A. Ali, H. Suguri, Distributed and scalable message transport service for high performance multi-agent systems. 2004 International Networking and Communication Conference, 2004, pp. 152–157, https://doi.org/10.1109/INCC.2004.1366596.

[16] J.M. Schlesselman, G. Pardo-Castellote, B. Farabaugh, OMG data-distribution service (DDS): architectural update. IEEE MILCOM 2004., 2004, pp. 961–967, https://doi.org/10.1109/MILCOM.2004.1494965.

[17] B. AL-Madani, H. Ali, Data distribution service (dds) based implementation of smart grid devices using ansi c12.19 standard, Procedia Comput. Sci. 110 (2017) 394–401, https://doi.org/10.1016/j.procs.2017.06.082.14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017) / 12th International Conference on Future Networks and Communications (FNC 2017) / Affiliated Workshops

[18] M. Cintuglu, A. Kondabathini, D. Ishchenko, Real-time implementation of secure distributed state estimation for networked microgrids. 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), 2020, pp. 1–6, https://doi.org/10.1109/WF-IoT48130.2020.9221310.

[19] T. Agarwal, P. Niknejad, M.R. Barzegaran, L. Vanfretti, Multi-level time-sensitive networking (tsn) using the data distribution services (dds) for synchronized three-phase measurement data transfer, IEEE Access 7 (2019) 131407–131417, https://doi.org/10.1109/ACCESS.2019.2939497.

[20] H.F. Habib, M.M. Esfahani, O.A. Mohammed, Investigation of protection strategy for microgrid system using lithium-ion battery during islanding, IEEE Trans. Ind. Appl. 55 (4) (2019) 3411–3420, https://doi.org/10.1109/TIA.2019.2904566.

[21] M.M. Esfahani, A. Hariri, O.A. Mohammed, A multiagent-based game-theoretic and optimization approach for market operation of multimicrogrid systems, IEEE Trans. Ind. Inf. 15 (1) (2019) 280–292, https://doi.org/10.1109/TII.2018.2808183.

[22] T.A. Youssef, M.E. Hariri, A.T. Elsayed, O.A. Mohammed, A dds-based energy management framework for small microgrid operation and control, IEEE Trans. Ind. Inf. 14 (3) (2018) 958–968, https://doi.org/10.1109/TII.2017.2756619.

[23] M. Starke, A. Herron, D. King, Y. Xue, Implementation of a publish-subscribe protocol in microgrid islanding and resynchronization with self-discovery, IEEE Trans. Smart Grid 10 (1) (2019) 361–370, https://doi.org/10.1109/TSG.2017.2739246.

[24] M.H. Cintuglu, T. Youssef, O.A. Mohammed, Development and application of a real-time testbed for multiagent system interoperability: a case study on hierarchical microgrid control, IEEE Trans. Smart Grid 9 (3) (2018) 1759–1768, https://doi.org/10.1109/TSG.2016.2599265.

[25] P. Charlton, R. Cattoni, A. Potrich, E. Mamdani, Evaluating the fipa standards and their role in achieving cooperation in multi-agent systems. 33rd Annual Hawaii International Conference on System Sciences, 2000.

[26] T. Sato, Smart grid standards: specifications, requirements, and technologies, Wiley, 2015.

[27] M. Yokoo, E.H. Durfee, T. Ishida, K. Kuwabara, The distributed constraint satisfaction problem: formalization and algorithms, IEEE Trans. Knowl. Data Eng. 10 (5) (1998) 673–685, https://doi.org/10.1109/69.729707.

[28] J. Dizdarevi, F. Carpio, A. Jukan, X. Masip-Bruin, A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration, ACM Comput. Surv. 51 (6) (2019) 116:1–116:29, https://doi.org/10.1145/3292674.

[29] M. Albano, L.L. Ferreira, L.M. Pinho, A.R. Alkhawaja, Message-oriented middleware for smart grids, Comput. Stand. Interfaces 38 (2015) 133–143.

[30] N. Naik, Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. 2017 IEEE International Systems Engineering Symposium (ISSE), 2017, pp. 1–7.

[31] T. Agarwal, P. Niknejad, M.R. Barzegaran, L. Vanfretti, Multi-level time-sensitive networking (tsn) using the data distribution services (dds) for synchronized three-phase measurement data transfer, IEEE Access 7 (2019) 131407–131417, https://doi.org/10.1109/ACCESS.2019.2939497.

[32] S.D.J. McArthur, E.M. Davidson, V.M. Catterson, A.L. Dimeas, N.D. Hatziargyriou, F. Ponci, T. Funabashi, Multi-agent systems for power engineering applicationspart i: concepts, approaches, and technical challenges, IEEE Trans. Power Syst. 22 (4) (2007) 1743–1752.

[33] A. Sargolzaei, K. Yazdani, A. Abbaspour, C.D. Crane III, W.E. Dixon, Detection and mitigation of false data injection attacks in networked control systems, IEEE Trans. Ind. Inf. 16 (6) (2020) 4281–4292, https://doi.org/10.1109/TII.2019.2952067.

[34] P. Stone, M. Veloso, Multiagent systems: a survey from a machine learning perspective, Autonomous Robots 8 (2000), https://doi.org/10.1023/A:1008942012299.

[35] N.A. El-Taweel, H.E.Z. Farag, Voltage regulation in islanded microgrids using distributed constraint satisfaction, IEEE Trans. Smart Grid 9 (3) (2018) 1613–1625, https://doi.org/10.1109/TSG.2016.2596098.

[36] X. Tong, M. Zhong, X. Zhang, J. Deng, Z. Zhang, Voltage regulation strategy of ac distribution network based on distributed pv grid-connected inverter, J. Eng. 2019 (2018), https://doi.org/10.1049/joe.2018.8680.

[37] M.M.A. Abdelaziz, H.E. Farag, E.F. El-Saadany, Y.A..I. Mohamed, A globally convergent trust-region method for power flow studies in active distribution systems. 2012 IEEE PES General Meeting, 2012, pp. 1–7, https://doi.org/10.1109/PESGM.2012.6344971.

[38] M. Nazari-Heris, B. Mohammadi-Ivatloo, G. Gharehpetian, A comprehensive review of heuristic optimization algorithms for optimal combined heat and power dispatch from economic and environmental perspectives, Renew. Sustain. Energy Rev. 81 (2018) 2128–2143, https://doi.org/10.1016/j.rser.2017.06.024.
[39] M. Rostami, S. Lotfifard, Distributed dynamic state estimation of power systems, IEEE Trans. Ind. Inf. 14 (8) (2018) 3395–3404.
[40] N.A. El-Taweel, H.E.Z. Farag, Operation challenges of feeder shunt capacitors in islanded microgrids. 2015 IEEE EPEC, 2015, pp. 191–196, https://doi.org/10.1109/EPEC.2015.7379948.
[41] Ni scan engine performance benchmarks, 2020, (????).
[42] A. Talaminos-Barroso, M.A. Estudillo-Valderrama, L.M. Roa, J. Reina-Tosina, F. Ortega-Ruiz, A machine-to-machine protocol benchmark for ehealth applications, Comput. Method. Program. Biomed. 129 (2016) 1–11, https://doi.org/10.1016/j.cmpb.2016.03.004.
[43] N.A. El-Taweel, H.E. Farag, Multi-agent coordination of distributed energy storage systems for mitigating renewable energy resources high ramp-rate issues. 2017 IEEE EPEC, 2017, pp. 1–6, https://doi.org/10.1109/EPEC.2017.8286199.
[44] E. Karunarathne, J. Pasupuleti, J. Ekanayake, D. Almeida, The optimal placement and sizing of distributed generation in an active distribution network with several soft open points, Energies 14 (4) (2021), https://doi.org/10.3390/en14041084.
[45] M. Conti, N. Dragoni, V. Lesyk, A survey of man in the middle attacks, IEEE Commun. Surv. Tutor. 18 (3) (2016) 2027–2051, https://doi.org/10.1109/COMST.2016.2548426.
[46] U. Meyer, S. Wetzel, On the impact of gsm encryption and man-in-the-middle attacks on the security of interoperating gsm/umts networks. 2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No.04TH8754) 4, 2004, pp. 2876–2883Vol.4, https://doi.org/10.1109/PIMRC.2004.1368846.