Dynamic Multi-agent OpenADR Virtual Nodes for Distributed Demand Response Schemes

Christos Patsonakis, Apostolos C. Tsolakis, *Member, IEEE,*, Dimosthenis Ioannidis,
and Dimitrios Tzovaras, *Senior Member, IEEE*
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
{cpatsonakis, tsolakis, djoannid, Dimitrios.Tzovaras}@iti.gr

***Abstract*—Demand Response (DR) has been described as one of the few solutions available for accessing the untapped energy potential of small/medium customers and especially of prosumers, towards supporting grid and DR reliability to a smart distribution grid. Investing on that notion, this manuscript employs virtual power plant (VPP) principles to deliver a novel, distributed, multi-agent virtual node scheme that aggregates consumers and prosumers and handles automatically OpenADR compliant DR requests. The introduced virtual node can act both as a virtual end node (VEN) receiving signals from a higher level (i.e., aggregator or utility) and/or as a virtual top node (VTN), dispatching optimal DR requests to customers. The proposed framework increases the DR results by not only optimally exploiting aggregated flexibility, but also through a dynamic bi-directional DR matchmaking process that can either recover monitored deviations within the virtual node (intra), or in-between nodes (inter) in real-time operation. The proposed design aims to elevate DR effectiveness, while enabling the participation of small to medium customers to energy markets.**

***Index Terms*—Demand Response, OpenADR, Virtual Node, Matchmaking, Multi-agent**

I. Introduction

Demand response (DR) programs have become a valuable asset for energy grid operators for mitigating energy shortage or excess, hence, increasing the overall system's reliability [1]. This allows to create/maximise profits and minimise risks, especially given the ever-growing penetration of Renewable Energy Sources (RES) [2], energy storage systems (ESS) [3] and electric vehicles [4]. Building on these changes, a trend has been observed in the energy markets towards different business models revolving around, e.g., aggregation and virtual power plants (VPPs) [5], both of which are heavily dependent on Information and Communication Technology (ICT) solutions for Demand Side Management (DSM) strategies, such as DR.

The efficient coordination of heterogeneous, distributed energy resourses (DERs) can improve, not only the overall economics of aggregators and retailers, but also for individual units and customers, while simultaneously reducing transaction costs and transmission/distribution losses. This coordination, through DR, allows to adjust consumption and generation patterns and tap into new, currently unexploited, flexibility at smaller scales. Nevertheless, in most cases around Europe where DR is already applicable, mainly industrial customers are offered the chance to participate in such markets, due to the lack of an ICT infrastructure that can handle small to medium residential and tertiary customers at large scales. However, the latter have been identified to have a significant untapped potential for DR schemes [6], a fact that is supported even more by the introduction of prosumers (consumers that also produce energy). Just by 2016, more than 33GW of residential solar PVs had been installed in the EU Member States, out of which 53% is exported to the grid [7]. This is expected to rise even more due to the EU's optimistic targets that aim towards a 32% RES share by 2030 [8].

To accelerate the uptake of DR programs in current energy markets, several challenges have to be addressed, one of the most important of which is interoperability [9]. Currently, there is a large discrepancy among the communication protocols employed by, e.g., DERs and the various energy stakeholders that are involved in energy markets. To mitigate the risk of creating yet again another data model, a well known open protocol, OpenADR [10], has been selected for investing upon and presenting an interoperable framework for smart grid information exchange among energy stakeholders and end-users.

The integration of small customers in energy markets introduces, however, additional challenges. First, there is a scalability issue, i.e., the more customers, the higher the computation and communication overhead required to unobtrusively and efficiently handle DR events that involve them. Second, there is a DR reliability issue, i.e., the smaller the capacity of the involved customers, the higher the uncertainty of achieving the DR event's requested result. To handle the sheer scale of smart meters, IoT devices and DERs brought by the integration of small customers, there is a need to shift to decentralized/localized monitoring and control schemes. Multi-Agent systems (MAS) pose as one of the most promising approaches in handling the integration of small and medium customers due to their inherent properties of autonomy, decentralization and locality regarding data management and decision making [11]. Although extensively researched with numerous scientific results, there are quite limited findings for MAS that tackle

This work is partially funded by the European Union's Horizon 2020 Research and Innovation Programme through DELTA project under Grant Agreement No. 773960.

DR reliability in terms of unsuccessful DR due to system estimation deviations or end-user behaviour. Even in cases where a reliability index is assigned to customers [12], this is handled proactively and not in real-time operation to adjust DR requests and deliver the expected outcome.

In this paper, the design and implementation of a novel MAS that aims to diminish the roadblocks that hinder the integration of small and medium customers in energy markets is presented. The proposed design employs a scalable and modular architecture that introduces a virtual node-based agent layer. Each agent is responsible for managing a (dynamic) set of customers whose individual resources are pooled together and exposed to the outside world as a unified flexible asset. Via a lightweight optimization engine, each agent, on input an OpenADR event, optimally dispatches OpenADR requests to its assigned customers. To provide for increased reliability in achieving the DR event's requested result, the designed agents are able to adapt to unforeseen failures and deviations during the active period of a DR event via a twofold approach. First, via an "Intra Matchmaking" algorithm that utilizes other available customers that the agent manages to make up for the deviation. Second, in cases where the previous approach fails, the agent employs an "Inter Matchmaking" algorithm which, in short, involves other fellow agents to be able to make up for the incurred deviation. This twofold approach is a first step in mitigating the risk of integrating small and medium customers in energy markets and, ultimately, delivering the requested amount of power/energy.

The manuscript is organized as follows: Section II introduces fundamental concepts regarding OpenADR. Section III presents previous research endeavours related to the the submitted work highlighting limitations and challenges identified. In Section IV, the overall system architecture is presented, followed by the detailed implementation for the aspects explored in Section VI.

II. Background: OpenADR

Open Automated Demand Response (OpenADR [10]) is an open and interoperable communication standard that facilitates smart grid information exchange among various energy stakeholders and end-users. OpenADR has gained worldwide acceptance with deployments around the world and has demonstrated to provide significant results for, e.g., average peak load reduction [13]. Its most recent version, OpenADR 2.0b, allows for automated DR actions and provides for a more generalized technical framework of communicating price and reliability signals. OpenADR partitions participating nodes into two types: 1) Virtual Top Nodes (VTNs) and, 2) Virtual End Nodes (VENs). VTNs publish information related to grid conditions (e.g., reliability events) to entities that control demand side resources. VENs are clients that have operational control over resources and are able to shift their energy demand in response to VTN requests. As an example, a VTN announces a DR event, which is received and responded to by VENs. The request may involve to reduce power to some devices or systems in the VEN's domain. We stress that a node can be a VTN and a VEN at the same time, which is relevant to the virtual node-based agent system that is presented in this work.

OpenADR signals are published via standard Internet application protocols, such as the Hyper Text Transfer Protocol (HTTP) or the XML Message and Presence Protocol (XMPP). Below, we briefly introduce the main services of OpenADR:

- **EiRegisterParty:** Used to identify entities, such as VENs, and is necessary to be invoked in advance of an actor interacting with other parties, such as VTNs.
- **EiReport:** Publish periodic or one-time information regarding the state of a resource.
- **EiOpt:** Declare and communicate the availability of a VEN to a VTN.
- **EiEvent:** Publish DR events containing signals that provide information about, e.g., the amount of energy curtailment requested or updates on electricity price. VENs typically confirm or reject participation in an event.

III. Related Work

Multi agent systems have been extensively employed for offering enhanced grid reliability in control systems [14]. When focusing on DR schemes, MAS-based implementations have been selected towards optimizing the communication among the various engaged stakeholders, while also distributing (partially in most cases) the computation requirements for decision making processes [11]. These attributes have led to the adoption of such architectures for DR services. In one of the very first deployments of MAS for DR [15], the focus was already shifted towards the effective interaction with the markets (market-based agent schemes [16]), but without analysing the actual delivered power/energy. Similarly, in a recent effort to combine industrial and residential customers, Golmohamadi et al. [17] presented a centralised MAS that optimises flexibility trading based on dedicated market agents that take into account customer processes and needs.

Elaborating more on the actual customer capacity (non-market-based), Karfopoulos et al. [18] have demonstrated that MAS-based DR can adequately support the participation of individual households in DR services. The authors also highlight the need for a highly dynamic decision logic due to both the volatile nature of RES generation, as well as, the non-deterministic end-user behaviour.

On an entirely different approach, Muthirayan et al. [19], to deal with agents that intentionally inflate their baseline so as to receive better market prices, propose frequent reporting from every agent in terms of baseline consumption and marginal utility forecasts. Even though this facilitates creating actual benchmark values for assessing the customer's reliability, it does not address the inherent problem of failed DR events, which are of course penalized. Towards that direction, recent research has worked towards providing accurate forecast models for consumption and generation, but also for prices and flexibility [20]. Emphasizing even more on the limitations introduced by uncertain factors, such as RES volatility, end-user decisions or even poor real-time monitoring of customers' assets [21], we have identified that in none of the above findings the failure of delivering the requested demand has been examined in real time operation for maximizing the system's overall DR reliability.

IV. MAS Architecture

In this section, the architecture of our novel, automated and distributed MAS for servicing OpenADR requests is presented. The proposed system's main innovation revolves around the following principles. First, it harnesses the untapped potential of small and medium customers (consumers and prosumers), thus, increasing the overall flexibility of the electricity grid. Second, it employs dynamic matchmaking algorithms to provide for increased DR reliability, fault-tolerance and delivery of energy in DR schemes.

The importance of fault-tolerance is highlighted as it allows energy stakeholders (e.g., aggregators and utilities) to incorporate low and medium voltage customers in their DR portfolios, while also maintaining a high-level of service reliability and availability. To achieve these, our MAS employs various ICT, such as peer-to-peer (P2P) networks, machine learning, linear programming solvers and asynchronous, event-driven system design principles. However, the focus of this work is to present how our MAS monitors and, in the presence of failures, or even deviations from the DR event's target, adapts either by employing its internal resources ("Intra Matchmaking"), or by coordinating with other agents ("Inter Matchmaking").

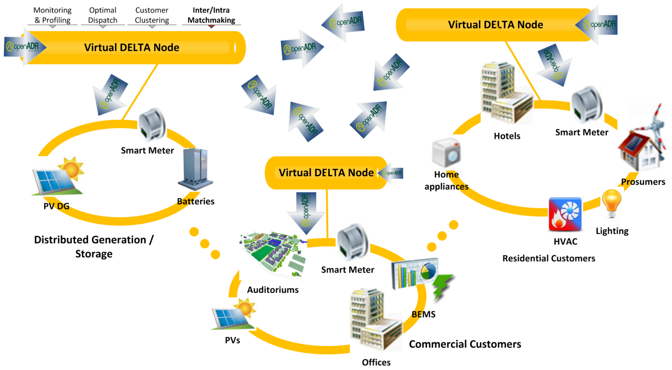

Fig. 1. Architectural overview of our distributed, virtual node-based MAS.

Figure 1 presents a high-level overview of the proposed system's architecture. It is comprised by a set of agents, which are referred to as virtual nodes (these terms are used interchangeably). As illustrated, each agent is responsible for monitoring and managing a dynamic set of customers, ranging from simple consumers, who are equipped only with, e.g., smart meters, to prosumers that are equipped with photovoltaics (PVs), batteries and/or other DERs. In the following subsections, the building blocks that comprise our MAS are introduced, i.e., the agent P2P network, which is the key enabler for, e.g., our matchmaking algorithms, as well as, the functionalities of the components that comprise each agent.

A. Agent Peer-to-Peer (P2P) Network

A fundamental requirement of any MAS is the establishment of a communications network that allows its participating agents to exchange data. Depending on their degree of centralization, communication networks can be classified in the following categories:

- **Centralized:** Information exchange takes place through a single centralized server. These networks are inherently limited in regards to scalability and, evidently, the centralized server constitutes a single point of failure.
- **Distributed:** All involved communications are point-to-point among agents who collectively share their communication links to facilitate message relay. One of the most prominent means of building and maintaining distributed P2P networks are structured overlays, such as distributed hash tables (e.g., [22]).
- **Hybrid:** Multiple centralized servers collectively manage and facilitate communications among agents. This approach addresses the single point of failure issue of the centralized case via standard redundancy techniques and provides for scalability, as communication load is managed and balanced among server replicas.

One of the main design principles of the presented MAS is its ability to scale and provide for service liveness, even in the presence of failures. Hence, the centralized approach is inappropriate. However, as it is required for the agents to be OpenADR compliant, this introduces a set of security requirements, which stem from OpenADR's compliance rules. Moreover, as the focus is to provide a system suitable for practical deployment, it is needed to acknowledge additional security constraints that aggregators and utilities are subject to, such as data privacy. To this end, a hybrid P2P network approach is employed. From a technical standpoint, OpenFire [23] is employed as the communication broker, which allows for both HTTP and XMPP communications, the latter of which is a requirement for OpenADR-compliant VTNs. OpenFire's builtin security features, such as support for Transport Layer Security (TLS [24]) and strong digital identities based on the X.509 standard [25], provide the necessary security and privacy features for practical deployments [26]. To provide for scalability, decentralization and fault-tolerance, a cluster of multiple OpenFire servers is created. This cluster behaves as one for its participating clients and communications are load balanced across the cluster's servers.

B. Agent Architecture

In this section, the functionalities of the individual components that comprise the agents of the proposed MAS (Figure 1) are introduced. One of the focal points of each agent is the "Monitoring and Profiling" component, whose functionalities are as follows. First, this is the entry point for new customers, i.e., it realizes OpenADR's "EiRegisterParty" service. The registration of a new customer is accompanied with data pertaining to her consumption, generation and storage capacities, geolocation, the market contexts that she is willing to be involved, and others. Second, this component is responsible for collecting and maintaining data pertaining to historical and forecasted consumption, generation, power flow and storage of its assigned customers (OpenADR's "EiReport" service). These data are used as the basis of building a dynamic, per customer profile that encompasses metrics pertaining to the accuracy of the reported forecasts, as well as, others that capture the reliability of the customer in the context of each individual market. The reliability metrics are output/updated by an in-house, under development, machine learning model that incorporates features relevant to each specific marketplace. Lastly, this component, based on the measurements and forecasts reported by its assigned customers, exposes aggregated historical and forecasted data, such as the agent's forecasted flexibility. These aggregated metrics can be queried by other fellow agents and play a pivotal role in the "Inter Matchmaking" algorithm, which we describe in Section V.

Agents input customer measurements and profiles to their "Customer Clustering" component to cluster their assigned customers based on various features, such as geolocation, type (consumer or prosumer) and their capacities. For each cluster, the agent produces a profile, which is directly involved in the decision making process for servicing input DR events. As the system is by nature dynamic, i.e., measurements and forecasts, as well as, customer response and reliability in the context of DR events may change over time, customers may be reassigned from cluster to cluster.

The "Optimal Dispatch" component is the focal decision point for servicing input DR events. This is a lightweight optimisation engine for consumers, prosumers and various DERs that are integrated into a unified optimisation problem and can cover multiple market contexts. Its objective function revolves around the minimization of the energy cost (based on the unit commitment problem) by taking into account parameters, such as the system marginal price (SMP), the profiles of customer clusters, the virtual cost of their flexibilities, and others. Put simply, this component, on input an OpenADR request, outputs a set of optimal OpenADR requests, which are dispatched to selected customers via OpenADR's "EiEvent" service.

V. DR Monitoring & Matchmaking

In this section, details are provided regarding the algorithmic flow of each agent during the active period of a DR event. The conditions under which each matchmaking algorithm is invoked, as well as, their involved (computational) steps are elaborated. To facilitate reader understanding and provide a concise description, a simple OpenADR load dispatch signal request is examined. More precisely, in the case study described, the active period of the input DR request is comprised by a series of time intervals, for each of which, a fixed powerflow setpoint is requested.

Figure 2 illustrates the algorithmic flow of the agent when the active period of the DR event begins. As illustrated, the agent first invokes its "Monitoring and Profiling" component. On a high-level, there are two constraints that should always be met. First and foremost, the individual powerflow measurement reports of the customers that are directly involved in the servicing of the DR event should comply to the setpoints that were output by the agent's "Optimal Dispatch" component. Second, the aggregated powerflow of the agent should comply to the setpoint of the initial DR request. Clearly, the second invariant involves also monitoring the powerflow reports of customers that are assigned to the agent, but are not involved in the DR request itself directly. If none of the aforementioned constraints is violated during the active period of the DR event, as illustrated in Figure 2, the agent will update the reliabilities of the VENs (customers) that were directly involved in the DR event, issue a successful completion report to the requesting VTN and terminate the monitoring process.

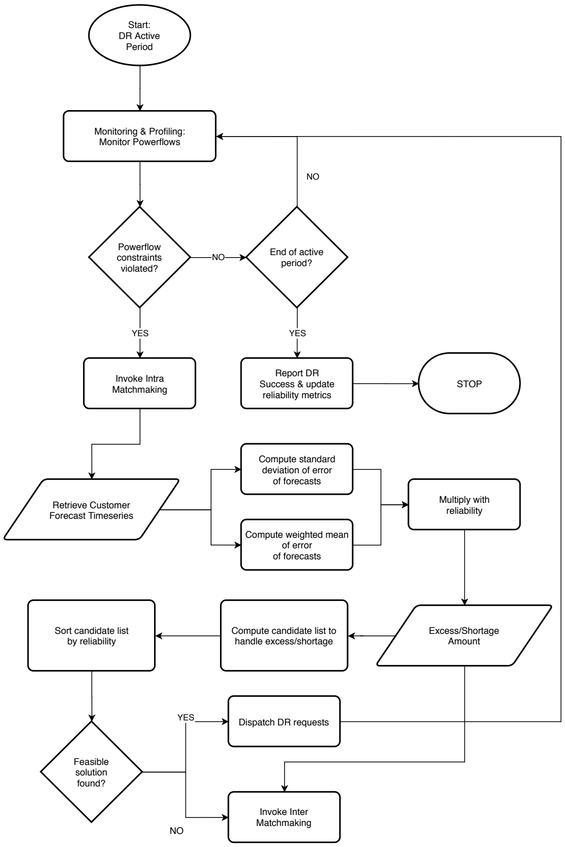

Fig. 2. Agent DR monitoring and Intra matchmaking flow diagram.

In cases where at least one of the powerflow constraints is violated, the agent will first invoke the "Intra" branch of the matchmaking algorithm, i.e., it first attempts to utilize the other customers that it manages to make up for the excess/shortage. Mitigating failures during the active period of a DR event imposes, in addition, the following time-related constraints. First, as OpenADR allows for intervals that lie in the order of seconds, we must employ a procedure that, apart from accurate, is computationally lightweight. Second, there needs to be enough time for the agent to engage in OpenADR's "EiEvent" service protocol with the candidates that will cover the excess/shortage.

To address these time-related constraints for the "Intra Matchmaking" case, a simple strategy is employed to calculate the weighted mean ($\bar{f^i}$) and standard deviation ($\sigma_{\bar{f^i}}$) of the error of the deviating customer's forecasted timeseries (f_t^i). Based on the calculated errors, we adapt these timeseries, for each time interval t, by multiplying them with the customer's reliability (r_i), as illustrated by the following equation:

$$f_t^{i'} = \frac{f_t^i - \bar{f^i}}{\sigma_{\bar{f^i}}} * r_i, \tag{1}$$

where i is the deviating customer.

Next, the agent evaluates whether there are other customers ($j \neq i, \forall j$) that can deliver, for each time interval t, the computed excess/shortage, which is captured by the following equation:

$$f_t^{i'} \leq \sum_{j \neq i}^{N} f_t^j * r_j, \tag{2}$$

where N is the agent's total customers.

The list of candidate customers (if any) is sorted according to their reliability (r_j) and the new DR requests are dispatched.

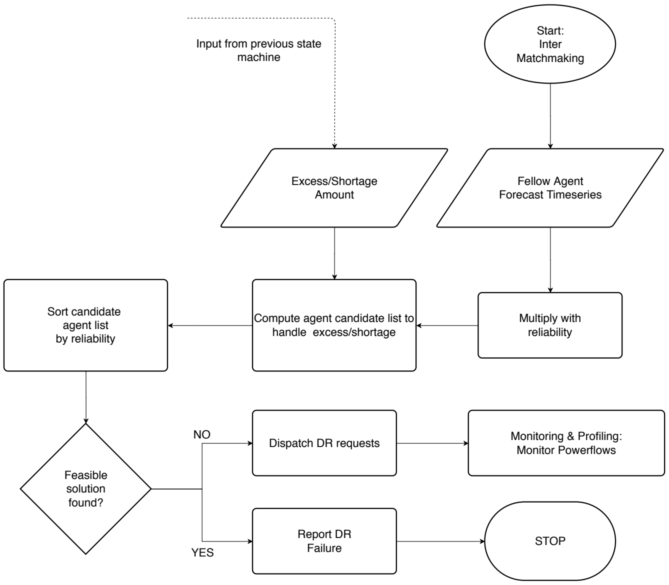

Fig. 3. Inter matchmaking flow diagram.

The "Inter Matchmaking" algorithm (Figure 3) is invoked in cases where the "Intra" branch fails to handle the excess/shortage of powerflow internally. It receives as input the amount of excess/shortage, which was calculated previously. In order for the agent to be able to come to a decision regarding which other fellow agent(s) it should dispatch DR requests to, it requires data pertaining to their individual aggregated forecasted flexibility. However, communicating this information during the active period of a DR event would impose additional time constraints. We eliminate this by having the agent preemptively query its other fellow agents prior to the start of the DR event's active period. To evaluate to which other agents DR requests will be dispatched, we follow a similar strategy as before. As illustrated in Figure 3, each agent maintains a local perception of the reliability of other fellow agents in the context of DR events. This metric is updated in a similar fashion as that of the VENs. Assuming the "Inter Matchmaking" algorithm finds a feasible solution to handle the excess/shortage, it dispatches the DR requests to the selected fellow agents. From hereon in, the selected agents follow the same flow that we have described up to this point.

VI. Virtual Node-based Agent Implementation

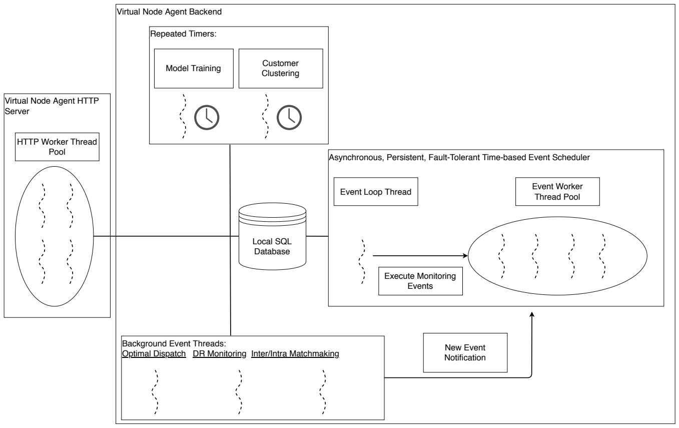

Fig. 4. Physical (implementation) diagram of our virtual node-based agent.

Figure 4 presents the physical diagram of the virtual node-based agent. On a high-level, it is comprised by three physically decoupled parts: 1) an HTTP server that exposes the agent's interfaces to the outside world, 2) the agent's backend, which encompasses all the components that we have described up to this point and, 3) a local SQL-based, relational database, which stores data that are relevant to the agent's operation, such as historical reports, forecasts and DR events. We integrate parts (1) and (2) via a standard Web Server Gateway Interface (WSGI [27]) based approach and parts (2) and (3) by employing the SQLAlchemy ([28]) toolkit.

The agent's backend is implemented as a multi-threaded, asynchronous, event-based application developed in Python ([29]). On the one hand, the agent is comprised by simple timer threads that execute on predefined time intervals: 1) the "Model Training" thread, which is responsible for training the agent's machine learning models and, 2) the "Customer Clustering" thread, which is responsible for updating the profiles of all the customer clusters maintained by the agent. On the other hand, the agent is also comprised by threads whose execution is triggered based on external input. For instance, the "Optimal Dispatch" thread is activated when the agent receives as input a DR request. Furthermore, the execution of events may lead to the internal scheduling of other events. We have already provided an example of such a case in, e.g., Section V, where the agent's monitoring component invokes the matchmaking component. To reliably synchronize the concurrent execution of all these events, we designed and implemented from scratch an asynchronous, persistent and fault-tolerant scheduler infrastructure for time-based events. This scheduler is comprised by one main thread whose sole role is to schedule the execution of newly created events to a pool of worker threads. This approach allows events to be executed in parallel and provides for increased throughput.

VII. Conclusion

Throughout the last two decades, there have been significant efforts in unlocking the flexibility potential of small to medium residential and tertiary customers for DR schemes. Either through market-based, or non-market-based solutions, most of the outcomes focus on optimizing the decision making and not the actual results, assuming that the DR will be delivered and, if not, it will simply be penalised. As this does not solve the actual problem that generated the need for issuing a DR request, it has been deemed necessary to investigate more towards understanding and resolving DR failures.

Towards that direction, a novel dynamic distributed multi-agent system has been presented, based on the OpenADR standard, where agents can act as VTNs and/or VENs, with layers above (e.g., aggregators), below (e.g., customers), or even among themselves. Both in terms of computation and business logic, the introduced MAS covers a wide range of customers, both in terms of size (small, medium, large) and type (consumers and prosumers). Apart from the optimal dispatch of OpenADR requests to customers, this work's main novelty revolves around the design of a two stage matchmaking algorithm that can resolve failures during the active period of a DR event. Based on a dynamic reliability index for either customers or other agents, each agent is able to resolve any potential failures either internally ("Intra Matchmaking"), or externally with other agents ("Inter Matchmaking"), respectively. This design is envisioned to improve significantly the success rate of DR events, hence: i) ensuring DR and, therefore, the grid's reliability, ii) minimising transmission losses and, iii) diminishing market risks. As part of an ongoing research endeavour, the proposed methodology will be deployed in real-life conditions to evaluate and validate the expected benefits.

References

[1] "Directive (eu) 2019/944 on common rules for the internal market for electricity and amending directive 2012/27/eu," https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32019L0944&from=EN, 2019.

[2] "Share of energy from renewable sources," https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=nrg_ind_ren&lang=en, 2020.

[3] L. Dusonchet, S. Favuzza, F. Massaro, E. Telaretti, and G. Zizzo, "Technological and legislative status point of stationary energy storages in the eu," *Renewable and Sustainable Energy Reviews*, vol. 101, pp. 158–167, 2019.

[4] "New passenger car registrations by fuel type in the european union," https://tinyurl.com/rkto7pa, 2019.

[5] "Tracking energy integration," https://www.iea.org/reports/tracking-energy-integration/demand-response, 2019.

[6] F. P. Sioshansi, *Consumer, Prosumer, Prosumager: How Service Innovations Will Disrupt the Utility Business Model.* Academic Press, 2019.

[7] "Study on residential prosumers in the european energy union," https://ec.europa.eu/commission/sites/beta-political/files/study-residential-prosumers-energy-union_en.pdf, 2017.

[8] "Directive (eu) 2018/2001 on the promotion of the use of energy from renewable sources," https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32018L2001&from=EN, 2018.

[9] M. Shafie-khah, P. Siano, J. Aghaei, M. A. Masoum, F. Li, and J. P. Catalão, "Comprehensive review of the recent advances in industrial and commercial dr," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3757–3771, 2019.

[10] O. Alliance, "Openadr 2.0 specifications," https://www.openadr.org/specification, 2013.

[11] A. González-Briones, F. De La Prieta, M. S. Mohamad, S. Omatu, and J. M. Corchado, "Multi-agent systems applications in energy optimization problems: A state-of-the-art review," *Energies*, vol. 11, no. 8, p. 1928, 2018.

[12] C. Silva, P. Faria, and Z. Vale, "Rating the participation in demand response programs for a more accurate aggregated schedule of consumers after enrolment period," *Electronics*, vol. 9, no. 2, p. 349, 2020.

[13] O. Alliance, "Enabling the standard for automated demand response," https://www.openadr.org/assets/docs/openadr\%20plma\%20seminar_final.pdf, 2013.

[14] C. G. Rieger, K. L. Moore, and T. L. Baldwin, "Resilient control systems: A multi-agent dynamic systems perspective," in *IEEE International Conference on Electro-Information Technology, EIT 2013*. IEEE, 2013, pp. 1–16.

[15] I. Praça, C. Ramos, Z. Vale, and M. Cordeiro, "Mascem: a multiagent system that simulates competitive electricity markets," *IEEE Intelligent Systems*, vol. 18, no. 6, pp. 54–60, 2003.

[16] R. Badawy, B. Hirsch, and S. Albayrak, "Agent-based coordination techniques for matching supply and demand in energy networks," *Integrated Computer-Aided Engineering*, vol. 17, no. 4, pp. 373–382, 2010.

[17] H. Golmohamadi, R. Keypour, B. Bak-Jensen, and J. R. Pillai, "A multi-agent based optimization of residential and industrial demand response aggregators," *International Journal of Electrical Power & Energy Systems*, vol. 107, pp. 472–485, 2019.

[18] E. Karfopoulos, L. Tena, A. Torres, P. Salas, J. G. Jorda, A. Dimeas, and N. Hatziargyriou, "A multi-agent system providing demand response services from residential consumers," *Electric Power Systems Research*, vol. 120, pp. 163–176, 2015.

[19] D. Muthirayan, D. Kalathil, K. Poolla, and P. Varaiya, "Mechanism design for demand response programs," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 61–73, 2019.

[20] Z. Wang, R. Paranjape, Z. Chen, and K. Zeng, "Multi-agent optimization for residential demand response under real-time pricing," *Energies*, vol. 12, no. 15, p. 2867, 2019.

[21] J. A. Taylor and J. L. Mathieu, "Uncertainty in demand response—identification, estimation, and learning," in *The Operations Research Revolution*. Informs, 2015, pp. 56–70.

[22] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. Springer-Verlag, 2002, p. 53–65.

[23] I. Realtime, "Openfire," https://www.igniterealtime.org/projects/openfire, 2020.

[24] T. Dierks and E. Rescorla, "The transport layer security (tls) protocol," Internet Requests for Comments, RFC Editor, RFC 5246, August 2008. [Online]. Available: https://www.rfc-editor.org/rfc/rfc5246.txt

[25] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet x.509 public key infrastructure certificate and crl profile," Internet Requests for Comments, RFC Editor, RFC 2459, January 1999. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2459.txt

[26] E. Gelenbe, G. Görbil, D. Tzovaras, S. Liebergeld, D. Garcia, M. Baltatu, and G. Lyberopoulos, "Nemesys: Enhanced network security for seamless service provisioning in the smart mobile ecosystem," in *Information Sciences and Systems*, E. Gelenbe and R. Lent, Eds. Springer International Publishing, 2013, pp. 369–378.

[27] P. Eby, "Pep 3333 – python web server gateway interface v1.0.1," https://www.python.org/dev/peps/pep-3333/#copyright, 2010.

[28] M. Bayer, "Sqlalchemy," in *The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks*, A. Brown and G. Wilson, Eds. aosabook.org, 2012, http://aosabook.org/en/sqlalchemy.html.

[29] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. CreateSpace, 2009.