Application of intelligent agent systems for Real-time Coordination of Power Converters (RCPC) in Microgrids

Conference Paper · September 2014

DOI: 10.1109/ECCE.2014.6953937

3 authors:

Maryam Nasri
Simon Fraser University

Herbert L. Ginn
University of South Carolina

Mehrdad Moallem
Simon Fraser University

Application of Intelligent Agent Systems for Real-time Coordination of Power Converters (RCPC) in Microgrids

Maryam Nasri, Herbert L. Ginn
Department of Electrical Engineering
University of South Carolina
Columbia, SC
{nasri, ginnhl}@cec.sc.edu

Mehrdad Moallem
School of Mechatronics System Engineering
Simon Fraser University
Surrey, BC, Canada
mmoallem@sfu.ca

***Abstract*— This paper presents results of a study to determine the most appropriate agent based architecture for implementation of controllers for stand-alone microgrids. The controller has to perform these main tasks: maintaining sufficient system voltage during supply overload conditions, balancing load flow, and managing voltage level in case of failure of some converters, or adding new converters to a system bus. The paper proposes application of agent technology in achieving of each aforementioned controller actions and Real-time Coordination of Power Converters (RCPC) in microgrids. The paper compares system complexity using numerical analysis of different distributed lookup algorithms based on defined metric values. The results aid in choosing publish/subscribe (pub/sub) model over distributed hash table (DHT) infrastructure as the most efficient and scalable solution of developing agent technology for the RCPC system. To test the applicability of the RCPC optimization method, a sample DC shipboard microgrid including 32 converters is used as a case study.**

I. INTRODUCTION

Microgrids are clusters of energy sources, storage systems, loads, local networks, innovative and efficient supplies, real-time technologies, and load controllers that are organized to offer an energy solution for a community while connected to power grids, or operated as an electrical island [1,2]. Because of microgrids smaller size and higher flexibility compare with power grids, different optimization algorithms were evaluated in their both connected and islanded modes [3]. The optimization mechanism can include some functions such as minimizing power loss, balancing load flow, increasing stability and reliability, and reducing fuel cost. To achieve most of the optimization factors, a comprehensive and coordinated control system is required which enables load control of each power component independently. In addition, for managing the coordination among local controllers, a higher control layer is required. Also, as the number of active converters can be changed in a microgrid, the control topology is required to meet upper time limits in real-time message exchange regardless of the number of involved converters.

In this paper we propose one optimization method using agent technology for real-time coordination of power converters in microgrids. Multi-agent system (MAS) has an event-driven real-time architecture to optimize required parameters of a distributed microgrid. The system is flexible in adding and removing distribution nodes, and it is desirable to self-heal after fault occurrence. Figure 1 shows a model of a shipboard microgrid. Shipboards are finite state machines able to connect to auxiliary power sources, otherwise they may operate near the threshold of power constrained [4]. They include a variety of power devices such as distributed generators (DGs), power converters, and loads. In sample microgird system, DGs include main turbo

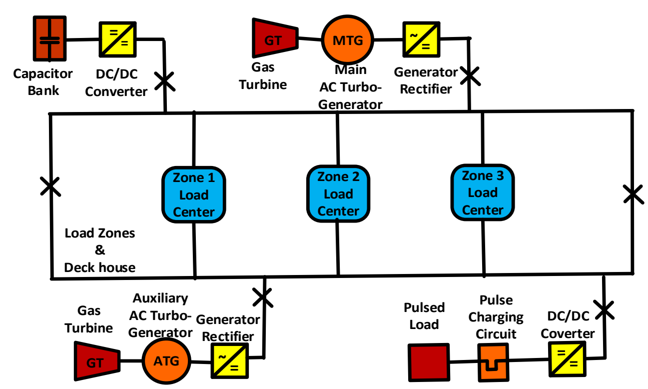


Figure 1. General topology for a shipboard DC system

978-1-4799-5776-7/14/$31.00 ©2014 IEEE

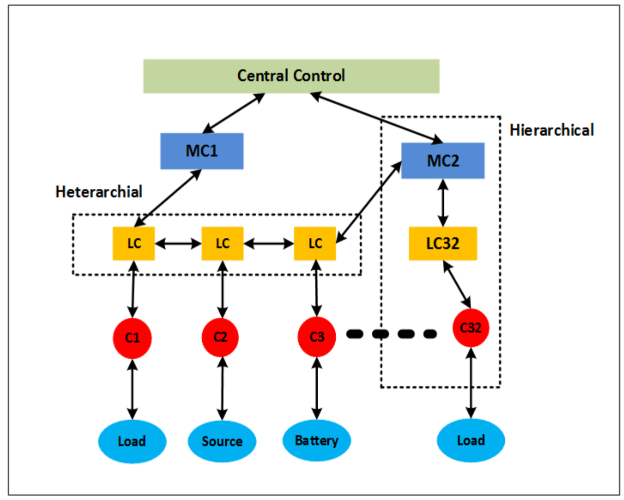
generators (MTGs), auxiliary turbo generators (ATGs), and capacitor banks (CBs). In this paper, section I provides introduction to micrograms, and RCPC high level design. Section II presents the definition of MAS and their application in RCPC. Moreover, algorithms and computational methodologies are described in section III. Designing RCPC control system using pub/sub model and DHT lookup algorithm is discussed in section IV. Case study is explained in section V, followed by section VI that is described management of adding new converters and failure of active converters using DHT. The paper is concluded in section VII.

II. Application of Muti-agent Technology in Control of Microgrids

An Intelligent Agent is an autonomous software-based (and/or hardware-based) system that is designed to take into account dynamic environmental requirements to deliver its design objectives. The environment can be physical (e.g. a protection switch) or computing space (e.g. a software program)[5], and [6]. MAS is a system comprising two or more agents [7].Combination of three technologies including web services, grid computing, and intelligent systems are used in MAS that it can also be designed, implemented and used in a power system. Furthermore, MAS needs to be implemented in a dynamic real-time computing system. Until recently, MAS has been rarely used in energy distribution applications. For example, a real-time adaptive VVO/CVR topology using MAS was employed for voltage regulation in power distribution networks [8-9].

As Figure 2 depicts, architecture of MAS used in power systems consists of three different control layers: reactive (local) layer, coordinator (middle) layer and deliberative (planning) layer [10]. Each agent designed in an MAS is placed in different layers based on its task (e.g. middle agent is assigned to the middle layer and applied as a bridge to communication processes). Therefore, each agent used in the power systems could be found in one of the three layers depicted above. Agents are located in the reactive layer if they are pre-programmed such as converter agents (CAs) to do certain tasks. Moreover, agents are placed in the middle and planning layers if they are context dependent, cooperating in their local tasks and competing with similar agents in other nodes in pursuing global goals. In addition, coordinator agents (e.g. middle agent) may communicate with agents in the other two layers.

Hierarchical control architecture displays communicating among N control layers which are connected to each other as a client server. There is no direct communication between modules of the same level. While this architecture proposes a hierarchical modeling methodology for real-time scheduling, its feasibility and optimality are not proven. The heterarchical control architecture is based on full local autonomy (distributed control) resulting in a control environment in which autonomous components (agents) co-operate in order to reach global objectives through local decision making [11]. In a multi-agent heterarchy where each agent represents an



`

Figure 2. A combination of central, middle, and local controllers in different control topology for RCPC microgrid including hierarchical, heterarchial, hybrid, and agent control technologies

individual power device (e.g. converters), each agent individually implements these low-level control algorithms for all of the converters they represent. Duffie claims that other advantages of heterarchical architectures include reduced complexity, increased flexibility, and reduced costs [12]. In hybrid architecture as we have used for RCPC control design, the purpose is to combine the predictability of the centralized and hierarchical control architectures with the agility and robustness against disturbances and high degree of adaptability of the heterarchical control architectures [13].

Figure 3 illustrates the application of agent technology for RCPC in a microgrid. Each CA is connected to one of 32 converters using Ethernet communication protocols (TCP/IP or UDP/IP) located in the network lower layer. A group of CAs is assigned to each middle agent (MA). The number of MAs and the topology of their connection to both CAs at the lower layer and planning agent (PA) at the upper layer are varied based on distributed agent algorithms and system architecture. There is only one PA in RCPC that is responsible for saving and mapping high-level system plans.

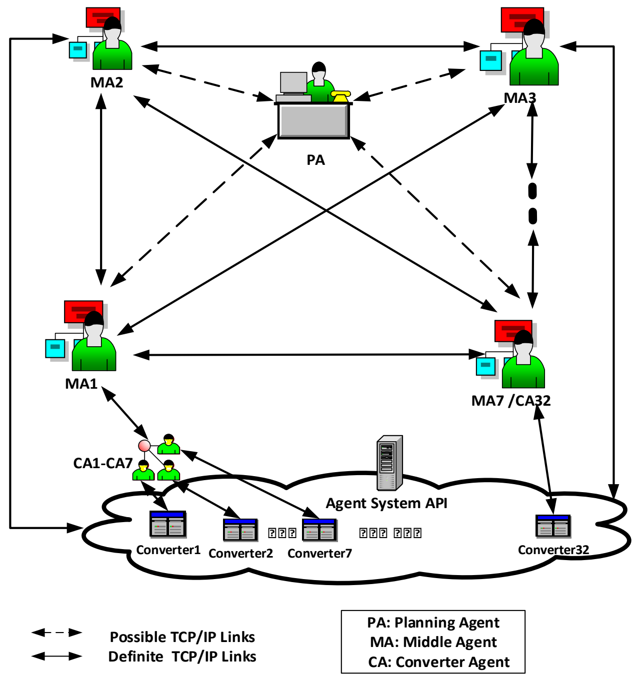


Figure 3. Communication design between three types of agents in RCPC

Searching a specific converter, adding, and deleting converters are considered as the main tasks in RCPC system. Message count is defined and used as a metric value to evaluate system running time for different algorithms in order to choose the most efficient algorithm among those who meet the required upper-limit. In addition to the time limitation, the system has other constraints such as maximum number of concurrent running tasks, and asynchronous running speeds between hardware simulated with MATLAB/Simulink and MAS API implemented with high level programming languages such as Java.

III. Algorithms and Computational Methodology

Different optimization algorithms can be applied in multi-agent technogy. In this section, a few of the most efficient algorithms extracted from literature are reviewed and customized for the RCPC. Message counts and in some algorithms number of groups (node ranges) are considered as metric values for RCPC. In addition, algorithms are compared using numerical analysis of their complexity formula based on defined metrics. The aforementioned algorithms shown in Table I include: 1) Belief-Desire-Intention (BDI) architecture, using bidding lookup algorithm; 2) Facilitator agent mechanism, using totally ordered multicast searching algorithm; 3) Pub/sub model using DHT lookup algorithm. Figure 4 compares running value of three bidding algorithms using their complexity (1), and (2). Figure 4.a displays that the sequential algorithm has the least complexity and therefore is the most efficient and persistent approach.

Computational weight for each iteration is O(mn) where n and m are numbers of agents and tasks, respectively. Considering the worst case scenario by assuming that all of tasks are being run in parallel, then m=n. Therefore, the bidding cost can be calculated from (1). The computational weight for each iteration is O(n+m) where n is number of auctioneer and m is number of bidders. Considering the worst case scenario when all the agents are bidding concurrently, the bidding cost is calculated from (2).

$$Cost \le (n * m) \quad \& \quad m = n \Rightarrow Cost \le (n * n) = n^2 \quad \text{where } 0 \le n \le 50 \qquad (1)$$

$$Cost \le (n + m) \quad \& \quad m = n \Rightarrow Cost \le (n + n) = 2n \quad \text{where } 0 \le n \le 50 \qquad (2)$$

TABLE I. Three Optimization Methods In Agent Technology

	Design Methods for Agent Technology	**Distributed Lookup Algorithms**
BDI Using Bidding Algorithm	Belief Desire Intention (BDI)	Bidding Algorithm
Facilitator Using Totally Ordered Multicast	Facilitator Agent Technology	FIFO + Causally Ordered Multicast
Pub/Sub Using DHT	Publish/ Subscribe	Distributed Hash Table (DHT)

To calculate the computational weight, each algorithm uses metric values for its calculations. To choose the best lookup method, three of the most efficient ones have been chosen and evaluated in Figure 5. In totally ordered multicast protocols, A broadcast message is sent to every working node connected to the microgrid [14]. A causally ordered protocol delivers messages according to the causal relations between the sending events [15]. A totally ordered protocol delivers messages in an arbitrary order. When it is causally ordered, this arbitrary order is consistent with the causal relations. In the other hand, pub/sub mechanism supports the simultaneity of actions for system consistency in distributed control. In this method, agents get access to the hash tables to coordinate and synchronize their data. As shown in Figure 5, DHT is the most efficient optimization technology. Since DHT algorithm is compatible with pub/sub agent technology, complexity of combination of these two methods is also graphed for a variation from 1 to 50 converters. As the message count displays a consistent rate in this interval for pub/sub over DHT, it will be used as RCPC design method in the next section.

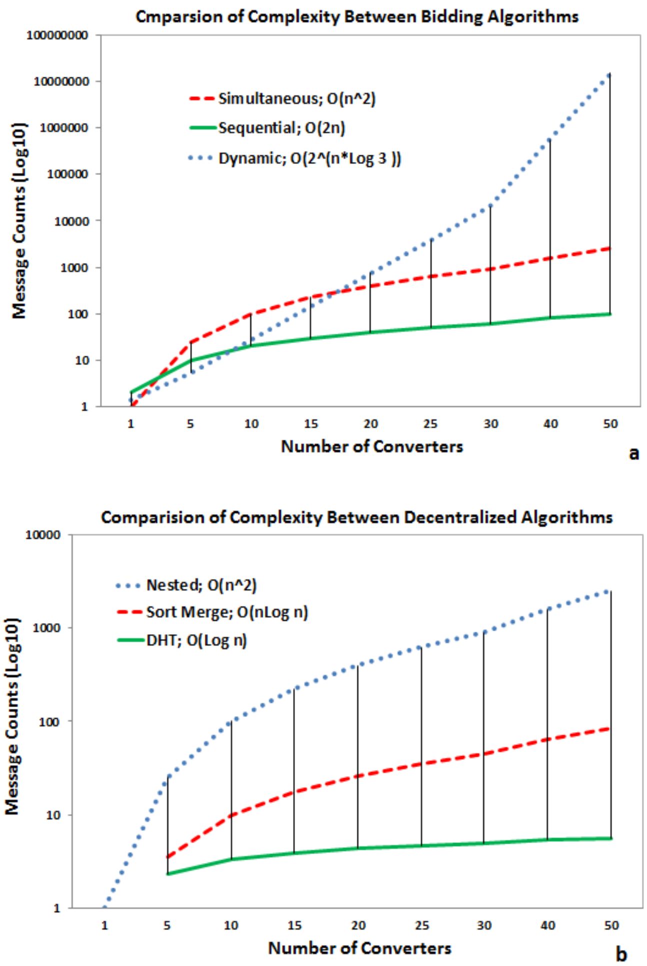


Figure 4. Numerical analysis for comparison bidding and decentralized lookup algorithms

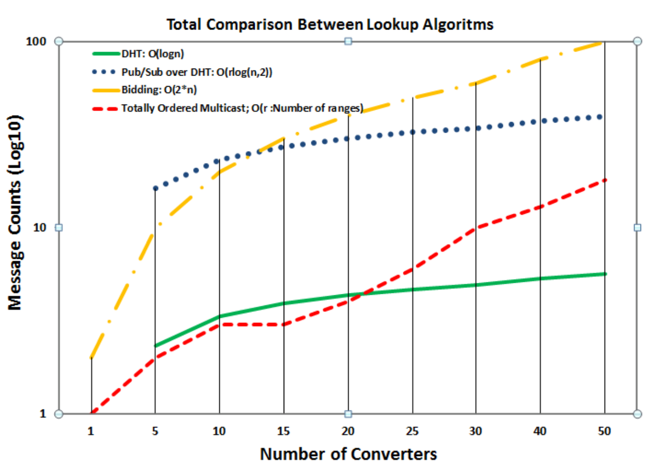


Figure 5. Numerical analysis for comparison between selected lookup algorithms

IV. Designing RCPC Control System Using Publish-Subscribe Model and DHT Lookup Algorithm

Pub/Sub is a data distribution algorithm which its main functionality is delivery of published notifications from every publisher (producer/event source) to all interested subscribers (consumers) using an overlay infrastructure [16]. There are two types of pub/sub systems: 1) topic based which consumers are subscribed based on their topics such as newsgroups, and 2) content based that publisher is given the ability to express its interest to subscribers by specifying a period of defined values over different attributes. Therefore, the content based pub/subs are made of {attribute, operator, value} tuples, where operator can be one of {<, =, >, ≤, ≥}, and publishers are a set of {attribute, value} pairs [17,18].

An overlay network is a network which has made on the top of an existing network. This is a good solution for supporting distributed algorithms such as pub/sub systems. DHT is an overlay network which is designed on application layer of TCP/IP internet network.

DHTs are a type of distributed searching algorithms. They use hashtable functionality to manage join and leave of the nodes in a wide-area environment. The sets of (key, value) which defined in hashtables help users to retrieve a value corresponding to the given key. Application of pub/sub over DHT in large-scale distributed systems has studied in some literatures [19,20]. It might be argued that in small-scale systems, pub/sub can be implemented on a centralized overlay network protocol with lower cost. But DHT is preferable for expanding networks because it won't require any changes in system architecture for adding the nodes to system over the time. In addition, there is no single point of failure in DHT against centralized algorithms. So the abilities to self-heal and scale up the system size easily and with low cost, will advance the usage of DHT over centralized algorithms. Section A of this chapter, explains the design of case study system based on pub/sub model, followed by section B which described the application of DHT as pub/sub infrastructure in this case study.

TABLE II. Definition of Middle Agents (MAs) for RCPC

	Key Nodes	Successor Nodes	Types of Nodes
1	MACL	CA1,…, CA7	Critical Load (CL)
2	MASCL	CA8,…, CA17	Semi-Critical Load (SCL)
3	MANCL	CA18,…,CA28	Non-Critical Load (NCL)
4	MAG1	CA29	DG1: MTG
5	MAG2	CA30	DG2: MTG
6	MAG3	CA31	DG3: ATG
7	MAG4	CA32	DG4: CB

A. Publish/Subscribe

In this subsection, we introduce a MAS using pub/sub and DHT for maintaining sufficient voltage level during supply overload voltage. As depicted in Figure 3, each of 32 nodes is connected to a converter agent through a converter, and a group of CAs based on system topology is assigned to each of middle agents. Table 2 introduces seven MAs and their associated CAs in RCPC. MAs are categorized based on their node types including critical load (CL), semi-critical load (SCL), non-critical load (NCL), and distributed generator (DG). Publishers and subscribers exchange messages during their matching process. Figure 6 illustrate a message structure which is included converter agent number (CAn) as a unique number between 1 and 32, and Sid and Eid as subscriber and event IDs respectively (3), and (4). At the last part of message, attributes and their values using operators have specified (3)-(10).

CAn	**Sid_{ijk}/ Eid_{ijk}**	**(attribute, operator, value)**

Figure 6. Publish/ Subscribe Message Structure

$$Eid_{ijk}\ (1 \le i \le 7, 1 \le j \le 4, 1 \le k \le 8) \tag{3}$$
$$Sid_{ijk}\ (1 \le i \le 7, 1 \le j \le 4, 1 \le k \le 8) \tag{4}$$
$Where$:
$i\ is\ node\ type\ i.e. CL, SCL, and\ NCL, DG1, DG2, DG3, DG4$
$j\ is\ physical\ area\ of\ nodes$
$k\ is\ message\ priority\ based\ on\ arrival\ time$

B. DHT

Recently, DHTs have emerged as an infrastructure for scalable, and efficient resource lookup in distributed networks because of their decentralized, and self-organized specifications. These characteristics make DHTs attractive for building distributed applications, such as distributed file systems [21]. Any DHT could be utilized for routing of n-dimensional index. In Chord method that is the most popular DHT structure, the complexity for routing table, lookup, and peer joint/leave are O(log n), O(log n), and $O((\log n)^2)$ respectively. The main idea of DHT is partitioning tables. Each node gets an identity by hashing its unique ID (CAn) and keys are also hashed into the same space. A key(k) with a hashed ID k is assigned to the first node whose hashed ID is equal to or follows k in a circular space: Successor (k), Put (key, value) → Lookup (key) → value.

V. Case Study

Table 3 displays the numerical values and description of 32 nodes in sample system [22]. The following steps are taken by agents for finding the match pairs of events and subscriber among all the 32 nodes. The pub/sub algorithm over DHT infrastructure is applied for finding the match nodes:

1) Both of MTG related agents subscribe to the load middle agents (MACL, MASCL, MANCL) (5), (6). As Figure 7.a displays, six total messages in this step is exchanges.

2) Auxiliary power resources subscribe to both main power resources (7), (8) and add four more message counts to the system complexity (Figure 7.a).

3) Load events release from CAs whenever load value increases more than 15% of its defined value, so generator needs to provide extra power. Four different loads (# 3, 15, 20, and 23) experience the overload at the same time. CAs route the events (E1, E2, E3, and E4) to the related load middle agents based on the first digit of their Eid numbers. So E1, E2, and E3, and E4 go to MACL, MASCL, and MANCL respectively (9)-(12). Routing happens based on numerical coefficient of Sid and Eid messages IDs (Figure 7.b)

$$S1 = \ (Sid_{411},\ 0 \le \Delta P \le 200) \tag{5}$$
$$S2 = \ (Sid_{432},\ 0 \le \Delta P \le 100) \tag{6}$$
$$S3 = \ (Sid_{621}, 0 \le \Delta P \le 4000) \tag{7}$$
$$S4 = \ (Sid_{741}, 0 \le \Delta P \le 520) \tag{8}$$
$$E1 = (Eid_{111},\ \Delta P \ge\ 16.5) \tag{9}$$
$$E2 = (Eid_{221},\ \Delta P \ge 14.25) \tag{10}$$
$$E3 = (Eid_{341},\ \Delta P \ge 0.57) \tag{11}$$
$$E4 = (Eid_{321},\ \Delta P \ge 0.75) \tag{12}$$
$$E5 = (Eid_{431}, \Delta P \le 3600) \tag{13}$$
$Where$: $Absolute\ Value\ of\ Load\ Changes\ (\Delta P)$
$$\Delta P = \ \left|P_{current} - P_{defined}\right| \tag{14}$$

a) All of the CAs related to loads route Eid messages to one of three load middle agents based on their first coefficient values (7.b). It is defined as number 1 for critical loads (CL), 2 for semi-critical loads (SCL) and 3 for non-critical loads (NCL).

b) Moreover, load middle agents forward events to the physically closest generator by checking the second coefficient value (7.b).

c) At the subscriber middle agents (MAG1, and MAG2), events are listed in a queue based on their load type and their arrival order which can be specified using first and third coefficient values of their Eid number.

d) The arrival order (3th coefficient number) sets in each agent individually. It means in each publisher and subscriber agent, there is a queue that events will place in it based on their arrival order to that specific agent. In this step another 10 messages are added to the system message count.

TABLE III. NODES VALUE AND DESCRIPTION FOR CASE STUDY (CL: CRITICAL LOAD, SCL: SEMI-CRITICAL LOAD, NCL: NON-CRITICAL LOAD)

CA No.	Node Type	P(kW)	CA No.	Node Type	P(kW)
1	CL	20	17	SCL	25
2	CL	30	18	NCL	3
3	CL	110	19	NCL	11
4	CL	2	20	NCL	38
5	CL	5	21	NCL	24
6	CL	3	22	NCL	3
7	CL	135	23	NCL	5
8	SCL	40	24	NCL	2
9	SCL	1.2	25	NCL	83
10	SCL	10	26	NCL	7
11	SCL	40	27	NCL	49
12	SCL	95	28	NCL(PL)	12000
13	SCL	50	29	DG1(MTG)	36000
14	SCL	20	30	DG2(MTG)	41000
15	SCL	95	31	DG3(ATG)	4000
16	SCL	1.2	32	DG4(CB)	520

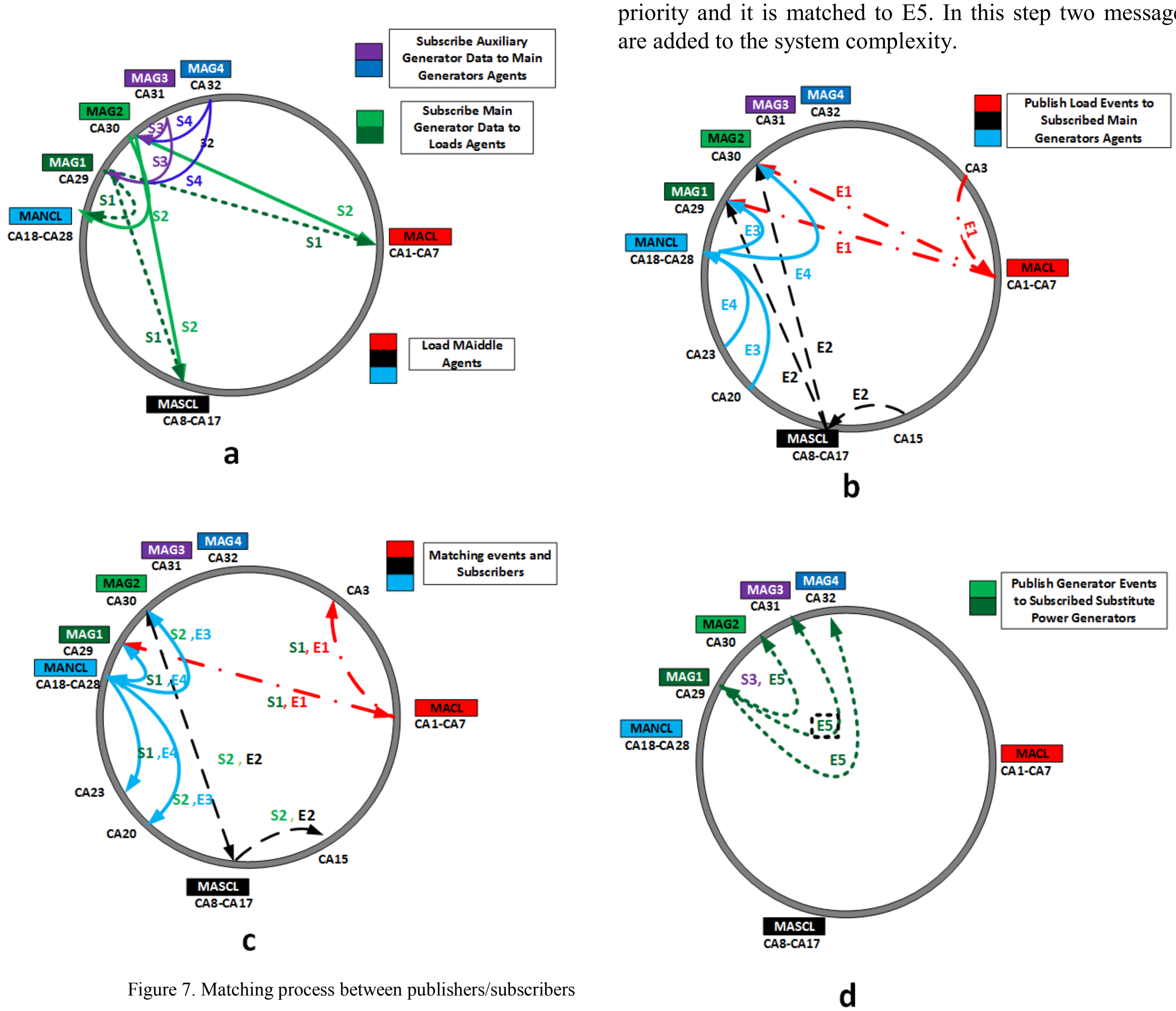


Figure 7. Matching process between publishers/subscribers

4) After completing matching steps each generator agent includes a finger table with publisher Eids. As the infrastructure is made on DHT, the weight of routing lookup table (subsection IV.B) is at most equal: Log32= 5 messages which is very efficient compared to the other searching algorithms (Figure 5). Since we matched four events, 20 message counts added to the previous system complexity based on system metrics (Figure 7.c).

5) The generator agents issue an event if the generator power falls less than 10% of its defined value. As Figure 7.d illustrates, E5 released in this situation (13).

6) Both of S3, and S4 are eligible to provide the required power for system since they both are physically neighbor with MATG1 and are also subscribed to it. Although considering the ΔP values of both subscribers, S3 has priority and it is matched to E5. In this step two messages are added to the system complexity.

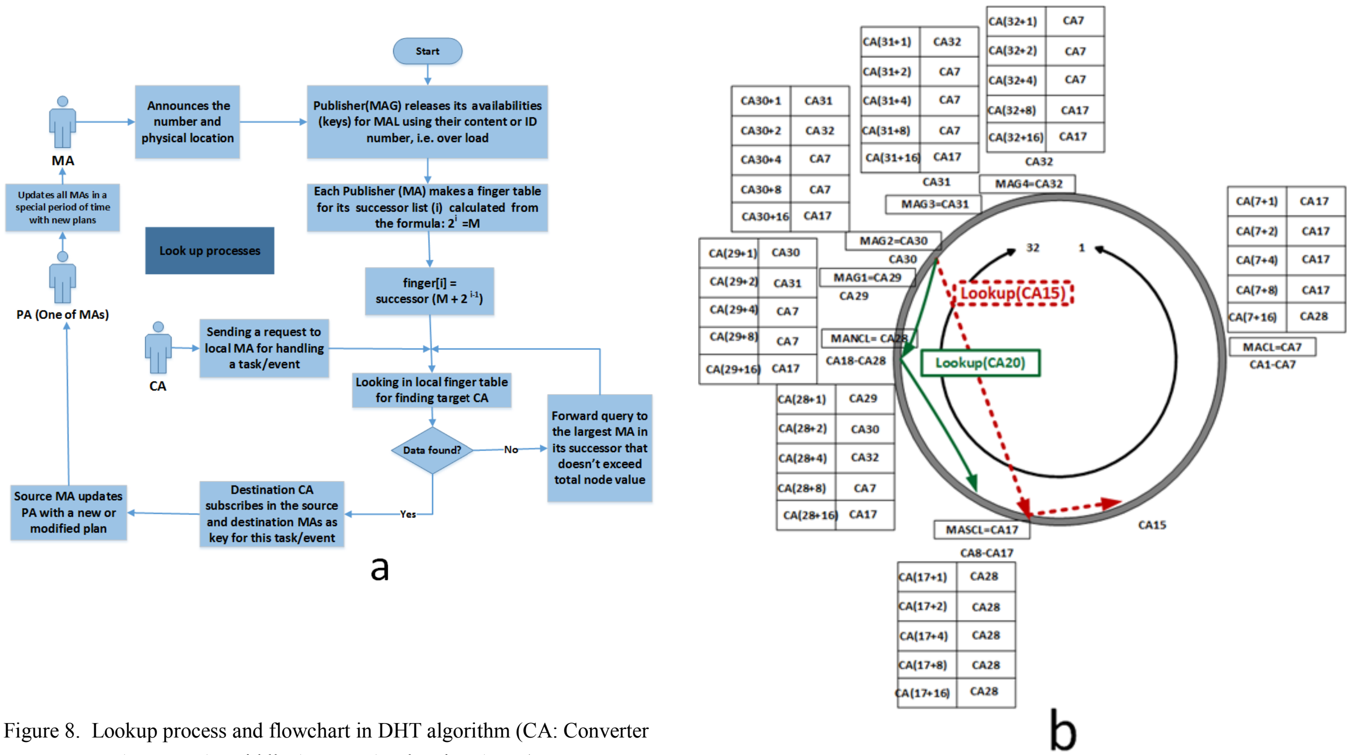


Figure 8. Lookup process and flowchart in DHT algorithm (CA: Converter Agent; MA: Middle Agent; PA: Planning Agent)

7) All of MAs routing table and message exchange history map to PA after finishing matching procedure which it costs seven more message counts. Therefore, in the whole processes of case study, 49 messages are delivered using agents.

Figure 8 illustrates the diagram and flowchart for lookup processes which are taken with DHT and pub/sub. The other procedures such as managing load balance in case of adding and failure of converters will discuss in the next section.

VI. Manging Adding and Failiure of Converters Using DHT

One of the most important characteristics of DHT is its ability for self-organizing. It means system automatically adapts to the arrival, departure and failure of nodes. So a single point of failure will not necessarily affect the entire system. Furthermore, the system is robust in case of adding nodes. Figure 9 displays the flexibility of RCPC system for handling the changes in the number of converters. In this Figure the steps of handling a node failure and replacing it with another node is explained by detail. As it was mentioned in subsection IV.B, the complexity of calculation is less than $(\text{Log } 32)^2$ for handing peer joint/leave in DHTs.

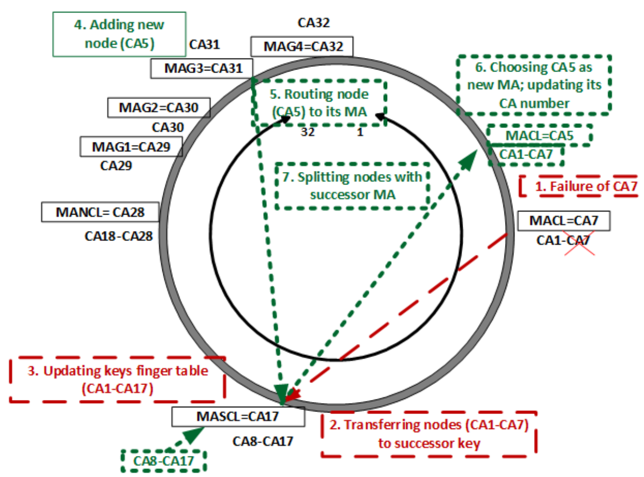


Figure 9. Managing converter adding and failure

VII. Conclusions

In this paper, a novel method for coordination of power converters in a sample microgrid using agent technology was presented. A combination of pub/sub and DHT was used for coordination of a middle sized microgrid. The paper contributions include application of the proposed agent technology in: 1) Providing required voltage level in case of system overloading; 2) Balancing load flow; 3)

Management of system voltage level in case of any converter failure.

In the case study, the results of modeling a multi-agent system using pub/sub algorithms has been analyzed numerically. It was found that the combination of pub/sub and DHT searching method were the most efficient, scalable optimization algorithms for medium size migrogrids. As shown in Figure 5, an increase in the number of converters from 5 to 50 did not affect the upper time limits of message exchange, and event handling. Furthermore, the management of adding and deleting converters based on DHT infrastructure displays the same level of efficiency as lookup process. The topic based pub/sub method has been modeled using the Jason Platform and the topic is chosen based on nodes IP Addresses. However, since we have advanced topic-based to content-based pub/sub model, the Jason platform is not a suitable platform for implementing the case study. Java Message Service (JMS) toolbox has been identified as the best option for developing API and the results will be presented in future work.

REFERENCES

[1] D. Bakken, “Smart Grids: Clouds, Communications, Open Source, and Automation”, Textboob, CRC Press, May 2014.

[2]http://www.horizonenergygroup.com/page.asp?p=Horizon%20Microgrid%20Solutions

[3]http://washingtontechnology.com/calendar/2012/11/military-smart-grids-and-microgrids-conference.aspx

[4] S. Abdelwahed, A. Asrari, J. Crider, R.A. Dougal, M.O. Faruque, Y. Fu, J. Langston, Y.Lee, H.A. Mohammadpour,A. Ouroua, E. Santi, K. Shoder, S.D. Sudhoff, Y. Zhang, H. Zheng, E. Zivi, “Reduced order modeling of a shipboard power system”, Electric Ship Technologies Symposium (ESTS), IEEE,2013, pp.256-263.

[5] M. Wooldridge, An Introduction to MultiAgent Systems – (Second Edition), Wiley & Sons, May 2009.

[6] S.D.J. McArthur, E.M. Davidson, V.M. Catterson, A.L. Dimeas, N.D. Hatziargyriou, F.Ponci and T. Funabashi, “Multi-Agent Systems for Power Engineering Part I and II,” IEEE Trans. Power Systems, Vol. 22, pp. 1743–1759, Nov. 2007.

[7] H. Fakham, A. Ahmidi, F. Colas and X. Guillaud, “Multi-Agent System for Distributed Voltage Regulation of Wind Generators Connected to Distribution Network,” in Proc. Innovative Smart Grid Technologies Conference Europe (ISGT Europe), IEEE PES, Gothenburg, Sweden, Oct. 2010.

[8] M. Nasri, H. Farhangi, A. Palizban, M. Moallem, “Multi-Agent Control System for Real-time Adaptive VVO/CVR in Smart Substation”, Electrical Power and Energy Conference (EPEC), 2012 IEEE, 1-7.

[9] Moein Manbachi, Maryam Nasri, Babak Shahabi, Hassan Farhangi, Ali Palizban, Siamak Arzanpour, Mehrdad Moallem, Daniel C Lee, “Real-Time Adaptive VVO/CVR Topology Using Multi-Agent System and IEC 61850-Based Communication Protocol”, IEEE Transaction on Sustainable Energies, issue 99, 2013.

[10] M. Shahidehpour and Y. Wang, Communication and Control in Electric Power Systems, Applications of Parallel and Distributrd Processing. IEEE Press, Willey InterScience, pp. 36-44, 2003.

[11] “Heterarchical control of a lithoshop” J. van Dongen, J.M. van de Mortel-Fronczak and J.E. Rooda

[12] N.A. Duffle, R.S. Piper, B.J. Humphrey, and J.E Hartwick, "Hierarchical and Non-Hierarchical Manufacturing Cell Control with Dynamic Part-Oriented Scheduling," Proc. of the 14th North American Mfg. Research Conf., Minneapolis, May 1986, pp504-507.

[13] Robert W. Brennan, Douglas H. Norrie, “Metrics for evaluating distributed manufacturing control systems”, Computers in Industry 51 (2003) 225–235, Elsevier.

[14] G. Florin C. Toinard. “A new way to design causally and totally ordered multicast protocols”, Acm Sigops Operating Systems Review Homepage archive, Volume 26 Issue 4, Oct. 1992, Pages 77-83.

[15] L. Lamport. Time, clocks and the ordering of events in a distributed system. CACM Vol 21,Number 7, July 1978.

[16] S. Tarkoma, “Publish/Subscribe Systems Design and Principals”, textbook, Wiley, 2012.

[17] V. Muthusamy, H.A. Jacobsen, “Small-Scale Peer-to-Peer Publish/Subscribe”, MobiQuitous P2PKM, 2005.

[18] P. Triantafillou, I. Aekaterinidis, “Content-based Publish-Subscribe Over Structured P2P Network”

[19] M. Castro, P. Druschel, A.M. Kermarrec, and A. Rowstron, “Scribe: A large-scale and decentralized application level multicast infrastructure”, IEEE JSAC, 20(8), oct 2002.

[20] A. Gupta, O. D. Sahin, D. Agrawal, and A. E. Abbadi. “Meghdoot: Content-based publish/subscribe over P2P networks”, In Middleware, pages 254–273, 2004.

[21] D. Tam, R. Azimi, and H. Jacobsen, “Building Content-Based Publish/Subscribe Systems with Distributed Hash Tables”, To appear in (i) the International Workshop on Databases, Information Systems and Peer-to-Peer Computing, September 7-8, 2003, Humboldt University, Berlin, Germany.

[22] X. Feng, T. Zourntos, K. L. ButlerPurry, and S. Mashayekh, “Dynamic Load Management for NG IPS Ships”, Power and Energy Society General Meeting, IEEE, 2010, pp.1-8.