Dynamic Resources Management Under Limited Communication Based on Multi-level Agent System

Zhigang Wang
CETC Key Laboratory of Data Link Technology
Xi'an, China
feitianwzg@163.com

Chilian Chen
CETC Key Laboratory of Data Link Technology
Xi'an, China
cclian@163.com

Chubing Guo
CETC Key Laboratory of Data Link Technology
Xi'an, China
715608993@qq.com

2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT) | 979-8-3503-9905-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICEICT57916.2023.10245884

***Abstract*—Under the condition of limited communication, the message interaction between different platforms cannot meet the demand of real-time and integrality, which significantly reduces the task-execution efficiency. Aiming at making full use of all kinds of payloads resources in the dynamically changing environment, we use a multi-level agent system model to describe the control relationship between different types of platforms and payloads, and propose an improved contract net protocol to coordinate tasks that different platforms need to undertake and manage different payloads of a platform. The experiment results of the semi-physical simulation show that the proposed method can efficiently assign the required resources to the dynamic tasks while interacting with little information between different platforms.**

Keywords—resources management, multi-level agent system, CNP, limited communication

I. Introduction

In a complex task environment, it's important to dynamic manage and control all kinds of platforms and payloads according to the change of targets and self-status. However, the control platforms may not receive and process other platform's status information in real time due to the irregular interference of the communication system, which directly affects the fusion accuracy of distributed sensors data [1], and thus reasonable and feasible decisions can't be made based on the incomplete and unreliable information [2]. Compared to the centralized control framework, the distributed control framework is more flexible, but it needs more information interaction between distributed platforms and the time used to achieve global consensus is out of control. Reference [3] proposed a communication topology of multi-agent system and built a realistic communication channel mode to actualize formation control under wireless network with communication constrains and uncertainty, however, the communication channel mode needs constant updates to keep the mode's accuracy in a highly dynamic changing environment, which significantly increase the amount of computation. Reference [4] carried out an in-depth study on the multi-UAVs cooperative control based on the high-order linear multi-agent systems time-delayed consensus theory, but the proposed method is difficult to apply to the engineering realization. The methods based on the reinforcement learning were proposed in reference [5-6] to allocate the heterogeneous resources and make decisions collaboratively, but the proposed methods required a lot of data for training and the trained model was difficult to adapt to complex task scenarios.

We use a mixed multi-level agent system framework to describe the control and synergic relationship between different types of platforms, and use an improved contract net protocol to coordinate platform-level tasks for different platforms and payload-level tasks for different payloads mounted in a platform, which will make the most of all kinds of resources to preplanned targets and emerging targets in the complex task environment.

II. Multi-level Agent System Framework

The multi-level agent framework consists of three levels: control platforms agent, execution platforms agent and payloads agent, as shown in Fig.1. Every agent is a virtual mapping of a real platform or payloads. A control platforms agent can manage and control several execution platforms, and an execution platform can carry and control several payloads. Each level agent plays a different role based on its capability. The control platforms agent is responsible to refine and decompose the received task intention, the decomposed primitive tasks should be executed by a single platform or a single payload. In addition, the control platforms agent should coordinate with other agents when available resources cannot meet the demand of a new task.

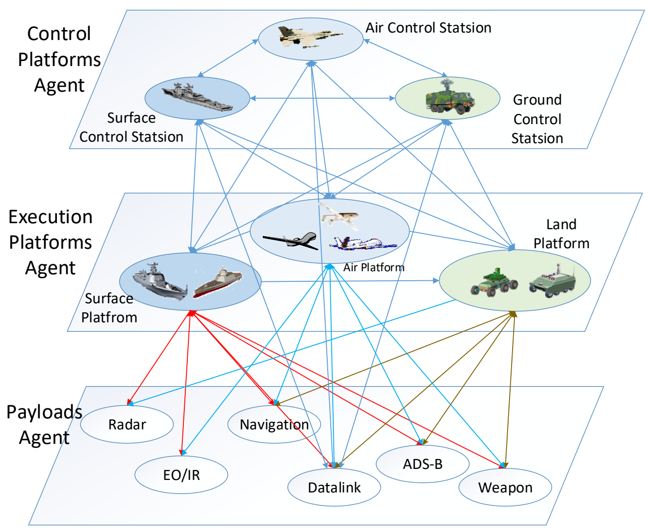

Fig. 1. Ccontrol relationship of the multi-level agent system

The primitive tasks are assigned to the execution platforms agent based on its platform type and payloads capability. When a new task is received, the priority of the new task should be compared with the priority of the task being performed, the high-priority task should be executed firstly. The priority of a task is determined by many factors, such as, the number of resources required by the task, the time required to perform the task, the execution status of the current task and the current status of the platform or the payloads. If the priority of two tasks is equal, the ongoing task should continue. Usually a primitive task that need a specific payload to execute shouldn't be assigned to a platform without the specific payload.

Payloads agent sets the work parameters based on the task requirements, reports its work status and sends task data periodically to the execution platforms agent. Payloads agent is the most basic unit of task execution.

Take the kill chain [7] as the example to interpret the task decomposition process. The kill chain is used to describe the orderly chain of interdependent links from target discovery to target engaging, including find, fix, track, target, engage and assess. However, these six steps is used just for command and control personnel to manage the battle progression, it's necessary to decompose the tasks and make the decomposed tasks executable for payloads, which is especially important for dynamic reconfiguration of the kill chain [8]. The kill chain tasks can be decomposed into search and detect task, electronic reconnaissance task, target identification task, target tracking task, electronic jamming task and missile task. The tasks mapping relationship with the kill chain steps is shown in Fig.2.

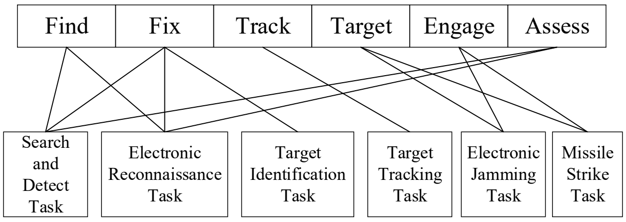

Fig. 2. Task decomposition of the kill chain

Each task can be represented as a set of task id, time, task area, target id, target position, task requirements and so on. The sequential order and dependency relationship of different decomposed tasks is represented in Fig.3.

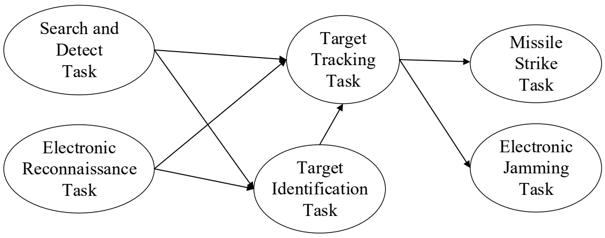

Fig. 3. Mutual relationship between different primitive tasks

III. IMPROVED CONTRACT NET PROTOCOL

The contract net protocol (CNP) is a distributed negotiation mechanism, proposed by Smith [9] to solve the task allocation problem in distributed artificial intelligence based on the market mechanism. It has been widely used in the production scheduling, multi-unmanned systems, resources scheduling [10], task collaboration [11], as the contract net protocol exhibits good dynamic performance and strong robustness. It's also easily extended to the large-scale mission scenarios.

The participants in the CNP are divided into tenderees and bidders. A task is generated and managed by a tenderee. The tenderee is responsible for selecting the successful bidder who can finish the task with a highest score. The bidder selects the best suitable task based on its status and received task request. The format of messages that interact between tenderees and bidders should be based on certain criteria to make the tasks understood clearly and the messages processed easily. The successful bidder is responsible for completing the task.

Usually the task allocation processes based on the CNP can be summarized as task publishing, bidding, contract signing and task execution, as shown in Fig 4. A new target or a new mission triggers the task publishing step. The tenderee needs to pack the primitive task information and evaluation criteria into a normalized message and broadcast the task message to all potential bidder.

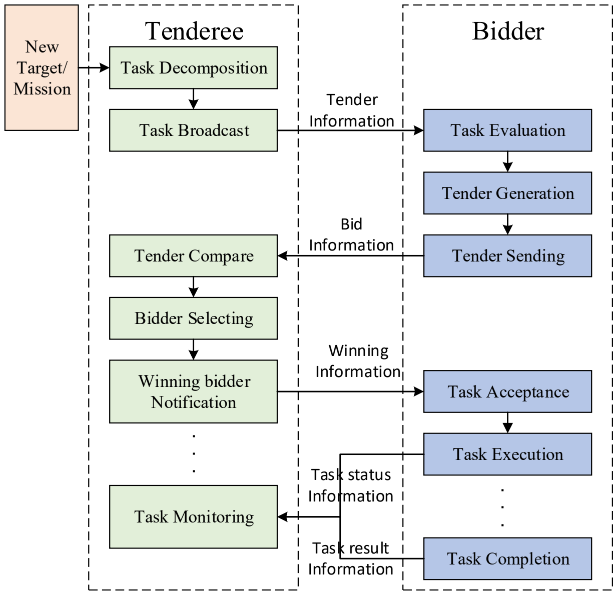

Fig. 4. Task allocation processes based on the CNP

When a task message is received by a potential bidder, the bidder should evaluate the task requirements with its capacity and current task status. Once the evaluation result shows that the bidder has the ability to execute the received task, it should generate and send a tender message to the tenderee as required. When the tenderee has received all potential bidder's tender messages or the deadline is arrived, the tenderee should compare all the tender and select the best one. Then the tenderee should send a notification to the winning bidder to confirm task performer. While the winning bidder executing the assigned task, it should send task execution status to the tenderee, which is monitoring the whole task process.

The basic CNP model is suitable for the scenario of single task and single winning bidder, and the efficiency will significantly reduce when the number of participants is large under the limited communication environment. A large number of potential candidates will increase the communication time and processing time, which will obviously affect the real-time capabilities of task assignment and resources allocation. We improved the CNP model from the following aspects:

- Divide all the participants into several clusters based on its type and position. Each cluster is controlled by a special platform, which plays the role of tenderee. When a new target is found, the control platform should select the suitable execution platform from its cluster firstly to perform the following tasks. Only when the suitable platform cannot be selected in the cluster, the control platform should coordinate with potential control platforms from other clusters or handover the target to other clusters to deal with.

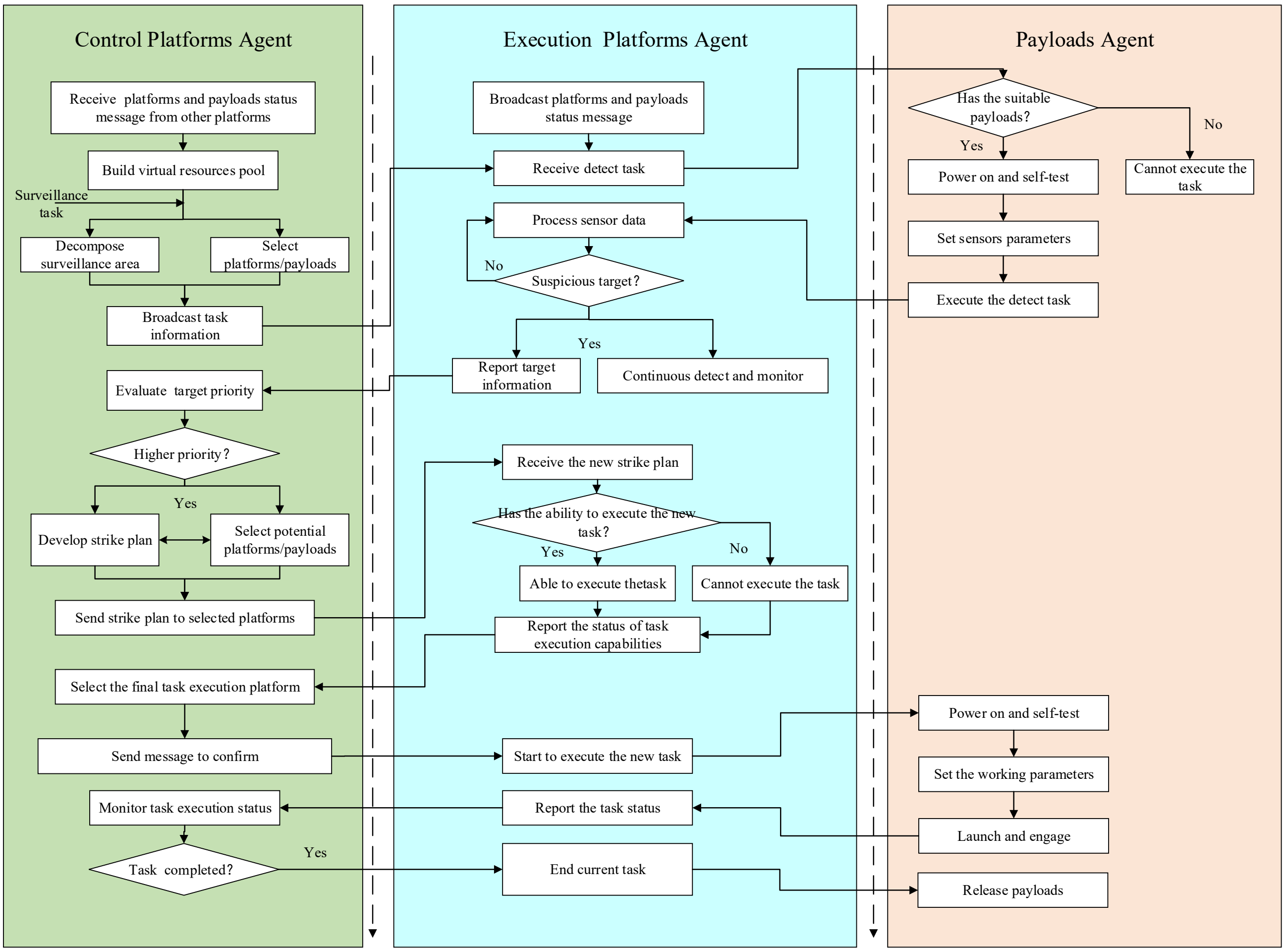

Fig. 5. Flow diagram of target-oriented improved CNP

- Send the tender information to the potential platform that has the suitable payloads to execute the task. The control platform needs to collect the basic capability information of all the platforms and payloads dynamic. Through the predefined mapping relationship between the primitive tasks and payloads, the selection of suitable platform can be done in an efficient way.
- Build the priority level model of tasks and dynamic adjust the use of resources based on the priority level. High threat level target needs to be processed as soon as possible, so the high-priority task has the right to use available resources and even the resources that are being used.
- Tender parallelly for multi-tasks triggered by a new target. Usually a target needs several primitive tasks to be dealt with, so the control platforms agent should start multiple tenders to form a combination of winning bidders. The target-oriented tender can guarantee the whole process of a target.

Each cluster can be presented as $Cluster^i = \{C_i, P_{i1}, P_{i2}, \cdots, P_{iM}\}$, where C_i denotes the control platforms agent in the cluster, P_{ij} denotes the execution platforms agent. Each platform can be presented as $P_{ij} = \{p_{ij1}, p_{ij2}, \cdots, p_{ijN}\}$, where p_{ijn} denotes the nth payloads of the platform P_{ij}. For a new target, the kill chain process can be decomposed into a tasks sequence $\{T_1, T_2, \cdots, T_n\}$ and each task T_i should correspond to a tender. The tender information can be described as a 5-tuple $\langle T_{id}, T_{type}, T_{info}, T_{time}, T_{condition} \rangle$, where T_{info} is the basic task information that the execution platform should know, T_{time} is the time requirements of the task, including the start time and end time, $T_{condition}$ is the task conditions. The bidder information can also be presented as a 5-tupe $\langle T_{id}, P_{id}, T_{cost}, T_{effectiveness}, P_{info} \rangle$, where the P_{id} is the bidder platform id, the T_{cost} is the cost to execute the task and the $T_{effectiveness}$ is the effectiveness that will be achieved when completing the task. Both the tender information and the bidder information should be packaged into a formatted message.

The flow diagram of target-oriented improved CNP is shown in Fig.5. The control platforms agent builds and updates a virtual resources pool by receiving platform and payloads status messages from other platforms. The accuracy and real-time of information in the virtual resources pool cannot be guaranteed as the communication is limited. But it still is an important reference information to select the potential platforms or payloads to execute a task. The execution platforms agent can acquire all the information of its payloads as the status information of all payloads is interacted internally, so the execution platforms agent has the only right to decide whether to accept a task or compute the cost of performing a task. The payloads agent changes the working status and sets the working parameters to perform the determined tasks. The working status of the payloads and task status are acquired in real time by the execution platforms agent to help to make dynamic decisions.

If the remaining platforms and payloads are adequate when a new target is found, the control platform can just select suitable platforms from available resources. When a necessary payload is being used and there is no other payload of the same type, the control platform should coordinate the use of the resource based on the priority level of the new task. If the new task has a higher priority level, the platform on which the payload is mounted should suspend the current task and start to execute the new task, and continue to execute the suspended task when the new task is completed. When there is no suitable payload in the cluster to execute the new task, the control platform should coordinate a platform with the specific payload to execute from other cluster.

IV. SEMI-PHYSICAL SIMULATION EXPERIMENT

To evaluate the effectiveness of the proposed method under the environment of limited communication, we establish a semi-physical simulation environment. Different types of platforms are connected to a wireless channel with datalink terminals and the wireless channel's attenuation can be controlled dynamically to simulate a limited communication environment. Different types of payloads are managed and controlled by the collaborative control terminal. The number of payloads amounted in a platform and the number of platforms in a cluster can be easily extended according to the task scenario. The control platforms can be pre-specified or automatically generated by algorithms.

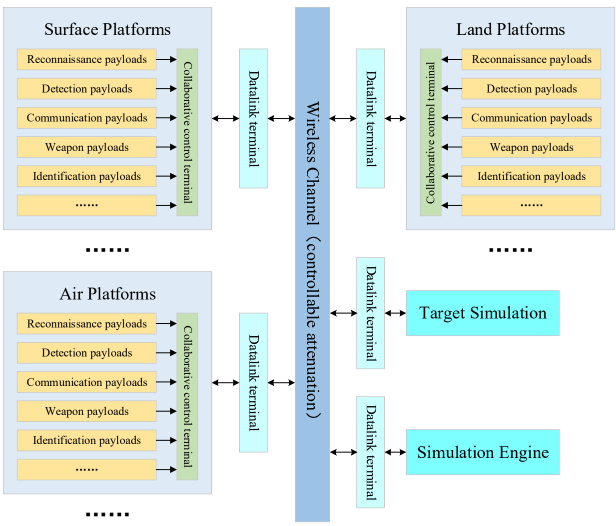

Fig. 6. The framework of semi-physical simulation environment

The target simulation can create different types of targets and simulate the actions and behaviors, which are the information inputs of the kill chains. The simulation engine is used to control the deduction speed of the task scenario.

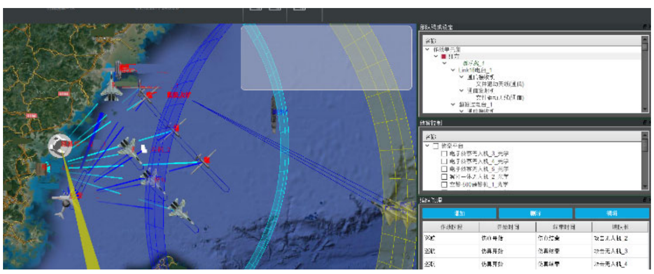

Fig. 7. Simulation result of the multi-targets scenario

The integrated situation of the typical task scenario is shown in Fig.7. We set two types of targets: scheduled targets and unplanned targets. For the scheduled targets, the resources allocation can be decided by the traditional optimization algorithms, while for the unplanned targets, the real-time performance of the resources allocation algorithms should adapt to the rapid change of the targets. The participating platforms and their payloads can be seen on the right of the figure. By using the improved CNP method, the appropriate resources can be allocated to every kill chain stage of a target and some resources can be adjusted to a higher priority target.

V. CONCLUSION

In this paper we use a multi-level agent system framework to describe the control relationship and collaboration relationship of different platforms, and on this basis, we use an improved CNP method to manage resources dynamically. The effectiveness of the proposed method is proved by the semi-physical simulation experiment. Further research is still needed in terms of the CNP messages protocols to shorten the interaction time and improve management efficiency.

REFERENCES

[1] D. Xue, B. Chen, W. A. Zhang, et al, "Kalman fusion estimation for networked multi-sensor fusion systems with communication constraints", Acta Automatica Sinica, 2015, 41(1): 203-208.

[2] Z.Li, "Research on cooperative control of multi-agent systems under communication constraints", [D], Yanshan University, 2021.

[3] N. Song, X. Hong, C. Zhou, "Formation control of netwoked multi-agent system with communication constrains", Journal of Dynamics and control, 2017,15(2):163-171.

[4] Y. CONG, "High-order Linear Multi-agent Systems Time-delayed Consensus Theory and Its Applications to Multi-UAVs Cooperative Control", Changsha: National University of Defense Technology, 2013.

[5] Z. Wang, S. Leng, K.Xiong, "Multi-agent resource allocation strategy for UAV swarm-based cooperative sensing", Chinese Journal on Internet of Things, 2023(7):18-26.

[6] C.Jin, H.Hu, et al, "Cooperative strategy of multipe unmanned aerial vehicles in limited communication environment based on reinforcement learning", Unmanned Systems Technology, 2022(5):12-20.

[7] Joint Chiefs of Staff. Joint targeting:joint publication 3-60[M]. Washington D.C.:Joint Chiefs of Staff,2013.

[8] L. Sun, Y. Chen, Z. Chen, "Invulnerability analysis of dynamically reconfigurable combat networks based on kill chain", Fire Control and Command Control, 2023(48):12-18.

[9] R.G.Smith, "the contract protocol: high-level communication and control in a distributed problem solver", IEEE Transactions on Computers, 1980(12):1104-1113.

[10] P.Jin, K. Li, "Distributed satellite resource scheduling based on improved contract network protocol", Systems engineering and electronics, 2022(44):3164-3173.

[11] X. Chen, B. Shen, et al, "MAV/UAV task collaboration method based on extended CNP", Unmanned Systems Technology, 2022(5):72-80.