

Negotiation Protocol Comparison for Task Allocation in Highly Dynamic Environments

K. Noack^a, L. Marsh and S. Shekh

^a Defence Science and Technology Group, Edinburgh, South Australia, Australia
Email: katherine.noack@dsto.defence.gov.au

Abstract: A robust negotiation protocol is required for a multi-agent simulation involving two adversarial teams in a highly dynamic and hostile environment. In this environment agent failure is possible due to a number of circumstances such as running out of fuel or being destroyed by other agents.

This paper compares three existing negotiation protocols: the Contract Net Protocol (Smith, 1980), the Distributed Contract Net Protocol (Cano and Carbo, 2006), the Extended Contract Net Protocol (Aknine et al., 2004) and two protocols developed by the authors (termed herein the 'Simple' and 'Hybrid' protocols). The objective of this paper is to determine which protocol is best suited to our application in terms of scalability, robustness against agent failure, communication overhead, and response time.

To evaluate these negotiation protocols an experiment was conducted, involving three different test cases, which varied the availability of agents at different stages of the negotiation process. In these test cases a team of software agents (the 'blue team') were tasked with destroying a number of stationary targets (the 'red team').

The experimental results showed that the Contract Net Protocol (CNP) was suitable for low risk environments due to its low communication overhead, while the Distributed Contract Net Protocol (DCNP) was more suitable for high-risk environments due to its greater robustness against agent failure. However, this robustness was achieved at the expense of greatly increased communication. An alternate approach that showed promising results was to use a Hybrid protocol that switched between CNP and DCNP depending on the environment. Additional work is required to develop the Hybrid protocol further.

Keywords: Negotiation protocols, task allocation, multi-agent systems, resource management

1. INTRODUCTION

Autonomous systems are of growing interest to Defence. As robotics, artificial intelligence and software development grow in maturity, it is anticipated that the shift from automation to autonomy will require a revolutionary advancement in command and control. A significant and challenging aspect in the command and control of autonomous systems is resource allocation and task management. The key to resource allocation and task management amongst cooperative agents is having a robust negotiation protocol tailored for the environment. The purpose of this paper is to compare and contrast a number of negotiation protocols with the aim of finding a protocol which works well in a highly dynamic and hostile environment.

In Section 2 we will describe the dynamic and hostile environment our project is based on, and detail the specific requirements our project has for a negotiation protocol. It will also cover common approaches to negotiation protocols in the literature. Additionally, Section 3 provides an overview of existing literature that compares and evaluates negotiation protocols. Section 4 describes the comparison of the candidate negotiation protocols we have selected, detailing the experiments and criteria used to measure each protocol. Section 5 includes a discussion of the experimental results and recommendations, and finally Section 6 summarises and presents the conclusions from the research.

2. PROBLEM DEFINITION

Our project is based on a multi-agent simulation concerned with the problem of coordinating a number of heterogeneous agents to achieve a main objective. The simulation consists of two adversarial teams ('red' vs. 'blue') in a highly dynamic environment where agent failure is possible due to circumstances such as running out of fuel or being destroyed by other agents. In order to coordinate the agent teams, a negotiation protocol was required that could handle this high-risk and highly dynamic environment. For a more detailed description of the project and simulation see Marsh (2013).

In selecting a negotiation protocol for this simulation environment, a number of competing requirements were identified including: minimal communication overhead; robustness to agent failure; scalability; a quick response time; and efficient and accurate allocation of agents. This paper focuses on communication overhead, robustness and task completion aspects.

The negotiation protocols chosen to be evaluated against our requirements were the Contract Net Protocol (CNP) (Smith, 1980), the Extended Contract Net Protocol (ECNP) (Aknine et al., 2004), and the Distributed Contract Net Protocol (DCNP) (Cano and Carbo, 2006). These protocols were chosen because they seem to address a number of our requirements such as robustness and communication overhead. These protocols are described in the following section.

2.1 Negotiation Protocols

The CNP (Smith, 1980) is a negotiation protocol that has been adopted by the Foundation for Intelligent Physical Agents (FIPA) (FIPA, 2001) and has been extended many times to address different requirements (Bozdag, 2008). The original CNP involves the auctioneer initiating the negotiation by sending a request for proposals to other agents. Participating agents then send their proposals, or bids, back to the auctioneer. The auctioneer assigns the agent whose proposal best meets the task requirements. When finished, the assigned agent sends a message to the auctioneer detailing their success or failure. We note that more than one agent may be assigned at a time. The advantage of the CNP is that it has a lower communication overhead compared to some other protocols and it is naturally load-balancing, as busy agents don't have to bid (Bedrouni et al., 2009). However, because the negotiation process is coordinated in the auctioneer, failure of this agent will cause the entire process to fail.

The ECNP (Aknine et al., 2004) is an extension of the CNP that has two phases of task proposal and allocation. This allows participating agents to propose temporary bids and update them if their situation changes. Agents can also participate in several negotiations in parallel. The ECNP has the advantages of being more fault tolerant than the CNP and allowing negotiations between many auctioneers and agents. The disadvantage of this protocol is the increased message traffic due to the additional allocation phase.

The DCNP (Cano and Carbo, 2006) is a decentralised variation of the CNP that does not have a central auctioneer role. Instead, the task allocation problem is solved through a global system rank that selects the most suitable agents for the task (Cano and Carbo, 2006). Using this protocol, after the initiating agent communicates the task, the global rank for this task is calculated by each participating agent after communicating their bids or capabilities to all other participating agents. The DCNP has the advantage of being more robust to auctioneer failure, at the cost of higher communication overhead due to increased

messaging between agents. The total number of exchanged messages for this protocol is $O(N^2)$, where N is the total number of agents (Cano and Carbo, 2006).

3. RELATED WORK

There are several papers that compare negotiation protocols. However, as far as the authors are aware, many of these papers do no compare the protocols quantitatively. Additionally, many focus on specific applications that are not relevant to our simulation. Some of the most relevant comparison papers are outlined below.

In Badica and Scafes (2011), three negotiation protocols were evaluated against negotiation outcome and communication complexity. The negotiation protocols evaluated were the Direct Task Assignment (DTA), the CNP and the Iterated Contract Net, in the context of agent-based disaster management systems. While this work varies the availability of contractor agents at different stages of negotiation, it does not look at the failure of the auctioneer. This is important to consider in our simulation of adversarial agents in a high-risk and dynamic environment.

Bozdag (2008) surveys extensions to the CNP that address some of the limitations, such as agents not submitting a bid unless they are idle. These extensions are qualitatively compared in the literature, according to a number of attributes, with the recommendation to perform an experiment in order to quantify results. Xu and Weigand (2001) also describe extensions of the CNP to four different contract model protocols and frameworks. This work discusses the capabilities and limitations of these different approaches, but does not quantitatively compare them. Finally, Dias (2006) addresses the challenges of market-based coordination approaches, and provides a review of the field, but does not evaluate specific negotiation protocols.

This paper differs from the work mentioned above in three ways. Firstly, we compare the negotiation protocols quantitatively, which was a recommendation in Bozdag (2008). Secondly, the protocols are evaluated against requirements specific to our application in high-risk environments and, thirdly, we compare some newer protocols, e.g. the DCNP. Existing comparison papers have typically focused on older techniques.

4. COMPARISON OF NEGOTIATION PROTOCOLS

A number of negotiation protocols from the literature were potentially suitable for our project. As it was unclear as to which protocol was the most appropriate for our purposes, we proceeded to evaluate the effectiveness of several promising protocols. This section describes the experimental setup and the results from the author's evaluation of the negotiation protocols.

The protocols from the literature that are compared are the CNP (Smith, 1980), the DCNP (Cano and Carbo, 2006), and the ECNP (Aknine et al., 2004). Two additional protocols are also evaluated, the Simple and Hybrid protocols. These two protocols were developed by the authors to address specific requirements. The Simple protocol is a variation of the CNP whereby the contractor does not communicate results back to the auctioneer. This was an initial attempt at reducing the communication overhead. The Simple protocol also allows agents to bid when they are busy. The Hybrid protocol attempts to gain benefits from both the CNP and the DCNP by switching between the two depending on the environment. For example, if an auctioneer or contractor does not receive a response, the agents infer that the environment is hostile and will switch from using the CNP to the DCNP, whichever is the more robust in the given context.

4.1. Experimental Setup

The five negotiation protocols were evaluated against a number of test cases, all variants upon a common scenario, in order to test specific requirements. In these test cases a team of software agents (the 'blue team') are tasked with destroying a number of stationary targets (the 'red team'). These test cases and agents were implemented in a multi-agent software simulation which executes agent and environment actions within a game tick. The simulation has a maximum time limit of 1000 game ticks.

In each test, a number of 'red' (targets) and 'blue' agents were randomly placed within a constrained geographical area and the detection of red agents was simulated at specified times throughout the 1000 game ticks. The initial location of agents, and the time red agents were detected (game ticks), were kept consistent for each protocol. Each protocol has been implemented according to details supplied in their respective papers. In our simulation, when a task is announced agents have two game ticks to respond with a bid. If no bids are received within the time limit, the task is re-announced for all protocols except the DCNP. The DCNP does not re-announce tasks due to the lack of auctioneer role after the initial task announcement. Additionally, the Simple protocol is the only protocol implemented that allows agents to bid when busy.

Test Cases

Three test cases were used to evaluate the negotiation protocols against our requirements. The basic test case (1) consists of a number of blue agents being tasked to destroy stationary red agents that are detected at specified times. The other two test cases are extensions of this test case, involving the destruction of blue agents, during the negotiation process, to test protocol robustness. For example, in the Destroy Auctioneer test case (2), the auctioneer agent is destroyed after announcing a task; and in the Destroy Contractor test case (3), the contractor is destroyed after being assigned a task.

Assumptions

The experimental environment consists of a number of assumptions, including:

- tasks can be completed by a single agent;
- agents have the same abilities and resources;
- agents cannot run out of resources;
- agents do not use scheduling;
- agents are cooperative and honest in their communication;
- agents can communicate with no failure or restrictions;
- any agent can be an auctioneer or contractor for a task;
- agents representing tasks (red agents) are stationary; and
- only one blue agent can be destroyed in a single time-step.

The assumption that agents don't use scheduling may have an impact on the results for the ECNP since the protocol requires bids to be computed according to task schedules (Aknine et al., 2004).

4.2. Results and Discussion

To determine how the algorithms are affected by the complexity of the problem we vary the number of blue and red agents (from 10 to 100, in increments of 10), and the probability of destruction (from 0 to 100 percent, in increments of 25). The destruction probability only affects test cases which involve destroying blue agents to simulate their destruction by red agents (test cases 2 and 3).

The appropriateness of a negotiation protocol to our problem is quantified as the total number of messages by the blue team, time taken to complete all tasks, length of auction process, and the percentage of tasks competed. Every test case variation (e.g. varying number of blue agents) was repeated 1000 times in order to obtain statistically significant results. The results below are averaged over these 1000 runs.

Table 1 shows the average length of the auction, for each protocol, from when the task was first announced to when the task was assigned to an agent. This is measured in game ticks. As shown below, all of the protocols had similar auction length, except for the ECNP (due to its additional task proposal and allocation phase). The ECNP also had greater variation in results because this protocol allows the repetition of earlier stages of the protocol if the requirements for task allocation aren't met. Approximately 17.38% of the auctions for the ECNP were longer than three game ticks with the maximum auction length being 188 game ticks. However, an extended auction process is rare with 0.109% of the ECNP results being greater than 10 game ticks. The DCNP showed no variation in results because tasks are not re-auctioned if the requirements for assignment aren't met.

Table 1. Length of auction from task announcement to task assignment.

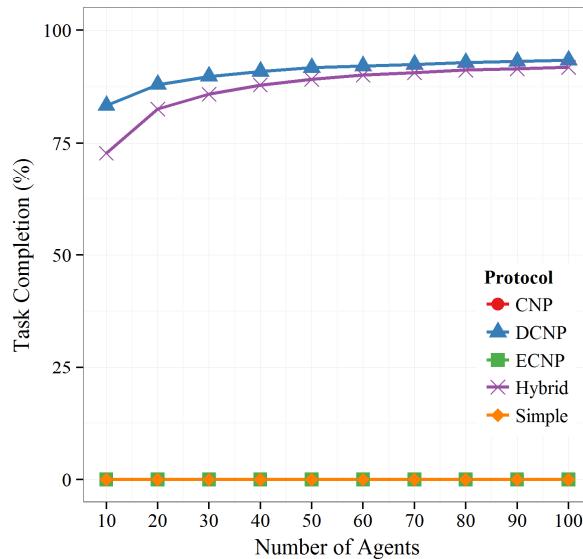
Protocol	Length of Auction (game tick)	Standard Deviation
CNP	2.00	0.088
DCNP	2.00	0.000
ECNP	3.24	0.743
Hybrid	2.00	0.007
Simple	2.00	0.089

Table 2 shows that all five protocols had very similar task completion for the Basic test case when the number of blue agents and the number of tasks are equal (e.g. 40 agents and 40 tasks etc.).

Table 2. Task Completion performance for the Basic test case when the number of tasks and agents are equal.

Protocol	Task Completion (%)	Standard Deviation
CNP	99.974	0.239
DCNP	99.971	0.242
ECNP	99.982	0.178
Hybrid	99.972	0.246
Simple	99.915	0.677

Figure 1 shows the task completion results for the Destroy Auctioneer (100% destruction probability) test case when the number of blue agents and the number of tasks are equal. The average of these results is shown in Table 3. For this test case only the DCNP and the Hybrid protocol were able to complete the tasks when the probability of destruction was 100%. As shown in Figure 1, the Hybrid protocol has a lower task completion compared to the DCNP for the Destroy Auctioneer test case. This is probably due to a delay when the Hybrid protocol switches from the CNP to the DCNP.

**Figure 1.** Task Completion performance for the Destroy Auctioneer test case when the number of tasks and agents are equal.**Table 3.** Task Completion performance for the Destroy Auctioneer test case when the number of tasks and agents are equal.

Protocol	Task Completion (%)	Standard Deviation
CNP	0.000	0.000
DCNP	90.700	4.850
ECNP	0.000	0.000
Hybrid	87.292	6.876
Simple	0.000	0.000

Figure 2 shows the number of messages (\log_{10}) for the Basic and Destroy Auctioneer test cases when the number of blue agents and the number of tasks are equal. The figure shows that as the number of blue agents and tasks both increase, the number of messages for the DCNP increases drastically, which supports the statement that the total number of exchanged messages is $O(N^2)$ (Cano and Carbo, 2006). Therefore the DCNP may be better suited to applications with fewer agents. As expected, the Hybrid protocol had a similar number of messages to the CNP in the Basic test case and a similar number of messages to the DCNP in the Destroy Auctioneer test case. This is due to the Hybrid protocol switching to the DCNP, when the agents

started failing in the Destroy Auctioneer test case, but remaining in the CNP for the Basic test case because no agent failure occurred.

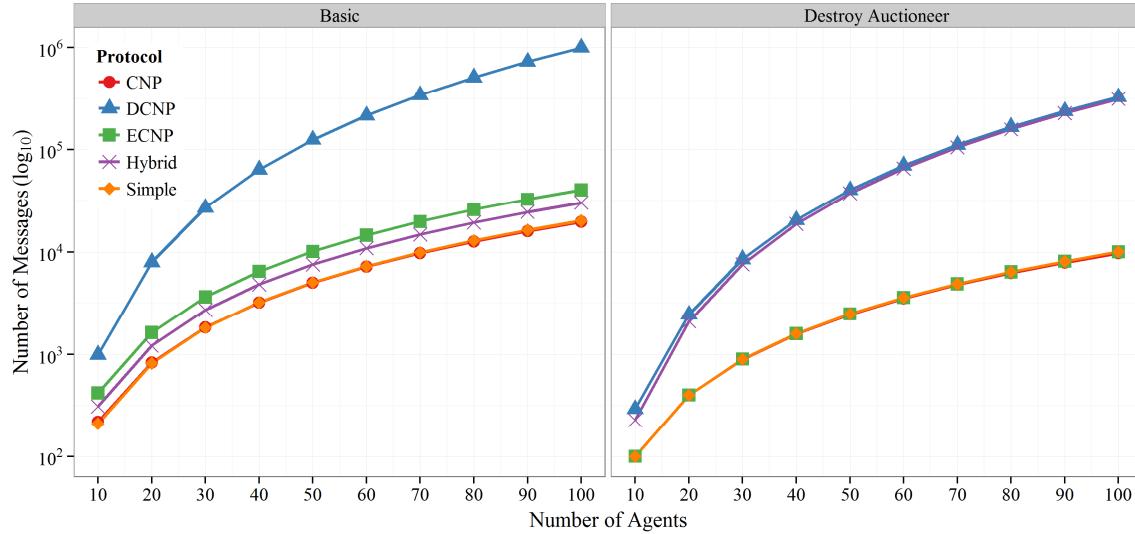


Figure 2. Message complexity of the protocols with an equal number of agents and tasks,

Table 4 shows the average number of messages for the Basic test case when the number of agents and tasks are equal while Table 5 shows this for the Destroy Auctioneer test case. Results are rounded to the nearest whole number.

Table 4. Average number of messages per protocol for Basic test case.

Number of Agents and Tasks	20	40	60	80	100
CNP	833	3234	7181	12632	19592
DCNP	7956	63731	215299	510688	997790
ECNP	1636	6467	14501	25728	40158
Hybrid	1221	4842	10863	19285	30108
Simple	820	3240	7260	12880	20100

Table 5. Average number of messages per protocol for the Destroy Auctioneer test case.

Number of Agents and Tasks	20	40	60	80	100
CNP	398	1582	3538	6256	9716
DCNP	2465	20405	69772	166490	326578
ECNP	400	1601	3602	6404	10006
Hybrid	2111	18673	65355	157824	312022
Simple	400	1601	3602	6403	10006

None of the protocols evaluated were able to handle the Destroy Contractor test case. If an agent was destroyed after being assigned a task, the task was never completed. Therefore, it's important that additional measures be considered to handle contractor agent failure.

5. RECOMMENDATIONS AND FUTURE WORK

The experiments have demonstrated that the CNP is a suitable protocol for low-risk environments due to its low communication overhead. In contrast, situations with a high possibility of auctioneer failure lend themselves to the use of the DCNP, which provides some degree of fault tolerance at the expense of greatly increased communication. However, since neither protocol is perfectly suited to our problem, the authors recommend combining the two protocols into a Hybrid protocol. Based on a preliminary implementation, the

Hybrid protocol appears promising, showing higher task completion than the CNP during high risk periods but lower communication overhead than the DCNP, which provides a degree of balance.

In both the CNP and DCNP, once a task has been assigned to a contractor, there is no assurance that the task will be completed, so contractor failure can ultimately lead to task failure. A simple solution is for the auctioneer to implement a timeout mechanism for re-auctioning the task if no results are received from the contractor after a set period of time. However, this could introduce unnecessary delays into the system, so an alternative may be to use a group of sentinels to monitor the system (Hagg, 1997). The sentinels provide an exception-handling mechanism, monitoring the system for undesirable events and acting accordingly to repair the system (Klein et al., 2003). In the case where contractor failure occurs midway through task completion, the sentinels can detect the problem and re-auction the task to the other contractors, thereby giving the task another chance for completion. In very high-risk environments where even the sentinels are at risk of failure, persistent sentinel (broker) teams can be used to monitor for sentinel failure and recruit new sentinels as required (Kumar et al., 2000). In general, any improvements to the CNP or DCNP should be carefully evaluated to ensure the functionality and performance of the original protocols is not compromised.

6. CONCLUSION

This paper provides a comparison of several negotiation protocols in the context of a highly dynamic and hostile environment. The results show that the CNP is the most effective protocol when the likelihood of risk is relatively low, while the DCNP is better suited to high risk environments, though at the expense of greatly increased communication. Future work could involve improving the performance of the Hybrid protocol, as well as exploring the use of sentinels to further increase protocol robustness in high-risk situations where even contractors are at risk of failure.

ACKNOWLEDGEMENTS

The authors would like to thank D. Gossink and T. Allard for their contributions to the project.

REFERENCES

- AKNINE, S., PINSON, S. & SHAKUN, M. F. (2004). An Extended Multi-Agent Negotiation Protocol. *Autonomous Agents and Multi-Agent Systems*, 8, 5-45.
- BADICA, C. & SCAFES, M. (2011). Experimental Evaluation of Service Negotiation Protocols in Agent Systems. 15th International Conference on System Theory, Controls, and Computing (ICSTCC), Sinaia. IEEE, 1-6.
- BEDROUNI, A., MITTU, R., BOUKHTOUTA, A. & BERGER, J. (2009). Coordination Strategies and Techniques. *Distributed Intelligent Systems*. Springer US.
- BOZDAG, E. (2008). A Survey of Extensions to the Contract Net Protocol. Delft University of Technology.
- CANO, J. & CARBO, J. (2006). Protocol for a truly distributed coordination among agents in competitive survival video games. *Developing Ambient Intelligence*. Springer Paris.
- DIAS, M. B. (2006). Market-Based Multirobot Coordination: A Survey and Analysis. *Proceedings of the IEEE*, 94, 1257-1270.
- FIPA. (2001). *FIPA Contract Net Interaction Protocol Specification* [Online]. Available: <http://www.fipa.org/specs/fipa00029/index.html> [Accessed 28 July 2014].
- HAGG, S. (1997). A Sentinel Approach to Fault Handling in Multi-Agent Systems. *Multi-Agent Systems Methodologies and Applications*. Springer Berlin Heidelberg.
- KLEIN, M., RODRIGUEZ-AGUILAR, J. A. & DELLAROCAS, C. (2003). Using Domain-Independent Exception Handling Services to Enable Robust Open Multi-Agent Systems: The Case of Agent Death. *Autonomous Agents and Multi-Agent Systems*, 7, 179-189.
- KUMAR, S., COHEN, P. R. & LEVESQUE, H. J. (2000). The Adaptive Agent Architecture: Achieving Fault-Tolerance Using Persistent Broker Teams. *MultiAgent Systems*, Boston, MA. IEEE, 159-166.
- MARSH, L., SHEKH, S., NOACK, K., GOSSINK, D. & ALLARD, T. (2013). A Multi-Agent System for Investigating Course of Action Planning. *20th International Congress on Modelling and Simulation (MODSIM)*. Adelaide, Australia.
- SMITH, R. G. (1980). The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, C-29, 1104-1113.
- XU, L. & WEIGAND, H. (2001). The Evolution of the Contract Net Protocol. *Advances in Web-Age Information Management*. Springer Berlin Heidelberg.