Performance Evaluation of Auctions WLAN for RoboCup Multi-Robot Cooperation

Mayez A. Al-Mouhamed[1] and Umair F. Siddiqi[2]

***Abstract*— A three-level architecture for a team of autonomous cooperative robots for RoboCup has been proposed. A dynamic Joint-Commitment scheme is proposed to support the formulation of relational behaviors for cooperative robots playing soccer. Passing the ball from one robot to another, receiving the ball and kicking, and handling exceptions invoke teamwork between two robots which know from each other that they are committed to a relational behavior. The joint commitment is established based on a finite state machine and a messaging system to: (1) synchronize the pass behavior, (2) reiterate the process and extend the pass to another partner, or (3) break the commitment and search for a new partner depending on dynamic game conditions. To provide dynamic joint-commitment and needed synchronization a fast, reliable, and power aware communication model is needed for Ad-Hoc wireless networks forming a cooperating multi-robot system. Current techniques are based on client-server, Publish/Subscribe, and Peer/Peer communication which are not suitable. For this we implemented and evaluated an auction based communication model based on (1) TCP Peer to Peer Scheme, (2) UDP Peer to Peer (UPTP) Scheme, and (3) UDP Broadcast and Token Passing (UBTP) scheme. Evaluation reports the distribution of auction completion times and power consumption for auction-based and peer-to-peer communication.**

***Index Terms*— behavior programming, distributed intelligence, multi-robot cooperation, robocup.**

I. Introduction

Distributed intelligence in multi-robot systems [1] is based upon the types of interactions exhibited: (1) the bio-inspired paradigm used in emergent swarms and mobile sensor network, (2) the organizational and social paradigm, and (3) the knowledge-based, ontological, and semantic paradigms. The domain space of distributed Intelligence establishes a hierarchy of interaction forms ranging from collective, cooperative, collaborative, and coordinative. The challenge is in the selection of the appropriate paradigm which dependents upon the specific constraints and requirements of the application of interest.

The multi-robot cooperative architecture is a general framework for implementing distributed artificial intelligent behaviors. A three-level functional architecture [2] is proposed for a team of cooperative mobile and autonomous robots playing soccer based on the joint intentions framework. The levels are (1) organizational, (2) relational, and (3) individual.

The *organizational level* determines a strategy with a goal, like attacking or defending, and a specific set of tactics to implement the strategy. Each tactic is a behavior assigned to a robot of the team. A tactic is chosen based on the current world state and internal robot state. In the *relational level* the robots negotiate (explicit or implicit) and eventually come to an agreement about some team/individual goal. Behavior assignments may also be temporarily modified as a result of inter-robot negotiations. The *individual level* handles own behaviors like the sense-think-act loop such as searching-ball, walking-to-ball, stand-behind-ball, kick-ball, etc.

A formulation of relational behaviors for cooperative real robots [3] has been proposed based on explicit teamwork between two team mates which know from each other that they are committed to the relational behavior and that they will not quit without informing the other team members first. It uses the Joint-Commitment Theory. In the ball pass a communication and synchronization framework is proposed in addition to two primitive behaviors, the (1) intercept and (2) aimAndPass behaviors, which provided successful coordinated execution.

However, to implement the dynamic cooperation a fast, reliable, and power aware communication model is needed when arranging the autonomous robots in an Ad-Hoc mobile Networks. Dynamic task allocation [4] in multirobot cooperation can be implemented as a distributed negotiation mechanism using an auction-based framework. Here the node interaction uses publish-subscribe communication model. This produces a distributed approximation of a global optimum of resource usage.

Extending RoboCup architecture to enable a multi-robot collaboration beyond communication limited to a soccer field is proposed in [5]. Here a wireless ad-hoc network is proposed to facilitate robots to cooperating in carrying out tasks such disaster recovery and emergency response. For each connected partition in the network, a routing algorithm determines a steward node for each destination. This improves current AODV routing which fails when there is no existing end-to-end path from source to destination.

A dynamic Architecture for Autonomous Federated Systems [6] is proposed by combining a dynamic, lightweight container with a flexible ”logical” layer. This allows (1) uploading and self-configuring modules at runtime, and (2) acquire and discard rules, at run time, that adapt the behavior of the application as needed.

Measurements of energy consumption in a visual sensor network are reported in [7] for CPU processing, flash memory access, image acquisition, and communication are characterized for different hardware states like sleep, idle, transmitting, and receiving, and webcam on/off. It reveals that wireless communication consumes a lot of energy.

(1) Department of Computer Engineering, College of Computer Science and Engineering (CCSE) King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia. mayez@ccse.kfupm.edu.sa

(2) Information Technology Center, King Fahd University of Petroleum and Minerals Dhahran 31261, Saudi Arabia. umairf@kfupm.edu.sa

Allocating a number of exploration tasks to a team of mobile robots in which each target location needs to be visited by a robot. Minimizing the sum of the travel costs of all robots for visiting all targets is an NP-hard problem. Auction-based algorithms [8] were first used for task allocation. The auction minimum total resource usage and minimum total time have also been evaluated in simulation [9] for object searching missions.

There is need for an Ad-Hoc Wireless Networking Communication Model that provides fast, reliable, and power aware features for the effective implementation of dynamic cooperative architectures in autonomous robotics. Specifically a fast and reliable collective communication must be part of the relational behavior to help implementing real-time joint-commitment in highly dynamic environments like soccer.

Here we propose a dynamic Joint-Commitment scheme to support the formulation of relational behaviors for cooperative robots. Teamwork between two team robots which know from each other that they are committed to the relational behavior. The dynamic nature are handled by a mutual messaging allowing the team members to: (1) synchronize the pass behavior, (2) reiterate the process and extend the pass to another partner, (3) break the commitment and search for another partner depending on dynamic game conditions. For this an effective messaging system is investigated. We review the client-server, Publish/Subscribe, and Peer/Peer communication models and propose an auction based communication model based on (1) TCP Peer to Peer Scheme, (2) UDP Peer to Peer (UPTP) Scheme, and (3) UDP Broadcast and Token Passing (UBTP) scheme. In the evaluation we study the distribution of auction completion times and power consumption for the above schemes.

The organization of this paper is as follows. In Section 2 background on multi-robot cooperation is presented. In Section 3 we describe our proposed auction-based joint commitment scheme. In Section 4 the experimental results are presented. We conclude in Section 5.

II. Background

A three-level functional architecture is proposed [2], [3] for a team of mobile and autonomous robots which are capable of carrying out cooperative tasks. Relationships among robots of the team are modeled using the joint intentions framework. The multi-robot cooperative approach is applied to a Robotic Soccer environment. The cooperative architecture is a general framework for implementing distributed artificial intelligence and intelligent control by using the concept of behaviors. Robot tasks are composed of subsumptive behaviors.

The organizational level establishes the current team strategy based on the (1) team state and (2) world state. Complexity is reduced by the decomposition of team strategies into individual behaviors, which in turn are composed of primitive tasks. Each strategy is a set of tactics. A running tactic represents an agent behavior that is assigned to a given robot at a given time. Examples of agents are the attacker, goally, supporter, defender, etc. The team state corresponds to the current set of behaviors under execution. The world or game state consists of (1) game situation and (2) evaluation of situation. The game situation describes the current game mode like kickoff, end-of-game, penalty-for, penalty-against, etc. The team evaluation of current game mode like (1) losing and close to the end of the game, (2) ball close to our goal, etc. The world-game state refers to what it has been achieved since the beginning of the game and how this may influence the selection of a tactic. Moreover, behaviors are assigned to the individual robots, after a selection from within behavior sets representative of alternative tactics for the strategy selected by the organizational level.

The organizational level determines a strategy and a specific tactic to implement the strategy. The strategy must specify not only the goal to be attained (e.g. attack, defense) but also criteria to check how close to the goal the team is. The tactic consists of behavior sets, whose elements are the behaviors assigned to each individual robot of the team. A tactic is chosen based on the current world state, but also on each agent's current internal state.

In the relational level the robots define their relationships by negotiating and eventually come to an agreement about some team and/or individual goal. Behavior assignments may also be temporarily modified as a result of inter-robot negotiations. The joint intentions framework (Tambe, Jennings, Cohen and Levesque), provides a foundation for team work modeling at the relational level of the architecture.

The individual level handles all the available robot behaviors which form the agent body such as search-ball, walk-to-ball, stand-behind-ball, kick, etc. A behavior corresponds to a set of goal-oriented primitive tasks which are sequentially and/or concurrently executed. A primitive task is a sense-think-act loop. The goal is to accomplish some objective like moving to a pose which includes a robot position and orientation. The sensing data is required to measure to progress in accomplishing the goal like the distance to an object.

At the robot architecture level, each individual robot is provided with all the three levels of the team functional architecture. However, the organizational level is only active in the current head robot and the other robots have disabled organization layer to provide some fault-tolerance. When the head has fault like loss of power or other, a new head is selected.

At the state machine level, the strategy is determined at the organizational level by a state-machine whose transitions are traversed upon the matching of specific world states, and whose states define the current strategy. Therefore, strategies change when the world state, as perceived by the team, changes. The tactic selection, including behavior selection, negotiation, and temporary behaviors modification, is implemented by relational rules at the relational level.

In our work the formulation is based on the Joint Commitment Theory for which the commitments among team mates are established in the relational behaviors. The above three layers are processed sequentially from the selection of a role, a commitment, and an individual behavior based on the robot's role and commitment. In the pass relational behavior two robots set up a long term commitment, in which several individual behaviors are executed. The pass relational behavior is based on the synchronization of both players' actions, which is achieved by communication, and the execution of their individual skills. One of the robots is referred to as the kicker; he starts having the ball and will try to kick the ball in the direction of the other robot, the receiver, who has to intercept the ball.

The Joint Commitment Theory is used to select relational behaviors. Predefined logical conditions can establish a commitment between two agents. Once a robot is committed to a relational behavior, it will pursue this task until one or more conditions become false, or until the goal has been accomplished.

The initiative for a relational behavior is taken by one of the agents, who sets a request for a relational behavior. A potential partner checks if the conditions to accept are valid. If so, the commitment is established. During the execution of the commitment the changing environment can lead to failure or success at any time, in which case the commitment will be ended. In general, a commitment consists of three phases: Setup, Loop and End. The setup and ending of a commitment are used for synchronization. Only in the Loop phase the robots select primitive behaviors concerning the commitment in order to achieve their joint goal.

III. Auction-Based Joint Commitment

The game status and other logical conditions may trigger the need for a joint commitment which lead two or more agents to cooperate, as part of a relational behavior, in a given task like the ball pass behavior. In general, a commitment consists of three phases: (1) *Setup*, (2) *Execution Attempt*, and (3) *End or Release*, where *Setup* leads to searching for a potential partner, *Execution Attempt* implements the commitment behavior with proper synchronization and interception, and *End* is to end the commitment or its interruption in the case of a dynamic change in game status.

Figure 1 shows the state diagram of the dynamic commitment. The initiator or kicker broadcasts an auction (broadcast $B\_Auction\_Rq$) announcing the detection of an opportunity for scoring and asking the receivers to make a bid ($Reply\_to\_Auction$) based on their game conditions like availability is some field area and visibility of the goal. The bids are received and analyzed by the initiator which may return a grant message ($Auction\_Select$) to the winner bid, which becomes the partner, and the other bidders become free. To re-evaluate the scene, the new states of the kicker and partner become *Aim and Pass* and *Standby*, respectively, in which synchronization is done through peer-to-peer messaging ($B\_Prepare$) for kicker and (B_Status) for partner. The dynamic game conditions may change for both the kicker and/or the partner which may lead to:

- If the positioning of the opponents changes and the kicker finds a way to directly kick the ball it must finish the commitment ($B\_Restart$) which causes the partner to become free,
- The kicker completes the pass of the ball (B_Pass) which causes the partner to change its state to *Intercept*, to attempt intercepting the ball, and dynamically kick the ball (B_kick),

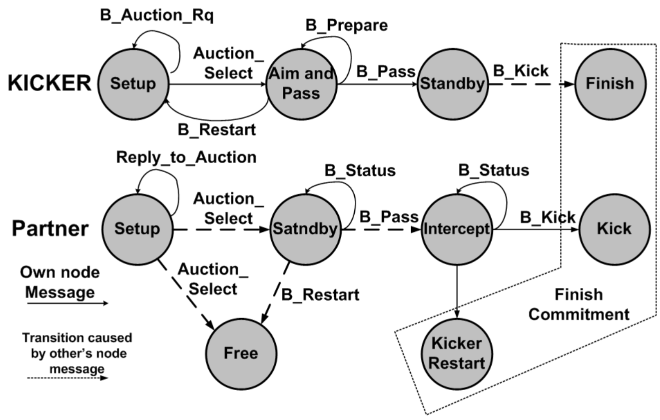

Fig. 1. Dynamic commitment based on auctions and peer-to-peer messaging.

- While intercepting the ball, the partner may find no opportunity to kick towards the goal. In this case, it restarts as a new kicker where a new commitment is to be attempted.

In the following we review the client-server, Publish/Subscribe, and Peer/Peer communication models and describe the proposed auction-based communication models.

A. Wireless Network Models

The wireless network can be developed using different networking models. In the following three network models are discussed.

1) Client/Server Model: In this network model, one node in the network is assigned as a server and other nodes as clients. The server generally performs the majority of the processing tasks. The clients initiate a connection with the server when they want to transfer instructions or data with the server. This model is not suitable for our Stargate based network because (1) in the network all nodes have equal computational power, therefore heavy computational load on one computer causes delays in the whole network, and (2) in the applications of the auction schemes any node can initiate communication (auction) with any number of the remaining nodes, so each node should have both client and server capabilities at the same time.

2) Publish/Subscribe Model: This model consists of publishers and subscribers. The publisher is unaware of the recipients of its messages and rather it publishes messages to a class of subscribers. The subscribers can receive messages from the classes in which they are interested without any knowledge about the publisher. In this way the publisher and subscriber are decoupled in this model. While this model looks suitable for the auction schemes, it has many extra features that are not needed by the auction schemes. First and foremost problem is that the target nodes for each auction are known to the initiator, and once an auction is done the list of nodes participated in the last auction becomes unimportant for other nodes because successive auctions cannot always be interrelated. Secondly, the target nodes for any auction are determined dynamically by the initiator based on the application requirements. Therefore, classes in this model need to be changes dynamically for each auction or too many classes need to be formed to meet

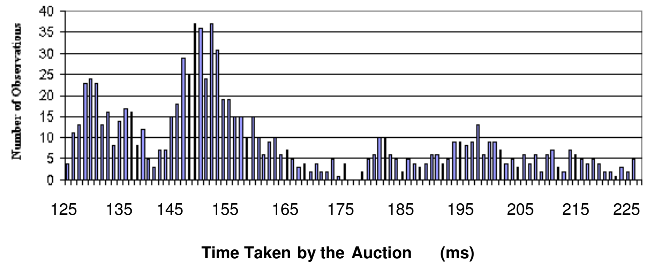

Fig. 2. Distribution of TCP based auction with scattered times.

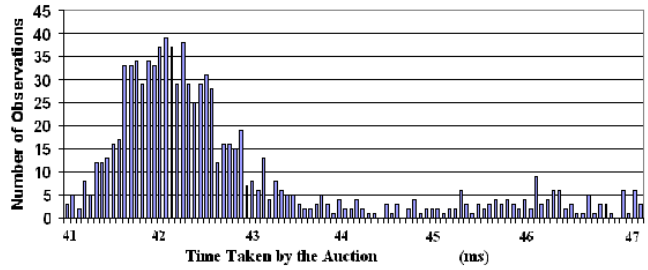

Fig. 3. Distribution of UPTP based auction

requirements of each auction. That is an unnecessary burden for short communications like used in the auction schemes.

3) Peer/Peer Model: In this network model, each node can connect to any other one or group of nodes and send/receive the data. The connections are ad-hoc and they lasts until the initiators or any other nodes want to terminate the connection. Features can be added into the basic Peer/Peer model to add functionality needed by the application. Peer/Peer model can use both TCP and UDP protocols. The following features of the auction schemes are best implemented in this network model: (1) The communication among nodes the auction is very short, (2) The behavior of each node is controlled by the node itself so there is no need for continuous data transfer, (3) Many features like broadcast or multicast can be used in Peer/Peer networking, and (4) Any node can initiates communication with any other node or group of nodes using multicasting.

B. Auction Algorithms

In the auction one node is assigned as a head node and other nodes are assigned as nodes only. The auction is always requested from the head node and other nodes only reply to the auction or perform synchronization related tasks if required by the auction scheme. In the following we present three schemes using UDP and TCP communication in different ways to accomplish the auction.

1) TCP Peer to Peer (TPTP) Scheme: In this scheme the head node communicates with each node that it wants to include in the auction using TCP packets. The auction request and reply from the node are both performed over the same TCP communication link. This scheme is also reliable but it is not scalable as the connection is repeated for each node.

2) UDP Peer to Peer (UPTP) Scheme: In this scheme the head node sends UDP packets to each node that it wants to include in the auction. The packet contains destination node IP, IP of the source node and resource ID of the resource for which the auction is being performed. The head node performs auction in a sequential manner, i.e. the first node and contacted first through the UDP packet and after it has responded through a UDP reply only then the head node contact the next node for the auction. If any node does not respond then the head node take-up that node and go the next node after (N) retries. This scheme is reliable if the value of N is kept large enough ($N = 10$) but it is not scalable.

3) UDP Broadcast and Token Passing(UBTP) Scheme: In this scheme the head node generates a UDP packet that it broadcasted to all nodes in the network. The packet contains the node ID of all nodes that it wants to include in the auction, the order of sequence in which the nodes should reply and the Identity Number of the resource for which the auction is being preformed. The node that is mentioned at the first position in the sequence order first replies to the head node and at the same time initiates a token packet. The token from the first node is forwarded to the second node in the sequence. The second node replies to the head node and transfers the token to the third node in the sequence. The third node responds in the same way and forwards the token to the fourth node. Similarly, all remaining nodes reply to the auction and forward the token. If any node fails to forwards the token then the next node automatically reply to the head node after a time-out of T milliseconds. The head node concludes the auction if it receives replies from all nodes otherwise it will broadcast a new auction request including the IDs of the nodes from which no bids were received so far. The process is repeated until all nodes responded or the originator gives up.

In the following we describe the hardware used and evaluate the distribution of the completion times of TPTP, UPTP, and UBTP schemes reports their power consumption.

IV. EVALUATION

The hardware consists of Stargate boards with wireless networking cards. The Stargate board is a powerful single board computer that consists of Intel 32-bit, 400 MHz XScale processor and 96 MB of memory in terms of SDRAM and Flash. The Stargate also have a daughter board that contains socket for the wireless card and Ethernet interface. The software of the Stargate comprises of Linux OS with drivers for all peripherals and Java Runtime Environment (JRE). In our experiments each Stargate have an Ambicom IEEE 802.11b wireless card. The wireless card has an additional 64 MB of memory for storing drivers and program files. In the following we study the above TPTP, UPTP, and UBTP schemes which are evaluated over the above Stargate system forming an 8-node wireless ad-hoc mobile network (WAHMN). The WAHMN is assumed to be be one single domain. Each auction is being generated by one head node and transmitted to all the other seven nodes. The head node can be any node in the above WAHMN.

The Distribution of completion times for the TPTP auction is shown on Fig. 2. The cost of having reliable communication is offset by the large (150 ms) and scattered completion times which may extend beyond 200 ms. Furthermore, the TCP communication has very high overhead that is not suitable for short communications like auction request and reply.

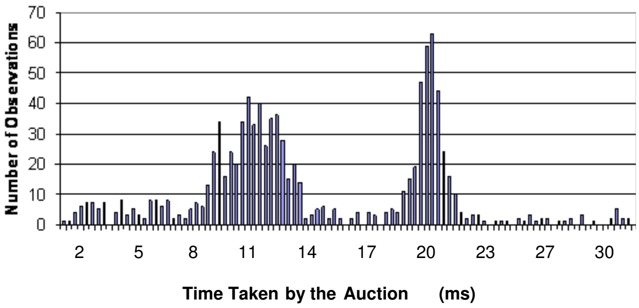

Fig. 4. Distribution of UBTP based auction

	TCP	UPTP	UBTP
AUCTION Range (ms) Power (mW)	[157, 208] 38.3	[44, 46] 9.4	[14.6, 23] 7.9
PEER-TO-PEER Range (ms) Power (mW)	[22, 29] 5.5	[6.3, 6.6] 1.34	

TABLE I
AVERAGE, 95% BOUND, AND POWER CONSUMPTION.

The Distribution of completion times for the UPTP auction is shown on Fig. 3. The typical auction time is clustered in narrow range from 41 ms to 47 ms. The cost of having reliable communication is offset by the large (150 ms) and scattered completion times which may extend beyond 200 ms.

The Distribution of completion times for the UBTP auction is shown on Fig. 4. In 40%, 55%, and 5% of the cases all the nodes responded after (1) the first auction (first cluster centered at 12 ms), (2) the second auction (second cluster centered at 21 ms), and (3) the third auction. One needs three auctions to make sure all the nodes of our WAHMN have been reached and successfully replied to auction.

It is clear that the UBTP scheme is performing better than TPTP and UPTP due to its shorter completion time while providing comparable reliability to the others. UBTP has also the advantage of being scalable with to arbitrary number nodes because it repeats the auction only when some nodes did not reply as opposed to the case of TPTP and UPTP for which the auction is repeated for each node.

Direct measurements of the power consumption using the TPTP, UPTP, and UBTP schemes reveals that the average energy consumptions per auction are 38.3 mW, 9.4 mW, and 7.9 mW, respectively. Table I summarizes the average completion times, the bound for 95% of times, and the power consumption for auction and peer-to-peer communications.

V. CONCLUSION

We described a dynamic relational behavior (ball passing) for autonomous cooperative robots playing soccer. The relational behavior allows: (1) synchronizing the pass behavior, (2) reiterating the pass the process and extend the pass to another partner, (3) breaking the commitment and searching for another partner depending on dynamic game conditions. To handle dynamic game situations an effective auction-based communication model is proposed based on (1) TCP Peer to Peer Scheme (TPTP), (2) UDP Peer to Peer (UPTP) Scheme, and (3) UDP Broadcast and Token Passing (UBTP) scheme. In the evaluation we study the distribution of auction completion times and power consumption for the above schemes. TPTP based communication has large and scattered time overhead and lack scalability for this environment. The stability and responsiveness is improved by using UPTP. However, the shortest and most stable times were obtained for UBTP. Therefore UBTP and UPTP are best to implement auction-based and peer-to-peer messaging needed in the implementation of proposed dynamic commitment scheme (Fig. 1).

VI. ACKNOWLEDGEMENT

The authors acknowledge computing support from King Fahd University of Petroleum and Minerals (KFUPM).

REFERENCES

[1] L. E. Parker. Distributed intelligence: Overview of the field and its application in multi-robot systems. *AAAI Fall Symp. on Regarding the Intelligence in Distr. Intelligent Systems.*, 2007.

[2] P. Lima et al. A functional architecture for a team of fully autonomous cooperative robots. *Manuela Veloso and Hiroaki Kitano, editors, RoboCup-99: Robot Soccer World Cup III, Lecture Notes in Computer Science, Springer-Verlag, 1999.*

[3] B.V. der Vecht and P. Lima. Formulation and implementation of relational behaviours for multi-robot cooperative systems. *Proc. of 8th RoboCup Inter. Symposium, Lisbon*, 2004.

[4] B.P. Gerkey and M.J. Mataric. Sold!: auction methods for multirobot coordination. *IEEE Trans. on Robotics and Automation,*, 18:758 – 768, 2002.

[5] A. Weitzenfeld et al. Beyond robocup: Ad-hoc networking for autonomous mobile robots. *International Conference on Mechatronics Technology*, 2006.

[6] A. Frei and G. Alonso. A dynamic lightweight architecture for ad-hoc infrastructures. *Third IEEE Inter. Conf. on Pervasive Computing and Communications (PERCOM05)*, 2005.

[7] C.B. Margi et al. Characterizing energy consumption in a visual sensor network testbed. *2nd Inter. Conf. on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM'6)*, 2006.

[8] M. G. Lagoudakis et al. Simple auctions with performance guarantees for multi-robot task allocation. *Proc. of the Inter. Conf. on Intelligent Robot and Systems*, 2004.

[9] A.R. Alejandro and Montano. L. Comparative experiments on optimization criteria and algorithms for auction based multi-robot task allocation. *IEEE Inter. Conf. on Robotics and Automation*, 2007.