Profiling Pseudonet Architecture for Coordinating Mobile Robots

Sriram Raghavan, Ravindran B
Department of Computer Science & Engineering
Indian Institute of Technology Madras,
Chennai, INDIA
Email: {srisai, ravi} @cs.iitm.ernet.in

Abstract— ***Area coverage*** **and** ***Navigation*** **are two fundamental requirements for robot applications. When multiple robots are fielded in a scene, coordination through communication becomes a natural pre-requisite. This paper focuses on the area coverage problem and proposes periodic exchange of state information related to** ***location*** **and** ***coverage*** **as a solution. The solution is based on Pseudonet communication architecture that enables exchange of messages between the robots by setting up a Bluetooth piconet or scatternet and maintaining the same throughout the process of covering the area.**

Keywords-Communication Architecture, Multi-agent System, Area Coverage, Distributed Coordination

I. INTRODUCTION

For inter-robot communication, wireless technology holds the key, as it endows the robots with the necessary degrees of freedom to be mobile. Over the years research related to robotics has witnessed significant growth in algorithms that empower robots to understand their environment, which is often hostile. Time has come when this body of algorithmic knowledge has to be coupled with wireless communication technologies which are becoming more sophisticated and powerful. In this paper, we describe an architecture called Pseudonet for communication and coordination among robots and show as to how it can be used effectively in area coverage problems. Pseudonet is equivalent to the "middleware" in typical computing environment as it bridges the communication requirements of the *robot-centric algorithms* to *modern sophisticated wireless technologies*. For physical transmission between the robots, we emphasize Bluetooth wireless technology, as it combines low-power, moderate-coverage and high speeds - which reflect the requirements in the robotic world.

The functional block diagram depicting the requirements of a Multi-Robot application scenario is shown in Figure 1. As mentioned earlier, Area Coverage and Navigation are integral to applications involving multiple robots and coordination among the robots. For example, to cover a given area using multiple robots, one needs the following:

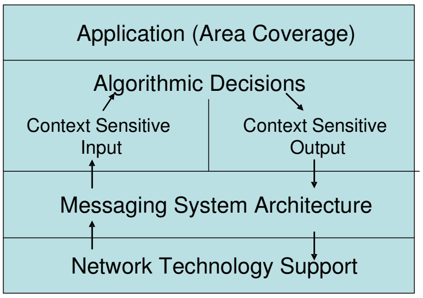

Figure 1. Functional Block Diagram of a typical Multi-Robot Coordination Scenario

- *Algorithm* to decide the next step(s) to cover – such a decision needs "awareness" on the part of the robot about the "environment" and "context".
- *Messaging* facility which enables coding of "environment" and the "context" by each robot and ability to communicate the same to the other robots through appropriate communication protocols.
- *Network Technology Support* for physically communicating the messages to the other robots with an appropriate choice of topology (star, tree, mesh, etc.) and mode (unicast, multicast or broadcast).

In area coverage applications, each robot takes part in the distributed discovery of the next position through exchange of messages. Each application-level action is translated into actions to be performed by each robot. In order to achieve this, the *coordination algorithm for coverage* must take each robot through a series of steps such as *localizing itself*, *selecting the next action* and *communicating* the same to other robots. This calls for direct communication between robots in order to maintain global knowledge. Besides, area coverage application requires that such transitions by each robot should result in covering the entire area – that too with a rider that the overlap in coverage is maintained near zero.

In order to conform to these stringent requirements of the application, the algorithms (that execute within each robot) should decide the next state of the robot based on a context-sensitive input. This has led to the discovery of several algorithms which suit different contexts. To enable the robots to participate in the context-sensitive algorithmic decision making, messaging system architecture is essential. Typically messaging system architecture consists of the definition of different types of messages to suit (different) contexts and their exchange sequences or protocols to realize the meaning associated with the algorithmic actions.

In the physical plane, the robots send their context information to all other robots – resulting in the familiar broadcast scenario. Besides, area coverage applications tend to cover larger areas by fielding robots in smaller colonies. So the physical transmission technology should have the natural ability to support a broadcast domain for each of the colonies and integrate the broadcast domains into single distributed scenario. Bluetooth enables broadcast in a robot colony through piconet and integrates broadcast domain by sharing slaves or masters across piconets.

Our emphasis in Pseudonet is to meet these functional requirements which help in the area coverage problem. In this paper, we explain the relationship between the functional block diagram described in Figure 1 and the Pseudonet architecture in detail.

The paper is organized as follows: Section 2 presents the State of the Art in Multi-agent Coordination Architectures. Section 3 presents the Pseudonet architecture and describes its packet structure and message types. Section 4 provides an overview of Multi-robot area coverage using Pseudonet. Section 5 describes the applied broadcast strategy used in Pseudonet and Section 6 provides an area coverage case study using one particular coverage algorithm. Section 7 concludes by giving a brief summary of the work done.

II.Related Work in Multi-agent Architectures

Multi-agent systems, in general, provide a framework for obtaining *control and data related information* from applications, support *distribution of processing* and handle *complexities arising out of scaling.* An example of this framework is the WARREN system based on RETSINA architecture [9]. The WARREN multi-agent system can be considered as a multi-user, distributed information gathering system and is widely deployed in management of financial portfolios.

Parker describes the ALLIANCE software architecture [7] that provides *fault-tolerant cooperative control in heterogeneous mobile robots* for performing missions composed of loosely-coupled tasks having ordering dependencies. ALLIANCE allows the robots to select the subtasks depending on mission requirements, robot current state and capabilities, and external environment conditions.

Gerkey and Mataric describe the Auction-Based MURDOCH architecture [3] for *multi-robot coordination.* The MURDOCH architecture is based on a simple distributed negotiation protocol that allocates tasks via a sequence of first-price one-round auctions. The process is triggered by the introduction of a task to the system and proceeds through five steps.

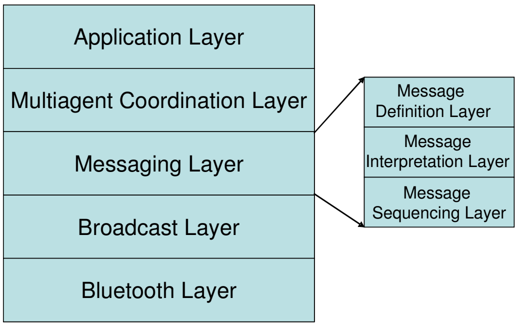

Figure 2. Pseudonet: A Multi-agent Coordination Architecture

Farinelli et al. [1] present a comprehensive survey of multi-agent robot coordination techniques and classify them based on the *coordination and system* dimensions; where *coordination dimension* takes care of cooperation, knowledge, coordination, and organization and *system dimension* tackles issues related to communication, team composition, system architecture, and team size.

A typical rescue task involves a team of robots to scan and cover an area in real-time, whose topology is potentially unknown. There are three dimensions in which mobile robot coordination can be classified. They are: *Coverage Problem*, *Coordination Problem*, and *Communication Problem*. The work by Cao et al. [8] provides a classification of the domain of multi-agent robotics along the dimensions of communication, computation and other capabilities. Butler et al. [5] describe algorithms that guarantee coverage of rectilinear environments by a team of robots.

In our paper, we profile a multi-agent coordination architecture called Pseudonet and solve the area coverage problem in a multi-robot setting. The solution is arrived at using a distributed approach and employing inter-robot communication as the only coordination tool. We illustrate the execution of an area coverage algorithm using Pseudonet explaining the messages exchanged in the process. Our work integrates Bluetooth Architecture with Pseudonet to support simple Bluetooth enabled mobile robots to perform area coverage by exchanging state information periodically.

III.Pseudonet – A Multi-agent coordination architecture

The Pseudonet architecture, as conceived by us and as used in our work, consists of 5 layers (See Figure 2). Of these, the uppermost layer is the application layer that is responsible for carrying out the ultimate application goal. The aim is achieved using multi-agent coordination function calls from the Multi-agent Coordination Layer. The area coverage problem is one such application. The next layer, the Multi-agent Coordination Layer consists of a series of multi-agent coordination algorithms that are responsible for guaranteeing task completion to the multi-agent applications they support. The algorithmic decisions are executed a through messaging function calls. The Messaging layer forms the messages corresponding to the algorithmic decisions handed down by the multi-agent coordination layer. It is also responsible for selecting the appropriate message type to send depending on the state of the robot. The Pseudonet Broadcast Layer follows the Messaging layer and it deftly combines a fully acknowledged transfer for a robot transmitting a message to the robot perceived as Master by the Piconet. The Bluetooth layer is the most fundamental layer, responsible for setting up the Bluetooth Piconet among the robot team members in order to facilitate robots to communicate with each other.

A. Pseudonet messages & packets

Pseudonet Packet Structure: The Pseudonet packet structure is built using four packet types, viz. Information packet, Synchronization packet, Acknowledgement packet and NegativeAcknowledgement packet as shown in Figure 3. The state information related to location and or coverage for each robot is sent using an information packet.

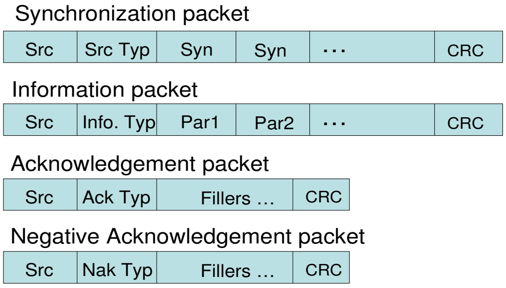

Figure 3. Pseudonet Packet Structure

When a robot initially synchronizes with the master robot in the piconet, the synchronization packet is used to provide information related to team size and the hopping-frequency information. While this hopping frequency information is a lower-layer operation, it is essential to ensure that each robot is ready to receive the polling messages. The acknowledgement and negative-acknowledgement packets are used whenever the applications require reliable information transmission. The source address is encoded in 3-bits to uniquely identify the sender and type field is encoded in 2-bits. As all messages are broadcast to the entire robot team, a separate destination field was not included in the design.

Pseudonet Message Types: The Pseudonet multi-agent architecture supports five types of information messages that provide state information about a particular robot. As a robot is moves within the coverage region and covers the cells, it periodically sends a *'state'* message using information packets that provides robot specific context information. The periodicity of this message depends on the robot coverage rate and the choice of coverage algorithm.

If a robot is surrounded on all sides either by covered cells or by other robots, it sends a *'help'* message to request support. A *help* message contains the location information in addition to the directions evaluated for future movement and the reasons for their failure. If any other team member has information about that location, it responds with an *'answer'* message directing the robot to the nearest uncovered cell.

When a robot discovers that it has revisited one or more cells often (this is achieved by maintaining a recent history of the visited cells), it sends a *'trap'* message which contains the trapped location sequence and requests an *answer*. In the absence of an answer message, the robot would arbitrarily select a direction and move k steps $(k > 1)$ along that direction to forcibly break loop to continue coverage. This ensures that a robot does not wait indefinitely for a non-existent response. A robot that covers a fixed ratio of cells within the area as programmed a priori sends a *'stop'* message to indicate its departure from the team. This message forcibly takes the robot to its dormant mode.

All messages (excluding the state message) are triggered by poor decision sequences on part of the robot/ team. These messages constitute communication protocol overhead and therefore need to be minimized.

The Pseudonet architecture can be used to compare algorithms by measuring the number of such messages generated during coverage. Such an analysis will provide a common baseline for comparison on the efficiency of algorithms. While Pseudonet supports only five message types for the area coverage application, it can be extended to other multi-agent applications by suitably modifying the information packet.

IV.Overview of Multi-robot area coverage Using Pseudonet Architecture

The area coverage operation using Pseudonet comprises of four phases: They are the initialization phase, the state exchange phase, the algorithmic decision phase and the termination phase. The actual exchange of messages in each phase will depend on

- nature of the area coverage problem,
- assumptions made about the environment; and
- capabilities of the robots

A. Different phases in Pseudonet Operation

The *initialization phase* involves robots discovering their neighbors, setting up a network and exchanging messages to decide which robot covers what part of the grid. After completing this phase the robots independently move to their respective locations to begin coverage.

During coverage, it is essential for the robots to exchange information related to location and coverage periodically so that the algorithm may permit them to select the next location in a manner that improves the efficiency (reduces overlap). To do this, the robots go through the *state exchange phase* followed by the *algorithmic decision phase* in which each robot selects an action as directed by the coverage algorithm.

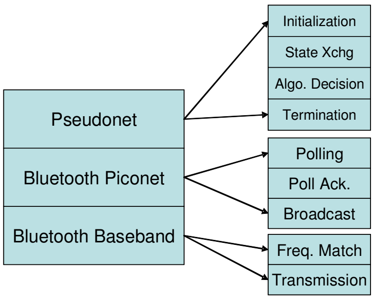

Figure 4. Different Phases in the operation of Pseudonet for Multi-Robot Coordination

After covering each cell, the robots collectively interact to understand completion status (of coverage) by exchanging information related to coverage of the grid up to that point. Since the total number of cells within the grid is known a priori, the robots can identify completion and it forms the *termination phase*.

All these four phases together constitute the Pseudonet Multi-agent architecture as shown in Figure 4. The state machine representing these stages is depicted in Figure 5. For each step taken at the Pseudonet level, the underlying technology layer, viz., Bluetooth handles three steps, viz., Polling, Poll acknowledgement and Broadcast. All message exchanges in Pseudonet are unacknowledged broadcasts by design and when viewed within the piconet. Each Pseudonet message constitutes a Bluetooth packet.

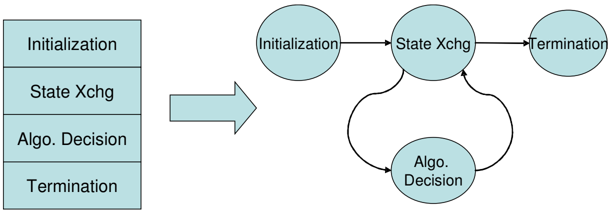

Figure 5. State diagram for Pseudonet Operation

Since the master is the initiator of communication, when it generates a packet, it sends the packet via the Bluetooth physical link as a broadcast along with a polling message (Poll step) inviting the next robot to transmit. The robot acknowledges the poll request by communicating its state information (Poll Acknowledgement step) to the master in the subsequent time slot. The acknowledging robot is always a slave device and it communicates its state information via the master (Broadcast step) to the team of robots.

In case a particular robot does not respond to its poll message, the master cognizes its absence from the team and informs other team members to alter team size appropriately. Such a robot may have either lost synchronization with the piconet frequency sequence or may have been destroyed due to unforeseen circumstances. In either case, that robot is unlikely to rejoin the team and therefore the remaining robots cover the grid in its assumed absence.

All packet transmissions occur through the Bluetooth Baseband layer that is responsible for identifying the next hop-frequency for transmission, hopping to that frequency (Frequency match step), synchronizing with the Bluetooth master's clock and physically sending the packets through the wireless medium (Transmission step). The scenario described hitherto maps Broadcast topology of Pseudonet to the piconet of Bluetooth, subject to the upper limit of 8 robots.

B. Pseudonet Setup at Bluetooth Layer

Pseudonet initialization phase occurs when the robots discover each other through the Bluetooth services provided as part of the Logical Link Control & Adaptation Protocol (L2CAP) layer in each robot. The initiating robot requests connection setup with the neighboring robots using a L2CAP layer's *connection_request()* message. The Host Controller Interface (HCI) translates these calls into procedures that perform the required signaling for detection and connection with the neighboring Bluetooth devices, i.e. robots. The Link Manager (LM) and Link Controller (LC) modules are then invoked to perform device inquiry and paging to setup the initial piconet.

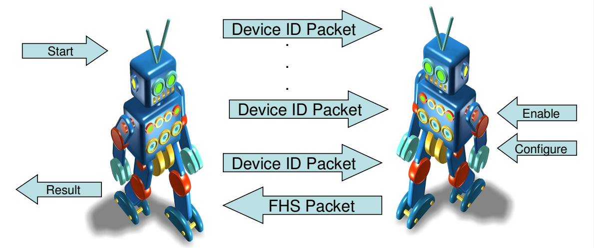

Figure 6. Pseudonet Initialization Phase - Discovery

To perform inquiry, the inquiring device broadcasts the Generic Inquiry Access Code (GIAC), common to all robots. The hopping sequence for this procedure is random and performed over 23 channels with a hopping rate of one every 2048 time slots (1 time slot = 625 µsecs). The GIAC transmission is sent in one-half slot and the inquiring device listens in the other half-slot at the associated hop frequency.

The inquiring device hops twice per time slot and the time for processing an inquiry packet is also half the time take taken as compared to a normal data packet. The scanned robot, on receiving the code (or part of it) sets a random back-off before responding to the inquiring robot. The back-off mechanism is employed to suppress multiple responses on the same frequency channel.

When the robot backs-off, it is tuned out of the channel and hence cannot receive any more messages. On re-entry, it waits for another GIAC. This is necessary to fully synchronize with the inquiring robot. On synchronization, it responds with a Frequency Hop Synchronization (FHS) packet. This requires the inquiring robot to keep inquiring for a long period, often longer than the period of random back-off. The idea behind increasing the duration of the inquiry procedure is purely to maximize the chance of coinciding with neighboring robots. Since this robot always listens on the corresponding response channel, it remains ready to receive the FHS packet and process it.

In order to connect two robots successfully it is first necessary to synchronize the frequency sequences before connection may be established. During paging, a robot attempts to connect to its neighbor (detected during the inquiry phase or known a priori) by sending an ID packet whose semantics is a simple connection request. The hop sequences are decided when the inquiry is performed and the devices are aware of the frequencies on which they may transmit or listen. In order to be able to respond, a robot needs to listen to the paging request on the same frequency.

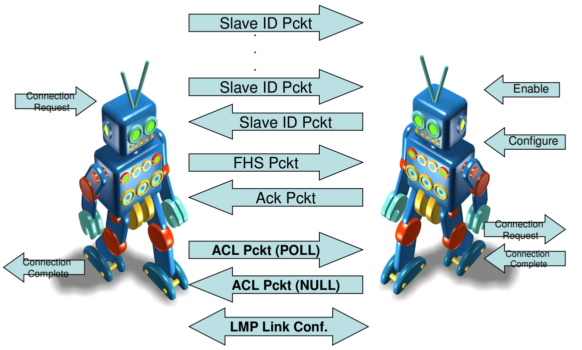

Figure 7. Pseudonet Initialization Phase - Connection

The scanning robot starts a device time and then starts off a periodic scan when it elapses. The robot thus performs periodic page scans of specified duration and at specified intervals. If the page scanner receives an ID packet during this period, it immediately replies with another ID packet having its own device address. The robot device address detected during the inquiry phase is unique. Multiple responses, interferences and packet losses are thus avoided.

The pager robot, on receiving the response packet, realizes that the page scanner is ready for receiving the pager's FHS packet and hence sends it. This FHS packet contains the necessary information for the page scanner robot to synchronize with the pager device by extracting CLK, and the AM_ADDR values. The page scanner device acknowledges the FHS packet with another ID packet. Now the two devices are ready synchronize and they move to the Master's hop sequence and synchronize with the Master's clock.

V. SUPPORTING BROADCAST IN PSEUDONET

In order that the robots are aware of what the other team members are attempting to achieve, it is essential to communicate effectively. Within a Bluetooth piconet, the master is the initiator of transmission and all slaves within transmit only to the master. Hence, any one-to-one communication between two robots, of which neither is a master, must take place through 2 hops. As mentioned in section 3, area coverage application requires broadcast support for minimizing overlap. Supporting broadcast at the application level through multi-step unicasting at the network level will result in linear increase in delay with team size. It is therefore necessary for broadcast support even at the network level. Within a Bluetooth piconet, a master can broadcast to all slaves at the same time with increased reliability due to repeated transmissions. If a robot sends its information to the master, the master can broadcast it to all the robots in one go. Pseudonet supports a 2-hop broadcast strategy.

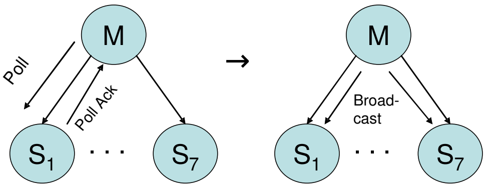

Figure 8. Pseudonet Broadcast Using Bluetooth Support

On establishing the piconet between all the robots, the frequency hop sequences are communicated. This paper assumes that all robots remain active in communication throughout the duration of coverage. The assumption is required to maintain synchronous hopping among the robots to support broadcast. The master initiates the information exchange sequence by communicating its state and desired future state along with a poll message to the slave robot in the first time slot as per the sequence communicated a priori (during piconet setup). The state information is encapsulated in an information packet and the poll message as a synchronization packet. Since all slaves receive this information, the master's action is global. In the slot following, the respective slave communicates its state to the master through an information packet. The master acknowledges that message and broadcasts the same to all in the subsequent slot. The master also sends a poll the next slave to obtain its state information. This broadcast now makes global the slave's state and desired future state.

An illustration of this method is shown in Figure 8 where slave S1 transmits its information to the master M during its period and this information is broadcast to all in the next time slot by the master. This process of polling, response and broadcast is repeated until all slaves are polled and the global state is known. Figure 9b shows a comparison between the delays experienced in unicast and Pseudonet broadcast modes.

Broadcast Algorithm for Bluetooth Master to perform State Information Exchange

---

```
Broadcast state, action information to all devices in piconet
For each slave, do
        Send poll to next slave
        Obtain response (state, action) from slave
        Interpret response message
        Broadcast slave's (state, action) information to all
        devices in piconet
```

```
End of For Loop
Repeat steps if previous state not equals current state
```

Figure 9a. Algorithm for Bluetooth Master in Pseudonet

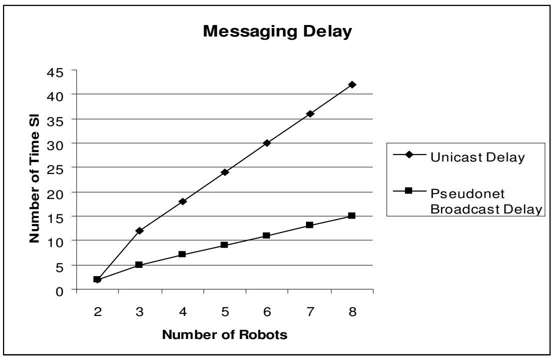

Figure 9b. Messaging Delay in Bluetooth for Unicast & Broadcast

VI. AREA COVERAGE CASE STUDY

Consider a scenario where a team of M homogeneous mobile robots capable of Bluetooth wireless communication are deployed for covering an area made of L^2 cells, where a cell is the footprint of a single robot. Let the robots employ a coverage algorithm called the *One-step-Communicate-All-Robots-Discovery* (OSCARD) in which each robot can detect other robots and the cells visited by the robots. A robot will communicate at every step with the team to decide on the next best step for movement. This algorithm is given in Figure 10.

OSCARD Coverage Algorithm

```
Given GRID of size (XMax, YMax) and M number of robots
Initialize: Spread the robots randomly in GRID.
For each pair of robots (i, j),
    compute $d_{ij}$ = Manhattan distance (robot i, robot j)
Repeat while coverage not complete,
For each robot, do
  Let (x,y) represent the current coordinates of the robot
  Avoid ALL previously covered cells; Identify and evaluate action list. Select action that doesn't reduce dij
  Communicate with other robots to exchange state, action information.
  Compute the 1-step new states for all robots and check for overlap.
  If no collision, communicate accept message and change state
  On collision, omit action from action list. Re-evaluate action list
  On trapped, choose random direction and break-loop
  If boundary along X-axis, select direction of which (YMax – y) or y is greater.
  If boundary along Y-axis, select direction of which (XMax – x) or x is greater.
  Else select an action with $d_{ij} \geq 2$ and change state.
End of For-loop
END of Algorithm
```

Figure 10. One Step Communicate – All Robots Discovery (OSCARD) Algorithm

In this algorithm, at each step, a robot localizes itself with respect to the area and identifies the available directions for its movement. Then it selects one among the available actions and communicates the same with the team. This communication is sent as broadcast to the team.

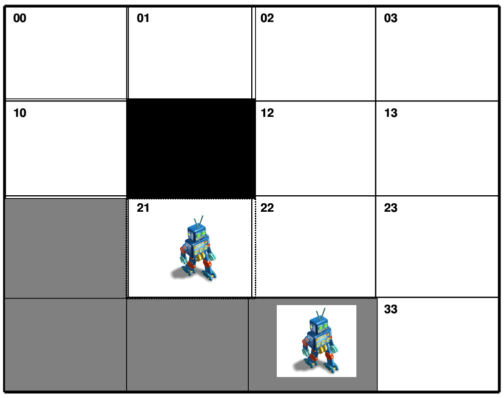

Figure 11. Intermediate Coverage Scenario using OSCARD algorithm

Consider one such intermediate coverage scenario depicted in Figure 11 where shaded cells represent covered regions. We explain the interplay between the Algorithmic Decision Layer and the Messaging Layer for the robot in the typical case of next action selection. For the next action selection, suppose the robot (bottom in figure) chose to move north, it sends its state information with *move_north* on its action field. The robot may have taken this decision after analyzing its action space and eliminating cells that have already been covered. Among the remaining actions, it uses some arbitrary decision mechanism to select the *move_north* action. Another robot intending to move to the same location raises an '*objection*' to that action when it transmits its state information. Once the state exchange phase completes for all robots, those having conflicting actions will undergo one more round of message exchanges to state their reason for selecting that action. The messages exchanged between the robots in this phase belong to the *help-answer* category where each robot claims ownership over the next state it wishes to move to. In this situation, a robot which is constricted on its movement is given priority to persist with the action while the other robots modify their actions by re-evaluating the action space. If more than one robot remains after this round of message exchanges, then the robot with a lower ID is arbitrarily chosen to persist with that action while the others are forced to alter their actions. A robot that admits to modify its action sends an *answer* message confirmation.

Each message exchanged is encapsulated in an information packet and sent with the source ID, list of evaluated actions (total of four actions for each robot), reasons for action failures, selected next action and the CRC. The acknowledgement and negative acknowledgement packets from robots other than the master robot were omitted in the implementation of this algorithm to simplify analysis. The transmission of each of the packets requires the robots to synchronize with the master robot and transmit the information packet. The master robot in turn forwards this packet to all the robots within its piconet. In order for Pseudonet to support data transmission across scatternets, the master robot of each piconet must be a slave robot in at least one other piconet.

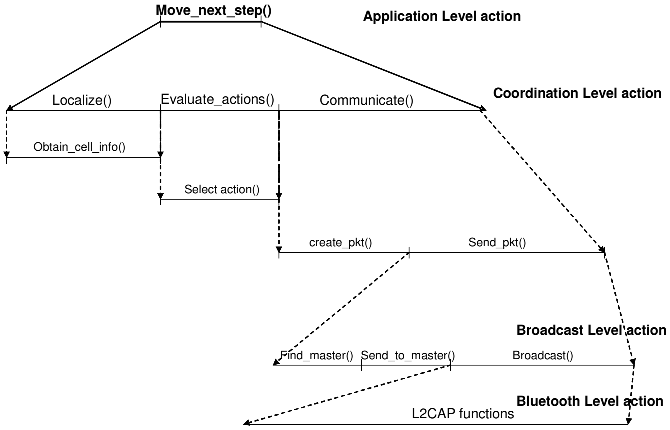

Figure 12. Action Decomposition in Pseudonet

We now describe how an application level action is mapped on to the lower level actions in Pseudonet. Consider an action such as the *move_next_step()* shown in Figure 12. From the application layer, this action gets translated into multiple service calls at each of the lower layers. It uses the services of from the procedures like *localize()* and *evaluate_actions()* at the coordination layer to determine its position and select pick the next action in each of the robots. The *localize()* action enables the robot to estimate distances to the boundaries. Evaluate_actions() scans the action space of a robots and omit already covered cells depending on the algorithm used. OSCARD, for example, omits all previously covered cells and chooses from a smaller list of uncovered cells for its next move. Each robot then communicates its state and action information to the team using the communicate action. To perform this action, an *information* packet is created and the message is embedded suitably before sending to the other robots.

If the robot gets into a *trap* or *covered* state for an extended interval of time, it sends a *help* packet to the team. This action, in turn will invoke the Broadcast layer service call to obtain the master device ID and send the information packet to it through the Bluetooth layer. This service call warrants the master of the piconet to be identified using the *find_master()* service at the broadcast layer and transmitting the packet to it. On receiving the message, the master robot acknowledges its receipt and broadcasts it to the entire team using the *broadcast_to_robots()* service call. The Bluetooth layer, on its part, uses the services provided as part of the L2CAP layer to synchronize the various robots at their respective frequency slots and send the messages to all. Each robot interprets the message appropriately using the Pseudonet Messaging layer and proceeds for coverage. In the case of a help message, a receiving robot will generate another information packet with answer as its content and sends it to the robot in need of the information. This communication is point-to-point (through the Bluetooth master) to avoid unnecessary traffic in the network. In this manner, a higher-level action such as the move_next_step is mapped onto various service calls at the lower layers in the Pseudonet Architecture.

VII.Conclusion

In this paper, we profiled a 5-layered Multi-agent Coordination Architecture called *Pseudonet based on Bluetooth technology*. We illustrated its use in a multi-robot area coverage application scenario and described its specifications. We addressed the associated issues and challenges in maintaining synchronous behavior across the robots during this application. Finally, we showed how Pseudonet can be used to study and improve the effectiveness of multi-agent coordination algorithms taking advantage of its different message types and provided a case study using one coverage algorithm.

In this paper, we have assumed an ideal scenario from the point of view of network behavior. Relaxing this assumption will require reassessment of the Bluetooth network and newer solutions. The authors are currently working in this direction. Another extension of this work involves assessment of system requirements in the case of heterogeneous networked robots for the same application. Efforts are currently in the works to provide new and novel solutions.

REFERENCES

[1] Farinelli A., Iocchi L., and Nardi D., “Multirobot Systems: A Classification Focused on Coordination”, *Systems, Man, Cybernetics-Part B*, *IEEE Transactions on,* Vol. 34, No. 5, Oct. 2004

[2] Farinelli A., Iocchi L., and Nardi D., “An analysis of coordination in Multi Robot Systems”, *Systems, Man, Cybernetics*, *IEEE Transactions on,* pp. 1487-1492, Washington DC, Oct. 2003

[3] Gerkey B. P., and Mataric M. J., “Sold!: Auction methods for multi-robot coordination”, *Robotics and Automation, IEEE Transactions on, Special Issue on Multi-robot Systems*, Vol. 18 No. 5, pp 758-768, Oct. 2002

[4] Parker L. E., Fregene K., Guo Y., and Madhavan R., “Distributed heterogeneous sensing for Outdoor Multi-Robot Localization, Mapping and Path Planning”, Proceedings *NRL Workshop Multi-Robot Systems*, pp. 21-30, Washington DC, 2002

[5] Butler Z. J., Rizzi A., and Hollis R. L., “Complete Distributed Coverage in Rectilinear Environments”, Proc. of the *Workshop on the Algorithmic Foundations of Robotics*, A. K. Peters, Jan. 2001.

[6] I. Rekleitis, G. Dudek and E. Milios, "Graph-based exploration using multiple robots", Proceedings of the *Fifth International Symposium of Distributed Autonomous Robotic Systems (DARS)*, October 4-6, 2000, Knoxville, Tennessee, pp 241-250, 2000

[7] Parker L., "ALLIANCE: An architecture for fault-tolerant multi-robot cooperation", *Robotics and Automation, IEEE Transactions on*, Vol. 14 No. 2, pp. 220-240, 1998

[8] Cao Y. U., Fukanga A., and Kahng A., “Cooperative Mobile Robotics: Antecedents and Directions”, *Autonomous Robotics*, Vol. 4, pp. 1-23, 1997

[9] Corkill D. D. and Lesser V., “The Use of Meta-Level Control for Coordination in a Distributed Problem Solving Network”, Proceedings of the *Eight International Joint Conference on Artificial Intelligence*, pp. 748-756, 1983