Multi-Robot Systems: Extending RoboCup Small-Size Architecture with Local Vision and Ad-Hoc Networking

Alfredo Weitzenfeld, *Senior Member, IEEE*, Luis Martínez-Gómez, Juan Pablo Francois, Alejandro Levin-Pick, Katia Obraczka, *Member, IEEE*, Jay Boice

***Abstract*—In this paper we describe preliminary results from a collaborative effort between ITAM's Robotics Lab and UCSC's Internetworking Research Group (i-NRG) focusing on extending the Small-Size League RoboCup system architecture. More specifically, our goal is to enable multi-robot collaboration beyond the limits of a soccer field environment. To this end, we have been developing a local vision wireless ad-hoc network architecture that will make it possible for robots to cooperate in carrying out tasks such as disaster recovery and emergency response. We present results from initial robot experimentation using ad-hoc networking while discussing future work.**

***Index Terms*—Autonomous Robots, Ad-Hoc Networking, RoboCup, Small-Size League, Search and Rescue.**

I. Introduction

In order for robotics to have an impact in real world applications researchers have to overcome extensive challenges from single robot designs all the way to multiple robot architectures. In such efforts it is common to exploit developments from various fields beyond robotics, such as artificial intelligence, biology, software engineering, human-computer interfaces, etc.

In this paper we present work resulting from a new collaboration between researchers in robotics at ITAM and researchers in networking at UCSC in the design of new communication protocols for the coordination of multiple mobile robots. This work highlights not only the inter-disciplinary nature of this research but also points out challenges in the individual domains. In the case of robotics, we are extending robotic architectures developed originally for RoboCup [1] by incorporating additional processing capabilities than required for the official competitions. In the networking domain we are extending protocols developed originally for networks with uninterrupted connectivity with new capabilities to make them applicable to scenarios with frequent and long-lived connectivity interruptions.

From the robotics perspective, the Robotics and Biorobotics Laboratories at ITAM are involved in the development of biologically inspired models to test hypothesis on animal behavior and their linkage to neuroscientific studies. These models are helping the development of new adaptive architectures such as rat-inspired learning and its application to robot exploration [2]. Additionally, in the context of RoboCup, ITAM's Eagle Knights competes in a number of soccer leagues including Small-size and Four-Legged where robots are programmed and in certain cases also built by the participating teams. RoboCup also includes non-soccer competitions. One noteworthy example is search and rescue known as RoboCup Rescue [3].

In recent years robots have demonstrated their usefulness in supporting life-threatening human tasks. Among these, Urban Search and Rescue (USAR) [4] has been an area where robotics is starting to have an important impact [5]. For instance, robots can play a crucial role in searching and rescuing survivors trapped under buildings collapsed due to major disasters such as earthquakes. One of the main challenges in these rescue operations are posed by the unstable nature of the collapsed structures, hard to reach spaces, lack of oxygen, and hazards resulting from fire, toxic gases, or other chemicals. In the past, specialized sensory equipment has been used in assisting rescuers, yet this technology is mainly used from outside the disaster perimeter. In the case of rescue robots, they are usually remotely operated, resulting in a number of limitations:

(a) The number of robotic devices required to control a large-scale search and rescue operation is significant, requiring a large number of trained human controllers.

(b) Coordination between human-controlled, teleoperated robotic devices is hard, limiting the possibility of shared decision support systems.

(c) Poor environmental conditions, such as low visibility, make human maneuvering of robotic devices difficult.

(d) Teleoperation relies on continuous availability of robust communication channels and power sources, including the use of wirelines.

In order to get closer to survivors, scientists are currently experimenting with mobile robots with various shapes, sizes and capabilities [6]. One unavoidable challenge is that search and rescue robots must become more autonomous while interacting with human controllers only for higher-level decision making. Robots can help in the overall search and rescue operation. In addition to producing maps of how to reach a survivor's location, robots will help in asserting survivors' conditions and existing hazards. A key consideration in carrying out these rescue missions will be the ability for robots to communicate with base stations even if far away. Ad-hoc networking will play an increasingly important role in such sparsely connected multi-robot systems.

1-4244-0537-8/06/$20.00 ©2006 IEEE

The i-NRG lab at UCSC is currently involved in several ad-hoc sensor networks related projects. Like the Eagle Knights Small-Size RoboCup team, these projects involve the integration of custom-built hardware with ad-hoc network protocols specifically designed for the environments in which they are used, as well as the data that is to be delivered. Experience with each of these projects is being leveraged into the Eagle Knights project. The following are descriptions of some of these projects.

The CARNIVORE system [7] (Carnivore Adaptive Research Network in Varied Remote Outdoor Environments) was born from the desire to further understand the interplay between coyotes, their predators and their ecosystem in the Santa Cruz mountains. Custom collars have been developed that contain a 3-axis accelerometer, GPS, storage space, and communication capabilities. Collared coyotes will continually sense and transmit data to static base stations deployed in the area, and the data will later be aggregated and used in analysis of their behavior. Similar to the Eagle Knights project, the network topology is quite sparse, resulting in a network that is rarely connected. Similar mechanisms will be used to ensure that messages are delivered in a timely fashion to the sink nodes.

Meerkats [8] is a battery-powered wide-area surveillance system incorporating both sophisticated vision algorithms and a power-management scheme to enable long network lifetime. Unlike the Eagle Knights project, the Meerkats network is static, allowing the use of more traditional ad-hoc networking. Detailed analysis of power consumption has enabled the network to be designed such that lifetime is maximized. Power monitoring enables a distributed resource manager to instruct nodes to turn on or off their components such as wireless card and USB camera.

The SEA-LABS project [9] (Sensor Exploration Apparatus utilizing Low Power Aquatic Broadcasting System) has been designed to monitor remote coral reefs. This project, since it is also battery-powered, must adhere to strict power-consumption guidelines in both sensing and transmission. A successful deployment in the Monterey Bay has provided initial data, and a full deployment in the Midway Atol is planned for the future. The devices, since they are used in such extreme environments, must require minimal maintenance and extremely long lifetime. Furthermore, the harsh environment and large distance between nodes (up to 8km) requires that the networking be designed with reliability as a key consideration.

These are just a few examples of mostly sensor networks, both static and mobile. In the case of multi-robot systems for disaster recovery and emergency response applications, robot teams collaborating in rescuing or reconnaissance operations need to be deployed in arbitrarily wide areas with tortuous terrain and subject to communication impairments such as interference, noise, signal fading, etc. Thus, new extensions to robots as mobile sensor networks are required to take into account stringent and adverse environmental conditions in search and rescue scenes. Thus, the initial goal of the existing collaboration between ITAM´s Robotics Laboratory and UCSC's i-NRG, is to add ad-hoc networking capabilities by extending the existing multi-robot platform.

The remainder of this paper is organized in three major sections, namely: Section II describes extensions to existing ITAM's Eagle Knights RoboCup Small-Size architecture by adding local vision and ad-hoc networking capabilities; Section III discusses current work at UCSC in developing protocols for environments with episodic connectivity; Section IV presents preliminary results from an experimental testbed composed of static and mobile nodes evaluating the ad hoc networking protocols for frequent and long-lived disconnection; finally, Section V presents conclusions.

II. Multi-Robot Coordination

This section overviews the RoboCup Small-Size league architecture and presents extensions to the individual robots necessary to provide local vision capabilities and support for ad-hoc networking.

A. RoboCup Small-Size Robot Architecture

RoboCup competitions initiated 10 years ago and have become a well-known venue where coordination among multiple robots teaming in a soccer game can be evaluated. ITAM's Eagle Knights [10] have been participating since 2003 in different soccer leagues. While there have been significant improvement in the performance of RoboCup teams over the years, several aspects of the competition were defined to simplify multi-robot coordination tasks. One clear example is the Small-Size League (SSL) having global aerial cameras simplifying visual processing with control centralized by an individual computer sending commands to all robots on the field. Additionally, the limited size of the soccer arena avoids many communication problems present in larger environments. The game involves two teams of five robots, up to 18cm in diameter each, playing on a 4m by 5.4m carpeted field, as shown in Figure 1.

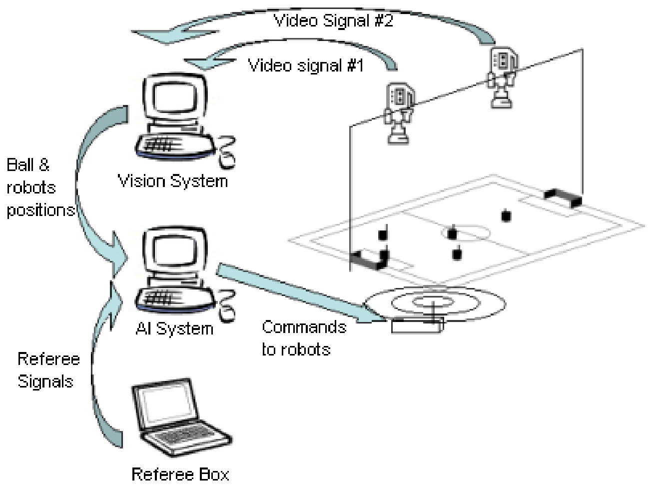

Fig. 1. ITAM's Eagle Knights RoboCup Small-Size league system architecture. A number of computers remotely control the state of the game. The Vision System receives images from the cameras mounted on top of the field and sends information about relevant objects to the AI System producing remote commands to the robots in the field. A Referee Box send game signals to both teams.

The system architecture consists of one or two remote computers sending action commands to the robots. Computers receive video signals from cameras mounted on top of the field and provide wireless signals to the five robots on the field. The main functional components of the small-size league system are shown in Figure 2: **Vision System**, **AI System**, **Referee Box**, and **Robots**.

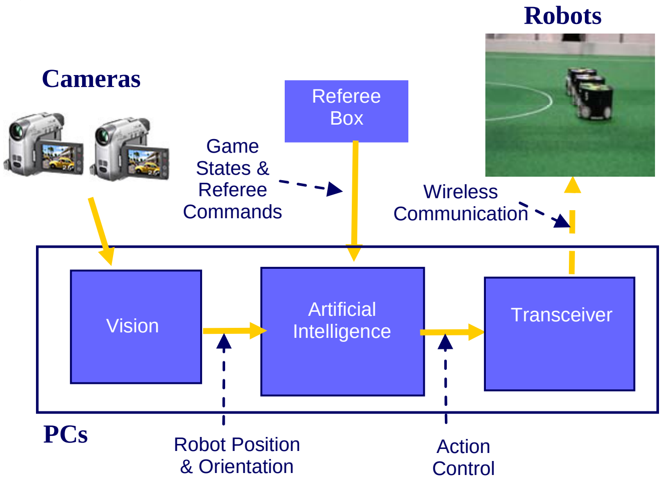

Fig. 2. ITAM's RoboCup Small-Size league block diagram. Visual input from cameras mounted on top of the field are processed by the Vision module to provide the AI module with robot positions and orientations. The AI module sends action command to the robots via a transceiver.

Vision System. The Vision System is the main source of input during a game. Its main task is to capture video in real time from the two cameras mounted on top of the field. The camera system needs to recognize the set of colors assigned to the objects of interest in the game, namely robots and ball, all in accordance with the SSL rules [11]. Once objects are recognized, the system identifies and computes the position of the ball together with position and orientation of the robots in the field. Robots of one team must have a blue colored 50mm in diameter circular patch on top while the other team must have a yellow colored patch. Additional patches are used to identify robots and compute their orientation. A particularly critical challenge in the Vision System is to adapt to different light conditions by performing dynamic color calibration. Positions and orientations of objects are transmitted to the AI System. The computation cycle is around 30 frames per second (More details can be found in [12].)

AI System. The AI or High Level Control System receives object positions and orientations, i.e. object localization, from the Vision System in order to produce robot action commands. These actions depend on strategic decisions made a priori depending on robot roles, e.g. goalkeeper, defense, and forward, and on the current state of the game, e.g. attacking or defending. Additional game state information comes from the Referee Box, e.g. regular play, free kick, etc. The AI System is composed of three main submodules: Behaviors, Collision Detection, and Motion Control as shown in Figure 3. Final robot action decisions are converted into commands that are transmitted to the robots via a wireless link through a transceiver. Transmission is asynchronous.

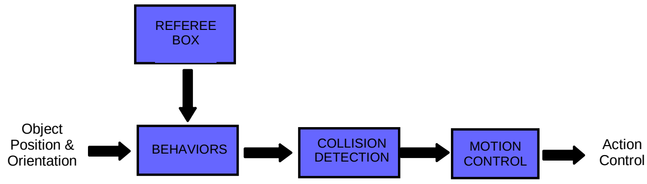

Fig. 3. AI System block diagram consisting of Behaviors, Collision Detection and Motion Control components. The Referee Box sends signals generated by a human referee during the game.

Figure 4 shows a sample set of behavior for a robot attacker described: Reach Ball, Circle Ball and Kick Ball. These behaviors are activated by external signals such as *ball_near* or *ball_far*.

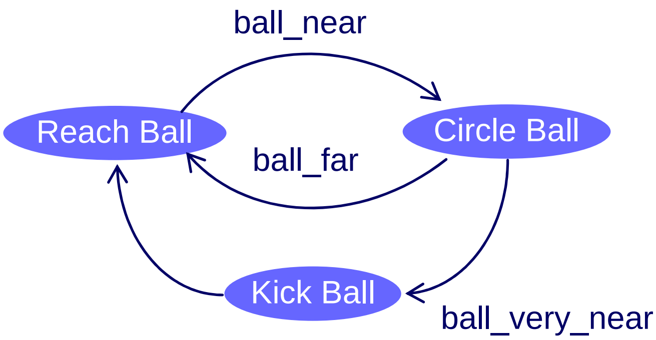

Fig. 4. Attacker behaviors described as a state machine. Three behaviors are defined: Reach Ball, Circle Ball and Kick Ball. These sample behaviors or states are activated from external signals such as *ball_near* or *ball_far*.

Referee Box. The Referee Box communicates additional decisions (penalties, goal scored, start of the game, etc.) generated by the human referee during a game. These decisions correspond to a set of predefined commands sent to the AI system via a serial link.

Robots. The Robots execute commands transmitted by the AI system through the transceiver to generate local robot actions, e.g. move, kick, and dribble. Robots in this league are mostly omni-directional having either three or four wheels with corresponding motors controlling movement. There is an additional motor controlling the dribbler that keeps the *golf* ball tight into the robot for a limited amount of time as specified by the rules. Additionally, the robot includes a solenoid controlled by capacitors to kick the ball. Local robot control is managed by a Texas Instruments TMS320LF2407A fixed-point single chip DSP (Digital Signal Processor) optimized for digital motor control. The DSP receives remote communication from the AI System via a Radiometrix RPC-914/869-64 local transceiver with radio frequency at either 914MHz or 869MHz with 64kbit/sec transmission rate similar to one attached to the PC. Teams alternate in radio frequency. Finally, rechargeable 9V/1600mA batteries are incorporated in the robot. The robot block diagram is shown in Figure 6. A picture of the Eagle Knights three wheeled robot used for this project is shown in Figure 7.

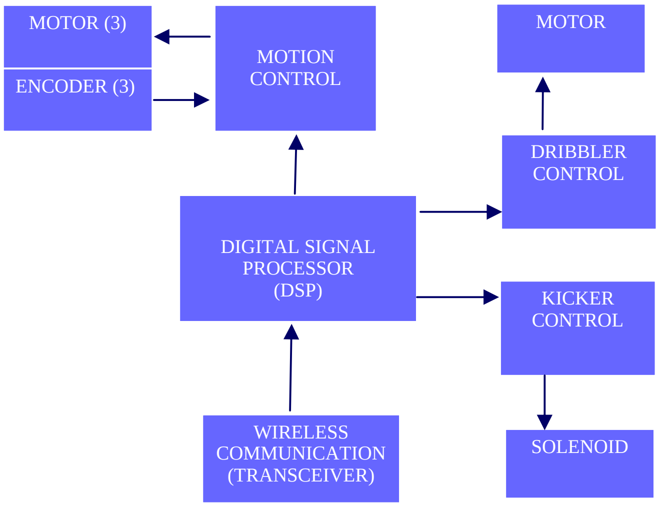

Fig. 6. Eagle Knights robot block diagram. A DSP receiving remote signals via a wireless transceiver control three (or four) motors for omni-directional movement. Additionally, the DSP control a dribbler and a kicker control mechanism.

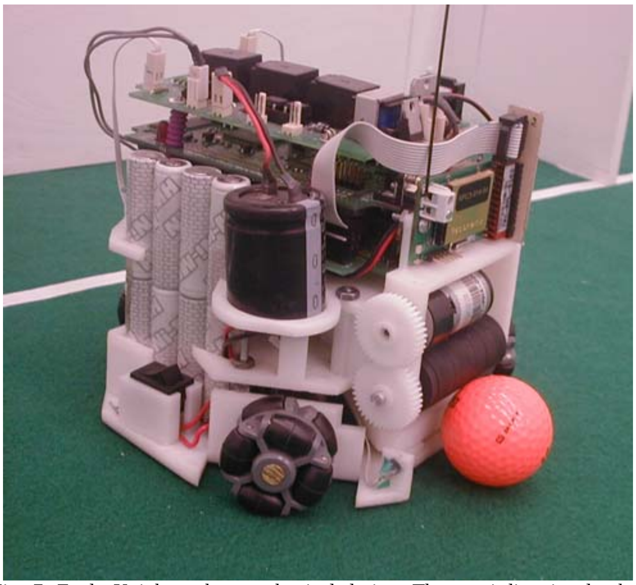

Fig. 7. Eagle Knights robot mechanical design. The omni-directional robot includes a kicker, dribbler and motion control all processed by a local DSP receiving signals from the remote AI System computer via a transceiver.

B. *Mobile Robot Architecture*

A major constraint in the small-size league architecture is the global vision system limiting mobility of the robots to the soccer field while keeping them under full camera view. By providing a local vision system as in the case of the Mid-Size and Four-Legged RoboCup leagues it becomes possible to avoid this restriction. For this purpose we have extended our robot design to include a local camera located where the dribbler and kicker used to be while adding a Crossbow Stargate [13] as shown in Figure 8. The Stargate itself is outfitted with a webcam and an 802.11 wireless card. It is a relatively powerful, small form factor single-board computer that has found applications in ubiquitous computing and wireless sensor networking. It is based on Intel's 400MHz X-Scale processor and has 32MB flash memory and 64MB SDRAM and provides PCMCIA and Compact Flash connectors on the main board. It also has a daughter board with Ethernet, USB and serial connectors. A Logitech QuickCam Pro 400 webcam is connected through the USB port, and communication carried out by an Ambicom Wave2Net IEEE 802.11b compact flash wireless card. The operating system is Stargate's version 7.2, an embedded Linux system (kernel version 2.4.19).

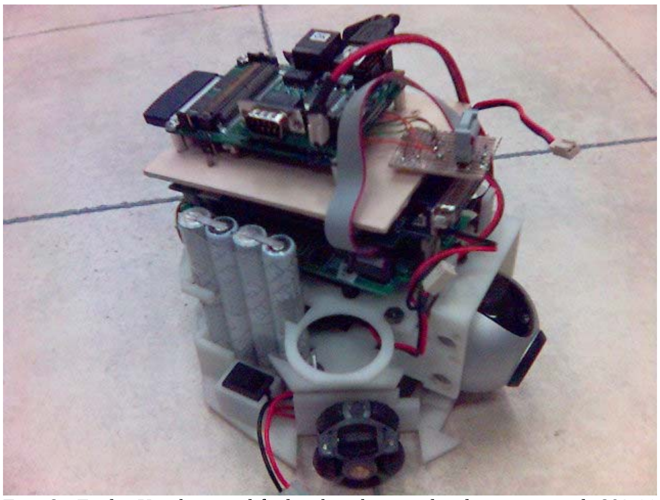

Fig. 8. Eagle Knights modified robot having local camera and 802.11 communication capabilities. The original robot architecture is maintained although replacing the transceiver with a direct linkage to the Crossbow Stargate (on top) managing wireless communication and local vision. Note how we replaced the kicker and dribbler with the camera due to camera.

The original communication transceiver was replaced by a direct wire connecting the main robot board with the Stargate while moving the Vision System and AI System computations to the local Stargate for processing as shown in Figure 9. Since the Stargate contains a Linux operating system, porting previous robot code written in C did not become a major issue although not all functionality was required. From Figure 4 programmed the **Reach_Ball** behavior.

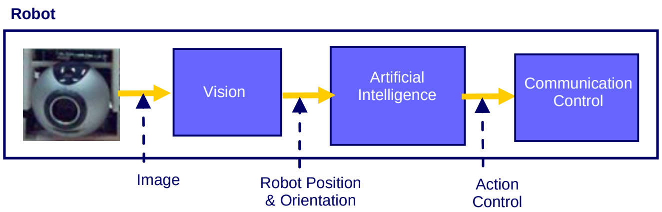

Fig. 9. Extended Small-Size robot architecture. Visual input from a camera mounted on the robot itself is processed by the Vision module to provide the AI module with robot positions and orientations. The AI module sends action command to the robot locally. Communication control is available for networking with other robots or a remote computer.

The block diagram for the robot design is shown in Figure 10. Due to size constraints we took out the kicker and dribbler to make space for the local camera. The Stargate was put on top of the robot as previously shown.

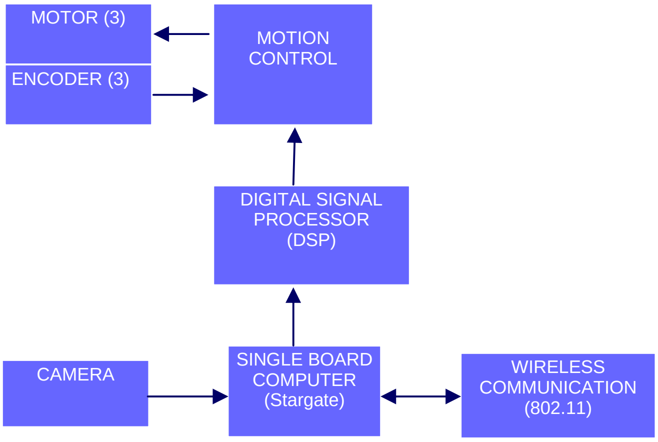

Fig. 10. Extended Small-Size robot block diagram. A DSP receiving remote signals via a wireless transceiver control three (or four) motors for omni-directional movement. Additionally, the DSP control a dribbler and a kicker control mechanism.

III. Wireless Ad-Hoc Networking

In the RoboCup Small-Size soccer league, robots are very close to each other on the field. This means that all robots are within transmission range of one another which makes routing of messages between computer and robot, or between robots, trivial; any robot can send a message to any other robot in a single transmission. For other applications, however, as the range of robot mobility is extended, nodes may be too far apart to directly communicate, requiring messages to be routed through intermediate robots to reach their destination. In such situations, known as multi-hop ad hoc networks, nodes must cooperatively establish routes and forward messages in order to maintain communication.

In terms of ad-hoc networking protocols, the Stargate used in our system architecture is shipped with AODV [14], the Ad hoc On-demand Distance Vector routing protocol. AODV has been designed under the assumption that end-to-end paths are available at least most of the time. In other words, it is assumed that the network is connected most of the time and that disconnections, when they happen, are short lived. However, in some situations such as disaster recovery or emergency response scenarios, end-to-end connectivity cannot be guaranteed; in fact, it may turn out that the network is disconnected for most of its operational lifetime. For this reason, we have developed StAR (Steward Assisted Routing), a routing protocol for networks in which links are often unavailable due to mobility or other interference. Figure 11 shows a sample network where typical ad-hoc protocols such as AODV will fail, highlighting the need for protocols that are robust to long-lived and/or frequent network disconnections such as StAR. Below, we describe both AODV and StAR.

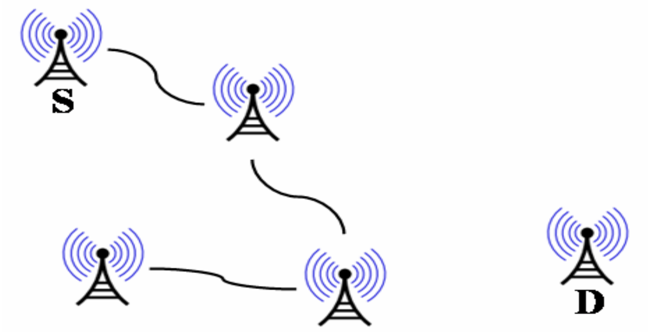

Fig. 11. An example network in which there is no route from S to D. Existing on-demand routing protocols fail to deliver messages when a route cannot be established. StAR will buffer data at the node nearest to the destination until a route is available.

A. AODV

Unlike traditional wired networks, multi-hop ad hoc networks (MANETs) require a routing protocol that can respond quickly to node failures and topology changes. AODV is an example of an on-demand routing protocol. It establishes a route between a source-destination pair only when the source node has data to send to the destination. This notion is in contrast to proactive routing protocols commonly used in the Internet, which can afford the luxury of maintaining all necessary routes since they rarely change. Because routes can change very quickly in a MANET, the signaling overhead required to maintain all routes at all times can be prohibitively high.

AODV's route establishment phase consists of two main control messages, the RREQ (route request) and RREP (route reply). A robot, when desiring to send a message to another robot, must send a route request for the destination. This request is broadcast to all neighbors and relayed by intermediate nodes until it reaches the destination, or a robot with a route to the destination, at which time a route reply message is unicast back to the source robot. This message sequence establishes the (temporary) route so that packets may be forwarded from source to destination. For a much more detailed description of AODV, the reader is referred to the AODV RFC [14].

The major failing point of AODV, and other on-demand routing protocols such as DSR [15], occurs when there is no existing end-to-end path from source to destination, and the route discovery phase fails. In this case, data packets are dropped, and the destination does not receive the intended messages.

B. StAR

The objective of StAR is to nominate, for each connected partition in the network, a *steward* for each destination. These stewards are the robots that are next expected to have communication with the destination. For example, if there is a single moving robot who communicates with all other stationary nodes, this robot is likely to be nominated as the steward for all destinations. Messages are sent to the associated steward, who will store them until a route to the destination (or a better steward) is available.

StAR routes messages using a combination of global (network-wide) contact information and local (intra-partition) route maintenance. The topological location of active destinations in the network is propagated through periodic broadcasts, or contact exchanges, between neighbors. These broadcasts occur at a fixed interval if there are nearby nodes, and contain only those entries in the routing table that may have changed since the last broadcast to the same set of neighbors. The message includes a unique sequence number indicating the broadcast from which the information came.

Initially, each node nominates itself as the local steward for each destination, and therefore does not route messages to any neighbor. As updates are received from neighbors that advertise better local stewards, routes are formed. The ranking of stewards is based on the most recently heard sequence number for a destination, or route length if two nodes have the same destination sequence number. In a connected network (i.e, a network in which there are connected routes between all robots), each tree will be rooted at the destination itself, and messages routed directly to the destination.

Thus, route maintenance results in one tree per destination of interest in each partition, where each tree is rooted at the locally nominated steward for that destination. Note that it is possible (and quite likely) that a node can be the steward for more than one destination at any given time, and the tree for each destination will contain precisely the same nodes and links.

IV. Experiments and Results

In addition to outfitting each robot with a local camera and ad hoc networking capabilities, we have loaded them with a simplified surveillance application. Each robot is defined as either a source (sensor) node or a destination (sink) node. It is the responsibility of source nodes to acquire images of their surroundings through the webcam at 5-second intervals, and transmit them to a designated sink. Because there may be no direct route to the sink at the time the image is taken, StAR ensures that the image is buffered at some intermediate node until a route toward the destination exists. We are currently experimenting with a wide range of network topologies using StAR on the extended Eagle Knight robot architecture for comparison with standard on-demand routing protocols.

In what follows, we define three experiments using four fully autonomous small-size robots in order to examine protocol performance under varied scenarios. In each experiment described below, we modify the mobility of the sensor and sink nodes to provide more or less connectivity in the network. All experiments last five minutes, during which time each sensor node captures a 230KB image every five seconds, resulting in a total of 30 images per sensor. We measure the number of images that are successfully sent to the sink to determine the effectiveness of the routing protocol.

A. *Experiment 1: Static Network*

We first examine the behavior in a network with four static nodes, two of which are sensors. The distance and obstacles between each node are different, as shown in Figure 12, which leads to intermittent connectivity between some node pairs. Most notably, the connectivity between the sink (node 7), and one of the sensors (node 3) is often unavailable due to the many walls between them, which requires images to be routed through node 1 at some points.

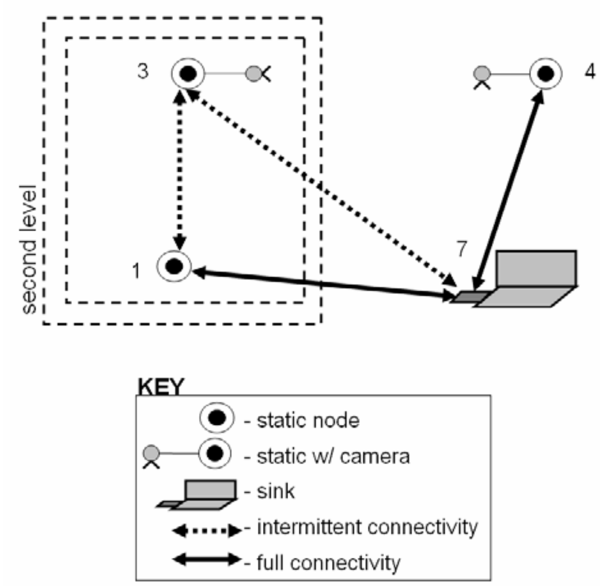

Fig. 12. Topology for Experiment 1: Static network. Sensor node 3 sends images to sink node 7 through intermediate node 1 when direct communication to the sink is unavailable.

Table I compares the delivery rates of AODV and StAR. Both protocols deliver more than 75% of captured images, however, StAR is able to deliver all 60 images, since it handles the intermittent connectivity between nodes 3 and 7 either by buffering the images at the source until a route can be established, either directly or through intermediate node 1.

TABLE I
Performance of AODV and StAR in Topology 1

	Image Deliveries	**Ratio Delivered**
AODV	46	76.67%
StAR	60	100.00%

B. *Experiment 2: Static Sensors with Mobile Intermediate Node*

In this experiment, all sensor nodes remain static, while an intermediate relay node moves to enable network connectivity. As shown in Figure 13, two of the sensor nodes 1 and 3 sometimes have connectivity with the sink, while the third sensor node 4, never has direct connectivity. Mobile node 2 enables connectivity between sensor node 4 and the sink, allowing images to be transmitted over a three-hop route (4 – 2 – 1 – 7).

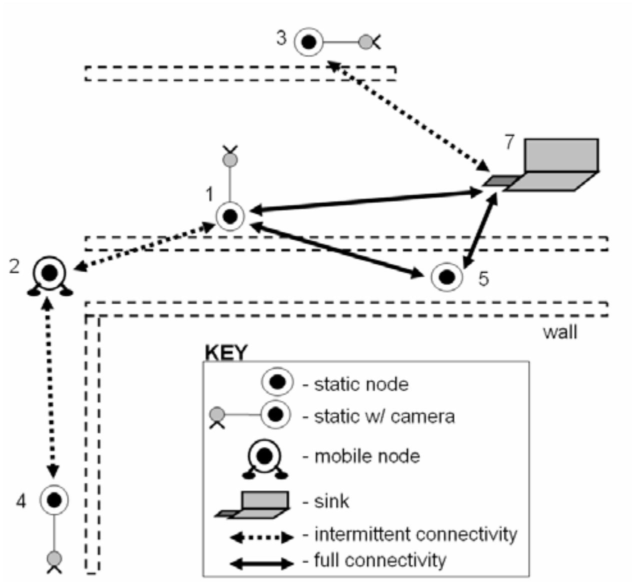

Fig 13. Topology for Experiment 2: Static sensors with mobile intermediate node. Static sensor node 4 sends images to sink node 7 through intermediate mobile node 2 and static node 1.

Table II shows the performance of the two routing protocols in experiment 2. AODV does not take advantage of the added connectivity provided by mobile node 2, and therefore fails to deliver any images from sensor node 4. Using StAR, however, the mobile node carries the images until a route can be established through node 1 to the sink. StAR is therefore able to successfully deliver all 90 images. Like the previous experiment, the poor connectivity between the sink and sensor node 3 makes it difficult for AODV to deliver images because of its inability to buffer the images until a route can be established.

TABLE II
PERFORMANCE OF AODV AND STAR IN TOPOLOGY 2

	Image Deliveries	**Ratio Delivered**
AODV	48	51.11%
StAR	90	100.00%

C. Experiment 3: Mobile Sensors with Static Intermediate Node

This experiment is representative of a situation where mobile sensor nodes are deployed to gather information before relaying it to static sink nodes. In this topology, shown in Figure 14, two mobile nodes with attached cameras had limited connectivity to static relay nodes. The static nodes all had intermittent connectivity due to obstacles and distance. The mobile nodes ranged at a distance from the sink, never coming into direct contact.

Again, as shown in Table III, StAR shows a large improvement over the standard AODV routing protocol. Because the source sensor nodes are able to buffer images until a relay node is available, and that relay node can in turn buffer the images until a direct path to the destination is available, the protocol delivers nearly every captured image.

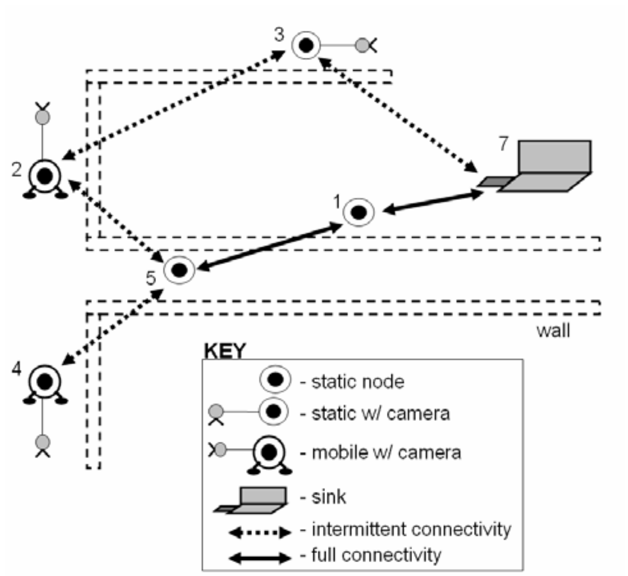

Fig 14. Topology for Experiment 3: Mobile sensors with static intermediate nodes. Mobile sensor nodes 2 and 4 send images to sink node 7 through intermediate static nodes 5 and 1.

Another discovery worth mentioning is that when we performed this type of experiment, the transmission of the images, although complete in terms of the number of images received, in some cases did not get the entire image across. Most probably this is due to the fact that if the mobile sensor node is in the middle of a transmission when it goes out of range, only part of the picture arrives, making it impossible to view it at the sink. This problem would likely be dealt with at the application layer.

TABLE III
PERFORMANCE OF AODV AND STAR IN TOPOLOGY 3

	Image Deliveries	**Ratio Delivered**
AODV	41	45.56%
StAR	89	98.89%

V. CONCLUSION

We presented preliminary results from collaborative research work between the robotics laboratory at ITAM and the internetworking research group at UCSC in incorporating vision-based sensing and ad-hoc networking capabilities in small autonomous mobile robots. The robots used were developed at ITAM in the context by the Eagle Knights RoboCup Small-Size league competitions. These robots are currently started to be used in search and rescue related applications where extensions to their architecture is necessary in order to have them execute outside the limited soccer field. The main hardware modifications have involved the inclusion of a Crossbow Stargate single-board computer connected to a local web camera and 802.11 communications device. In terms of software, algorithms previously designed for remote execution have been ported to the Stargate for local processing. Additionally, we have ported ad-hoc communication protocols developed by the networking group at UCSC to operate on the Stargates.

As proof of concept, we carried out a number of experiments to showcase and evaluate the communication capabilities of the resulting robotic system. We have experimented with various static and mobile multi-node configurations to test how effectively sensor nodes can deliver images to a sink. We show that the proposed routing protocol was quite efficient handling disruptions due to both node mobility and poor link quality.

Our long-term goal in this collaborative effort is to be able to deploy multiple robots in real world applications such as search and rescue where advanced communication capabilities are required. Our current work in this direction is to extend the capabilities of both the robots and networking in adding more autonomous networking related control in the robots to enable them to take communication-related decisions during network failures, for example, by searching for locations where communication can be reestablished.

It should be noted that we have chosen to extend the Small-size league architecture since the robots were built by our group and can easily be modified and extended with other devices if so desired, such as having two cameras, etc. Other robotic platforms were considered as well such as the already discontinued Sony AIBO incorporating ad-hoc communication services. From evaluations previously done at our robotics lab, the Small-size robot used in this project has at least twice the speed of the Sony AIBO, while our latest Small-size generation has more than four times the AIBO speed. Current plans involve using our latest small-size robot models. Finally, this project does not limit itself to ground robots but also to unmanned aerial vehicles (UAVs) in developing hybrid ad-hoc networks.

ACKNOWLEDGMENT

This work has been supported in part by UC-MEXUS CONACYT, CONACYT grant #42440, LAFMI, and "Asociación Mexicana de Cultura" in Mexico.

REFERENCES

[1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In Proceedings of the IJCAI-95 Workshop on Entertainment and AI/ALife, 1995. http://www.robocup.org.

[2] Barrera, A., and Weitzenfeld A., 2006, Bio-inspired Model of Robot Adaptive Learning and Mapping, IROS – International Robots and Systems Conference, Beijing, China, Oct 9-13 (accepted for publication).

[3] RoboCup Rescue, Urban Search and Rescue Robot Competitions, 2004 (http://www.isd.mel.nist.gov/projects/USAR/competitions.htm).

[4] Rescue Robotics, IEEE Robotics & Automation Magazine, 9 (3), September 2002.

[5] Orfinger, B., Robot Responders at WTC Site Fit Into Tight Spaces, Disaster Relief, Oct 2001, (http://www.disasterrelief.org/Disasters/011015robots/).

[6] Osuka, K., Murphy, R., Schultz, USAR Competitions for Physically Situated Robots, IEEE Robotics & Automation Magazine, 9 (3): 26 - 33, September 2002.

[7] Petkov, V., Rutishauser, M., Boice, J., Obraczka, K., Williams, T., Costa, D., More wily than a coyote: Monitoring wildlife in real-time using a multi-tier, disruption-tolerant network, under submission.

[8] Margi, C., Lu, X., Zhang, G., Manduchi, R., and Obraczka, K., Meerkats: A Power-Aware, Self-Managing Wireless Camera Network for Wide Area Monitoring, under submission.

[9] Bromage, M., Obraczka, K., Potts, D., SEA-LABS: A Wireless Sensor Network for Sustained Real-Time Monitoring of Coral Reefs, under submission.

[10] Martínez-Gómez, L.A, Torres, M., Soto, M., Weitzenfeld, A., 2006, Eagle Knights 2006 - Small Size RoboCup Soccer Team, RoboCup 2006, Bremen, Germany, June 14-18.

[11] Verret, Ball, Kiat Ng. Laws of the F180 League - Release 3.00a. http://www.itee.uq.edu.au/~wyeth/F180%20Rules/index.htm.

[12] Martínez-Gómez, L.A., and Weitzenfeld, A., 2004, Real Time Vision System for a Small Size League Team, Proc. 1st IEEE-RAS Latin American Robotics Symposium, ITAM, Mexico City, October 28-29.

[13] Stargate Platform, http://platformx.sourceforge.net/

[14] C. Perkins and E. Belding-Royer, 2003, RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing.

[15] David B Johnson and David A Maltz, 1996, Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing Volume 353.