

Autonomous Communication Relays for Tactical Robots

H. G. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta

J. M. Spector

Space and Naval Warfare Systems Center
San Diego, CA 92152-7383, USA
Email: nguyenh@spawar.navy.mil

Department of Electrical Engineering
University of Florida
Gainesville, FL 32611-6200, USA

Abstract

The high-bandwidth digital radio link between a mobile robot and its remote control station degrades quickly as the robot penetrates the interior of a building or becomes shielded by intervening terrain. This paper describes a current project that uses mobile autonomous communication relay nodes to overcome this problem. Each node is a small slave robot equipped with sonar, ladar, and 802.11b-based ad hoc networking radio. The relay robots follow the lead robot and automatically stop where needed to maintain a solid communication network between the lead robot and the remote operator. With their on-board external sensors, they also act as rearguards to secure areas already explored by the lead robot. As the lead robot advances and RF shortcuts are detected, relay nodes that become unnecessary will catch up to the lead robot and be reused, using maps generated by the lead robot. All relay deployment and redeployment functions occur without the operator's awareness.

1. Objectives

One of the weaknesses of current mobile robots operating in real-world scenarios is the communication link to the operator's console. Hard cables reduce mobility and often become entangled and broken, rendering the robot inoperable. User surveys have identified radio-frequency (RF) communications as more desirable [1]. However, most RF communication systems currently employed on teleoperated robots in the field are analog, which often experience signal interference, multipath, and attenuation. Spread spectrum digital systems are more immune to these problems and provide a level of transmission security, but operate at shorter ranges and mostly on line-of-sight (LOS).

To extend the range of these digital radios and provide non-line-of-sight service, we are exploring the use of relay nodes. These nodes could be dropped by the mobile robots where required. However, in tactical and reconnaissance missions, the robot's convoluted path may often lead to situations where intermediate relay nodes are no longer needed (e.g., RF shortcuts are encountered).

To maximize resources and allow for extended explorations, unneeded relay nodes should be reclaimed and reused. We accomplish this function through the use of mobile relay robots that follow the lead robot in convoy fashion (Figure 1), stop and act as relay nodes where needed, and catch up to the lead robot to be redeployed when no longer needed. These activities are all performed without the operator's involvement. With minimal additional sensory hardware, these relay nodes will also act as rearguards, preventing areas previously tagged as clear of hostile elements by the lead robot from being re-occupied without detection.



Figure 1. Four relay robots convoying behind the lead robot (laboratory demonstration scenario).

2. Approach

This project is being conducted in two phases. Phase 1 addresses the deployment of relay units and establishing a relaying network. The specific steps to be accomplished in phase 1 include:

1. Developing a convoying strategy to allow four mobile relay robots to follow a teleoperated lead robot into a building.
2. Developing a strategy for deploying the relay nodes at appropriate locations. Since there can be many RF nulls (locations where the RF signal

strength is locally low), this most likely involves a relay robot stopping only when the signal strength to the closest node to the rear has decreased beyond a set point and further forward movement fails to improve it.

In phase 2, the re-deployment and rearguard functions are addressed. The ability of the relay robots to find and catch up to the lead robot means that a map is needed. (Two robots can be in RF range of each other, but far enough to be outside visual range. Navigation by RF direction is also very difficult in a complex environment.) Thus the specific steps of phase 2 are:

1. Acquiring a real-time mapping ability for the lead robot. The lead robot will map the environment as it passes through it.
2. Adding the ability for the lead robot to pass the map back to a relay node that needs it.
3. Developing the navigational skill to allow a relay robot to catch up to the lead robot to be reused.
4. Adding rearguard functions (detection of intruders) to deployed relay nodes.

We are leveraging our existing pool of laboratory robots for this demonstration, using *ROBART III* as the lead robot and four ActivMedia *Pioneer 2-DX*'s as the relay robots (Figure 1). Details on these robots and their configurations were given in our previous paper [2]. A transition to the real world will probably require more rugged tracked robots that can handle unpredictable terrain.

3. System Development

ROBART III, developed in-house, is an advanced technology-base development and demonstration platform for physical security and non-lethal tactical response. To prepare *ROBART III* for use in this project, a SICK *LMS200 2-D* laser radar (ladar) has been added to support the real-time mapping function. A KVH fiber-optic gyro has also been incorporated for improved dead-reckoning accuracy. The master onboard processor has been upgraded from a 68HC11-based microcontroller to a more powerful Bright Star Engineering (BSE) *ipEngine*. The credit-card-sized *ipEngine* hosts a 66 MIPS PowerPC CPU, 4 MB of flash memory, 16 MB of RAM, 16,000-gate FPGA, and dual RS-232 and 10Base-T Ethernet ports. The FPGA can be configured to provide additional input/output ports for various sensors.

For use as the relay/rearguard nodes, we equipped four *Pioneer 2-DX* robots with a suite of navigation and security sensors, processors, and RF modems. A SICK *LMS200* ladar was added to allow the robots to

perform the conveying function. A magnetic compass facilitates orientation for navigation using global maps. To provide rearguard functions, we installed on each *Pioneer* a Sony pan-tilt-zoom camera and a microphone (Figure 2). These will be used in conjunction with the onboard sonars and ladar for intrusion detection in areas that the lead robot has passed through and are supposed to be clear of hostile elements.



Figure 2. A Pioneer robot configured for communications relay and rearguard functions.

To provide more processing power, we replaced the two BSE *ipEngine* boards mentioned in our previous paper [2] with a BSE *nanoEngine*, an even smaller processor card with a 200 MHz StrongARM SA1110 microprocessor. We built a *nanoEngine* daughterboard (Figure 3) to provide power regulation and various I/O interfaces to the *nanoEngine*, including 10Base-T Ethernet, USB, general-purpose IO, LCD, and five RS-232 ports (two of which are provided by a DUART on the daughterboard). Devices that will make use of these serial ports include: the Pioneer microcontroller, the SICK ladar, the Sony camera control lines, the electronic compass, and the voice synthesizer.

Video and audio processing is handled by the Indigo Vision *VP604* board. The *VP604* resulted from a contract from SSC San Diego's Man-Portable Robotic System project [3,4] to Indigo Vision, to combine the functions of their existing *VP500* video digitizer/encoder and *VP400* decoder into a miniature

hardware codec (4.14" x 2.14" in size). The *VP604* supports four input analog video channels and one output video channel in S-video, RGB, or composite (CVBS) format, plus analog audio input/output, and communicates over a standard 10/100 Base Ethernet network. A daughterboard for the *VP604* was similarly developed that provides regulated power, access to all ports, a microphone differential preamplifier with Automatic Gain Control (AGC), and a 675mW audio amplifier with manual volume control for the output audio channel. Indigo Vision is working on advanced motion detection functions for the *VP604* under a separate contract from SSC San Diego.



Figure 3. A *nanoEngine* and its daughterboard.

4. Compact Ad Hoc Networking Radios

There are several problems with currently available IEEE 802.11-type wireless modems that make them difficult to use in a mobile-robot-based network. Most are either rather large or require two units (access point and bridge) to operate in relay mode. They are also inefficient in network reorganization in the presence of node mobility. To solve these problems, we have worked with BBN Technologies to implement a new ad hoc networking solution developed by BBN under DARPA/IPTO's Software for Distributed Robotics (SDR) program.

BBN's ad hoc networking software uses a proactive link-state protocol. Each node in the network has complete information about the characteristics of all links. It can execute a routing algorithm of its choice and determine the paths most suitable for the chosen criteria. Each node uses broadcast messages (sent at intervals determined by the network criteria and the environment) to determine the characteristics of the links and set up the routing table. The routing table is recomputed whenever certain network events occur, such as when the link quality between two

nodes has dropped below a preset level appropriate for a desired scenario. Thus the routing table can be updated before a link is broken, and the network is automatically maintained in a proactive fashion, for optimal information transmission and minimal lag. There is no delay incurred for route re-selection due to broken links.

This software has been incorporated into a set of compact ad hoc networking wireless modems, each the size of a pack of playing cards (Figures 4 and 5). Each stand-alone wireless modem contains an 802.11b wireless LAN card (the ORiNOCO *WaveLAN* PC Card Gold), a *nanoEngine*, and a Radio Interconnect Board (RIB) developed by us. It also has connectors for external power and antenna, as well as Ethernet and serial communication ports.



Figure 4. Top side of the Compact Ad Hoc Networking Radio, showing the *Wavelan* PC Card.

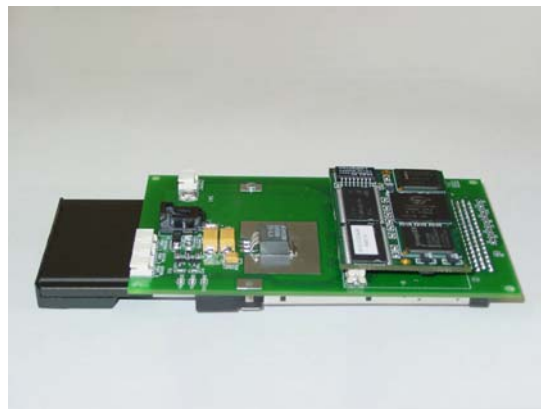


Figure 5. Bottom side of the Compact Ad Hoc Networking Radio, showing the *nanoEngine* processor card.

5. Relay Deployment Strategies

There are several strategies for deploying mobile relay nodes for LOS connectivity [5]. Since our communications links are mostly LOS, we considered similar strategies, but using link quality to determine connectivity. These strategies include:

(1) The lead robot proceeds alone, with all relay nodes remaining at the base station. It then stops and calls for a relay node when its link to the base station is about to break. A relay node will move to the location of the lead robot, freeing it to advance further. When the lead robot finds its link to the relay node about to break, it again pauses and calls for a second relay node to take the place of the first relay node, allowing it to proceed towards the lead robot and repeat the process.

(2) All relay nodes follow the lead robot in conveying fashion. When the link between the base station and the last relay node in the convoy is about to break, that node stops while the rest of the convoy continues on. Then when the link between the now stationary relay node and the last node in the remaining convoy is about to break, that node also stops and becomes a stationary relay node. The process continues until all nodes have been deployed.

(3) Hybrid strategies that fall in between the above two extremes, for example, allowing the lead robot to move while a relay node is en route, or pulling a small number of relay nodes using predictive heuristics.

Of these choices, strategy (1) results in the least energy expenditure by the entire system. Each relay node only has to move the minimum distance required. It also results in the most delays for the lead robot. The system uses more energy in strategy

(2) since the relay robots have to follow the lead robot on its meandering exploratory paths. However, the lead robot does not suffer any delay due to relay deployment. The hybrid strategies in (3) can be developed as compromises, reducing the delays at the cost of a little more energy consumption. However, since our objective is complete transparency of the relaying function, any delay imposed on the lead robot is unacceptable. Thus we chose to implement strategy (2), as illustrated in Figure 6. As the relay nodes are dropped off from the end of the convoy, the lead robot's advance is not impacted, and the operator can be assured of a reliable communications link without any additional distraction or burden.

6. Algorithm Development

We are currently writing software for the relay robots using a set of tools developed by the Robotics Laboratory at the University of Southern California, namely the *Stage* simulator and the *Player* robot device server. *Player* comprises a set of socket-based device drivers that provide simple Unix-file-like read/write access to individual devices on the robots [6,7]. Most devices associated with the *Pioneer 2-DX* have been modeled. *Stage* (currently maintained by HRL Laboratories) is a graphical user interface and simulator for the robot devices and environment [7]. It loads a binary image file for use as a map of the environment, spawns simulated *Player* devices as specified in a configuration file, and runs external high-level programs that control the robots' behaviors. Figure 7 is a partial screen shot of *Stage* running the conveying program using retroreflective beacons (discussed below). High-level software developed on *Stage* is transferable with few modifications to the robots, where real *Player* devices replace the simulated instances.

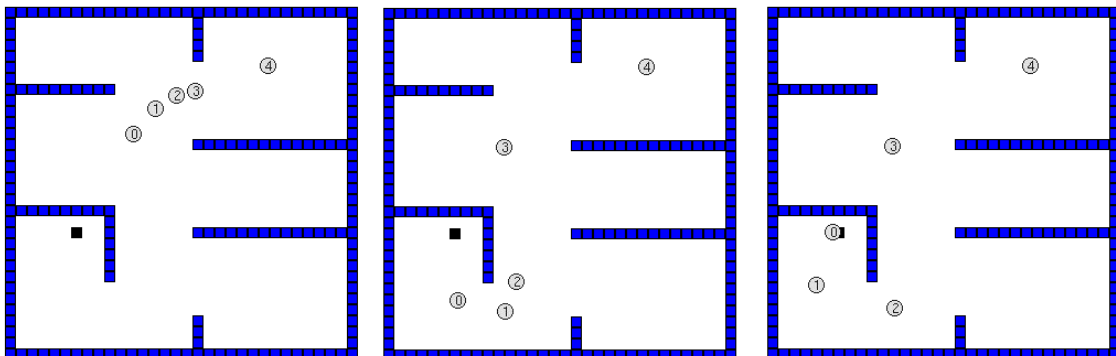


Figure 6. A demonstration of the conveying strategy (from a simulation developed by the University of Massachusetts, Amherst, that establishes LOS links between mobile robots [5]). Robot 0 is the leader, who is trying to reach the goal (black square). Robot 4 is the base station. The others act as relay nodes.

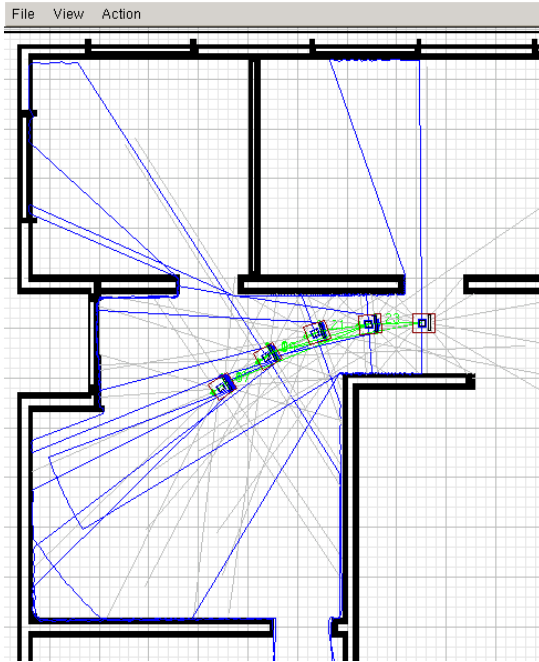


Figure 7. The conveying algorithm running on Stage. The lead robot is moving randomly. The dark lines from the robots outline areas visible to each ladar. Lighter lines represent sonar pings.

7. Laser Retroreflective Beacons

Our conveying algorithm (a copy of which runs on each relay robot) uses the Pioneer's SICK ladar for identifying and following the robot ahead, and both ladar and sonars for obstacle avoidance. The sonars and ladar complement each other and provide robust obstacle detection. For beacon identification and following, we use the *Player* retroreflective laser barcode device, normally used for locating fixed retroreflective tags mounted on walls. We configured this device for smaller tags that can be mounted on the back of each robot (see Figure 8). The tags were constructed using 1" strips of 3M *Scotchlite* retroreflective tape, arranged in 5-bit binary patterns. The *Player* laser barcode device requires the first and last bits to be 1's (reflective), resulting in eight possible IDs (17 through 31, in increments of 2). Under this configuration, the beacons are detectable to 4 m and identifiable to 2 m.

We first developed and tested the conveying algorithm on the Stage simulator (Figure 7), then transferred it to the *nanoEngines* running embedded Linux on the Pioneer robots, where it was fine-tuned and demonstrated in the real world.

8. Real-time Mapping

A capability for real-time mapping is required for the second phase of our project, to fulfill both the exploratory function of the tactical robot and to guide unneeded relay robots to rejoin the convoy. Odometry noise has always been a problem for real-time mapping of an unknown environment. It introduces uncertainty in the robot's position relative to its own map, especially when the robot has completed a circuit and come back to a previous point in the environment. This problem is known as simultaneous localization and mapping (SLAM). Historically, there have been several approaches developed to solve this problem, but none was completely successful. They required either unique environmental features that can be used by the robot for registration, or multiple passes through the data set, which is not a good fit for real-time systems. However, recently two similar techniques have emerged that have the potential to deliver robust SLAM in real-time [8,9]. We are considering using the CMU algorithm [8], distributed as part of the CARMEN open-source software package [10]. This algorithm combines an incremental maximum likelihood estimator with a posterior pose estimator to incorporate new ladar data into a map and to maintain consistency with older data, closing cycles in the map.



Figure 8. A relay robot with a retroreflective beacon on the back. This unit's beacon ID is 21 (10101₂).

9. Conclusions and Future Work

We are demonstrating a method for automatic maintenance of a high-bandwidth digital communications link between an advancing tactical robot and a base station. Our demonstration uses only indoor robots, although it may have just as easily used tracked robots in outdoor environments. In fact, SSC-San Diego's Man Portable Robotic System project, which has fielded ruggedized outdoor robots for use in military and rescue scenarios [3,4], has expressed interests in adapting our technology for use in the field.

The mobility of the relay nodes allows the network configuration to be more flexible. It can change in real time and adapt to the needs of the advancing tactical robot, moving relay nodes into new locations as needed. However, the luxury of having a group of cooperative robots may not be available to most users of tactical robots in the near future. Therefore we are exploring an adaptation of this project for use with just one robot.

In this scenario, a deployer module is designed to hold a number of relay "bricks" (each the size of our compact ad hoc networking radio, with additional batteries). The deployer unit is either attached to the tactical robot's chassis, or is a separate, towed module. Instead of using a distributed relay-deployment algorithm, control is centralized in the deployer, which automatically launches a relay node where needed to maintain the communications link.

The small size of these static relay nodes (dictated by the size of the deployer unit and the number of nodes it has to carry) limits the capacity of the energy source that can be packaged in each node. To maximize the operational time of the network, we are also looking at incorporating BBN Technologies' energy-conserving network protocols [11,12], developed under a contract from DARPA IPTO's SDR program. There are several tactics that can be employed to conserve energy in an ad hoc network, including: (1) appropriate use of sleep periods for each node, and (2) limiting transmit power to the minimum level necessary to permit communication between any two nodes. A system with these capabilities will significantly enhance the capabilities of existing tactical robots.

Acknowledgments

This project is funded by the Defense Advanced Research Projects Agency / Information Processing Technology Office (DARPA/IPTO) as part of the Mobile Autonomous Robot Software (MARS) program, under AO# M158.

References

1. H.G. Nguyen and J.P. Bott, "Robotics for Law Enforcement: Applications Beyond Explosive Ordnance Disposal," *SPIE Proc. 4232: Technologies for Law Enforcement*, Boston, MA, 5-8 November 2000.
2. H.G. Nguyen, H.R. Everett, N. Manouk, and A. Verma, "Autonomous Mobile Communication Relays," *SPIE Proc. 4715: Unmanned Ground Vehicle Technology IV*, Orlando, FL, 1-5 April 2002.
3. Man Portable Robotic System, Space and Naval Warfare Systems Center, San Diego, <http://www.spawar.navy.mil/robots/land/mprs/mprs.html>.
4. R.T. Laird, et al., "Issues in Vehicle Teleoperation for Tunnel and Sewer Reconnaissance," *IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, 28 April, 2000.
5. J. Sweeney, T. J. Brunette, Y. Yang, and R. Grupen, "Coordinated Teams of Reactive Mobile Platforms," *IEEE Int. Conf. On Robotics and Automation*, Washington, D.C., 11-15 May, 2002.
6. B. Gerkey, et al., "Most Valuable Player: A Robot Device Server for Distributed Control", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1226-1231, Wailea, HI, 29 Oct. - 3 Nov., 2001.
7. B. Gerkey, R.T. Vaughan, and A. Howard, "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems," *11th Int. Conf. on Advanced Robotics (ICAR 2003)*, Coimbra, Portugal, 30 June - 3 July, 2003.
8. S. Thrun, "A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots," *International Journal of Robotics Research*, **20**(5), pp. 335-363, 2001.
9. J.-S. Gutmann and K. Konolige, "Incremental Mapping of Large Cyclic Environments," *IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*, 2000.
10. Carnegie Mellon Robot Navigation Toolkit, <http://www-2.cs.cmu.edu/~carmen/>.
11. J. Redi, and B. Welsh, "Energy Conservation for Tactical Robot Networks", *Proc. IEEE Conf. on Military Communications (MILCOM) 1999*, pp. 1429-1433.
12. J. Redi, W. Watson, and V. Shurbanov, "Energy-Conserving Reception Protocols for Mobile Ad Hoc Networks," *Proc. IEEE Conf. on Military Communications (MILCOM) 2002*, pp. 850-855.