

# When Good Comms Go Bad: Communications Recovery For Multi-Robot Teams

Patrick Ulam, Ronald C. Arkin  
Mobile Robot Lab, College of Computing  
Georgia Institute of Technology  
Atlanta, USA  
{pulam, arkin}@cc.gatech.edu

**Abstract**—Ad-hoc networks among groups of autonomous mobile robots are becoming a common occurrence as teams of robots take on increasingly complicated missions over wider areas. Research has often focused on proactive means in which the individual robots of the team may prevent communication failures between nodes in this network. This is not always possible especially in unknown or hostile environments. This research addresses reactive aspects of communication recovery. How should the members of the team react in the event of unseen communication failures between some or all of the nodes in the network? We present a number of behaviors to be utilized in the event of communications failure as well as a behavioral sequencer to further enhance the effectiveness of these recovery behaviors. The performance of the communication recovery behaviors is analyzed in simulation and their application on hardware platforms is discussed.

**Keywords:** multi-robot systems; ad-hoc networks, communications recovery, behavioral sequencing

## I. INTRODUCTION AND MOTIVATION

Teams of mobile robots have become increasingly reliant upon communication networks for tasks ranging from inter-robot coordination [7,12,13] to distributed sensing [17]. The mobility inherent in the nodes of these networks presents a large number of challenges in maintaining communication connectivity that must be overcome in order for the team's goals to be accomplished successfully.

Due to the lack of hardwired network infrastructure in most mobile robot applications, a standard means of creating a network between groups of robots is the ad-hoc network. These highly dynamic networks are not without numerous problems, however, ranging from unpredictable signal attenuation as the robots move to accomplish their mission, uncertain message routing between nodes, to communications hardware failure [16]. Uncertain and adversarial environments such as those found on the battlefield, where enemies may actively try to subvert communications, further compound these unreliable network conditions.

Two approaches may be taken to cope with the problem of uncertain communication channels. The first is to proactively adjust the robot's behavior to try to avoid communication failure before it occurs [15]. The second involves a reactive strategy in which communication failures are dealt with as

they occur. While it would be ideal in most situations to avoid communications failures through proactive means, this option is not always possible, especially in scenarios that require the robots to operate in environment in which they have little or no *a priori* knowledge. Because failures in these environments are not predictable, reactive means of dealing with communications failures must be developed and utilized to provide for truly robust communication between team members. In order for such reactive measures to be effective, team members must be able to re-establish communications in a timely manner and reposition themselves so that after reforming the network the mission may continue successfully.

Communications recovery is an often overlooked aspect of mobile robotics. While some research has looked at means of communications recovery such as the methods used in maintaining line of sight networks [4,14,18] and many efforts have focused on proactive means of preventing communications failures [3,20], these may not be wholly suitable in highly dynamic and unknown environments.

This research presents a suite of reactive communications recovery behaviors that allow a robot to autonomously re-establish communications after a failure occurs. The performance of these behaviors is analyzed in the context of a surveillance mission in an urban environment. The motivating question of this research is: What should the members of a multi-robot team do in the event of a communications failure and how can the team re-establish network connectivity to all nodes quickly and then continue with their mission?

It is hypothesized that recovery from communications failures will become more effective as team size increases due to the increasing number of nodes available for connectivity. It is also predicted that the additional sophistication in communications recovery behavior selection will increase mission success rates and communication recovery speed.

## II. COMMUNICATION RECOVERY BEHAVIORS

To implement reactive communication recovery, a suite of four primitive communication recovery behaviors formulated as motor schemas [6] were integrated into the *MissionLab* multi-robot simulation environment [11]. The behaviors used were:

- *Probe*: A behavior that creates an attraction to nearby open space.
- *Retrotraverse*: A behavior that creates an attraction to waypoints stored at given distance intervals during mission execution.
- *Move-To-Higher-Ground*: A behavior that creates an attraction to nearby inclines [5].
- *Move-To-Nearest-Neighbor*: A behavior that creates an attraction to the last known position of the nearest teammate.

The behaviors were chosen to encompass two types of behaviors: context-specific recovery and general-purpose recovery. The context-specific behaviors, *Move-To-Higher-Ground* and *Probe*, are included to utilize implicit cues from the environment as to what could be causing the communications failure and provide a means by which to rectify the communication failures. The *Move-To-Higher-Ground* behavior, for instance, takes advantage of nearby inclines to provide a better vantage point for the robot's teammates, and allowing the robot to serve as a communications bridge of sorts for teammates on either side of the hill, or to raise the robot's elevation above what could be an attenuating obstacle. The general-purpose behaviors, *Retrotraverse* and *Move-To-Nearest-Neighbor*, do not require specific environmental features to be effective and as such they are expected to prove effective in a wider range of situations. All these behaviors were incorporated into motor schema assemblages that include an obstacle avoidance behavior and a low gain noise behavior.

Perceptual features were added to the *MissionLab* multirobot specification system to provide basic network connectivity information: *CommBroken* and *CommEstablished*. These perceptual features can trigger a transition in the robot's Finite State Automata (FSA) based on the robot's communication connectivity to its immediate neighbors in the network or to nodes that are several hops away. They also serve as a filter by monitoring the temporal length of the communications disturbance such that brief lapses in connectivity are not interpreted as total communication failures. Figure 1 depicts a typical FSA created in *MissionLab* that utilizes the communications-sensitive perceptual features with a single communications recovery behavior.

### III. EVALUATION METRICS

This research has been framed in the context of a team of robots conducting surveillance in an urban environment. Three primary metrics were selected for use in evaluation. *Mission completion rates* were measured as a percentage of the number of robots that successfully traversed the mission area to their goal position. The *area covered* in a surveillance mission is an important metric as it can be directly mapped to the amount of area surveyed for items of interest during the mission. Therefore, statistics are also collected concerning the

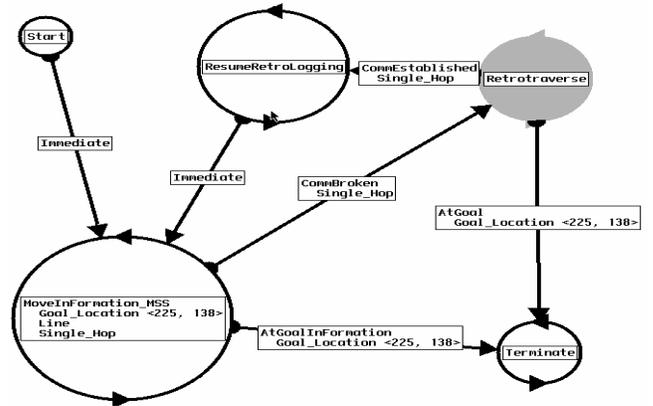


Figure 1. Example FSA for a coverage mission with a single recovery behavior

area covered by the robots during their traversal from their starting point to their goal. A very conservative measurement for coverage is used by assuming that the robot is only able to adequately view areas in a one-meter radius around the robot in sufficient resolution to be considered covered. The final metric calculated is *recovery time*. This metric serves as an indicator of the effectiveness of a communications recovery behavior in recovering from network failures. The recovery time is measured from the point a lesion (break in the network) is detected in the network until the time the network has been fully re-established and has stabilized.

### IV. EXPERIMENTAL SETUP

For the simulation experiments, the *MissionLab* multi-robot simulation environment was utilized. Experiments were conducted on two urban settings based on Military Operations on Urban Terrain (MOUT) sites at both Quantico and Ft. Benning (Figures 2, 3, 4, and 5). Topographical variation was added by hand to increase the diversity of the environments. Twenty missions were run in each environment. Each mission originated from a different point spaced evenly on the western side of the environment and had the team perform a traversal though the MOUT site to individual goals on the central eastern side of the site. The starting points and goals were held constant over all trials.

To provide network connectivity and signal attenuation data during the experiments, a series of network models developed by BBN was integrated into *MissionLab*. These include a line-of-sight model, a free-space attenuation model, and an attenuation model based upon MITRE's OPAR specification [8]. The free-space attenuation model was used for the communication-recovery experiments after it was determined that it most closely resembled the network characteristics found on-site at Ft. Benning during runs using physical robots.

Two series of experiments were run, one with a team consisting of two robots, the other with a team composed of four robots. For the teams composed of two robots, experiments were run using all possible enumerations of the

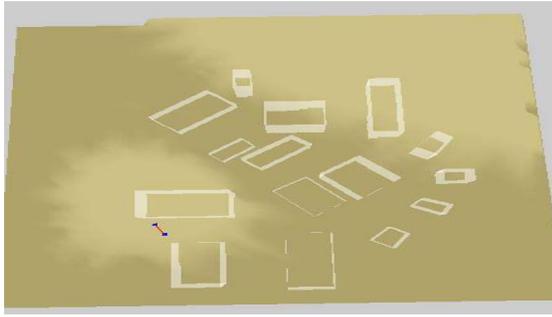


Figure 2. Quantico MOUT site overlay used in simulation runs. Light areas on the overlay indicate areas of significant elevation.



Figure 3. Photograph of Quantico MOUT Site

recovery behaviors such that each robot would only execute a single recovery behavior upon communications failure. Trials were also conducted in which, upon detecting a communications failure, that instead of executing a recovery behavior, the robot stops and waits for its teammate to recover communication with its own assigned behavior.

In the trials run with teams of four robots, we assigned combinations of recovery behaviors to the team in groups of two, such that two robots would execute their assigned recovery behavior and the other two team members would possibly execute another recovery behavior.

As the team size grows larger, questions concerning responsibility for communications recovery become increasingly important. For the experiments encompassing larger scale teams we investigated three different strategies for determining communication recovery responsibility:

- **Single-robot (Solo) Responsibility:** Only the robot who has lost communications contact with the rest of the network attempts to re-establish communications.
- **Team Responsibility:** All robots on the team perform recovery behaviors once any lesion in the in the network has been detected.
- **Nearest-Neighbor Responsibility:** The robot closest to the lesion attempts to recover communications in conjunction with the robot that has become disconnected with the rest of the network.

Each configuration of behaviors and communications recovery responsibility strategies was run twenty times over both environments. Control trials were also conducted for both the

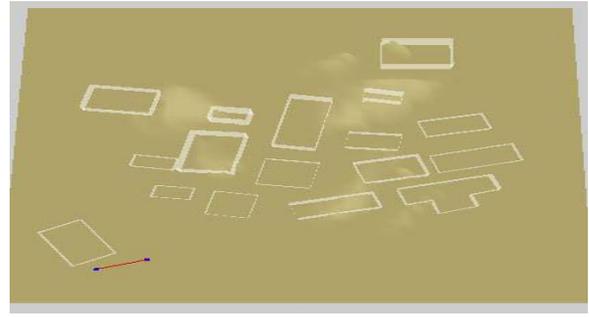


Figure 4. Fort Benning MOUT site overlay used in simulation runs. Light areas on the overlay indicate areas of significant elevation



Figure 5. Photograph of Benning MOUT site

two-robot and four-robot experiments where the robots did not attempt to recover communications upon failure.

## V. RESULTS

Statistics concerning mission success rates, communication recovery time, and mission area coverage for the experiments conducted with two robots are summarized in Table 1. Of the trials in which both robots executed the same recovery behavior, the *Move-To-Nearest-Neighbor* behavior proved most successful in terms of mission completion. The combinations of recovery behaviors among the team members that included the *Move-To-Nearest-Neighbor* behavior also proved quite successful in completing the missions. The success of this behavior lies largely in its ability to reposition the robots close to one another after a network failure. This serves the dual purpose of ensuring that no new obstacles attenuate the signal to the other robot as well as positioning both robots on the same side of the attenuating obstacle. By placing the robots close to one another on the same side of an obstacle, the robots can avoid oscillatory behavior caused by losing communication repeatedly by the same obstruction that originally caused the failure. Such oscillatory action was often seen in the trials that utilized *Retrotraverse* as one of the communications recovery behaviors.

Communication recovery times for the two robot teams show that the *Move-To-Nearest-Neighbor* behavior combinations provide the most rapid recovery of communication. *Retrotraverse* also provided recovery times significantly below that of the control. Not surprisingly, the contextual-recovery behaviors did not perform as well as the general-purpose behaviors. When executed in tandem with the general-purpose recovery behaviors, however, performance

**Table 1:** Performance statistics for two-robot surveillance missions with communication recovery behaviors.

Recovery Behavior Combination	Mission Success Rate (%)	Mission Success 95% Confidence Interval (+/-)	Area Covered (m <sup>2</sup> )	Area Covered 95% Confidence Intervals (+/-)	Communication Recovery Time (time steps)	Communications Recovery 95% Confidence Interval (+/-)
Control	95	5	1491	58	140	37
Retrotraverse x 2	34	14	1193	194	80	1
Nearest-Neighbor x 2	98	3	1452	48	69	8
Probe x 2	25	12	4186	644	362	86
Higher-Ground x 2	46	16	1220	132	474	269
Retrotraverse / Stop	13	10	879	139	89	13
Nearest-Neighbor / Stop	84	9	1394	50	77	9
Probe / Stop	44	16	2474	623	424	135
Higher-Ground / Stop	28	15	855	176	326	316
Retrotraverse / Nearest-Neighbor	93	7	1523	125	82	8
Retrotraverse / Probe	38	14	2331	249	306	93
Retrotraverse / Higher-Ground	22	11	1167	193	140	27
Nearest-Neighbor / Probe	99	3	1581	78	79	21
Nearest-Neighbor / Higher-Ground	90	10	1476	47	196	154
Probe / Higher-Ground	58	15	3015	517	532	189

was much better, though communications recovery time was still above the control trials.

The area covered by the robot team during these surveillance missions is at its greatest when the *Probe* behavior was a part of the communication recovery strategy. *Probe* served to direct the robot executing it into open areas where the mission could continue in a manner that allowed the robots to spread out significant distances and thus cover more ground. Most other behavioral combinations covered approximately the same amount of terrain with the exception of the *Move-To-Higher-Ground/Stop* and dual *Retrotraverse*, which tended to limit coverage due to the fact that it strictly follows previously trodden paths in its attempt to regain communications.

The statistics collected from the trials run with four robots executing the team responsibility strategy are shown in figures Table 2. Statistics from the trials run using the nearest-neighbor responsibility strategy are shown in Table 3 with the solo responsibility strategy data appearing in Table 4.

The most notable result from the experiments with teams of four robots is the evidence that the communications recovery responsibility used plays a very large role in determining mission success. In the trials utilizing the team recovery and nearest-neighbor recovery strategies, mission success was under 50% for all trials except when all robots performed the *Move-To-Nearest-Neighbor* recovery behavior. The solo recovery strategy proved most effective in terms of mission completion. By allowing the severed robot to attempt communications recovery by itself, the remainder of the team did not jeopardize the mission's success and hence allowed for greatly improved mission completion rates.

In all four-robot trials, communication recovery times improved slightly over two-robot trials in most cases as hypothesized, except in the cases that used the *Move-To-Higher-Ground* or *Probe* behavior. These trials actually resulted in significantly longer recovery periods. The area covered in the four-robot trials increased as would be expected

by the addition of two more robots, though the area covered did not double with the doubling of the team size. The combinations utilizing the *Probe* behavior once again show significantly larger areas covered in all the trials run with teams of four robots.

For all experiments, there existed two major types of recovery failures that resulted in less than desirable mission completion rates and recovery timings. The first was oscillation induced by the recovery behaviors. A large number of communications failures encountered during mission execution resulted from team members attempting to pass on opposite sides of an obstacle. Once this occurred, the recovery behaviors would activate and often would result in communications being recovered at the point prior to the failure. Without producing a significant amount of movement towards one side or the other of the obstacle, the robot's movement would often result in an oscillation where the robots would recover from a communications failure only to have the same failure re-occur once the mission was resumed.

The second major type of recovery failure found was the improper use of the context-dependent recovery behaviors. Without an incline to climb, the *Move-To-Higher-Ground* recovery behavior was ineffective in healing the network. The *Probe* recovery behavior faced a similar problem. Without nearby obstacles to direct it, the *Probe* behavior cannot react in a useful manner.

## VI. RECOVERY BEHAVIOR SEQUENCER

Drawing from the experience from the experiments described above, the recovery behaviors were found to be, in many cases, of limited use without more sophisticated techniques to choose between them in a situation-dependent manner. To improve performance the shortcomings of the recovery behaviors found in the initial experiments had to be addressed. The first area for improvement was in the process of recovery behavior selection. None of the primitive recovery behaviors by itself proved to be an adequate solution to all of the possible communication failures that could occur over the

**Table 2:** Performance statistics for four-robot surveillance with communications recovery behaviors activated with the team-responsibility strategy.

Behaviors	Mission Success Rate (%)	Mission Success 95% Confidence Interval (+/-)	Area Covered (m <sup>2</sup> )	Area Covered 95% Confidence Intervals (+/-)	Communication Recovery Time (time steps)	Communications Recovery 95% Confidence Interval (+/-)
Control	79	6	2308	115	235	35
Retrotraverse x 4	0	0	1057	165	67	1
Nearest-Neighbor x4	80	11	2260	154	93	3
Probe x 4	10	4	9474	672	409	42
Higher-Ground x4	10	10	1565	254	1025	232
Retrotraverse x2 / Nearest Neighbor x2	4	4	1400	232	111	34
Retrotraverse x2 / Probe x2	3	3	5604	560	531	68
Retrotraverse x2 / Higher-Ground x2	3	2	1323	203	1074	224
Nearest Neighbor x2 / Probe x2	32	12	6121	327	327	43
Nearest-Neighbor x2 / Higher Ground x2	32	5	1560	222	228	62
Probe x2 / Higher-Ground x2	11	8	6050	501	571	567

**Table 3:** Performance statistics for four-robot surveillance with communications recovery behaviors activated with the nearest-neighbor responsibility strategy.

Behaviors	Mission Success Rate(%)	Mission Success 95% Confidence Interval (+/-)	Area Covered (m <sup>2</sup> )	Area Covered 95% Confidence Intervals (+/-)	Communication Recovery Time (time steps)	Communications Recovery 95% Confidence Interval (+/-)
Control	79	6	2308	115	235	35
Retrotraverse x 4	3	3	1338	149	68	1
Nearest-Neighbor x4	75	15	2310	137	74	7
Probe x 4	24	8	9658	862	346	377
Higher-Ground x4	14	10	1806	218	214	42
Retrotraverse x2 / Nearest Neighbor x2	2	2	1533	195	75	3
Retrotraverse x2 / Probe x2	6	4	5699	544	395	55
Retrotraverse x2 / Higher-Ground x2	3	3	1438	205	137	32
Nearest Neighbor x2 / Probe x2	38	13	5811	665	241	34
Nearest-Neighbor x2 / Higher Ground x2	26	13	1998	128	134	22
Probe x2 / Higher-Ground x2	12	10	6041	545	351	44

**Table 4:** Performance statistics for four-robot surveillance with communications recovery behaviors activated with the single-robot responsibility strategy

Behaviors	Mission Success Rate(%)	Mission Success 95% Confidence Interval (+/-)	Area Covered (m <sup>2</sup> )	Area Covered 95% Confidence Intervals (+/-)	Communication Recovery Time (time steps)	Communications Recovery 95% Confidence Interval (+/-)
Control	79	6	2308	115	235	35
Retrotraverse x 4	32	9	1898	167	142	7
Nearest-Neighbor x4	76	12	2206	202	138	118
Probe x 4	79	6	4998	597	472	98
Higher-Ground x4	67	7	2382	223	448	142
Retrotraverse x2 / Nearest Neighbor x2	59	9	2164	201	151	17
Retrotraverse x2 / Probe x2	59	8	4380	508	286	39
Retrotraverse x2 / Higher-Ground x2	38	10	2194	232	204	19
Nearest Neighbor x2 / Probe x2	72	7	4033	445	257	53
Nearest-Neighbor x2 / Higher Ground x2	68	10	2261	208	125	15
Probe x2 / Higher-Ground x2	69	8	4346	554	787	250

course of the mission. If the robot could select recovery behaviors in a manner to best utilize the context-dependent recovery behaviors, it was believed that performance would improve significantly. The second shortcoming to address involved the notion of cognizant failure [10]. If the robots could recognize a recovery behavior as ineffective in a particular situation and switch to a more effective behavior, communication recovery times and mission completion rates could be improved greatly.

Situated-activity based systems have looked at similar problems in context-dependent action selection [2,9]. Drawing inspiration from this paradigm and in particular Firby’s RAPs architecture [9], we designed a recovery behavior sequencer to provide better contextual use of recovery behaviors and to provide a notion of cognizant failure whereby an ineffective recovery behavior’s execution can be discovered, halted, and replaced with another more effective recovery behavior.

The behavior sequencer designed to achieve these goals can be roughly divided into three parts. The first is the communications recovery behavior library. This library contains all the possible recovery behaviors that may be selected for execution as well as information concerning the situational context in which they can be applied. This will ideally prevent the execution of a recovery behavior outside of its proper context (e.g., *Move-To-Higher-Ground* when there are no significant inclines to climb). The second component is the failure history module, which contains activation records for the different recovery behaviors in the library as well as statistics concerning the duration of currently executing recovery behaviors and records of failed recovery attempts using each behavior. The final component is the sequencer itself. Utilizing input derived from current communication statistics, environmental perceptual features, and information residing in the failure history module, the sequencer will perform one of four actions: choose a recovery behavior to execute, fail the currently executing recovery behavior and choose another, switch recovery behavior execution without indicating behavioral failure, or do nothing and allow the currently executing recovery behavior to continue running. A block diagram of the recovery behavior sequencer can be seen in Figure 4.

Action selection is handled by the sequencer in a number of ways. The first is matching possible recovery behaviors by comparing current environmental and communication features to the context provided for each behavior in the library. If multiple behavioral contexts are matched, the failure history module is consulted to determine which behavior has been the most successful in the past. That behavior is then chosen and executed. The second means by which actions can be selected is through the use of monitors. A monitor in this context involves the detection of a specific perceptual feature that causes the sequencer to interrupt the currently executing behavior and re-evaluate the recovery behavior selection. An example would be a monitor that is triggered by the presence of steep inclines in the vicinity of the robot. This would cause the sequencer to reconsider the recovery behavior in current use and potentially choose the *Move-To-Higher-Ground* behavior for execution. The final method by which a recovery

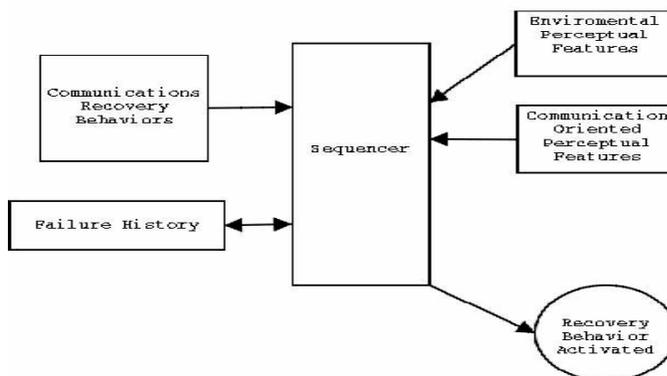


Figure 4. Block diagram of communications recovery behavior sequencer

behavior may be selected is due to the failure of the currently executing recovery behavior. In this case, the environment and communications states are re-evaluated and a new more appropriate recovery behavior is chosen.

Cognizant failure was incorporated into the sequencer principally through tracking of recovery behavior execution time. The sequencer checks the execution time of the currently running recovery behavior at each time step. When the execution time has surpassed a given threshold, a failure is indicated in the history module and a new recovery behavior is selected.

## VII. SEQUENCER EVALUATION

The recovery behavior sequencer was evaluated in the same manner as the previous experiments described in this paper. Utilizing teams of four robots, twenty trials were run over the two environments utilizing the same three communication recovery responsibility strategies as in the previous four-robot testing. Statistics concerning mission completion, communications recovery time, and mission area coverage were collected. Figures 5, 6, and 7 depict the results of the experiments.

The mission completion rates for all the sequencer runs slightly exceeded the control runs, with the solo recovery once again proving to be most effective in mission completion. Coverage exceeded the control by a significant margin when the recovery behavioral sequencer utilized the team recovery and nearest-neighbor recovery strategies and was only surpassed by the trials that utilized *Probe* in the original experiments. The recovery times for the sequencer-based experiments with team-based recovery matched the best runs in the previous set of four-robot experiments. The other responsibility strategies resulted in recovery times that were not significantly faster than the control.

## VIII. DISCUSSION

The behaviors discussed here provide a basis in which to implement reactive communication recovery. These results show evidence that the *Move-To-Nearest-Neighbor* behavior is the most effective general-purpose communications recovery tested to date. Retrotravel, while providing quicker

communication recovery times than the other behaviors in larger scale teams, often results in oscillation in the robots' movement, impairing the team's ability to successfully complete its mission. Contextual recovery behaviors, while providing some advantages in coverage, do not appear to be effective without a means to better determine their applicability in a particular communications recovery task.

Three different responsibility strategies were investigated for use in larger scale teams requiring reactive communications recovery. It was found that without integrating additional capabilities into the robot controller's action-selection mechanism as well as facilities to determine the failure of a particular recovery behavior, the strategies which focus on communication recovery over mission completion result in severe degradation in mission completion rates although communications recovery rates do increase. Utilizing a behavioral sequencer to select among multiple recovery behaviors based upon past success and current environmental features provides an effective means to combat a number of problems found in the experiments conducted using static combinations of recovery behaviors. By employing a simple means of cognizant failure (timeout) to switch away from behaviors that did not succeed in communications recovery, mission completion rates increased significantly over trials that utilized fixed recovery behavior selection.

There does not appear to be a decisive choice in recovery behaviors nor responsibility strategies for all the metrics measured, though the trials that used the sequencer provided higher mission completion rates than those conducted with static behavioral combinations. Of the experiments conducted with the sequencer, the results using team recovery behavior appear to provide an adequate compromise between coverage, mission completion, and communication recovery speed.

### IX. HARDWARE IMPLEMENTATION

The individual recovery behaviors, with the exception of *Move-to-Higher-Ground*, have been tested in a number of scenarios on actual robotic platforms. These communications recovery behaviors were demonstrated on two ATRV-JR robots equipped with laser scanners for obstacle detection, differential GPS for localization, and 802.11b wireless bridges for communication. On Georgia Tech intramural fields, proof-of-concept missions were run utilizing the simulated free-space attenuation network model described earlier in this paper. Communications recovery behaviors were also tested using actual network connectivity data found while on-site at the MOUT site in Fort Benning, Ga. Figure 8 illustrates a typical experimental hardware run on the Georgia Tech campus. Initial results with two robots indicate consistent results with those found in simulation and additional data for missions run on hardware platforms are currently being collected for in depth comparison to the simulation results.

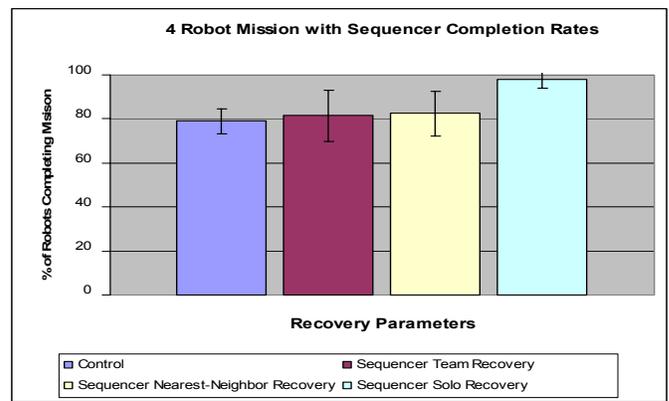


Figure 5. Mission completion rates for 4 robots utilizing the sequencer. Error bars indicate 95% confidence intervals. The team utilizing the sequencer and the solo recovery strategy performs significantly better than the control as well as all previous static behavioral combinations. Higher values are better.

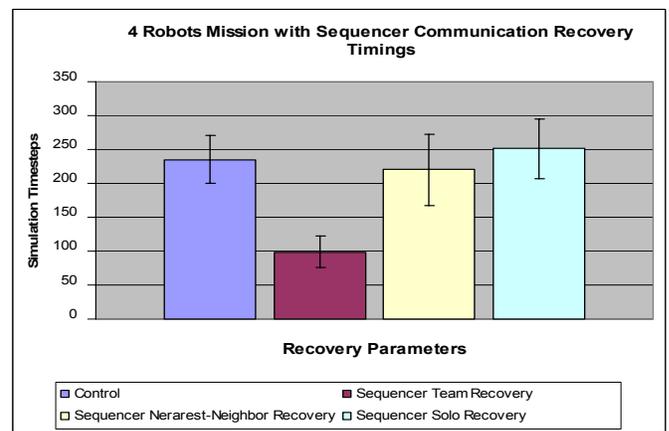


Figure 6. Time spent executing recovery behaviors until communication is reestablished for 4 robot trials utilizing the sequencer. The team utilizing the sequencer and the solo recovery strategy recovers communication significantly faster than the control but somewhat slower than some static combinations such as teams all performing *Retrotraverse* or the *Move-To-Nearest Neighbor* behavior. Lower values are better.

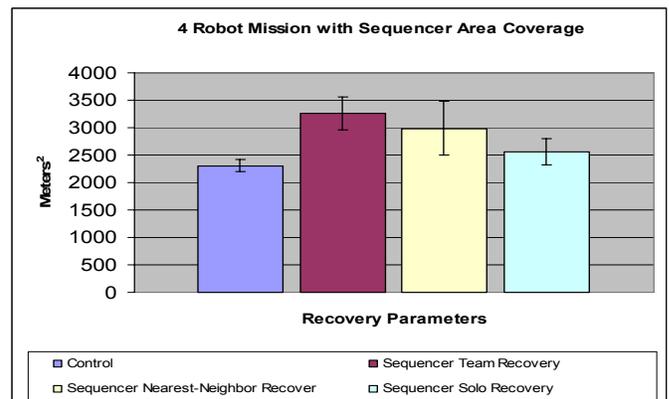


Figure 7. Area coverage for the 4 robot team utilizing the sequencer. The teams utilizing the sequencer and the solo or team recovery strategies perform significantly better than the control and all static combinations that did not utilize the *Probe* behavior. Higher values are better.



a) Two robots approach an artificial building.



b) Communications is broken and the robots begin the execution of their recovery behaviors. The robot on the left executes *Retrotraverse*, the robot on the right executes *Move-To-Nearest-Neighbor*.



c) Communications is reestablished and mission is resumed.

Figure 8: Communication recovery experiment run on the Georgia Tech campus.

## X. CONCLUSIONS AND FUTURE WORK

A series of reactive communications recovery behaviors have been presented and their performance measured in teams of two and four robots performing a surveillance mission in an urban environment. The utility of three different communications recovery responsibility strategies was also examined and it was found that the choice of which robot(s) responds to a communications failure can greatly affect mission completion rates and communication recovery speed. Further, a behavioral sequencer was used to alleviate some of the problems found during static recovery behavior selection.

This research leaves a number of open questions concerning reactive communications recovery for future investigation. Issues still remain as to the best methods for determining responsibility for communications recovery in large-scale teams [1] beyond these existing strategies. By no means is our suite of recovery behaviors exhaustive, where a possible avenue for further research may include further evaluations of other communications recovery behaviors such as a shepherding behavior in which a robot is dynamically assigned

to guide 'lost' robots back to the network [19]. Additional investigations concerning the impact of unmanned aerial vehicles on communications recovery are also expected to prove fruitful and will be considered in the near future.

## REFERENCES

- [1] A. Agah and G.A. Bekey, "In a team of robots the loudest is not necessarily the best", Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics, October, 1995.
- [2] P. Agre, D. Chapman, "Pengi: An implementation of a theory of activity", Proceedings AAAI-87, 1987.
- [3] S. Anderson, R. Simmons, and D. Goldberg, "Maintaining line of sight communications networks between planetary rovers", Proceedings of the Conference on Intelligent Robots and Systems (IROS), October 2003.
- [4] R.C. Arkin, J. Diaz, "Line-of-sight constrained exploration for reactive multiagent robotic teams," AMC 7<sup>th</sup> International Workshop on Advanced Motion Control, 2002.
- [5] R.C. Arkin, "Behavior-based robot navigation for extended domains", Adaptive Behavior, 1(2), pp. 201-225, 1992.
- [6] R.C. Arkin., "Motor-Schema-Based Mobile Robot Navigation", International Journal of Robotics Research, Vol. 8, No. 4, August pp. 92-112., 1989.
- [7] T. Balch, R.C. Arkin, "Communication in reactive multiagent robotic systems", Autonomous Robots, 1(1) , pp. 27-52, 1994.
- [8] G. Comparetto, S. Kao, J. Marshall, and N. Schult "OPNET path attenuation routine (OPAR) description document", MITRE Tech. Rep., V. 1.0, Aug. 2001.
- [9] J. Firby, "Adaptive execution in complex dynamic domains", Ph.D. Thesis, Yale University Technical Report YALEU/CSD/RR #672, January 1989.
- [10] E. Gat. "Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots", Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 809-815, 1992.
- [11] Georgia Tech Mobile Robot Laboratory, User Manual for *MissionLab* version 5.0, [www.cc.gatech.edu/ai/robot-lab/research/MissionLab/](http://www.cc.gatech.edu/ai/robot-lab/research/MissionLab/), January 2002.
- [12] H. Hu, I. Kelly, D. Keating, D. Vinagre, "Coordination of multiple mobile robots via communication", Proceedings of SPIE/98 Mobile Robots XIII Conference, pp. 94-103, Nov. 1998.
- [13] R.R. Murphy, "Marsupial and shape-shifting robots for urban search and rescue", IEEE Intelligent Systems 15(2) p. 14-19, 2000.
- [14] H.G. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta, and J.M. Spector, "Autonomous communication relays for tactical robots," IEEE 11<sup>th</sup> International Conference on Advanced Robotics (ICAR 2003), 2003.
- [15] M. Powers and T. Balch, "Value based communications recovery for mobile robots", unpublished.
- [16] R. Ramanathan and J. Redi, "A brief overview of ad-hoc networks: Challenges and directions", IEEE Communications Magazine, pp 20-22 May 2002.
- [17] A. Stroupe and T. Balch, "Behavior-based mapping and tracking with multi-robot teams using probabilistic techniques", Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA '03), May, 2003.
- [18] J. Sweeney, T.J. Brunette, Y. Yang, and R. Grupen, "Coordinated teams of reactive mobile platforms", Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA '02), May, 2002.
- [19] R. Vaughan, N. Sumpter, A. Frost and S. Cameron, "Robot control of animal flocks" Proceedings of the International Symposium of Intelligent Control, 1998.
- [20] A. Wagner, R.C. Arkin, "Internalized plans for communications-sensitive robot team behaviors", the Conference on Intelligent Robots and Systems (IROS), October 2003.