Establishing Line-of-Sight Communication Via Autonomous Relay Vehicles

Md Mahbubur Rahman¹, Leonardo Bobadilla¹, Brian Rapp²

Abstract—In communication-denied scenarios or contested environments, Line-of-sight (LoS) communication (by infrared or visible light) becomes one of the most reliable ways to send information between mobile units. In this paper, we consider the problem of verifying and repairing the visibility-based communication network formed among a number of servicing vehicles and military units that are dispersed in an environment with obstacles. We first formulate the problem in terms of algebraic graph theory to verify its connectivity. If any disconnection occurs due to a unit going out of the visibility polygons of all the vehicles, we propose a solution to repair it by dispatching a single vehicle that is optimal in terms of visibility and relocation cost. We tested our ideas through computer simulations in several case studies based on different mobility models.

1. INTRODUCTION

Communication between units located in various different positions of an environment is a matter of vital military importance. Communications can be interrupted by natural features such as terrain and atmospheric effects, and electromagnetic signals can also be hampered by interference. Intentional jamming of communications by the enemy may pose a serious risk. It is possible that the Line-of-Sight (LoS) based communication solves many of these problems. This form of communication is more difficult to intercept or jam, because to do so would require the attacking element to be located directly on that LoS. Because mission-related movements of land forces may naturally cause them to lose LoS with their friendly units, it is desirable to provide additional nodes or relays that can maintain communication between the units. An autonomous ground vehicle can fulfill this role, by moving from place to place as needed for the purposes of establishing relayed contact.

Once a number of autonomous ground vehicles have been established in the field (see Figure 1), their computational capacities can be used to provide services to the mobile units. This can provide additional military value, by analyzing tactical data, detecting threat patterns, or searching for information that would otherwise not be readily available. If communications to the outside world are interrupted, degraded, or simply too low-bandwidth for this purpose, then a locally-available high-performance computer may be the best solution. However, the mobility of the deployed units may cause them to go outside of visibility polygons of the vehicles. Therefore we require a solution that is able to check this type of disconnection and repair it accordingly. In this paper we propose solutions that can effectively report any LoS based disconnection and select a vehicle to be dispatched in order to reconnect the lost units while maintaining the existing connectivity.

Since the serving vehicle must have LoS communication with all the units it needs to serve, our ideas are naturally connected to Art Gallery problems [21], [22]. Art-gallery based approaches have been used in vehicles to solve sensor [8] and landmark placement [6] problems. Another computational geometry problem that is connected to our ideas is the Watchman Route [19] problem. Some of the differences between the traditional watchman route and our setup are: 1) we are not only concerned about the shortest path but also about a path that will keep the most visibility with all units and 2) our path must respect the differential constraints of the vehicle.

Also related to our problem are visibility-based pursuit evasion schemes whose goals are to find a path that will guarantee that an evader is captured regardless of its motion [9]. Our work is also closely connected to path planning approaches that attempt to maintain visibility with a single static landmark [4], [20].

The idea of a powerful mobile unit uploading, downloading, and distributing data to a set of static nodes has been explored in data muling [2], [24]. One important difference between our formulation and the data muling approach is that instead of the proximity of sensor nodes, the communication is based on line-of-sight. In the area of communication, Free-Space Optical Communications (FSOC) [12] is being considered as an alternative for military network-centric operations. Particularly related is the work in [16] where the problem of two mobile nodes that try to maintain LoS alignment is studied.

The contributions of the paper are the following: 1) We propose a way to convert the problem of maintaining a visibility-based network into a graph problem, and apply algebraic graph theory method to check the validity of the communication network. We develop and implement an algorithm and analyze its computational complexity; 2) We propose solutions to recover the network once it gets disconnected. The proposed approach finds the goal region for relocating a single vehicle and check whether this relocation repairs the disconnection; and 3) We implemented our ideas and tested them in a number of case studies through computer simulations.

The rest of the paper is organized as follows. Section II introduces the necessary mathematical notation, preliminaries

¹M. Rahman and L. Bobadilla are with the School of Computing and Information Sciences, Florida International University, Miami, FL, 33199, USA. mrahm025@fiu.edu, bobadilla@cs.fiu.edu
²B. Rapp is with the Army Research Lab, USA. brian.m.rapp2.civ@mail.mil
and the problems of interest. Section III presents algorithmic solutions to the problem we formulated in Section II. Section IV presents a number of detailed case studies. Finally, Section V includes a discussion of the results and directions for future work.

II. PRELIMINARIES

Assume that a mission takes place on a closed planar environment, \( W = \mathbb{R}^2 \) as shown in Figure 1. There are \( n \) vehicles, \( A_1, A_2, \ldots, A_n \), equipped with high-performance computing devices that need to serve \( m \) units \( B_1, B_2, \ldots, B_m \) scattered in a field. \( O \) is the set of obstacles that are modeled as polygons and impenetrable by both the units and vehicles. Therefore the collision-free space is defined as \( E = W \setminus O \). The mobile units present in the field can move freely in \( E \) and are modeled as point robots. A unit \( B_j \) has configuration space \( B_j \) where a particular configuration \( b_j \in B_j \) is defined as \( b_j = (x, y) \in E \). The vehicles are modeled as car-like robots i.e., they cannot move sideways which incorporate differential constraints on their movements. A vehicle \( A_i \) has a configuration space \( C_i \) and the positions \( q_i \in C_i \) are defined as, \( q_i = (x, y, \theta) \in E \times [0, 2\pi] \) [17]. Vehicle dynamics for \( A_i \) are defined as \( \dot{x}_i = u_i^f \cos \theta_i \), \( \dot{y}_i = u_i^f \sin \theta_i \), and \( \dot{\theta}_i = \frac{1}{M} \tan \alpha_i \) [11], where \( u_i^f \) is the forward speed and \( u_i^\alpha \) is the steering angle of the vehicle. We define the entire system state space to be \( X = C_1 \times C_2 \times \cdots \times C_n \times B_1 \times B_2 \times \cdots \times B_m \). Let \( X_{obs} = \{ x \in X : x \cap O \neq \emptyset \text{ where } O \in \mathcal{O} \} \) be the obstacle state space. The collision-free state space is then \( X_{free} = X \setminus X_{obs} \).

A. Communication State Validity

In our visibility-based network, communication can only be established among servicing vehicles and between servicing vehicles and mobile units through Line-of-Sight (LoS). In a complex environment filled with many obstacles, LoS is disrupted frequently by the motions of the units and vehicles. To formalize the problem, we therefore characterize the communication range of a particular vehicle by a visibility polygon. The visibility polygon \( V(p) \) for a point \( p \in E \) is defined as [5]:

\[
V(p) = \{ w \mid w \in E \text{ and } \overline{pw} \cap E = \overline{pw} \}.
\]

Therefore \( V(q_i) \) is the visibility polygon for vehicle \( A_i \) where we assume that the configuration \( q_i \in C_i \) approximates the points (without rotation) where the vehicle is currently placed in \( E \).

A mobile unit must reside inside the visibility polygon of at least one vehicle in order to receive service. Additionally, a servicing vehicle must stay in the visibility polygon of at least one other servicing vehicle and collectively they must form a connected relay network to make the state \( x \in X \) completely communication-valid.

Definition 2.1: A state \( x \in X \) is considered communication-valid if and only if each unit is visible by at least one servicing vehicle and all the servicing vehicles form a connected graph. We define this set of configurations as \( X_{comm} \subseteq X \).

According to the above definition we must satisfy the following two conditions in order to have a communication-valid state:

\[
\forall j, \exists i \text{ s.t. } b_j \in V(q_i) \text{ for } 1 \leq j \leq m \text{ and } 1 \leq i \leq n
\]

\[
\{(q_j, q_k) | q_k \in V(q_j) \text{ for } 1 \leq j, k \leq n, k \neq j \} \equiv CC(x)
\]

where \( CC(x) \) is a connected component formed by all the vehicle-vehicle connections.

The mobile units move independently from the vehicles, following one of the many mobility models [1]. Consequently the servicing vehicles may lose visibility with one or more units, making the relay network communication-invalid. Any such mobility event also results in the changes of configurations for the relocated units and modifies the current state space \( x \in X \). Therefore, we have a decision problem in which we want to know whether a given state is communication-valid \( (x \in X_{comm}) \). This problem can be formulated as follows:

Problem 1: Communication State Validation

Given the workspace \( W \), a set of obstacles \( O \), a set of configurations \( C \) for servicing vehicles, and \( B \) for mobile units, determine whether a state \( x \in X_{comm} \) or not.

B. Invalid-to-Valid Communication State Restoration

Initially, we assume that the visibility-based network is connected as shown in Figure 1 and \( x \in X_{comm} \). Since the mobile units are allowed to move freely throughout the environment \( E \), the system becomes communication-invalid frequently. A unit is marked as disconnected if and only if it is not visible to any of the servicing vehicles. A set, \( D \subseteq \{B_1, B_2, \ldots, B_m\} \), of disconnected units is defined based on a given state \( x \in X \) and we dispatch an available vehicle \( A_i \) from its current location \( x_i^1 \) to a newly computed goal region \( X_{G}^i \) in order to reconnect the strayed units. As the event of vehicle relocation must not break the existing partially connected network, the selection of vehicles to be moved must be done carefully. This motivates the following problem:

Problem 2: Communication Validity Restoration

Given \( W \) and \( O \), the current state space \( x \in X \), and a set of disconnected units \( D \), select a number of vehicles to relocate...
and compute their new goal region, $X_G$, that will reconnect all the units in $D$.

### III. METHODS

#### A. Checking Communication Validity/Feasibility

In this section we provide the solution to verify whether the current state $x \in X$ is communication-valid which solves Problem 1 demonstrated in Section II. Initially, we are given with the positions of units and vehicles and we do not know which vehicle is providing service to which unit. In order to solve problem 1, we propose Algorithm 1 that works based on graph theoretic network connectivity [25], [23] solutions.

A visibility-based graph can be constructed where the node set is composed of all the components (vehicles and units) and an edge is added between two nodes if the corresponding components are visible to each other. However checking the algebraic connectivity [25] on this graph is not sufficient. For an example the graph shown in Figure 2(b) is connected but not communication-valid. These types of graphs result if there are obstacles between vehicles. Therefore our proposed validation is two-fold and we generate the following two types of undirected graphs based on a state space $x$.

**Vehicle Relay Graph ($G_A$):** This state dependent undirected graph is a mapping $g_A : X \rightarrow G_A(V_A, E_A)$, where $V_A = \{A_1, A_2, \ldots, A_n\}$ is the set of vehicle nodes (see Figure 2(a)). $E_A$ denotes the set of edges defined as,

$E_A = \{e_{ij} | q_i \in V(q_j)\}$

where $q_i$ and $q_j$ are the positions of vehicles $A_i$ and $A_j$ in the environment $E$. This implies that an edge $e_{ij}$ exists if and only if the vehicle $A_i$ is inside the visibility polygon, $V(q_j)$, of vehicle $A_j$. We then compute the $n \times n$ Laplacian matrix, $L(G_A) = D E G(G_A) - A D J(G_A)$, where $A D G(G_A)$ is the familiar $(0, 1)$ adjacency matrix, and $D E G(G_A)$ is the diagonal matrix of vertex degrees [18], also called Valency matrix of $G_A$. The entries of $L$ are as follows [3]:

1. $l_{ij} = \begin{cases} -1 & \text{if an edge exists between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}$

2. $l_{ii} = -\sum_{k=1, k \neq i}^{n} l_{ik}$

In line 2 of Algorithm 1, we check the second smallest eigenvalue $\lambda_2(L(G_A))$ of $G_A$ to see whether it is positive. A non-positive value indicates that the relay network formed by all the vehicles does not exist and the network is communication-invalid (line 3). If $\lambda_2 > 0$, then we go to the second step of validation where we check the entire network connectivity (lines 5 - 11).

**Unit Graph ($G_B$):** The unit graph $G_B$ is computed in line 5 which is also undirected and is a mapping $g_B : X \rightarrow G_B(V_B, E_B)$. In contrast to $G_A$, the graph $G_B$ includes all the $m$ units and $n$ vehicles in its node set $V_B$. Accordingly, $V_B = \{A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m\}$ is indexed by the vehicles followed by the units where all units have indexes greater than $n$. The edge set $E_B$ has the following form,

$E_B = \{e_{ij} | b_j \in V(q_i) \}$ where $n < j \leq n+m$ and $0 < i \leq m$.

This means that an edge is added if an only if a unit’s position $b_j$ is visible from some vehicle’s position $q_i$ (see Figure 2(b)).

Finally we form a state dependent graph $G(V, E)$ as shown in Figure 2(c) which is the union of the above formed two graphs $G_A$ and $G_B$. Accordingly the vertex set $V = V_A \cup V_B$ and the edge set $E = E_A \cup E_B$. Therefore we conclude that the graph is communication-valid if the second smallest eigenvalue $\lambda_2$ of the $(m + n) \times (m + n)$ Laplacian matrix, $L(G)$ of graph $G(V, E)$ is greater than zero (lines 7 - 11).

**Algorithm 1** communicationCheck($x, O$)

1: $G_A = g_A(x)$
2: if $\lambda_2(L(G_A)) \leq 0$ then
3: return false
4: end if
5: $G_B = g_B(x)$
6: $G = G_A \cup G_B$
7: if $\lambda_2(L(G)) \leq 0$ then
8: return false
9: else
10: return true
11: end if

**Analysis of Algorithm 1:** The graph creation in lines 1 and 5 use a visibility polygon computation algorithm to estimate the edges of the graphs. Each polygon computation takes $O(n^3)$ [5] and for $n$ vehicles the running time is $O(n^2)$. But the most dominant factor is in computing the eigenvalues (lines 2 and 7) which generally takes $O(n^3)$ in the worst cases. Therefore the running time of Algorithm 1 is $O(n^3)$.

#### B. Recovering a Communication-valid State with a Single Vehicle

As the units are on the move, this may result in disconnections to their respective servicing vehicles. Here we propose a solution that dispatches a single vehicle in order to reconnect a strayed unit from the visibility-based network. Any movement inside a network triggers Algorithm 2 that identifies any disconnections and relocates a vehicle which best re-establish a communication-valid state without affecting existing network connections. The set of disconnected units $D$ is defined as,

$D = \{B_j | \forall i, b_j \notin V(q_i) \}$ where $1 \leq i \leq n$.

Therefore $D$ is the set of units that are not visible to any of the vehicles due to obstacles. In other words, the set of all the units with degree zero in the graph $G_B$ compose the disconnected set $D$. If there is any such visible unit, i.e.,
\( D \neq \emptyset \), we attempt to resolve disconnections for each \( B_j \in D \) using Algorithm 2.

Next we define the set \( H_i \) as the set of hard constrained units of a vehicle \( A_i \), which are only visible from \( A_i \), and to which no other vehicles can provide service. A unit is said to be a hard constrained unit if and only if it is visible from only one vehicle. Lines 1–3 of Algorithm 2 compute \( H_i \) for all the vehicles according to the following equation:

\[
H_i = \{ b_j | b_j \in V(q_i) \text{ and } \forall k \neq i, b_j \notin V(q_k) \}. \tag{7}
\]

Alternately, the nodes with degree one in the graph \( G_B \) are the members of hard constrained set of the vehicles connected through their only edges. In line 4 we compute the intersecting polygon \( V(H_i) \) of all visibility polygons of all members in \( H_i \). Initially the set of candidate vehicles for relocation is, \( C = \{ A_1, A_2, \ldots, A_n \} \). However, we may not be able to relocate all the vehicles in \( C \) as this may break the existing connected graph topology \( G_A \) among the vehicles. Therefore we check the second smallest eigenvalue of the Laplacian matrix of a graph generated by removing each of the corresponding vehicle nodes \( A_i \in C \) along with its incident edges from graph \( G_A \). We remove the nodes from \( C \) that make \( \lambda_2 \leq 0 \) (line 6 of Algorithm 2).

The new goal polygon \( X_G^i \) of a candidate vehicle \( A_i \in C \) must be inside the visibility polygons of 1) the disconnected unit \( B_j \) and 2) at least one other vehicle that is a part of the existing relay network. Moreover if there is any hard constrained unit and \( H_i \neq \emptyset \) then \( X_G^i \) must be inside the polygon \( V(H_i) \). As the visibility polygons are concave in an environment filled with obstacles, we may get multiple goal polygons. In such cases, we take the largest one. Therefore we compute \( X_G^i \) for \( A_i \in C \) as follows (see line 8):

\[
X_G^i = \begin{cases} 
\max_{A_k \neq A_i, 1 \leq k \leq n} V(b_j) \cap V(q_k); & \text{if } V(H_i) = \emptyset \\
\max_{A_k \neq A_i, 1 \leq k \leq n} V(b_j) \cap V(q_k) \cap V(H_i); & \text{otherwise}
\end{cases} \tag{8}
\]

Once the goal regions for all the candidate vehicles in \( C \) are computed, we only retain the vehicles that has nonempty goal region (line 10). We then select the optimal vehicle \( A_{s_i} \) in terms of the motion cost. In brief, the motionCost() method in line 14 computes the relocation cost of a vehicle from its current position \( x_i^s \) to the computed goal region \( X_G^i \) using a motion planning algorithm such as Rapidly Exploring Random Tree Star (RRT*) [14].

IV. EXPERIMENTAL RESULTS

A. Checking Communication-Valid State

In the first case study, we validate the correctness of Algorithm 1 to check the communication-valid state space. The results from our experiments on different setups of the environment are shown in Figure 3. In Figure 3(a) we have few trivial environments where only one graph is communication-valid (bottom right with \( \lambda_2(G_A) = 2 \) and \( \lambda_2(G) = 3 \)). A complex environment filled with three obstacles, two vehicles and six units are presented in Figure 3(b). Here the relay network is connected as the second smallest eigenvalue of the vehicle graph’s Laplacian is \( \lambda_2(G_A) = 2 > 0 \). But the union graph including vehicles and units results in \( \lambda_2(G) = 0 \) which indicates that the set up is not communication-valid.

Another environment is shown in Figure 3(c) where the relay network is not communication-valid \( \lambda_2(G_A) = 0 \), although the union graph is connected. Finally in Figure 3(d) we demonstrate a communication-valid network with three vehicles where both the relay graph and union graph are connected \( \lambda_2(G_A) = 1 \) and \( \lambda_2(G) = 0.5024 \).

B. Regaining a Communication-valid State by Single Vehicle Movement

We used the Bonnmotion Library [1] to generate different mobility models. Also the CGAL library [7] was used to perform the geometric polygon computation and Python programming language was used for visualization. SMP library [13] was used for RRT* algorithm implementation.

In Figure 4, we present the test cases for random waypoint mobility model where four servicing vehicles are assigned to monitor six deployed units. While the units are moving randomly, the Unit \( E \) gets disconnected from the
network. Therefore we demonstrate the computation of our proposed Algorithm 2 in Figures 4(a)-(d) in order to recover the network. As the events of relocating vehicle 1 and 3 make vehicle 2 and 4 disconnected from the network respectively, they are eliminated from consideration and the candidate vehicle set becomes $C = \{2, 4\}$. In Figure 4(b) we compute the goal region $X^2_G$ for vehicle 2. We have three regions to consider from the intersection of the visibility polygons marked by dotted lines. Among those we select the region labeled as “E, 4” as $X^2_G$ which is the largest among the three. Similarly we compute the region “E, 3” as the goal region $X^3_G$ for vehicle 4 shown in Figure 4(c). Finally, as shown in Figure 4(d), we generate two motion paths corresponding to the vehicles 2 (red) and 4 (blue) using the RRT* [15] motion planning algorithm for a Dubins car [11]. We select vehicle 4 for relocation by following the blue trajectory as it gives an optimal cost compared to the red trajectory which requires a longer path to travel.

In Figure 5(a) we have three available vehicles forming a relay network while serving six units. Unit D gets disconnected and the candidate vehicle set for relocation is $C = \{2, 3\}$ as relocating vehicle 1 makes the relay network broken. As both of them have hard constrained units ($H_2 = \{B\}; H_3 = \{E, F\}$), we use (8) to calculate the intersecting polygon $X^2_G$ and $X^3_G$ as their respective goal regions. The resulting region that optimizes the visibility is shown as dashed area for vehicle 2, which is the intersecting region of vehicle 3, hard constrained unit $H_2 = \{B\}$, and disconnected unit D. Vehicle 2 is then relocated into the purple dashed region following the trajectory generated by the RRT* algorithm. At time = 5 as shown in Figure 5(b), the hard constrained unit B of vehicle 2 changes position and does not break the connectivity. At time = 7 (Figure 5(c)) we dispatch vehicle 1 to serve the disconnected unit E after the same computations done for the above cases. However at time = 15 (Figure 5(d)), the system becomes
non-recoverable (unit \( A \) gets disconnected) as we cannot move either of the vehicles 1 and 2 due to no common visibility polygon among hard constrained units, one other vehicle and disconnected unit \( A \). Vehicle 3 cannot be moved due to relay connectivity.

**Nomadic Mobility Model:** Units move in groups according to the nomadic mobility model and an example scenario is presented in Figure 6. We deployed two servicing vehicles in order to provide service to six units that are distributed into two groups (see Figure 6(a)). Unit \( B \) gets disconnected in next time-stamp as shown in Figure 6(b). Accordingly, the vehicle 1 goes to the intersection of the visibility polygons of the other vehicle 2 and unit \( B \) (purple dashed). This little movement is highlighted by a red curvature generated by the RRT* algorithm.

In Figure 6(c), the unit \( A \) is disconnected and the vehicle 2 is dispatched to its calculated goal location \( X_D^2 \) (intersection of the vehicle 1 and unit \( A \)). Then in the next time-stamp, unit \( F \) is disconnected. Only vehicle 1 has a common intersection with vehicle 2, hard constrained unit set \( H_1 = \{ B, C, D \} \) and disconnected unit \( F \). Therefore we relocate vehicle 1 to repair the LoS based visibility network. We observed that the Nomadic mobility model is easier to repair than the Random waypoint model with a single vehicle movement as the units move in groups.

**V. CONCLUSIONS AND FUTURE WORK**

In this paper, we presented a computationally efficient and practically implementable methodology to check and repair the LoS based communication network on a field mission. Our proposed methodology uses a two step algebraic graph theoretical solution to proactively verify the current status of the visibility based network, given a particular mission state that includes all the vehicles and units position. Secondly, we developed an efficient combinatorial algorithmic solution that relocates a single vehicle in order to repair a disconnected network caused by the mobility of the units. Finally, we showed illustrative examples to demonstrate the effectiveness of the proposed model using different mobility models. Several interesting directions are left for future work.

In our formulation we dispatch only one vehicle to repair any disconnections. One immediate extension of this work is to remove this constraint and allow multiple vehicles to relocate. Although this may cause temporary disconnections in the existing network, it will give optimal communication quality. Additionally a patrolling scenario can be formulated in case of an insufficient number of servicing vehicles. Another extension of our work is to remove assumptions about the known world, \( W \), and obstacles, \( O \). We assumed that the obstacles are known beforehand and the layout can be perfectly decomposed. Ideally, the robot equipped with sensors can create a strategy based on visibility events [17] to explore the environment and find good LoS locations. We are exploring the related problem of finding competitive strategies for a kernel polygon search and if they can be implemented in the mobile vehicle with sensors [10].

In our experimental setup we tested two well known mobility models, 1) random waypoint and 2) nomadic. We found our proposed methodology very effective in most of the cases to resolve disconnection if they start from an initial connected setup. In the future we would like to explore more mobility models and see the usability of the proposed model in a highly cluttered environment with a swarm of units.

**REFERENCES**


