# Robotic Routers

Onur Tekdas and Volkan Isler
Department of Computer Science, Rensselaer Polytechnic Institute
Technical Report # 08-01
{tekdao,isler}@cs.rpi.edu

***Abstract*— Mobile robots equipped with wireless networking capabilities can act as robotic routers and provide network connectivity to mobile users. Robotic routers provide cost efficient solutions for deployment of a wireless network in a large environment with limited number of users.**

**In this paper, we present motion planning algorithms for robotic routers to maintain the connectivity of a single user to a base station. We consider two motion models for the user. In the first model, we assume that the target's motion is known in advance. In the second model, user moves in an adversarial fashion and tries to break the connectivity.**

## I. Introduction

Suppose a user, working in a large farm or a warehouse, needs continuous network connectivity, perhaps for inventory tracking or surveillance. The user can be a human carrying a mobile device or an autonomous robot. One approach to provide network connectivity to such a user is to deploy static wireless routers and provide connectivity over the entire environment. Even though this solution is viable in some cases, in general maintaining the network (e.g. deployment, replacing batteries) may be costly and inconvenient. Further, since at any given time the set of nodes needed to provide connectivity forms only a small subset of all nodes deployed in the environment, this solution may be too costly in large environments. We believe that robotic routers can provide a better solution for this application: a small number of robots can autonomously deploy themselves and reconfigure to maintain the connectivity of the user. Maintenance operations such as recharging can be performed in an autonomous fashion.

We have run into a similar problem in our laboratory which is located at the end of a long, rectangular building (See Figure 1). The Computer Science Department owns only a small portion of this floor and our wireless network covers only the part of the floor owned by the department. Therefore, when one of our robots navigates in the halls and needs wireless connection, using other robots to maintain connectivity becomes an appealing solution.

In order to demonstrate the potential utility of introducing mobility, consider the two examples shown in Figure 2. In both examples, the star, square and circle indicate the base station, user and robotic router, respectively. The connectivity model is visibility based (i.e. two nodes are connected if they see each other). In the left figure, roughly $|V|/3$ stationary routers are needed where $|V|$ is the cardinality of the set of vertices of the environment. However, a single robotic router which is as fast as the target, suffices for maintaining

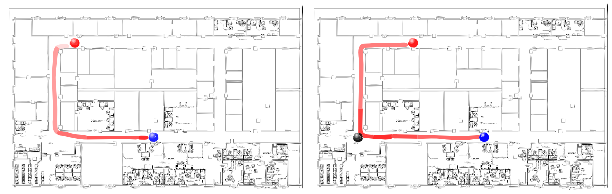

Fig. 1. **Left:** The blue circle (in the middle of the bottom corridor) is the stationary blue robot. Fading red circles show the signal strength as the red robot moves to the upper corridor (actual measurements). When the red robot reaches the position shown on the map, the signal strength becomes zero. **Right:** By moving a third robot (black circle) to the position shown on the map, we can reestablish the communication. The colors indicate the signal strength from black to red and black to blue nodes.

the target's connectivity. Therefore, in this environment, introducing mobility is beneficial. However, in the right figure, unless the routers are extremely fast, the number of robotic routers can be as high as the number of static routers.

In this paper, we study the problem of designing motion strategies for a network of robotic routers in order to maintain the connectivity of a single user to a base station. The paper is organized as follows. After an overview of related work, we formalize the *Robotic Routers Problem* in Section III. In Sections IV and Section V, we study the robotic routers problem for two different user mobility models. Finally, in Section VI we show the practical feasibility of our solutions with simulations.

## II. Related work

The problem of maintaining connectivity of a team of robots has been studied for various tasks including ren-

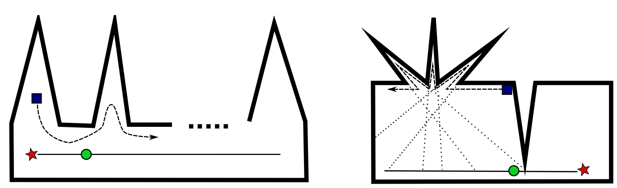

Fig. 2. In both figures, the star indicates the base station and we assume a visibility based connectivity model: two nodes can communicate if they can see each other. **Left:** We need one static router per triangular room for stationary case. However, a single robotic router (green circle) which is as fast as the user (blue square) can maintain the target's connectivity by following the drawn line. **Right:** As the user visits the spike-shaped rooms, a very fast router can maintain connectivity. However, if the router is not fast enough, as many as the number of routers used in stationary case may be needed.

dezvous [1], formation control [2] and flocking [3]. In these problems, all the network entries whose connectivity needs to be maintained can be controlled. The difference between these problems and the problem addressed in the present work is that, here we are maintaining the connectivity of a target moving independently from the network.

In this regard, Robotic Routers problem is related to the problem of maintaining the visibility of a single moving target. For this problem, dynamic programming [4], sampling-based [5], and game-theoretic [6] strategies have been presented. These problems focus on maintaining the visibility of the target until it disappears for the first time from the field of view of a single robot. In the problems we address, the connectivity model can be more general than visibility and there are additional constraints such as connectivity to the base station. Further, we address the issue of controlling multiple robots.

## III. Problem formulation

In this section, we first present the terminology and notation used throughout the paper, and formalize the robotic router problem.

### A. Definitions

A *robotic router* is a robot which has wireless communication capability. Robotic routers are subject to communication and motion constraints such as limited communication range and a bounded maximum speed. The *base station* is a static node to which the *user (or target)* wishes to establish connection. All entities move inside a shared *workspace* which we represent as a set of points. The single user, base station and a number of robotic routers establish a mobile ad-hoc network which we call *mobile router network*.

In addition to these entities, we use two concepts frequently: *connectivity* and *motion* model. In the mobile router network, two nodes are *connected* if they satisfy the given connectivity requirements which may depend on the position of nodes, communication range of nodes or possible occlusions. The *user is connected* if it is directly connected to the base station or it is connected through point-to-point links in the mobile router network. The *motion model* of the user is discussed in Section III-C.

### B. Notation and Assumptions

In this paper, we represent workspace $\mathcal{W}$ as a set of $n$ points. This set contains all possible locations of relevant entities (i.e. the user, the base station and the robotic routers). We represent the time domain in unit time steps. All motion models discussed in this paper are represented in discretized domain as a *trajectory*. Let $T$ be the end time of user motion, we express the trajectory of the user with the position vector $u$ of size $T$, i.e. $u(t)$ is the position of the user at time step $t$. Similarly, $r_i$ is the position vector of $i^{th}$ robotic router and $r_i(t)$ is the position of $i^{th}$ robotic router at time step $t$. Since the base station does not move, it's trajectory is a constant, $b$.

A configuration $q = (q_1, ..., q_m)$ is a vector of locations of robotic routers in the mobile router network. As we discretize the time domain, the speed of user and robotic routers are expressed as step sizes i.e. the distance that a node can move in one time step. In this model, we define the neighbor points that $i^{th}$ robotic router can move from the point $q_j$ as $N_r(q_j)$. To simplify the notation, we assume that all robotic routers have the same speed. However one can remove this assumption by defining different neighbor functions for each robotic router. For the user, we use the neighbor function $N_u$. Similarly, $N_c(q)$ is a set of neighbor configurations that can be reached from *configuration* $q$ in one time step. A trajectory $r_i$ is a *valid trajectory* if $\forall t,\ r_i(t+1) \in N_r(r_i(t))$.

Throughout the paper we assume that the connectivity model is given beforehand. In other words, we are given a matrix $A$ such that $A(i,j)$ is 1 if the mobile router network nodes located at $i$ and $j$ are connected and 0 otherwise. Here, we assume that the connection range of all nodes are same but it may depend on the communication capabilities of nodes. One can generalize this to varying ranges by using a matrix of higher dimensions. The user located on $q_u$ is *connected* by robots in configuration $q = (q_1, ..., q_m)$ if one of the following holds: $(i)$ $A(q_u, b) = 1$ $(ii)$ $\exists q_i$ s.t $A(q_u, q_i) = 1$ and $q_i$ is connected to $b$ through point-to-point link(s) of type $(i)$ or $(ii)$. Let $q(t)$ be the configuration of mobile router network at time step $t$, the user is *continuously connected* if it is *connected* in $q(t)$ for all $1 \leq t \leq T$.

### C. Motion model

In most applications, the workspace, location of the base station, wireless range and speed properties of robotic routers do not change. Hence, the trajectory of the user becomes the most important variable in determining the robotic router strategies. In this work we consider two motion models. In the first model we assume that we know the trajectory of the user in advance. This assumption can be made for some applications, e.g. in our experiments we control the (user) robots which may have fixed trajectories. In general, a user may be willing to declare its trajectory when requesting the connectivity service.

However, in some cases it is not feasible to know the user trajectory in advance. In such cases, we may consider the worst case trajectory where user tries to disconnect as quickly as possible. This case analysis can give us a guarantee on whether we can connect the user for any possible trajectory or not. We model this scenario as a *pursuer-evader game* where the user tries to break the connection from the mobile router network as quickly as possible. At the same time, the robotic routers try to extend the connection time as long as possible, preferably infinitely. We call this user motion strategy as *adversarial user trajectory* and the shortest such trajectory as the *shortest escape trajectory*.

### D. Formulation of the robotic routers problem

Here, we formalize the robotic routers problem for two motion models: known user trajectory and adversarial user trajectory.

**Known user trajectory:** Let $\mathcal{W}$ be the workspace, $A$ be the connectivity model, $b$ be the position of the base station, $m$ be the number of robotic routers and $u$ be the trajectory of the user. Find *valid robotic router trajectories* $r_i$, $\forall i$ such that the user is connected to the base station for the maximum possible amount of time.

**Adversarial user trajectory:** Let $\mathcal{W}$ be the workspace, $A$ be the connectivity model, $b$ be the position of the base station, $m$ be the number of robotic routers, $q_u$ be the initial location of the user and $q$ be the initial configuration of mobile router network. Find out whether there exists a user escape trajectory for *pursuer-evader game* and find the *shortest escape trajectory* $u$ and corresponding *valid robotic router trajectories*, if such trajectory exists.

Note that, in both problems we assume that the number of robotic routers $m$ is given. However, it is easy to obtain the minimum number of routers required for continuous connectivity simply by performing a binary search on $m$.

## IV. Known user trajectory

In this section, we present *KnownUserTrajectory* algorithm for robotic routers problem for the case where the user trajectory is known a priori. The solution uses dynamic programming to obtain robotic routers' trajectories. We build a table $C(q,t)$ which stores the maximum connection time of the user until time $t$ with routers ending in final configuration $q = (q_1, ..., q_m)$. Using $C(q,t)$ we find robot trajectories which maximize the connection time of the user. We define the $C(q,t)$ recursively as follows:

$$C(q,t) = \max_{q' \in N_c(q)} C(q', t-1) + d$$

$$\text{where } d = \begin{cases} 1 & \text{if } u(t) \text{ is connected by } q \\ 0 & \text{otherwise.} \end{cases}$$

$$C(q,1) = \begin{cases} 1 & \text{if } u(1) \text{ is connected by } q \\ 0 & \text{otherwise.} \end{cases}$$

The value: $\max_{\forall q} C(q,T)$ is the maximum connection time for the given user trajectory $u$. *KnownUserTrajectory* algorithm can be easily modified to return the corresponding robotic router trajectories $r_1, r_2, ..., r_n$ by backtracking. If $\max_{\forall q} C(q,T) = T$, we can find robotic router trajectories which continuously connect the user. Otherwise we find robotic router trajectories which maximize the connection time of the user.

The correctness and optimality of *KnownUserTrajectory* algorithm can be proved directly by induction. The running time of the algorithm is determined by the computation of $C(q,t)$. There exists $Tn^m$ tuples and for each tuples we determine the maximum in recurrence which takes $O(n^m)$ time. Hence the running time is $O(Tn^{2m})$.

## V. Adversarial user trajectory

In this section, we present *AdversarialUserTrajectory* algorithm for the robotic routers problem for the case where the user moves in such a way to break the connection as quickly as possible. The algorithm is as follows:

---

*Algorithm 1:* AdversarialUserTrajectory

1: $\forall q_u \forall q \;\; E[q_u, q] \leftarrow \infty$
2: $\forall q_u \forall q$
3: **if** $q_u$ is not connected in $q$ **then**
4: $\quad E[q_u, q] \leftarrow 1$
5: **end if**
6: **for** $k = 2$ to $n^{m+1}$ **do**
7: $\quad \forall q_u \forall q$
8: $\quad$ **if** $\min_{q'_u \in N_u(q_u)} \max_{q' \in N_c(q)} E[q'_u, q'] = k - 1$ **then**
9: $\quad\quad E[q_u, q] \leftarrow k$
10: $\quad$ **end if**
11: **end for**

---

First, we create a table of dimension $m+1$ and size $n^{m+1}$. In the first step (line 1), we initialize all the entries as infinity. At the end of the algorithm, the entry $E[q_u, q]$ gives the length of the shortest escape trajectory starting from user location $q_u$ and robotic routers' configuration $q$. We will show that if an entry $E[q_u, q]$ is left infinity at the end of the algorithm, then there exists no escape trajectory. In lines 2-5, we set $E[q_u, q] \leftarrow 1$ if $q_u$ is not connected in the initial configuration $q$. After the initialization steps, we repeat the procedure between lines 7-10 for $n^{m+1} - 1$ times. In this procedure, we apply the min-max relation (line 8) to the entire table. In each iteration $k$, we set $E[q_u, g]$ only if the shortest escape trajectory length is $k$.

With an additional step, we can find if there exists initial robotic router configuration $q$ corresponding to initial user location $q_u$ which satisfies the connectivity. If $\exists q, \; E[q_u, q] = \infty$, then we can initialize our robotic routers to configuration $q$ to keep the connectivity. Moreover, if $\forall q_u \exists q, \; E[q_u, q] = \infty$, we can say that $m$ robotic routers are sufficient to maintain the connectivity independent from the initial location and the trajectory of the user.

The running time of the algorithm is $O(n^{3(m+1)})$: there exists $n^{m+1}$ entries and for each iteration we scan the entire table which takes $O(n^{m+1})$ time. We iterate this procedure for $n^{m+1}$ times which sums up to the claimed running time. In Theorem 2 we show the correctness and optimality of the algorithm.

*Theorem 2:* Suppose there exists a *shortest escape trajectory* such that robotic routers are initially in configuration $q$ and user is at location $q_u$. Let $e(q_u, q)$ be the length of this trajectory.

1) $E[q_u, q] = k$ if and only if the length of the shortest escape trajectory $e(q_u, q)$ is $k$.
2) $E[q_u, q]$ is $\infty$ if and only if there exists robotic router trajectories for any possible user trajectory which satisfies the continuous connectivity.

*Proof:*

*Proof of (1):* We show that $E[q_u, q] = k \Leftrightarrow e(q_u, q) = k$ by induction on $k$.

Basis: $E[q_u, q] = 1 \Leftrightarrow e(q_u, q) = 1$ holds due to the initialization step between lines 2-5. If the escape trajectory is of length 1, this means that the user is disconnected in the initial configuration and we set $E[q_u, q] \leftarrow 1$. Similarly, if $E[q_u, q]$ is set to one in the initialization step, there exists a trivial escape trajectory of length 1.

Inductive step: let us assume that $\forall k,\ E[q_u, q] = k \Leftrightarrow e(q_u, q) = k$ holds. We show that $E[q_u, q] = k + 1 \Leftrightarrow e(q_u, q) = k + 1$. We prove this statement by showing that both directions of the conditional statement hold.

First we prove: $E[q_u, q] = k + 1 \Rightarrow e(q_u, q) = k + 1$. For contradiction, suppose that $E[q_u, q] = k + 1$ but $e(q_u, q) \neq k + 1$. Due to the inductive step we have: $e(q_u, q) \geq k + 1$ (Condition 1). This is because, due to the inductive hypothesis, $e(q_u, q) < k + 1$ would imply $E[q_u, q] < k+1$, which is a contradiction. When $E[q_u, q]$ is set to $k + 1$, due to the min-max relation, following holds: $\exists q'_u \in N_u(q_u),\ \exists q' \in N_c(q)$ such that $E[q'_u, q'] = k$ and $\forall q'' \in N_c(q),\ E[q'_u, q''] \leq k$. From the inductive hypothesis and the inequality $\forall q'' \in N_c(q),\ E[q'_u, q''] \leq k$, we have $\forall q'' \in N_c(q),\ e(q'_u, q'') \leq k$. This gives us $e(q_u, q) \leq k + 1$ (Condition 2). This is because the user can choose to go to $q'_u$ and follow an escape trajectory of length $k$ afterwards. From conditions (1) and (2), we have $e(q_u, q) = k+1$ which contradicts with the original claim. Thus, $E[q_u, q] = k+1 \Rightarrow e(q_u, q) = k + 1$ holds (Condition 3).

Next, we prove: $e(q_u, q) = k + 1 \Rightarrow E[q_u, q] = k + 1$. Again, for contradiction, let us assume that $e(q_u, q) = k+1$ but $E[q_u, q] \neq k + 1$. From the inductive hypothesis, $E[q_u, q] \geq k + 1$ holds (Condition 4). Let $\pi$ be an escape trajectory of length $e(q_u, q) = k+1$ with initial positions of the players given by $q_u$ and $q$. Let $q'_u \in N_u(q_u)$ be the user location in the second step of $\pi$. Since, the escape trajectory length is exactly $k + 1$, $\forall q'' \in N_c(q),\ e(q'_u, q'') \leq k$. Because, otherwise robotic routers network can increase the connection time by going to $q'$ where $e(q'_u, q') > k$. Moreover, $\exists q' \in N_c(q)$, such that $e(q'_u, q')$ is exactly $k$ (otherwise by going $q'_u$, user achieves an escape trajectory of length less than $k + 1$ which is a contradiction). By the induction hypothesis: $\forall q'' \in N_c(q),\ E[q'_u, q''] \leq k$, thus applying the min-max relation yields $E[q_u, q] \leq k + 1$ (Condition 5). From conditions (4) and (5), we have $E[q_u, q] = k + 1$. This is a contradiction with the original claim. Therefore $e(q_u, q) = k + 1 \Rightarrow E[q_u, q] = k + 1$ holds (Condition 6).

From conditions (5) and (6), the inductive step is proved. Finally, we showed: $\forall k\ E[q_u, q] = k \Leftrightarrow e(q_u, q) = k$.

*Proof of (2)*:

The proof of the second statement is straightforward. $E[q_u, q]$ is either marked as $k \leq n^{m+1}$ or $\infty$ and the user either has a shortest escape trajectory of length $e(q_u, q) \leq n^{m+1}$ or it can not avoid the connection from the robotic router trajectory. Since the number of iterations in algorithm can not exceed $n^{m+1}$ the claim above holds for $E[q_u, q]$. Let us assume that there exists an escape trajectory and its length is: $e(q_u, q) > n^{m+1}$. Since the number of permutations of tuples: $(q_u, q)$ is $n^{m+1}$, we can find a cycle in the sequence of tuples. However, then we can find a shorter escape trajectory by avoiding the cycle which is a contradiction. ■

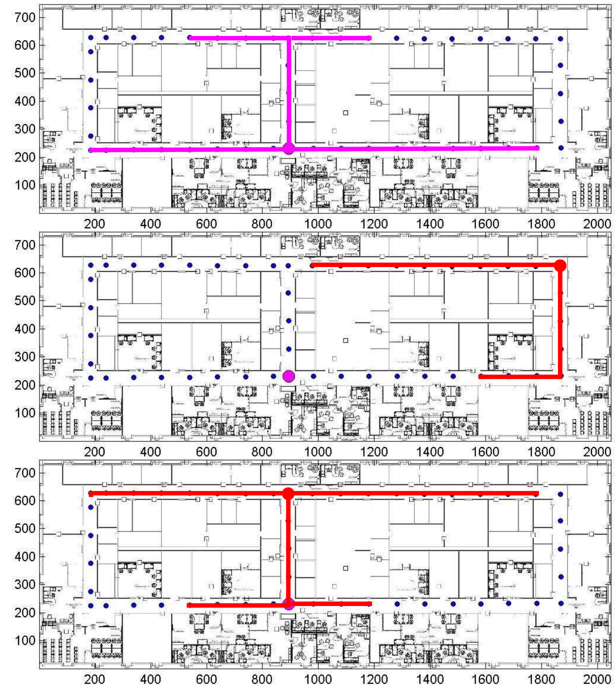


Fig. 3. Top figure shows the connectivity from the base station's location. Other two figures illustrate the connectivity of locations marked with red circles.

## VI. SIMULATIONS

We demonstrate a practical application of mobile router networks with simulations where the environment is our floor layout. We discretize the hallways into discrete locations almost uniformly (some degeneracy exists near the corners of halls). We construct the connectivity table according to the following rule: if the distance between two locations is less than a fixed distance $\tau$, and they are on the same hallway, then these two locations are connected. If two locations are not in the same hallway, their connectivity is based on the Manhattan distance between them. To obtain the connectivity threshold, we subtract a fixed penalty from $\tau$ for each corner on the path between the two locations. See Figure 3 for examples. In the following simulations, the robotic routers are twice as fast as the user. The base station is located at the bottom of the middle vertical hallway.

In order to obtain a baseline, we computed (by enumeration) the minimum number of static routers to cover the environment. It turns out that at least 4 static routers, as shown in Figure 4, are necessary to satisfy coverage and connectivity constraints.

In the following simulations, we start with a network of a single robotic router. For a given (known) user trajectory, we compute the corresponding robot trajectory which keeps the user connected during its trajectory. Next, we find an escape trajectory in which a single robotic router is not

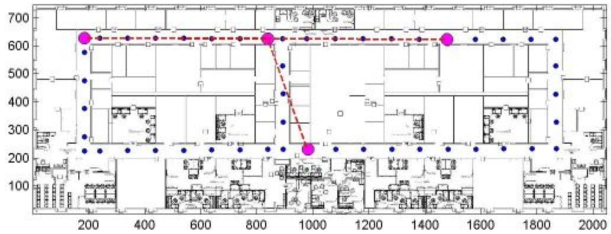


Fig. 4. The minimum number of static routers to satisfy the connectivity and coverage constraints is 4. The optimum deployment and its network topology is shown.

sufficient to maintain the connectivity. Finally, we show that two robotic routers are sufficient to keep the user connected whatever initial location or trajectory he chooses. We show how two robotic routers keep the user connected even if the user tries to avoid the connection. Videos of all simulations are available online [7].

Figure 5 shows the result of our first simulation. The top two figures show the (known) user trajectory and the corresponding robot trajectory computed by our algorithm. We identify the locations of nodes at the critical time steps with time labels. Following figures show snapshots of the connectivity graph of active nodes at these critical time steps. By connectivity graph of active nodes, we indicate the connectivity links (edges) between base station and the user, and the active nodes (vertices) in this connection path.

Figure 6 shows our second simulation. In this simulation, other than the last turn, the user follows the same trajectory as the previous simulation (see top left figure). Until this last turn, the robotic router also follows the same trajectory as the previous simulation (see top right figure). However, in the last step, the robotic router can not keep the user connected. The last figure shows the snapshot at the disconnected state.

We find the minimum number of required robotic routers for all possible user trajectories by trying $AdversarialUserTrajectory$ algorithm with increasing number of robotic routers until there exist corresponding robotic router trajectories for all possible initial locations and trajectories of the user. For our floor, we found that 2 robotic routers are sufficient to supply a continuous connection.

In the last simulation, relying on the sufficiency of two robotic routers, we solve the known trajectory algorithm with two robotic routers for a path which is not feasible for single robotic router network. The top three figures in Figure 7 show the user trajectory and corresponding robotic router trajectories. The user starts from the top left corner of workspace and completes a cycle by crossing from the middle vertical hall and coming back to the top of the vertical hall. Subsequent figures show the locations and connectivity graph of the nodes as snapshots at critical time steps.

## VII. Conclusion

In this paper, we addressed the problem of planning the motion of a network of robotic routers. We studied two different user models. In the first model, the user's trajectory is known whereas in the second model the user moves in

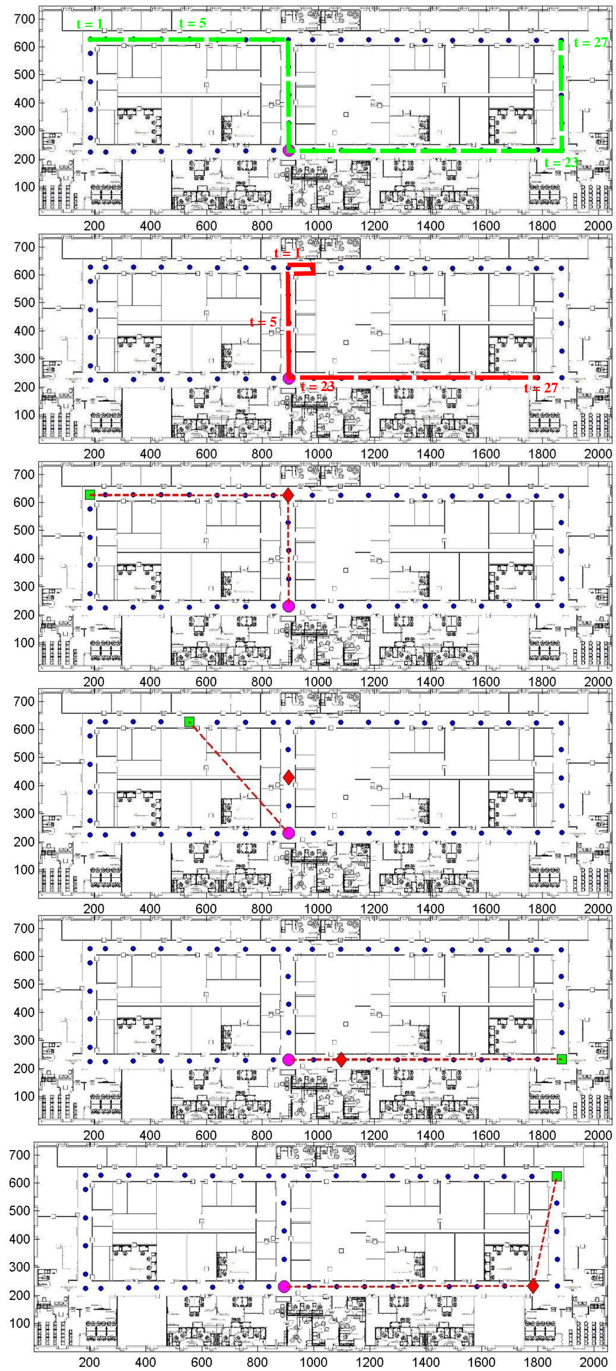


Fig. 5. The known user trajectory and corresponding computed robot trajectory are shown at the first and the second figure. The remaining figures show the snapshots of the user's connectivity graph (base station - circle with magenta color, robotic router - diamond with red color and user - square with green color). The third figure shows the initial configuration of nodes where the user is connected to the base station through the robotic router. Fourth figure shows the configuration at the time step when the user is directly connected to the base station. Fifth figure shows the configuration when the direct link between the user and base station is broken and connectivity is supplied through the robotic router. Last figure shows the final configuration of nodes.

an adversarial fashion. Even though the algorithms compute optimal solutions, their running times are exponential in the number of robots in the network. Currently, we are working on improving this running time.

## VIII. Acknowledgment

This work is supported in part by NSF CCF-0634823 and NSF CNS-0707939. The authors would like to thank Eric Meisner and Wei Yang for generating Figure 1.

## References

[1] A. Ganguli, J. Cortes, and F. Bullo, "Multirobot rendezvous with visibility sensors in nonconvex environments," *IEEE Transactions on Robotics*, Nov. 2006, to appear.

[2] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–639, 2004.

[3] M. M. Zavlanos and G. J. Pappas, "Flocking while preserving network connectivity," in *IEEE Conference on Decision and Contro*, 2007.

[4] S. M. LaValle, H. H. González-Baños, C. Becker, and J.-C. Latombe, "Motion strategies for maintaining visibility of a moving target," in *Proceedings IEEE International Conference on Robotics and Automation*, 1997, pp. 731–736.

[5] R. Murrieta-Cid, B. Tovar, and S. Hutchinson, "A sampling-based motion planning approach to maintain visibility of unpredictable targets." *Auton. Robots*, vol. 19, no. 3, pp. 285–300, 2005.

[6] S. Bhattacharya, S. Candido, and S. Hutchinson, "Motion strategies for surveillance," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[7] O. Tekdas, "Robotic routers network simulation videos." [Online]. Available: http://rsn.cs.rpi.edu/routers.html

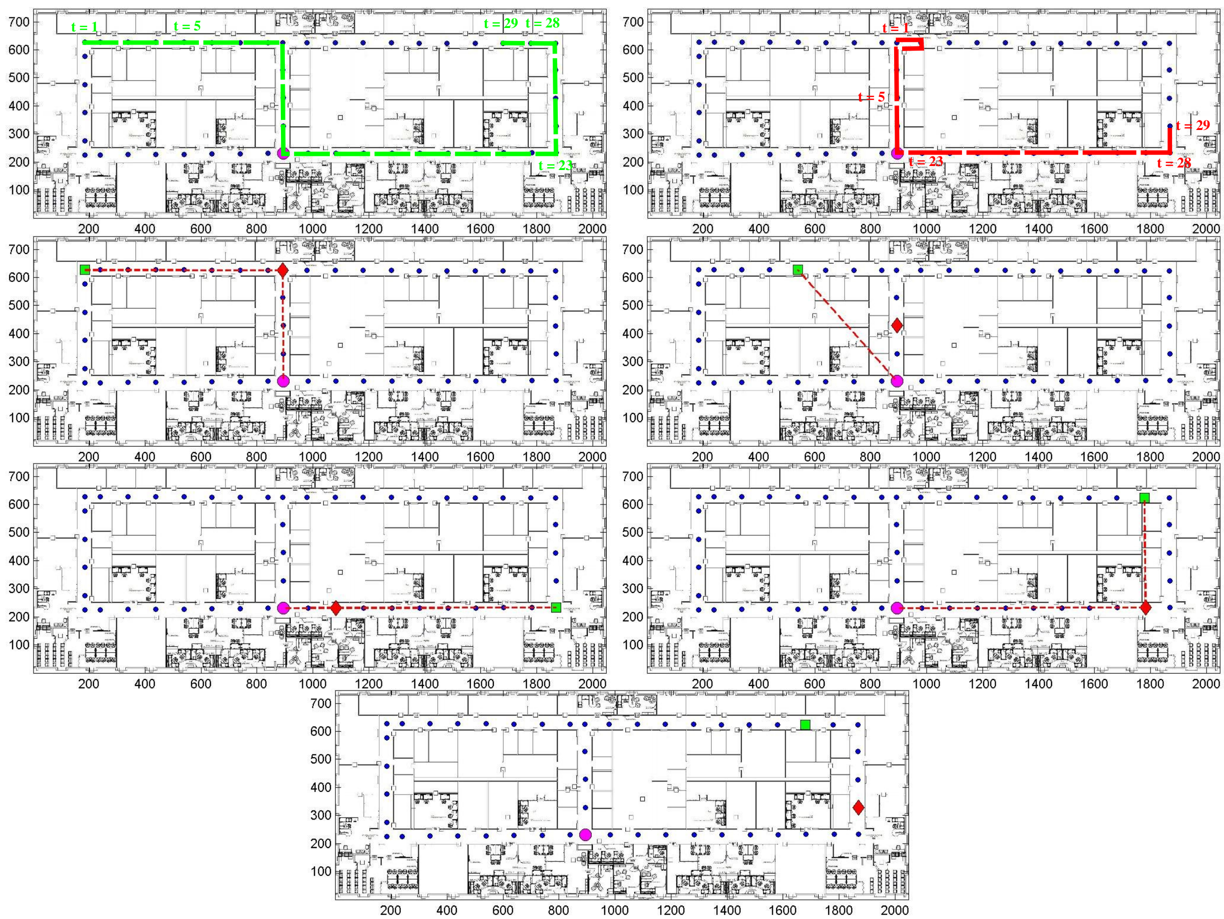


Fig. 6. Top two figures shows an escape trajectory for the user and the corresponding robot trajectory which maximizes the connection duration of the user. The remaining figures show the snapshots from the solution of algorithm including the connectivity graph of active nodes (base station - circle with magenta color, robotic router - diamond with red color and user - square with green color). First figure from the second row shows the initial configuration of nodes where user is connected to base station through the robotic router. Second figure from the second row shows the configuration at the time step when the user is directly connected to the base station. First figure from the third row shows the configuration when the direct link between the user and base station is broken and connection is supplied through the robotic router. Second figure from the third row shows the last configuration in which the user is connected to the network. Finally, bottom figure shows the configuration where the connection is broken inevitably.

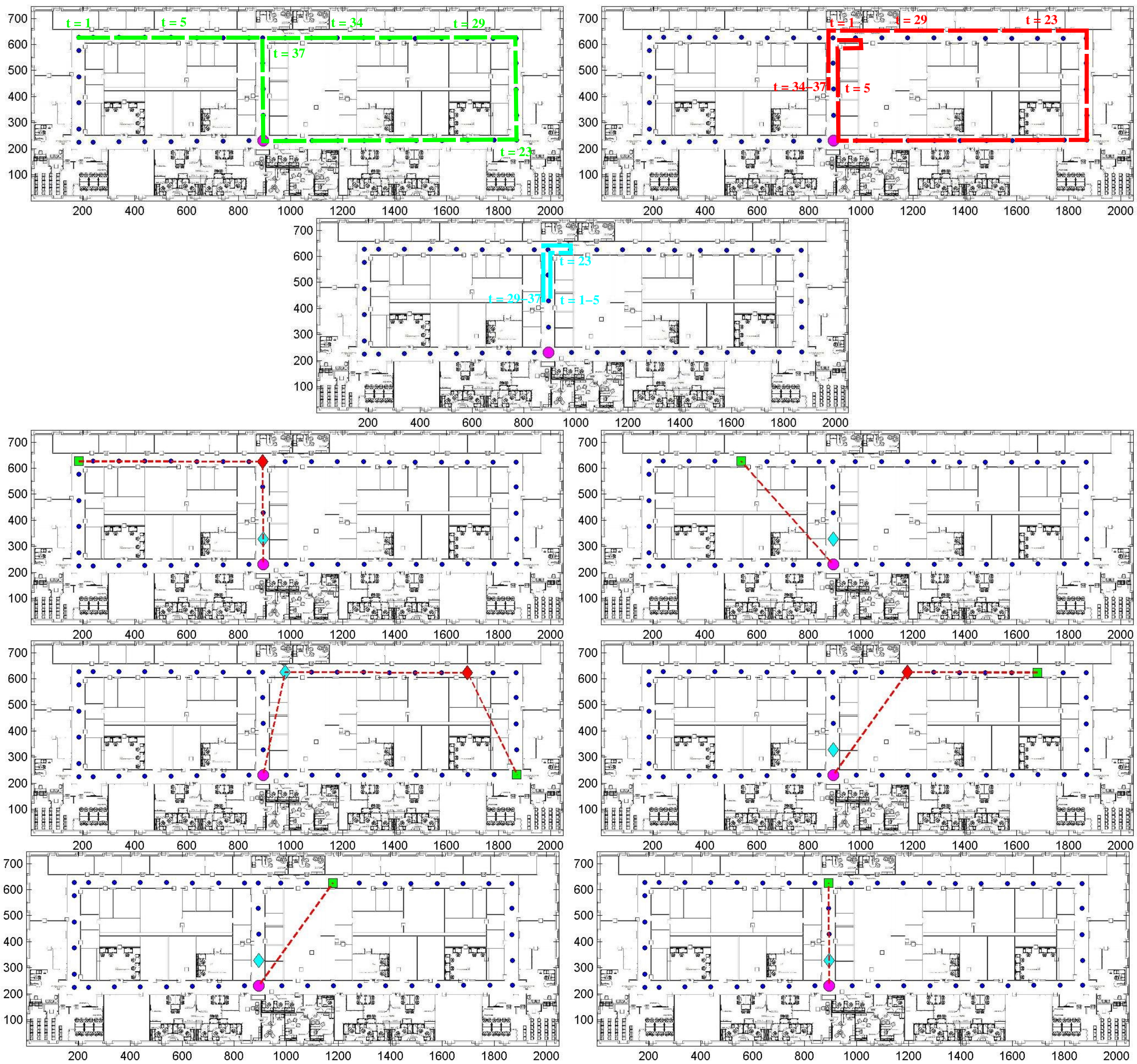


Fig. 7. Top three figures show the user trajectory and corresponding robot trajectories. Subsequent figures are snapshots from the solution of the algorithm including the connectivity graph of active nodes (base station - circle with magenta color, two robotic routers - diamond with red and cyan color, and user - square with green color). The first figure on the third row shows the initial configurations of all nodes where the user is connected to the base station through the red robotic router. The second figure on the third row shows the configuration at the time step when the user becomes directly connected to the base station. The first figure on the fourth row shows the configuration when the user is connected to base station through three links. The second figure on this row shows the configuration when the three-link connection reduced to a two-link connection. The first figure on the last row shows the time step when the user is directly connected to the base station. The last figure shows the final positions of the nodes at the last time step.