Robotic Routers

(joint work with Volkan Isler)

Onur Tekdas

Rensselaer Polytechnic Institute

May 22,2008

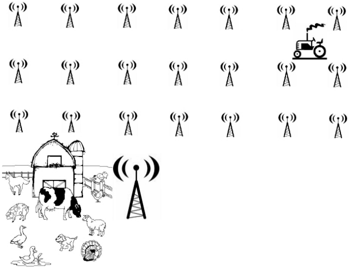
Table of contents

Motivation

- Imagine a user working in a large farm and needs network connection
- **Standard Solution:** A network of static wireless routers which covers the entire environment
    - A small subset of routers are **active** in a given time
    - This solution is **costly** for large environments

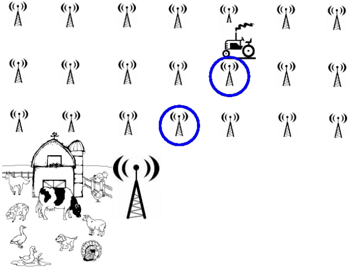
Motivation

- Imagine a user working in a large farm and needs network connection
- **Standard Solution:** A network of static wireless routers which covers the entire environment
  - A small subset of routers are **active** in a given time
  - This solution is **costly** for large environments



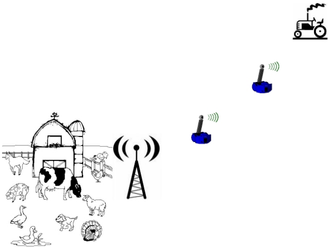
Motivation

- Imagine a user working in a large farm and needs network connection
- **Standard Solution:** A network of static wireless routers which covers the entire environment
  - A small subset of routers are **active** in a given time
  - This solution is **costly** for large environments
- **Alternative Solution:** A *small number* of robots with wireless communication capabilities can *autonomously* deploy themselves and reconfigure to maintain connectivity according to the location of the user



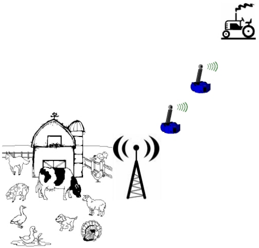
Motivation

- Imagine a user working in a large farm and needs network connection
- **Standard Solution:** A network of static wireless routers which covers the entire environment
  - A small subset of routers are **active** in a given time
  - This solution is **costly** for large environments
- **Alternative Solution:** A *small number* of robots with wireless communication capabilities can *autonomously* deploy themselves and reconfigure to maintain connectivity according to the location of the user



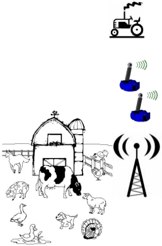
Motivation

- Imagine a user working in a large farm and needs network connection
- **Standard Solution:** A network of static wireless routers which covers the entire environment
  - A small subset of routers are **active** in a given time
  - This solution is **costly** for large environments
- **Alternative Solution:** A *small number* of robots with wireless communication capabilities can *autonomously* deploy themselves and reconfigure to maintain connectivity according to the location of the user



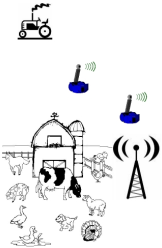
Motivation

- Imagine a user working in a large farm and needs network connection
- **Standard Solution:** A network of static wireless routers which covers the entire environment
  - A small subset of routers are **active** in a given time
  - This solution is **costly** for large environments
- **Alternative Solution:** A *small number* of robots with wireless communication capabilities can *autonomously* deploy themselves and reconfigure to maintain connectivity according to the location of the user

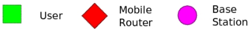


Motivation

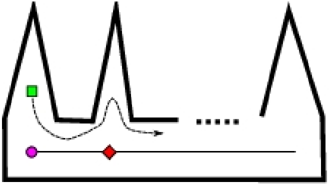
- Imagine a user working in a large farm and needs network connection
- **Standard Solution:** A network of static wireless routers which covers the entire environment
  - A small subset of routers are **active** in a given time
  - This solution is **costly** for large environments
- **Alternative Solution:** A *small number* of robots with wireless communication capabilities can *autonomously* deploy themselves and reconfigure to maintain connectivity according to the location of the user

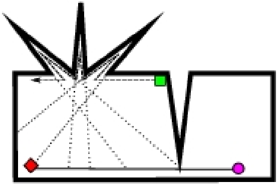


Is mobility useful?

- In this example, let us assume that the connectivity between two nodes is **visibility based**
- We use a **base station** as the access point of mobile router network to WAN
- In both figures: 

- In **left figure**, one mobile router is sufficient while $\lfloor \frac{n}{3} \rfloor$ static routers are necessary
- In **right figure**, if robot is not faster than user mobility may not gain any advantage





Problem parameters and definition

- The location of the base station
- The initial locations of robots
- The motion capabilities of robots (i.e. speed)
- The connectivity model between nodes
  - We present algorithms work for **any connectivity model**
  - The user is connected if it is connected to the base station through point-to-point links in the mobile router network
- The motion model of the user (i.e. speed and how it moves)

Problem definition

Given parameters above find robot strategies which **maximizes the connection time** of the user

Our contributions

In this work

- We focus on single user case
- We consider the motion models below and present **optimal solutions** for these models

- **Known Trajectory:**
  - The user trajectory is known in advance
  - For example, user can be a mobile robot which follows a predetermined trajectory
- **Adversarial Trajectory:**
  - The user tries to break the connectivity as quickly as possible
  - This ensures that whether the mobile router network can maintain the connectivity for any possible user trajectory

Our contributions

In this work

- We focus on single user case
- We consider the motion models below and present **optimal solutions** for these models

- **Known Trajectory:**
  - The user trajectory is known in advance
  - For example, user can be a mobile robot which follows a predetermined trajectory
- **Adversarial Trajectory:**
  - The user tries to break the connectivity as quickly as possible
  - This ensures that whether the mobile router network can maintain the connectivity for any possible user trajectory

Our contributions

In this work

- We focus on single user case
- We consider the motion models below and present **optimal solutions** for these models

- **Known Trajectory:**
  - The user trajectory is known in advance
  - For example, user can be a mobile robot which follows a predetermined trajectory
- **Adversarial Trajectory:**
  - The user tries to break the connectivity as quickly as possible
  - This ensures that whether the mobile router network can maintain the connectivity for any possible user trajectory

Working Example

- Our network only covers a small portion of floor hence the robotic router network is the only feasible solution

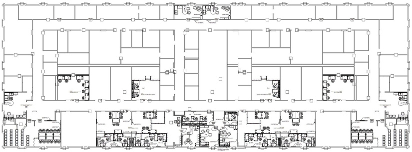


Figure: Floor plan

Connectivity model

- This figure shows the connection between two robots
- The fading red circle shows the connectivity strength (actual measurement)
- **Connectivity model** is based on geodesic distance but we penalize corners

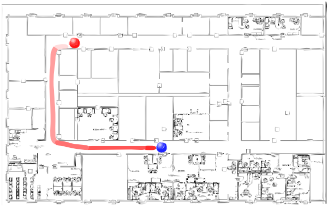
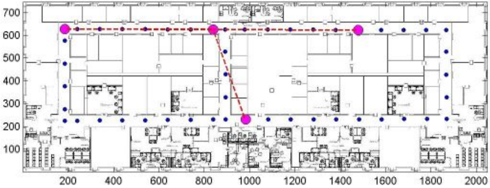


Figure: 1

Static router network

- We need **four** static routers for entire coverage (baseline)

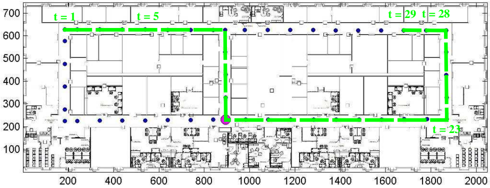


Adversarial user trajectory

- In the following figures:

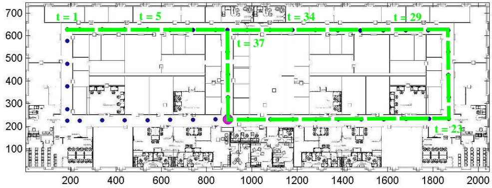
user | mobile routers | base station

- For the following user trajectory one mobile router is not sufficient
- User trajectory:



Adversarial user trajectory

- Using binary search on number of robots, we can find the minimum number of required robots to maintain the connectivity for adversarial user.
- In this particular problem, minimum number of required robotic router is **two**
- User trajectory:



Known user trajectory algorithm

- This is a dynamic programming algorithm
- The table is constructed using the C function below, where:
  - q and q' is the current and next location configuration of robots
  - t is the time step and $u(t)$ is the position of the user at t

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

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

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

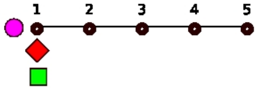
- $\max_{\forall q} C(q,T)$ is the maximum connection time (T is the end time of the user trajectory)

Adversarial user trajectory algorithm

Algorithm

AdversarialUserTrajectory

$\forall q_u \forall q \quad E[q_u, q] \leftarrow \infty$
$\forall q_u \forall q$
if q_u *is not connected in* q **then**
  $E[q_u, q] \leftarrow 1$
end if
for $k = 2$ *to* n^{m+1} **do**
  $\forall q_u \forall q$
  if $\min_{q'_u \in N_u(q_u)} \max_{q' \in N_c(q)} E[q'_u, q'] = k - 1$ **then**
    $E[q_u, q] \leftarrow k$
  end if
end for



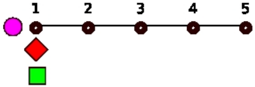
robot \ user	1	2	3	4	5
1	∞	∞	∞	∞	∞
2	∞	∞	∞	∞	∞
3	∞	∞	∞	∞	∞
4	∞	∞	∞	∞	∞
5	∞	∞	∞	∞	∞

Adversarial user trajectory algorithm

Algorithm

AdversarialUserTrajectory

$\forall q_u \forall q \;\; E[q_u, q] \leftarrow \infty$
$\forall q_u \forall q$
if q_u *is not connected in* q **then**
    $E[q_u, q] \leftarrow 1$
end if
for $k = 2$ *to* n^{m+1} **do**
    $\forall q_u \forall q$
    if $\min_{q'_u \in N_u(q_u)} \max_{q' \in N_c(q)} E[q'_u, q'] = k - 1$ **then**
        $E[q_u, q] \leftarrow k$
    end if
end for



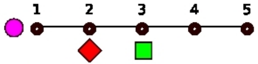
user / robot	1	2	3	4	5
1	∞	∞	1	1	1
2	∞	∞	∞	1	1
3	∞	∞	1	1	1
4	∞	∞	1	1	1
5	∞	∞	1	1	1

Adversarial user trajectory algorithm

Algorithm

AdversarialUserTrajectory

$\forall q_u \forall q \quad E[q_u, q] \leftarrow \infty$
$\forall q_u \forall q$
if q_u *is not connected in* q **then**
  $E[q_u, q] \leftarrow 1$
end if
for $k = 2$ *to* n^{m+1} **do**
  $\forall q_u \forall q$
  if $\min_{q'_u \in N_u(q_u)} \max_{q' \in N_c(q)} E[q'_u, q'] = k - 1$ **then**
    $E[q_u, q] \leftarrow k$
  end if
end for



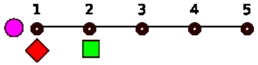
robot \ user	1	2	3	4	5
1	∞	∞	1	1	1
2	∞	∞	∞	1	1
3	∞	∞	1	1	1
4	∞	∞	1	1	1
5	∞	∞	1	1	1

Adversarial user trajectory algorithm

Algorithm

AdversarialUserTrajectory

$\forall q_u \forall q \;\; E[q_u, q] \leftarrow \infty$
$\forall q_u \forall q$
if q_u *is not connected in* q **then**
    $E[q_u, q] \leftarrow 1$
end if
for $k = 2$ *to* n^{m+1} **do**
    $\forall q_u \forall q$
    if $\min_{q'_u \in N_u(q_u)} \max_{q' \in N_c(q)} E[q'_u, q'] = k - 1$ **then**
        $E[q_u, q] \leftarrow k$
    end if
end for



robot \ user	1	2	3	4	5
1	∞	∞	1	1	1
2	∞	∞	∞	1	1
3	∞	∞	1	1	1
4	∞	∞	1	1	1
5	∞	∞	1	1	1

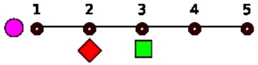
Infinity

Adversarial user trajectory algorithm

Algorithm

AdversarialUserTrajectory

$\forall q_u \forall q \quad E[q_u, q] \leftarrow \infty$
$\forall q_u \forall q$
if q_u *is not connected in* q **then**
  $E[q_u, q] \leftarrow 1$
end if
for $k = 2$ *to* n^{m+1} **do**
  $\forall q_u \forall q$
  if $\min_{q'_u \in N_u(q_u)} \max_{q' \in N_c(q)} E[q'_u, q'] = k - 1$ **then**
    $E[q_u, q] \leftarrow k$
  end if
end for



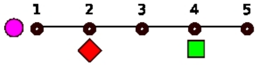
robot \ user	1	2	3	4	5
1	∞	∞	1	1	1
2	∞	∞	∞	1	1
3	∞	∞	1	1	1
4	∞	∞	1	1	1
5	∞	∞	1	1	1
		Infinity	Infinity		

Adversarial user trajectory algorithm

Algorithm

AdversarialUserTrajectory

1: $\forall q_u \forall q \quad E[q_u, q] \leftarrow \infty$
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**



robot \ user	1	2	3	4	5
1	∞	∞	1	1	1
2	∞	∞	∞	1	1
3	∞	∞	1	1	1
4	∞	∞	1	1	1
5	∞	∞	1	1	1

Infinity Infinity 2

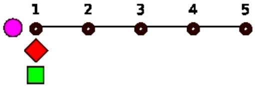
2

Adversarial user trajectory algorithm

Algorithm

AdversarialUserTrajectory

$\forall q_u \forall q \;\; E[q_u, q] \leftarrow \infty$
$\forall q_u \forall q$
if q_u *is not connected in* q **then**
    $E[q_u, q] \leftarrow 1$
end if
for $k = 2$ *to* n^{m+1} **do**
    $\forall q_u \forall q$
    if $\min_{q'_u \in N_u(q_u)} \max_{q' \in N_c(q)} E[q'_u, q'] = k - 1$ **then**
        $E[q_u, q] \leftarrow k$
    end if
end for



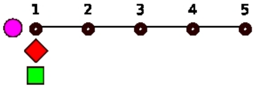
robot \ user	1	2	3	4	5
1	∞	∞	1	1	1
2	∞	∞	2	1	1
3	∞	∞	1	1	1
4	∞	∞	1	1	1
5	∞	∞	1	1	1

Adversarial user trajectory algorithm

Algorithm

AdversarialUserTrajectory

$\forall q_u \forall q \;\; E[q_u, q] \leftarrow \infty$
$\forall q_u \forall q$
if q_u *is not connected in* q **then**
  $E[q_u, q] \leftarrow 1$
end if
for $k = 2$ *to* n^{m+1} **do**
  $\forall q_u \forall q$
  if $\min_{q'_u \in N_u(q_u)} \max_{q' \in N_c(q)} E[q'_u, q'] = k - 1$ **then**
    $E[q_u, q] \leftarrow k$
  end if
end for



robot \ user	1	2	3	4	5
1	4	3	1	1	1
2	4	3	2	1	1
3	4	3	1	1	1
4	4	2	1	1	1
5	3	2	1	1	1

The correctness and optimality

Theorem

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 ∞ if and only if there exists robotic router trajectories for any possible user trajectory which satisfies the continuous connectivity.

Conclusion and future work

Conclusion:

- We consider the motion planning of network of robotic routers
- We presented two dynamic algorithms for **known user trajectory** and **adversarial user trajectory** model.
- Both algorithms are optimal
- The running time of algorithms are $n^{O(m)}$.

Future work:

- Finding efficient approximation algorithms

Acknowledgment

This work is supported in part by **NSF CCF-0634823** and **NSF CNS-0707939**

The authors thank to **Eric Meisner** and **Wei Yang** for generating Figure 2

Thanks for listening

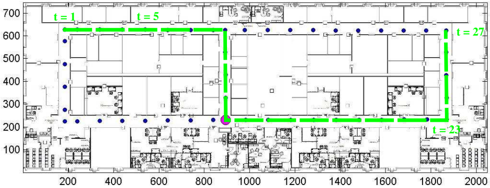
- Any questions or comments?

Robotic Routers Demo

Special thanks to **Wei Yang** for robotic routers implementation.

Known user trajectory

- With the given user trajectory **one** mobile router is sufficient
- User trajectory:



Sketch of the proof

- We show that $E[q_u, q] = k \Leftrightarrow e(q_u, q) = k$ by induction on k.
- Basis: $k = 1$ (trivial)
- Inductive step: Assume $E[q_u, q] = k \Leftrightarrow e(q_u, q) = k$ holds prove:
  1. $E[q_u, q] = k + 1 \Rightarrow e(q_u, q) = k + 1$
  2. $E[q_u, q] = k + 1 \Leftarrow e(q_u, q) = k + 1$

Sketch of the proof (cont.)

Proof ($E[q_u, q] = k + 1 \Rightarrow e(q_u, q) = k + 1$):
For contradiction, suppose that $E[q_u, q] = k + 1$ but $e(q_u, q) \neq k + 1$

1. $e(q_u, q) \geq k + 1$: From inductive step: If $e(q_u, q) < k + 1$ then $E[q_u, q] < k + 1$. Contradiction.
2. $e(q_u, q) \leq k + 1$: If $E[q_u, q] = k + 1$, due to the min-max relation:
   - $\exists q'_u \in N_u(q_u)$, $\exists q' \in N_c(q)$ such that $E[q'_u, q'] = k$
   - $\forall q'' \in N_c(q)$, $E[q'_u, q''] \leq k$

	q'_u	q_u	
q'	k	.	.
q	$\leq k$	$k + 1$	.
q''	$\leq k$	.	.

   - Hence,$\forall q'' \in N_c(q)$, $e(q'_u, q'') \leq k$
   - This gives us $e(q_u, q) \leq k + 1$