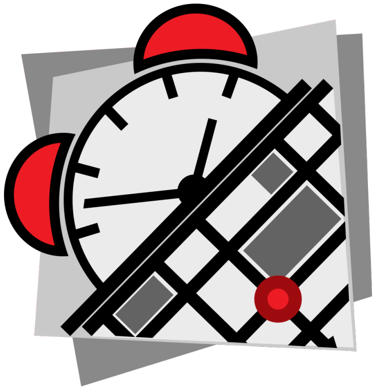
A Space Time Alarm



ROYAL INSTITUTE OF TECHNOLOGY

Universität Bremen

LBS2014, Vienna, 26th November 2014

Adrian C. Prelipcean, Takeshi Shirabe - Geoinformatics, KTH, Royal Institute of Technology
Falko Schmid - Cognitive Systems, Bremen Universität

Outline

Introduction

Research interest:

1. Traveler’s decision making processes and outcomes

2. Aiding travelers
   - Actively
   - Passively

Calm technology

… weaves itself into the fabric of everyday life until it sinks into the background (Weiser 1996).

Examples:

- Smart houses - adapt to the inhabitant’s needs
- Air conditioning - maintains constant temperature
- Street signs - absorb information without consciously reading

Is calm technology relevant to LBS? If so, how?

Here is one of our first attempts...

Space Time Alarm Clock

Given a destination and deadline, it:

Makes sure you arrive at the destination in time.

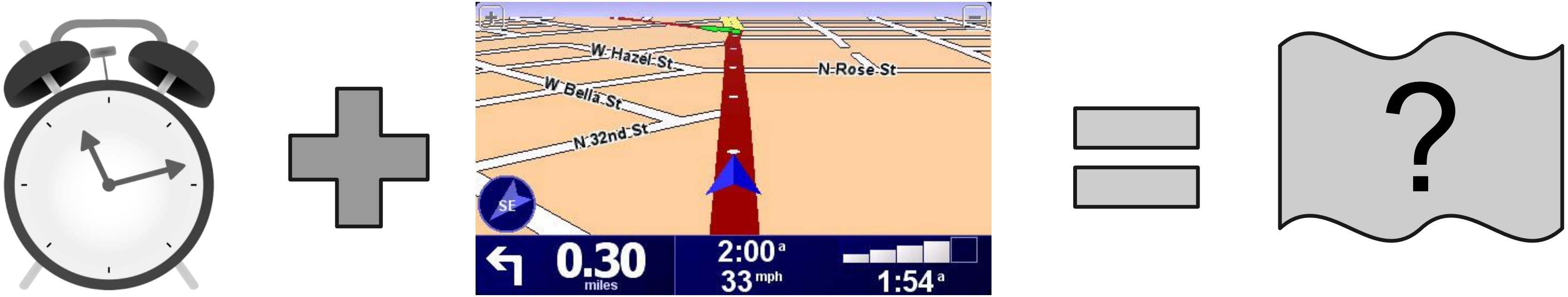
Tells you the consequence of each move.

- Forward – 3 minutes and 6 seconds to the destination
- Left – 6 minutes and 37 seconds to the destination
- Turn around – almost 10 minutes to the destination

Does NOT dictate what you should do.

Research Question

Is it possible to design a method that helps reach a specified destination by a specified deadline without demanding continuous **attention**?



source: http://linuxhub.net/wp-content/uploads/2010/01/alarm-clock.png and http://www.roadmapgps.com/models/tomtom-go-510/scr-navigation-map-6.jpg

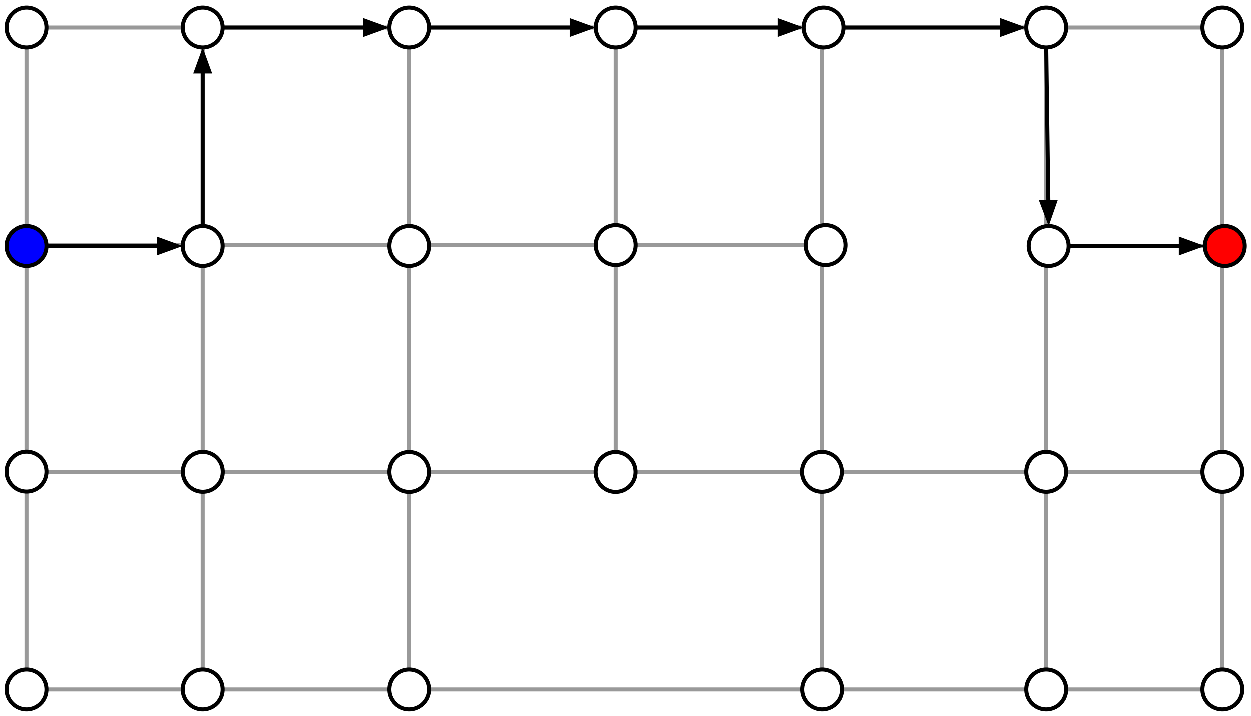
Space Time Alarm

Space Time Alarm is a component of the **Space Time Alarm Clock** that:

- continuously tracks the user in space and time,
- determines whether he will arrive at the destination in time, and
- alarms the user, if not.

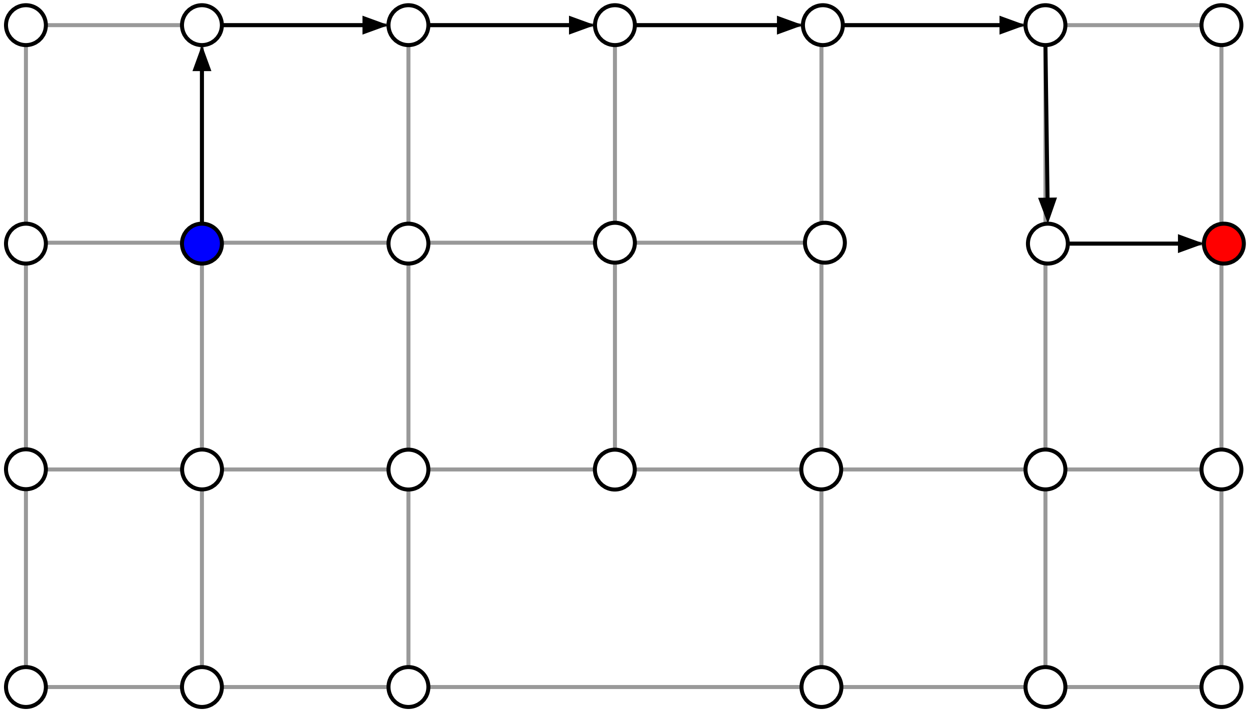
How could this be done with current technology?

Current technology (e.g. navigation systems)



1. destination specified
2. shortest path is computed
3. arrival time is estimated

Current technology (e.g. navigation systems)



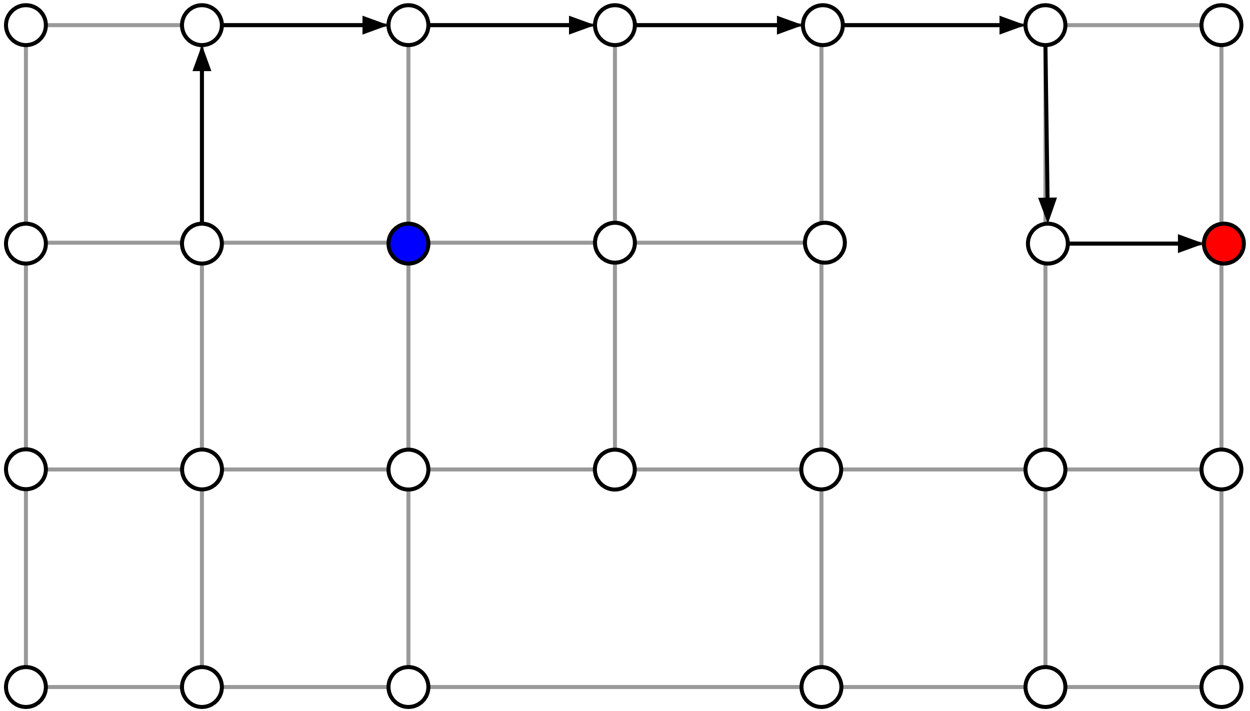
1. destination specified

2. shortest path is computed

3. arrival time is estimated

4. the user **follows** the shortest path

Current technology (e.g. navigation systems)



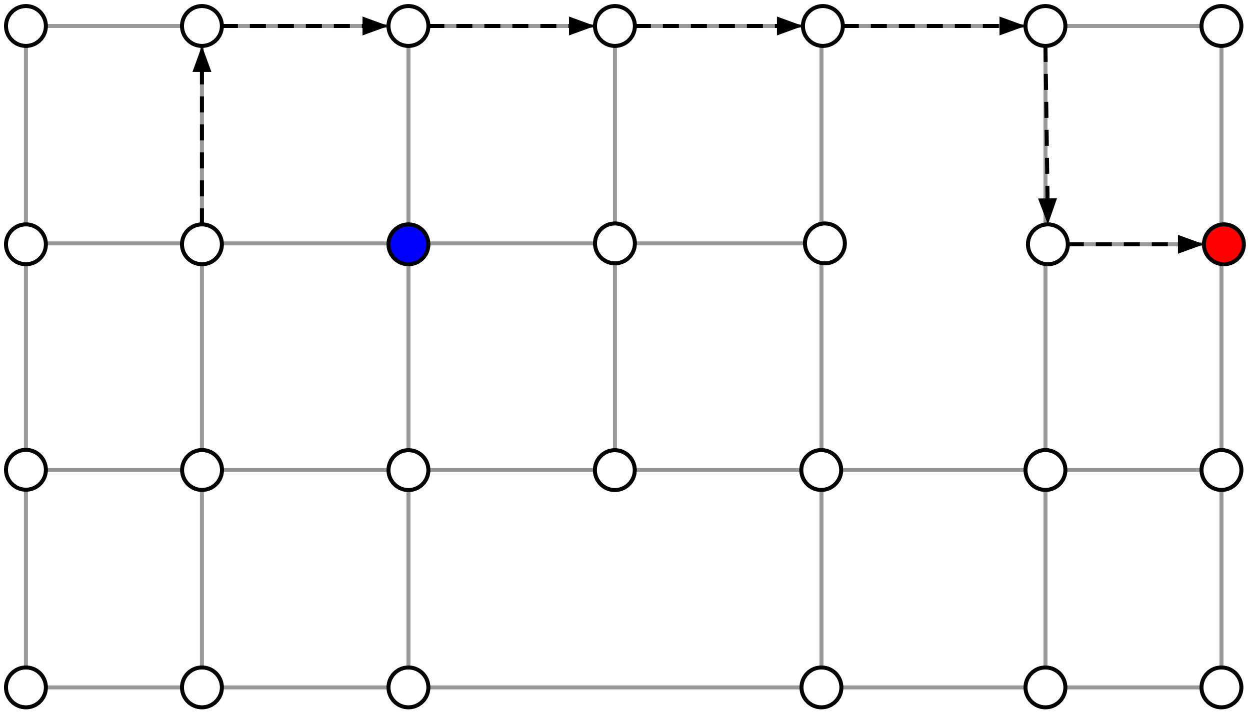
1. destination specified

2. shortest path is computed

3. arrival time is estimated

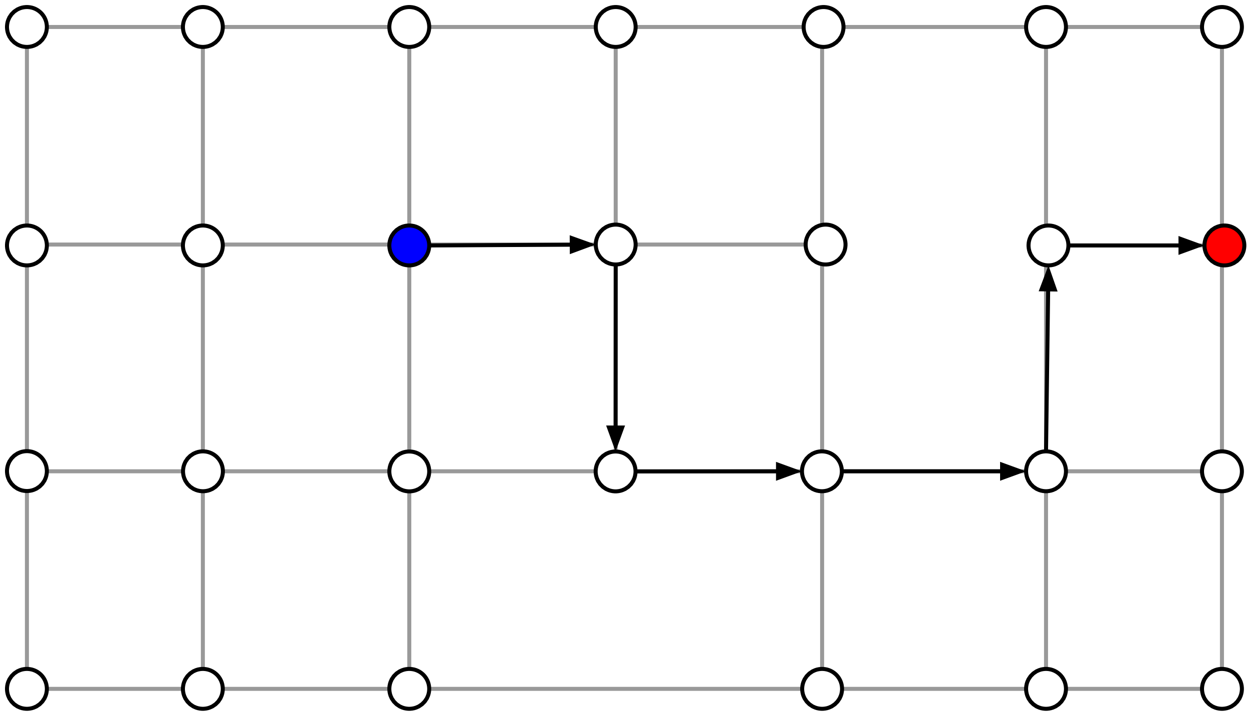
4. the user **does not follow** the shortest path

Current technology (e.g. navigation systems)



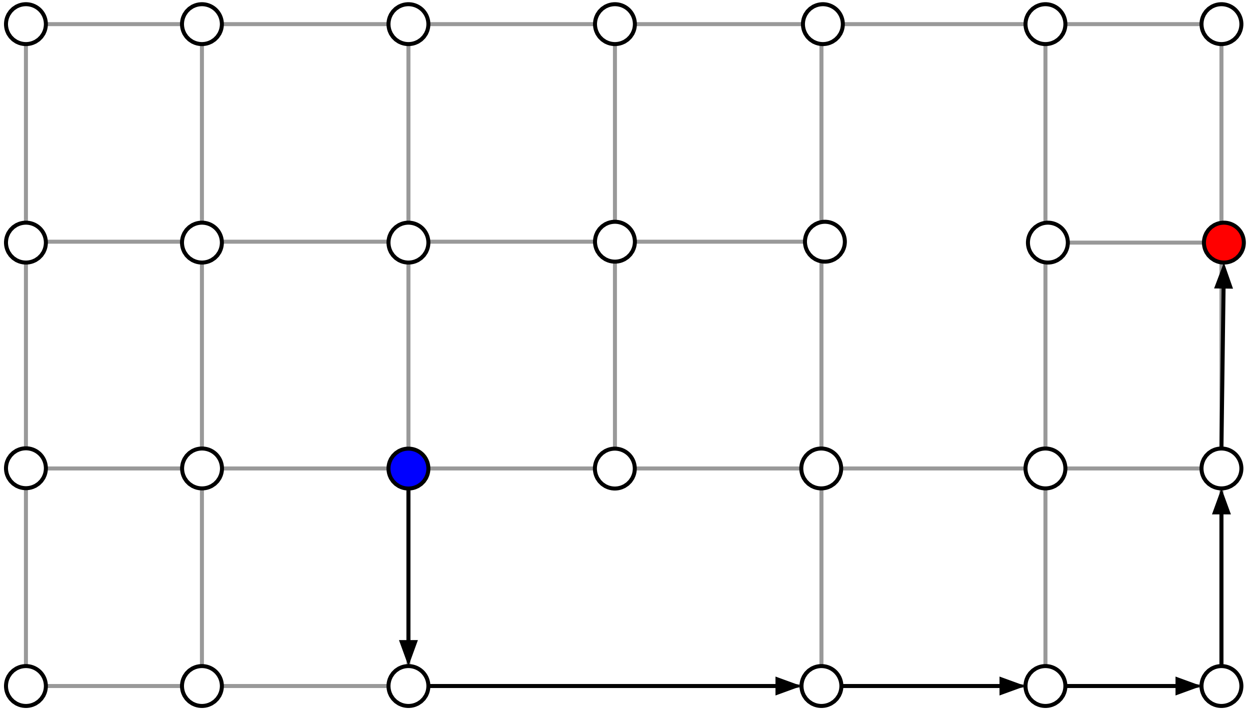
1. destination specified
2. shortest path is computed
3. arrival time is estimated
4. the user does not follow the shortest path
5. the shortest path becomes **obsolete**

Current technology (e.g. navigation systems)



1. destination specified
2. shortest path is computed
3. arrival time is estimated
4. the user does not follow the shortest path
5. the shortest path becomes **obsolete**
6. **new shortest path** is computed
7. arrival time is re-estimated

Current technology (e.g. navigation systems)



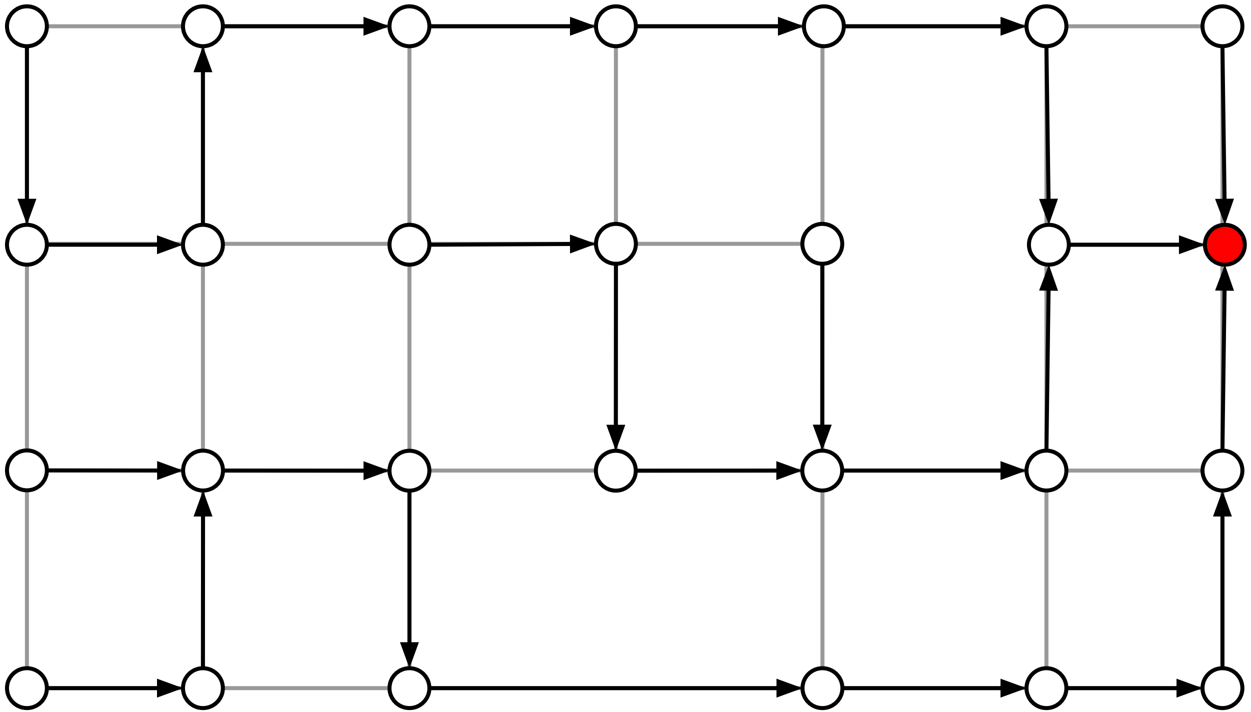
1. destination specified
2. shortest path is computed
3. arrival time is estimated
4. the user does not follow the shortest path
5. the shortest path becomes **obsolete**
6. **new shortest path** is computed
7. arrival time is re-estimated

Current technology (e.g. navigation systems)

Navigation systems may not be the right tools for the present purposes, because...

- The estimated arrival time is only valid when a user follows the shortest path to the destination,
- Any deviation from the shortest path triggers the re-computation of a new shortest path to the destination,
- When a new shortest path is generated, the arrival time is re-estimated, which implies…. when this information is available to the user, it might be already too late.

Using a shortest path tree



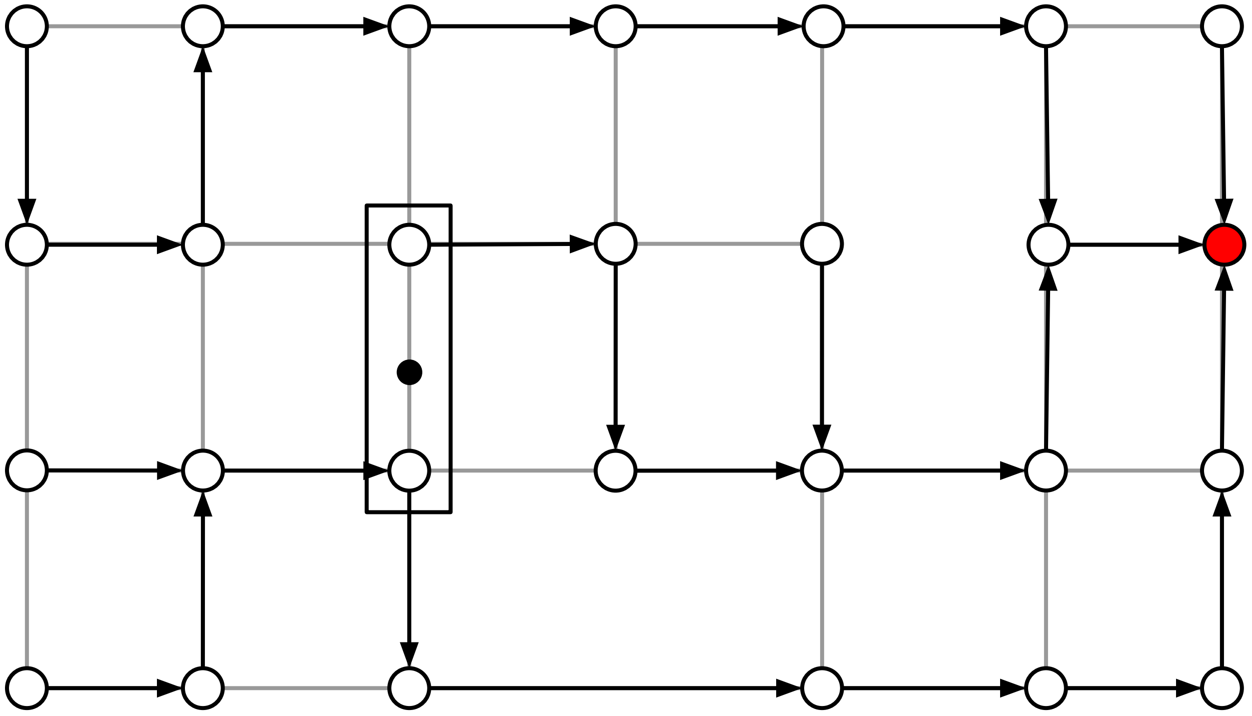
1. destination specified

2. shortest path tree is computed

3. travel time is estimated from **every node** in the network

4. shortest path tree needs **not to be recomputed** wherever the user goes

Using a shortest path tree



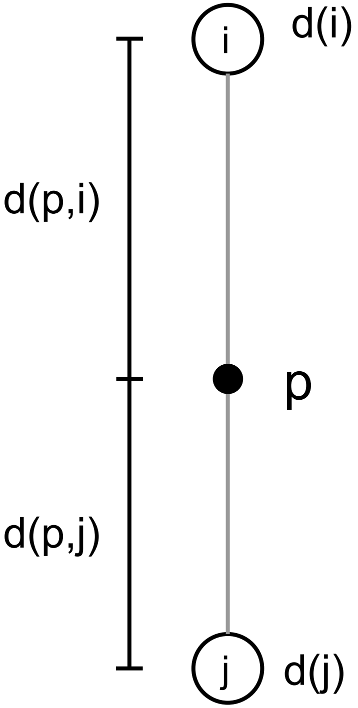
1. destination specified

2. shortest path tree is computed

3. travel time is estimated from **every node** in the network

4. shortest path tree needs **not to be recomputed** wherever the user goes

Estimating the arrival time



d(p) = min(d(i)+d(p,i), d(j)+d(p,j))

t(p) = current time + d(p)

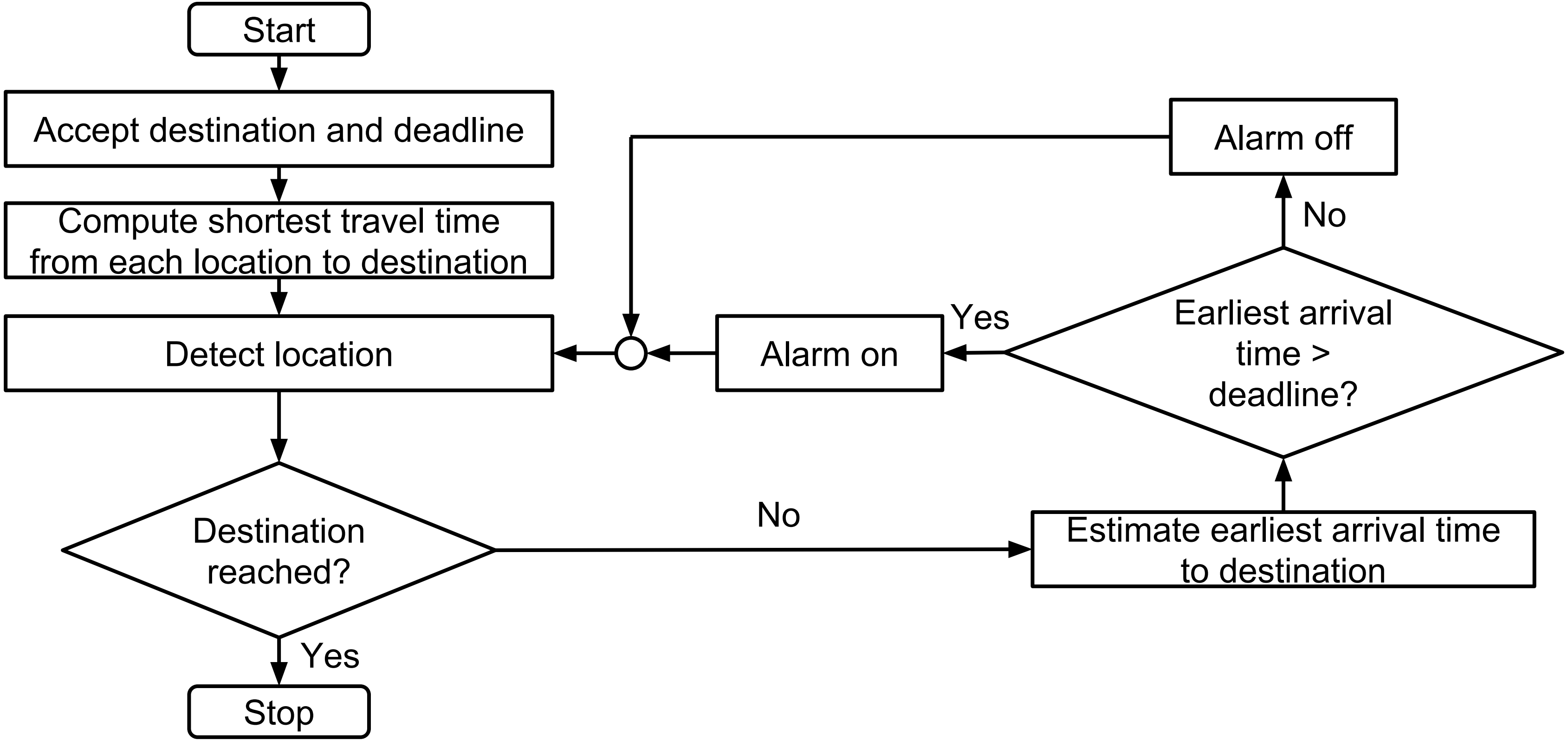
d(p) - travel time from “p” to destination

d(p,i) - travel time from “p” to node “i”

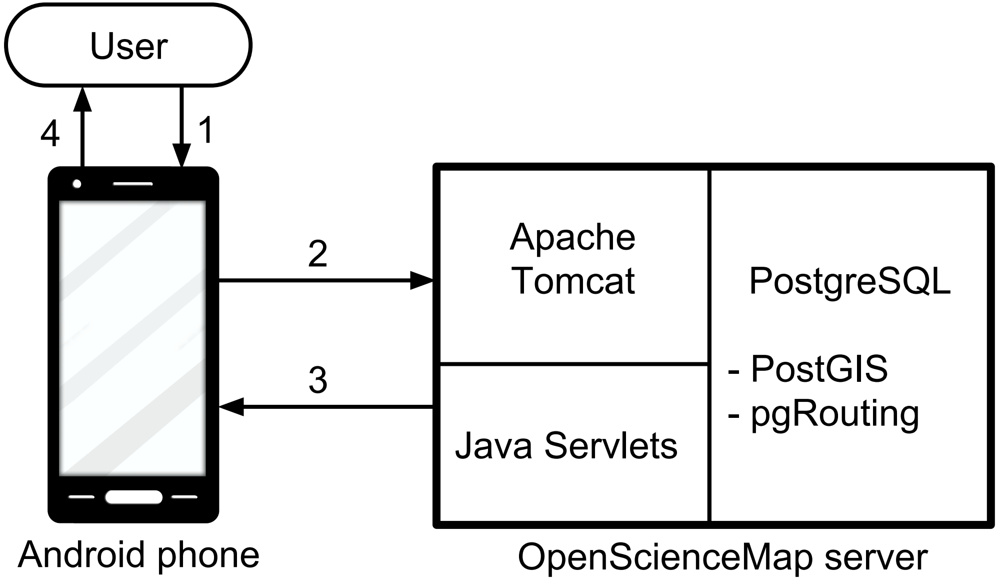
t(p) - estimated arrival time from position “p”

Users’ movement is confined to the road network.

The Space Time Alarm concept



A Space Time Alarm Implementation



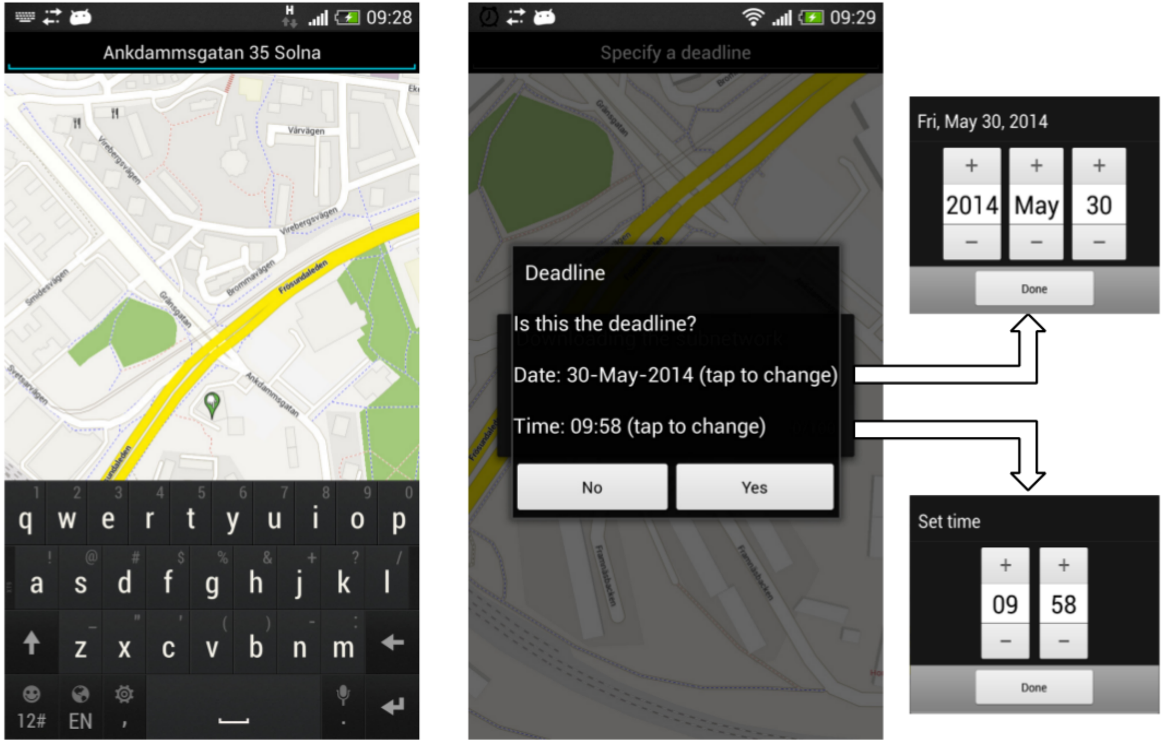
1 - User specifies **destination** and **deadline** (Input)

2 - Smartphone sends **current location**, **destination** and **deadline** to the server

3 - Server responds with the **shortest path tree for the subnetwork**

4 - Smartphone detects **user's location** and **alerts him when he is late**

Using Space Time Alarm



Difficulties of Space Time Alarm

- Every client has to store and access a shortest path **tree**, which causes bottlenecks
- The current implementation is limited to pedestrians, because other means of transportation would require more dynamic networks and the shortest path tree needs to be computed more than once.

Summary

- “Calm technology” can be useful (at least) for pedestrians.
- Introduced **Space Time Alarm**, which informs the user whether he will arrive at the destination in time, **wherever he goes, whenever it is**.
- The Shortest Path Trees allows the Space Time Alarm to provide its functionality
- Using Space Time Alarm with dynamic networks needs further investigation

Thank you for your attention!

Q&A / Discussion

Adrian C. Prelipcean <acpr@kth.se>
Takeshi Shirabe <shirabe@kth.se>
Falko Schmid <schmid@informatik.uni-bremen.de>