AARHUS UNIVERSITET

JUNE, 2009

EnTracked: Energy-Efficient Robust Position Tracking for Mobile Devices

Mikkel Baun Kjærgaard, Aarhus University
Jakob Langdal, Aarhus University
Torben Godsk, Aarhus University
Thomas Toftkjær, Aarhus University

AGENDA

- Motivation
  - Periodic tracking
  - Dynamic tracking
- EnTracked
- Methods
- Evaluation
- Conclusions

ACM MobiSys 2009

EnTracked: Energy-Efficient Robust Position Tracking for Mobile Devices

Mikkel Baun Kjærgaard, Jakob Langdal*, Torben Godsk†, Thomas Toftkjær‡
Department of Computer Science
Aarhus University, Denmark
mikkelbk@cs.au.dk, jakobl@cs.au.dk, tbg@cs.au.dk, toughcar@cs.au.dk

ABSTRACT

An important feature of a modern mobile device is that it can position itself. Not only for use on the device but also for remote applications that require tracking of the device. To be useful, such position tracking has to be energy-efficient to avoid having a major impact on the battery life of the mobile device. Furthermore, tracking has to robustly deliver position updates when faced with changing conditions such as delays due to positioning and communication, and changing positioning accuracy.

This work proposes EnTracked — a system that, based on the estimation and prediction of system conditions and mobility, schedules position updates to both minimize energy consumption and optimize robustness. The realized system tracks pedestrian targets equipped with GPS-enabled devices. The system is configurable to realize different trade-offs between energy consumption and robustness.

We provide extensive experimental results by profiling how devices consume power, by emulation on collected data and by validation in several real-world deployments. Results from this profiling show how a device consumes power while tracking its position. Results from the emulation indicate that the system can estimate and predict system conditions and mobility. Furthermore they provide evidence for that the system can lower the energy consumption considerably and remain robust when faced with changing system conditions. By validation in several real-world deployments we provide evidence that the real system works as predicted by the emulation.

Categories and Subject Description: C.2.4 [Computer-Communication Networks]: Distributed Systems

General Terms: Experimentation, Measurement

*Associated with the Alexandra Institute A/S, Denmark
†Associated with the Danish Agricultural Advisory Service, National Centre, Denmark
‡Associated with Systematic A/S, Denmark

1. INTRODUCTION

An important feature of a modern mobile device is that it can position itself. Not only for use locally on the device but also for remote applications that require tracking of the device. Examples of such applications are geo-based information applications [6] or proximity and separation detection for social networking applications [12] just to mention a few. To be useful, such position tracking has to be energy-efficient to avoid having a major impact on the power consumption of the mobile device. Optimizing the operation of mobile devices for energy efficiency is an important issue and research is trying to address it from many angles, for instance, by trying to lower the impact of network protocols on power consumption [15] or by optimizing the execution at the operating system level [4]. Furthermore, tracking has to be robust in order to deliver position updates within limits when faced with changing conditions such as delays due to positioning and communication, and changing positioning accuracy.

To quantify the impact of position tracking on power consumption, we measured the power consumption of a Nokia N95 phone for 30 minutes, while the phone periodically positioned itself using the built-in GPS receiver and then send the position data using UMTS to a remote service hosted on an internet-connected server. The measurements were repeated with different time intervals between the periodic position updates. The average power consumption measured is plotted in Figure 1. The results highlight that even for moderate time intervals between position updates such as sixty seconds the power consumption is as high as 0.6 watt, which is twelve times more consumed power than when the phone is idle and the double amount of power compared to when the phone is used with the screen turned on.

From Figure 1 one might propose to minimize the power consumption by using large intervals between position updates, but then maintaining position accuracy becomes a problem, as a pedestrian target can walk or run quite far during two to five minutes. To address this problem previous research such as [14, 19] has proposed dynamic tracking to try to minimize the frequency of needed position updates by only requiring updates after the target has moved more than a specified distance or has moved out of a specified area.

The systems proposed in previous research for such dynamic tracking have several drawbacks. First, the research assumes that power consumption for positioning and sending is instantaneous meaning that the power consumption per position sensing and sending is a constant and that you

Motivation

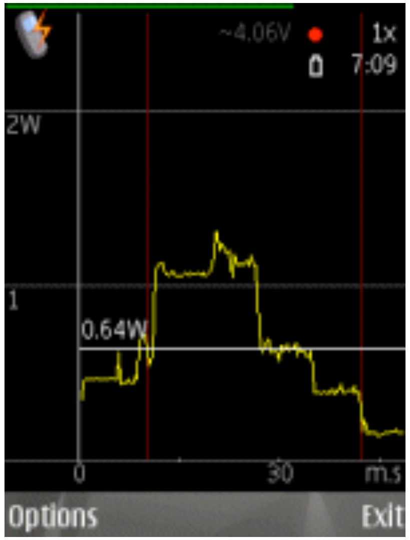
› Mobile devices depend on battery power

› Power consumption impacts application use

MEASURING POWER CONSUMPTION

› Profiling Power Consumption on N95 8GB

› Nokia Energy Profiler Tool



Feature	Avg. Power [watt]
Idle	0.0621
Accelerometer	0.0503
GPS	0.324
UMTS Radio idle	0.466
UMTS Radio active	0.645

POSITION TRACKING

Position-Based Application	Tracking Server	Positioned Target

- Tracking
  - For server to provide application with updates of the target position

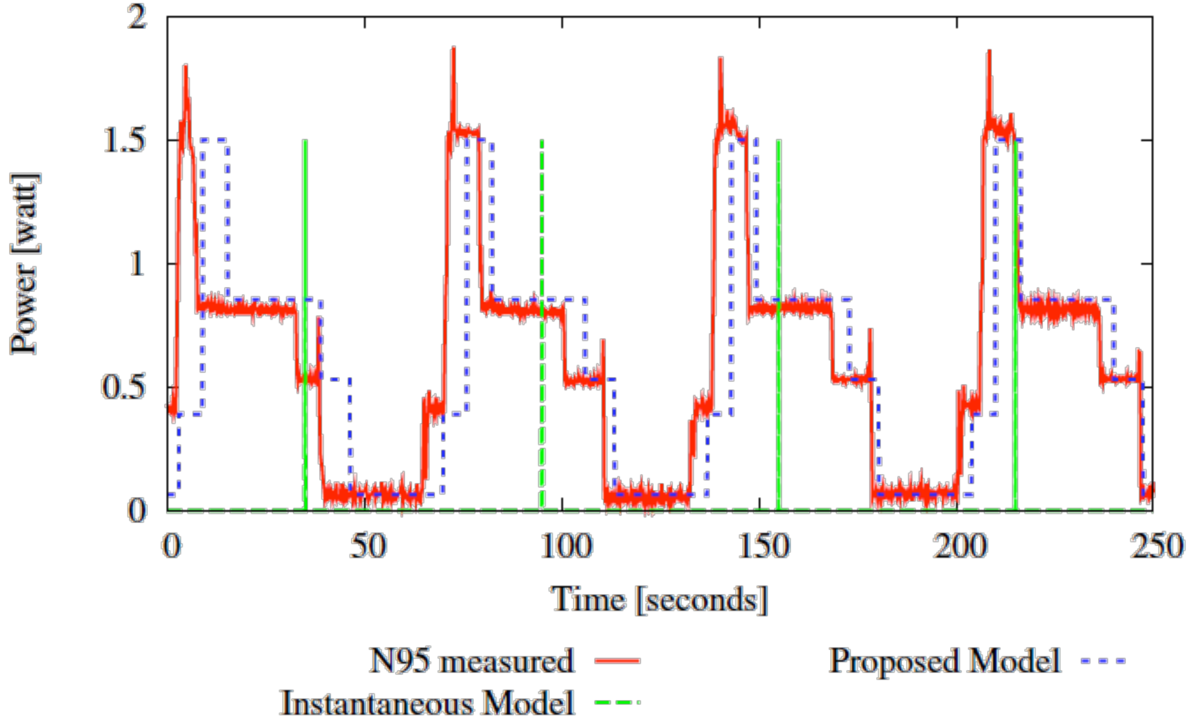
PERIODIC TRACKING

- Send position update every X seconds
- Upper limit on latency of position updates
- No guaranties for accuracy, walk 90 m or drive 1 km
- Simple to implement

A. Leonhardi and K. Rothermel. A comparison of protocols for updating location information.
Cluster Computing, 4(4):355–367, 2001.

PERIODIC TRACKING

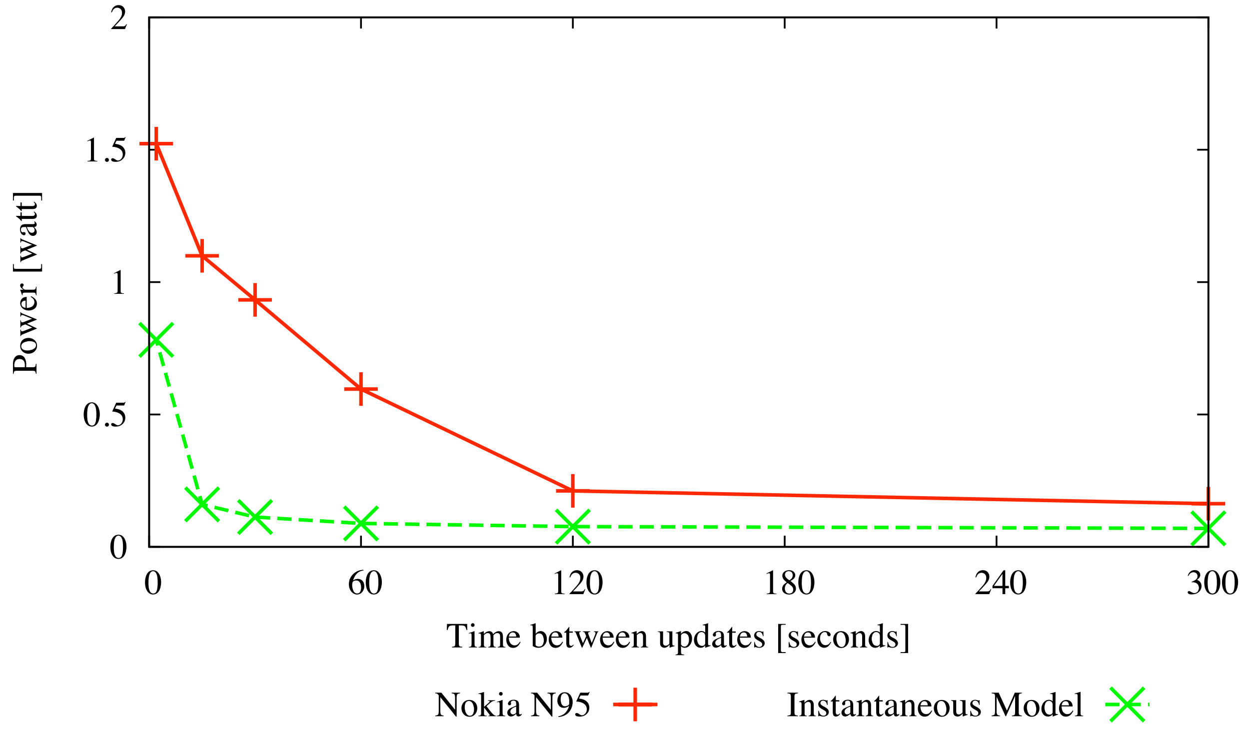
› X=60 seconds



T. Farrell, R. Lange, and K. Rothermel. Energy-efficient tracking of mobile objects with early distance-based reporting. In Proceedings of 4th Int. Conference on Mobile and Ubiquitous Systems (Mobiquitous 2007).

PERIODIC UPDATING

› Averages for 1-300 seconds



DYNAMIC TRACKING

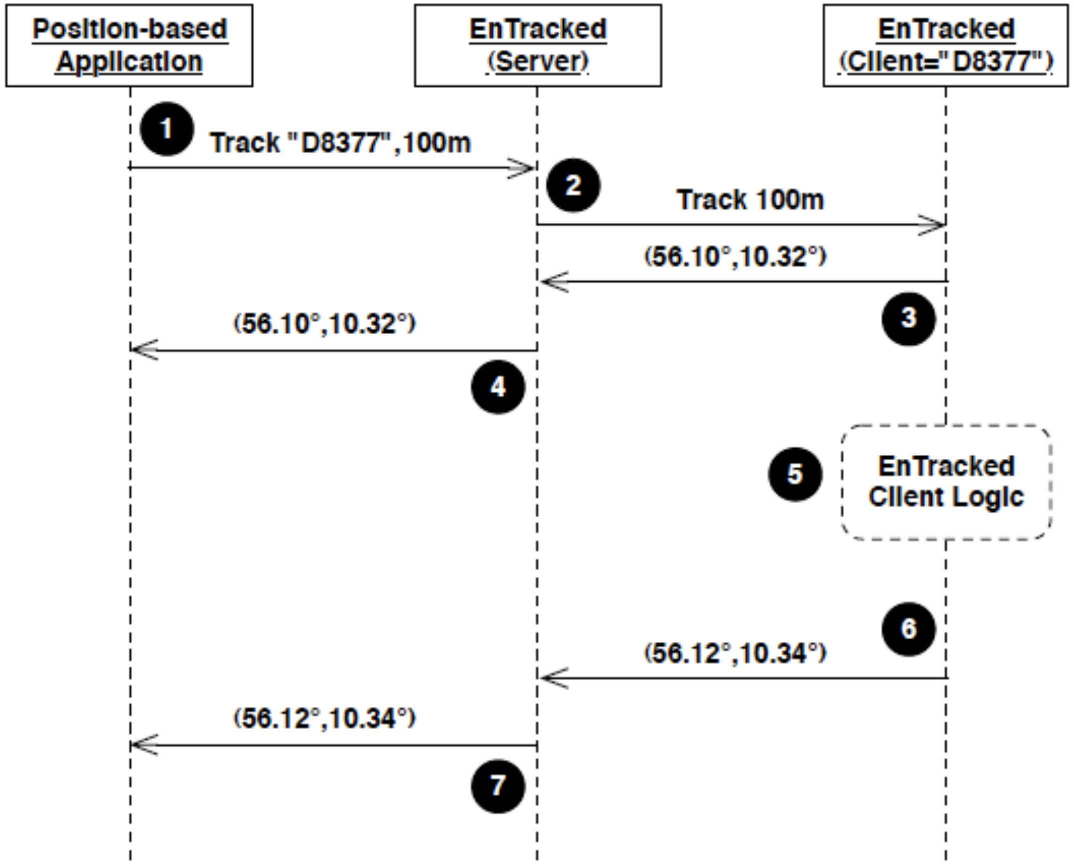
- Adapt position updates to satisfy error threshold
- Sources of thresholds
  - Map zoom level (50-500 m)
  - Proximity detection algorithms (1–10000 m)

    A. Küpper and G. Treu. Efficient proximity and separation detection among mobile targets for supporting location-based community services. Mobile Computing and Communications Review, 10(3):1–12, 2006.
- Complex implementation

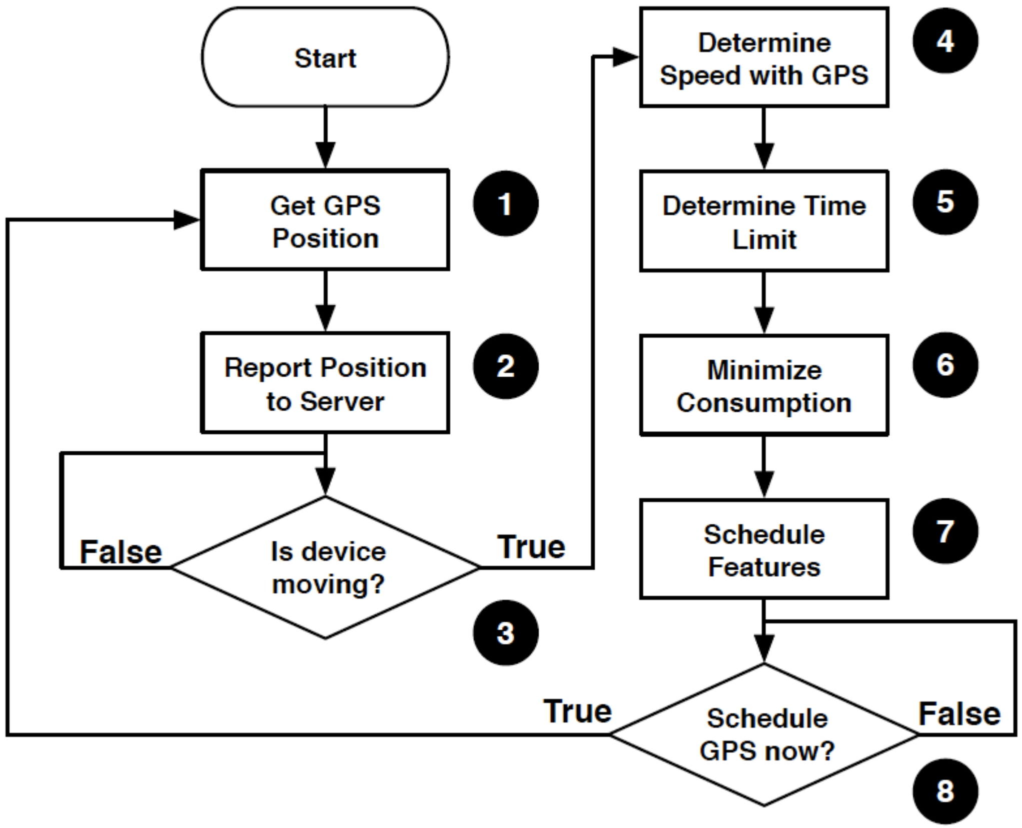
EnTracked

- EnTracked – a dynamic tracking system
- Goals
  - Minimize energy consumption (turning features off)
  - Optimize robustness (position within error thresholds)
- Adapt based on estimating and predicting system conditions and mobility

ARCHITECTURE



EnTracked – CLIENT SIDE LOGIC



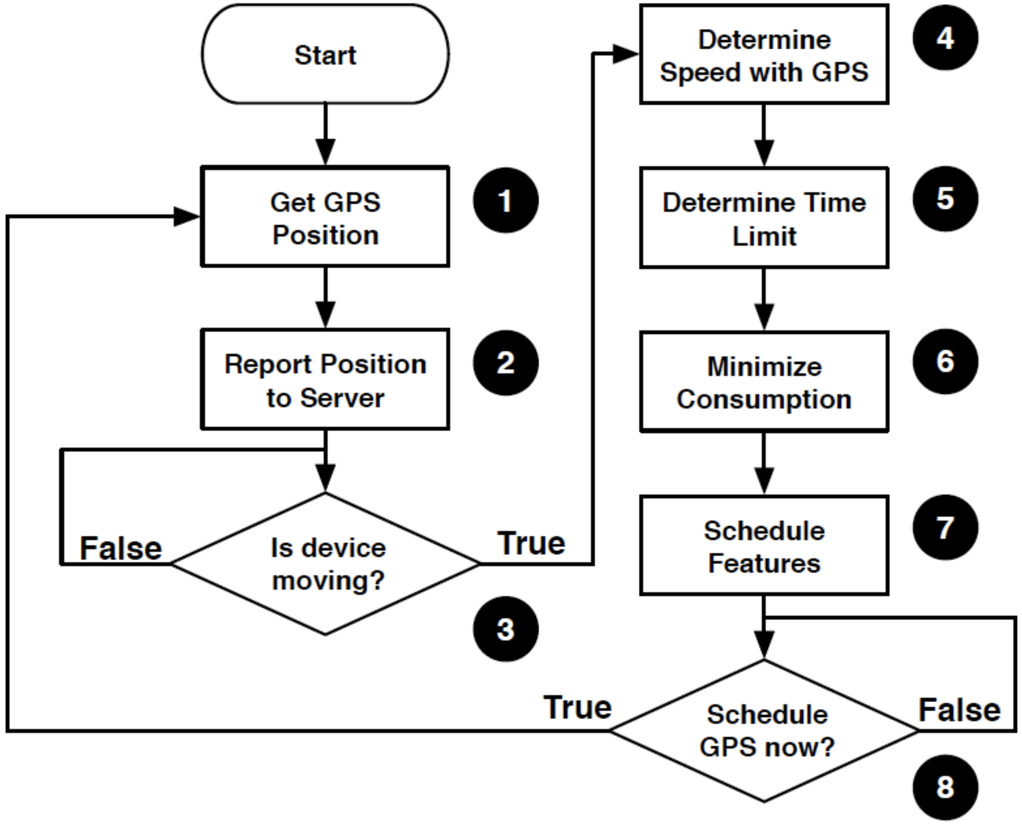
DERTEMINE TIME LIMIT

› Time limit for delivering a new fix to server

$$t_{limit} = \frac{d_{limit} - e_{model}}{v_{est}}$$

d_{limit}: user error limit
e_{model}: estimated error of server-known position
V_{est}: GPS speed estimate + speed estimate accuracy

EnTracked – CLIENT SIDE LOGIC



MINIMIZE CONSUMPTION

$$req_g(x) = \begin{cases} 1 & \text{if } x <= 30 \\ 6 & \text{if } x > 30 \end{cases}$$

Delay functions

$$req_s(x) = 1$$

Recursive Power Function

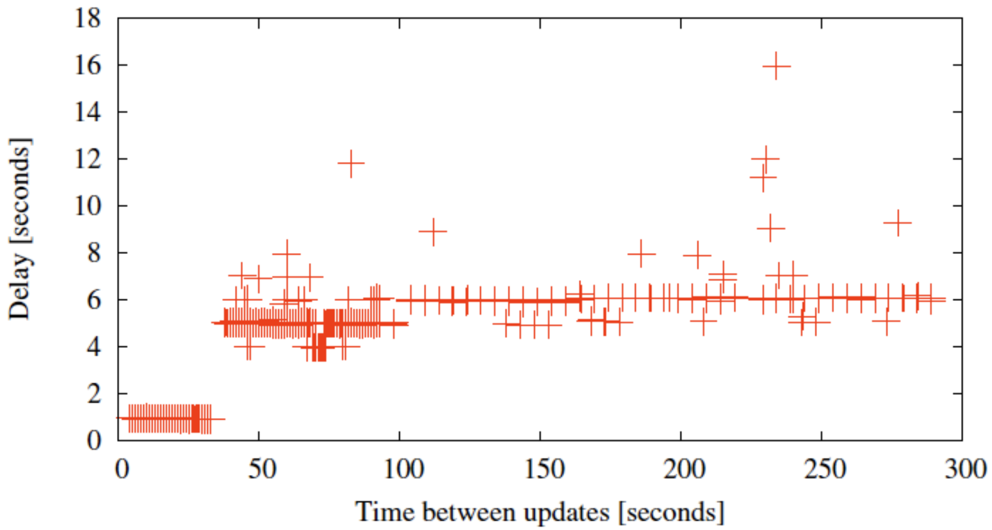
$$C_{u,p}(t, x, x_0) = \begin{cases} 0 & \text{if } t = 0, x = x_0 \\ p + C_{u,p}(t-1, x) & \text{if } t > 0, x = 0 \\ C_{u,p}(t-1, x-1) & \text{if } t > 0, x > u \\ p + C_{u,p}(t-1, x-1) & \text{if } t > 0, 0 < x <= u \\ \text{Undefined} & \text{else} \end{cases}$$

$$(g'', r'', s'') = \arg\min_{g,r,s}\{C_{30,0.324}(t - req_g(g), g, g_0) + C_{31,0.466}(t, r, r_0) + C_{6,0.645}(t, s, s_0) | t = t_{limit} - req_s(s)\}$$

Minimization function

DELAY MODEL FOR GPS

› Time To First Fix for Assisted-GPS with Nokia N95 8GB



$$req_g(x) = \begin{cases} 1 & \text{if } x <= 30 \\ 6 & \text{if } x > 30 \end{cases}$$

MINIMIZE CONSUMPTION

$$req_g(x) = \begin{cases} 1 & \text{if } x <= 30 \\ 6 & \text{if } x > 30 \end{cases}$$

Delay functions

$$req_s(x) = 1$$

Recursive Power Function

$$C_{u,p}(t, x, x_0) = \begin{cases} 0 & \text{if } t = 0, x = x_0 \\ p + C_{u,p}(t-1, x) & \text{if } t > 0, x = 0 \\ C_{u,p}(t-1, x-1) & \text{if } t > 0, x > u \\ p + C_{u,p}(t-1, x-1) & \text{if } t > 0, 0 < x <= u \\ \text{Undefined} & \text{else} \end{cases}$$

$$(g'', r'', s'') = \arg\min_{g,r,s}\{C_{30,0.324}(t - req_g(g), g, g_0)$$
$$+C_{31,0.466}(t, r, r_0) + C_{6,0.645}(t, s, s_0) | t = t_{limit} - req_s(s)\}$$

Minimization function

RECURSIVE POWER FUNCTION

- r_0: seconds since an active radio was asked to power off
- t_{limit}: the limit for getting a new fix
- r'': when to turn the radio on

RECURSIVE POWER FUNCTION

$C_{6,0.645}(t_{limit}\text{-}req_s(r),r,r_0)$, $t_{limit}=10$, $r_0=0$

	0	1	2	3	4	5	6	7	8	9
0	0	0,645	1,29	1,935	2,58	3,225	3,87	4,515	4,515	4,515
1	-	0,645	1,29	1,935	2,58	3,225	3,87	4,515	5,16	5,16
2	-	-	1,29	1,935	2,58	3,225	3,87	4,515	5,16	5,805
3	-	-	-	1,935	2,58	3,225	3,87	4,515	5,16	5,805
4	-	-	-	-	2,58	3,225	3,87	4,515	5,16	5,805
5	-	-	-	-	-	3,225	3,87	4,515	5,16	5,805
6	-	-	-	-	-	-	3,87	4,515	5,16	5,805
7	-	-	-	-	-	-	-	3,87	5,16	5,805
8	-	-	-	-	-	-	-	-	3,87	5,805
9	-	-	-	-	-	-	-	-	-	**3,87**

$r''=9$

MINIMIZE CONSUMPTION

$$req_g(x) = \begin{cases} 1 & \text{if } x <= 30 \\ 6 & \text{if } x > 30 \end{cases}$$

Delay functions

$$req_s(x) = 1$$

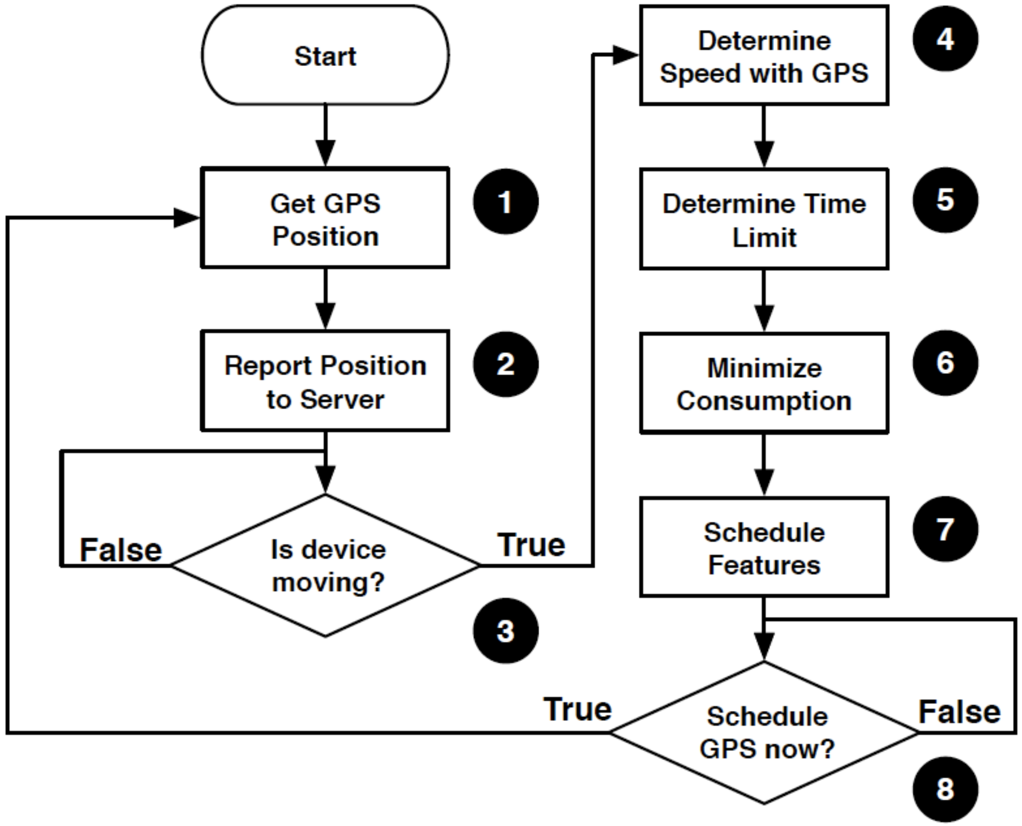
Recursive Power Function

$$C_{u,p}(t, x, x_0) = \begin{cases} 0 & \text{if } t = 0, x = x_0 \\ p + C_{u,p}(t-1, x) & \text{if } t > 0, x = 0 \\ C_{u,p}(t-1, x-1) & \text{if } t > 0, x > u \\ p + C_{u,p}(t-1, x-1) & \text{if } t > 0, 0 < x <= u \\ \text{Undefined} & \text{else} \end{cases}$$

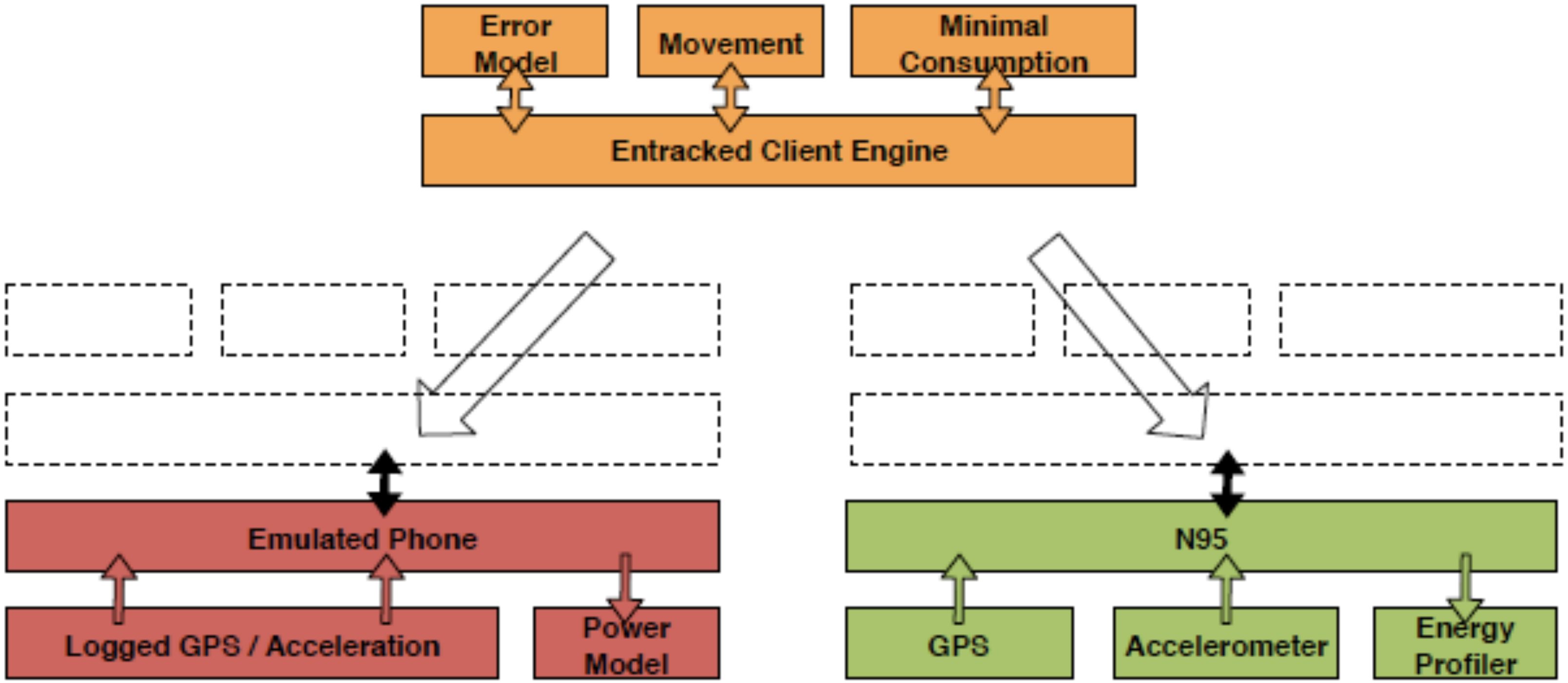
$$(g'', r'', s'') = \arg\min_{g,r,s}\{C_{30,0.324}(t - req_g(g), g, g_0)$$
$$+ C_{31,0.466}(t, r, r_0) + C_{6,0.645}(t, s, s_0) | t = t_{limit} - req_s(s)\}$$

Minimization function

EnTracked – CLIENT SIDE LOGIC



IMPLEMENTATION

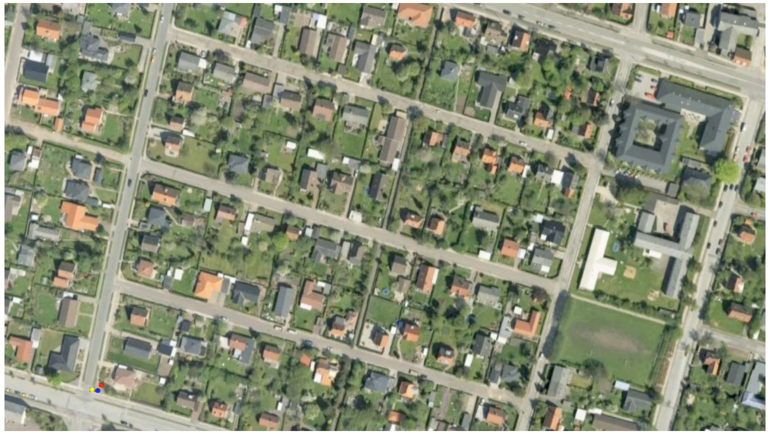


DEPLOYMENT

- EnTracked Server
  - Implemented in Java
  - Running on a public IP adresse
- EnTracked Client
  - N95 8GB
  - Implemented in Python for S60
- Connection between client and server over UMTS using TCP.
- Nokia Energy Profiler
- Position error using manual-entered ground truth

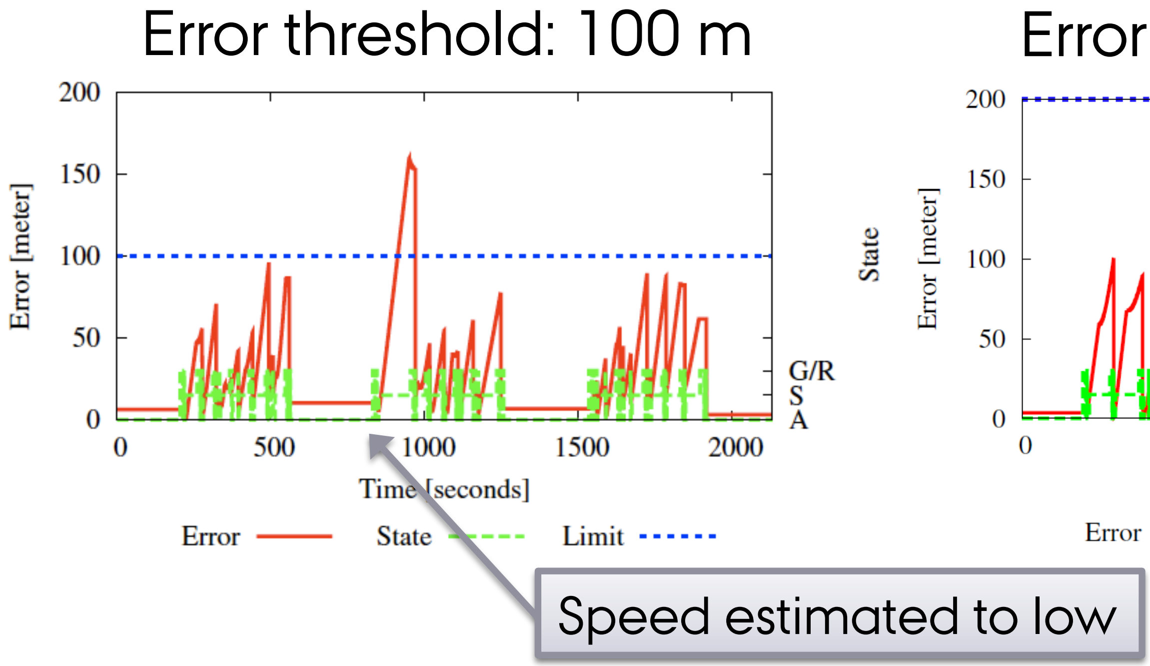
EnTracked - Video

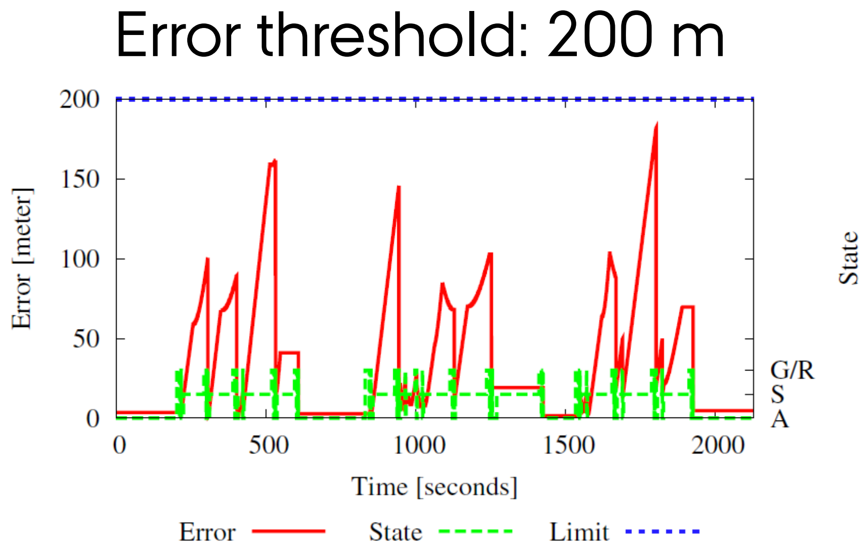
- Yellow (worm) – Ground Truth
- Red - EnTracked with user error threshold 100 meter
- Blue - EnTracked with user error threshold 200 meter



ROBUSTNESS

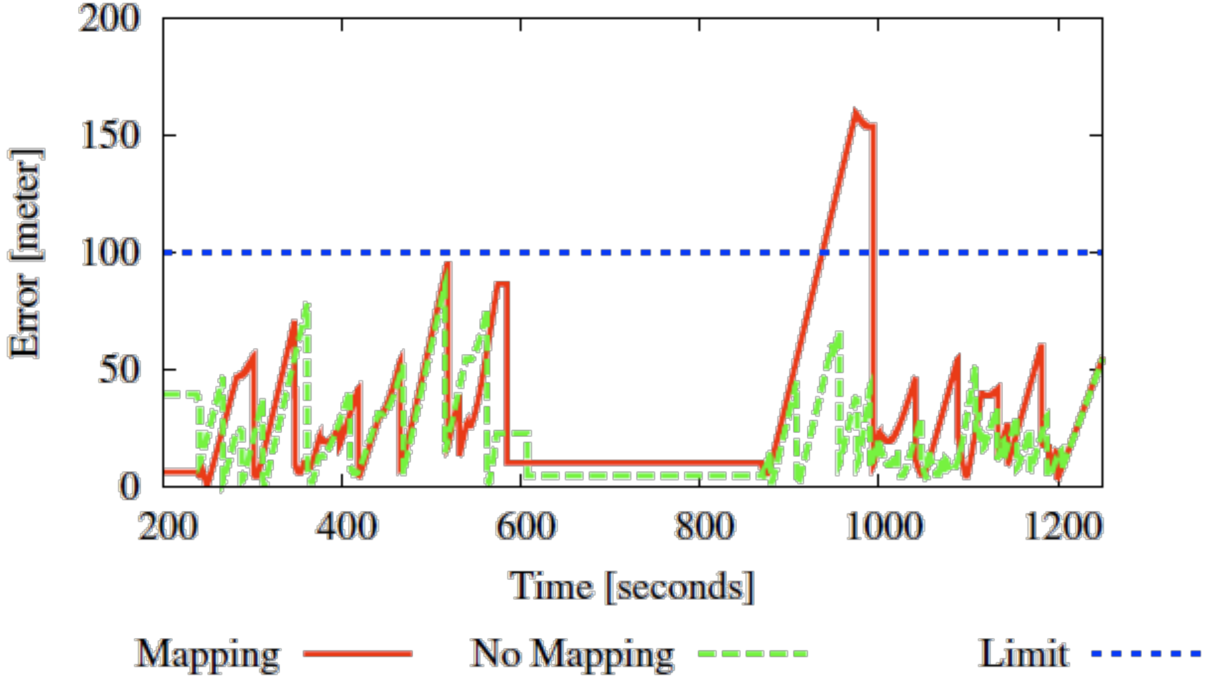
› Error of server-known position





ROBUSTNESS

› Trust position estimates less

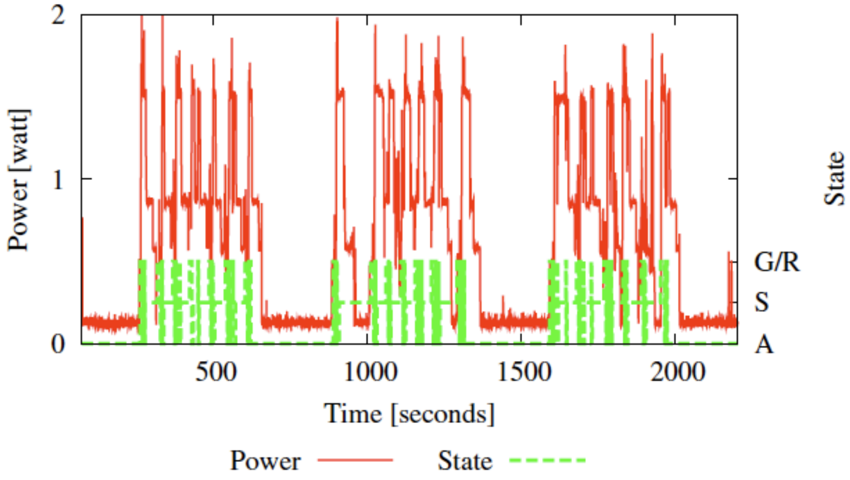


› Other option would be to trust speed less

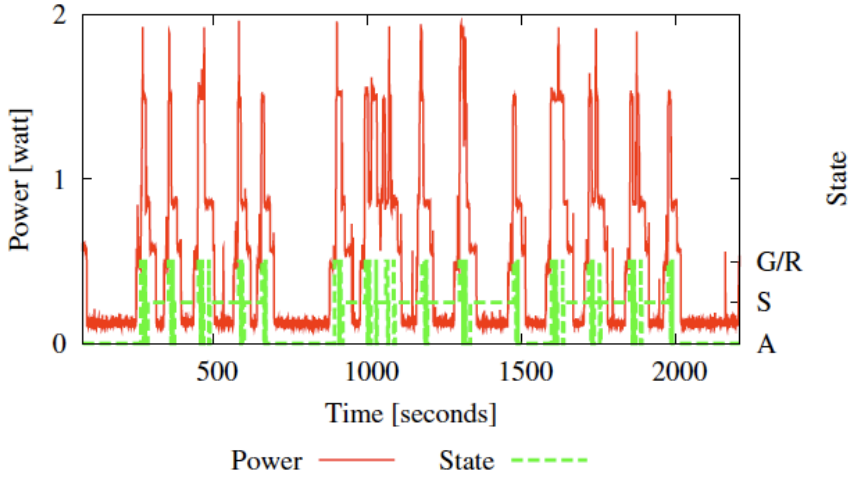
ENERGY EFFICIENCY

› Measured with Nokia Energy Profiler

Error threshold: 100 m



Error threshold: 200 m



DISCUSSION

- Depend on request patterns
  - Need position twice a day (Use a request protocol)
  - Anytime within limit (Use EnTracked)
- Depend on movement patterns
  - Moving fast all the time (Small savings)
  - Moving fast only a few times a day (Large savings)

Conclusions

- EnTracked – a dynamic tracking sytem
- Methods for changing system conditions
  - Speed and speed accuracy
  - Position accuracy
  - Mobility
- Minimization algorithm to select optimal plan
- Evaluation in terms of both emulation and deployment for energy-efficiency and robustness