# URBANEYE

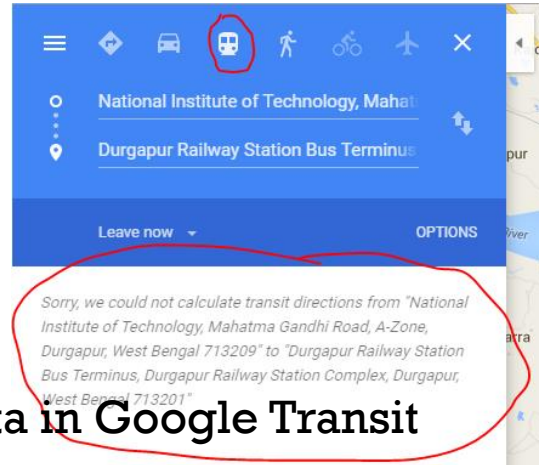
## An Outdoor Localization System for Public Transport

Aviral Shrivastava, Bivas Mitra, Niloy Ganguly, Sandip Chakraborty
{Indian Institute of Technology, Kharagpur}

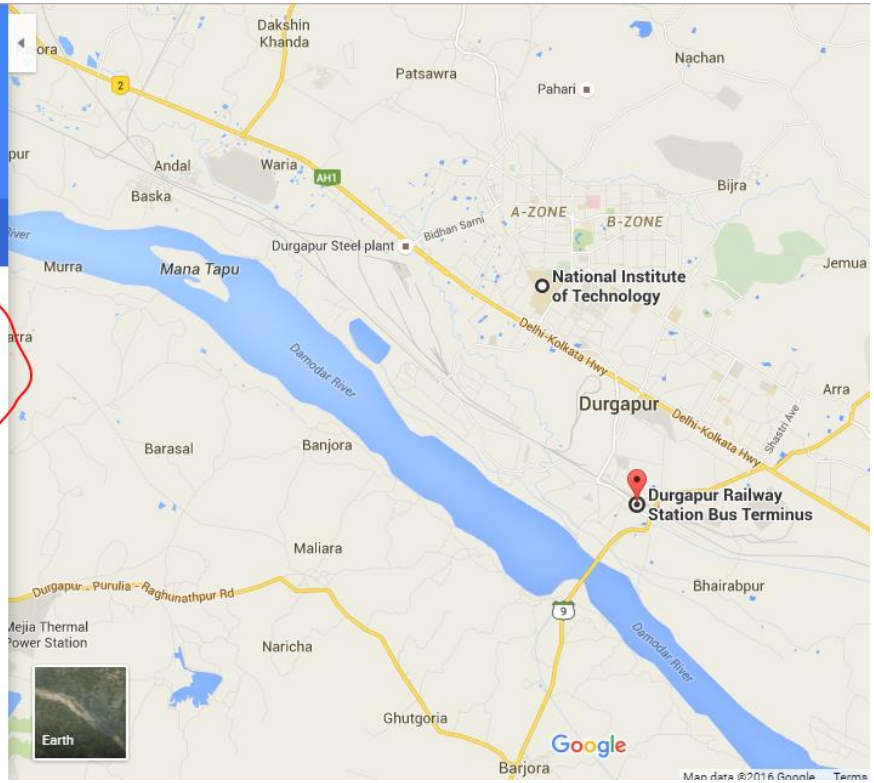
Sujoy Saha, Subrata Nandi
{National Institute of Technology, Durgapur}

- Rohit Verma
PhD Student
Complex Network Research Group (CNeRG)
Indian Institute of Technology, Kharagpur

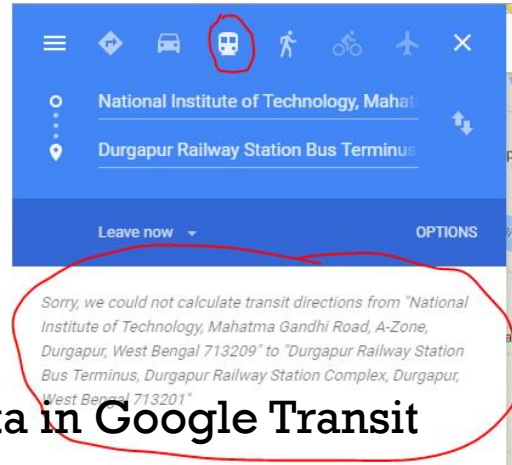
# PROBLEMS OF PUBLIC TRANSPORT IN DEVELOPING COUNTRIES



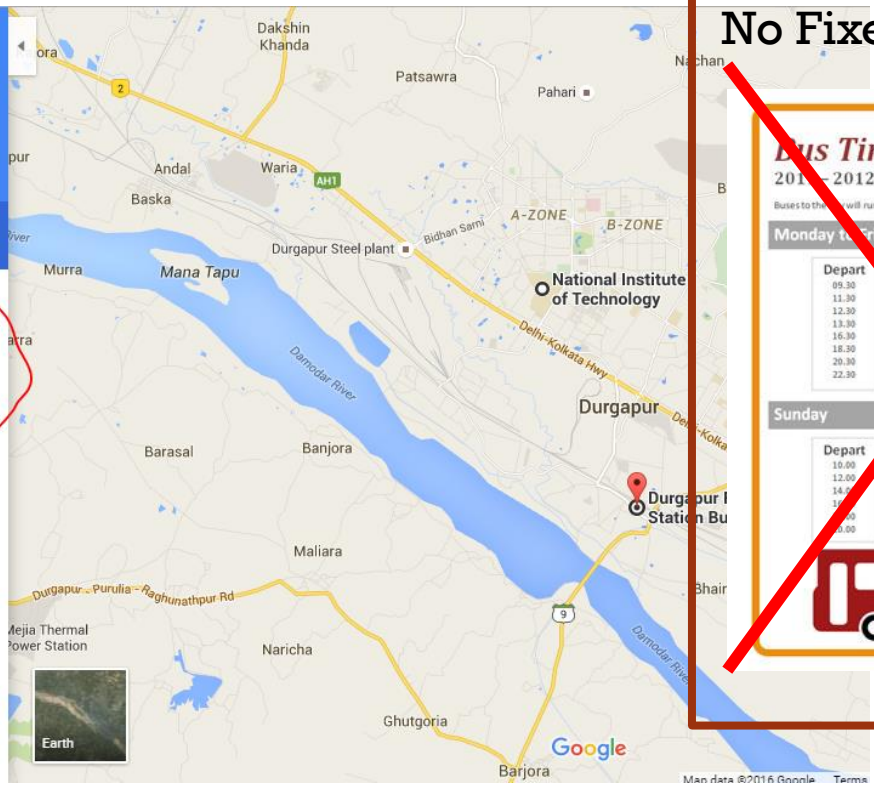

No Data in Google Transit



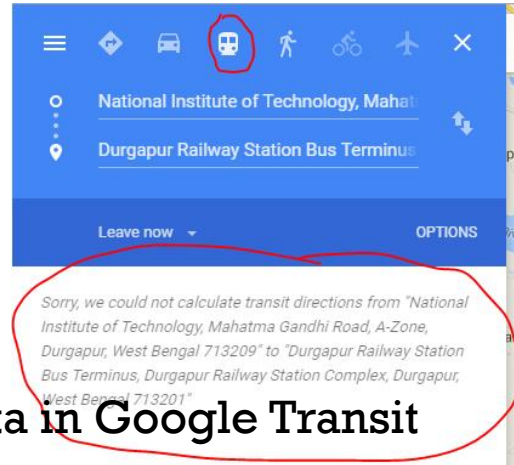
# PROBLEMS OF PUBLIC TRANSPORT IN DEVELOPING COUNTRIES



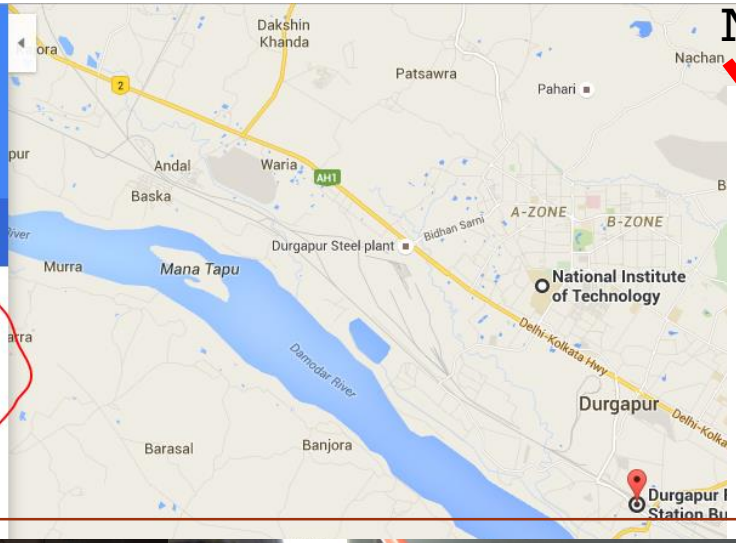

No Data in Google Transit



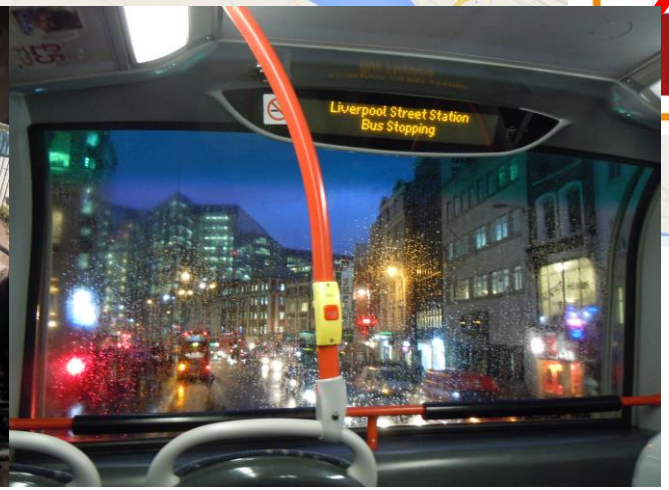

# PROBLEMS OF PUBLIC TRANSPORT IN DEVELOPING COUNTRIES




No Data in Google Transit




No Fixed Time Table


No Information Boards

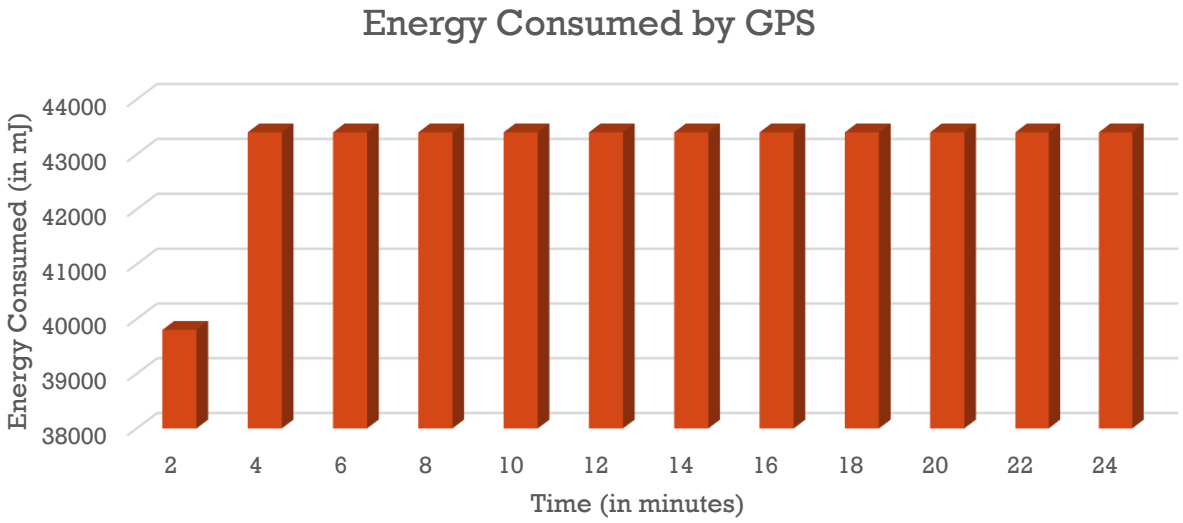
# OBJECTIVE

- An application which runs on a commuter's mobile phone is a feasible solution

- The app should give the following
  - Current location of the commuter
  - Time that the bus would take to reach the destination

# A G.P.S BASED SOLUTION

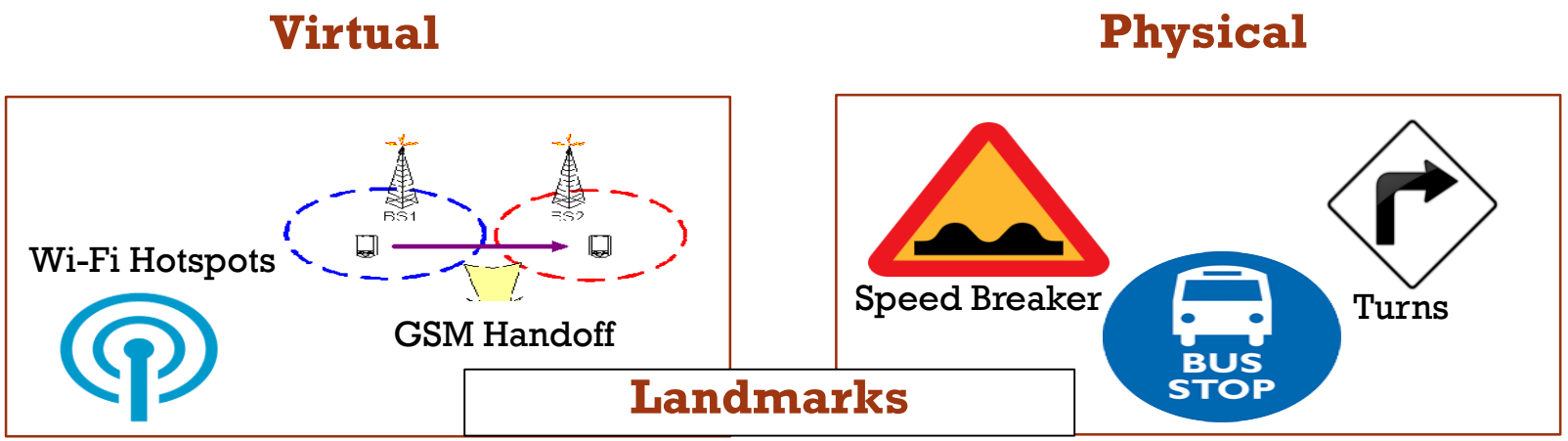
- Empower the user with a GPS based mobile application which she can use anytime
- But!!



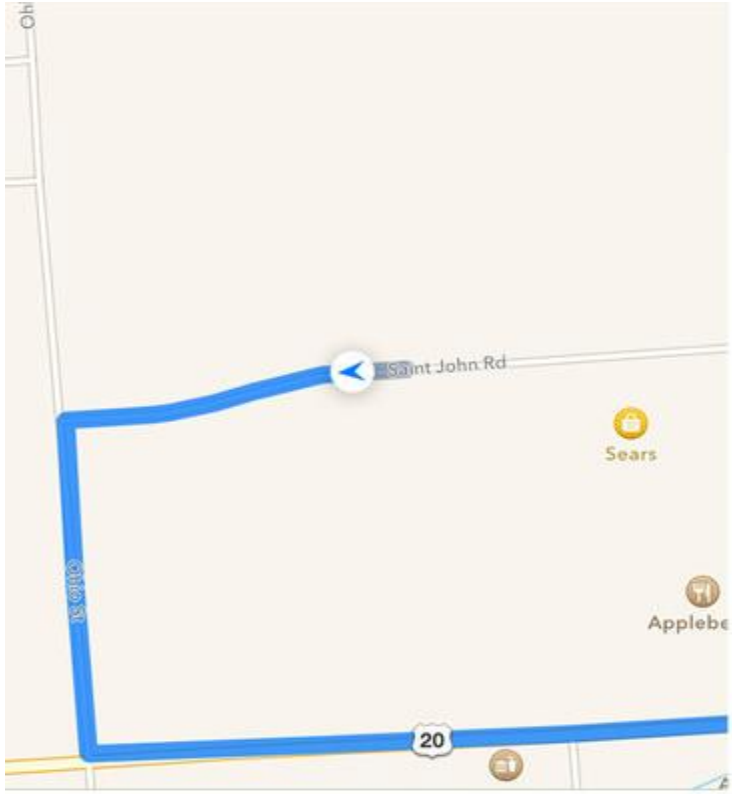
# WHAT CAN BE DONE?

Use of **mobile sensors** to localize vehicles without using GPS

**Landmarks**: Specific anomalies on the route which can be detected using the mobile sensors



# COLLECT SENSOR DATA



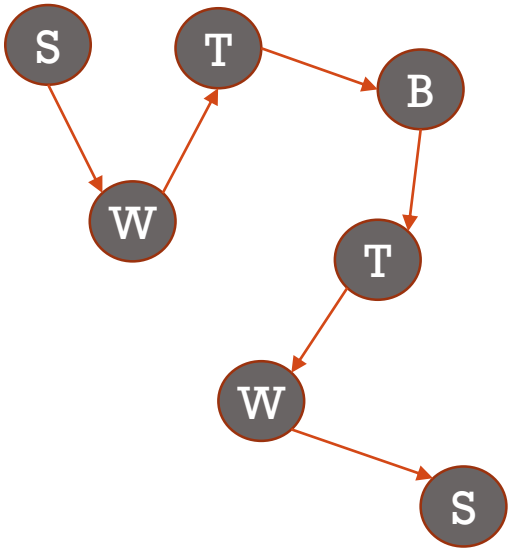

- Collect mobile Sensor data on a route
- Annotate position of landmarks when encountered


Collect → Build Database & Traversal Graph (Automata) → Detect Landmark using Sensors → Localize

Database Generation

Navigation

# DATABASE AND TRAVERSAL GRAPH

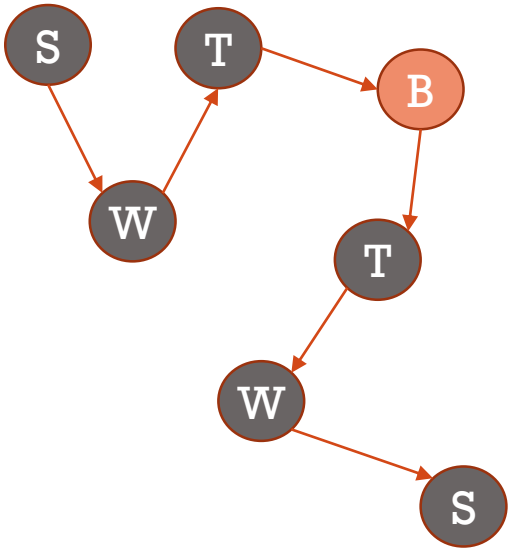




Collect → Build Database & Traversal Graph (Automata) → Detect Landmark using Sensors → Localize

Database Generation

Navigation

# DETECT LANDMARK USING SENSORS

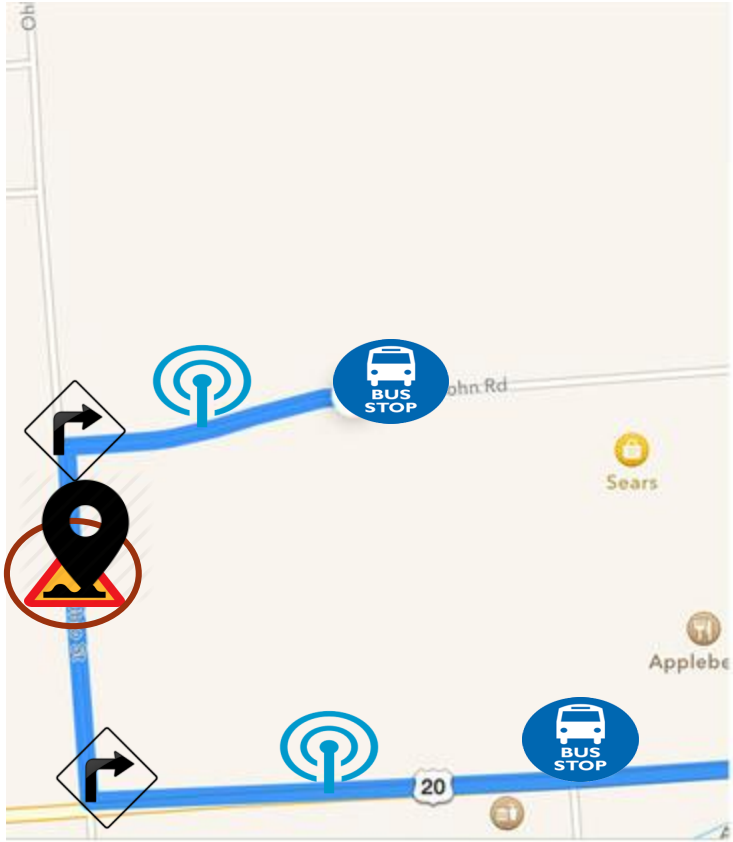



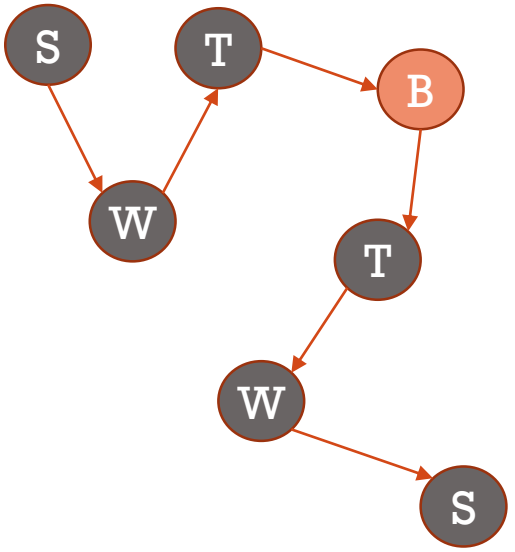

Collect → Build Database & Traversal Graph (Automata) → Detect Landmark using Sensors → Localize

Database Generation

Navigation

# LOCALIZE

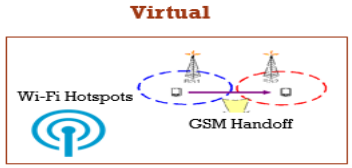


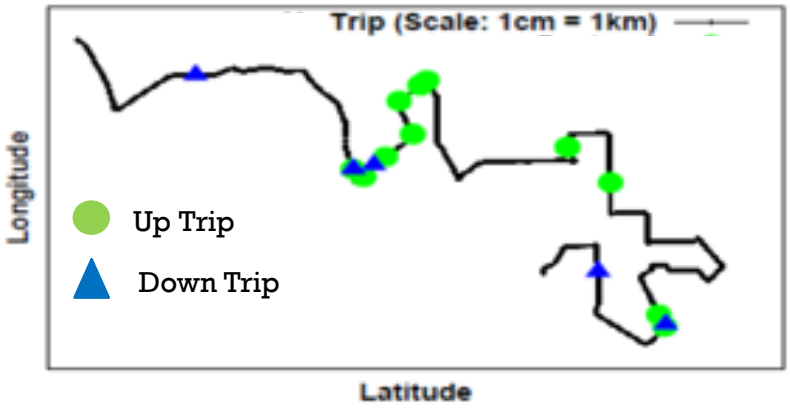

Collect → Build Database & Traversal Graph (Automata) → Detect Landmark using Sensors → Localize

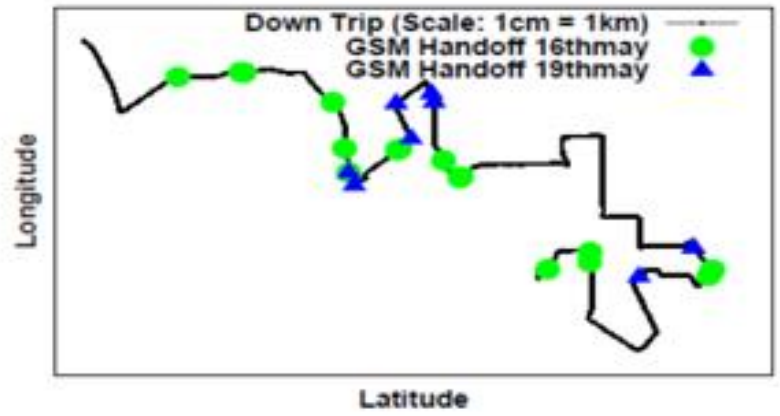
Database Generation

Navigation

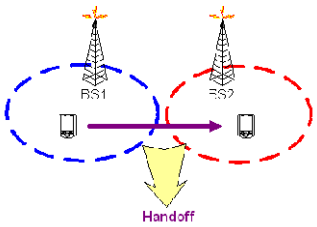



# ARE VIRTUAL LANDMARKS FEASIBLE?



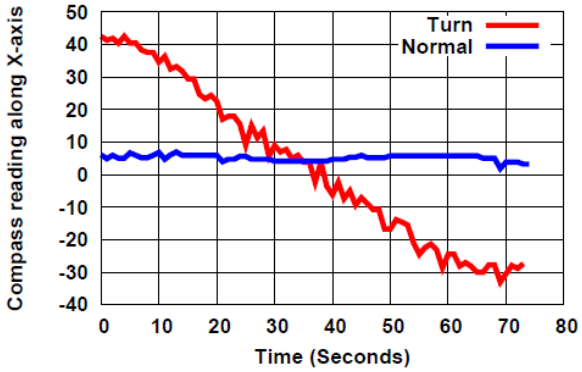

- Absence of fixed Wi-Fi hotspots
- Only stray hotspots detected



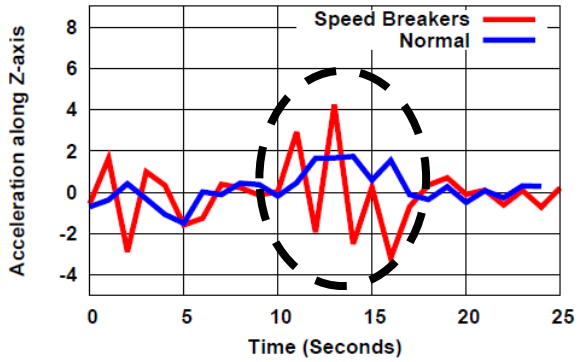

- Unplanned placement of cell towers.


# WHAT ABOUT THE PHYSICAL ONES?

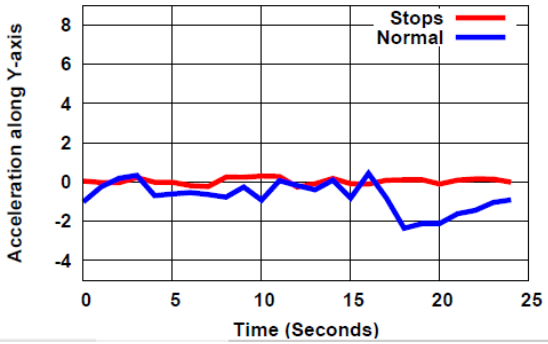
(a) Normal vs Turn




(b) Normal vs Speed Breaker

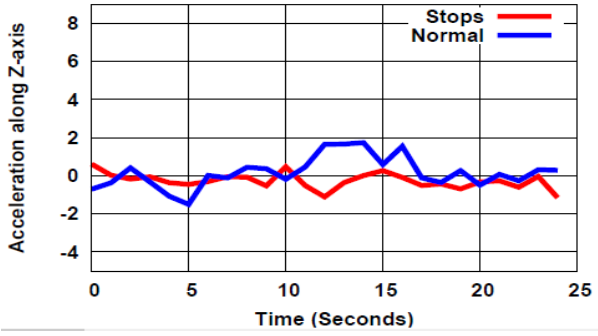



(c) Normal vs Stop (Y-axis)





(d) Normal vs Stop (Z-axis)



# AN ISSUE WITH PHYSICAL LANDMARKS

## Volatile Landmarks

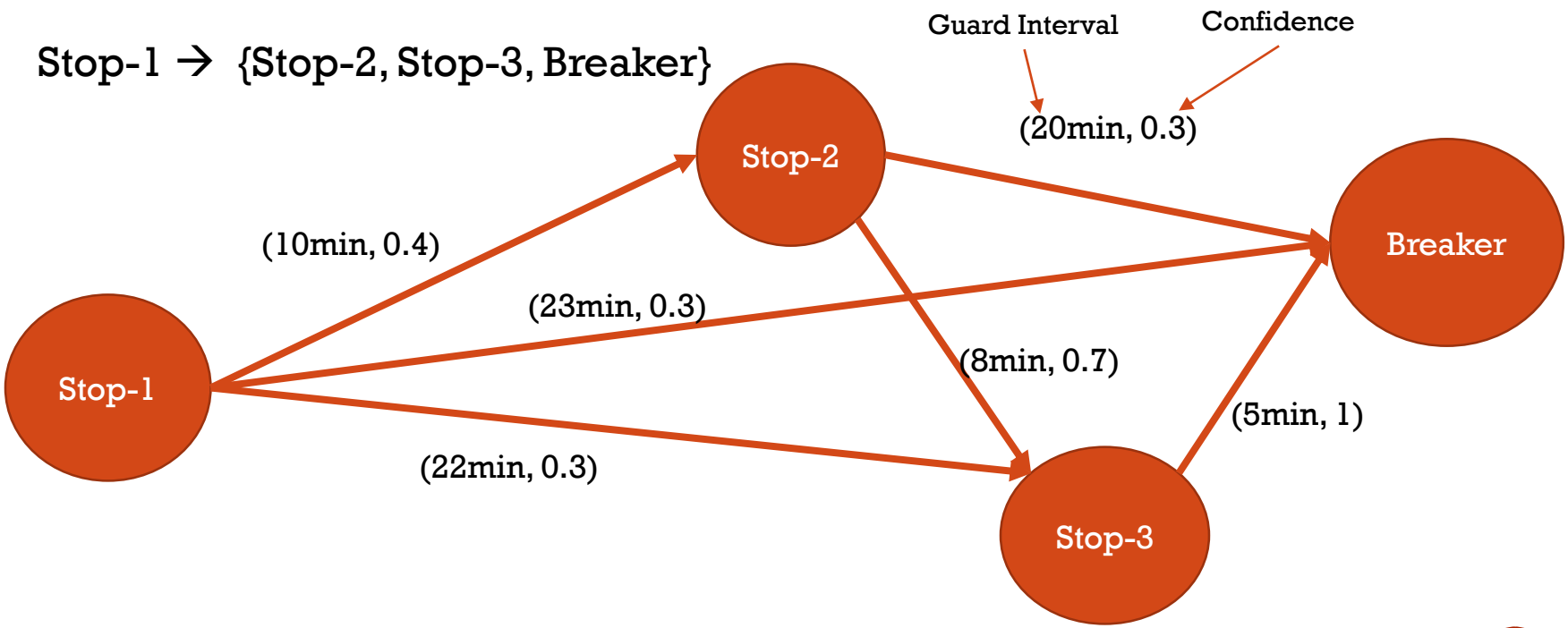
May or may not occur on a route.


Bus skips a designated bus-stop

Bus stops at no-stop locations

# HANDLING VOLATILE LANDMARKS

Confidence of Landmarks: Probability of a bus encountering a landmark given that it has already encountered the previous landmark



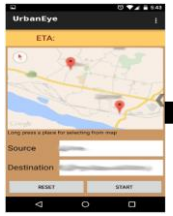

Probabilistic Timed Automata (PTA)

# WE THUS PRESENT - URBANEYE

## Builds Landmark Database


Collect Inertial Sensor Data

Generate Landmark Database

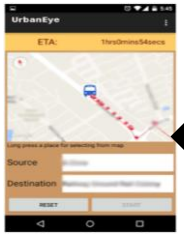



**Gets Source & Destination**

Collect Inertial Sensor Data

Detect Landmark and Localize

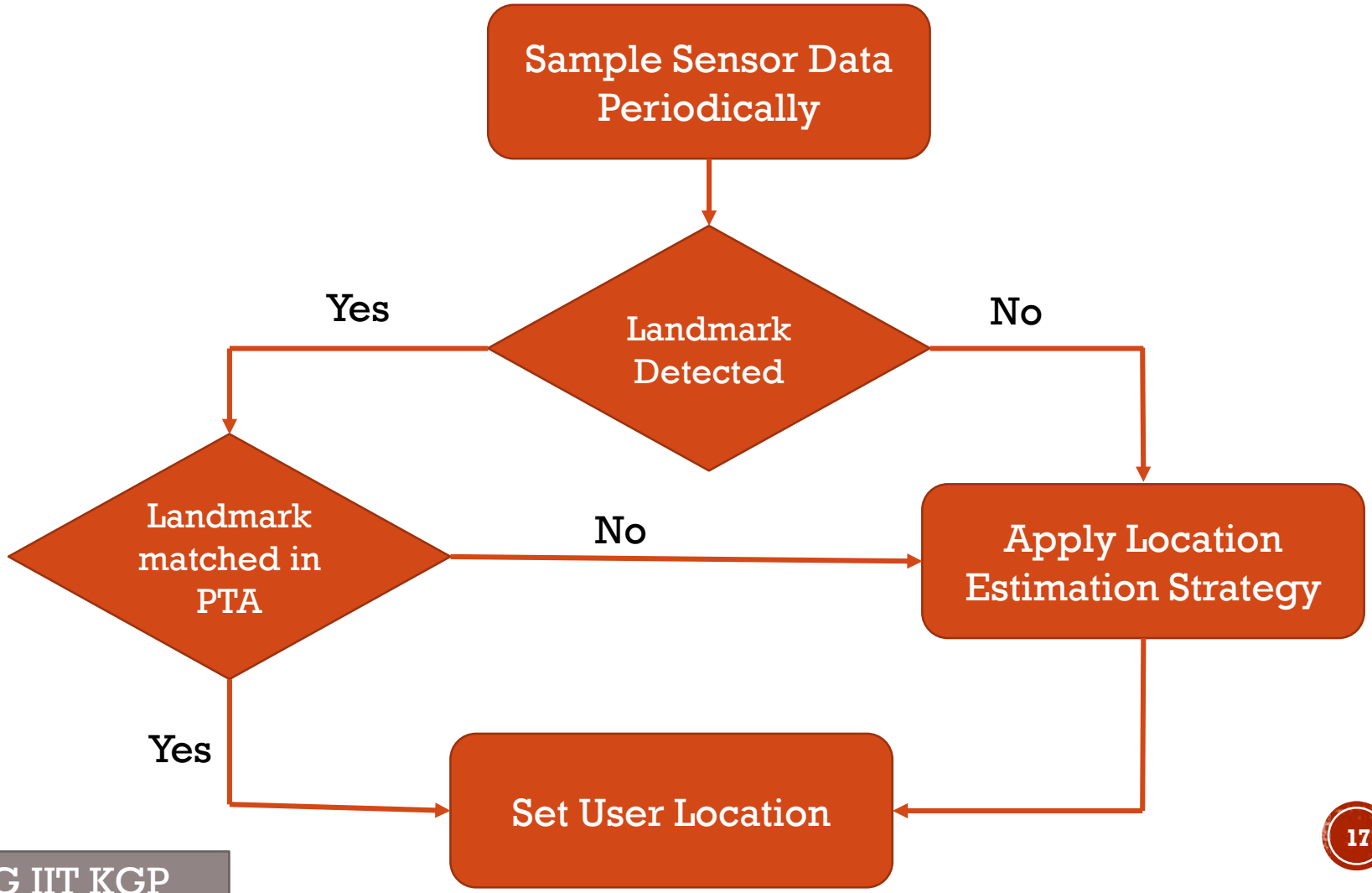
Estimate Time to Destination



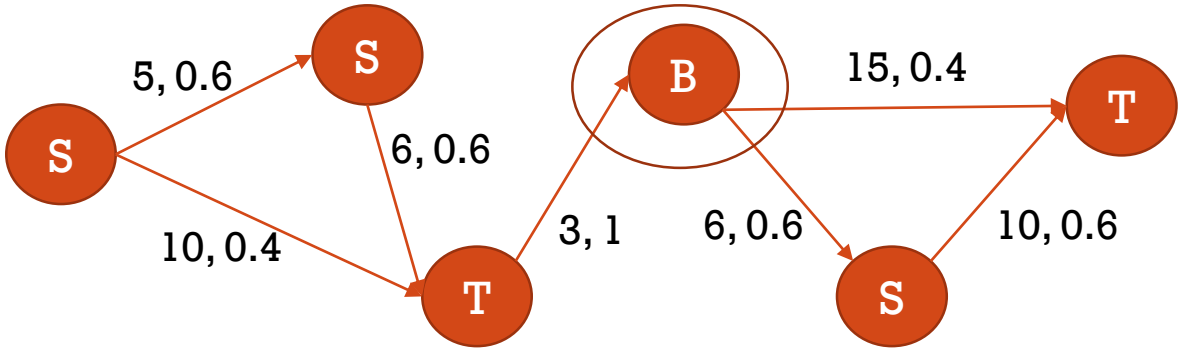

**Energy–Efficient Navigation**

## Performs Navigation

# NAVIGATION: LOCALIZATION



# NAVIGATION: TRAVEL TIME ESTIMATION




Assuming there are a total of *n* landmarks between the source and destination and *m* landmarks have already passed,

$$ETA = \sum_{i=m}^{n-1} p_{ji} * g_{ji}$$

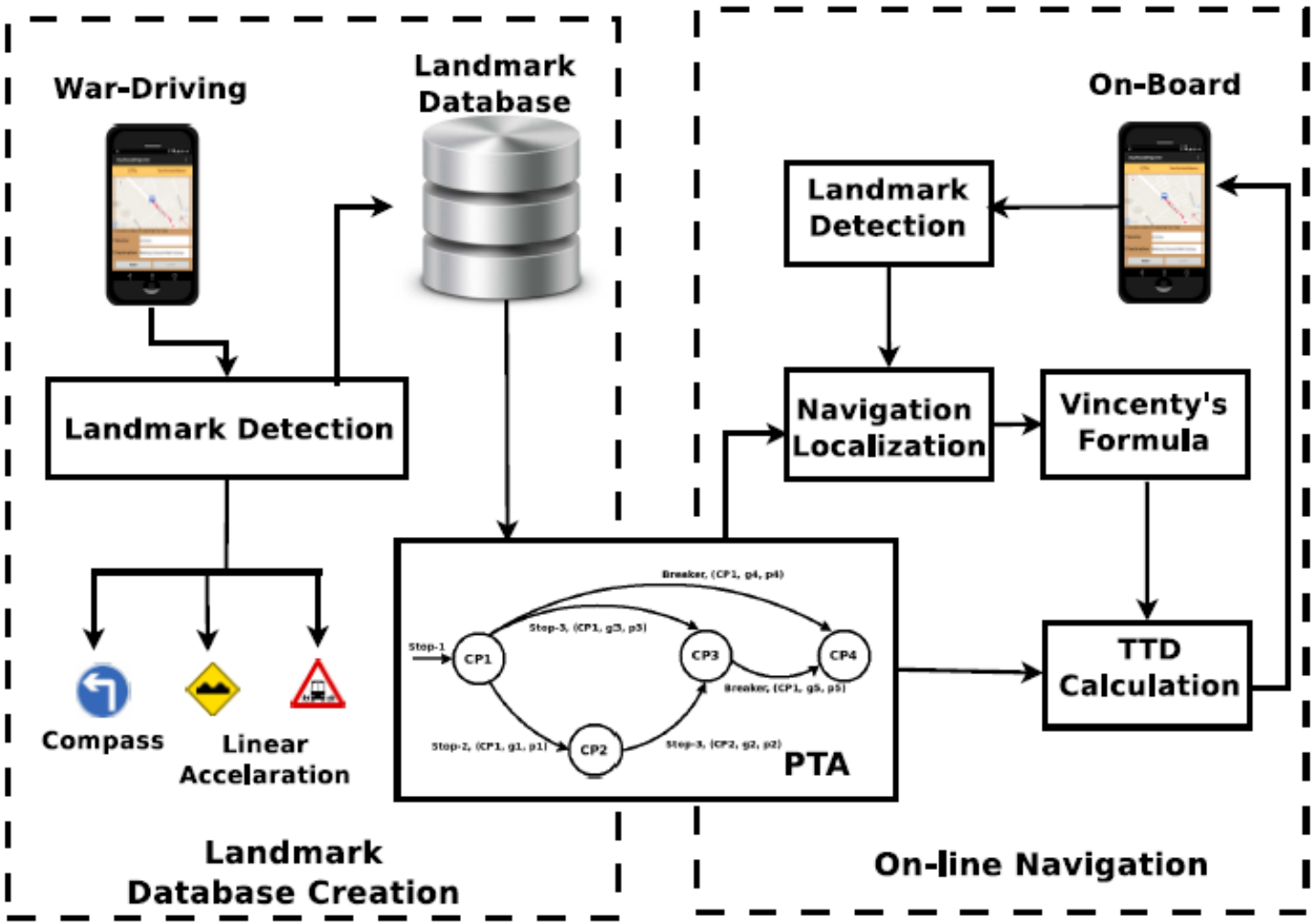
where *j* is the landmark from which *i* was reached

In this example, n = 6 and m = 3

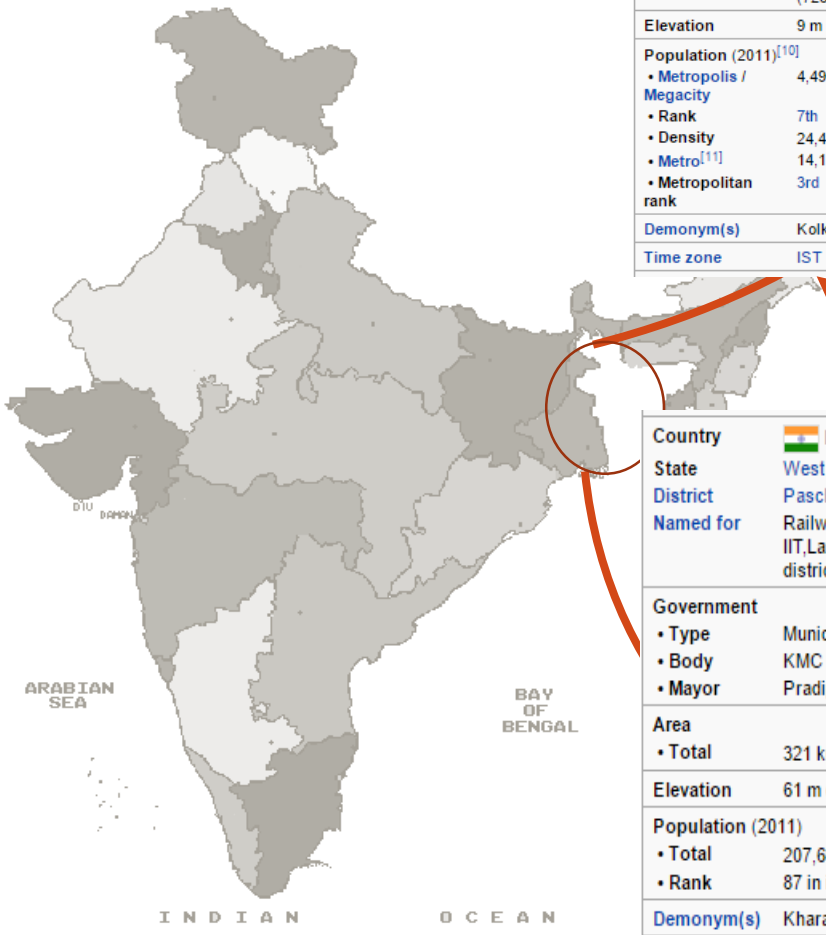
Hence we have,

ETA = (15*0.4) + (6*0.6) + 10*0.6 = 15.6

# SYSTEM ARCHITECTURE



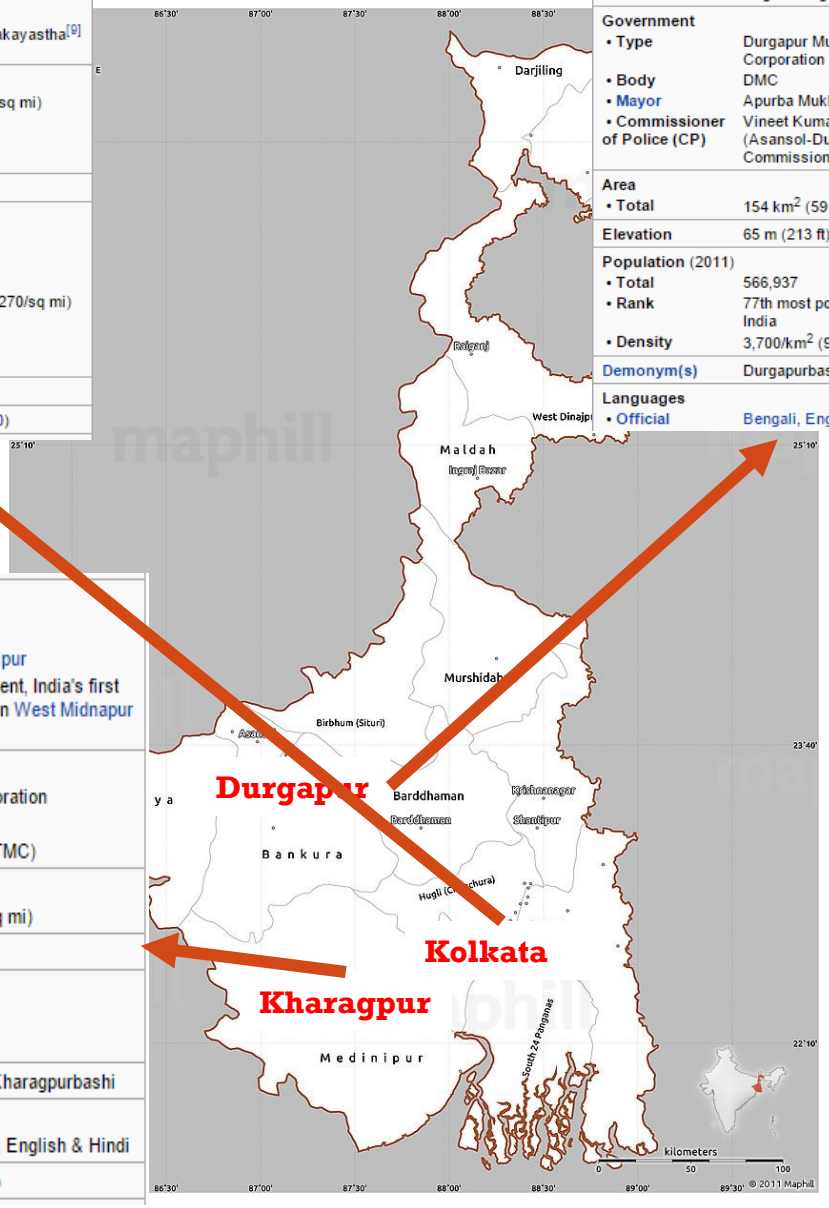
# EXPERIMENTS




| | |
|---|---|
| Country | India |
| State | West Bengal |
| Division | Presidency |
| District | Kolkata[A] |
| **Government** | |
| • Type | Mayor–Council |
| • Body | KMC |
| • Mayor | Sovan Chatterjee[7] |
| • Sheriff | Ranjit Mallik[8] |
| • Police commissioner | Surajit Kar Purakayastha[9] |
| **Area** | |
| • Metropolis / Megacity | 205 km² (79.15 sq mi) |
| • Metro | 1,886.67 km² (728.45 sq mi) |
| Elevation | 9 m (30 ft) |
| **Population (2011)**[10] | |
| • Metropolis / Megacity | 4,496,694 |
| • Rank | 7th |
| • Density | 24,429/km² (63,270/sq mi) |
| • Metro[11] | 14,112,536 |
| • Metropolitan rank | 3rd |
| Demonym(s) | Kolkatan |
| Time zone | IST (UTC+05:30) |

| | |
|---|---|
| Country | India |
| State | West Bengal |
| District | Paschim Medinipur |
| Named for | Railway settlement, India's first IIT,Largest city in West Midnapur district |
| **Government** | |
| • Type | Municipal Corporation |
| • Body | KMC |
| • Mayor | Pradip Sarkar(TMC) |
| **Area** | |
| • Total | 321 km² (124 sq mi) |
| Elevation | 61 m (200 ft) |
| **Population (2011)** | |
| • Total | 207,604[1] |
| • Rank | 87 in india |
| Demonym(s) | Kharagpurian, Kharagpurbashi |
| **Languages** | |
| • Official | Bengali, Telugu, English & Hindi |
| Time zone | IST (UTC+5:30) |

| | |
|---|---|
| Country | India |
| State | West Bengal |
| District | Barddhaman |
| Established | Late 1950s |
| Founded by | Dr. Bidhan Chandra Roy (former chief minister of West Bengal) |
| Named for | Durgamohan Chattopadhyay, former Zamindar of Sagarbhanga |
| **Government** | |
| • Type | Durgapur Municipal Corporation |
| • Body | DMC |
| • Mayor | Apurba Mukherjee |
| • Commissioner of Police (CP) | Vineet Kumar Goyal, IPS (Asansol-Durgapur Police Commissionerate) |
| **Area** | |
| • Total | 154 km² (59 sq mi) |
| Elevation | 65 m (213 ft) |
| **Population (2011)** | |
| • Total | 566,937 |
| • Rank | 77th most populated city of India |
| • Density | 3,700/km² (9,500/sq mi) |
| Demonym(s) | Durgapurbashi/Durgapurians |
| **Languages** | |
| • Official | Bengali, English |



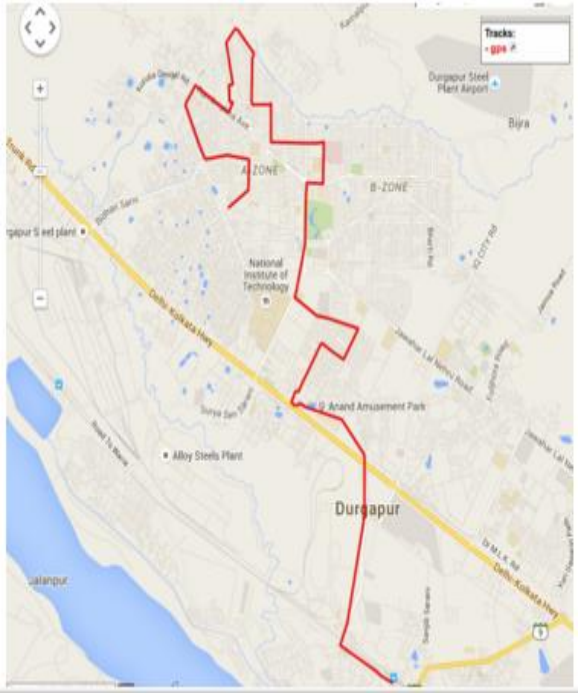
# WAR DRIVING


- We collected sensor trails for up and down trips (**60 trails**) over a month's duration.
- Total coverage of the routes was around **75kms**
- Different types of devices were used, for e.g. **Google Nexus4, Micromax A092, Samsung Galaxy Tab 3**

# EVALUATION : DETECTION OF LANDMARKS

Detection Accuracy for a route in Durgapur

| Landmark | Actual | Detected (%) | False Positive (%) | FPE* (%) |
|---|---|---|---|---|
| Turn | 32 | 31 (96.8) | 10 (31.2) | 0 (0) |
| Speed Breaker | 9 | 8 (87.5) | 3 (37.5) | 0 (0) |
| Bus Stops | 42 | 34 (80.9) | 13 (38.2) | 2 (4.76) |




- Bus Stops have comparatively low detection because of volatility
- Applying PTA guard intervals reduces the false positive cases considerably





*FPE: False positives after elimination by PTA guard intervals

# EVALUATION : DETECTION OF LANDMARKS

Accuracy metrics for the 3 cities

| City | Turns | | | Speed Breakers | | | Bus Stops | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | A | P | R | A | P | R | A |
| Durgapur | 1 | 0.94 | 0.94 | 1 | 1 | 1 | 0.93 | 0.83 | 0.78 |
| Kharagpur | 1 | 1 | 1 | 1 | 0.94 | 0.94 | 0.88 | 0.88 | 0.78 |
| Kolkata | 1 | 0.97 | 0.97 | 1 | 0.89 | 0.89 | 0.94 | 0.81 | 0.77 |

- Almost close to 1 values for Precision, Recall and Accuracy for turns and speed breakers
- Bus stops again have comparatively low values because of volatility

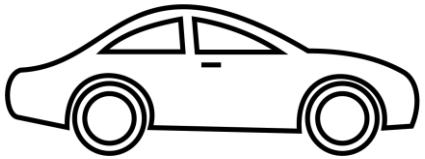
# NAVIGATION EVALUATION

- We compare localization accuracy w.r.t **Dejavu**.
- We compared the travel time estimates with that of **Google maps**.
- Energy consumption comparisons were done against **Dejavu** and **GPS**.

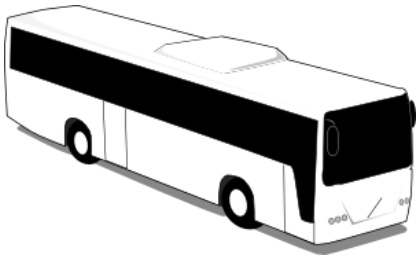
Aly, Heba, and Moustafa Youssef. "Dejavu: an accurate energy-efficient outdoor localization system." *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2013.

# COMPETING HEURISTIC

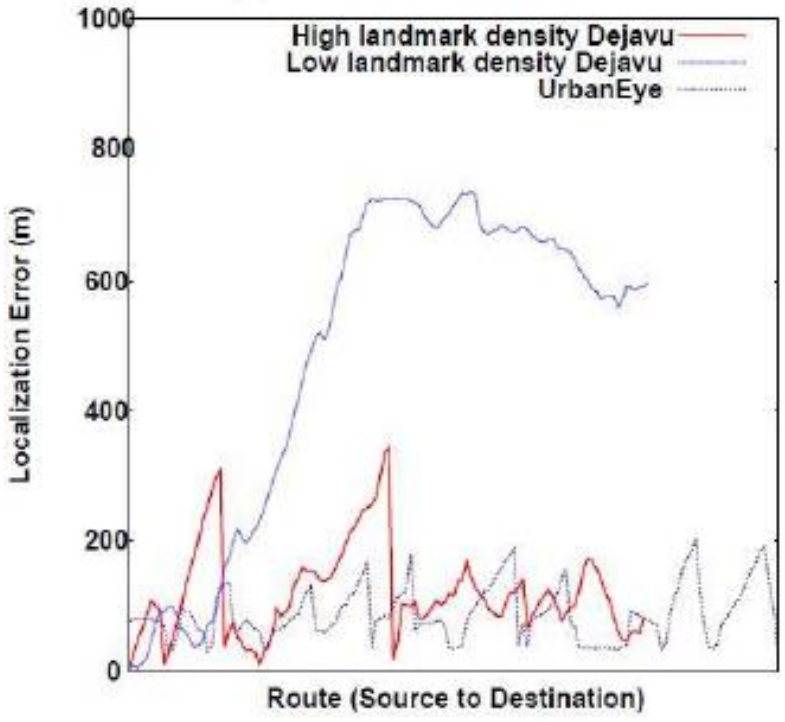
**Dejavu (Alexandria, Egypt)**




**UrbanEye (Kolkata, India)**



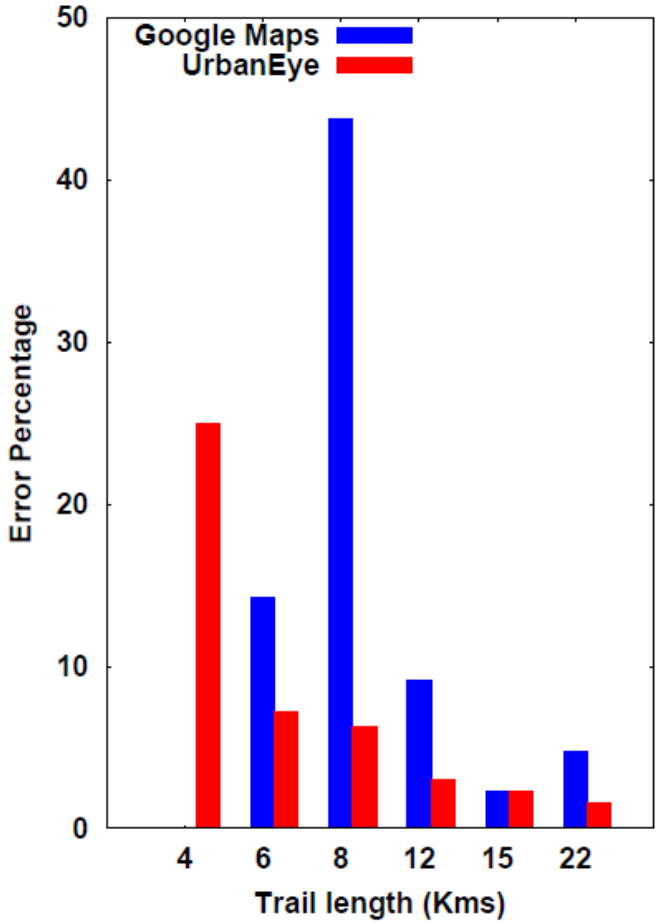
# EVALUATION: LOCALIZATION ACCURACY




Localization Error over a route

- Dejavu performs very bad for low density landmarks routes
- The average localization error is 50m

# EVALUATION : TRAVEL TIME ESTIMATE




- Simulated the bus route for Google Maps
- Google maps doesn't consider wait time at bus stops
- Hence, UrbanEye gives better estimate than Google Maps

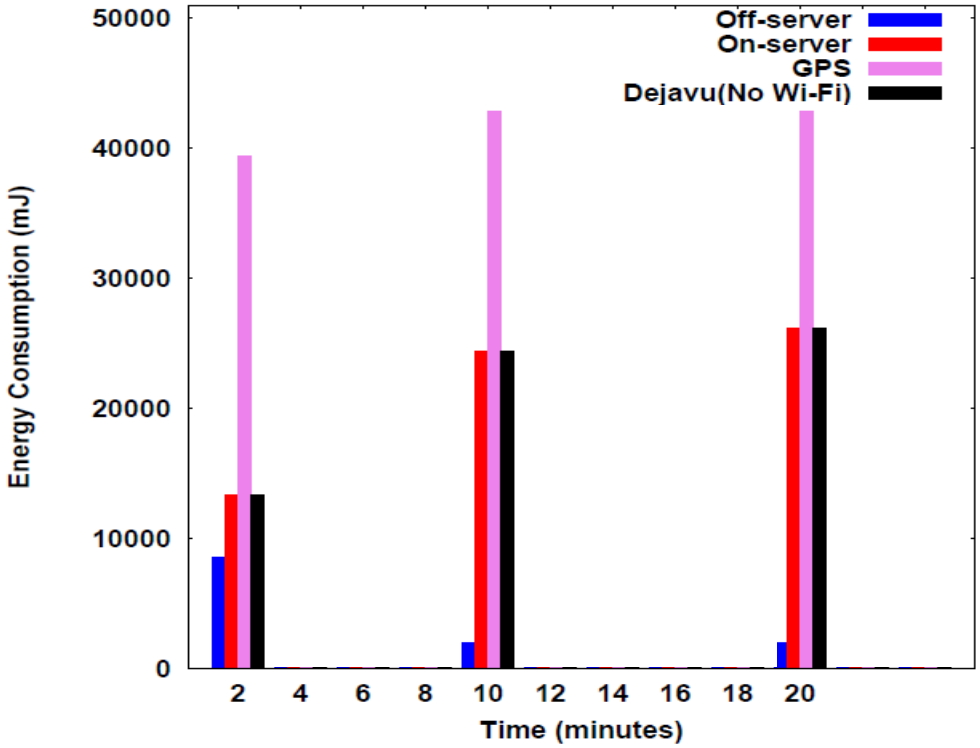
Error Percentage is given as;

$$\frac{\text{abs (Actual Time – Estimated Time)}}{\text{Actual Time}} \times 100$$

# ON-SERVER AND OFF-SERVER URBANEYE

- We developed two versions of the application
  - **On-server** : Navigation is carried on server and sensor data is offloaded
  - **Off-server** : Navigation is performed on the device
- We hence evaluate how much overhead does offloading sensor data have over energy consumption

# EVALUATION : ENERGY CONSUMPTION




- UrbanEye consumes 50% less energy compared to GPS
- The on-server version consumes same energy as Dejavu
- The off-server version consumes 86% less energy than GPS

# CONCLUSION

- This is the first work which gives proper data structure and framework for **localization under uncertainty**
- The **PTA** efficiently utilizes the in uncertainty
- Compared to a deployed system **Google Maps** and research system **Dejavu**, UrbanEye fairs quite well

# ACKNOWLEDGEMENT


- Special Thanks to
  - Information Technology Research Academy, India
  - Xerox Research Centre India
  - ACM India - IARCS

# THANK YOU!

UrbanEye: http://www.cnergres.iitkgp.ac.in/urbaneye/

Follow the work of Complex Network Research Group (CNeRG), IIT KGP at:
Web: http://www.cnergres.iitkgp.ac.in/
Facebook: https://web.facebook.com/iitkgpcnerg
Twitter: https://www.twitter.com/cnerg