

Bluetooth Tracking without Discoverability

Simon Hay and Robert Harle

Winter Augmented Reality Meeting, 12-13 February 2009

Motivation

- Colleague searching
- **Computing for the Future of the Planet**
 - Location-aware power management
 - Dynamically optimising heating, cooling and computing devices based on room usage
 - Personal energy meter
 - Contextual information crucial for apportioning usage

Harle, R., Hopper, A.: The potential for location-aware power management.
UbiComp '08: Proceedings of the 10th International Conference on Ubiquitous Computing (Sep 2008)

Hopper, A., Rice, A.: Computing for the Future of the Planet.
Phil. Trans. R. Soc. A, 366(1881):3685—3697 (Oct 2008)



Goals

- Locate users to a small area, ideally placing them in a single room
- Track users as they move to permit dynamic optimisation on timescales of the order of seconds
- Track multiple users simultaneously
- Easily adopted by a high percentage of building users
- Avoid the issue of users forgetting to enable tracking in some way

Low-infrastructure Location Systems

- Quick deployment across entire buildings using standard infrastructure
- No retro-fit of dedicated infrastructure
- No need to issue occupants with custom tracking devices

Ubiquitous Sensing



- GSM/3G
- WiFi
- Bluetooth
- Camera
- Accelerometers
- Speaker/microphone

Mechanisms for Bluetooth Tracking

- Proximity based
 - Nominal range of 10m in free space, less indoors
 - Simple, reliable
- Signal strength based
 - ‘Fingerprinting’ (radio maps)
 - Radio propagation models

Problems with Bluetooth Tracking

- High tracking latency
 - Each scan takes >10.24 s: very slow update rate which does not support dynamic tracking
- Devices in the system must be discoverable
 - Security risk
 - Potential privacy risk
 - Impossible on latest handsets
- Connection disruption

Inquiry-Free Tracking

- *Connection-based* tracking
- devices are characterised as proximate if one can connect to the other
 - i.e. if beacon B can connect to handset H then B and H are proximate
- Each query targets a specific device, rather than every device
 - Multiple queries needed to find multiple local devices
 - Each query completes faster, allowing for multiple queries

Bluetooth Connections

- **A**synchronous **C**onnectionless **L**ink (ACL)
 - Most fundamental; must be established before any other connection can be made
- **L**ogical **L**ink **C**ontrol and **A**daptation **P**rotocol (L2CAP)
 - Packet based layer
 - Provides guaranteed packet sequencing and selected degree of delivery reliability
- **R**adio **F**requency **C**ommunications (RFCOMM)

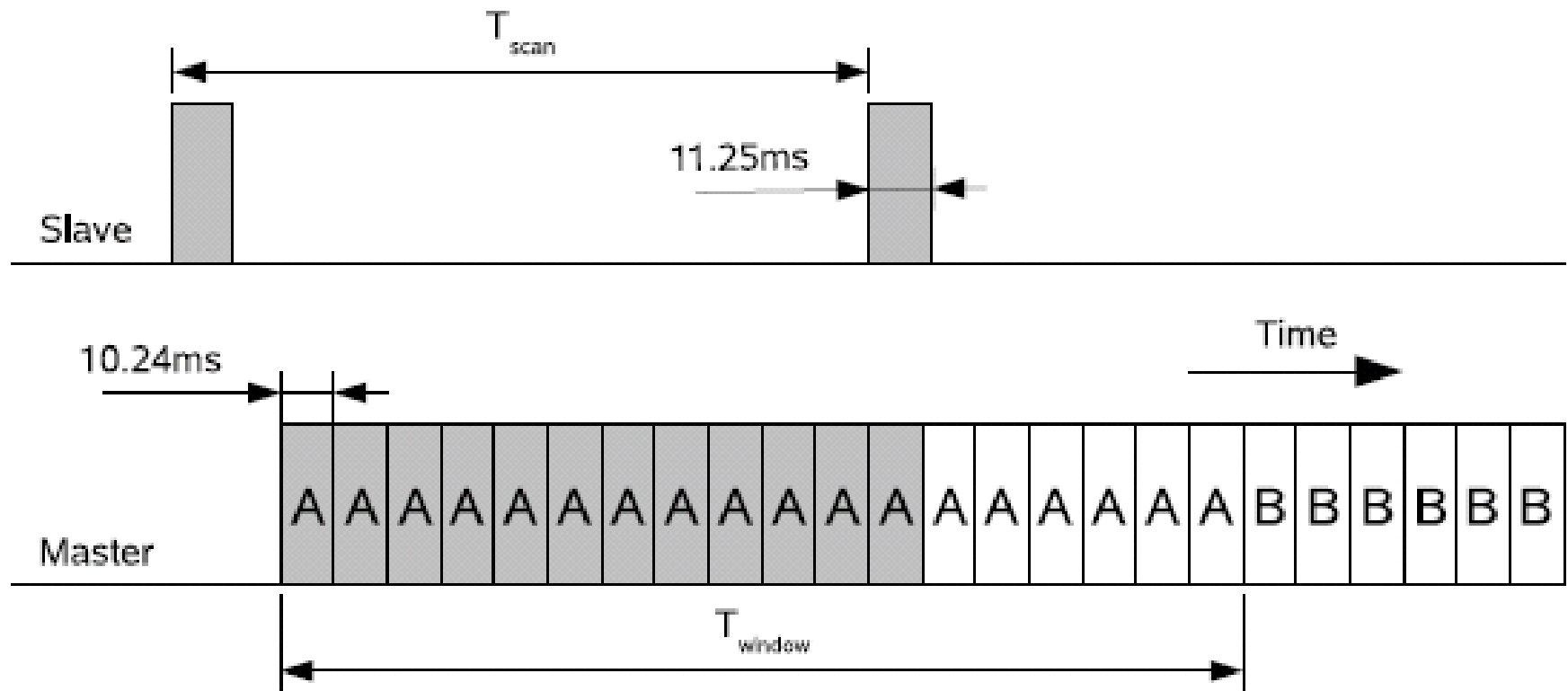
Connection Authorisation

- RFCOMM connections require explicit *pairing*
- Requires human input: impractical to pair every handset with every host
- L2CAP connections almost universally authorisation-free
 - Limited in use for communications, but sufficient for tracking
 - Low-level tasks support continual connection monitoring

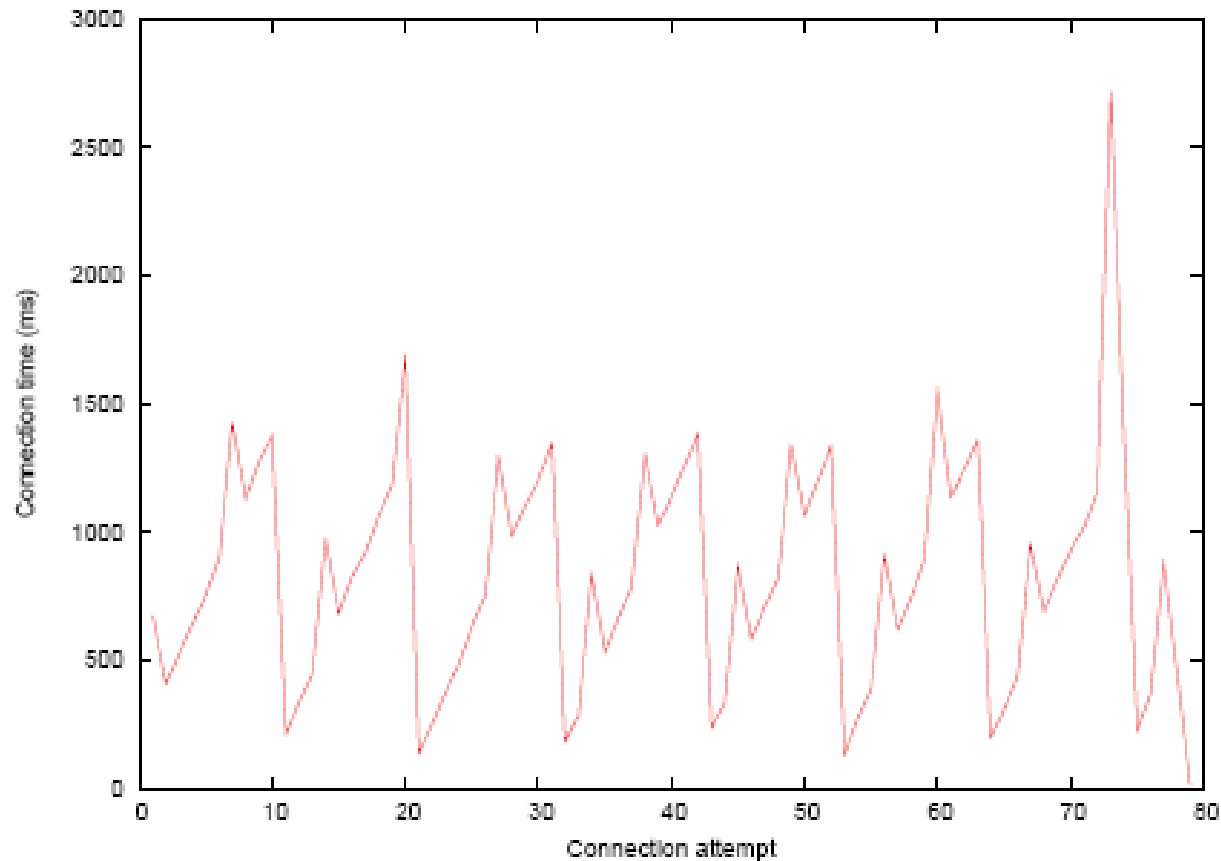
Connection Time

- Expected latencies in tracking dictated by connection times
- Bluetooth uses frequency hopping over 79 channels
 - Sequence derived from device address and Bluetooth clock
 - For two devices to communicate they must have the same hopping sequence and phase
 - To reach this state, Bluetooth defines the *paging* protocol

The Paging Process



Measured L2CAP connection times for iPhone



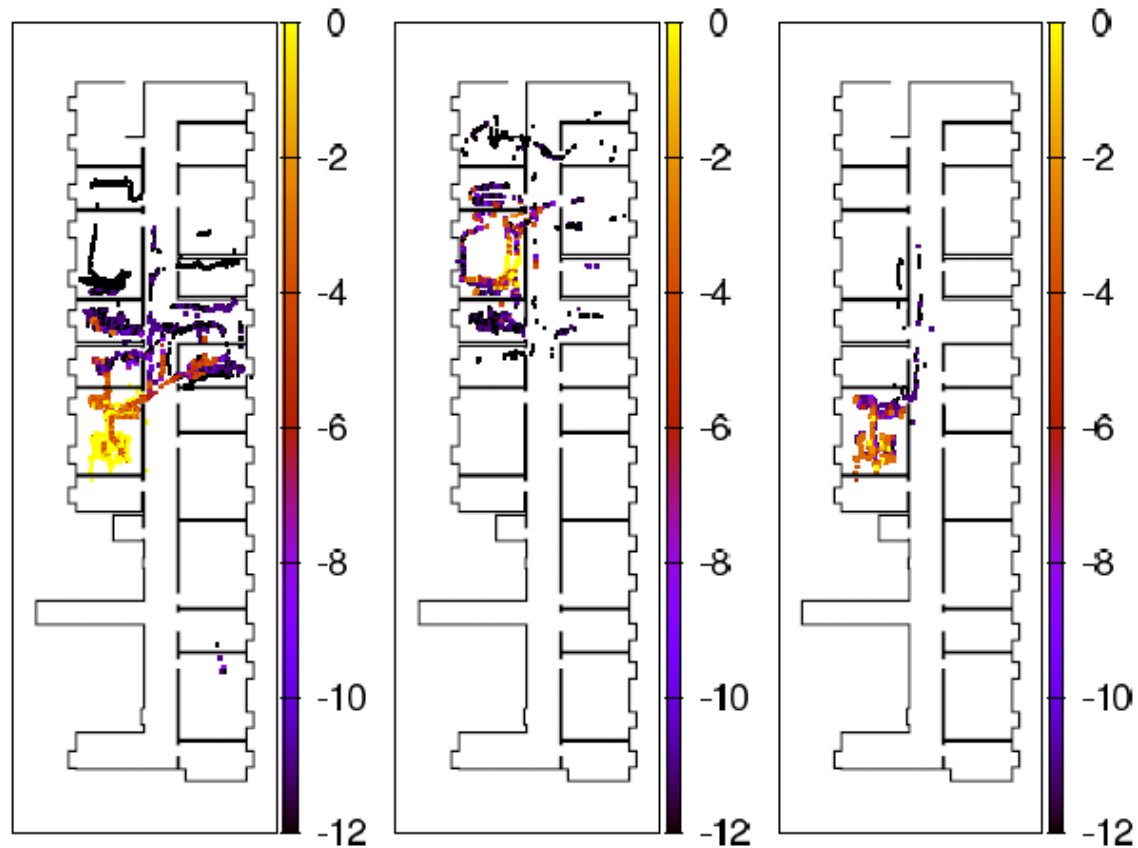
Disconnection Time

- L2CAP
 - Orderly shutdown by either end
 - Severed when out of range
 - Complete disconnection after *LTSO* without communication
 - Not re-established if handset re-enters radio range before *LTSO*
 - *LTSO* should be set to a small value (default usually 20s)
- ACL will disconnect when no higher-layer connections have existed for a certain time (usually 2s)

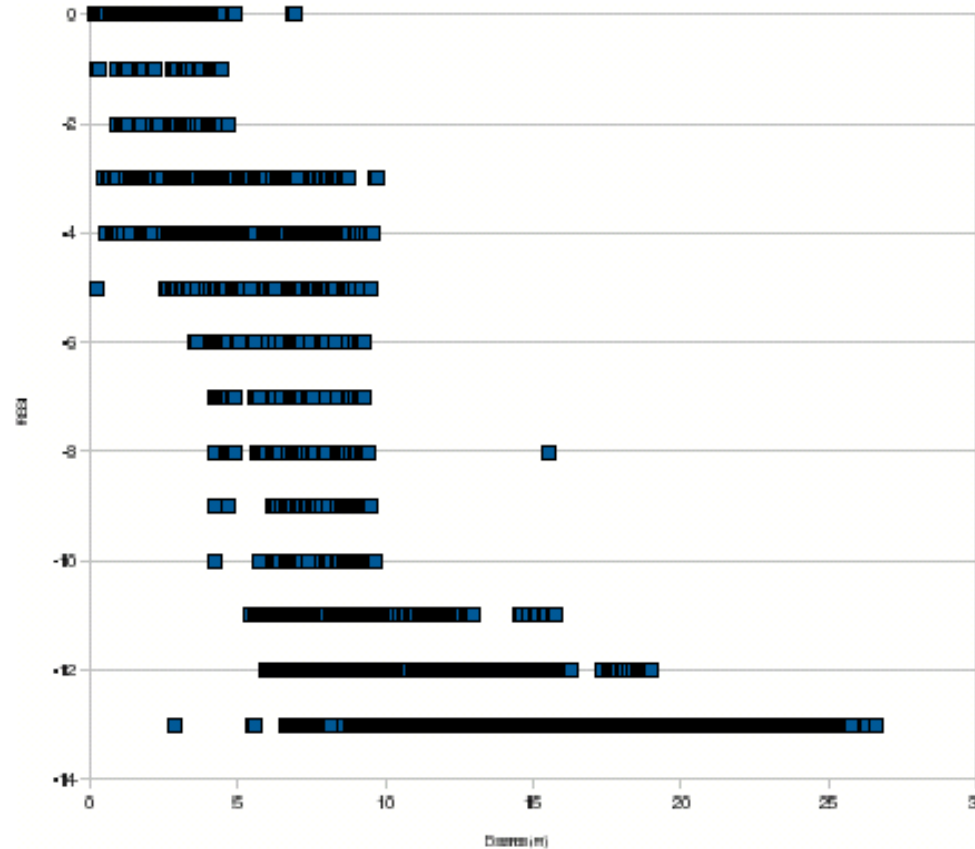
Connection Monitoring

- Forced disconnection indicates two devices are no longer co-located
- We can infer more by measuring connection quality
- Possible metrics:
 - Received **S**ignal **S**trength **I**ndicator (RSSI)
 - Link **Q**uality (LQ)
 - Echo response time

Measured RSSI values from three hosts



Measured RSSI against Euclidean distance



Battery Costs

Test ID	Handset State	G1 (mW)	N80 (mW)
1	Idle	12.81	19.44
2	WiFi connected but idle	170.66	438.08
3	Bluetooth on, discoverable, unconnected	15.45	22.17
4	Bluetooth on, continually being scanned by host	16.07	31.80
5	Continuous Bluetooth echo at maximum rate	321.07	234.43
6	Continuous Bluetooth RSSI measurement	74.97	89.20
7	Bluetooth echo every 30s (with reconnect)	23.76	26.17
8	Bluetooth echo every 20s (with reconnect)	25.19	28.17
9	Bluetooth echo every 15s (with reconnect)	28.02	–
10	Bluetooth echo every 10s (with reconnect)	30.53	42.27
11	Bluetooth echo every 5s (with reconnect)	40.13	50.61
12	Bluetooth RSSI every 30s (with reconnect)	29.93	28.05
13	Bluetooth RSSI every every 20s (with reconnect)	35.86	29.93
14	Bluetooth RSSI every every 10s (with reconnect)	47.59	36.04
15	Bluetooth RSSI every every 5s (with reconnect)	75.72	51.88

System Architecture

- Large building, with large number of fixed hosts in known locations
- All hosts have class 2 Bluetooth devices and networked in some way
- Centralised system
 - Maintains database of handsets
 - Addresses, owners, known properties (T_{scan} etc.)
 - Maintains model of where people are
 - Instructs hosts to monitor handsets

Distributed Clock Offsets

- Important to minimise connection times
- Ensure host always has handset's current frequency in paging train A
- One host: lengthy first connection, then cache Bluetooth clock offset
- Many hosts: distribute clock offsets
 - Use NTP to synchronise system clocks
 - Report two offsets:
 - Between system clock and host Bluetooth clock
 - Between host Bluetooth clock and handset Bluetooth clock

Search

- Simplest level: emulate inquiry-based system
- Use 'home' hosts and connection monitoring techniques
 - Vast majority of workers are sedentary at any given time
 - Only search for users who are moving away from their local host
- Limit the extent of search where possible
 - If home reported very weak RSSI t seconds ago, only hosts within vt can reasonably observe the user now

Bootstrapping

- Constant round-robin polling
 - Cycling through 100 handsets can take >8 minutes
 - Extra load on hosts, reducing update rate
- Out-of-band events
 - *Preferred solution*
 - Computer system events (e.g. associated user logs in)
 - Building security (e.g. user gains access by swiping ID card)

Inverted System

- Handset as master, connecting to hosts as slaves
- No bootstrapping necessary; privacy implications
- More power hungry
- Requires custom software on handsets
- Requires data channel to receive host positions and addresses
- Care needed to ensure hosts not saturated with connection requests

Evaluation of Connection-based Tracking

Advantages

- Faster update rate
- Variable update rate
- No scanning
- Privacy

Drawbacks

- Devices must be known
- Security policies are not universal
- Connection limits

Inquiry-based vs Connection-based Tracking

	Inquiry-based Handset-scans-Host	Inquiry-based scans-hostset	Connection-based
Works on Unmodified Handsets	Most	Some	Almost all
Requires Custom Handset Software	Yes	No	Yes
Deployability	Med	High	High
Requires Discoverable Handset	No	Yes	No
Requires Separate Data Channel	Yes	No	No
Location Update Frequency	~0.1Hz	~0.1Hz	Varies (0-20Hz)
Supports dynamic update rates	<0.1Hz	<0.1Hz	Yes
Scalability	Simple	Simple	Complex
User Privacy	High	Low	Med-High
Handset Power Drain	High	Low	Varies (Med)

Thank you!

Any questions?

Email: sjeh3@cam.ac.uk

More information:

<http://www.cl.cam.ac.uk/~sjeh3/>

<http://www.cl.cam.ac.uk/research/dtg>