Topical Issues: Location Tracking

Part II
Dr Robert Harle

# 1 Location, Location, Location

**Background**

- Many computing eras
  - Mainframe, desktop, networked
- Today the **mobility** era is important
  - Devices more portable but also more *personal*. We carry them around with us and us them to run our lives.
  - They get used in many different *contexts* (communications, camera, navigation, diary, alarm, business...)
  - Adding mobility adds to the context a device must have to respond appropriately: *location* is one of the “next big things” and you'll need to appreciate the location capabilities of devices in the future.
  - But location determination is a hard problem. In this course we'll spend a few lectures looking at the technologies and principles behind today's location systems.

Computing has gone through a series of eras, including mainframe computing, desktop computing, networked computing, etc. There are many innovations in the market right now but probably the overriding theme of the current era is *mobility*. Mobile phones have developed into the most ubiquitous computing platform—there are now over 4 bn phones in the world to the desktop machine’s 1.5 bn.

Initially mobile computing was about making non-mobile systems portable—Powerpoint on the train etc. But as these systems have evolved we’ve added new capabilities, chief among them (almost) ubiquitous networking, and discovered new uses for them.



Mobile devices are much more personal than their desktop counterparts (we don’t tend to share phones for example) and therefore are more of a digital extension of the user. The inherent mobility has fostered interest in location-awareness: knowing *where* someone is gives a strong hint as to *what* they are doing, and allows mobile devices to respond accordingly.

The development of location-aware applications has been possible primarily because of the GPS system—a truly amazing piece of engineering. Almost everyone is familiar with GPS these days. It’s become such an important system that we *rely* on (both for location and time[1]) that you really ought to have an understanding of how it works, and there’s a lecture dedicated to it in this course.

For now it suffices to consider GPS as providing us with ubiquitous location *outdoors* to approximately ten metres. But GPS isn’t perfect—it can be slow to start up, can fail near tall building (‘urban canyons’), and almost never works inside buildings. In this lecture we study the fundamental principles of deterministic location and look at systems that have tried to fill the gaps in GPS.

## 1.1 What Accuracy do we Need?

**What Accuracy Do We Need?**

- Everyone knows about GPS. An amazing piece of engineering that gives us ~10m accuracy outdoors. And completely fails indoors!
- But: even if it worked indoors, 10m is too large to be useful
- Why? Because the scale is different. 10m outdoors is pretty unambiguous – landmarks are separated by much more. Indoors, however, landmarks are much closer. We usually work with:
  - Room-level accuracies
  - sub-metre level accuracies

Before we start looking at location techniques, it is useful to consider what loca-

[1]See the Royal Academy of Engineering’s recent report “Global Navigation Space Systems: Reliance and Vulnerabilities” for an interesting look at just how much we have come to rely on GPS. Scary. `http://www.raeng.org.uk/news/publications/list/reports/Global_Navigation_Systems.pdf`

tion accuracies are appropriate in different contexts. For most GPS applications, the world is considered to be a 2D map of immense size. The landmarks of interest on that map are usually very well separated by tens, hundreds, even thousands of metres. Thus a GPS location with a 10 m accuracy is usually unambiguous and therefore sufficient.

Indoors, however, things are quite different. Buildings have floors so 2D mapping is out. A GPS fix accurate to 10 m (in 3D) isn't so useful here: you can't determine the floor the user is on, never mind the office they are in! Even outdoors we can have issues: two nearby roads that run parallel to each other can cause confusion in a GPS system, or close-by entrances to two buildings.

So the accuracy of location we need depends on the smallest separation we want to distinguish between. The *scale* of the map (which we need to make any kind of useful inference from location data) is what counts. In my experience with indoor location, two different scales are of use:

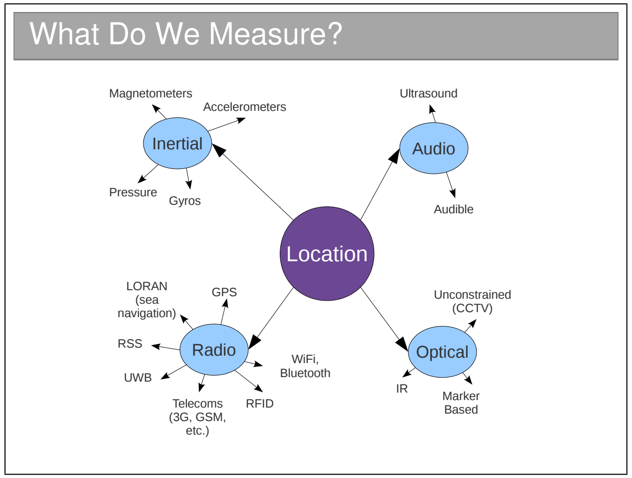
**Room level.** Knowledge of the room we are in (and probably those others we are with) says a lot about our context. Many of the devices we use can be considered on a room scale. For example, we can imagine computers that unlock when we are in the room, phonecalls that route automatically and lighting that responds to our presence.

**Sub-metre level.** We can also imagine more precise location providing room devices with better context. Perhaps a computer should not unlock itself unless I am directly in front of it (not just in the corner of the room), or the phone selection algorithm may wish to distinguish between multiple phones in the room for call routing. Typically, our devices or spaces of influence are separated by a metre or so and thus we need sub-metre accuracies to exploit this context.

### 1.2 What do we Measure?

Very rarely can we just measure 'location' directly (the only example that comes to mind is a tape measure and that isn't great for tracking your car...). Instead we measure whatever we can that can be turned into a distance or bearing and derive location from that.

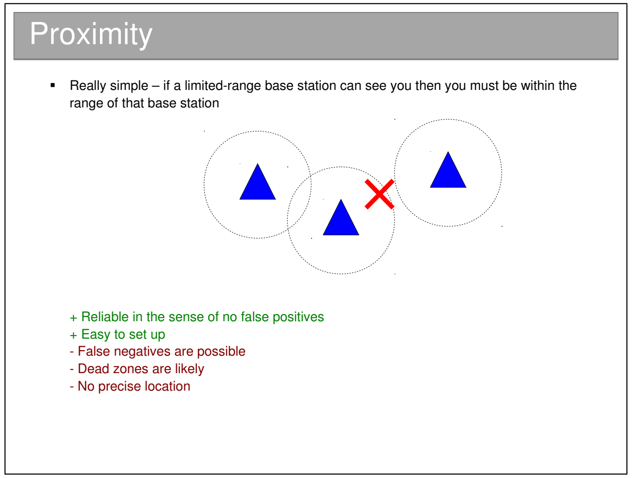





So what exactly do we measure? Over the years we have seen location systems that use a variety of different physical phenomena—a feel for the different media available is in the slide above. They all have their advantages and disadvantages and many can be used in different ways.

Our approach herein will be to look at specific classifications of location system, discussing the underlying principles and giving some concrete examples.

## 2 Proximity-Based Systems



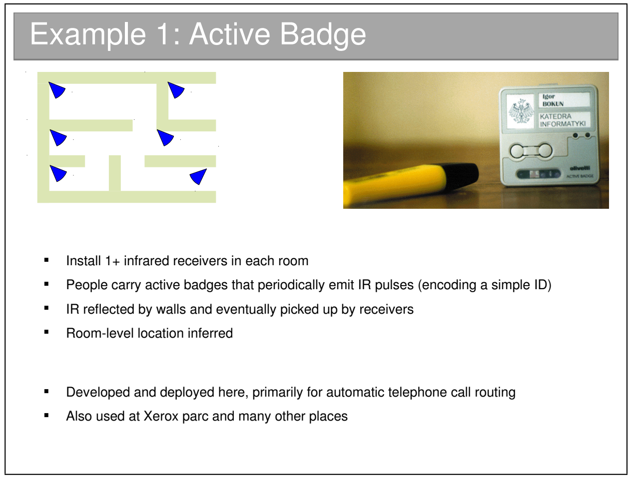
Proximity-based systems are the simplest location systems available to us. The idea is that we have a series of landmark beacons in the world and whenever our device can see one, it is reported as being co-located at that position. A simple example would be a location system based on RFID tags: a network of readers could be installed throughout a building and whenever they spot a tag they locate the associated object at the reader position. Simples!

Of course, the reality of such systems is not so rosy. We want to be able to localise a device wherever it is in a building but getting complete coverage is usually difficult, especially when you consider that the we'd rather the coverage areas for each beacon didn't overlap. And the location results are usually rather coarse: we can get finer accuracy by using beacons with smaller coverage areas (lower power transmitters for example), but then we need a lot more to retain global coverage.

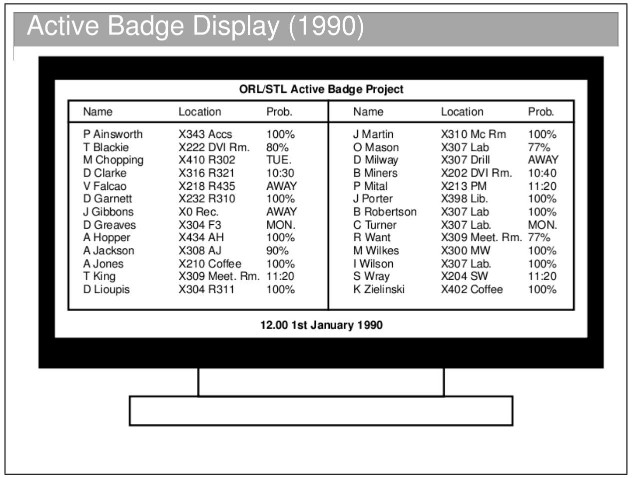
Proximity systems can be based on any medium that can have a limitable, and preferably stable, operational range—infra-red light and radio signals are common choices.

## 2.1 Examples

### 2.1.1 Active Badge (1989–1992)



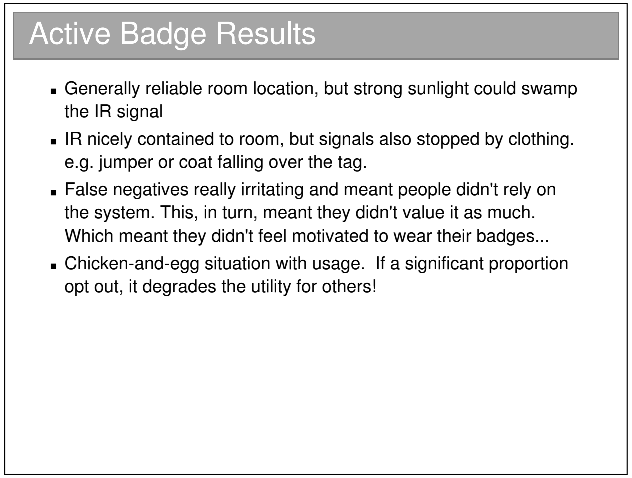



| Name | Location | Prob. | Name | Location | Prob. |
|---|---|---|---|---|---|
| P Ainsworth | X343 Accs | 100% | J Martin | X310 Mc Rm | 100% |
| T Blackie | X222 DVI Rm. | 80% | O Mason | X307 Lab | 77% |
| M Chopping | X410 R302 | TUE. | D Milway | X307 Drill | AWAY |
| D Clarke | X316 R321 | 10:30 | B Miners | X202 DVI Rm. | 10:40 |
| V Falcao | X218 R435 | AWAY | P Mital | X213 PM | 11:20 |
| D Garnett | X232 R310 | 100% | J Porter | X398 Lib. | 100% |
| J Gibbons | X0 Rec. | AWAY | B Robertson | X307 Lab | 100% |
| D Greaves | X304 F3 | MON. | C Turner | X307 Lab. | MON. |
| A Hopper | X434 AH | 100% | R Want | X309 Meet. Rm. | 77% |
| A Jackson | X308 AJ | 90% | M Wilkes | X300 MW | 100% |
| A Jones | X210 Coffee | 100% | I Wilson | X307 Lab. | 100% |
| T King | X309 Meet. Rm. | 11:20 | S Wray | X204 SW | 11:20 |
| D Lioupis | X304 R311 | 100% | K Zielinski | X402 Coffee | 100% |

The Active Badge system used small, powered (i.e. active) tags that were worn as badges. Each badge had a unique identifier and would periodically (0.1 Hz) transmit it over an infra-red (IR) channel.

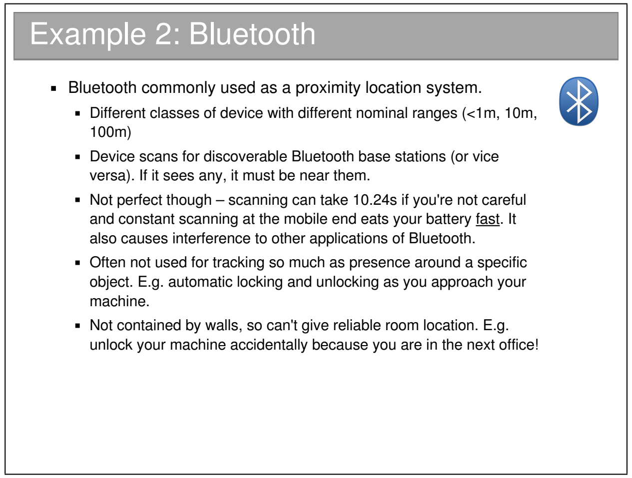
Networked IR receivers were put up, roughly one per office. The great thing about IR is that it will bounce all over the room before it dissipates (you know this from TV remotes) and doesn't penetrate the walls. This means we get natural room containment and so we can reliably associate users with rooms. It does mean we need at least one receiver per room, but we are not particularly sensitive to where that sensor is sited.



The system was extensively deployed and was arguably a great success. Occasionally there were problems in strong sunlight (which would overpower the sensors) but generally the system worked well and was popular.

You might reasonably ask why we are not all carrying around Active Badges or their descendents these days. On reflection, I think it was ahead of its time and suffered for it. When it was demonstrated, the technology was only just capable of supporting the system. A redesign now would make the tags smaller, cheaper and with much longer battery lifetimes. Remember that it was developed in a world where global networking was still around the corner and location-awareness had never really been considered. Had it been invented today, I suspect it would be much more successful.

### 2.1.2 Bluetooth




Bluetooth is ubiquitous on modern mobile platforms, which makes it rather attractive for locating someone. The simplest approach is to use a proximity system: if a base station can see you, then you must be within range of it. Software like BlueProximity will do this for you.

Now, Bluetooth comes in three flavours, which have different transmit powers and hence different ranges. Nominally we have:

| Class | Max Power | Range |
|---|---|---|
| 1 | 100mW | 100 m |
| 2 | 2.5 mW | 10 m |
| 3 | 1 mW | 1 m |



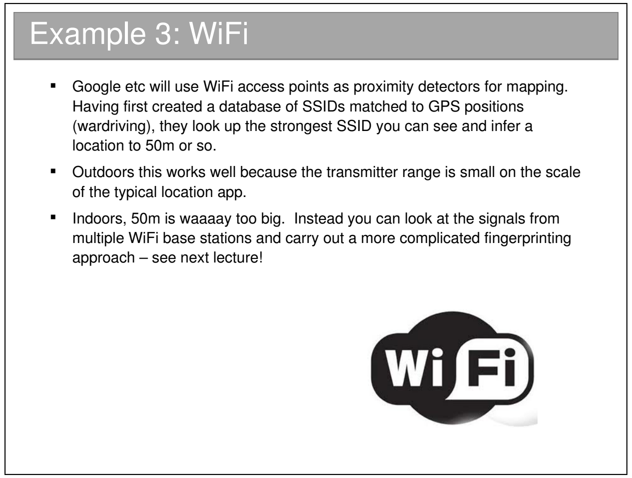
It depends on the scale of your problem as to which you want to use. If it's indoors, locating someone to a radius of 100 m probably doesn't even get you the building they're in! A radius of 10 m is better, but it's really only going to get you 'portion of a building' accuracy (it will be hard to pinpoint people to specific rooms because the radio penetrates the walls). You'll also be needing lots of Bluetooth hosts to cover an entire building...

In reality, creating a Bluetooth tracking system isn't trivial. The simplest method that works for any device is to leave the mobile device discoverable and have every host constantly scan for in-range devices. This is generally bad because:

- Bluetooth discovery sucks. To discover all the devices in range, you may need to let each discovery query run for 10.24 s (this is all to do with power saving at the mobile end). It means you get a pretty awful update rate for tracking.
- Discoverability is considered a security risk.
- Most modern phones/devices won't even *allow* you to leave discoverability on indefinitely because of the previous point!

Nonetheless, some have had success. The website http://www.bluetoothtracking.org leaves a scanning station next to a highway, and reports that it sees 3,200+ handsets an hour at peak times! This isn't so much tracking as an instantaneous measure of position, but it tells you something about what's out there...

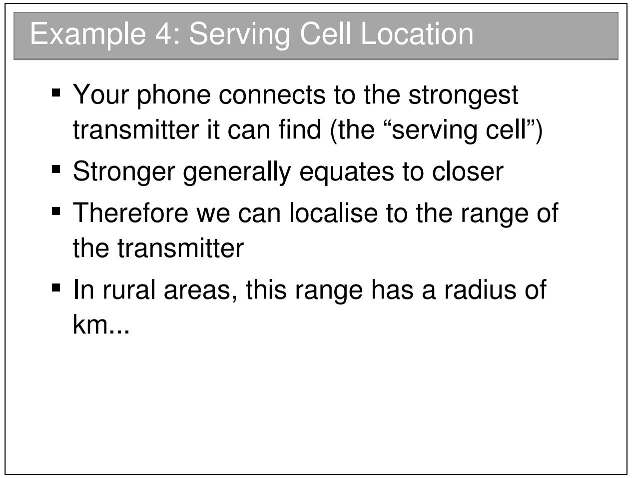
### 2.1.3 Example 3: WiFi



WiFi is a little better on these counts, since security is stronger and discovery is comparably fast. But WiFi is very power hungry compared to Bluetooth (as any smartphone user will tell you). So not everyone will be happy with having it permanently on and scanning for base stations.

As it turns out, WiFi generally has a pretty big range, and we often get access points with overlapping coverage. In this case, we can use *fingerprinting* to locate with more accuracy—we return to this later in the course.

## 2.2 Example 4: Serving Cell Phone Location




The mobile phone network is designed such that your phone talks to the strongest base station that it can hear at any given time—this base station is known as the *serving cell* of your handset.

For mobile telephony networks, the strongest station is almost always the nearest. Therefore, the network operator can localise any phone to within the range of its serving cell—proximity-based location!

How accurate it is depends heavily on the serving cell and its location. In rural areas, cells are sparse and so their coverage is very large (many km), but also more predictable (free space propagation). In built up areas, lower power (and hence smaller) cells are often used in a more dense distribution, giving better accuracy. It is in the interests of the network provider to have ubiquitous coverage so holes are rare. We are still talking many tens or hundreds of metres of accuracy, though.



# References


[1] G. Borriello, W. Brunette, M. Hall, C. Hartung, and C. Tangney. Reminding adbout Tagged Objects using Passive RFIDs. In *Proceedings of the 6th International Conference on Ubiquitous Computing (UbiComp 2004), Nottingham, UK*, September 2004. An RFID system where a mofified watch alerts the user to any forgotten items as he leaves an area (all items are RFID-tagged).

[2] D. Hahnel, W. Burgand, D. Fox, K. Fishkin, and M. Philipose. Mapping and Localization with RFID Technology. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA '04)*, 2004. A paper from about the use of RFID tags in mapping and localization.

[3] A. Harter and A. Hopper. A distributed location system for the active office. *IEEE Network*, 8(1), 1994. An evolution of the Active Badge system that looks at making the system distributed and scalable.

[4] Simon Hay and Robert Harle. Bluetooth tracking without discoverability. In Aaron Quigley and Tanzeem Choudhury, editors, *LoCA 2009. LNCS*, volume 5561, Heidelberg, 2009. Springer. A paper to be presented in a few weeks that describes a wide area Bluetooth proximity system that does *not* need discoverability to work.

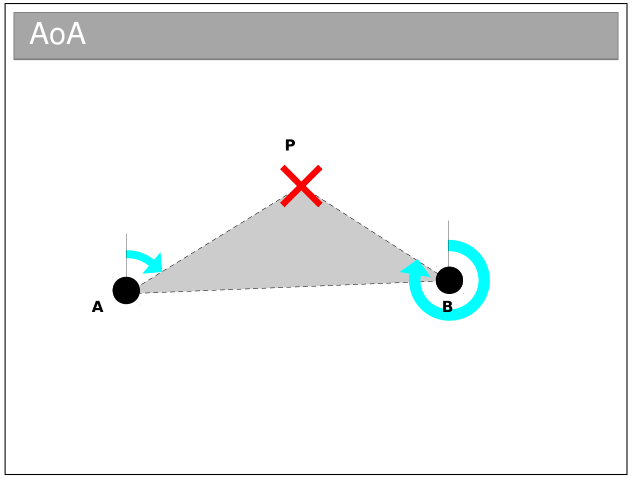
[5] N. Nasser. Automatic location systems for mobile phones. pages 59–64, May 2008. A good general intro to tracking mobile phones.

[6] L. M. Ni, Y. Liu, Y. Cho Lau, and A. P. Patil. LANDMARC: Indoor Location Sensing using Active RFID. In *Proceedings of the First International Conference on Pervasive Computing and Communications*, March 2003.

[7] BlueProximity Project. `http://blueproximity.sourceforge.net/`. Designed to lock or unlock your machine as you walk away or approach, based on whether it can sight your phone using Bluetooth.

[8] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, January 1992. The original Active Badge system as developed at ORL.

# 3 AoA Systems



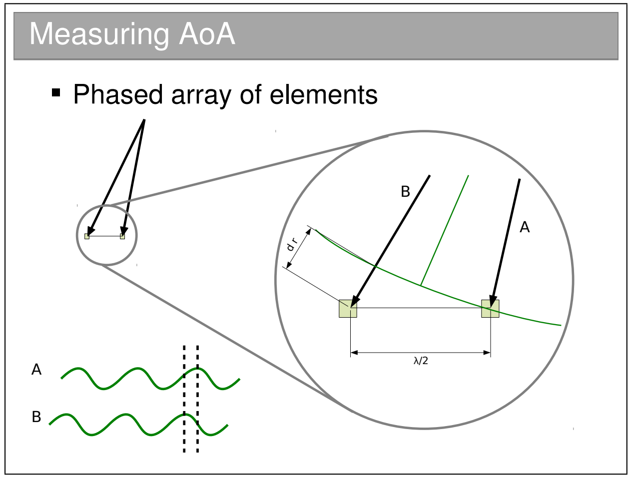

Most people think of the word 'triangulation' when they are asked to compute a position. Thing is, not many people seem to understand what it means and, like chinese whispers, it has ended up with a lax and unclear definition.

We're going to take a reasonably strict definition. 'Triangulation' applies to so-called 'Angle of Arrival' (AoA) systems where we can somehow measure the angle of incidence of a target's signal.

The principle is simple. Take two measurement stations at $\mathbf{A}$ and $\mathbf{B}$ and an object to be located at $\mathbf{P}$. Assume the object is transmitting in some way (usually but not always radio). The two stations measure the incident signal angle and form a triangle in 2D space based on the two bearings and $\mathbf{A}$ and $\mathbf{B}$. The third triangle vertex will be at $\mathbf{P}$.



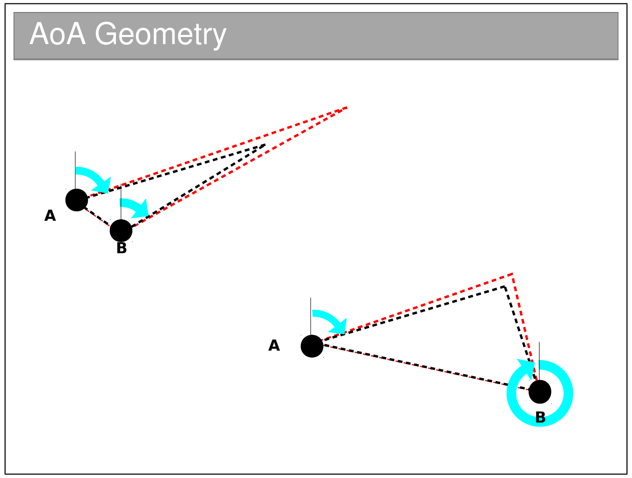
## 3.1 How can you measure the AoA?




Typically we would use an antenna array and measure the phase difference between signals received at different elements. For example, consider a two-element array. As the bearing to the source changes, the signal hits each element at slightly different times due to the slightly different path lengths (say, a difference of $dr$). This means the signals at any any two elements will have a different phase if the path difference between them is not a whole number of wavelengths.

If we choose the element separation to be one-half of the wavelength $\mid dr \mid$ has a maximum of half the wavelength (making the signals exactly out of phase). This occurs only when the array is parallel to the bearing to the source. All measured phases in between will imply a different transmitter bearing. Hence we measure the phase difference (e.g. by letting the two signals interfere and analysing the resulting pattern) and from that derive a bearing estimate.

## 3.2 Siting the Stations

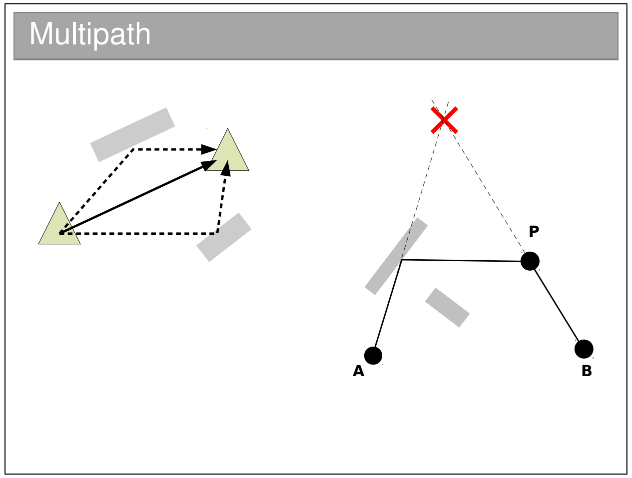



Every measurement has error associated with it, so it's interesting to think about how sensitive this approach is to errors in the bearings. This is all about the geometry.

Consider the vectors from the base stations to $\mathbf{P}$. If these vectors are near parallel (similar bearings), the triangle must be very tall and thin. A small error in bearing gives a big change in the estimate of $\mathbf{P}$. Conversely, if the vectors are near perpendicular, a small bearing error doesn't have so great an effect on the estimated $\mathbf{P}$. Therefore the geometry of our stations relative to our source is very important—ideally a line drawn between them should intersect at the true target and the closer we get to this setup the better. i.e. make the triangles are wide and short as possible.



## 3.3 Multipath




In the real world there's another problem to worry about: *multipath*. This is the term used to indicate that a signal propagates from source to receiver via multiple paths, usually due to reflections.

In an AoA system, this is a major problem if the direct (*line-of-sight*) path doesn't get through, but a bounced signal does. This means the bearing is all wrong and we get garbage in our location calculation.

How do we address this? We put redundancy into our system. In 2D we only need two bearings in principle, but we actually use as many as possible. We are then applying a *multiangulation* approach whereby we process an over-determined system, looking for consistency in the data.

For example, we might take six bearings, five of which mutually agree and one which does not—we can throw out the latter as multipath. A typical way to do this is to compute a location from all bearings (perhaps using an iterative non-linear model) and then evaluate the residuals (errors) for each bearing. In the case described the five correct bearings should pull the estimate towards the true position with five units of influence and the bad signal away from it with only one unit of influence. We would therefore expect the estimate to be closer to the true position than not, and the residual for the bad signal would be the highest. We throw out the highest residual measurement and reprocess, repeating this until either the overall error estimate is below some acceptable bound, or we have too few bearings left to compute a location (a failure).

If we can, we distribute the measurement positions *around* the source to minimise the effects of bearing errors.

### 3.4 Example: Pirate Radio and Enemy Transmitters

AoA systems are usually used whenever you want to locate the source of a transmission over which you have no control and you don't have the luxury of having a permanent set of listening stations surrounding its approximate position (if you do you can use TDoA—see later).

Say you are trying to localise a radio transmission in the desert from your helicopter (you *do* have a helicopter, right?). You take a bearing to the signal from wherever you are (as determined using GPS). Then you fly perpendicular to that bearing for a bit and take a new bearing at a new position. You can now estimate the location of the transmitter. To improve your estimate you should repeat the process until you have enough bearings in agreement to pinpoint the transmitter.

You can use a similar approach to locate a pirate radio transmitter. First you tune to the station and get a bearing to the source. Then you move around a bit and repeat. You use the two measurements you have to very roughly estimate the transmitter location and then move to another location that gives you the best geometry to pinpoint it more accurately with AoA.

## 4 ToA or ToF systems

**ToA Systems**

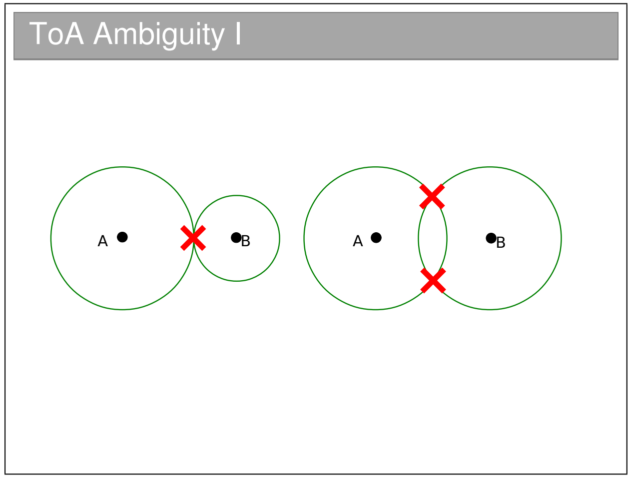
- Measure how long it takes for a signal to propagate from some know place (a base station) to the target
- Convert the time difference to a lateration (distance) measure using its known speed
  - This defines a radius about each base station, on which the device must lie

Our next class of location system is a *Time of Arrival* (ToA) or equivalently a *Time of Flight* (ToF) system. The idea is that we somehow measure how *long* it takes for a signal at the source to reach a set of receiver stations at known locations.

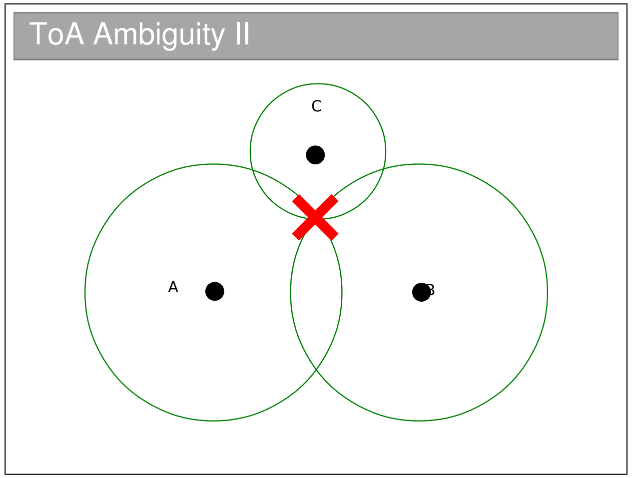
Times don't help us much directly, so we convert them to distances on the assumption that we know the speed at which the signal travelled. Again we are looking to form a triangle to get a position, but instead of having the angles we now have the triangle side lengths. Computing the source position from this information is *trilateration*.

Imagine our stations are at $\mathbf{A}$ and $\mathbf{B}$, and a signal (propagating at speed $c$) arrives at times $t_A$ and $t_B$, respectively. Then we know:

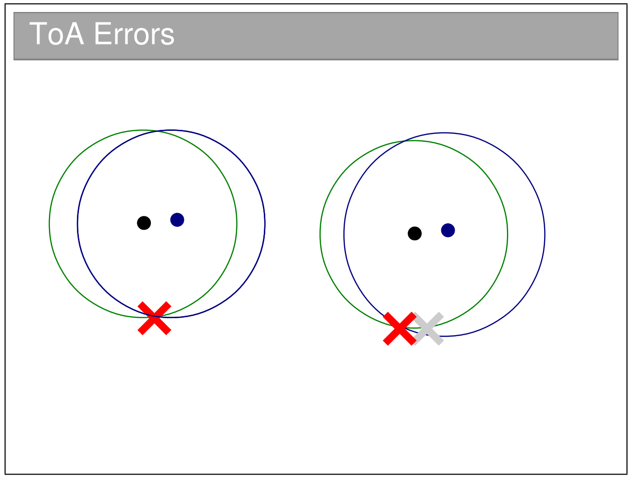
$$|\mathbf{P} - \mathbf{A}| = ct_A \quad (1)$$
$$|\mathbf{P} - \mathbf{B}| = ct_B \quad (2)$$




You can think of this as intersecting circles of set radii centred on the station locations (the above equations each describe a circle in 2D). The problem is that this almost always gives ambiguity in position. To solve this, we must ensure that we have a minimum of three measurement stations and, as before with AoA, an overdetermined system is even better.




## 4.1 Noise and Geometry

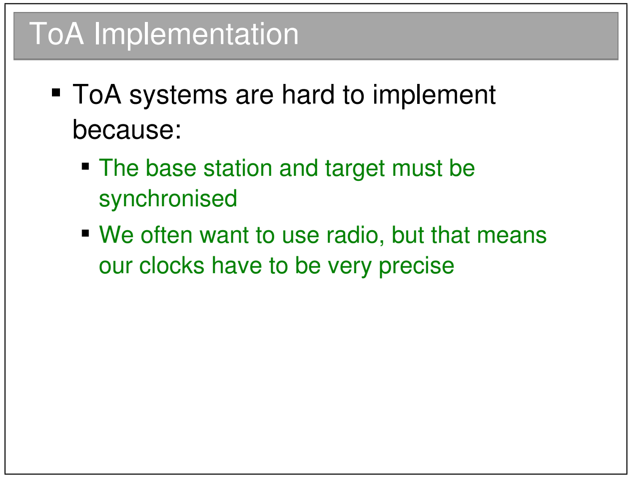



What about error in the timings? As with angulation, the magnitude of the error is dependent on the geometry of the receivers. The ideal geometry for 2D has three receivers, each at a vertex of an equilateral triangle that contains $\mathbf{P}$.

Problems arise when the vectors between the receivers and the source are close to parallel. If this happens, a small change in circle radius will have a relatively big effect on the intersection points.



## 4.2 Synchronisation and Timing




ToF systems are simple to understand but often a pain to implement. The problems stem from the need to measure the travel time of a signal, which comes with at least two issues:

**Station Sync.** The receiving stations have to start their virtual stopwatches at exactly the same moment as the source emits its signal. The problem is that synchronising clocks across multiple sites with sufficient accuracy is *not* easy, especially when we're using signals that travel at the speed of light!

**Timing Accuracy.** Assume we are dealing with radio waves propagating at the speed of light. If we want our timings to give distances within just 10 m of the correct value, we need to be able to time to accuracies of 10/300,000,000 = 33.333 *nano*seconds. That's expensive kit...

Even when you have an accurate clock, when do you trigger the timing measurement at the receiver? In the real world we never get a perfect pulse to trigger from, and the channel almost certainly has noise that could mistakenly trigger it if we're just naïvely using a threshold.

Instead, we can rely on the same pulse profile being received at each receiver, just at different times. We sample the received signals at high rate and then *cross-correlate* the results to figure out the timing difference:

$$[f \star g]_i = \int_{j=-\infty}^{\infty} f_j g_{i+j} \tag{3}$$

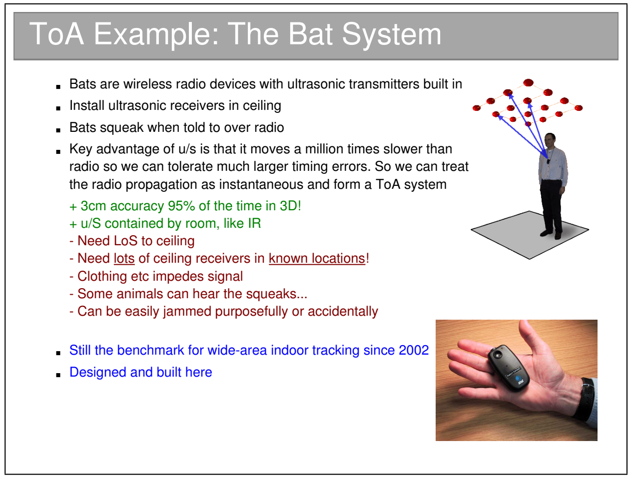
where $f$ and $g$ represent the sampled signals at two different receivers.

## 4.3 Multipath

Just like with angulation, multipathed signals cause headaches. They cause measured ToF values to be larger than they should be (never smaller) and thus cause distances to be overestimated.

The solution is the same as with angulation, however. *Multilateration* uses ToF readings from more than three spatially distinct receivers to estimate the position and to throw out outliers (multipathed signals).

## 4.4 Example: The Bat System




The Bat system (originally "Active Bat System") came out of a CUCL Ph.D. by Andy Ward that was developed by AT&T Research Cambridge. It is a ToA system that times ultrasonic pulses from small, wearable devices ("Bats") to a set of wired receivers in the ceiling. A radio system (433 MHz) polls a Bat, telling it to send an ultrasonic pulse. To get a position, the system simultaneously starts a clock and polls the Bat being located via the radio. The Bat responds by sending an ultrasonic pulse, which is heard by some subset of the ceiling receivers. Each receiver that hears a pulse notes the time it heard it. When these times are collated, a multilateration algorithm is used to estimate the Bat position.

The choice of ultrasound is important for a number of reasons:



**Easy Synchronisation.** The great thing about ultrasound is that it moves approximately a million times slower than radio waves do (330 m/s vs 300,000,000 m/s). If the system starts a clock and polls a Bat 30 m away, the Bat receives the instruction after $10^{-7}$s. If we ignore this transmit time altogether, we only introduce a lateration error of 0.000033m. In essence, we can get away with treating the radio propagation as instantaneous.

**Easy Containment.** One of the nice properties of IR for the Active Badge was that it is naturally contained within bounded spaces (rooms). Ultrasound has this property too, so we can pinpoint the correct room without even having enough data to trilaterate!

But beware: whereas IR bouncing all over the room was good news for the Active Badge, ultrasonic reflections are potentially bad news for the Bat system since the receivers will see multipathed signals too. Each receiver reports only the *first* pulse it hears—we hope that this is from the direct path, or that a multilateration algorithm will have enough good measurements to discard it if not. The thinking behind putting the receivers in the ceiling was to make multipath less likely—by making the ultrasonic emitters of the Bat point upwards, and wearing them at chest height, there will usually be direct paths to the ceiling receivers.

### Performance

- The Bat system achieves **3cm accuracy** *in 3D space* 95% of the time!
- The position update rate is variable, with a maximum of around 15 Hz. This is a nominal value chosen to ensure that each ultrasonic pulse has fully dissipated before the next is sent.

**Deployment** The Bat system was deployed across three floors of the old AT&T Research building near Engineering. It was also deployed in a single room in engineering, and subsequently along the entire length of the SN corridor in the WGB (i.e. the DTG area). Today, it still runs in the WGB and is used for location research (usually as a ground truth for other systems).

**Issues** The Bat system is arguably the most accurate large-scale person-tracking in existence, but it isn't perfect. One of the problems with a system that can potentially achieve cm-precision is that *the accuracy to which you can locate your receivers becomes a limiting factor*! We want to get the receiver locations measured with an accuracy that is an order of magnitude smaller than the expected location accuracy.

That means location to a few millimetres across hundreds of square metres. Good luck with that...

In the current deployment we used laser surveying stations (the type that architects use) and went to great effort. Realistically we probably measured to within 15 mm. Over time, however, receivers are bound to move or be knocked and that accuracy has doubtless faded.

The next issue concerns the number of receivers. Ultrasonic containment is nice on one hand, but means that wherever you face in a room, there must be at least three receivers in the ceiling to get a position fix. That means you need a lot of receivers (all accurately positioned!). The 550 $m^2$ deployment in the WGB (that's 23 rooms/corridor areas) uses a whopping 409 receivers, all carefully surveyed..!

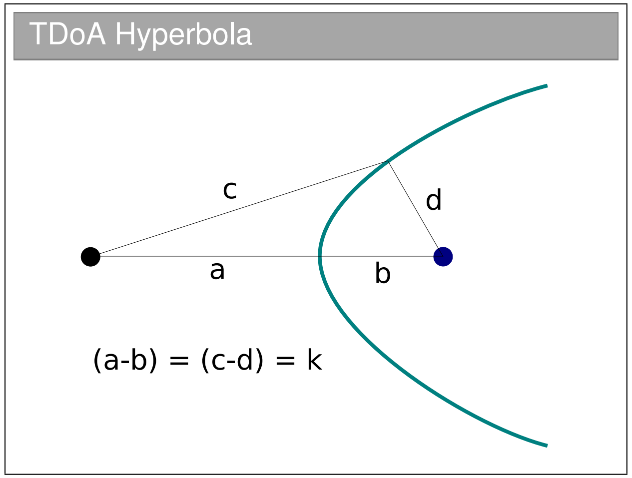
# 5 TDoA Systems

Synchronisation is a big issue in ToA systems—to time how long it takes for a signal to propagate between two points means that we have to have a clock at each site and those clocks must be synchronised. In the real world, we're pretty good at synchronising two systems when there's a reliable piece of wire between them (think NTP and better).

For a location system, we have a problem. The locatable device needs to be mobile (or you don't need a location system!) and mobile devices won't have the serious hardware you need to synchronise two systems together to nanosecond precision over radio. So, pure radio systems can't realistically synchronise the mobile node with the receivers, and ToA won't therefore work.

Instead we can synchronise our receivers together (usually with bits of wire) and use a *Time Difference of Arrival* (TDoA) system to handle the fact that we can't know precisely when the mobile node transmits.

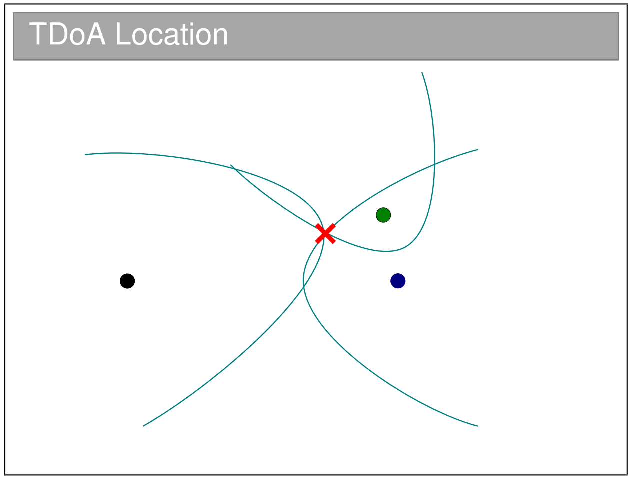





The method is best illustrated by example. Take our stations $\mathbf{A}$ and $\mathbf{B}$ and assume they log the same signal at times $t_A$ and $t_B$ (note these times are in the same frame of reference, such as GMT, but are *not* the ToF values for the signal because we don't know when the signal was sent). If we assume $t_B > t_A$ then we can state that station B is $c(t_B - t_A)$ further away from P than A. i.e.

$$| \mathbf{P} - \mathbf{B} | - | \mathbf{P} - \mathbf{A} | = c(t_B - t_A) \quad (4)$$

This is actually the definition of a hyperbola with centre at the midpoint between the two stations. i.e. From a pair of stations we can restrict $\mathbf{P}$ to lie on a hyperbola in 2D space despite not knowing when the transmission began.



Now all we do is look at multiple pairs of receivers to derive multiple hyperbolae and look for the intersection in the same way as we did with circles in a ToA system.

Note that we need three pairings to get three hyperbolae for an unambiguous 2D fix, but that we can get these from just three base stations since the pairings need not be completely independent.

Additionally, there is no requirement for the measurements for all pairings to be derived from the same signal. We might get the A-B pairing one second, and the B-C the next. This is fine, so long as the object isn't going to move significantly in the time it takes to collect all of the pairings you want.

## 5.1 TDoA Example: Phone Tracking

Hollywood would have you believe that you can be tracked so accurately through your mobile phone that they'd know if you tripped. Fortunately (or perhaps unfortunately?) they can't. But it's interesting to know what *can* be done.

Firstly, you need to understand some terminology: the network operator has a series of *Base Transmitting Stations (BTSs)*; your phone is a *Mobile Station (MS)*. GSM communication uses Time Division Multiple Access (TDMA) i.e. there are set time slots during which only one phone talks to the BTS.

How does an MS synchronise with a BTS so that it talks at the right moment? Each BTS regularly sends out a *synchronisation burst* which the MS can 'lock on' to. This means the BTS and MS are reasonably well synchronised (actually the BTS buffers the timeslots to allow for small sync errors), but not well enough for location estimates. Note also that any two BTSs are *not* synchronised.

**U-TDoA Phone Location**



### TDoA Example: Phone Tracking

- Radio masts are known as Base Transmitting Stations (BTSs) and they are **NOT** synced together
- **U-TDoA** ($\rightarrow$ normal TDoA)
  - Phone transmits a signal heard by different BTSs
  - They record when in their local clock frame
  - Have to deploy special GPS devices called Location Measurement Units (LMUs) at the BTSs in order to get a common time reference
  - Then we can compute the time difference between arrival at BTS1 and BTS2, etc.
  - Intersect hyperbolae $\rightarrow$ position

Uplink Time Difference of Arrival (U-TDoA) has been adopted by all the major US phone providers in response to the E-911 government mandate there (this is a law that requires a mobile phone to be locatable to various accuracies when the emergency 911 number is called).

It is basically standard TDoA on a mobile phone signal, except that we need to augment the BTSs with some kit to sync them up. This kit is deployed by the operator and is called a *Location Measurement Unit (LMU)*. To save money, operators usually deploy LMUs at only a subset of their BTSs (the more the better as far as location accuracy goes).

Each LMU monitors the signals received by the attached BTS and uses GPS to timestamp them in a global time frame. To position, the primary LMU for a given MS (usually just the closest) collects the receive times from LMUs using the data network, computes time difference pairs and thus a location using TDoA.

**E-OTD Phone Location**

## TDoA Example: Phone Tracking II

- **E-OTD** (Enhanced-Observed Time Difference)
  - The phone measures differences between the corresponding arrivals in its own timeframe
  - This gives us a TDoA value. Collect enough and we can position as usual
  - BUT we can also make the processing easier by ensuring that every pair we derive a TDoA from contains the same BTS. Then w have a load of time differences (=distance differences) from one specific BTS. This means only one unknown in the system...

A Cambridge company (Cambridge Positioning Systems, now part of Cambridge Silicon Radio) developed a technique known as Enhanced-Observed Time Difference (E-OTD) which is really a kind of constrained TDoA. The main change is an inversion of the system so that the MS measures the time differences between the BTS bursts it hears. This is further complicated by the fact that we must assume that two BTSes will not transmit at the same time.

So how do we make this work? The first piece of information we need is the transmission times for each BTS in a global time frame. We do this by sticking an LMU somewhere in the system. This LMU computes its position using GPS, retrieves the BTS positions from a database and then listens for the same pulses that the MS does. From this information, it can figure out the time that each BTS actually transmitted its pulse. So, BTSs A, B and C might transmit at absolute times $t_a$, $t_b$ and $t_c$.

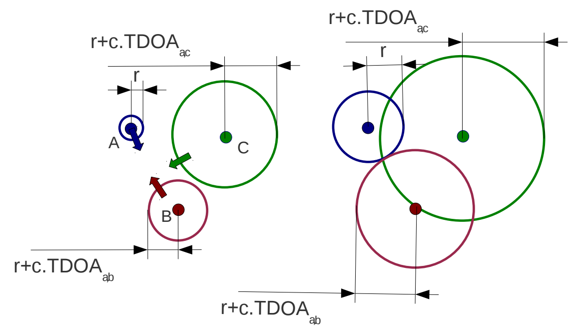
Meanwhile, the MS hears the same signals and, not having a global time reference, measures the differences in the reception times relative to any reference BTS with an LMU attached (say, $A$): $\Delta t_{a,b}$, $\Delta t_{a,c}$.

We can now compute the TDoA values:

$$TDoA_{a,b} = \Delta t_{a,b} - (t_b - t_a) \quad (5)$$
$$TDoA_{a,c} = \Delta t_{a,c} - (t_c - t_a) \quad (6)$$



## TDoA Example: Phone Tracking III




At this point we could compute $TDoA_{b,c}$ and apply the usual TDoA calculation. This would work fine, but turns out to be slight overkill. This is because we have implicitly tied all the TDoA time difference measurements to include a specific BTS (the reference one with the LMU). Therefore we can apply a different analysis which *may* be easier to understand/easier to implement:

- The only real unknown here is the time it takes for a signal to get from $A$ to the MS.
- Let's just set this arbitrarily to $r$. As with TOA, we can draw a circle around $S$ with radius $r$
- Now we can draw circles around $B$ and $C$ with radii $r + c.TDoA_{a,b}$ and $(r + c.TDoA_{a,c})$, respectively.
- If we vary $r$ until the three circles meet, we have our location!

### 5.1.1 Comparison

**U-TDoA vs E-OTD**

- U-TDoA
  - Positioning at the server → bad for privacy but more attractive to operators
  - Works on all handsets without modification
  - Can increase accuracy by deploying more LMUs
  - Needs lots of LMUs → expensive
- E-OTD
  - Positioning at the handset → privacy
  - Requires software on the phone
  - Fewer LMUs

- In principle, E-OTD needs fewer LMUs deployed which means lower deployment costs. U-TDoA is very expensive to deploy.
- U-TDoA can deploy more sensitive receiving equipment on its LMUs and thus more BTSs will hear the phone than vice-versa.
- U-TDoA typically achieves sub-80 m accuracy and can use 40+ BTSs per position (greater redundancy gives greater accuracy). E-OTD typically uses around 8 BTSs per position and achieves accuracies closer to 150 m.
- E-OTD only works on modified handsets, U-TDoA works on all.
- E-OTD accuracy is dictated by the handset capabilities (clock, processing, etc.). U-TDoA can use more powerful, bulky equipment.
- E-OTD requires the active participation of the handset so has a natural privacy-preserving mechanism. U-TDoA can be performed without the MS owner knowing.



**Which won?**

- U-TDoA is the operators choice. Strong backwards compatibility and the ability to track any active phone without consent – who wouldn't want that..!

Many US operators adopted E-OTD a few years back, but then decided that it couldn't reach the accuracies that it had to reach for the FCC e911 mandate. The result is that most US operators have now coughed up and use U-TDoA.

Note that both E-OTD and U-TDoA struggle in the same 'urban canyons' that GPS struggles with (for the same reasons).

# 6 Concluding Remarks

**Conclusion**

- We've only just scratched the surface of location determination
- Next time we'll look at a non-deterministic location technique that is becoming very popular...

In this lecture we've looked at some of the principles of deterministic location determination, including proximity, AoA, ToA and TDoA. We've really only scratched the surface, however. In a later lecture we will look at a different approach entirely, which changes the problem into one of pattern matching: fingerprinting.

It's also worth noting that the techniques we look at in this course are generally one-time position fixes. You can do a lot more by feeding such information into a filter that intelligently *tracks* you from one position to the next, taking account of your motion abilities (top speed, etc.) and the constraints in the environment (walls, furniture, etc.). We won't go into any detail on this, but if you find yourself doing this sort of thing, you should look up the use of Kalman filters and Particle filters for location tracking.