IT University of Copenhagen

Mesh Mobs
Virtually augmented crowds

Tim Kindberg (matter2media.com)
Jakob Bardram
Sebastian Büttrich
Morten Esbensen
Steven Houben
Rilla Khaled
Thomas Pederson
Gian Perrone
Juan David Hincapié Ramos
Aurélien Tabard

IT University Technical Report Series

TR-2011-146

ISSN 1600-6100

November 2011

Tim Kindberg
Jakob Bardram
Sebastian Büttrich
Morten Esbensen
Steven Houben
Rilla Khaled
Thomas Pederson
Gian Perrone
Juan David Hincapié Ramos
Aurélien Tabard

IT University of Copenhagen

ISSN 1600-6100

ISBN 978-87-7949.245-5

Copies may be obtained by contacting:

IT University of Copenhagen
Rued Langgaards Vej 7
DK – 2300 Copenhagen S
Denmark

Telephone: +45 72 18 50 00
Telefax: +45 72 18 50 01
Web: www.itu.dk

1. Introduction

This paper is the result of a six-week research project on the topic of crowd computer interaction, which was conducted at ITU in October/November 2011 while the first author was resident as a visiting professor. The project led to a proposal for mesh mobs, a new type of smart crowd, in which physically collocated people interact via their WiFi-enabled mobile phones. The paper provides a brief background to crowd computer interaction, motivates the concept of mesh mob and presents some design and implementation issues pertaining to applications for mesh mobs.

Our interest in applications for crowds stems from two perspectives. The first is experiential. Crowd computer interaction is, broadly speaking, multi-person interaction with (and via) devices, designed to bring about both individual and collective experiences and phenomena. While this notion (and that of applications for crowds) has existed for decades, only relatively recently has it become a subject of systematic study. A first workshop on the topic, co-organised by the first author, was held at CHI 2009 [Brown et al. 2009]:

> *Crowds are an integral feature of contemporary life, and emerging technologies raise questions around how new interactional forms might work in crowd contexts. In this workshop, we will examine crowds as a unit of interaction for technology design, focusing primarily on collocated assemblies of people who interact in the same timeframe. ... The predominant themes of the workshop are interactions within crowds, diverse crowd configurations and individual and collective experience.*

We report on some related work below, including analyses, applications and system support; but progress has been rather ad hoc and there is plenty of room for fresh exploration.

The second factor that led to this project is more of a systems perspective, specifically a long-standing interest in *volatile* systems, i.e. those in which the sets of participating users, software and hardware components are subject to routine change [Coulouris et al. 2011]. This assumption is in stark contrast to most systems algorithms, which typically assume that failures and other forms of component change are relatively rare.

As an example of a volatile system, consider a physically collocated crowd carrying smart phones. People come and go at random but in principle whoever is instantaneously present could be provided with a collective crowd-based application. We considered the following more specific examples.

Mass protests (and disturbances). Protesters could benefit from apps that allow them to report what happens, both to one another and to the outside world. Since bandwidth may be lost or taken away, in some ways these scenarios are applicable also to disasters such as 7/7 in London and 9/11 in New York. Protests are not constrained to any particular space and indeed the crowd as a whole may move from space to space.

Running events (races). While these take place over a predefined course and are highly organised, they are interesting in that the spatial density of the crowd of runners changes considerably over the course. Since the participants are intensely involved in the existing experience, any app for them would need to be very lightweight – pertaining, for example, to measurements such as who passed whom and when. The app and associated devices would need to have extremely low overheads in every sense.

Music festivals. Festivals like Glastonbury or Roskilde lend themselves to apps for entertainment. Again, the environment is constrained overall and organised, but there is plenty of scope for crowd formation and dissolution as performers take and leave the stage and, of more interest in this context, as the crowd 'does its own thing'. Reeves et al. [2010] studied the ways in which people within the crowd of football supporters solicit participation from others. Perhaps here is where issues of cultural differences and community formation could best be studied.

Online participants. Our focus is on physically collocated people but interactions between them and people who are physically absent but online is also of interest. For example, football fans exchange texts with their friends who could not come to the match [Reeves et al. 2010]. In the game Uncle Roy Is All Around You [BlastTheory 2003], online players collaborate with physical-world players. The two operate in different but intersecting spheres. During protests and disturbances – and indeed during disasters – people use Twitter and other social media to report from the ground but also to comment from afar.

Taking the experiential and systems perspectives together, some immediate questions arise. These are the starting point for this paper:

- What kinds of experiences/applications would be valuable for crowds such as these?
- How should one design for crowd-based experiences?
- What types of systems/algorithms can continue to function within the parameters of such an experience, despite the volatility?

Section 2 describes related work. Mesh mobs and the motivation for them are described in Section 3. Section 4 describes the 'sync and hold' architecture for mesh mobs. An implementation on top of Android is outlined in Section 5. Section 6 deals with design issues. Section 7 briefly discusses some other factors that were discussed during the research, and Section 8 concludes.

2. Background and related work

Roughton et al. [2011] provide a taxonomy of crowds, interaction styles and applications. They provide plenty of examples of applications, especially of games, and observe how they relate to the interaction models and intentions of individuals in the crowds. But they do not build upon studies of particular types of crowd, ones that already exist in the absence of these applications. Instead, the study is mainly concerned with crowds that are brought together by applications, particularly games. Similarly, O'Hara and Glancy [2009] observed crowds drawn to games and shows on BBC Big Screens. Some people play the game, others watch. There is a flow of people between the two roles and their study describes this.

Reeves et al. [2010], on the other hand, studied football fans and how they behave en masse, and then drew tentative inferences about what types of application and system support might be appropriate.

Our perspective is similar to that of Reeves et al. [2010], who consider crowds as autonomous and self-defining entities in their own right. As they put it: "Crowds are distinct in that they offer a setting where large-scale participation is a key characteristic. Furthermore, this participation is not necessarily mediated by some singular 'spectacle' as with audience-performer scenarios, and any performance-like activities are more distributed, fluid and shared amongst members of the crowd. The relationships (and interaction) between members are much more varied than that between performer and audience, where (on the whole) there is a maintenance of shared attention. In crowds, individuals and the group maintain a shifting focus for participants."

2.1 Mobs

There is much popular interest in different types of 'mob' such as smart mobs [Rheingold 2002] and flash mobs, which go beyond the old-fashioned pejorative sense of the word. We consider those here in order to draw out some characteristics of crowds. This is not meant to be an exhaustive account but it is illustrative of what was considered within the project.

Classic mobs

People have always gathered in the same space without necessarily having the full sanction of the authorities in order to take collective actions or for collective experiences. This includes people who collect together to protest, riot, loot or rave. There are new communication systems that affect how classic mobs organise and behave nowadays, but this type of mob (in the original sense) existed long before the internet. The dictionary definition is of interest here: Mob *Origin*: 1680–90; short for Latin *mōbile* vulgus the movable (i.e., changeable, inconstant) common people [dictionary.reference.com] – a nice reference back to volatility.

Smart mobs

Smart mobs are self-organising collections of people who develop networks to make them more capable than their members would be alone. For example, people with a common political goal keep one another informed and discuss the issues. They are typically not physically collocated. The collective is typically relatively long-lasting (days or months or more are needed to become 'smart'), though there is the separate question of the rate of membership turn-over – mobs have an identity over and above the identities of the individuals that make them up. Interaction between members is not necessarily by broadcast, so it may be asymmetrical and intransitive. However, typically there is at least one shared repository of content produced by the members.

Machine mobs

In crowd-sourcing systems such as Amazon's Mechanical Turk (https://www.mturk.com), crowds of people distributed over the globe undertake independent tasks from a common pool. Although they share a common goal and common artefacts, they know nothing of one another by default. Their work is controlled from the centre.

Flash mobs

Flash mobs are one-off, collocated assemblies who come together largely spontaneously to create spectacles such as pillow-fights for other (unsuspecting) people in public places. Flash mobs are self-organised but stem from notices and discussions on the internet.

Subtle mobs

In subtle mobs [Speakman 2009], the participants gather in a public place and listen through headphones to synchronized audio files that refer to the listener as a character in a narrative. No one knows for sure who is participating, out of all the people around wearing headphones or earphones. The organiser controls registration and sends out audio files with instructions on where to go and exactly when to begin the soundtrack. But the experience plays out according to the whims of individual participants, affected though that is by the narration they listen to and whatever happens to be going on around them at the time. Subtle mobs are a mixture of the designed (the audio tracks) and the contingent (confluence with random events in the setting).

Ironic mobs

These mobs consist of young people, typically, gathered in common spaces such as schools, pubs and clubs. Their mobiles have Bluetooth switched on and discoverable and they project an aspect of their identity in their Bluetooth names, for discovery by others nearby [Kindberg & Jones 2007]. (This is a practice that has fallen into relative disuse since the mid 2000's, now that Bluetooth is supported with greater restrictions on smart phones.) Crowd members also sometimes provoke one another in the form of Bluejacking – sending a vCard containing a message to people nearby. There is an asymmetry of knowledge: steps are needed to identify who has which Bluetooth name. Bluetooth is thus an 'ironic' medium (in the sense of dramatic irony) in that some people (an audience) know something at the radio level that others do not know, and vice versa. There is some evidence of development of social mores over time, such as the use of provocative names in pubs.

2.2 Crowd Characteristics

The above types of mobs exemplify various characteristics that we can ascribe to crowds in general and which are of relevance to the design of both applications and the infrastructure needed to support them. Again, the following list is not exhaustive.

Social psychology. This is the discipline that seeks to "understand and explain how the thought, feeling and behaviour of individuals are influenced by the actual, imagined or implied presence of other human beings" [Allport 1985]. Crowds such as football supporters may exhibit group behaviours, as distinct from the behaviours of individuals. Moreover, individuals' behaviours may be influenced by the people around them. We speak, in particular, of group dynamics and peer pressure, and say that crowds have moods –one of the things that make them interesting. A recent, and extreme, example took place this spring in Bristol where the first author lives, when police clashed with local people [Guardian 2011] on a night that began with many people on the street in a mellow atmosphere. The Guardian's reporting is contested locally in some respects. That raises the question of what is the 'truth' about a crowd, particularly one in which emotions are strong: how can what takes place in a crowd best be captured?

Physical extent and density. Members may be collocated in a relatively confined area of the city, say, or spread all over the globe with the internet their only connection. Some crowds are combination of the two. In so far as they share a common physical setting, we can ask how dense they are and how they physically arrange themselves in terms of flows and clustering.

Temporal duration and volatility. In some cases, the members simply come together for a one-off event. Others have sufficient contact over sufficient time to develop their own organisation (e.g. political campaigns) or their own mores (e.g. football supporters). We can also ask whether the crowd continue to have a separate identity even if its membership changes to a significant extent over time.

Organisation & control. Some crowds form relatively spontaneously and are self-organising; in others there is some element of external control or at least suggestion. There are hybrids too, such as protests that are planned and organised but which behave spontaneously when people collect together. The motivations of individual participants and the social psychology of the crowd are relevant to what types of organisation are appropriate and possible.

Limited knowledge and awareness. Crowd members have something in common that has caused them to collect in the first place. It is often also useful to consider what, specifically, by the nature of the crowd or of the intermediation channels between them, the members do *not* know, or are not aware of, about one another.

2.3 System support for crowds

The underlying system infrastructure needs to provide some kind of useful invariant, despite the volatility of the crowd. For example, it could provide continuous connectivity despite the loss of wide-area connectivity, as in the B.A.T.M.A.N. protocol [B.A.T.M.A.N. 2011], which has been implemented for routing between mobile phones, here at ITU.

Another possibility would be to provide a higher-level abstraction such as publish-subscribe communication, in which any member of the crowd could publish information, and all members of the crowd would be able to receive that information. The Haggle project [Pelusi et al. 2006] sought to do that, based on opportunistic ways of achieving communication despite high rates of connection failures. Opportunistic communication is similar to delay-tolerant/disruption-tolerant networking but makes, unlike the latter, no *a priori* assumptions at all about the topology of the network. This model is extremely volatile: the set of entities participating and the state of connectivity is assumed to change often. The opportunistic model is also data-oriented. That is, how one propagates data depends to some extent on the semantics of the data itself. Wireless Sensor Networks operate similarly, although the assumptions for those networks tend to be somewhat less volatile than with crowds of moving people.

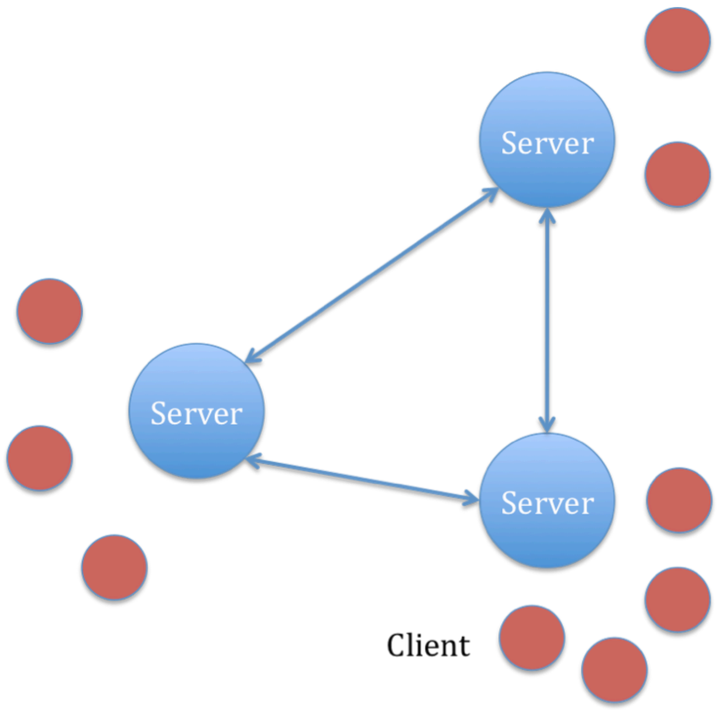

Figure 1. The gossip/epidemic algorithm

As an application of Haggle, Martín-Campillo et al. [2010] describe the problem of triage during a disaster. Medical workers with mobile devices need to report on who needs how much help and where. The authors describe generalised forwarding policies based on various factors. Those include pre-existing social structures and known movements of individuals, neither of which applies to crowds. The forwarding policy they decide to use in a disaster is not very clearly explained in the paper but it does involve the active participation of the personnel, who are under orders to return to a contact point in a certain timeframe.

Zhao et al. [2004] were the first to describe how some nodes can move and act as 'ferries' of data within mobile ad hoc networks. But they base their analysis on sparse networks, unlike (potentially) crowds.

Tennent et al. [2005] implemented an epidemic ('gossip') algorithm (Fig. 1) on top of WiFi for communication among PDAs and laptops. Point-to-point communication links are assumed to exist most of the time. The authors describe three applications built on top of the algorithm. The same group at Glasgow were investigating 'seamful games' [Bell et al.], in which players had to become aware of where WiFi access points were and were not within range. While they did not address crowds as such, they were concerned with human sensibilities as regards the affordances of the network(s) used by collections of people.

There are many concrete applications that could be prototyped without, at this early stage of research, trying to fix on what the lower layers should be. For example, Twimight [Twimight 2011] is a Twitter client designed to send tweets over Bluetooth when wider connectivity goes down.

3. Mesh mobs

The foregoing considerations led us to consider how to make collocated crowds of people smarter, and at the same time to introduce aspects of 'subtlety' and 'irony' that could be exploited in games.

We therefore propose the concept of a mesh mob: a physically collocated crowd in which the network is implemented by, and with the conscious involvement of, the people in the crowd.

Conventional WiFi meshes and sensor networks are typically mounted on parts of the inanimate physical world (though sometimes on animals too). In a mesh mob the network is mounted on the people in a crowd in the form of their WiFi-capable mobile phones. In addition, there may be more powerful network nodes in vans and on bicycles. There might even be network nodes in the form of drones flown overhead.

Mesh mobs bring the affordances of the network up to the human level in order to solve the more challenging problems of managing (or playing with) that network. The operating assumptions of opportunistic systems such as Haggle [Pelusi et al. 2006] are that:

1. System-level communication happens under the covers, away from explicit human involvement; humans are the carriers of data but they don't *consciously* participate in its carriage.
2. The system algorithms try to cope with human movement across unrelated sets of people, i.e. with the making and breaking of communication links that results. For example, my phone starts communication with your phone in a bar; you later catch a bus and pass it to someone else, who gets home and passes it on again, etc.

But crowds are different in two ways. First, they are a whole, rather than a set of only piecemeal-related people distributed across much larger areas. Second, since crowds have an identity, we can consider circumstances in which the crowd has a *collective interest* in the transmission of data, and is willing to *participate actively* in it. An image that inspires this idea is of a collection of people passing water or sand in buckets in a chain, one to the other, to put out a fire.

In order to make the network function satisfactorily, the members must collaborate. The mesh is made up of relatively low-power radio transceivers. Unless the nodes move to where they are needed, overall connectivity will break. Healing the network may be extremely challenging, since the crowd may have its own unpredictable dynamics in terms of physical flow and density, and usage patterns over space and time.

Our research goal is specifically to address mesh mobs as cybernetic systems consisting of people and the applications on their phones that implement and use the network. While we would like to automate as far as possible those network tasks that are simply laborious, we expect that some tasks will benefit from the application of human intelligence acting on information supplied by software. The benefit may be in terms of functionality or, in a gaming context, in terms of (satisfying) challenge.

We can relate mesh mobs to the various types of mob described in Section 2.1. First, in general, a mesh mob is a physically collocated smart mob, in that the members create a network and pass knowledge along it to satisfy their mutual goals. There is a mixture of people in this system:

- Those who implement the network itself.
- Users of the network, such as people who tweet over it.
- Others: bystanders who may or may not know that the mesh mob is taking place around them.

Second, there is an element of 'subtlety' and 'irony' (see Section 2): people do not necessarily know who is part of the mesh mob; and people who are part of the mesh know things not known to others.

Lastly, in so far as mesh mobs have collective work to do to maintain the network, they are what we call machine mobs. Roles and tasks need to be divided among individuals in such a way as to maintain the network for the crowd as a whole. However, this crowd-sourcing element may be very different from systems controlled from the centre, such as Amazon's Mechanical Turk. A machine mob may, on the contrary, be self-organising and self-aware.

Here are four concrete motivations for mesh mobs:

- They are useful in situations where wide-area bandwidth has disappeared locally. Disasters, protests and festivals, for example.
- There is a class of games that can be built on a mechanism by which virtual things are physically moved. Of course, there are other ways of achieving that movement, e.g. with a GPS-sensing app connected over 3G to a server. But those don't exercise crowd coordination in as challenging a way.
- There is a strong but tenuous relationship between the *human*-physical and the virtual, which can be played upon, Bluetooth-style, for extra affordances in the experience.
- Since mesh mobs are collocated, they provide an opportunity for face-to-face community togetherness, awareness and, arguably, empowerment. The physical proximity of people in a mesh mob runs contrary to the tendency for activities to take place more and more exclusively online.

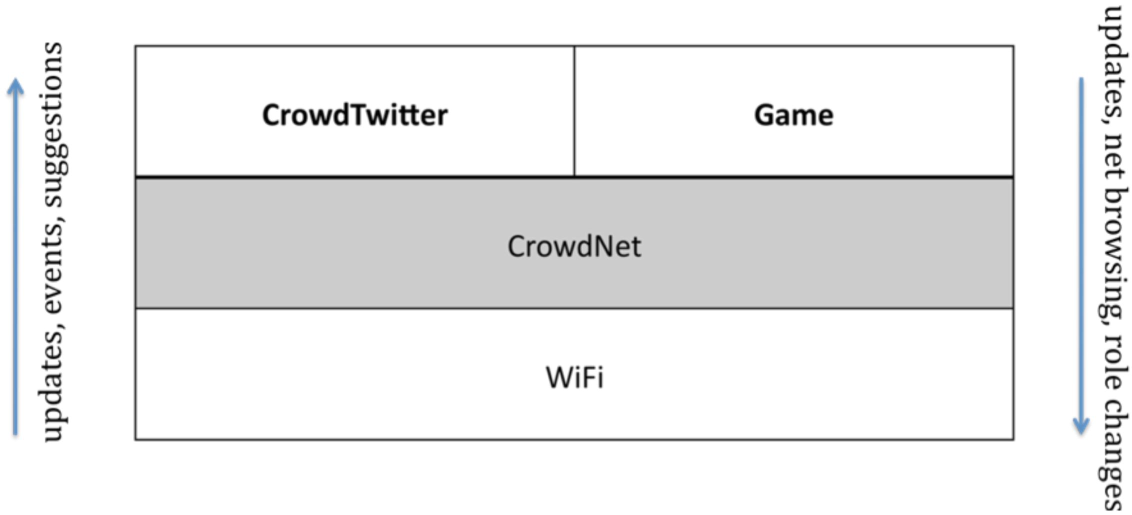

Figure 2. Layers in the mobile phone

For example, a protest crowd without 3G would be interested in seeing that its collective tweets about what is happening on the ground will make it to twitter.com as soon as possible. If they can 'ferry' this 'precious cargo' to a point outside with 3G, then that objective will be realised. Additionally, tweets from outside – and information about which tweets from the crowd itself have been published so far – can be ferried back to the crowd.

To take our second scenario from the introduction, in a race, there could be virtual runners who are competing in parallel with the physical runners. If the runners collaborate, they can send their virtual runners ahead by radio.

At a festival, our third example from the introduction, rival groups of fans could try to pass on music tracks or images competitively, in such a way that eventually one particular track or image will prevail and the others will cease to be available, according to a voting scheme. People may infiltrate one another's 'crowds' in order to have their media propagate across the musical divide … or collect together to try to increase the density of their media and thus its power to propagate beyond.

4. Sync and Hold: a 'strawman' realisation of mesh mobs

This section presents a concrete (though, at the time of writing, only partly implemented) proposal for both human roles and system components in mesh mobs. It is presented as a 'strawman', i.e. a point for discussion and initial experimentation.

We present a generic architecture for the *crowdnet*, the network that routes data across a crowd, including media such as text, images and videos, or any application-specific data. The crowdnet may be used solely for intra-crowd communication – as, for example, in a closed game. Equally, there may be nodes with wider connectivity to the internet that are within or reachable from the crowdnet, in which case the crowdnet may route data to and from the internet as well.

In principle, several crowdnets could operate within one physical crowd, implemented by different people.

A number of crowd-aware applications may run on any given crowdnet node (Fig. 2). Those applications generate data to be sent over the crowdnet and receive data from it. In addition, the crowdnet maintains information about itself that may be supplied both to users of the crowdnet and those who are minding (and implementing) it. For example, people trying to tweet across a crowdnet could be given information about how many others are currently able to receive their tweets.

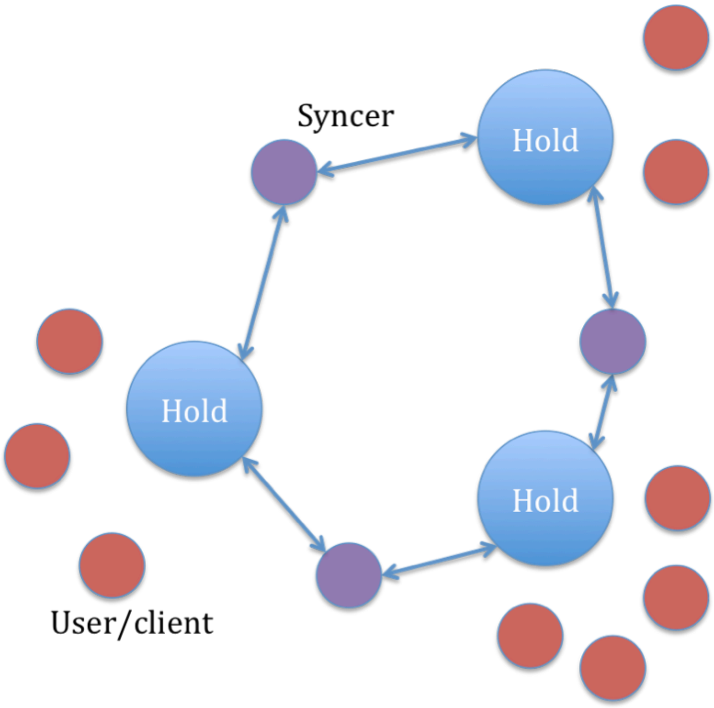

Figure 3. Users, Holds and Syncers.

The crowd

Each mesh mob is assumed to be associated with 'mob data' in the form of a name, a password and a region where the crowd congregates, which can be presented as a map on the phone (the current implementation uses Open Street Maps). The password is a credential for participating in the mesh mob. The mob data could be created at a website and disseminated in any suitable way prior to the crowd gathering, if that is appropriate. Equally, once the crowd has gathered, the mob data can further be disseminated directly on the basis of trust decisions between people on the ground. For example, people can pass the data around directly to new participants from phone to phone by Bluetooth, NFC or QR code.

In addition, there could be application-specific mob data such as a twitter account for that crowd, which everyone would use. Fittingly, the crowd as a whole would have a single Twitter identity but with many different voices within it. This sharing of an account works well with the participants in the collective interactive experience known as Fortnight [Fortnight 2011].

4.1 Roles

In a mesh mob we consider three roles for people with associated roles for their mobile phones: users, holds and syncers (Fig. 3).

Users generate and consume application data such as 'tweets'. This data is transferred to and from nodes in the crowd that act as repositories, called *holds*. We use the metaphor of Zhao et al. [2004] that people act as ferries of data through the crowd, and we say that they keep that data in the *hold* that is their mobile phone. We will talk equally of holds in the human and system senses. Users write to and read from holds over wireless connections.

Holds are limited in their physical resources: the power of their radios, their storage capacity and their battery capacity. We therefore anticipate that a mesh mob will in general have several holds both at the same time and over the course of time, covering different physical areas at different times, for example. The collective battery capacity and storage capacity of a crowd may be very large. It is an important challenge to multiplex the crowdnet over those collective resources in such a way that individuals do not lose their mobile phone's functionality altogether.

Holds will acquire and propagate different sets of data from different users, by default. Therefore, we need a way for holds to be synchronized. *Syncers* are people with mobile phones who synchronise holds: they bring each to the same set of updates.

This is similar to a conventional 'gossip' or 'epidemic' system in which a set of servers (holds) accepts updates and queries from clients. Those servers manage state that is supposed to be eventually consistent. They achieve this by, from time to time, picking one of the other servers at random and engaging in a synchronisation protocol. Each tells the other about any updates it has not yet received.

That model assumes the set of servers is known, and ignores any contextual factors about which servers have a network connection available to synchronise themselves.

In our mesh mob proposal, we turn synchronisation into a third human role, for three reasons. First, two holds may be out of radio range and therefore unable to synchronise directly, while a third person may be in radio range of each and thus able to move data between them. Second, we view the role of a syncer as a place to inject human intelligence about which hold needs synchronizing with which other. Third, if a hold acts as a WiFi access point, which is so in our initial implementation, it cannot also connect to another hold.

4.2 The state of a hold

A 'healthy' crowdnet can be measured by its state of connectivity and the latencies and throughputs across it. Holds are key components of the crowdnet. There are several state attributes that are relevant to maintaining a healthy crowdnet:

- The hold's *position* on the map.
- The hold's *capabilities* such as whether it has a direct connection to the internet.
- The hold's *load* measured through such factors as the volume of data it stores and the rate of updates from connected clients.
- The hold's available storage and battery *resources*.
- The recent *synchronisation history* of the hold: which other holds it was synchronized with, and when.

The crowdnet propagates hold states as well as application data – using the same mechanism – in order that humans can examine the state of the crowdnet as a whole and take actions as necessary.

4.3 Location

It is useful to know the positions not only of holds but in principle also of syncers and users.

Position could be determined using on-board location-sensing such as GPS. But that has disadvantages: it can take a long time to initialise; it may not be available at all in certain locations; it is battery-draining; and it is not always accurate enough.

An alternative is for people to self-report their locations by clicking on a map. That has none of the disadvantages of GPS. It is, however, subject to people forgetting to update their position, or lying about their position (an aspect that could be a feature rather than a bug in a gaming context).

This raises interesting questions about how to design a system that will engender accurate self-reports of locations. Moreover, any such system could allow for the collective determination of a person's position. I may not know where I am on the map but others may be able to help me. Equally, in case I lie or make a mistake, others may be able to verify and contradict my position.

WiFi signal-strength acts as ground truth: if I assert that I am a hold near you, you can scan for me and verify my radio visibility and signal strength.

5. Implementation on Android

In order to provide applications for many people, we considered what could be built with Android smart phones, preferably without rooting them (adapting the underlying OS). First, a few facts about Android:

- Many (all?) android phones can scan for, and connect to, wireless access points.
- Andoid phones with WiFi can become wireless access points.
- Android phones cannot form ad hoc networks unless they are rooted. (In forthcoming Android version 4.0, WiFi Direct will be supported. While not precisely ad hoc mode, this would be an interesting technology to exploit.)

We implemented holds as WiFi access points and web servers. Anyone can scan for the holds within radio range, without connecting to the holds. We encode information as a hint into the holds' 32-character SSIDs, including the lat/long coordinates of the hold and information about the remaining state attributes as described above. People monitoring the network can see information about where the holds are (or were) and their state by looking at a map. However, in general, not everyone will be able to see the same information, and information may be out of date.

Each access point is protected by the WPA crowd password. People without the password cannot connect to the corresponding crowdnet.

The lat/long coordinates of a hold can be extracted from whatever location-sensing facilities are available on the handset. Equally, people can report their locations themselves by clicking on the map.

Syncers are realised by connecting to two holds in turn. This is, unfortunately, time-consuming since a full WiFi association can take seconds to make. Also, a minimum of three exchanges are required to ascertain which updates need to be transferred in each direction.

5.1 Time

Clock synchronisation is challenging in a volatile system so we do not attempt it. Instead, we measure time backwards in a crowd as follows. Every node records the time at which it created an item. It therefore knows at any point in the future how long ago the item was created. If we ignore the time taken to transfer items between nodes over radio, that information can be transferred and maintained. Inaccuracies of a few seconds (the time taken to transfer items) will creep in as holds are synchronised. Those inaccuracies should not be of great importance in a system that anyway has very large latencies.

This procedure amounts to ‘age-stamping’ items instead of time-stamping them.

5.2 Identification

Crowdnet nodes have unique MAC addresses, which we use to uniquely identify the data objects they create by appending a sequence number. The item is also age-stamped. Thus, every object carries identification of which node created it, where it lies in the sequence from that node, and how many seconds ago it was created.

6 Mesh Mob Apps and Experiences

The foregoing strawman proposal is not the only possible architecture for mesh mobs. Moreover, even if we use that ‘sync and hold’ architecture, it leaves many questions unanswered and plenty of room for design innovation. In this section we outline some of the factors that a designer for mesh mobs should take into account.

At a basic level, not all applications are suitable for mesh mobs. Probably a shared spreadsheet, for example, would be an unsatisfactory program if ported to a crowdnet! Crowd-aware applications must be able to withstand high (and highly variable latencies) and low throughputs. There may be a complete loss of data. The data produced and consumed by a crowd-aware application should be immutable, since concurrency control techniques are likely to be unusable on such volatile systems.

On the other hand, this approach is good enough for ‘best effort’ reporting by users of what is transpiring during a protest. It is also good enough for games where the network issues are transformed into an entertaining challenge.

6.1 Designing for Mesh Mobs

The following are a few of the many questions one might ask, in no particular order, when designing for mesh mobs. We leave them as questions since there has been little time to consider them.

1. What kind of functional goal or experience is required?
2. Are members self-motivated or motivated by the wellbeing of the crowd? Do they need to be recruited, rewarded or constrained? What ‘persuasive’ techniques are relevant?
3. What is the organizational mechanic in terms of tasks, roles and rewards/consequences?
4. How do you bootstrap a mesh mob as simply as possible?
5. How do you make it last as long as required?
6. How do you manage the various types of volatility such as people arriving, leaving and wanting to change their roles; phone batteries failing; human flows and densities changing; and radio propagation changing?
7. What is the relationship between the physical world and the virtual world, in terms of what can be inferred about the one from the other?
8. What are the physical signalling arrangements between the members (raising arms, wearing hats etc.)?

9. How is the state of the network manifested in the interfaces of those using it – in particular, in how it affects their use and in terms of signals to the users, perhaps to encourage them to join the network itself?

6.2 Heuristics

We expect heuristics to emerge for humans to use in maintaining the crowdnet. Those heuristics consist of ways of behaving under certain circumstances, especially in order to fix a current problem with the crowdnet.

For example, if a syncer detects (by scanning) only one hold in range – a hold that has not been synchronized recently – it (she or he) can move until it can simultaneously scan a second hold, perhaps looking in places where other holds were known to be recently.

Similarly, if a hold detects that it has not been synchronized for some time, it can inform all the nodes using it, and crowd-aware applications on those nodes can inform the users. It is ultimately in the interest of those users to become a syncer. Otherwise updates will not be propagated.

As another example, a syncer that finds itself synchronizing a hold with low battery levels could become a hold to replace it.

Signalling systems are required so that people in the mesh mob will know how to behave to maintain it. One signalling system is for people to be informed via the applications they use of urgent (and perhaps not so urgent) network conditions.

Similarly, the people involved may employ direct signalling to one another. For example, people who are holds or syncers could use a gesture to communicate that fact to those around them. Equally, they could wear distinctive armbands.

7. Discussion

Perhaps the biggest single question about mesh mobs is that of how they can and should be organised appropriately and efficiently. We proposed a strawman model in which people play the roles of holds and syncers, temporarily at least, and in which they organise themselves through those roles to maintain the crowdnet for their over-arching purposes.

However, a variety of models are possible, differing in at least three ways.

First, the nature and quantum of crowdnet-maintenance 'work' may be different. The tasks may be heterogeneous or heterogeneous. People may perform a single task at a time rather than playing a role for an extended period. Tasks may be formulated more in terms of the application, rather than the maintenance of the crowdnet. People may multi-task – using the application and maintaining the crowdnet at the same time – or do one thing at a time.

Second, the organisation and direction of the work could vary. It is conceivable, for example, that the system may be left in control, automatically computing where people should move and what they should do, with no need for human discretion. There may be more or less teamwork as opposed to individual work. Some crowd members could direct others.

Third, there is the question of motivation, both intrinsic and extrinsic. However the work is broken down, a mesh mob has a voluntary and informal infrastructure [Hincapié-Ramos et al. 2011]. Given these factors, we must answer several questions in order to arrive at a feasible and experientially satisfying cybernetic whole. Why should an individual participate in the maintenance of the crowdnet? At what level and period of activity is the maintenance work sustainable? How can people be recruited to the maintenance activity and a critical mass be achieved? What rewards may be required?

Other issues

Several issues that were not covered above entered into our discussions. We outline them here, in no particular order.

Security

The crowdnet is by default open to those in possession of the collective password and therefore good only for communicating non-personal data. One would not tweet to one's personal twitter account using it, for example. That is one reason why we suggested the use of a shared, crowd-specific twitter account.

Applications could use end-to-end encryption where that was required. However, rather than viewing the open network as a problem to be solved, an alternative perspective is to view it as a creative opportunity: to design applications that are truly for the crowd as opposed to the individuals that make it up. What would – could – 'CrowdTwitter' be like in that respect?

We acknowledge that the system we have described would be easy to attack. Someone has only to pose as a 'friend' of the crowd and obtain the crowd's credentials (WPA key). That person can then freely read the crowd's data and mount a denial-of-service attack.

Instrumenting the crowd

It would be very useful to know aspects of the physical dynamics of the crowd. For example, is it largely standing still? Are there prominent flows or movements?

In addition to data from the applications and data about the network itself, it would be possible for individuals to sense, for example, accelerations, velocities and ambient noise levels, and to send that over the crowdnet for collation.

Modelling and simulation

It would be highly desirable to model and compare the effects on the crowdnet of different network protocols, human movement behaviours, and high-level game mechanics.

The parameters involved would be the hold attributes identified in our sync and hold protocol, plus models of human behaviour and movement, and radio ranges (which themselves are a function of human density, among other things).

Return receipts and feedback

Someone sending a tweet over the crowdnet would require feedback about the state of the communication of that tweet. Has it gone beyond the hold it first arrived at? How many people have received it in the crowd, ideally as an estimated proportion of how many people are (electronically) active in the crowd as a whole? Has it reached a node connected to the internet, and therefore been sent to twitter.com?

Equally, data such as tweets from people who are online but not physically present are of interest to people in the crowd. How are those routed from nodes with internet connections to those without? At the system level, we can propagate them using the same sync and hold protocol. But how, if at all, should the people implementing those roles behave differently between intra-crowd communications, data sent out to the internet, and data incoming from the internet?

8. Conclusions

We have described a new type of 'mob', mesh mobs, consisting of collocated people whose WiFi-enabled mobile phones make up the network they use, the crowdnet. Mesh mobs are designed for a new type of functionality and experience: one in which the members of a crowd 'ferry' some 'precious cargo' around. The virtual bits of this important electronic resource are consciously transported physically between the crowd members. We can also think of this as collectively cared-for communication. The goal of that communication may be functional (e.g. reports of a protest) or more experiential (e.g. moves in a game).

We outlined a strawman architecture for constituting a mesh mob: a crowdnet with one or more crowd-aware applications running on top of it. We have described the beginnings of an Android implementation.

This work (the result of a six-week residency) remains at an early stage. It is our hope that the development can continue, through MSc projects at ITU and as an open-source project for development by the wider community. The implementation is available for development at https://github.com/matter2media/crowdnet. We encourage playful experimentation with these ideas.

Acknowledgements

Tim Kindberg would like to thank ITU, and Jakob Bardram in particular, for the opportunity to conduct this research as a visiting professor: for the excellent hospitality and the stimulating discussions during my stay.

References

[Allport 1985] G.W. Allport. The historical background of social psychology. In G. Lindzey & E. Aronson (Eds.), The handbook of social psychology. New York: McGraw Hill.

[BlastTheory 2003] Uncle Roy Is All Around You. http://www.blasttheory.co.uk/bt/work_uncleroy.html

[B.A.T.M.A.N. 2011] B.A.T.M.A.N. (better approach to mobile ad-hoc networking) project home page http://www.open-mesh.org/.

[Bell et al. 2006] Marek Bell, Matthew Chalmers, Louise Barkhuus, Malcolm Hall, Scott Sherwood, Paul Tennent, Barry Brown, Duncan Rowland, Steve Benford, Mauricio Capra, and Alastair Hampshire. 2006. Interweaving mobile games with everyday life. In Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI '06), Rebecca Grinter, Thomas Rodden, Paul Aoki, Ed Cutrell, Robin Jeffries, and Gary Olson (Eds.). ACM, New York, NY, USA, 417-426.

[Brown et al. 2009] Barry Brown, Kenton O'Hara, Timothy Kindberg, and Amanda Williams. 2009. Crowd computer interaction. In Proceedings of the 27th international conference extended abstracts on Human factors in computing systems (CHI EA '09). ACM, New York, NY, USA, 4755-4758. http://www.metamanda.com/crowdcomputing

[Coulouris et al. 2011]. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair. Chapter 19 of Distributed Systems, Concepts and Design, 5th edition. Pearson 2011.

[Fortnight 2011] Fortnight. project home page http://fortnightproject.com

[Guardian 2011] The Guardian. Bristol riot over new Tesco store leaves eight police officers injured. http://www.guardian.co.uk/uk/2011/apr/22/bristol-riot-police-injured.

[Hincapié-Ramos et al. 2011] Juan David Hincapié-Ramos, Aurélien Tabard, Jakob E. Bardram. GridOrbit – An Infrastructure Awareness System for Increasing Contribution in Volunteer Computing . Proc. CHI2011.

[Kindberg & Jones 2007] Tim Kindberg and Tim Jones. Merolyn the Phone: A Study of Bluetooth Naming Practices. In proc. Ubicomp 2007.

[Martín-Campillo et al. 2010] Abraham Martín-Campillo, Jon Crowcroft, Eiko Yoneki, Ramon Martí and Carlos Martínez-García. 2010. Using Haggle to create an electronic triage tag. In Proceedings of the Second International Workshop on Mobile Opportunistic Networking (MobiOpp '10). ACM, New York, NY, USA, 167-170.

[O'Hara & Glancy] Kenton O'Hara and Maxine Glancy. 2009. Watching in Public: Understanding Audience Interaction with Big Screen TV in Urban Spaces. Chapter in Social Interactive Television

[Pelusi et al. 2006] L. Pelusi, A. Passarella, M. Conti. 2006. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. IEEE Communications Magazine, Vol. 44, No. 11. (November 2006), pp. 134-141.

[Reeves et al. 2010] Stuart Reeves, Scott Sherwood, and Barry Brown. 2010. Designing for crowds. In Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10). ACM, New York, NY, USA, 393-402.

[Rheingold 2002] Howard Rheingold. Smart Mobs: The Next Social Revolution. Basic Books

[Roughton et al. 2011] Adam Roughton, John Downs, Beryl Plimmer and Ian Warren. 2011. The crowd in the cloud: moving beyond traditional boundaries for large scale experiences in the cloud. In Proceedings AUIC 2011. ACM, Perth, Australia.

[Speakman 2009] Duncan Speakman. Subtle mobs. http://subtlemob.com/

[Tennent et al. 2005] Paul Tennent, Malcolm Hall, Barry Brown, Matthew Chalmers, and Scott Sherwood. 2005. Three applications for mobile epidemic algorithms. In *Proceedings of the 7th international conference on Human computer interaction with mobile devices \& services* (MobileHCI '05). ACM, New York, NY, USA, 223-226.

[Twimight 2011] Twimight project home page. http://code.google.com/p/twimight/.

[Zhao et al. 2004] W. Zhao, M. Ammar, and E. Zegura. 2004. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '04). ACM, New York, NY, USA, 187-198.