Urbansim: A Simulation Tool for Peer-to-Peer Information Sharing in Smart Cities

Samuel R. Cauvin[a,*], Andrew J. Skinner[a], Alexander I. G. Stuart-Kregor[a], Wamberto W. Vasconcelos[a]

[a]*Dept. of Computing Science, University of Aberdeen, U.K.*

Abstract

We report on a simulation tool, called Urbansim, to explore the performance of peer-to-peer information sharing in urban "smart city" environments. Our simulation tool allows the modelling of street furniture (e.g., bus stops, lamp posts, rubbish bins, etc.) which, in addition to their regular purpose, also serve as peers generating, storing, requesting, and relaying information to/from other peers. We can also model moving components, for example, people carrying mobile phones, as well as vehicles such as buses, ambulances, police cars and fire engines. We list and justify requirements and propose a reference architecture to meet them. We survey available technologies, selecting and justifying which ones were used for the various components of our architecture. We model a scenario and evaluate how the classic flooded request peer-to-peer mechanism performs in a smart city scenario. Additionally we present a usage scenario of this simulation tool to explore a peer-to-peer advertising mechanism within a smart city and evaluate the scenario with regards to revenue and information propagation.

Keywords: peer-to-peer, smart cities, simulation, advertisement auctions

1. Introduction

We report on a tool, called Urbansim, for modelling and simulating peer-to-peer information sharing in urban "smart city" environments. Our tool allows the modelling of street furniture (e.g., bus stops, lamp posts, mail boxes, etc.) with digital capabilities which, in addition to their regular purpose, act as peers in an ad-hoc large area network. The tool also allows the modelling, also as peers, of moving components such as people carrying mobile devices, vehicles such as buses, ambulances, police cars and fire engines. These peers are modelled with capabilities to generate, store, request, relay and delete information (e.g., GPS location, temperature, traffic information, etc.) using alternative peer-to-peer mechanisms.

Mobile telephony and access to the Internet may not always be available. These can become unavailable or intermittent due to extreme weather, earthquakes, urban guerilla warfare, civil unrest and large-scale accidents caused by humans. We model an alternative, adopting peer-to-peer mechanisms [1, 2, 3] whereby components communicate to one another directly without a centralised point-of-contact. Street furniture such as bus shelters, mail boxes, rubbish bins, benches, lamp posts, traffic lights, billboards, and so on, can be *dual purposed*, that is, in addition to their original function, they can be equipped with digital capabilities (*e.g.*, Wi-Fi and/or Bluetooth signal, digital storage space, processing power, and sensors) which would provide basic functionalities for a distributed infrastructure to further support peer-to-peer mechanisms within an urban space.

The term "smart city" refers to urban spaces where, in addition to their physical infrastructure (*e.g.*, transport, public spaces, and so on), there is also availability of communication, social infrastructure, and information [4]. This can be used to improve services, social inclusion and participation, business, and general quality of life in cities. A city's sustainability, economic competitiveness and ability to attract business have recently been measured as how "smart" the city is [5]. We adopt the vision of a smart city within which

*Corresponding author

Email addresses: `r01src15@abdn.ac.uk` (Samuel R. Cauvin), `a.skinner.11@aberdeen.ac.uk` (Andrew J. Skinner), `a.stuart-kregor.10@aberdeen.ac.uk` (Alexander I. G. Stuart-Kregor), `w.w.vasconcelos@abdn.ac.uk` (Wamberto W. Vasconcelos)

people and street furniture interact sharing information and knowledge. Additionally, vehicles such as buses, trams, ambulances, fire engines, police cars and taxis, will have, like the street furniture, digital capabilities; by moving around over larger areas than pedestrians and at greater speeds, these vehicles will be able to gather and disseminate more information and knowledge.

The provision of information, such as GPS location, temperature and weather forecast, bus routes and timetables, traffic updates, places to eat, etc., from street furniture or vehicles would be cheaper and consume less energy than if the same information were obtained via mobile telephony and the Internet. Furthermore, this approach would increase the resilience of the current communication infrastructure (namely, mobile telephony and Internet) by having them *combine* in extreme circumstances (*e.g.*, high bandwidth demands, failure in centralised equipment, and so on), when some of the demand would be seamlessly handled by the peer-to-peer mechanisms and the smart city infrastructure. Finally, we note the business potential for provision of information and knowledge as well as the excellent communication channel the smart city infrastructure creates among citizens, city authorities, and service providers.

Our research provides a computational model of an actual urban space, in which there is an ecosystem consisting of various components stationed or moving around, all adopting peer-to-peer mechanisms to facilitate information dissemination. The model and mechanisms are available as a simulation tool to support urban planners, mobile phone app developers, service providers, and emergency services, among others.

The structure of this paper is as follows. Section 2 lists and justifies requirements and presents an architecture for our tool, and which meets the requirements. Section 3 describes the implementation of Urbansim. Section 4 provides an example scenario for our tool, and evaluates some performance metrics in that scenario. Section 5 describes and evaluates the use of our tool in a more complex scenario, namely one in which auctions enable advertisements to have access to devices. Section 6 provides an overview of existing and related research in this field. Section 7 discusses the limitations of the tool, drawing conclusions and outlining future work.

2. Requirements and Architecture

We aim at providing means for the modelling and simulation of peer-to-peer networks in urban "smart city" scenarios. We envisage a tool in which various stakeholders such as urban planners, mobile phone "app" developers, digital service providers, businesses, and citizens, are able to create realistic (albeit simplified) computational models. Our tool should allow them to study the behaviour of these models in order to understand their dynamics, and explore the space of possible solutions. This main objective gives rise to three requirements below.

The first requirement is to enable users to model urban environments. We thus need to model and simulate an urban environment with its topography and geography, also including street furniture, people and vehicles, and their likely movement (*e.g.*, buses along their routes, pedestrians moving at the appropriate speed along pavements, and so on) as well as a realistic network of roads, pavements, and paths.

The second requirement is to enable users to model and simulate peer-to-peer networks within the urban environments mentioned above. We aim to study issues like performance, feasibility and reliability of a wireless ad-hoc peer-to-peer network over an urban area. The peer-to-peer network needs to be modelled as realistically as possible, considering, for instance, the diversity of peers (e.g., with distinct processing power, storage capability, etc.), connection among peers taking place within a realistic range and for an adequate amount of time, data lost during transmission, as well as peers appearing and disappearing constantly. Individual peers stand for items of street furniture, people's mobile devices, and vehicles, all following the same peer-to-peer protocol to exchange information.

The third requirement is to enable users to model and simulate information propagation over peer-to-peer networks. We need to gather quantifiable evidence about peer-to-peer networks in smart-city scenarios, hence we must record information flow through the city-wide network.

We adopt a modular and customisable architecture, shown in Figure 1, consisting of a simulator, a model, and data components. The physical model component provides a machine-processable representation of the urban scenario, which should adhere to industrial standards (for ease of integration with existing technologies and off-the-shelf representations of cities). This representation of the world is input to the simulator component, providing physical constraints such as roads, buildings, etc., to be covered by the simulation. Our physical model also contains descriptions of participants such as street furniture, mobile

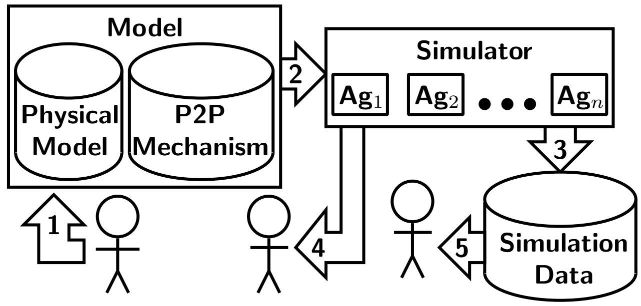


Figure 1: Architecture

devices and vehicles to be used in the simulation, aiming to accurately portray scenarios to be simulated. The tool relies on a design activity whereby a team of experts decides upon the urban scenario, the location of street furniture, routes of vehicles, movement of people, and so on.

We achieve generality and modularity by explicitly representing the peer-to-peer mechanism to be deployed in the simulation. This mechanism will be uniformly adopted by all components of the simulation, that is, it will run in street furniture, in mobile devices, and in vehicles. A machine-processable description of the peer-to-peer mechanism, together with the physical model, are passed to an agent-based simulator.

We use an agent-based simulator, associating components with software agents. The physical model indicates what the components are, and these are represented in the simulation (together with their position and movement), as a software agent, adopting the P2P mechanism. Agents simulate movement as well as the core functionalities of peers namely, connecting to one another, storing and deleting information, offering and requesting information, and relaying information around the network. The simulator records simulation data at specified intervals, thus allowing for off-line data analysis, in addition to real-time behaviour observation. The data analysis and behaviour observation may lead to changes or refinements of the input models, enabling the usual virtuous life-cycle of simulation models.

Our tool can be used in an experimental set up to automatically generate random parameters within set limits across multiple tests. The results of these tests give rise to a number of statistics in files with significant names.

3. Implementation

In this section we present technologies incorporated to our tool, and give details of how we implemented various functionalities.

3.1. Technologies

An important decision concerned the technology chosen for the actual peer-to-peer network simulation. We intended to adopt a multi-threaded simulation approach so as to achieve scalability, to allow for variability among peers and arbitrary levels of sophistication in individual peers. We considered three main candidates for these simulations:

- Java Agent Development (JADE[1]) Framework offers various support features, for debugging and developing agents, and is compliant with industrial standards (*e.g.* FIPA-ACL[2]), but its scalability was not ideal.
- PeerSim[3] offers great scalability and is specifically aimed at the simulation of peer-to-peer networks. However, it did not offer multi-threading, and its support for debugging and development was very limited.

[1] `http://jade.tilab.com/`
[2] `http://www.fipa.org`
[3] `http://www.peersim.sourceforge.net`

- Multi-Agent Simulation Of Networks (MASON[4]) is a library for agent-based simulations, offering user interfaces with real-time visualisation features. MASON allows the run-time inspection of agents which is useful for debugging.

These three platforms were stable, and offered useful on-line support (*e.g.*, manuals, tutorials, and discussion forums). We adopted MASON due to its light-weight multi-threading capabilities and ease-of-use.

Realistic simulations of urban environments require an accurate model of traffic as it moves through a city. We adopted the Simulation of Urban MObility (SUMO [6]) technology, as it provides many important road network features such as traffic lights and bus routes. SUMO is able to take data from OpenStreetMap[5] and convert it into its own format for use with simulation – this confers great versatility, as we can model real urban scenarios, available in OpenStreetMap. SUMO is stable, has an active community of users and developers, and has a convenient interface. We notice that SUMO has previously been combined with JADE to control traffic lights [7].

We aimed to provide customisability and modularity in our simulated peers' devices. To this end, we have software agents customisable to represent features of many types of devices. We achieved a great deal of customisation by using Java reflection[6] allowing distinct implementations to be loaded at run-time, which allows us to simulate many distinct types of devices within our network.

3.2. Simulation

This section provides an overview of the main functionalities of the simulation.

We adopt a cycle-based simulation approach with each cycle of the simulation happening at 1 second intervals; this 1-second cycle is sub-divided into milliseconds to simulate individual peers/devices, with each device using 1000 milliseconds per cycle. Some of the functions in the simulator have an associated time cost (*e.g.*, scanning for other devices within wireless range will take a set amount of time depending on the wireless technology modelled).

SUMO is implemented in C/C++ adopting a client/server architecture. SUMO operates as a server, allowing clients to communicate with it through a TCP/IP socket using the Traffic Control Interface (TraCI) protocol. By using the Java library TraCI4J[7] we can interact with the road simulation. TraCI4J is able to start a local instance of SUMO or connect to an existing server. For simplicity we create a local instance of SUMO and connect to it through the network loopback interface. SUMO then loads its required configuration files and is paused, ready for the start of the simulation. An "observer" gathers events from the simulation, such as messages exchanged between devices, states of sub-components of a device and the state of the agent running on the device. MASON provides a scheduling feature used for each step of the simulation.

Each step (term used interchangeably with cycle) of our simulation executes in a specific order, as shown in Algorithm 1. At the start of each simulation step the road network in SUMO is advanced. TraCI4J advances the SUMO simulation by a set time period, one second in our implementation. Once SUMO has advanced we then inquire if any new mobile components (pedestrians or vehicles) have appeared or disappeared within the simulated space, so that we can create or destroy the corresponding device in our simulation. We then update the position of mobile components in the simulated environment through the TraCI4J connection to the SUMO server. To improve performance and also to reduce network traffic, queries to SUMO are grouped together. We then create an array with all devices to be simulated and pass this on to MASON for a multi-threaded execution. MASON uses a pool of threads and associates these with devices which are simulated for a single step until all devices have progressed.

We model devices running in components of our smart city scenarios, such as a mobile phone being carried by a person or a bus shelter with solar-powered computing capabilities. We specify our devices as three separate sub-components, namely, an interface, a battery and storage space, and devices are simulated via software agents. The specifications of the many kinds of devices and their sub-components are stored in files which are read and used to create software agents. Our agents use Java continuations to enable them to resume from the previous cycle/step of the device simulation. The simulation of the device takes into account how much time there is left in the simulation step and if the battery still has power; the simulation

[4] `http://cs.gmu.edu/~eclab/projects/mason/`
[5] `http://www.openstreetmap.org/`
[6] `https://docs.oracle.com/javase/tutorial/reflect/`
[7] `https://github.com/egueli/TraCI4J`

Algorithm 1: Simulation Step

```
Advance SUMO simulation
Update mobile devices positions
// Step devices using multiple threads
forall Devices do
    Lock CoRoutine Mutex
    Update Device
    if GotTime & Battery then
        do
            Start Battery Drain
            Run CoRoutine
        while GotTime & Battery & Running
    else if Dead Battery then
        Remove Connections
    Unlock CoRoutine Mutex
Update the user interface
Log data using Observer
```

of the device consumes battery as the device sends and receives messages of the peer-to-peer protocol, and messages are stored in input and output buffers.

We developed a simple flooded request protocol, whereby peers will exchange messages to share information. If a request is received by a peer which has that information then the peer sends the information back; if the peer does not have the information, it then passes on the request to another peer. The peer mechanism is "running" on the device, and subject to the specification of its components (i.e., battery, interface and storage space). Peers become neighbours when they detect one another within a certain range (specified by the interface), and then they exchange messages, requesting and providing information. In our simulation, the peer-to-peer message exchange mechanism will increase the battery drain, as it processes messages in the input and output buffers. The interface establishes what information the peer needs and what information it can provide. The mechanism limits the forwarding of messages with a "time-to-live" (TTL) value which is part of the message payload.

The interface of the device establishes the maximum throughput of its wireless connection, and this is used to simulate the message exchange among peers. Care was taken to ensure that a message being sent at the end of a simulation cycle is received at the start of the next cycle. We represent peer connections and amount of message exchanges in the graphic interface, and all messages are logged for off-line analysis. We cater for situations in which a message is sent to a peer who has moved to outside the communication radius, so the message will not be received.

The bandwidth between two devices is calculated using linear interpolation of known bit rates at set distances. This is simple to implement and allows users to customise the granularity of the bandwidth over distance calculation, depending on the required accuracy of the simulation. We can track the amount of bandwidth that the device has used to ensure that it can never over allocate bandwidth during a single simulation step. As sending a message from one device to another requires that both have some available bandwidth and because devices are multi-threaded, this aspect of the simulation must be thread safe. If one device has less bandwidth available than the other the lower value is used so that the device with more bandwidth does not over allocate resources.

The road simulation is handled using TraCI4J and SUMO. While this simulation is computationally demanding, there are a few performance improvements that can be made. The most important of these is that multi queries are used when updating the position of devices. Multi queries allow many individual position queries to be bundled into a single larger query, reducing network latency and overhead as fewer packets are sent across the network. This also allows SUMO to process the queries faster than several individual queries.

The performance can also be improved by sending TCP IP data packets individually instead of being

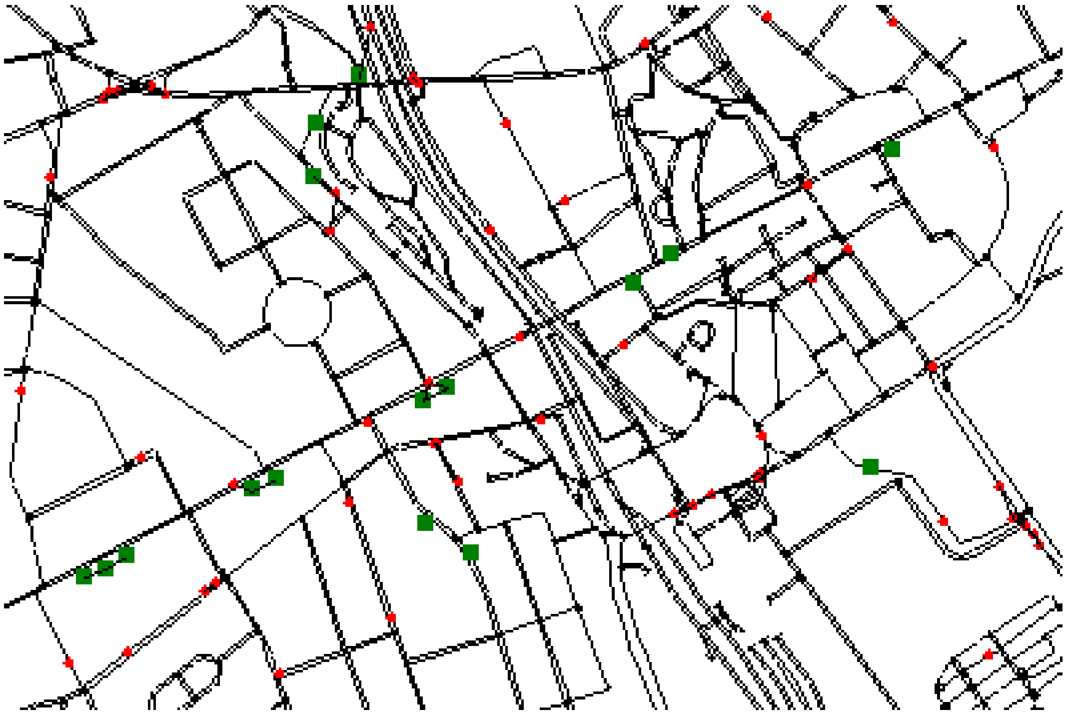


Figure 2: Image of the Simulation Running

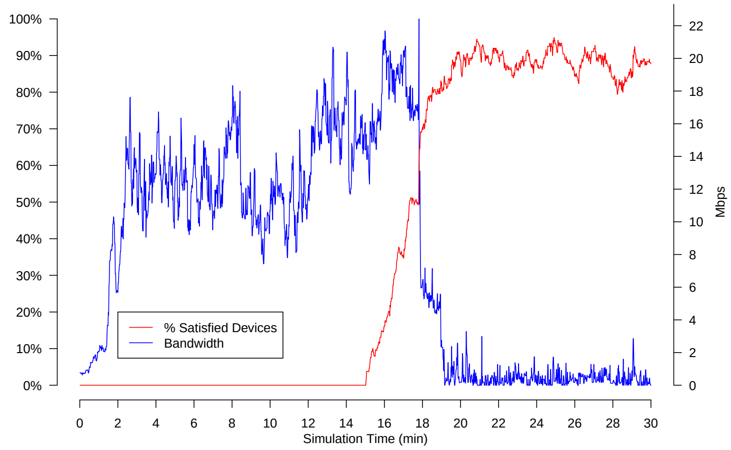
grouped together. The grouping of TCP packets is a method to reduce bandwidth, but in the case where the SUMO instance is running using the local loop back interface it causes a significant delay (over 100 milliseconds) on each query. When there is at least one query per simulation step it reduces the performance of the simulation to a maximum of 10 steps per second, without any devices in the simulation.

4. Scenario Modelling

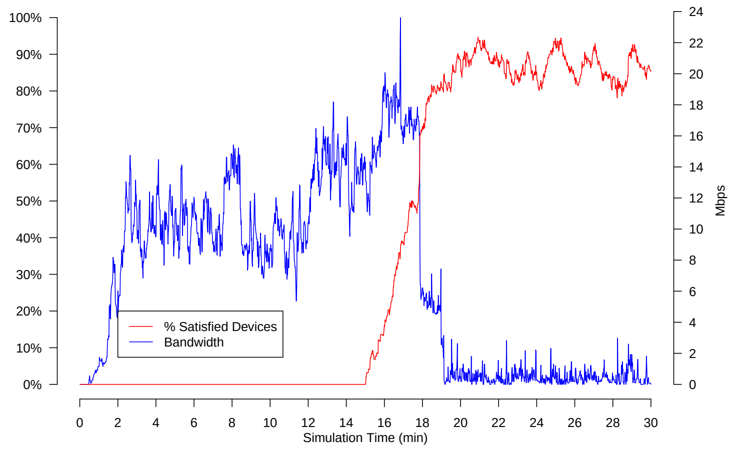
There is no existing test data set that is comparable to our simulation, so for the purposes of evaluation we created a series of simulation scenarios. An image of these simulations running in UrbanSim can be seen in Figure 2.

First a control scenario was created to provide a baseline for the proceeding simulation scenarios. The control simulation is designed to test the propagation of a small file through a group of devices operating with our designed behaviours within 30 minutes. As the simulated environment is initially not populated with vehicles carrying devices it was decided that the file would be released 15 minutes after the start of the simulation. The file was released from a device at a bus stop that is in roughly the centre of the simulated environment. Buses and cars both carry devices, but buses have set routes taken at regular intervals whereas cars have their routes randomly generated before execution. This single set of randomly generated routes is used in every simulation, meaning that the cars in every simulation make the exact same journey. We measure social welfare as indicated by those participants who got the file(s) they wanted/needed; these devices are deemed to be "satisfied".

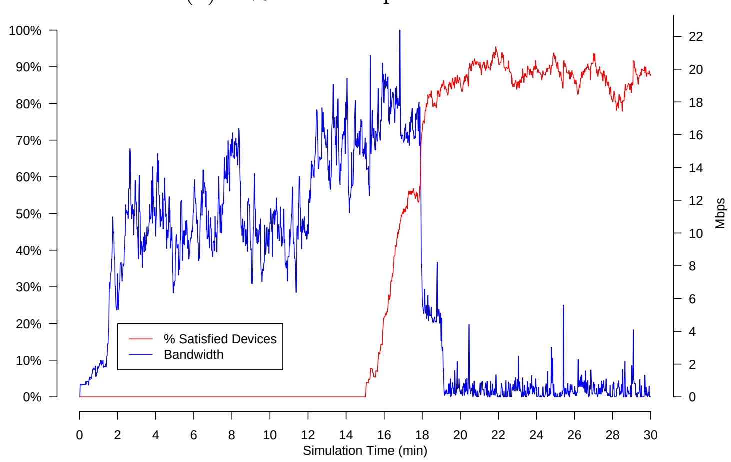
The results from the control simulation can be seen in Figure 3a. The control simulation shows that the bandwidth of the system increases as cars and buses carrying devices enter the simulation. This is caused by all the devices flooding the network with requests for the file. It also shows that there is variance of up to 5Mbps in the bandwidth, caused by movement of the cars and buses. There is a spike in the bandwidth after 8 minutes that is likely caused by cars or buses stopped at a busy intersection. The file is released at the chosen location at the 15 minute mark, after which the percentage of satisfied devices (those with the file) begins to climb. The bandwidth increases at this point due to the size of the file travelling over the network, which is larger than a simple file request. When the satisfied devices ratio reaches 50% the bandwidth of the system significantly drops, as once a device gets the file they cease sending file requests. It is also likely



(a) Control Simulation

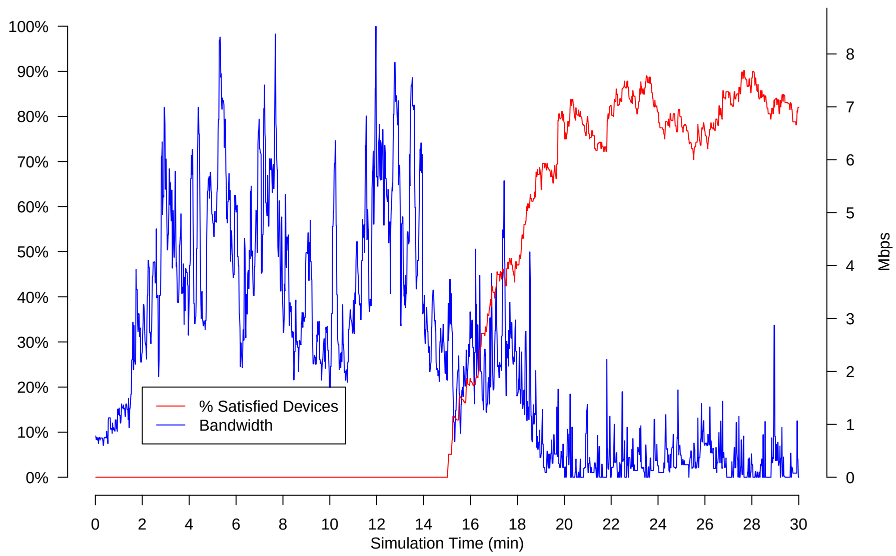


(b) 50% of bus stops with devices

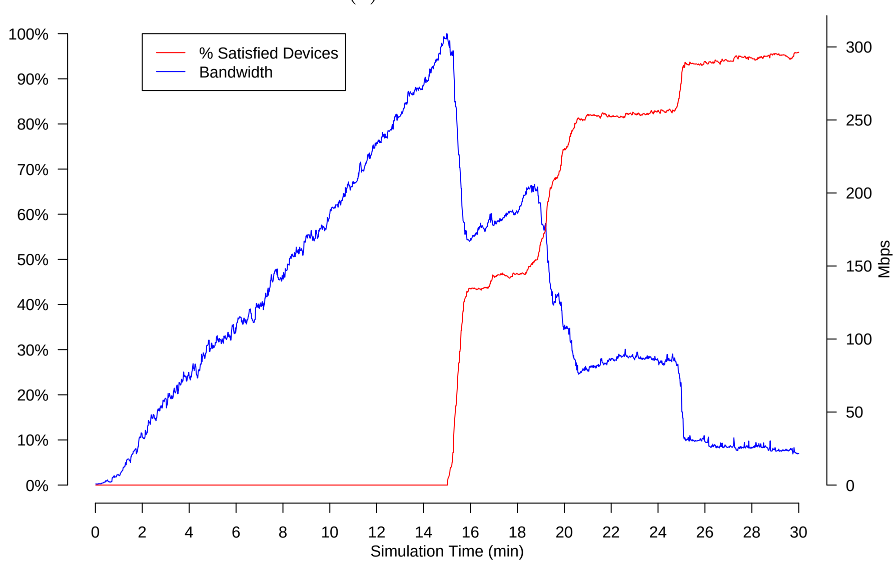


(c) Buses without devices

Figure 3: Scenario Results: (a) Control Simulation, (b) 50% of bus stops have devices, (c) Buses do not devices



(a) 50% fewer cars



(b) 650% more cars

Figure 4: Scenario Results: (a) Simulation run with 50% fewer cars, (b) Simulation run with 650% more cars

that any requests made at this point are have to traverse significantly less devices to reach one in possession of the file. The bandwidth continues to decrease and the number of satisfied devices increases until around 20 minutes into the simulation. At this point around 85% of all devices have the file and are satisfied. The bandwidth then remains below 3 Mbps as only new cars or buses entering the simulation are requesting the file.

The next scenario that was tested had only 50% of the bus stops fitted with a device, the results are shown in Figure 3b. There is very little change between this experiment and the control, possibly because the bus stops in this test do not have a large contributing factor to the file propagation.

The third scenario removes devices from buses, giving the results in Figure 3c. Again there is very little change in results, however bandwidth before the file released is about 1.5 Mbps less and after release there are more bandwidth spikes.

The fourth scenario has 50% fewer cars in the simulation. The results in Figure 4a show that reducing the number of cars had a significant effect on the bandwidth of the simulation. Before the file is released bandwidth is down to around 4 Mbps from 12 Mbps in the control. It also takes around 4 minutes longer for the file to reach approximately 80% of the devices. This indicates that cars in this simulation transport a significant proportion of the data.

The final scenario was used to test how the simulator copes with a very large number of devices, with the results as shown in Figure 4b. In this case there were an average of 668 devices in the simulation compared to the control average of 112 devices. The large number of devices on the road network resulted in the main roads becoming congested, with traffic at a stand still in many areas. The simulator was able to cope with the large number of devices, however it did take much longer to run the simulation. At the peak, the peer-to-peer network was using around 325Mbps of bandwidth. It is interesting to see in the simulation that the file only took 6 minutes to reach around an 85% of the devices, this then continued to increase towards 100%. The high satisfaction rate is most likely attributed to the fact that all devices were stuck in traffic next to each other, compared to the other scenarios where devices are rapidly passing each other.

4.1. Scalability

The primary limiting factor of this system is the number of devices present in a simulation. This directly impacts many other components of this system, and causes the amount of memory and processor time required to run the simulation to increase significantly.

The impact to processing time can be reduced by using more powerful hardware. With the simulator's ability to increase the number of threads that it uses, the hardware can be effectively utilised. This greatly reduces the execution time when a significant number of devices are simulated. The simulation is however limited to a single physical machine.

The number of mobile devices also affects the requirements of the road network simulator SUMO. It is however possible with some modification of the project to move the SUMO simulation onto a separate machine. This would introduce a limiting factor of network bandwidth between the road and main simulations.

The behaviour of the agents on the devices also affects the execution time of the simulation. An agent that is asleep will not require any processing time, neither will it send any messages for other devices to process.

Table 1 was created using some of the data from the above scenarios. It shows the increase in processing time in relation on the number of devices in the simulation. Included is the peak bandwidth for each simulation and the amount of data generated.

Scenario	Avg. Devices	Runtime	Peak Bandwidth	Data Generated
Control	112	62s	23 Mbps	420 MB
50% fewer cars	64	42s	9 Mbps	140 MB
650% more cars	668	579s	325 Mbps	6041 MB

Table 1: Data for 30 simulated minutes

5. Another (more sophisticated) Scenario: Advert Auctions

We made use of Urbansim to model and simulate advert auctions within a smart city [8]. In this scenario, companies and organisations create their advertising campaigns (for instance, electronic versions of posters to be displayed on billboards, videos with or without sound, and so on) and the peer-to-peer mechanisms will disseminate these advertisements within the city. However, the dissemination will be in a controlled fashion, respecting the context of the device, the nature of the advert, and other restrictions. Moreover, we consider a business model whereby companies and organisations pay each time their advertisements are shown. We address the situation when there might be competition for space (on mobile devices, billboards, screens in buses, etc.) and the advertisements are equipped with a budget and means to take part in auctions to gain access to space.

We developed this more complex scenario with our tool to show how we can model individual components with arbitrary functionalities. We explore the design of a distributed mechanism for managing advertisements in a smart city, making use of peer-to-peer technologies to support auctions. Our advertising mechanisms support various functionalities such as the billing, the advert and simulated device models, the auction mechanism itself, as well as the peer-to-peer mechanisms and functionalities to log results. The overall system behaviour is defined by these components and their fine-grained configuration.

In order to make decisions about which adverts should be shared or displayed, a selection mechanism is required. This mechanism would assign adverts to neighbouring devices, in accordance with specified rules/conditions established by the companies and organisations managing their advertisement campaign. Our mechanism also ensures that devices receive adverts that are most appropriate to them (e.g., a video with sound should not be shown on a bus-shelter billboard).

There are many ways to determine which adverts should be sent and to whom and which adverts should be displayed. Our information model caters for, for instance, the advertisement's release date, expiry date and *geofence*[8]. As there might be more than one advertisement to be displayed at any one moment on a particular device, an appropriate selection mechanism is called for. Inspired by on-line auctions for advertisement space as done by, among others, Google[9], Yahoo[10] and GMail[11], we explored a simple auction mechanism to support advertisments "competing" with one another for space/time on a device. Not only this was a natural way to support fair competition among companies and organisations, but it creates a business model whereby those providing the smart city computational infra-structure could have access to revenue. We explain the auction mechanism in Section 5.1.

To further explore business and economical aspects, we created a system of currency based on nominal credits, thus enabling us to emulate any actual currency. Each copy of an advertisement carries with itself credits it uses to take part in auctions. When an advertisement wins an auction it pays whatever it bid (in credits) to the device. Devices accrue credits from the advertisements which are successful in auctions and are selected to be displayed.

Advertisements need a way of bidding in auctions. For simplicity, our advertisements are created with a single bid value which remains constant. This bid is the maximum amount of credit the advertiser is willing to spend per visualisation of the advert. However, having an overall budget and bid value enable the designer of the advertisement campaign to have a concept of limited currency and expenditure; these also foster competition among advertisers.

5.1. Auction mechanism

We have modelled the Generalised Second Price (GSP) auction [9]. A more detailed discussion of the GSP auction is given in Section 6. We briefly describe how our solution works, full details of the algorithm can be found in [8]. When an auction takes place, adverts are ranked in descending order according to their bid to fill the display slots on the device. Each advert is charged the amount of the next highest bid that is less than the bid of the advert itself. In the case that there are fewer adverts than display slots, the last advert is charged a reserve price and the auction finishes with some display slots unallocated.

[8] A geofence is a virtual boundary representing the geographic limits within which an advert can move/travel among devices.
[9] `http://www.google.com`
[10] `http://www.yahoo.com`
[11] `http://mail.google.com`

Consider a device, D with three display slots. Device D holds adverts A and B with bids 4 and 2 respectively. Advert A would be charged 2, which is the bid of B, while B would be charged the reserve price since there are three display slots but only two available adverts.

Various different aspects of the auction mechanism were considered especially in relation to repeated auctions. To increase diversity of advertisements to be displayed as well as to prevent abuse from rich companies or organisations (with generous budgets for their advertisement campaigns), an advert could be prevented from taking part in auctions for a period of time if they have been successful previously. Another possible variation to increase diversity is for devices/peers to disseminate copies of advertisements to their neighbours, and refuse to receive them if they get sent back.

The messages used in the peer-to-peer mechanism are relatively simple. They consist primarily of two fields, namely, a message type and a message payload. The message type allows the protocol to determine how the message should be processed once it is received. The message types and applicable payloads are detailed in Table 2. The payload of a message is a string, consisting (in all but the first type) of a serialised object such as the actual advertisements, billing records and device details. We note that since advertisements have expiry dates, we decided to not make use of a time-to-live (TTL) when propagating advertisements within a smart city. The REQUESTDETAILS message does not have any payload. The acknowledgement messages, namely, ADVERTACK and BILLINGACK, require a payload of the object that they are acknowledging, otherwise we cannot guarantee what is being acknowledged.

Message type	Payload
REQUESTDETAILS	null (no payload expected)
DEVICEDETAILS	DeviceDetails Object
ADVERT	Advert Object
ADVERTACK	Advert Object from previous ADVERT message
BILLINGRECORD	BillingRecord Object
BILLINGACK	BillingRecord Object from previous BILLINGRECORD message

Table 2: Message types and their payload

5.2. *Peer-to-Peer protocol*

The following description relates two devices, A and B showing how they interact via our protocol. Although only two devices are being used as an example, multiple devices can follow the protocol simultaneously, with messages being sent to and received from many peers/devices.

1. Device A initiates a connection with all neighbours and sends a REQUESTDETAILS message.
2. Device B responds with a DEVICEDETAILS message, including a DeviceDetails object in the payload.
3. Device A can then send an ADVERT message, the payload contains a single advert object.
4. Device B receives the advert and responds with an ADVERTACK message, the payload is the same as the ADVERT message.
5. Device A can also send a BILLINGRECORD message, the payload contains a BillingRecord object.
6. Device B then replies with a BILLINGACK message, the payload contains the same BillingRecord object.

Steps 1 and 2 must be the first steps that are executed, sending the associated messages. These form the first stage of the protocol (details exchange), and without this interaction the rest of the protocol cannot proceed. Once this interaction has occurred, the device may execute the following stage comprising step 3 followed by step 4 (advert exchange) or step 5 followed by step 6 (billingrecord exchange).

The acknowledgement messages are essential for the protocol as these messages allow devices to maintain only one copy of an advert or a billing record in the network of peers/devices. If an acknowledgement is not received, it is assumed that the message was not successfully sent and it will be recreated and sent to a different peer. These acknowledgement messages were put in place because during early versions of our modelling, advertisements were unaccounted for and the overall number of advertisements decreased rapidly.

5.3. Results of Simulation

We designed a series of experiments with our model, to study the dynamics of the simulations. The results are discussed below. Figure 5a shows the increase in revenue accrued by devices as the overall advert

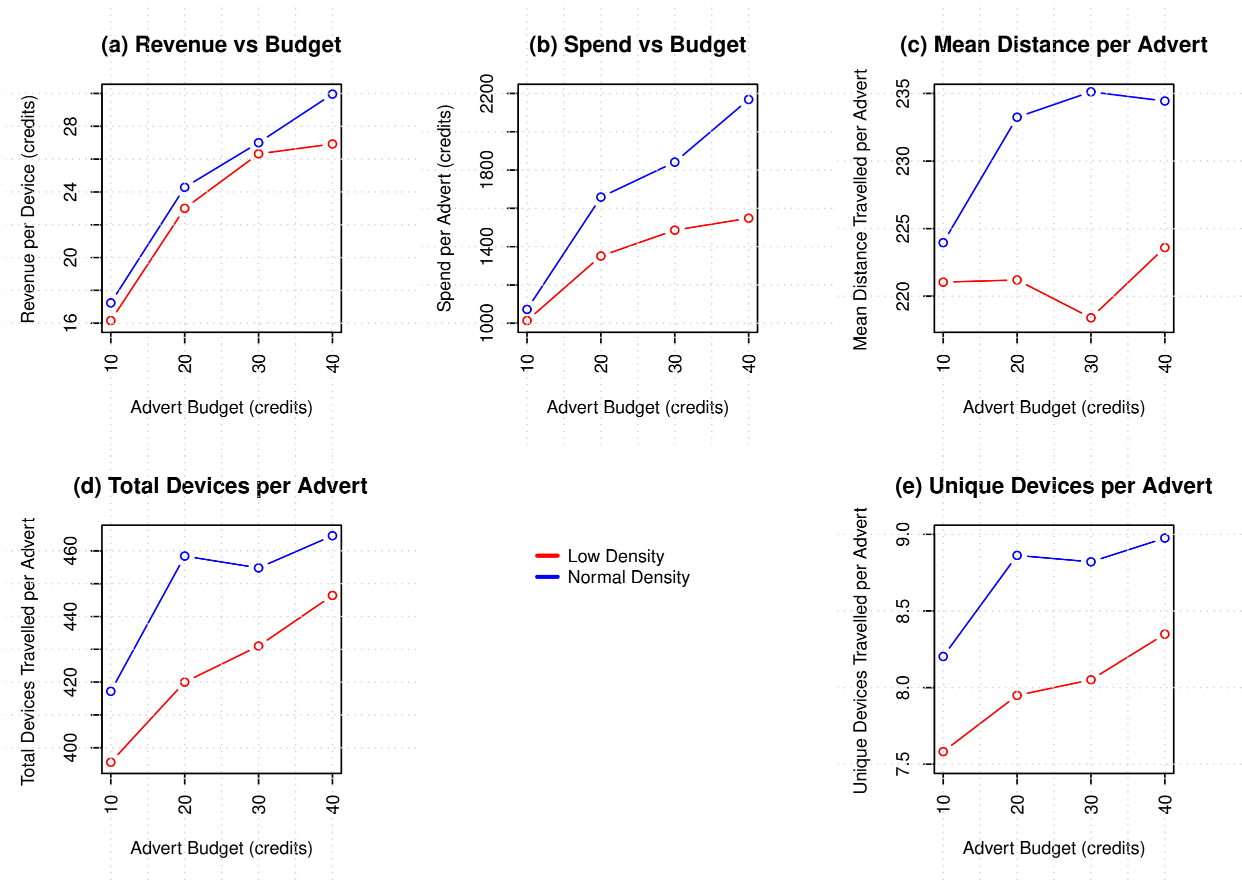


Figure 5: Evaluation Results

budget increases. As the budget starts out at 10 credits, both densities show devices accruing a moderate amount of credit. Devices accrued a mean of 17.26 credits in the denser network and 16.14 credits in the sparser network, giving a difference of 1.12. After the budget increases to 20 credits, the difference between the densities remains similar.

Once the budget reaches 30 or 40 credits, an interesting observation can be made. With a sparser network, increasing the budget from 30 to 40 has much less of an effect than in the denser network implying that a limit has been reached. This is most likely because in the sparser network, there are fewer devices to hold auctions, and therefore the advert budget is less likely to be depleted at the same rate as it would in a denser network. This is in line with what can be observed from using an advert budget of 40 in a denser network. The revenue accrued by devices increases as expected, with devices accruing a mean of 29.97 credits in the denser network.

WV: Can we say something about the density – how many peers per 10 square metres, etc.?

SC: Density apparently represents peers in the network. Alex couldn't find exact numbers, but knew it was in the tens as opposed to the hundreds. The low might be somewhere around 16, or perhaps 21

Figure 5b shows the increase in advert spending as the overall advert budget increases. The spend of each advert provides some interesting information about the behaviour of the network. An important observation to make is that the Y-axis scale begins at 1000, despite the fact that the lowest value used for the advert budget was 10 credits. This indicates that the budgeting mechanism for advertisements did not work as intended.

Figure 5c, shows that the mean distance travelled per advert remains mostly unchanged at around 221 metres regardless of the value of advert budget, with the exception of when the budget is 30 where it drops to approximately 218 metres. Considering that the mean distance travelled in the sparse network with an

WV: This is not good... We could omit the graph and delete this paragraph...

SC: If we do this, do we need to change the other graphs? They all use advert budget as their x-axis, and this value seems to be inaccurate based advert budget of 40 is very similar to when the budget is 10 or 20, the result for a budget of 30 could be considered anomalous and may require repeated testing.

The comparatively low values for distance travelled when the advert budget is 10 are to be expected. With a small budget, the budget will be spent and the advertisement will be removed sooner than if its budget had been greater. When the budget is 20, 30 or 40, the denser network appears to allow adverts to travel a greater average distance. Since adverts are not restricted in terms of which devices they can be sent to, it is not surprising that the budget does not seem to have a strong effect on how far the adverts travel, with the exception of low budgets such as the first result in Figure 5c.

Figures 5d and 5e respectively show how many unique devices and non-unique devices adverts were sent to. This gives an overall indication of an adverts propagation as well as how many times it was shared. The proportions between the individual points of Figures 5d and 5e appear to be similar in both graphs.

It is interesting to observe that with each increase in their budget, advertisements travel through more devices in the sparser network. A possible reason for this is that in a sparser network, there are fewer opportunities to exchange adverts with other devices. As a result, it is likely that there will be less variety of adverts available for the auction, and therefore less competition. If an advert is unable to be transferred to another device, and it frequently succeeds in auctions, its budget will deplete at a faster rate. Assuming that this is the case, adverts may be less likely to propagate if their budget is not large enough.

6. Related work

Exploring other work relating to peer-to-peer computing in smart city scenarios reveals a varied landscape of current and proposed work. One such area of research relates to traffic and travel in a smart city, with topics such as sensing anomalies in traffic data [10, 11, 12], real-time travel-time estimation [13, 14], personalised directions adapting to real-world conditions [15] and intelligent taxi choice in a dispatch/scheduling system [16, 17, 18, 19]. Other research has considered city-wide statistics, such as air quality [20, 21], noise pollution [22, 23, 24], fuel consumption [25] and electricity consumption (for electric vehicles) [26, 27, 28, 29, 30]. There has also been research into the social applications of smart cities [31, 32, 33], how disaster scenarios can be supported by smart cities [34, 35], where retail stores should be placed [36], and how to predict the price of real estate within smart cities [37].

Our research work also relates with mobile peer-to-peer networks. Mobile Chedar [38] is an extension to Chedar (CHEap Distributed ARchitecture) middleware [39], designed to allow mobile phones to access the Chedar peer-to-peer network and to also create Chedar peer-to-peer networks between other mobile phones. To do this it uses the mobile phone's Bluetooth[12] features to communicate to Chedar gateway peers. Gateway peers are traditional networked machines that act as Bluetooth wireless access points. It is also possible for Mobile Chedar to create a network between the Bluetooth of other mobile phones. There is a limitation of Bluetooth that mandates a star topology, with a master device and slave devices. It is normal for the gateway peers to be the master devices in Mobile Chedar. This allows them to connect to the conventional network and then route messages to other gateway peers, who then forward the messages through Bluetooth to other mobile devices.

Previous research used social computing and an extension of Gnutella to create a mobile peer-to-peer network [40]. The research uses a super-peer architecture, with better performing peers being given more responsibility in the network. In this project the super peers are conventional desktops machines. They have significantly more processing power, storage and bandwidth than mobile peers. The super peers are then used to assist the mobile peers in resource intensive tasks. The mobile peers can connect to each other and super peers through the wireless network. Mobile devices are only able to communicate through this network and any searches that they make will be passed to a super peer for processing.

Vehicular ad-hoc networks (VANET) are networks created between vehicles as they move through a road network. For the testing of these VANETs accurate road and traffic simulation is needed. This has lead to the development of many tools, that accurately simulate road networks on a large scale. A common goal of VANET research is to reduce road congestion, through efficient management of traffic signals. The ad-hoc network created between vehicles often use peer-to-peer technologies to allow data propagation between

[12] `https://www.bluetooth.com`

vehicles. This data is then passed to the traffic signals themselves which act on the data or send the data to another device, allowing it to be transferred through a conventional network to traffic controllers [41].

Exploration of VANETs using ad-hoc networks in conjunction with cellular phone networks [42] has shown their usefulness and feasibility as the price of cellular data continues to decrease. This type of VANET has potential, as car manufacturers are increasingly integrating mobile phone technologies into their cars[13]. This potentially furthers the reliance on mobile network infrastructure, but will have the benefit that newer cars will have wireless connectivity.

There are many existing proposals for the simulation of peer-to-peer networks. They are often very similar, with the majority being developed for research and utilising existing open source projects. PeerFact[14] started development in 2005 and has been developed since 2006 by the German Research Society with the goal of benchmarking P2P systems. The simulator is also being used to research P2P-based service-oriented architectures [43]. With nearly a decade of development PeerFact has been used in a range of research[15]. It has a wide range of features such as an ISO-style stack for network communication and recently they have added support to the simulation for modelling power consumption and movement of agents in a geospatial environment [44]. The project is focused on being modular and open, hence its active development. PeerFact has split the development of P2P technologies along clear boundaries thus allowing for individual components to be changed and adjusted, but this also complicates the design of any potential protocol that one may want to test. Additionally, any protocols that are developed would require refactoring for a real device as there is a large contrast between them. The latest version of PeerFact uses its own geospatial network simulation – this could have potential benefits, but adds another level of complexity when developing a simulation. Our project uses an already tried and tested road network simulation with the aim of using existing technologies to reduce complexity and the potential for bugs.

The auctions used by various search engines, such as Google and Yahoo, to power their advertising platforms have been studied relatively well [45, 46]. Such auctions and their effects are very important, since search engine companies such as Google make vast amounts of their income from advertising. For instance, in 2014, Google made over $59 billion in advertising revenue[16]. Therefore, it makes sense to carefully study the processes involved given the amount of money concerned. The Generalised Second Price (GSP) auction is commonly used for selecting which adverts should appear alongside search results [9]. The GSP auction is an extension of Vickrey second price auction [47] designed for multiple items, each bidder places a single secret bid. Each bidder pays the price of the bid below them: 1^{st} highest pays the 2^{nd} highest price, 2^{nd} highest pays the 3^{rd} highest price, ..., n^{th} highest price pays the $(n+1)^{th}$ highest price. Unlike the Vickrey auction, bidding honestly is not always a dominant strategy in GSP auctions [9].

The GSP auction itself has been explored in many ways, including various assumptions about bidder independence, bidder budgets [48] and also CTRs along with bidder and slot matching [49, 48]. Additional work was carried out on the the peer-to-peer auction mechanism, placing more of an emphasis on currency [50]. In this system, adverts spend "credits" to be carried by devices, and the owners of these devices can convert these credits into rewards.

In [51] it is reported a knowledge-intensive peer-to-peer mechanism to support computer games. This approach is generic, in that game "logic" and knowledge about the game are captured via general-purpose production rules. Such an approach can be adapted and extended to make use of our smart city infrastructure to reduce message-passing, improve quality and timeliness of information/knowledge shared, decide on information and knowledge to volunteer and request, and facilitate knowledge storage (what to keep, what to delete, what to summarise). Information and knowledge should be represented in a machine-processable format, adopting existing standards and making use of current technologies stemming from research on Semantic Web[17]. We envisage the use of explicit if-then-else rules such as those of "If-this-then-that" (IFTTT[18]) which, in addition to being used to decide on courses of action, can also be manipulated as explicit knowledge, and shared, deleted, updated, and stored.

[13] `http://www.openautoalliance.net/`
[14] `https://sites.google.com/site/peerfactsimkom/home`
[15] `https://www.sites.google.com/site/peerfactsimkom/publications`
[16] `https://abc.xyz/investor/pdf/2014_google_annual_report.pdf`
[17] `https://www.w3.org/standards/semanticweb/`
[18] `https://ifttt.com/`

We have already explored various technical and technological aspects of the proposed research. In [52] we proposed a solution for distributed wireless ad-hoc networks of mobile phones via Bluetooth and peer-to-peer mechanisms. In [53] we developed a game for mobile phones which simulates post-disaster scenarios; players have to interact with one another via peer-to-peer mechanisms in order to perform better in the game. In [54] we presented means to support disaster simulation games for mobile phones. The proposed research will make use of core technologies and design features of our previous work, combining, extending and adapting them as needed.

7. Discussion, conclusions & future work

We have presented a tool for simulating peer-to-peer communication within a smart city environment. We have illustrated the use of this tool with two scenarios, one demonstrating information propagation through a smart city, and the other showing distributed advert auctions.

One aspect of our solution that is lacking is the feasibility of the peer-to-peer networks. The simulated peer-to-peer network tested in the evaluation phase would be feasible in the simulated environment. However, for the simulation to be truly accurate in a real world context it would require significant development of the wireless bandwidth calculations (see Section 3.2). Additional details, such as radio shadowing[19] would also need to be accounted for.

This research has some interesting features that are not present in existing solutions and has met the requirements presented in Section 2. However, there are aspects which could be improved. For instance, sending messages between devices needs to be re-implemented, as we detected an intermittent issue with bandwidth over-allocation, whereby a device is able to allocate more bandwidth than it should be able to. This is likely due to the time slicing model adopted in the simulation. To resolve this, we intend to move to an event-based model.

When it came to using Urbansim for the advert auction scenario, it was intended that no further development would be required; however some modifications did have to be made. The first change introduced was to expose an existing method to get a device's current position, as we require this position to enable geofencing for adverts. This method had to be declared in the device interface provided by Urbansim to agent classes in order to be used. The second change made was to enable Urbansim to run without a GUI (Graphic User Interface). The GUI would not be needed during the automated testing, therefore the simulation had to be controlled entirely from the command-line. Finally, the XML logging feature of the simulator was turned off along with any unnecessary console printing. This particular feature was not being used and was creating large amounts of unused data per simulation.

This research creates an entry point by developing an interesting and novel simulation environment, that we envisage developing further in the following areas. Our current model of wireless communications in the simulation is greatly simplified. It would be interesting to modify communication to add a calculation of wireless signal strengths through different media. A line of sight between wireless antennae, for instance, has significantly better throughput than that of a connection that must pass through a building. We would also like to make it possible for a device to have multiple types of modes of connection, since it is common for a mobile phone to have 3/4G, WiFi and Bluetooth. Moving to an event based model would require an event subsystem to be created. Doing this will make synchronisation with the SUMO simulator non-trivial and events for messages sent between devices would methods to update them in transit.

In the future we could explore our work in the context of two social science paradigms. Mobilities research, which explores the "interdependent digitised systems of mobility" that are becoming common in the 21st century [55], will help us think through the social relations of movement in the city. A specific example could be to look at the connection between speech acts and pedestrian trajectories in the philosophy of de Certeau [56]. Science and Technology Studies (STS), meanwhile, would enable us to examine pedestrian activity in the light of the kinds of speech enabled by mobile devices (*e.g.*, digital messaging), and the creation and uptake of the technology through social and technological networks [57]. Using these approaches, we could look at how people are influenced by data provenance, namely who created the knowledge or information,

[19] Radio shadowing refers to fluctuations in received signals due to objects obstructing the path between sender and receiver.

how this was disseminated (*e.g.*, path, speed, who was involved), if there were changes, who carried these out, and so on.

The research inquires into the vision of "smart cities" in which knowledge and information should be everywhere, for everyone. Individuals would consume, intermediate, transform, store and, ultimately, create knowledge and information. Our simulation tool would enable the study, from anthropological and social science perspectives, of the motivations that people may have for taking an active part in a knowledge infrastructure, while at the same time remaining open to the possibility that people may choose not to share information in a peer-to-peer fashion. The dynamic between sharing and not-sharing could be the focus for case studies in the future.

[1] M. C. Castro, A. J. Kassler, C.-F. Chiasserini, C. Casetti, I. Korpeoglu, Peer-to-peer overlay in mobile ad-hoc networks, in: Handbook of Peer-to-Peer Networking, Springer, 2010.

[2] R. Rodrigues, P. Druschel, Peer-to-peer systems, Communications of the ACM 53 (10) (2010) 72–82.

[3] I. J. Taylor, A. Harrison, From P2P and Grids to Services on the Web – Evolving Distributed Communities, Springer, 2009.

[4] A. Caragliu, C. Del Bo, P. Nijkamp, Smart cities in Europe, Journal of Urban Technology 18 (2) (2011) 65–82.

[5] D. Gibbs, R. Krueger, G. MacLeod, Grappling with smart city politics in an era of market triumphalism, Urban Studies 50 (11) (2013) 2151–2157.

[6] D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, Recent development and applications of SUMO - Simulation of Urban MObility, International Journal On Advances in Systems & Measurements 5 (3–4).

[7] A. Tiago M. L., D. A. Paulo J. M., R. Rosaldo J. F., R. Ana Paula C., JADE, TraSMAPI and SUMO: A tool-chain for simulating traffic light control, in: 8th International Workshop on Agents in Traffic and Transportation, AAMAS, 2014.

[8] A. Stuart-Kregor, A peer-to-peer mechanism to support advertisement auctions in smart cities, Final Year Dissertation, Department of Computing Science, University of Aberdeen (2015).

[9] B. Edelman, M. Ostrovsky, M. Schwarz, Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords, Working Paper 11765, National Bureau of Economic Research (November 2005). doi:10.3386/w11765.
URL http://www.nber.org/papers/w11765

[10] B. Pan, Y. Zheng, D. Wilkie, C. Shahabi, Crowd sensing of traffic anomalies based on human mobility and social media, in: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2013, pp. 344–353.

[11] L. X. Pang, S. Chawla, W. Liu, Y. Zheng, On mining anomalous patterns in road traffic streams, in: Advanced Data Mining and Applications, Springer, 2011, pp. 237–251.

[12] W. Liu, Y. Zheng, S. Chawla, J. Yuan, X. Xing, Discovering spatio-temporal causal interactions in traffic data streams, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 1010–1018.

[13] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, J. Eriksson, VTrack: accurate, energy-aware road traffic delay estimation using mobile phones, in: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, ACM, 2009, pp. 85–98.

[14] Y. Wang, Y. Zheng, Y. Xue, Travel time estimation of a path using sparse trajectories, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 25–34.

[15] J. Yuan, Y. Zheng, X. Xie, G. Sun, T-Drive: Enhancing driving directions with taxi drivers' intelligence, IEEE Transactions on Knowledge & Data Engineering 25 (1) (2013) 220–232. doi:10.1109/TKDE.2011.200.

[16] S. Phithakkitnukoon, M. Veloso, C. Bento, A. Biderman, C. Ratti, Taxi-aware map: Identifying and predicting vacant taxis in the city, in: Ambient Intelligence, Springer, 2010, pp. 86–95.

[17] K. Yamamoto, K. Uesugi, T. Watanabe, Adaptive routing of cruising taxis by mutual exchange of pathways, in: Knowledge-Based Intelligent Information and Engineering Systems, Springer, 2008, pp. 559–566.

[18] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, M. Pazzani, An energy-efficient mobile recommender system, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 899–908.

[19] X. Xie, Y. Zheng, L. Zhang, N. J. Yuan, T-Finder: A recommender system for finding passengers and vacant taxis, IEEE Transactions on Knowledge & Data Engineering 25 (1) (2013) 2390–2403. doi:doi.ieeecomputersociety.org/10.1109/TKDE.2012.153.

[20] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, B. Nath, Real-time air quality monitoring through mobile sensing in metropolitan areas, in: Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing, ACM, 2013, p. 15.

[21] X. Chen, Y. Zheng, Y. Chen, Q. Jin, W. Sun, E. Chang, W.-Y. Ma, Indoor air quality monitoring system for smart buildings, in: Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2014, pp. 471–475.

[22] S. Santini, B. Ostermaier, A. Vitaletti, First experiences using wireless sensor networks for noise pollution monitoring, in: Proceedings of the Workshop on Real-World Wireless Sensor Networks, ACM, 2008, pp. 61–65.

[23] N. Maisonneuve, M. Stevens, M. E. Niessen, L. Steels, Noisetube: Measuring and mapping noise pollution with mobile phones, in: Information Technologies in Environmental Engineering, Springer, 2009, pp. 215–228.

[24] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, W. Hu, Ear-phone: an end-to-end participatory urban noise mapping system, in: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, ACM, 2010, pp. 105–116.

[25] F. Zhang, D. Wilkie, Y. Zheng, X. Xie, Sensing the pulse of urban refueling behavior, in: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2013, pp. 13–22.

[26] I. Dusparic, C. Harris, A. Marinescu, V. Cahill, S. Clarke, Multi-agent residential demand response based on load forecasting, in: Proceedings of the 1st IEEE Conference on Technologies for Sustainability, IEEE, 2013, pp. 90–96.

[27] A. Marinescu, I. Dusparic, C. Harris, V. Cahill, S. Clarke, A dynamic forecasting method for small scale residential electrical demand, in: Neural Networks (IJCNN), 2014 International Joint Conference on, IEEE, 2014, pp. 3767–3774.

[28] E. Galván-López, A. Taylor, S. Clarke, V. Cahill, Design of an automatic demand-side management system based on evolutionary algorithms, in: Proceedings of the 29th Annual ACM Symposium on Applied Computing, ACM, 2014, pp. 525–530.

[29] C. Harris, R. Doolan, I. Dusparic, A. Marinescu, V. Cahill, S. Clarke, A distributed agent based mechanism for shaping of aggregate demand on the smart grid, in: International Energy Conference, IEEE, 2014, pp. 737–742.

[30] M. Momtazpour, P. Butler, M. S. Hossain, M. C. Bozchalui, N. Ramakrishnan, R. Sharma, Coordinated clustering algorithms to support charging infrastructure design for electric vehicles, in: Proceedings of the ACM SIGKDD International Workshop on Urban Computing, ACM, 2012, pp. 126–133.

[31] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, W.-Y. Ma, Mining user similarity based on location history, in: Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2008, p. 34.

[32] C.-C. Hung, C.-W. Chang, W.-C. Peng, Mining trajectory profiles for discovering user communities, in: Proceedings of the 2009 International Workshop on Location Based Social Networks, ACM, 2009, pp. 1–8.

[33] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma, Mining interesting locations and travel sequences from GPS trajectories, in: Proceedings of the 18th International Conference on World Wide Web, ACM, 2009, pp. 791–800.

[34] R. Lee, K. Sumiya, Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection, in: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, ACM, 2010, pp. 1–10.

[35] X. Song, Q. Zhang, Y. Sekimoto, T. Horanont, S. Ueyama, R. Shibasaki, Modeling and probabilistic reasoning of population evacuation during large-scale disaster, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 1231–1239.

[36] D. Karamshuk, A. Noulas, S. Scellato, V. Nicosia, C. Mascolo, Geo-spotting: Mining online location-based services for optimal retail store placement, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 793–801.

[37] Y. Fu, H. Xiong, Y. Ge, Z. Yao, Y. Zheng, Z.-H. Zhou, Exploiting geographic dependencies for real estate appraisal: A mutual perspective of ranking and clustering, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 1047–1056.

[38] N. Kotilainen, M. Weber, M. Vapa, J. Vuori, Mobile Chedar: a peer-to-peer middleware for mobile devices, in: Proceedings 3rd International Conference on Pervasive Computing & Communications Workshops, IEEE, 2005.

[39] A. Auvinen, M. Vapa, M. Weber, N. Kotilainen, J. Vuori, Chedar: peer-to-peer middleware, in: Proceedings of the 20th International Parallel & Distributed Processing Symposium, 2006.

[40] H. M. Tran, K. Van Huynh, K. D. Vo, S. T. Le, Mobile peer-to-peer approach for social computing services in distributed environment, in: Proceedings of the Fourth Symposium on Information and Communication Technology, ACM, 2013, pp. 227–233.

[41] S. Dornbush, A. Joshi, Streetsmart traffic: Discovering and disseminating automobile congestion using VANET's, in: Vehicular Technology Conference, IEEE, 2007.

[42] J. Rybicki, B. Scheuermann, M. Koegel, M. Mauve, Peertis: A peer-to-peer traffic information system, in: Proceedings of the 6th ACM International Workshop on VehiculAr InterNETworking, ACM, 2009.

[43] D. Stingl, C. Gross, J. Ruckert, L. Nobach, A. Kovacevic, R. Steinmetz, PeerfactSim.KOM: A simulation framework for peer-to-peer systems, in: International Conference on High Performance Computing and Simulation, 2011, pp. 577–584.

[44] D. Stingl, B. Richerzhagen, F. Zollner, C. Gross, R. Steinmetz, PeerfactSim.KOM: Take it back to the streets, in: International Conference on High Performance Computing and Simulation, 2013, pp. 80–86.

[45] G. Aggarwal, A. Goel, R. Motwani, Truthful auctions for pricing search keywords, in: Proceedings of the 7th ACM Conference on Electronic Commerce, ACM, 2006, pp. 1–7.

[46] T. Qin, W. Chen, T.-Y. Liu, Sponsored search auctions: Recent advances and future directions, ACM Transactions on Intelligent Systems and Technology (TIST).

[47] W. Vickrey, Counterspeculation, auctions, and competitive sealed tenders, The Journal of Finance 16 (1) (1961) 8–37. `doi:10.1111/j.1540-6261.1961.tb02789.x`.
URL `http://dx.doi.org/10.1111/j.1540-6261.1961.tb02789.x`

[48] I. Ashlagi, M. Braverman, A. Hassidim, R. Lavi, M. Tennenholtz, Position auctions with budgets: Existence and uniqueness, The BE Journal of Theoretical Economics 10 (1).

[49] G. Aggarwal, S. Muthukrishnan, M. Pál, General auction mechanism for search advertising, in: Proceedings of the 18th International Conference on World Wide Web, ACM, 2009, pp. 241–250.

[50] L. K. Hristov, A peer-to-peer mechanism for intelligent adverts in smart cities, Final Year Dissertation, Department of Computing Science, University of Aberdeen.

[51] M. S. Gibson, A knowledge-based approach to multiplayer games on peer-to-peer networks, Ph.D. thesis, The University of Aberdeen (Jul. 2014).

[52] B. J. Buchan, Peer-to-peer networks using mobile devices and Bluetooth, Final Year Dissertation, Department of Computing Science, University of Aberdeen.

[53] A. Hoxhaj, Simulating a post disaster scenario through a collaborative peer-to-peer app for mobile devices, Final Year Dissertation, Department of Computing Science, University of Aberdeen.

[54] L. Douglas, Simulating a post-disaster scenario on mobile devices using centralised infrastructure to support ad hoc peer-to-peer networks, Final Year Dissertation, Department of Computing Science, University of Aberdeen.

[55] J. Urry, Mobilities, Wiley, 2007.

[56] M. de Certeau, The Practice of Everyday Life, University of California Press, 1984.

[57] L. Watts, Sand14: Reconstructing the future of the mobile telecoms industry, Fibreculture Journal 20.