MOBIX: System for managing MOBility using Information eXchange

Jhoanna Rhodette Pedrasa
University of New South Wales, Kensington, Sydney, Australia
National ICT Australia (NICTA), Australian Technology Park, Eveleigh, Australia
University of the Philippines, Diliman, Quezon City, Philippines
Email: j.pedrasa@student.unsw.edu.au

Aruna Prasad Seneviratne
National ICT Australia (NICTA), Australian Technology Park, Eveleigh, Australia
University of New South Wales, Kensington, Sydney, Australia
Email: a.seneviratne@nicta.com.au

***Abstract*—It is evident that mobile devices of the future will have multiple wireless interfaces. For small, energy-constrained devices, determining network availability by keeping all radio interfaces turned on at all times will negatively impact battery lifetime even when these interfaces are idle. Predicting future network availability from user history requires a period of training and learning user habits. This method will fail when users deviate from their routines constantly or move to locations not visited before.**

We propose a different approach to determining network availability of mobile nodes which leverages on the fact that nodes on the move will meet other nodes who will be able to share conditions of networks they have recently encountered. This paper presents MOBIX, a system where nodes exchange information about network conditions using short-range communication such as Bluetooth. Our simulation results show that the required number of nodes needed for 100% success is not unrealistic of densely populated metropolitan areas. Even with relatively low population densities, we can expect a data store hit more than 50% of the time. Although our evaluation used WiFi, our scheme can easily be extended for other technologies such as GSM and WiMax.

I. Introduction and Motivation

In the future mobile world, users will be carrying devices with multiple radio interfaces. These interfaces will have varying energy profiles and network characteristics, ranging from low power, low transmission range Bluetooth to relatively high power, longer range interfaces such a WiFi or WiMax. Despite rapid progress in battery technology, small, mobile devices of the future will still be energy constrained. Thus, turning on all wireless interfaces all the time even for the purpose of detecting available network points of attachments to decide which interface to use will significantly shorten the battery life of most portable devices.

Current methods rely mostly on the radio interfaces itself for detecting network availability. This limits decisions on what is the "best" point of attachment to radio layer parameters, such as RSSI. Another approach to the problem of determining network availability hinges on the repetitiveness of activities in user's lives, such as that used in [14]. Their method predicts future network conditions from past experiences. In our work, we explore a different approach altogether by having mobile users exchange reports of network conditions with other nodes they encounter using a short-range, low-power communication channel, such as Bluetooth.

Although it may seem counter-intuitive, we argue that there are benefits to such a system. Firstly, as previously mentioned, to turn on all of the multiple radio interfaces to monitor for current conditions will be an extreme drain on battery power. For instance, the WiFi interface can consume more than 60% of total system power, even when idle [12]. The Bluetooth interface on the other hand consumes very little power even when active. Secondly, only RSSI can be measured by relying on the radio layer. Throughput and delay can only be estimated based on inherent properties of the interface such as maximum theoretical bandwidth. On the other hand, in our proposed system we can share information such as actual throughput and delay experienced by other users on their connected networks. Finally, we can make better decisions as our information hinges not just on a single measurement at a specific point in time but on multiple measurements gathered by other devices nearby.

We have made the following contributions in this work:

- To the best of our knowledge, we present the first scheme to disseminate reports over the Bluetooth interface for the purpose of learning the network availability of other wireless interfaces, specifically WiFi.
- We studied the ratio of successful network connections using decisions based on reports alone. We show by simulations that at even at low number of mobile nodes, we can gather relevant data from other nodes more than 50% of the time. Additionally, we estimate that the required population density needed to have 100% data store hit is is not unrealistic for densely populated metropolitan areas.
- We show that the average age of useful reports is less than 10 minutes. Furthermore, it is possible to lower this value to less than one minute by adding more generator nodes.
- We demonstrate that because of the inherent low power

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6795
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6795

consumption of Bluetooth compared to WiFi even when active, we can achieve energy reductions of greater than 50%.

We briefly discuss related work in section 2 and the assumptions made in designing MOBIX in Section 3. We elaborate on system functions in section 4 and in section 5, we discuss how we evaluated the system. We share the results of our simulations in section 6 and finally, make our concluding statements in section 7.

II. Related Work

The problem of determining network availability has been the subject of much research. [14] used semi-Markov models to predict WLAN availability from user context such as time of day, GSM location area, available WLANs, and number of Bluetooth devices found. Similarly, [13] estimated WiFi network conditions using past WiFi and Cellular ID information and acceleration approximations. Prediction-based schemes require a period of learning however, and will fail when users deviate from their patterns or move to location not visited before. Our system may be used to complement these prediction schemes, as MOBIX does not need any prior knowledge and can work in unfamiliar environments as long as there are enough mobile nodes.

The work by [13] showed that network availability is very high in urban areas, but the energy cost of network interfaces is a significant problem. They thus proposed to leverage the complementary energy profiles of GSM and WiFi by developing policies for choosing between the two interfaces with the goal of extending battery life. Other works that seek to conserve energy of portable devices by switching between radio interfaces include [12] and [1]. Their approach however is aimed at conserving energy during data transfer, while our scheme seeks to minimize energy during the process of determining accessible networks.

Another approach to network availability involves the use of network maps or QoS maps. In [2], network coverage was modeled as a two dimensional map of the geographic coverage of each network. This network map was used in determining whether the node is in a transition zone and a vertical handover is warranted. A more sophisticated map was proposed in [4], where two dimensional representations of VoIP QoS metrics are made available to mobile VoIP users. These QoS maps are non-trivial to produce however and require a central server for the aggregation of map data. Our system on the other hand, is fully decentralized and does not require map generation.

Our techniques for information dissemination is closely related to similar work on vehicular ad-hoc networks (VANET). [9] examined selective data dissemination by estimating the novelty probability of reports from a spatio-temporal perspective. They concluded that age is a better indicator of novelty than distance. We chose not to use the combined ranking algorithm they presented as it requires sorting the data store twice and showed only slight improvements to results. Similarly, [22] used a ranking method based on perceived supply and demand of reports, with supply estimated using a machine learning algorithm.

Our work can be classified under mobile encounter networks, discussed in [8], a form of mobile peer to peer networks created by mobile devices. As with VANETs, their goal was information diffusion, ie broadcasting advertisements or gas-price alerts. As such, their work was focused more on optimizing data penetration, the percentage of users receiving the reports. Our system's goal on the other hand is not to inform as many nodes as possible, as the reports generated are not relevant to all the nodes but rather to a small subset of peers only.

III. Assumptions

We designed the system using the following assumptions:

- Devices have multiple radio interfaces with varying transmission ranges, bandwidth, and energy requirements. Each device will have at least one interface for short-range communication, eg Bluetooth or Zigbee.
- Over-all power consumption will be a major concern. Despite improvements in battery technology, mobile devices will still be energy constrained and minimizing energy requirements will thus be a priority.
- Processing power on the nodes will increase such that the mobile device can run monitoring tasks and other software in the background without negatively impacting the performance of other applications or the over-all power usage of the device. Decisions can also be made locally by the mobile node with minimal help from the network.
- Nodes will have large enough on-board storage capacity for storing gathered reports.
- Each node knows its position at any point in time, either through hardware (eg GPS receivers) or software (eg by extrapolating its coordinates from received signal strength).
- The system is fully decentralized and uses only short-range peer-to-peer communications for information exchange.

IV. System Overview

In brief, our system works as follows: Nodes generate reports on network conditions at certain points in time and keep these reports in a buffer or data store. When a node encounters another node, they exchange information by sending a portion of reports from their data stores. The node then calculates the relevance of each report and keeps the top N reports in the data store. When a packet has to be sent to the Internet, the node retrieves reports generated within a maximum radius from its current location and determines the integrity of each report by calculating its trustworthiness value. It then combines these reports to make a decision on which interface and network point of attachment to use for data transfer.

A. Node Architecture

A MOBIX node is a software agent executing on a user's device, which is actively gathering reports about network conditions from mobile peers it encounters. An *encounter* is

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6795
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6795

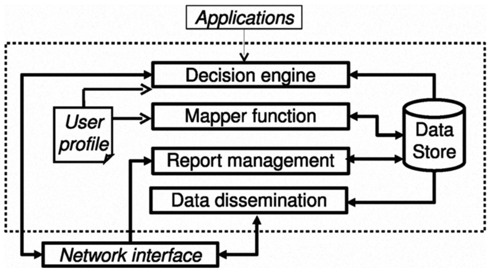


Fig. 1. Different modules in a MOBIX node.

defined as the event of two nodes establishing communication with the goal of exchanging data [18]. An encounter typically occurs when two devices come within communication range and successfully exchange reports with each other. A *report* contains the network conditions on all available interfaces experienced by a mobile node at a particular location and a particular point in time.

Nodes can operate in two modes, either as a *generator* or as a *forwarder* node. A generator node is a node that has other interfaces aside from Bluetooth / Zigbee which are active, or are at least in a mode that allows it to gather relevant network data. A generator node generates reports and stores and forward reports it receives from other nodes. A forwarder node on the other hand only has its Bluetooth interface actively turned on for receiving and forwarding reports.

The main functions of a MOBIX node are shown in Figure 1. We discuss these modules in more detail in the preceding section and briefly describe their functionality here.

Central to the system is the *data store*, where generated and received reports are stored and from which reports are chosen to be sent during an encounter. This data store is accessed by all the other modules but only the report management module can modify its contents.

The *data dissemination module* manages all aspects of data transfer among mobile peers. It decides when to transmit a report, which reports are sent, how many reports to send per transmission, and what the transfer mechanism is. It also implements the communication protocol used to exchange reports between mobile nodes.

The *report management module* is in charge of all aspects of data storage on each mobile node. It determines when a new report is generated by the node (if it is a generator node), which received reports to insert in the data store, and how reports are deleted.

The *mapper module* aggregates received reports in both space and time. It maps the raw data into a quality measure for use by the decision engine, ie RSSI to predicted throughput. The *decision engine*, in turn, takes the user profile and application requirements and tries to match it with available network resources using information gathered in the data store.

Fundamental to the success of our system are the proper management of reports for optimum information dissemination and ensuring the integrity of received reports. We discuss these in the next sections.

TABLE I
CONTENTS OF A REPORT.

Field	Description
repId$_R$	Globally unique ID of report R
ts$_R$	Timestamp of when R was generated
pos$_R$	Node position when R was generated
PoA$_{R,1}$	Access point name / Base station ID of 1st network point of attachment
RSSI$_{R,1}$	Received signal strength for $PoA_{R,1}$ at position pos_R and time ts_R
Op$_{R,1}$	Operator owning or administering $PoA_{R,1}$
...	..
PoA$_{R,n}$	Access point name / Base station ID of nth network point of attachment
RSSI$_{R,n}$	Received signal strength for $PoA_{R,n}$
Op$_{R,n}$	Operator owning / administering $PoA_{R,n}$

B. Report Management

A report is the basic piece of information exchanged between nodes. In this section we discuss the contents of a report, when it is generated and disseminated, and how it is used to make decisions.

1) Report Generation: Table I lists the contents of a report used in our evaluation. Here we assume that there exists some function $Y(repId_R) \rightarrow k$ that returns the type of node k which generated report R with unique ID $repId_R$. Each report includes a timestamp and the node's geographical location when the report was created, and the Received Signal Strength Indicator (RSSI) for all available interfaces, excluding the Bluetooth interface, and all possible Point of Attachment (PoA). In this evaluation, RSSI was used as a baseline network parameter as it is a basic indicator for all wireless interfaces. In the future we will extend the system to incorporate results when other network statistics such as throughput and delay are sent in the report as well. The operator information is included since user's choices will be restricted by the contracts they hold with their data providers.

A report may be generated using several strategies, which can be grouped as:

- Time-based: Reports are produced periodically using a timer on the mobile node. This is easier to implement but finding the optimum generation interval may be difficult. Important events can be lost if the interval is too long while too many redundant reports may be generated on the instances when the device is stationary. This can be solved by varying the report generation interval depending on how fast the device is moving.
- Event-based: Reports are triggered when an event occurs. This is the mode used in some vehicular traffic dissemination algorithm such as [22] where a report is generated every time the vehicle reaches the end of a road segment. In our case, events can be defined as when the node enters or leaves a coverage area, the measured indicator (ie throughput) passes a certain threshold, or every time a user has traveled a certain distance.

We used a time-based approach in producing reports for the rest of our paper, as it is the most straightforward to implement while still allowing us to evaluate the system's performance sufficiently. Event-based report generation is left for future study.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6795
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6795

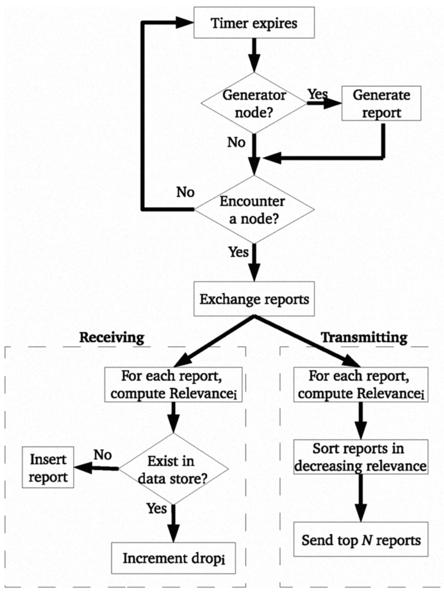


Fig. 2. Main process on a MOBIX node.

2) Report Distribution: The data store is where generated and received reports are stored and from which data is retrieved when a decision needs to be made. The data store has a finite buffer size and can hold at most *B* reports. As with report generation, we transmit reports periodically as long as another node is within transmission range. Although flooding the network leads to the highest bandwidth consumption as opposed to epidemic or proximity-based information dissemination, we argue that the size of the reports are small enough (about 170 bytes/report in our simulation) that bandwidth is not an issue. Additionally communication is essentially free using Bluetooth so there are no financial costs involved for transmitting reports.

Figure 2 shows the main process running on a MOBIX node. As mentioned previously, we are using time-based report generation for our simple evaluation. At periodic intervals, the node creates a report if it is a generator. It then exchanges reports with other nodes within transmission range by broadcasting the top *N* reports in its data store.

Reports are arranged in the data store according to relevance factor, which is an indication of how new and how useful this report will be to the node. Intuitively, we know that younger reports will most likely bring fresher information while reports generated within the node's current vicinity will be more pertinent than those generated farther away. Thus we compute the relevance factor of report *R* to be:

$$Relevance_R = W_{age} * age + W_{dist} * dist \qquad (1)$$

where W_{age} and W_{dist} are the weighting factors of age and distance, respectively, ranging between 0 and 1.0.

When a MOBIX node encounters another node, it exchanges reports in its data store. It first sorts the data store from highest to lowest relevance factor according to Equation 1. It then transmits the top *N* reports. For each report it receives, the node checks that the report does not yet exist in its data store. If the report is a duplicate, it is dropped and the drop count for that report is incremented. Otherwise, the relevance factor is computed and the report is inserted in the data store. Reports which are oldest or farthest away are deleted from the data store when the buffer becomes full.

C. Data Integrity

Data trust in traditional schemes is essentially dependent on trusting the entity where the information came from. Such entity-centric trust is usually based on a priori relationships and hinges on a single source of trust, eg via certification authorities. Even in reputation-based systems, trust is formed over lengthy interactions as nodes build up their reputations over time. Due to the highly mobile and ephemeral nature of node encounters in our system, however, trusting the data by verifying the source will not be appropriate, almost impossible. Instead we propose to use a *data-centric* approach to evaluating the trustworthiness of reports, similar to that adopted in [15].

The trustworthiness of each report is evaluated as a function of its static and dynamic properties and expressed as

$$w_{R,k} = F(k, R) = F(s(k), d(R), t(R))$$

where $w_{R,k}$ is the trust level of report R generated by node of type k. This function returns a value in the interval [0,1] and is a measure of the over-all trustworthiness of the report. The higher the value, the more trustworthy a report is.

The *static trustworthiness*, $s(k)$, depends on the attributes of the node k which generated the report. Nodes can either be *mobile* or *fixed* and owned *publicly* or *privately*. Privately owned nodes, whether mobile or fixed, have relatively low default settings as there is no a-priori means of establishing their integrity. Publicly owned fixed nodes, such as those set up by government agencies, have higher trust settings as they are assumed to have no biases towards specific networks or operators and will generate reports faithfully. Being fixed, these nodes will have some additional protection that makes them resistant to physical tampering.

The *dynamic trustworthiness*, $d(R)$ and $t(R)$, are a function of the distance and age of report R, respectively. As the strength of radio signals vary according to distance and multi-path interference, reports generated farther away would not be as reliable as those produced closer to where the node currently is. This is especially true in boundary conditions experienced near the edges of network coverage. Similarly, reports which are introduced more recently would likely reflect current conditions more accurately than older reports. Cryptographic means can be used so that these geo-timestamps cannot be tampered with by other nodes.

To illustrate, Table II lists examples of trustworthiness values for various report types, where: $d(R) = 1 - 0.25\lfloor \frac{dist_R}{5} \rfloor$ if $0 < dist_R \leq 20$ m and $d(R) = 0$ if $dist_R > 20$ m; $t(R) = 1 - \frac{age_R}{900}$ if $age_R \leq 900sec$ and $t(R) = 0$ if $age_R > 900$ sec. These trust levels become important in decision-making as they will act as weights when combining the reports.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6795
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6795

TABLE II
EXAMPLE TRUSTWORTHINESS VALUES.

Node Type	$s(k)$	dist	$d(R)$	age	$t(R)$	$w_{R,k}$
Private, mobile	0.5	7	0.75	510	0.43	0.16
Private, fixed	0.5	23	0	800	0.11	0
Public, fixed	0.8	12	0.5	300	0.67	0.27
Operator, fixed	0.6	0.01	1	10	0.99	0.59

D. Decision Engine

When a node has a packet to send to the Internet, it needs to make a decision on which interface and network point of attachment to use for this transaction. To do so, the node retrieves all the reports from the data store generated within a maximum radius of its current location. It determines the integrity of each report by calculating the trustworthiness value and uses this to combine the reports into a single RSSI measure for each possible network found. The decision engine then chooses which point of attachment to use by matching application requirements and user preferences with the combined RSSI measures that meet a minimum trust level.

The combined RSSI for point of attachment n at time t, denoted by $RSSI(PoA_{t,n})$, can be evaluated using a variety of techniques. The most straightforward is to calculate the average of the RSSI with the trust level of each report treated as weights:

$$RSSI(PoA_{t,n}) = \frac{\sum w_{R,k} \cdot RSSI_{R,n}}{\sum w_{R,k}}, \forall R : dist < d_{max} \tag{2}$$

The trust level of the combined RSSI can be evaluated as

$$w(PoA_n, t) = \frac{1}{K} \sum_{k=1}^{K} w_{R,k}$$

where K is the total number of reports which contributed to the combined RSSI measurement. More trusted reports thus have a greater influence on the combined RSSI model. However, when there are instances of misleading reports (whether accidental or intentional), false reports may still have significant influence on the outcome.

Two other approaches frequently adopted for data fusion are Bayesian Inference (BI) and Dempster-Shafer Theory (DST)[16]. Using these technique, the decision engine must first translate application requirements into network-centric properties, eg minimum bandwidth to RSSI. Instead of combining the reported RSSI values per se, the decision logic treats each report as *evidence* either supporting or rejecting the hypothesis H that a certain point of attachment meets minimum application requirements. The trust level for each report is calculated similarly as in Section IV-C and are combined as probabilities and beliefs for BI or DST, respectively.

BI is based on well-understood probability theory. The posterior probability of a hypothesis H given e_k independent pieces of evidence is expressed in terms of Bayes' theorem

$$P[H|e] = \frac{P[H]\Pi_{k=1}^{K} P[e_k|H]}{P[H]\Pi_{k=1}^{K} P[e_k|H] + P[\bar{H}]\Pi_{k=1}^{K} P[e_k|\bar{H}]}$$

where $P[e_k|H]$ is the probability that node k confirms hypothesis H given that H is true (node is reliable) while $P[e_k|\bar{H}] = 1 - P[e_k|H]$ is the probability that node k is reporting unfaithfully.

BI uses probabilities to attach weightings to the possible state of the system. It considers evidence as being either true or false, and does not deal well in capturing the *unknown* state. Thus evidence with low trust levels will automatically have high mistrust levels. In cases where the majority of reports have low trust levels, it leads to conclusions opposite from what is expected [15].

DST allows for a certain level of uncertainty in the system by replacing probabilities with *belief* and *plausibility*. Non-supporting evidence does not necessarily refute the hypothesis H; rather, a node has p degree of belief in H and 0 degree of belief in its absence. The trust level of each report k are treated as *mass* m_k either confirming or refuting H. The combined *belief* value for hypothesis H, *bel(H)*, of two masses $m_1(A)$ and $m_2(B)$ is combined using Dempster's rule for combination [5]

$$bel(H) = m_1(A) \oplus m_2(B) = \frac{\sum_{A \cap B = H} m_1(A) m_2(B)}{1 - \sum_{A \cap B = \emptyset} m_1(A) m_2(B)}$$

with the trust levels of nodes 1 and 2 treated as masses m_1 and m_2, respectively.

The decision engine accepts the hypothesis H if *bel(H)* is larger than some preset confidence value. Although DST is a powerful tool for handling uncertainty, it suffers from counter-intuitive behavior when the data is highly conflicting [21]. A number of alternative combination rules have been raised to address these problem, such as those proposed in [20] [7] and [6].

We use the weighted average approach, inspired by [15], to combine reports in our preliminary evaluation of the system. Although it is not proven to be resilient to attacks, the evaluation of the system with regards to security is not the main focus of our paper and is the subject of ongoing work.

V. EVALUATION

We evaluate our system through simulation using two well-known mobility models, random waypoint and Manhattan grid.

A. Evaluation Criteria

The key motivation for mobile nodes to exchange reports is for them to learn about available networks and their conditions and not just rely on their own radio interfaces. If this assumption holds, there will be reports in a node's data store generated within the vicinity of its current location when it needs to establish a network connection. A simple way to evaluate this is to have a mobile forwarder node attempt a file transfer and make a decision on which base station to connect to based on reports alone. If no reports within the vicinity are found, the packet is dropped and no connection attempt is made. The file transfer is deemed successful if the node receives a FIN packet within a specified interval. This FIN packet is sent when the fixed host has received the 1MB file in its entirety. The total number of successful file transfer attempts is thus a good indicator of our system's performance, as a FIN packet will only be received if relevant reports were

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6795
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6795

TABLE III
NUMBER OF NODES

No. of Nodes	Equivalent Population Density
50	200 nodes per km^2
100	400 nodes per km^2
200	800 nodes per km^2
250	1000 nodes per km^2

found in the data store and a successful WiFi connection was established. More formally, we measure the success ratio as

$$Ratio = \frac{\textit{Total number of successful attempts}}{\textit{Total number of file transfer attempts}} \quad (3)$$

Additionally, we compared the difference in decisions based on received reports alone and if the mobile device made its own measurements. For the purpose of analysis, we also looked at the average age of reports used in making decisions.

B. Simulation environment

We used the NS2 simulator with the NS Miracle[10] plug-in to allow for multiple wireless interfaces. The Bluetooth interface was emulated at the radio layer by using a WiFi interface with a 10m transmission radius. Mobile nodes were set to move at an average walking speed around a 500m x 500m simulation area. We divided the simulation area equally into four quadrants and placed a WiFi access point in the center of each quadrant. Each access point had a transmission range of 125m. The combined coverage area of the four access points was about 73% of the total simulation area; thus, there were some locations in the grid with no coverage.

We varied the number of mobile nodes to determine the effect of population density, as summarized in Table III. The simulation time increased dramatically as we increased the number of nodes, so the decision was made to use a smaller simulation area of 500m x 500m instead of the usual $1km^2$ grid. We also varied the percentage of generators to be either 1%, 5%, 10%, 25% and 50% of the total number of mobile nodes. The node attempting the file transfer is always a non-generator node.

We used a simplified system where reports are generated and transmitted periodically. Generator nodes create reports every 5 seconds and all mobile nodes transmit reports at the same rate (but not at the same time, nodes are not synchronized). This period was chosen as it gave the best balance between the percentage of redundant reports received and the possibility of missing an encounter. Reports are sorted according to age (W_{age}=1.0 and W_{dist}=0.0) or distance (W_{age}=0.0 and W_{dist}= 1.0) to determine which factor is a better indicator of relevance. The data store size is fixed at 100 reports and the top 10 reports are sent per transmission.

Nodes start moving at time zero and after 5 minutes, a predefined forwarder node attempts to transfer a 1MB file to a fixed host. The attempt is declared successful if an acknowledgment packet is received within 30 seconds. If no report within the specified radius exists in the database, the node tries again after 1 sec. The node attempts a file transfer every minute and the simulation ends after 25 attempts, or at 30 minutes. Each scenario was tested using 200 simulation runs with a unique movement file per run.

TABLE IV
MOVEMENT FILE PARAMETERS

Parameter	Value
Random Waypoint and Manhattan Grid	
Mean node speed	1.9 m/sec
Speed delta	±0.6 m/sec
Mean pause time	15 sec
Pause delta	± 5 sec
Simulation area	500m X 500m
Simulation time	1800 sec
Manhattan Grid only	
No. of blocks	5 x 5
Turn probability	0.5
Seconds ignored	first 1800 sec

The decision engines combines reports using the weighted averaging approach (Equation 2), with the weights of reports calculated according to its trustworthiness value. As all nodes are mobile, private nodes, their default trustworthiness $s(k)$ are set to 0.5. The dynamic trustworthiness of reports are evaluated using $d(R) = 1 : dist_R \leq 10\text{m}, d(R) = 0 : dist_R > 10\text{m}$ and $t(R) = 1 : age_R \leq 900\text{sec}, t(R) = 0 : age_R < 900\text{sec}$.

C. Mobility Model

Two mobility models were used in the simulations, namely the random waypoint model and the Manhattan grid model. These models were chosen as they represent very diverse mobility patterns. In random waypoint, users move with their speed and direction chosen randomly. Except for the bounding box of the simulation area, there are no restrictions on where users can go. Such an unrestricted random walk can only be possible in large open fields commonly found in rural settings. In reality however, high population densities are characteristic of large metropolitan areas where buildings obstruct movement and people are restricted to walking on streets. Such a scenario is depicted by the Manhattan grid model. Table IV summarize the parameters used when the movement files were generated using the two mobility models.

We used a tool [11] to generate perfect simulation movement files for the random waypoint mobility model and fed this into NS2. This tool effectively shortened our transient period as the nodes start in steady-state distribution. Although the random waypoint model does not represent any realistic user movement, it does give us invaluable insight on how the system performs when users move in an uncoordinated and totally unrestricted manner.

The movement files for the Manhattan Grid model were produced using the BonnMotion movement generator [3]. In this model, nodes move only in predefined paths composed of a number of horizontal and vertical streets which divide the simulation area into blocks. Nodes can turn left, right or go straight with a certain probability at an intersection. In our simulation, we divided the area equally into 5x5 blocks and set the turn probability to 0.5. The BonnMotion code sets the start point of all nodes at (0,0) so we ignored the first 1800 seconds of movement.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6795
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6795

VI. Results

We present our simulation results on the percentage of successful file transfer attempts and evaluate the quality of decisions made using information exchange. We additionally calculate the theoretical energy savings derived from using our system.

A. Success Ratio

Figure 3(a) shows the average ratio of successful file transfer with varying population density for random waypoint and Manhattan grid plotted at 95% confidence interval. The number of generators were set at 50% the total number of nodes and age used as the relevance factor for ranking reports. As mentioned previously, only 73% of the simulation area is under WiFi coverage so we plot the baselines values obtained if the node did not rely on reports alone.

We can see that as the number of mobile nodes increases the success ratio also increases. This is not surprising, since the probability of encountering a node carrying useful information increases as the number of mobile nodes increases. At 250 nodes, the success ratio is approximately 50% for the Manhattan grid model. Considering that the baseline value is around 78%, more than half the time the node needs to make a decision at least one relevant report can be found in its data store.

From the graph we can observe a linear relationship between the number of nodes and the success ratio. Although we do not have a large number of data points, we can use this to approximate the minimum population density required to reach the baseline value. In other words, we approximate the number of mobile nodes needed in order to have a data store hit 100% of the time. Extrapolating the data from the graph, we calculate this value to be ~780 nodes for Manhattan grid and ~520 nodes for Random Waypoint. This translates to a population density of 3,120 nodes/km^2 and 2,080 nodes/km^2, respectively. As a point of comparison, Sydney's Waverly district has a population density of 6,900 people/km^2[17] while Tokyo metropolitan area has 11,526 people /km^2[19]. Thus even if only half of the population is mobile at any given time and assuming all of them carry a mobile device, the calculated population density is not unrealistic of current highly urbanized cities.

We can further see from Figure 3(a) that the mobility model has a significant impact on the results. The system consistently performs better using the Manhattan grid model, even in other experiments where we vary the number of generators and change the relevance factor. Nodes have a higher probability of making an encounter on a Manhattan grid model over the random waypoint because of path constraints. Still, even with the unrestricted node behavior exhibited in random waypoint, we can still expect a data store hit 50% of the time at 250 nodes.

Next we look at the effect of varying the percentage of generators on the system. We keep the number of mobile nodes at 100 and randomly choose 1%, 5%, 10%, 25% and 50% of the nodes as generators, with the rest as forwarders. Figure 3(b) plots the average success ratio with varying generators at 95% confidence interval. We can see that the success ratio increases as the number of generators increases. In particular, there is a significant increase in success ratio going from 1% to 10% generators. However, adding more generator nodes after 25% does not improve performance as dramatically. For instance, the success ratio using Manhattan grid is 36.22(±1.41) at 25% generators. Doubling the number of generators to 50% increases the success ratio by about 4% to 40.61(±1.58). This observation is also true for random waypoint.

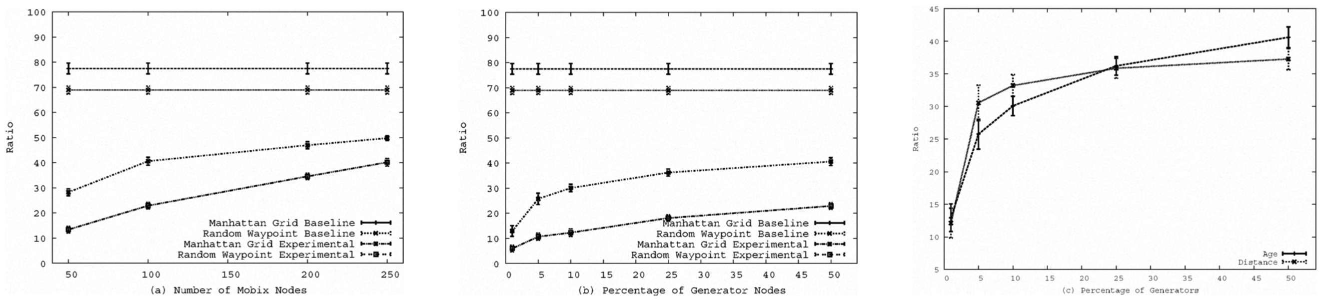
TABLE V
Average distance travelled by 50 reports and average number of nodes holding a copy in the data store n minutes after the reports were generated. (100 nodes, 50% generators)

Time (After n min)	Distance Travelled (m)		Number of Nodes	
	Sort by Age	Sort by Distance	Sort by Age	Sort by Distance
2	137.12	140.82	3.08	2.27
4	202.44	136.07	2.82	1.73
6	191.24	152.23	1.45	2.02
8	122.89	93.43	0.49	1.27
10	28.92	48.6	0.1	0.82

We now compare the performance of the system using different relevance factors in sorting the data store. Figure 3(c) shows the success ratio of 100 nodes with varying number of generators using solely age (W_{age}=1.0, W_{dist}=0.0) or distance (W_{age}=0.0, W_{dist}= 1.0) for the Manhattan grid model. We can see that between 1%-10% generators, sorting the data store according to distance performs slightly better than age. After that however, the success ratio using distance remains at 35% even as we increase the number of generators while age gives a slightly higher success ratio at 50% generators.

To understand these results, we look at the data store contents and track the movement of a sample of 50 unique reports using Manhattan grid model. Table V lists the average distance traveled by a report and the average number of nodes holding a copy of it n minutes after the report was generated. We can see from the table that reports propagate farther from its generation point using age as the relevance factor, reaching on the average more than 200m after 4 minutes. On the other hand, sorting by distance actually limits the report's movements to within its vicinity. Reports stay in the data store for a shorter time using age as relevance factor, as evidenced by the average number of nodes holding a copy of the report dropping to less than one after 6 minutes. This means that some of the 50 reports tracked have been deleted from the system entirely and not a single node has a copy of it in its data store. In contrast, reports stay in the data store longer using distance as the relevance factor. Thus, it appears that age is a better indicator of relevance than distance as it allows reports to propagate further and refreshes the data store much quicker, by removing old reports as younger reports come in. It is also much easier to manage the data store since we do not have to constantly sort the reports as the node moves around. These results are similar to that obtained by [9] where they concluded that age-based ranking performs better than distance-based ranking for disseminating data in mobile peer-to-peer networks.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6795
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6795



(a) Varying number of nodes (b) Varying percentage of generators (100 nodes) (c) Varying relevance factor (Manhattan grid, 100 nodes, 50% generators)

Fig. 3. File transfer success ratio with varying (a) number of nodes, (b) percentage of generators, and (c) relevance factor for sorting data store.

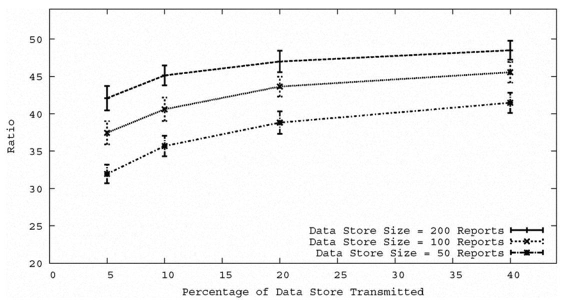


Fig. 4. File transfer success ratio with varying percentage of data store sent per transmission and varying data store size. (Manhattan grid, 100 nodes, 50% generators)

In the previous discussions, we have kept the data store size constant at 100 reports and sent the top 10 reports in the data store per transmission. We now explore the effect of varying the number of reports per transmission and the data store size on the success ratio. Figure 4 shows the results for Manhattan grid with 100 nodes and 50% generators as we vary the data store size from 50 reports to 200 reports and the percentage of the data store sent per transmission. We can see that increasing the data store size significantly increases the success ratio. On the other hand, sending more reports per transmission may not significantly improve performance. Observe that regardless of data store size, the knee of the curve occurs between 10% and 20% of the data store, after which little additional information is gained by sending more reports per transmission. Note also that the success ratio achieved by 100 nodes sending the top 20% of its data store with a size of 200 reports is the same as for 200 nodes with a data store size of 100 reports. Thus for low population densities, it is paramount for nodes to have larger data store capacities and to send about 10% to 20% of the data store contents per transmission to make up for the lesser probability of node encounters.

B. Quality of Decisions

In this section we look at the quality of the decisions made using MOBIX. We compare the decisions on which access point to connect based solely on reports and if the node made its own measurements on Table VI. We can see from the table that the same decisions were reached 97% of the time. Less

TABLE VI
COMPARISON OF DECISIONS BASED SOLELY ON REPORTS AGAINST BASELINE DECISIONS WITH VARYING POPULATION DENSITY.

No. of Nodes	Same AP	False Positive	False Negative	Different AP chosen
50	97.3	0.601	2.08	0.0555
100	97.1	0.848	1.94	0.0727
200	97.4	0.745	1.78	0.0862
250	97.5	0.575	1.65	0.0821

than 3% of the time, the decision engine falsely concluded that an access point is within range when in reality there is none (false positive) or that it is not within radio range of an access point when actually there is (false positive). Note that these are approximately the same results we obtained even if we vary the number of generators, the mobility model, and the relevance factor. These false conclusions occurred at the edges of network coverage where the RSSI value approaches the minimum threshold (-70dBm).

Next, we look at the average age of reports on which decisions were based upon. This gives an indication of how fresh useful reports are in the data store. Figure 5 shows the average age of reports used in making decisions with varying percentage of generator nodes, population density, data store size and reports sent per transmission. We observe from Figure 5(a) and Figure 5(b) that adding more generators, either by increasing the percentage of generators or increasing the total number of nodes, greatly shortens the average age of reports. For instance, the average age at 10% generators (at a total of 100 nodes) is almost double the average age at 50% generators. Additionally, we see that sorting by age means that the data stores are refreshed much quicker compared to sorting by distance. This validates our results obtained in the previous section where we concluded that age is a better relevance indicator than distance.

Figure 5(b) shows that decisions are based on older reports using the Random Waypoint mobility model compared with Manhattan Grid. Encounters are less frequent in random waypoint because of the lack of restrictions on node movements, so nodes need to rely on earlier encounters and search deeper into its data store for relevant reports. Finally, we see that increasing the data store size and the percentage of reports sent per transmission (Figure 5(c)) does not significantly impact the average age of useful reports. Thus for this specific scenario

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6795
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6795

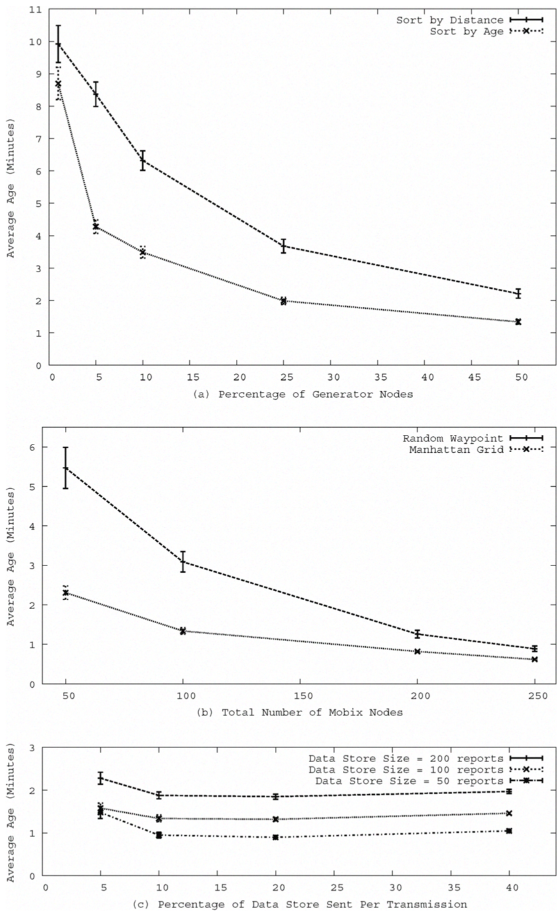


Fig. 5. Average age of useful reports (in minutes) with varying (a) percentage of generators, (b) number of nodes, and (c) percentage of data store sent per transmission.

of 100 nodes and 50 generators, we can conclude that a data store size of 50 reports and 100 reports is too small as reports are deleted too quickly.

C. Energy Savings

One would wonder why nodes would resort to exchanging reports in our simulation scenario when nodes can merely switch on their WiFi interfaces all the time to make its own measurements. The obvious limitation of relying on radio interfaces alone is that measurements will only be limited to what the radio layer can detect (RSSI), whereas by exchanging reports nodes can learn about other network parameters such as throughput and delay as well. Another aspect however is the significant energy savings of using Bluetooth for network resource detection instead of turning on the WiFi interface. For small energy-constrained devices such as PDAs and mobile phones, a WiFi radio represents a significant proportion of the over-all system power, even at low-power idle state [12].

Table VII lists the power consumption for Bluetooth and WiFi interfaces as measured in [12]. We can see that even if the Bluetooth interface is active 100% of the time, it will still be drawing more than 50% less power than the WiFi interface at idle state. Thus, even at very high population densities where nodes have a lot of neighbors within transmit range and the

TABLE VII
MEASURED POWER CONSUMPTION FOR VARIOUS WIRELESS INTERFACES.

Interface	Low-Power Idle	Active Tx
Cisco PCM-350 WiFi	0.390 W	1.60 W
Linksys WCF12 WiFi	0.256 W	0.890 W
BlueCore3 Bluetooth	0.025 W	0.120 W

Bluetooth device is busy all the time, it is still more energy-efficient to use Bluetooth rather than powering on the WiFi interface.

We use the values in Table VII to approximate the total energy expended by the Bluetooth interface over the 30 minute simulation period. There are four access points in the simulation area, so let us assume that each report contains four possible points of attachments and has a total size of S=172 bytes. Additionally, assume the nodes broadcast a packet every 5 seconds without checking if other nodes are in range or not. With a population of 100 nodes, we get that each node received 100 packets on average by the end of the simulation run. BlueCore3 is based on version 1.2 of the Bluetooth specification where the maximum data rate is 1 Mbps. In reality, the actual bandwidth is much lower than specified and the paper [12] measured this value to be 564 kbps and 544 kbps at 2 meters and 7 meters, respectively, for BlueCore3. We get the average and use a bitrate of 554 kbps in our calculations.

The total energy consumed by the Bluetooth interface of one MOBIX node over 1800 simulation sec, E_{total}, can be calculated as

$$E_{total} = E_{tx} + E_{rx} + E_{idle} \text{ W-s} \tag{4}$$

where E_{tx} is the total energy expended to transmit packets, E_{rx} is the total energy expended to receive packets, and E_{idle} is the total energy expended the rest of the time when the Bluetooth device is not active. We can calculate E_{tx} by

$$D_{tx} = \frac{S * N_{rep} * N_{pkts,sent} * 8}{1000} \text{ kb}$$

$$Time_{tx} = \frac{D_{tx}}{BW} \text{ sec}$$

$$E_{tx} = Dev_{tx} * Time_{tx} \text{ W-s} \tag{5}$$

where D_{tx}, total data transmitted over 1800 sec, in kb
S, average report size, in bytes
N_{rep}, number of reports sent per packet
$N_{pkts,sent}$, total packets sent over 1800 sec
$Time_{tx}$, time to send D_{tx}
BW, measured Bluetooth bandwidth
Dev_{tx}, power consumed when transmitting, in W

Here we are assuming that packets are transmitted just once, and that there are no retransmissions. Similarly, we can calculate E_{rx} using Equation 5 by replacing $N_{pkts,sent}$ with $N_{pkts,rcvd}$, the average total number of packets received over the simulation run. We presume that the node consumes the same amount of power when transmitting and receiving a packet. We can approximate E_{idle} by

$$Time_{idle} = 1800 - (Time_{tx} + Time_{rx}) \text{ sec}$$

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6795
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6795

TABLE VIII
TOTAL ENERGY EXPENDED BY THE BLUETOOTH INTERFACE OVER 1800 SEC WITH VARYING NUMBER OF REPORTS SENT PER TRANSMISSION. [R=172 BYTES, $N_{pkts,sent}$=360, $N_{pkts,rcvd}$=100, BW=554 KBPS, Dev_{tx}=Dev_{rx}=0.120 W, Dev_{idle}=0.025 W]

N_{rep}	E_{tx}	E_{rx}	E_{idle}	E_{total}
10	1.07 W-s	0.3 W-s	44.71 W-s	46.09 W-s
20	2.15 W-s	0.6 W-s	44.43 W-s	47.17 W-s
40	4.29 W-s	1.19 W-s	43.86 W-s	49.34 W-s
80	8.58 W-s	2.38 W-s	42.71 W-s	53.68 W-s

BlueCore3 Bluetooth active 100% of the time	216 W-s
Cisco PCM-350 WiFi idle 100% of the time	702 W-s
LinkSys WCFI2 WiFi idle 100% of the time	461 W-s

$$E_{idle} = Time_{idle} * Dev_{idle} \text{ W-s} \tag{6}$$

where Dev_{idle} is the power consumed by the interface when not active.

Table VIII shows the calculated total energy consumed by the Bluetooth interface over 1800 sec at various number of reports sent per transmission. The average size of a report is very small that even if nodes send 80 reports per transmission, the average throughput is just 28.13 kbps. The Bluetooth channel is active about 5% of the time, and the total energy consumed is between 46-54 W-s, around 12% of energy consumed by the LinkSys WCFI2 WiFi card and 8% of the Cisco PCM-350, both in low-power idle state. As mentioned earlier, we can still get power savings of more than 100% in the worst case scenario when the Bluetooth interface is active all the time.

VII. CONCLUSION

Determining network conditions in a heterogeneous, mobile environment is an essential ingredient for optimizing wireless resource utilization. We have presented a system for discovering available networks and their conditions by exchanging reports with other nodes through a short-range communication channel. Using simulation, we show that it is possible to achieve at least 50% successful data store hits even at low population densities and that the required density for 100% data store hit rate is not unrealistic of densely populated areas. Furthermore, we have shown that decisions made based on reports alone are similar to baseline decisions 97% of the time. Finally, we have approximated that theoretically our scheme can achieve more than 50% energy savings by using Bluetooth instead of powering on the WiFi interface.

We are currently extending the security aspects of our proposed scheme. We are looking into incorporating more sophisticated data fusion techniques that will be more resilient to attackers, specifically falsely-reporting nodes. Additionally, we will be performing simulations that will incorporate other network parameters such as bandwidth and latency aside from RSSI in reports.

REFERENCES

[1] Y. Agarwal, T. Pering, R. Want, and R. Gupta. Switchr: Reducing system power consumption in a multi-clients, multi-radio environment. In *ISWC '08: Proceedings of IEEE International Symposium on Wearable Computing*, 2008.

[2] S. Balasubramaniam and J. Indulska. Vertical handover supporting pervasive computing in future wireless networks. *Computer Communications*, 27(8):708 – 719, 2004. Advances in Future Mobile/Wireless Networks and Services.

[3] Bonnmotion: A mobility scenario generation and analysis tool. [Available online] http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/.

[4] N. Dao, R. Malaney, E. Exposito, and X. Wei. Differential voip service in wi-fi networks and priority qos maps. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 5, pages 2653–2657, Dec. 2005.

[5] A. P. Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society*, 30 (Series B):205–247, 1968.

[6] D. Dubois and H. Prade. On the combination of evidence in various mathematical frameworks. In J. Flamm and T. Luisi, editors, *Reliability Data Collection and Analysis*, pages 213–241. Kluwer Academic Pub, Brussels, 1992.

[7] T. Inagaki. Interdependence between safety-control policy and multiple-sensor schemes via dempster-shafer theory. *Reliability, IEEE Transactions on*, 40(2):182–188, Jun 1991.

[8] V. Korhonen and J. Kurhinen. Information exchange in mobile encounter network. In *Networking and Services, 2007. ICNS. Third International Conference on*, pages 70–70, June 2007.

[9] Y. Luo, O. Wolfson, and B. Xu. A spatio-temporal approach to selective data dissemination in mobile peer-to-peer networks. *Wireless and Mobile Communications, 2007. ICWMC '07. Third International Conference on*, March 2007.

[10] Ns-miracle: Multi-interface cross-layer extension library for the network simulator. [Available online] http://www.dei.unipd.it/wdyn/?IDsezione=3966.

[11] S. PalChaudhuri, J.-Y. Le Boudec, and M. Vojnovic. Perfect simulations for random trip mobility models. *Simulation Symposium, 2005. Proceedings. 38th Annual*, pages 72–79, April 2005.

[12] T. Pering, Y. Agarwal, R. Gupta, and R. Want. Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 220–232, New York, NY, USA, 2006. ACM.

[13] A. Rahmati and L. Zhong. Context-for-wireless: context-sensitive energy-efficient wireless data transfer. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 165–178, New York, NY, USA, 2007. ACM.

[14] U. Rathnayake and M. Ott. Predicting network availability using user context. In *MobiQuitous 2008: Proceedings of the fifth Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, New York, NY, USA, July 2008. ACM.

[15] M. Raya, P. P. Papadimitratos, V. Gligor, and J.-P. Hubaux. On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks. In *IEEE Infocom 2008*, 2008.

[16] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[17] *Catalogue 1301.0: Year Book Australia 2008.* Australian Bureau of Statistics, 2008. [Available online] http://www.abs.gov.au/AUSSTATS/abs@.nsf/Productsby Catalogue/B995EE21BC6844BFCA25728E0014D682? OpenDocument.

[18] F. Tan, S. Ardon, and M. Ott. Ubistore: Ubiquitous and opportunistic backup architecture. *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pages 71–78, March 2007.

[19] *Statistical Handbook of Japan 2008.* Statistics Bureau, Ministry of Internal Affairs and Communications, Japan, June 2008.

[20] R. R. Yager. On the dempster-shafer framework and new combination rules. *Inf. Sci.*, 41(2):93–137, 1987.

[21] L. A. Zadeh. A simple view of the dempster-shafer theory of evidence and its implication for the rule of combination. *AI Magazine*, 7(2):85–90, 1986.

[22] T. Zhong, B. Xu, and O. Wolfson. Disseminating real-time traffic information in vehicular ad-hoc networks. *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1056–1061, June 2008.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6795
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6795