Asynchronous Reputation Systems in Device-to-Device Ecosystems

Dimitris Chatzopoulos
System and Media Lab
Hong Kong University of
Science and Technology
dcab@cse.ust.hk

Pan Hui
System and Media Lab
Hong Kong University of
Science and Technology
panhui@cse.ust.hk

ABSTRACT

Advances in device–to–device (D2D) ecosystems have brought on mobile applications that utilise nearby mobile devices in order to improve users' quality of experience (QoE). The interactions between the mobile devices have to be transparent to the end users and can be of many services – opportunistic networking, traffic offloading, computation offloading, cooperative streaming and P2P based k-anonymity location privacy service, to name a few. Whenever mobile users are willing to "ask for help" from their neighbours, they need to make non trivial decisions in order to maximise their utility. Current motivation approaches for mobile users that participate in such environments are of two types: (i) credit-based and (ii) reputation-based. These approaches rely either on centralised authorities or require prohibitively many messages or require tamper resistant security modules. In this paper we propose a trust-based approach that does not require synchronisation between the mobile users. Moreover, we present the three-way tradeoff between, consistency, message exchange and awareness and we conclude that our approach can provide first-rate data to neighbour selection mechanisms for D2D ecosystems with much less overhead.

1. INTRODUCTION

The popularity of smartphones is continuously increasing with almost 1.5 billions of them, which is around 20 percent of the world population, sold only in 2015 [25]. Their capabilities exceed the ones of conventional servers of the previous 5 to 10 years but these devices are only used for personal use by their owners. On the other hand, mobile applications are becoming more resource-hungry and this motivates research on the area of Mobile cloud computing (MCC) [18, 10, 9, 12]. MCC approaches offload the most computationally expensive parts of mobile applications to cloud surrogates in order to provide better quality of experience to the end users. Advances on MCC have been mainly focused on the offloading decisions, the connectivity issues with the cloud surrogate as well as in pricing models. However, under-utilised and capable smartphones with available battery can be found nearby and their owners are willing to share their resources [2]. The concepts of wisdom of crowd and collective intelligence have been utilised by mobile application developers to create a vast spectrum of novel applications that can collectively leverage resources from other mobile devices. Apart from applications with computation offloading functionalities, *context-aware applications* have been developed for D2D ecosystems, as well.

Device-to-device ecosystems are composed by mobile devices that are able to communicate without the support of any fixed infrastructure. In such ecosystems, mobile users form mobile ad-hoc networks (MANETs), communicate wirelessly via WiFi-direct, Bluetooth, or even NFC, move unpredictably and form temporary and delay-tolerant networks (DTNs). D2D ecosystems are challenging due to **(1)** their unpredictability, which is caused by users' mobility, **(2)** the limited, compared to conventional computer, computational resources and battery and **(3)** the incentives required to motivate mobile users to participate.

Resource sharing has to be transparent from the user and needs to respect some sharing constraints (i.e. *If the battery level is more than L% and the CPU utilisation is less than U%, a Dalvik virtual machine instance, with XYZ characteristics can be initialised*). This functionality can be realised via the characteristics of the Hidden Market Design (HMD) [22]. HMD mechanisms work transparently as a set of background processes, without needing any input from the application user. Users who wish to participate only need to specify the amount of their resources they are willing to share via a simple user interface [7].

Given that any mobile user is self-interest and he ideally uses others' resources without sharing any of his, **cooperation enforcing mechanisms** have been proposed. Modern mobile devices are able to, not only forward each others' packets like in the traditional MANET cases, but also exchange resource-demanding services. Later, in Section 3 we discuss such applications. Depending on the design of the cooperation mechanism, extra processing overhead and accounting messages are needed in order to maintain and share information related to mobile users' serviceableness in the whole ecosystem. In this work, we argue that lightweight, in terms of **(i)** message exchanging, **(ii)** processing requirements and **(iii)** storage needs, cooperation enforcing mechanisms can provide enough information to neighbour selection mechanisms (NSMs) on D2D ecosystems.

HotPOST 16, July 04-08, 2016, Paderborn, Germany
© 2016 ACM. ISBN 978-1-4503-4344-2/16/07…$15.00
DOI: http://dx.doi.org/10.1145/2944789.2944873

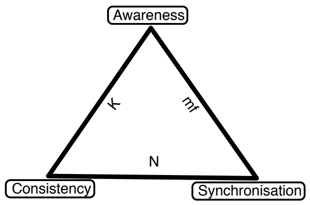


Figure 1: Three way tradeoff of trust estimation in D2D ecosystems.

In order to justify our argument we define the **price of inconsistency** as the overhead caused to the mobile users by not selecting the most suitable helpers due to lack of complete information.

Additionally, we discuss the **cost of synchronisation**, which depends on **(i)** the number of the messages needed to be spread in order to inform every mobile user in the ecosystem and **(ii)** the required storage to save all the received evaluations that lead to complete information.

We consider **recommendation** as a service, which is taking place whenever a mobile user is sharing her experience(s) with other mobile users. Based on the past interactions with the user that is giving the recommendations and her trustworthiness, her recommendations are evaluated. We formulate trust as a random variable, which depicts how much user trusts another user on helping her with her task. The calculation of trust is erroneous because a user can not be familiar with all the interactions of all the other users. The price of inconsistency depends on this error. It is worth to remind that any credit based cooperation enforcing mechanism has to guarantee that the price of inconsistency is zero. On the other hand, in such a mechanism, many more messages should have been exchanged.

Neighbour Selection Mechanisms (NSMs) find the most suitable nearby users based on the score of each candidate. These scores are produced using collected data. This data are created either by the NSMs themselves or by other cooperation evaluation mechanisms. Usually, an application that is suitable for D2D ecosystem has a module to evaluate the contribution of each helper. Depending on the application and the request type, the evaluation can be either based on the benefits caused by the help or by the saved costs. Our argument is based on the fact that each mobile device does not need to share all the data produced by her evaluation with others. On the other hand, depending on the context and the current knowledge of each mobile user, experience sharing between mobile users can be helpful to them to form opinion about others.

There exists a **three-way trade-off**, as shown in Figure 1, between the required size of data that lead to a robust estimation about nearby devices (K), the amount of data that should be broadcasted to everyone whenever two mobile users interact (N) and the freshness of the stored data (mf).

2. RELATED WORK

Credit based schemes stimulate user cooperation in terms of resource sharing by means of virtual currency (credits). The key idea is that users providing a service should be remunerated, while nodes receiving a service should be charged [4, 3, 19]. Reputation based schemes discourage misbehaviour by estimating users reputation and punishing the ones with bad behaviour. The main difference between these two approaches, and also the reason we decided to base our approach in a trust and reputation based system, is the fact that trust is a subjective concept since it is based on data collected by the mobile user that produces trust scores as well as from data she received by other trusted mobile users. This **subjectiveness implies a flexibility in the amount of the required message exchange and also determines the amount of inconsistency**. On the other hand, credit based systems require full consistency, otherwise any malicious user can cheat and make the system collapse.

In our approach we do not force users to exchange messages after any interaction but we allow them to ask for recommendations. The frequency with which a mobile user is updating her knowledge base for other mobile users defines her **awareness**. Awareness differs from consistency on the fact that it does not require full knowledge about everything but only enough information to produce a **robust** trust score.

Application	References
P2P based k-anonymity location privacy	[1] [13] [24]
Cooperative Streaming	[16] [11]
Face Recognition	[17]
Video Compression	[8]
Sensing	[14] [15]
Computation offloading	[20]
Opportunistic Networking	[6]

Table 1: Applications for D2D ecosystems.

3. APPLICATIONS FOR D2D ECOSYSTEMS

Applications on D2D ecosystems can be of many types. Traditional packet forwarding and routing in DTNs will regain popularity on the arrival of 5G technologies because they allow users' traffic to be routed via other proximal mobile devices. Moreover, new smartphones will be equipped with more than one cellular transceivers and will be able to connect with multiple networks at the same time. Also, applications are able to be executed in more than one devices [23], following the paradigm of computation offloading, that was initially proposed for MCC architectures.

When a device receives a task from another node, she needs to allocate additional resources in order to process the task. Context-aware applications require help from other nearby devices in order to estimate the context. For example, mobile crowd sensing applications make use of devices' sensors to perform local measurements and share their data with each other. Context is a multifunctional variable of time and the ambient conditions and is a type of information that is worth sharing between mobile devices regardless of whether they have past interactions or not. The use of these resources will cost the device in terms of battery and, in case of dataplan sharing, in terms of money. These costs can be expressed as a function of the needed resources and the network overhead. The requirements of the offloadable tasks can vary based on the functionalities of the application. We present a few sets of applications in Table 1.

The main difference between applications that have introduced for MANETs and the aforementioned ones is the variety in the possible requested help. Packet forwarding-like applications evaluate the help of each node only by whether she forwards or routes the packets she receives towards the destination. D2D applications have a computation offloading part that should be the main component in the used metrics. In other words, in traditional applications, all the mobile users have the same role and usually the same needs while in modern D2D applications mobile users can diversify in many ways. Devices with different capabilities and variety on mobile applications are two of them.

4. SYSTEM MODEL

In this work we focus on the exchanged data between mobile users in D2D ecosystems and their use and store by Neighbour Selection Mechanisms . These data are exchanged in order to help in the selection of nearby users. We consider a D2D ecosystem with a set of mobile users $\mathcal{U}$. We define as **interaction** between two mobile users the service and the message exchanges between them and we denote with N the number of the messages that were exchanged after the interaction. Any user $u \in \mathcal{U}$, at time t may need help in executing an application $\mathcal{A}_u$. In this work we do not consider how u will select from whom of the other users she will ask for help. Instead, we propose a lightweight way of evaluating and keeping information related to the help of other devices. Let's assume that $\mathcal{A}_u$ is split by u into smaller tasks and u has decided to ask from v to help her with the task $\mathcal{A}_u^v$. We formulate every possible application $\mathcal{A}$ as a combination of services. The set of all possible services is denoted by $\mathcal{S} = \{S_1, S_2, \ldots, S_{|\mathcal{S}|}\}$ and then, each application is a vector in the power set of $\mathcal{S}$, $\mathcal{A} \in 2^{|\mathcal{S}|}$.

Every mobile user, via a simple interface [22], shares some of her resources. The set of shareable resources is denoted by $\mathcal{R} = \{R_1, R_2, \ldots, R_{|\mathcal{R}|}\}$ and there is a direct mapping from an application vector to the set of the **minimum required resources** in order for this application to be executed properly. Without loss of generality, we use normalised, to one, values for the resources and the services.

Example 1: In the video compression application, one device connects with another one, it sends a video, then the helping device compresses the video and later when they meet again, the helping device sends the compressed video to the initial device. This means that the application requires S_1 = computation and S_2 = local network connection. These two services are mapped to R_1 = CPU, R_2 = Memory and R_3 =TCP socket with nearby device.

Example 2: In the Cooperative streaming application, multiple devices are connected with each other and some of them are connected to the Internet and download parts of the same video and share these parts with each other. The cooperative streaming application requires S_2 = local network connection, S_3 = Internet connection and S_4 = network buffer. These three services are mapped to R_1 = CPU, R_2 = Memory, R_3 =TCP socket R_4 = mobile DB.

So $\mathcal{A} = (a_1, a_2, \ldots, a_{|\mathcal{S}|})$ can be described by the set of the services it is based on and we assume that there exist a mapping function $r(\cdot)$ such that:

$$r(\mathcal{A}) \rightarrow \mathcal{R} \tag{1}$$

At the end of each interaction, both devices are able to evaluate the interaction and based on the implemented cooperation enforcing mechanism they will act accordingly. We define a history matrix on each user u for user v, $\mathcal{H}_u^v \in \mathcal{I}^{K,|\mathcal{S}|+1}$, where $\mathcal{I}$ is the unit interval and K is the number of the interactions which v has saved. The number of the columns of $\mathcal{H}_u^v$ are two more that the number of the services because each row contains In the second last column, the id of the mobile device that shared the stored entry with user u about user v. If this device is u herself, the the value is equal to zero. The last column keeps the timestamp of the interaction while the i-th column stores the evaluation of the i-th service that user v promised to provide.

$$\mathcal{H}_u^v = \left(\begin{array}{cccc|c|c} \mathcal{H}_u^v(1,S_1) & \mathcal{H}_u^v(1,S_2) & \ldots & \mathcal{H}_u^v(1,S_{|\mathcal{S}|}) & id_1 & t_1 \\ \mathcal{H}_u^v(2,S_1) & \mathcal{H}_u^v(2,S_2) & \ldots & \mathcal{H}_u^v(2,S_{|\mathcal{S}|}) & id_2 & t_2 \\ \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\ \mathcal{H}_u^v(K,S_1) & \mathcal{H}_u^v(K,S_2) & \ldots & \mathcal{H}_u^v(K,S_{|\mathcal{S}|}) & id_K & t_K \end{array}\right)$$

Recommendation is one of the $|\mathcal{S}|$ services, which is taking place whenever a mobile user is sharing her experience(s) with other mobile users. Based on the past interactions with the user that is giving the recommendations and her trustworthiness, her recommendations are evaluated. The way $\mathcal{H}_u^v$ will be used as well as the value of K depends on the cooperation enforcing mechanism. We consider the case of using a combination of a trust and a reputation system. We refer to trust using the following definition: *"Trust is the ability to accurately predict another person's behaviour."*. We formulate trust as a random variable $\theta_u^v(\mathcal{A}_u^v)$, which depicts how much user u trusts user v on helping her with her application part $\mathcal{A}_u^v$. In order to determine $\theta_u^v(\mathcal{A}_u^v)$ we define a function $f(\cdot)$ such that:

$$\begin{aligned} \theta_u^v(\mathcal{A}_u^v) &= f(\mathcal{H}_u^v \cdot [\mathcal{A}_u^v|0|\cdot] \\ &- \sum_{i=1}^{K} \sum_{j \in \mathcal{U}, j \neq v} (1 - \theta_u^j(\mathcal{A}_u^v)\mathbf{1}_{j=id_i}\mathcal{A}_u^v(i))) \end{aligned} \tag{2}$$

where $\mathcal{A}_u^v(i))$ is the i-th row of $\mathcal{A}_u^v$ and $\mathbf{1}_{j=id_i} = 1$ if $j = id_i$ and 0 otherwise.

Given that $\mathcal{A}_u^v$ can be mapped, via Equation 1, to a set of minimum required resources and that u is not familiar with all interactions of v with the remaining $\mathcal{U} - \{u, v\}$ mobile users, we argue that $\theta_u^v(\mathcal{A}_u^v)$ is erroneous. Moreover, in the case where u had access to all the stored passed data with v's interactions with other users, she could have built a more robust estimation of $\theta_u^v(\mathcal{A}_u^v)$. We define $\tilde{\theta}_u^v(\mathcal{A}_u^v)$ as the trust score u could have built about v if she had access to all v's interactions ($K = \infty$). All these interaction can be known to u if the cooperation enforcing mechanism was credit based, then the enforced integrity guarantees would have allowed u to be familiar with v's interactions. On the other hand, in such a mechanism, many more messages would have been exchanged. Then the price of inconsistency is given by:

$$POI = ||\tilde{\theta}_u^v(\mathcal{A}_u^v) - \theta_u^v(\mathcal{A}_u^v)|| \tag{3}$$

An abstract formulation of the problem we are dealing with in this paper is shown in Equations 4-6.

$$\min_{K \in \mathbb{Z}} \quad ||\tilde{\theta}_u^v(\mathcal{A}_u^v) - \theta_u^v(\mathcal{A}_u^v)|| \tag{4}$$

$$\begin{aligned} s.t \;\; \theta_u^v(\mathcal{A}_u^v) &= f(\mathcal{H}_u^v \cdot [\mathcal{A}_u^v|0|\cdot] \\ &- \sum_{i=1}^{K} \sum_{j \in \mathcal{U}, j \neq v} (1 - \theta_u^j(\mathcal{A}_u^v)\mathbf{1}_{j=id_i}\mathcal{A}_u^v(i))) \end{aligned} \tag{5}$$

$$\begin{aligned} \tilde{\theta}_u^v(\mathcal{A}_u^v) &= f(\mathcal{H}_u^v \cdot [\mathcal{A}_u^v|0\cdot] \\ &- \sum_{i=1}^{\infty} \sum_{j \in \mathcal{U}, j \neq v} (1 - \theta_u^j(\mathcal{A}_u^v)\mathbf{1}_{j=id_i}\mathcal{A}_u^v(i))) \end{aligned} \tag{6}$$

Since K is an fixed integer, whenever a new mobile user is entering the system, other mobile users will try to create data for her and she will also collect data for others while after she has K entries for another mobile user, she will carefully discard past data in order to store new ones.

5. PROPOSED APPROACH

We propose a lightweight approach that works in distributed way in each mobile user independently and without any need of coordination. Our approach is based on the use of the first and second moment of $\theta_u^v(\mathcal{A}_u^v)$. For the use of $f(\cdot)$ we select the framework of Beta distribution. In order to find out $\theta_u^v(\mathcal{A}_u^v)$ we need to first calculate the parameters of Beta distribution which are $\alpha_u^v(\mathcal{A}_u^v)$ and $\beta_u^v(\mathcal{A}_u^v)$. $\alpha_u^v(\mathcal{A}_u^v)$ is the weighted sum of all the positive interactions u has collected about v for all cases where the services of $\mathcal{A}_u^v$ were used while $\beta_u^v(\mathcal{A}_u^v)$ is the weighted sum of the negative ones. The weights in these sums are the trust score of the mobile user that provided the entry. If the entry was provided by u the weight equals to 1. In order to calculate these two parameters we only need $\mathcal{H}_u^v$ and no communication with other devices. The first two moments of Beta distribution are given by:

$$\begin{aligned}\mu_u^v(\mathcal{A}_u^v) &= \frac{\alpha_u^v(\mathcal{A}_u^v)}{\alpha_u^v(\mathcal{A}_u^v)+\beta_u^v(\mathcal{A}_u^v)}\\ \sigma_u^v(\mathcal{A}_u^v) &= \frac{\alpha_u^v(\mathcal{A}_u^v)\beta_u^v(\mathcal{A}_u^v)}{(\alpha_u^v(\mathcal{A}_u^v)+\beta_u^v(\mathcal{A}_u^v))^2(\alpha_u^v(\mathcal{A}_u^v)+\beta_u^v(\mathcal{A}_u^v)+1)}\end{aligned}$$

We consider recommendations as one type of services in the D2D ecosystems. Any new coming mobile user does not have any collected data for the other mobile users. Whenever a mobile user u has in her neighbour list a candidate for help v with empty $\mathcal{H}_u^v$, she assumes that $\alpha_u^v(\mathcal{A}_u^v) = \beta_u^v(\mathcal{A}_u^v) = 1$, which gives v a trust score of 0.5 with a uniform distribution and the highest possible variance σ_u^v. We assume that every mobile user has a confidence score (i.e. maximum acceptable σ_u^v) in her opinion about other mobile users and in order to satisfy this confidence score she requests information about others from other trusted friends.

Given that the information that is produced by our proposal is going to be used by Neighbour Selection Mechanisms that are targeting on improving the QoE of D2D applications, it is important to not marginalise mobile users for their selfish attitude in the past. On the other hand, free riders should not have the same confrontation as the altruists. During the calculation of $\alpha_u^v(\mathcal{A}_u^v)$ and $\beta_u^v(\mathcal{A}_u^v)$, we use a multiplication factor on each entry i of $\mathcal{H}_u^v$:

$$mf = \frac{\lambda}{t - t_i} \tag{7}$$

where λ is a possitive tuning parameter, t is the current timestamp and t_i the timestamp that the entry i was collected. mf slows down the decrease of the variance and feeds the need for new entries. In the general case, any mobile user requests for others' recommendation whenever her current evaluation has bigger variance that the imposed threshold. If her current entries is less than K, she just enters more entries in her history matrix. If her history matrix is full, she discards entries with small contribution to the distribution. The contribution of each entry is calculated by multiplying the entry by mf times the mean of the trust distribution of the mobile user who offered the entry. If the entry was inserted by the user herself, we multiply only by mf.

6. EVALUATION

In this section we show the validity of our initial argument. In Section 6.1 we present a static analysis of users that are uniformly distributed and we show how their population size and the connectivity between them affects the number of the messages needed to maintain the integrity of a credit based system. Also we show how the number of the available data from past interactions affect the consistency of the the users regarding the serviceability of the others. After that, we use three mobility traces in Section 6.2 to show that the trust score follows the same trend as the credit redistribution in a community of mobile users with different helping profiles.

6.1 Evaluation using Random Geometric graphs

We produce instances of static random geometric graphs using MATLAB. We distribute uniformly users in a $[0,1] \times [0,1]$ area. In Figure 2a we show how many retransmissions are required in order for one message to arrive to all the users of the network in the case of 1000 users or 2000 users. The x-axis of both figures 2a and 2b show the connectivity threshold. By connectivity threshold we define the ratio between the coverage radius of a smartphone to the whole examined area[1]. Moreover, Figure 2b shows how the graph diameter is decreasing and the fraction of the users in the major connected component is converging to 100% when the connectivity threshold is increasing.

Figure 2c shows how the amount of the stored interactions, K, affects the diversity of the trust scores between mobile users. We have randomly selected a probability for each user to be helpful to others in the begging of the simulations, which last for 10000 slots, and on every quarter of the simulation time we change the profile of the users to either not helpful or to completely altruistic that help everyone and we used $\lambda = 1$ in the equation 7.. As we can see from the plot, if the collected data are not enough, the trust estimation can not follow the profile change.

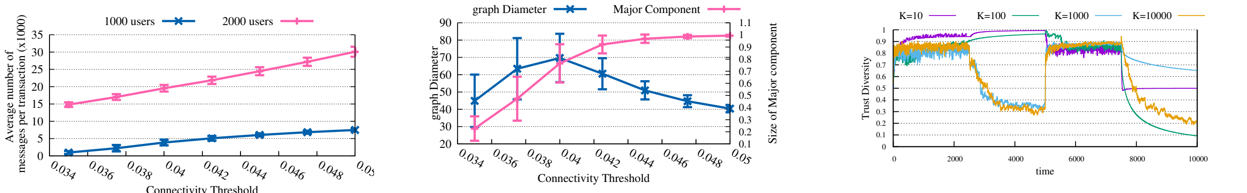
6.2 Evaluation using mobility traces

We implement an event-driven simulator to depict the performance of our proposal. We used three datasets, Infocom 05 and Infocom 06 from the Haggle project [21] and Humanet [5], which contain user mobility traces in different environments. The duration of the simulation is one day. We select the first day of the first two, while Humanet is already only one day long. We considered all the mobile users, which are 41 for Infocom 05, 78 for Infocom 06, and 56 for Humanet.

We performed an experiment to show the effects of the incentive scheme and the reputation mechanism on rewarding the collaborating nodes and sidelining the selfish ones. Given the limitations of the datasets, we diminished the initial budget of each mobile user to 2000 credits, so that the selfish nodes can finish their budget during the one-day simulation period. All the mobile users are initialized with the same trust score, which is 0.5. As explained in Section 5, if we do not know anything about others, we set $\alpha = \beta = 1$, which means that the trust score is 0.5.

Figure 3 shows how the credits are distributed among the users and how the reputation of the users changes after the simulation. These results are presented separately

[1] For the simulation purposes we assumed that all the smartphones have the same coverage radius and that are uniformly distributed in a squared area.

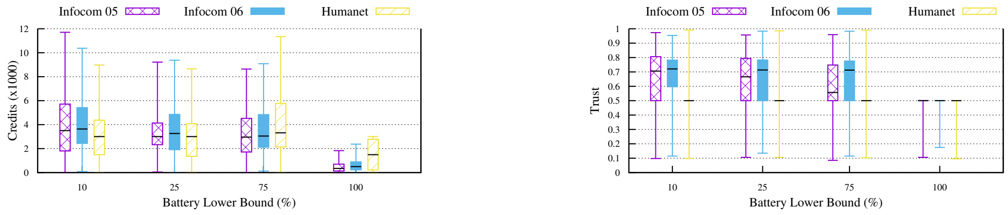


(a) Transmitted messages per transaction for different values of the coverage area.

(b) Graph diameter and size of the major connected component for different values of the coverage area.

(c) The higher the size of the collected data (K), the less the diversity in the trust estimation between users.

Figure 2: Results from Simulations. In the two leftmost Figures we present the required messages (N) and the formation of the graph that produced by Random Geometric graphs. In the third Figure we show how the available storage (K) for the collected data from recommendations affects the adaptability of our algorithm when users change helping profile.



(a) Credits.

(b) Trust.

Figure 3: Redistribution of credits and trust scores among the different user categories, which are defined based on the battery lower bound of their devices. We define three categories, with battery lower bound 25%, 75%, and 95%, which represent altruistic, conservative, and selfish users, respectively. The reputation score of the selfish users decreases with time, making them undesired by other nodes for offloading. This way, they quickly consume their credits, and without being able to gain more their possibilities for collaboration with other mobile users are limited.

for users belonging to one of the three categories previously defined: altruistic, conservative, and selfish. In the y1-axis we show the box-plot representing the minimum, 25^{th}, 50^{th}, 75^{th} percentiles, and the maximum of the cumulative number of credirs at the end of the simulation for each group. In the y2-axis we show the box-plot of the cumulative reputation of the users in each group. The trend from left to right is decreasing for both metrics, until it drops to a minimum for the free-riders (nodes with battery threshold 95%, as explained in the Introduction, via a simple interface that follows the principles of hidden market design).

It is important to notice that these values are strictly related to each other. When a node selects another node from the list of the known nodes, it does so according to their reputation score: The higher the reputation, the higher the chances for a node to be selected. Based on this intuition, in general, a node will only increase its reputation or maintain the same one (if a node has worst reputation than the others, it will not be selected very often, so the chances that it fails to successfully execute a tasks are lower). This intuition is confirmed by the results of Figure 3, where we can see that the almost all selfish nodes finish their credits, while the altruistic ones increase their budget. In the long term, if the selfish nodes want to be part of the system, they should start gaining some credits and improve their reputation. To do so, they should be more generous and change the bounds accordingly, e.g. decrease the battery lower bound.

However, as we can see from the trend in Figure 3, the reputation score is decreasing with the same speed as the credits and this fact implies that both credit-based and reputation and trues-based schemes can result, in the long term, in the same outcome.

7. CONCLUSION AND FUTURE WORK

In this paper we analysed a mechanism that provides trust scores to cooperation enforcing mechanisms for mobile devices in D2D ecosystems. We argued that mobile users do have a by-default motivation to help their neighbours and that's why they need schemes that stimulate user cooperation. Moreover, we focus on reputation based mechanisms because the credit-based ones, have high communication cost in order to guarantee integrity, if possible. Evaluation using static and dynamic graphs show that our selection is the proper one for D2D ecosystems.

Our future work will focus on understanding how the inherent parameters of D2D ecosystems affect the trust estimation and on implementing our proposal as a component of a cooperation enforcing mechanism on Android.

8. ACKNOWLEDGEMENTS

This research has been supported, in part, by General Research Fund 26211515 from the Research Grants Council of Hong Kong and the Innovation and Technology Fund ITS/369/14FP from the Hong Kong Innovation and Technology Commission.

9. REFERENCES

[1] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, Jan 2003.

[2] C. Bermejo, R. Zheng, and P. Hui. An empirical study of human altruistic behaviors in opportunistic networks. In *Proceedings of the 7th International Workshop on Hot Topics in Planet-scale mObile*

computing and online Social neTworking, pages 43–48. ACM, 2015.

[3] S. Buchegger and J.-Y. Le Boudec. Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 226–236. ACM, 2002.

[4] L. Buttyán and J.-P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mob. Netw. Appl.*, 8(5):579–592, Oct. 2003.

[5] J. M. Cabero, V. Molina, I. Urteaga, F. Liberal, and J. L. Martin. CRAWDAD data set tecnalia/humanet (v. 2012-06-12). http://crawdad.org/tecnalia/humanet/, June 2012.

[6] Y. Cao and Z. Sun. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys Tutorials, IEEE*, 15(2):654–677, Second 2013.

[7] D. Chatzopoulos, P. Hui, and D. Huang. Fides: A hidden market approach for trusted mobile ambient computing. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pages 81–82, April 2015.

[8] D. Chatzopoulos, K. Sucipto, S. Kosta, and P. Hui. Video compression in the neighborhood: An opportunistic approach. In *IEEE ICC 2016 Ad-hoc and Sensor Networking Symposium (ICC'16 AHSN)*, Kuala Lumpur, Malaysia, May 2016.

[9] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. Clonecloud: Elastic execution between mobile device and cloud. In *Proceedings of EuroSys*, 2011.

[10] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. Maui: Making smartphones last longer with code offload. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 49–62, New York, NY, USA, 2010. ACM.

[11] K. Eustice, V. Ramakrishna, N. Nguyen, and P. Reiher. The smart party: A personalized location-aware multimedia experience. In *2008 5th IEEE Consumer Communications and Networking Conference*, pages 873–877, Jan 2008.

[12] M. S. Gordon, D. A. Jamshidi, S. Mahlke, Z. M. Mao, and X. Chen. Comet: Code offload by migrating execution transparently. In *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, OSDI'12, pages 93–106, Berkeley, CA, USA, 2012. USENIX Association.

[13] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, MobiSys '03, pages 31–42, New York, NY, USA, 2003. ACM.

[14] Y. Huang, A. Tomasic, Y. An, C. Garrod, and A. Steinfeld. Energy efficient and accuracy aware (e2a2) location services via crowdsourcing. In *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 436–443, Oct 2013.

[15] M. Keally, G. Zhou, G. Xing, and J. Wu. Remora: Sensing resource sharing among smartphone-based body sensor networks. In *Quality of Service (IWQoS), 2013 IEEE/ACM 21st International Symposium on*, pages 1–10, June 2013.

[16] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou. Microcast: Cooperative video streaming on smartphones. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 57–70, New York, NY, USA, 2012. ACM.

[17] R. Kemp, N. Palmer, T. Kielmann, and H. Bal. *Mobile Computing, Applications, and Services: Second International ICST Conference, MobiCASE 2010, Santa Clara, CA, USA, October 25-28, 2010, Revised Selected Papers*, chapter Cuckoo: A Computation Offloading Framework for Smartphones, pages 59–79. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[18] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *Proceedings of IEEE INFOCOM*, 2012.

[19] P. Michiardi and R. Molva. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Advanced communications and multimedia security*, pages 107–121. Springer, 2002.

[20] A. Mtibaa, K. Harras, K. Habak, M. Ammar, and E. Zegura. Towards mobile opportunistic computing. In *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pages 1111–1114, June 2015.

[21] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD data set cambridge/haggle (v. 2006-01-31). Downloaded from http://crawdad.org/cambridge/haggle/, Jan. 2006.

[22] S. Seuken, D. C. Parkes, E. Horvitz, K. Jain, M. Czerwinski, and D. Tan. Market user interface design. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 898–915. ACM, 2012.

[23] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura. Serendipity: Enabling remote computing among intermittently connected mobile devices. In *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '12, pages 145–154, New York, NY, USA, 2012. ACM.

[24] R. Shokri, G. Theodorakopoulos, P. Papadimitratos, E. Kazemi, and J. P. Hubaux. Hiding in the mobile crowd: Locationprivacy through collaboration. *IEEE Transactions on Dependable and Secure Computing*, 11(3):266–279, May 2014.

[25] The Statistics Portal. Number of smartphones sold to end users worldwide from 2007 to 2015 (in million units). http://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/, 2016. Online; accessed 14 March 2016.