Strategic Analysis of Trust Models for User-Centric Networks

Marta Kwiatkowska
Department of Computer Science
University of Oxford

David Parker
School of Computer Science
University of Birmingham

Aistis Simaitis
Department of Computer Science
University of Oxford

We present a strategic analysis of a trust model that has recently been proposed for promoting cooperative behaviour in user-centric networks. The mechanism for cooperation is based on a combination of reputation and virtual currency schemes in which service providers reward paying customers and punish non-paying ones by adjusting their reputation, and hence the price they pay for services. We model and analyse this system using PRISM-games, a tool that performs automated verification and strategy synthesis for stochastic multi-player games using the probabilistic alternating-time temporal logic with rewards (rPATL). We construct optimal strategies for both service users and providers, which expose potential risks of the cooperation mechanism and which we use to devise improvements that counteract these risks.

1 Introduction

User-centric networks are designed to encourage users to act cooperatively, sharing resources or services between themselves, for example in order to provide connectivity in a mobile ad-hoc network. The effectiveness of such networks is heavily dependent on their cooperation mechanisms, which are often based on the use of incentives to behave unselfishly. In this paper, we present an analysis of a cooperation mechanism for user-centric networks [3], which combines a *reputation*-based incentive, used to establish a measure of *trust* between users, and a *virtual currency* mechanism used to "buy" and "sell" services.

The cooperation model proposed in [3] was analysed formally by the authors using probabilistic model checking [1, 2]. They verified several performance properties, specified in the probabilistic temporal logics PCTL and CSL, on discrete- and continuous-time Markov chains models and, in [1], also used Markov decision processes to assess the worst-case performance of service providers.

In this paper, we take a different approach and study the cooperation mechanism using *strategy-based analysis*. The system is modelled as a *stochastic multi-player game*, in which service providers and customers are modelled as players with objectives, expressed in the logic probabilistic alternating-time temporal logic with rewards (rPATL) [4]. We model and analyse the cooperation mechanism using PRISM-games [5], a probabilistic model checker for stochastic multi-player games. We use rPATL model checking to identify weaknesses in the cooperation mechanism and then perform *strategy synthesis* to discover important insights into the model: firstly, we construct and visualise potential attacks or undesirable behaviour; secondly, we develop improvements to the system that alleviate these problems and check their correctness.

Related work. Game-theoretic techniques have been applied to a wide variety of problems in the context of computer networks, from network security [8] to self-organisation in ad-hoc networks [6]. Of particular relevance to this paper is the work in [7], which gives a game-theoretic analysis of cooperative incentive schemes in mobile ad-hoc networks and proposes the combination of trust and currency mechanisms used in [3]. Its effectiveness is analysed using a combination of theoretical and simulation results. By contrast, we adopt a semi-automatic approach where the strategies are synthesised automatically by the tool from rPATL specifications, and are then analysed to understand and improve the cooperation

To appear in EPTCS.

mechanism. The logic rPATL has been previously used to analyse cooperation incentives in micro-grid energy management and decentralised agreement in sensor networks [4], but a detailed strategy-based analysis was not performed.

2 Modelling the Cooperation Mechanism

2.1 The cooperation mechanism

The basic ideas behind the cooperation mechanism of [3] can be summarised as follows. We assume a general model of *providers* offering *services* to *requesters*. Cooperation between *users* of the network (requesters and providers) is managed through a combination of reputation and virtual currency.

Reputation is captured by a discrete *trust measure*, denoted $trust_{ij}$, representing the extent to which user i trusts user j, based on previous interactions between them and the recommendations provided by the other users in the network. This is used to determine whether a service request from j is accepted by i. A *trust level* T_{ij} is computed as a weighted sum $T_{ij} = \alpha \cdot trust_{ij} + (1-\alpha) \cdot recs_{ij}$, where $recs_{ij}$ is an "indirect" trust measure, taken as the the average value of $trust_{kj}$ for other users k (whereas $trust_{ij}$ is called a "direct" measure of trust). By default, i will decide to accept j's request if T_{ij} is not below a pre-specified *service trust level*, denoted st_i. The parameter $\alpha \in [0,1]$ controls the relative influence that the direct and indirect measures of trust have on this decision.

The reputation scheme is then integrated with a virtual currency system, where services are bought and sold between users, and the cost paid to i by j for a service is a function of $trust_{ij}$. Assuming model parameters for minimum and maximum costs C_{min}, C_{max} and threshold T', the cost is defined as

$$C(trust_{ij}) = \begin{cases} C_{min} + \frac{C_{max}-C_{min}}{T'} \cdot (T' - trust_{ij}) & \text{if } trust_{ij} < T' \\ C_{min} & \text{if } trust_{ij} \geq T' \end{cases}$$

Procurement of a service proceeds in several phases. First, a requester j chooses a provider i and makes a request. If $T_{ij} \geq st_i$, the request is accepted. In this case, the two users then "negotiate" the service cost, using the function of $trust_{ij}$ given above. The negotiation may, however, fail: with probability c_i, user i cancels the accepted request; this represents the "cooperative attitude" [1] of the provider i. If not cancelled, the service is delivered and the requester chooses whether or not to pay the negotiated price to the provider. If payment is made, the provider increases the trust measure of the requester by one unit. If not, the measure is decreased by td_i units. On encountering a requester for the first time, a provider shares the trust measure with the other providers.

2.2 A stochastic game model

We build a model of the cooperation mechanism of [3] as a (turn-based) stochastic multi-player game (SMG). An SMG comprises a finite set of *players* and a finite set of *states*. In each state, exactly one player chooses (possibly randomly) from a set of available *actions*. When an action is taken, the result is a *probabilistic transition*, i.e. a successor state is chosen according to a discrete probability distribution. The choices for each player are made by a *strategy*, which selects an action (or distribution over actions) based on the history of the SMG so far. The strategies needed in this paper are *memoryless* (i.e. history independent) and *deterministic* (i.e. do not use randomisation).

We developed the SMG model using the PRISM-games model checker, taking the PRISM model of [1] as a starting point.[1]. The SMG model has one player for each user in the network. The choices

[1] All model/property files are available at: `http://www.prismmodelchecker.org/files/sr13trust/`

made by a "requester" player model the decision of which provider is selected at each point in the system execution. In the basic model, the "provider" players do not have any choices to make; later (in Sections 3.2 and 3.3), we will add choices for these players in order to synthesise strategies that can be used to improve the cooperation mechanism. The stochastic aspects of the SMG model are primarily required to model the fact that negotiations fail probabilistically.

We adopt the same basic network configuration as used in the original analysis of the protocol [1], which comprises 3 providers and 1 requester. Even though this network is relatively small, it still captures the fundamental aspects of the protocol. For instance, observe that the decision whether to provide a service to a requester does not depend on the trust level of other requesters in the network, so incorporating more requesters does not offer any more information about the dynamics of trust and provided services. On the other hand, as we will show, using three service providers already allows us to identify malicious strategies for the requester that can be generalised to an arbitrary number of providers (see, e.g., the discussion about the number of unpaid requests in Section 3.1).

The parameters of the cooperation mechanism are also taken from [1] and are as follows. The trust measure is an integer in the range 0 to 10 and is initially 5. We use α=0.8 to compute trust levels, unless stated otherwise, and the service trust threshold st_i is set to 5 for all providers. We use a negotiation failure probability of c_i=0.05 for all providers i, and the parameters used to compute prices are fixed at C_{min}=2, C_{max}=10 and T'=8.

3 A Strategy-based Analysis

We now analyse the cooperation model described above, showing how the interplay between the two key components of the protocol, trust and virtual currency, affects the cooperation dynamics. Our analysis is based on strategy synthesis for properties in the temporal logic rPATL [4]. The logic combines features of the multi-agent logic ATL, the probabilistic logic PCTL, and operators to reason about expected reward or cost measures. A simple example of an rPATL formula is $\langle\langle\{1,2\}\rangle\rangle \mathsf{P}_{\geq 0.75}[\mathsf{F}^{\leq 5} goal]$, which asks "do players 1 and 2 have a (combined) strategy to ensure that the probability of reaching a '*goal*' state within 5 steps is at least 0.75, regardless of the strategies of other players in the game?". Alternatively, we can use a numerical query such as $\langle\langle\{1,2\}\rangle\rangle \mathsf{P}_{\max=?}[\mathsf{F}^{\leq 5} goal]$: "what is the maximum probability of reaching a '*goal*' state within 5 steps that can be ensured by players 1 and 2?". An example of property to reason about rewards (or costs) is $\langle\langle\{3\}\rangle\rangle \mathsf{R}^{r}_{\leq 10}[\mathsf{F}^{\star} goal]$, which asks "does player 3 have a strategy to ensure that the expected amount of reward r cumulated before reaching a '*goal*' state is at most 10?". The $\star$ parameter lets us specify what the total reward should be if a '*goal*' state is *not* reached: we can assign zero reward ($\star$=0), infinite reward ($\star$=∞) or allow reward to accumulate indefinitely ($\star$=c). For precise details of the logic rPATL and its semantics, we refer the reader to [4].

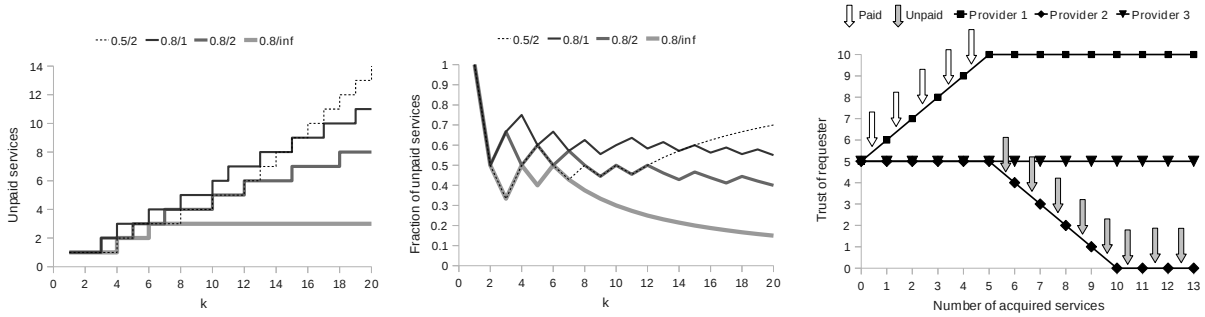
3.1 Unpaid requests

First, we consider the extent to which the requester can obtain services without paying for them. We analyse the maximum (expected) number of unpaid services that the requester can obtain if its goal is to get k services in total. This is expressed in rPATL as:

$$\langle\langle\{requester\}\rangle\rangle \mathsf{R}^{unpaid}_{\max=?}[\mathsf{F}^{c} services = k],$$

where *unpaid* denotes a *reward structure* assigning 1 to every unpaid request. The results for various combinations of model parameters α and td_i are shown in Figure 1 (we use 0.5/2 to indicate that $\alpha = 0.5$

and trust is decreased by $td_i = 2$ units upon an unpaid service; $td_i = \text{inf}$ means that trust is reset to 0 upon an unpaid service).



(a) *Number* of unpaid services. (b) *Fraction* of unpaid services. (c) Strategy for 0.5/2 and $k = 13$.

Figure 1: Maximum unpaid services the requester can achieve in obtaining k services.

Figures 1a and 1b show the number and fraction, respectively, of services that are unpaid, for a range of k. From Figure 1b, in particular, we see that, for parameters 0.5/2 and 0.8/*inf*, the behaviour is fundamentally different from the other two - the portion of requests converges to 1 and 0, respectively. For 0.8/*inf*, this behaviour is expected, because the trust measure is decreased to 0 upon non-payment; however, the behaviour of 0.5/2 represents an attack on the trust model allowing the requester to receive an unlimited number of unpaid services for a fixed cost. We synthesise an attacker (requester) strategy for our model with 3 providers, for the case of acquiring $k = 13$ services: for a cost of 5 services, the requester can get an unlimited number of unpaid services. We depict the strategy in Figure 1c. Arrows represent "request-and-pay" (white arrow) and "request-and-do-not-pay" (grey arrow) actions of the optimal requester strategy, depending on the number of services acquired so far.

This attack is possible if $st_i \leq (1-\alpha) \cdot T_{\max}$ for some provider i, where $T_{\max}$ is the maximum trust level among all providers. We note that it is only viable if the network is sufficiently small since the fixed cost increases with the number of providers sharing the trust information: to achieve the required indirect trust measure $recs_{ij} \geq \frac{st_i}{1-\alpha}$, the requester must pay for a number of services proportional to the number of providers. However, in order to work, this requires that all providers share their initial direct trust measure even though they have not encountered the requester.

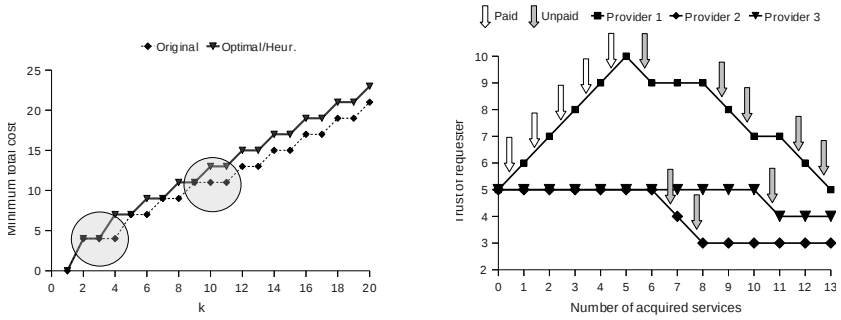
3.2 Cost of obtaining services

We now turn our attention to the virtual currency system, and study the minimum price at which the requester can buy k services. For this, we use rPATL formula:

$$\langle\langle \{requester\} \rangle\rangle \mathsf{R}^{cost}_{\min=?}[\mathsf{F}^{\infty} services = k].$$

Intuitively, the requester has a strategy to get one unpaid service for each paid service by executing the following sequence: pay, not pay, pay, not pay, etc. However, a plot of the above property (see highlighted sections of line 'Original' in Figure 2a), shows deviations from this pattern, where the requester can get 4 services for the price of 2 and, similarly, 11 services for the price of 9.

We synthesise a strategy achieving this and depict it in Figure 2b. We can see that all paid requests are directed to one provider and the others only receive unpaid requests. In fact, by exploiting the reputation system, the requester is even able to obtain 2 unpaid requests from provider 2.



(a) Minimum cost to obtain k services. (b) Example strategy for $k = 13$.

Figure 2: Cost of k services for requester and a strategy example.

Next, we devise a fix by changing the model to allow providers to manage the way they share trust information between themselves: they can choose whether to share trust information after interaction with the requester. We synthesise the optimal trust information sharing strategy for cooperating providers, whose behaviour is shown as 'Optimal/Heur.' in Figure 2a and can be seen to avoid the above shortfall. Manual examination of the synthesised strategy reveals a suitable heuristic whereby providers share trust information only when its direct trust of the requester is smaller than that of the others. We implement this heuristic in the model and find that it yields the same model checking results as the optimal strategy.

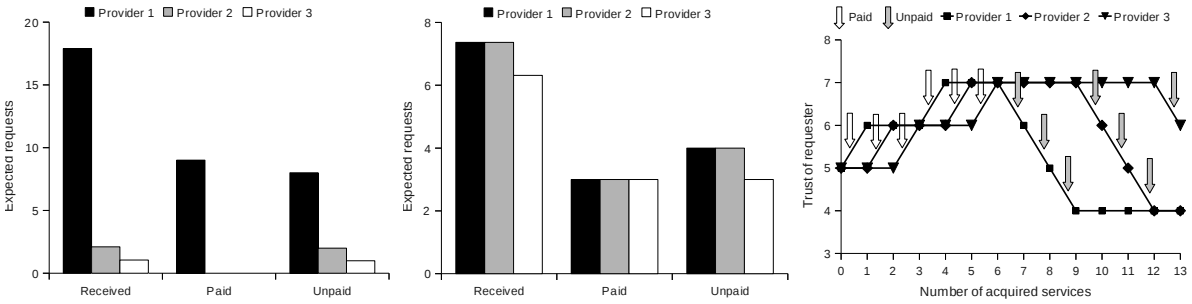
3.3 Provider selection incentives

Another interesting feature revealed by the analysis of the strategy in the previous section is that the proposed virtual currency system provides an incentive for the requester to only ever pay for services from one provider (see Figure 3a). This is in fact optimal behaviour because, in the computation of the service cost, only the direct trust measure is used. This may or may not be a desired feature for the mechanism. We can show that a simple change that incorporates the maximum difference between trust into the pricing model (i.e., cost is now computed as $original\_cost + \max_k |trust_{ij} - trust_{kj}|$, where $original\_cost$ is the cost assigned by the pricing scheme from Section 2.1) incentivises the requester to disperse its requests between service providers.

Figure 3b shows the distribution of requests between providers and Figure 3c depicts the actions of the optimal strategy in the new pricing scheme. This strategy contrasts with the strategy for the original mechanism from Figure 2b because paid requests are now distributed uniformly across all the service providers. This analysis of strategies has been performed using the "strategy implementation" feature of PRISM-games, which allows the user to synthesise an optimal player strategy for some rPATL formula, and then evaluate a second rPATL property on the modified SMG in which one coalition's strategy is fixed using the previously synthesised one. In this instance, we used the following rPATL formulae:

$$\langle\langle \{requester\} \rangle\rangle \mathsf{R}^{cost}_{\min=?}[\mathsf{F}^{\infty} services = k] \quad \text{and} \quad \langle\langle \emptyset \rangle\rangle \mathsf{R}^{r}_{\min=?}[\mathsf{F}^{c} services = k],$$

where the first formula was used to synthesise the strategy and the second formula is the one used to analyse it (r represents reward structures for Received, Paid, and Unpaid).



(a) Original pricing scheme. (b) Modified pricing scheme. (c) Optimal strategy for 13 services.

Figure 3: Distribution of requests among providers.

4 Discussion and future work

We have presented a strategy-based analysis of a cooperation mechanism for user-centric networks, using automated verification of stochastic multiplayer games. We have identified several undesirable properties of the model, including attacks on the reputation system and inefficiencies of the virtual currency mechanism. These would have been difficult to discover using conventional model checking. Furthermore, we have shown that an analysis of optimal strategies for the model can help us understand the incentives that the model introduces to the system and to devise and verify improvements.

Our approach, which is based on probabilistic model checking, builds and analyses a more detailed system model than other game-theoretic analysis techniques, such as [7]. On the one hand, this may impose limitations on the scalability of our approach. On the other hand, we are able to look at the protocol in fine detail and, as we have shown in this paper, identify subtle problems that arise even with a small number of system components, but which may also generalise to larger models.

There are many interesting directions for future work. We plan to further develop our probabilistic model checker PRISM-games to provide a wider range of analysis techniques. For example, we plan to incorporate additional reward operators dealing with limit averages and discounted sums. We would also like to investigate extensions of our techniques to incorporate partial-information strategies or more complex solution concepts such as Nash and subgame-perfect equilibria.

Acknowledgments. The authors are part supported by ERC Advanced Grant VERIWARE, the Institute for the Future of Computing at the Oxford Martin School and EPSRC grant EP/F001096/1.

References

[1] A. Aldini & A. Bogliolo (2012): *Model Checking of Trust-Based User-Centric Cooperative Networks*. In: *Proc. 4th International Conference on Advances in Future Internet (AFIN'12)*, pp. 32–41.

[2] A. Aldini & A. Bogliolo (2012): *Trading Performance and Cooperation Incentives in User-Centric Networks*. In: *Proc. International Workshop on Quantitative Aspects in Security Assurance (QASA'12)*.

[3] A. Bogliolo, P. Polidori, A. Aldini, W. Moreira, P. Mendes, M. Yildiz, C. Ballester & J. Seigneur (2012): *Virtual Currency and Reputation-based Cooperation Incentives in User-Centric Networks*. In: *Proc. 8th International Wireless Communications and Mobile Computing Conference (IWCMC'12)*, pp. 895 –900, doi:10.1109/IWCMC.2012.6314323.

[4] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker & A. Simaitis (2013): *Automatic Verification of Competitive Stochastic Systems*. *Formal Methods in System Design*. To appear.

[5] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker & A. Simaitis (2013): *PRISM-games: A Model Checker for Stochastic Multi-Player Games*. In: *Proc. 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'13)*, *LNCS* 7795, Springer, pp. 187–193.

[6] M. Felegyhazi, J.-P. Hubaux & L. Buttyan (2006): *Nash equilibria of packet forwarding strategies in wireless ad hoc networks*. *Mobile Computing, IEEE Transactions on* 5(5), pp. 463 – 476, doi:10.1109/TMC.2006.68.

[7] Z. Li & H. Shen (2012): *Game-Theoretic Analysis of Cooperation Incentive Strategies in Mobile Ad Hoc Networks*. *IEEE Transactions on Mobile Computing* 11(8), pp. 1287 –1303, doi:10.1109/TMC.2011.151.

[8] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya & Q. Wu (2010): *A Survey of Game Theory as Applied to Network Security*. In: *Proc. 43rd Hawaii International Conference on System Sciences (HICSS'10)*, pp. 1 –10, doi:10.1109/HICSS.2010.35.