IRONMAN: Using social networks to add incentives and reputation to opportunistic networks

Greg Bigwood and Tristan Henderson
School of Computer Science
University of St Andrews
St Andrews, Fife, UK
{gjb,tristan}@cs.st-andrews.ac.uk

Abstract—**Opportunistic networks enable users to communicate in the absence of network infrastructure. But forwarding messages in such a network incurs costs for nodes in terms of energy and storage. This may lead to nodes being selfish and not forwarding messages for other nodes, resulting in degraded network performance. This paper presents a novel incentive mechanism for opportunistic networks that uses pre-existing social-network information to detect and punish selfish nodes, incentivising them to participate in the network. Trace-driven simulations demonstrate that our mechanism performs better than existing mechanisms, and that social-network information can also be used to improve existing incentive mechanisms.**

I. INTRODUCTION

Even with the modern ubiquity of the Internet, there still exist certain scenarios where current infrastructure networks may be unable to provide a communication medium. Such scenarios include areas when infrastructure networks are unavailable or overloaded, e.g., developing nations, disaster recovery scenarios, or even at busy concerts or sporting events. In these scenarios we can leverage the social network of human encounters to provide a mechanism for exchanging information: an *Opportunistic Network* [1]. As people encounter each other, their wireless devices such as mobile phones can communicate wirelessly, using every available opportunity, to forward information from person to person. These opportunistic networks have been proposed for applications ranging from messaging, participatory sensing and crowdsourced data-retrieval, to ubiquitous mobile social networks.

Opportunistic networking relies on cooperation between nodes, that is, the users participating in the network, to perform efficiently. Opportunistic routing protocols depend on nodes forwarding messages for each other, as otherwise the only delivery mechanism would be for the creator of a message to encounter the message destination node and deliver the message directly. Cooperative forwarding, however, incurs a cost to the forwarding nodes, both in terms of energy (battery power) and storage (the space required to store forwarded messages). Both of these are a constrained resource in mobile devices such as those used in opportunistic networks.

Due to these costs, nodes may wish to avoid the costs associated with participation in an opportunistic network, by not forwarding messages for other nodes. Figure 1 shows the results of an opportunistic network simulation where nodes
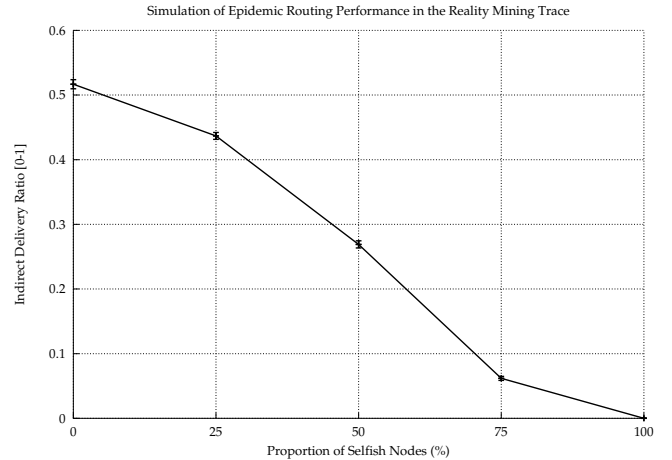

Fig. 1. Simulation of an opportunistic network application using Epidemic routing [2], and the Reality Mining mobile-phone trace [3]. As the proportion of selfish nodes in the network increases, network performance in terms of delivery ratio decreases.

act selfishly: the performance of the network, i.e., the number of messages delivered, decreases rapidly as the percentage of selfish nodes increases. If we can detect and discourage selfish behaviour, it might be possible to achieve the same performance as if no nodes are selfish, even if all the nodes have a propensity for selfish behaviour.

But how can we create incentives for nodes to cooperate? This paper investigates how to use social-network information to do so. We present an incentive mechanism for opportunistic networks, IRONMAN (Incentives and Reputation for Opportunistic Networks using sociAl Networks), that uses social-network information to bootstrap the detection and discouragement of selfishness. We demonstrate IRONMAN's superior performance over two existing incentive mechanisms, and show how to improve existing mechanisms by using social-network information.

The contributions of this paper are to show that:
- existing incentive mechanisms are inappropriate for opportunistic networks.
- social-network information can bootstrap a trust mechanism to discourage selfishness in opportunistic networks.

Next, we discuss existing incentive mechanisms. We then introduce our social-network-based incentive mechanism in

Section III. Section IV describes a set of trace-driven simulations using three real-world traces, two routing protocols and four different incentive mechanisms. Section V shows that our protocol performs well across all traces and protocols. Finally we conclude and discuss future work.

II. RELATED WORK

Incentives, reputation and trust have been extensively studied in peer-to-peer and mobile ad hoc networks, and more recently, in sensor and delay-tolerant (DTN) networks. We draw on work from these other fields to tailor an approach suitable for opportunistic networks.

To combat selfishness, it must first be detected. Several approaches use "watchdog" mechanisms [4]–[7], where a third node oversees a message exchange between two nodes to verify its authenticity. Such an approach, however, is inappropriate for opportunistic networks, as routes are rarely static and the inter-contact times are large; neighbours are not consistently available to monitor the behaviour of one another. The most common detection approach is for all nodes to monitor all their own encounters, and to exchange opinions when they interact. Nodes then use these collated opinion data to make decisions on the trustworthiness of individual nodes.

Many different mechanisms exist to create incentives to discourage selfishness [8]: from bartering (a direct exchange of services), to currency (behaviour that benefits other users earns measurable credits exchangeable for services). In the middle of this spectrum lies asynchronous bilateral trading: nodes perform actions to benefit one another, but not necessarily at the same point in time. Nodes can maintain a concept of credits using this approach [8], but limits on the number of credits are frequently needed to prevent credit explosion. Kangasharju et al. use a similar approach for opportunistic networks [9] .

Wang and Li's routing protocol for selfish and rational wireless ad hoc networks assumes nodes compensate each other for forwarding cost [10]. Their scheme however, assumes that nodes in the multicast group do not charge each other for data exchange. Suri and Narahari go further and develop a scheme BIC-B, in which all nodes may charge one another for forwarding [11]. BIC-B however, assumes a centralised payment arbiter can allocate compensation for forwarding cost based on knowledge of the forwarding paths, making it inappropriate for use in opportunistic networks.

Yu et al.'s reputation system for peer-to-peer networks has nodes build opinions of other nodes by analysing the quality of service (QoS) that they receive from these nodes [12]. A rating-discovery algorithm maintains consistency of ratings across the network. Peer-to-peer networks, however, have different properties to opportunistic networks. Even though there is potentially high churn in peer-to-peer networks, it is generally assumed that direct connectivity between any two nodes in the peer-to-peer network is possible, which is unlikely to be true in an opportunistic network.

For a disconnected opportunistic network, it is therefore necessary to rely on the encounters between nodes as the only way to exchange data and incentive mechanism control traffic. To verify opinions, we must be able to prove that the opinion is based on a real experience–how can we prove that messages were exchanged, or that encounters occurred? One way to validate encounters is to use encounter tickets [13], a cryptographic mechanism nodes can use to prove encounters and message exchanges took place. This allows nodes to build up a history of message exchanges to use to construct an opinion of other nodes. Nodes can exchange encounter tickets and opinions during encounters.

Lu et al. propose an encounter-ticket-based incentive mechanism [14], but this requires a trusted authority (an out-of-band oracle), which makes it inappropriate for opportunistic networks. Li et al. use a history-based approach [15] which floods control messages to the network, potentially consuming lots of nodes' resources.

RELICS [16] encourages cooperation through ranking. Nodes estimate the likelihood of message delivery for each of the nodes they encounter, and use this to rank nodes. A node's rank is improved by being on the forwarding path of successful delivery. Nodes adjust their energy expenditure to meet a desired delivery ratio threshold (decided *a priori*). By expending more energy (forwarding more messages), nodes can hope to deliver more messages, increasing their rank with other nodes. Similarly if their delivery ratio is above the threshold, nodes drop their energy expenditure.

To summarise, none of the existing mechanisms work where infrastructure connectivity and delivery acknowledgements cannot be assumed. We now present such a mechanism.

III. AN INCENTIVE MECHANISM FOR OPPORTUNISTIC NETWORKS

Opportunistic networks exploit the interconnections of individuals as they go about their daily lives. In society we form ties and connections with people around us, be it work colleagues, friends or family. Our goal is to use a record of these social-network data from self-reported social networks (SRSNs) to bootstrap an incentive mechanism for opportunistic networks. SRSNs can be obtained through interview, or from an online social network (e.g., Facebook friends lists).

By viewing those members of the opportunistic network that are also in a node's SRSN as more trustworthy, we can exploit the implicit trust relationships provided by the users. Detecting selfish behaviour quickly is important, as it reduces the amount of transmissions to (and due to) selfish nodes, and therefore the energy wasted. We must balance this, however, against being too cautious and presuming all nodes are selfish. Most existing mechanisms are not bootstrapped to work from network start-up; we use SRSNs to provide reputation for nodes. We assume that individuals have implicit trust with members of their SRSN: therefore, when the network starts up, nodes assign higher trust values to nodes in their SRSN.

A. Detecting selfishness

We now present our IRONMAN mechanism. Consider the following scenario (Figure 2): Alice wishes to send a package to Bob. She first meets Eve, however, and gives Eve the
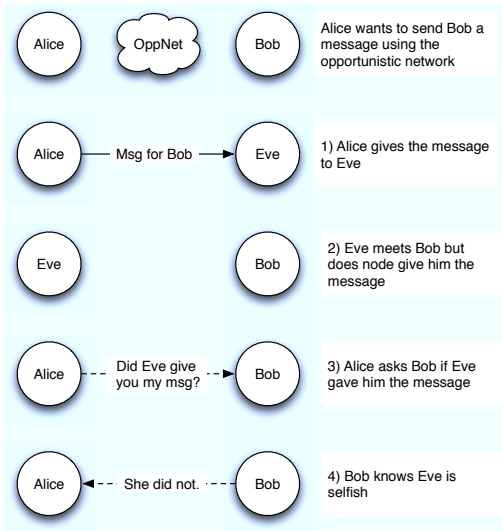
Fig. 2. Nodes keep history of their encounters and message exchanges. When nodes meet these histories are exchanged to detect selfishness.

package, believing Eve will meet Bob before Alice does. Eve then meets Bob, and because Eve is selfish, does not give Bob the package. Yet later, Alice meets Bob, and they discuss their encounters. Alice mentions to Bob that she gave Eve a package for Bob. Bob knows he met Eve, and therefore knows Eve was being selfish by withholding the package. We extend this analogy to opportunistic networks.

If nodes can store a history of encounter times and messages exchanged, and exchange histories during encounters, we can detect selfishness and altruism as seen in Algorithm 1. If a node detects another as selfish, the detecting node decrements its rating for the selfish node by the *behaviour constant x*. Similarly, when nodes pass on a message for which they are not the source: the receiver marks them as altruistic, and their rating of the node receiving the message is incremented by *x*. Additive increase and decrease are used to reduce the effect of false positives, which can arise when a node pushes a message out of its buffers due to congestion (giving the appearance that it deliberately dropped the message).

Algorithm 1 IRONMAN Selfishness detection

1: $x \leftarrow$ *behaviour constant*
function EncounterNode(*B*):
1: history_tuples $\leftarrow$ [(*exchange_time*, *msg_id*, *msg_source*, *node_seen*)]
2: exchange forwarding history with *B*
3: **for all** *message_exchanges* in foreign_history **do**
4: **if** *exchange_time* > last encounter with *B* **then**
5:   **if** *msg_destination* == *my_id* **then**
6:    **if** last encounter with *node_seen* > last encounter with *B* **then**
7:     **if** *node_seen* did not give us *msg_id* **then**
8:      Rating$_{node\_seen}$ $\leftarrow$ Rating$_{node\_seen}$ $- x$
function ReceiveMessage(*other_node*, *msg_src*):
1: **if** *other_node* $\neq$ *msg_src* **then**
2: Rating$_{other\_node}$ $\leftarrow$ Rating$_{other\_node}$ $+ x$

Nodes store local ratings of encountered nodes, and ex-

change these during encounters. An encountered node's *trust score* is the sum of the local rating and foreign ratings. Upon receiving a message the node checks if the source of the message is the node forwarding. If so, and if the *trust score* is not greater than the *trust threshold*, then the receiving node will discard the message and notify the forwarding node that it has been detected as selfish.

Nodes do not accept messages for which they believe the source of the message to be selfish. To allow nodes that have been deemed as selfish to improve their trust score, nodes do pass messages to selfish nodes, allowing them to forward these messages and therefore improve their ratings. This approach does not punish nodes that are rarely given messages to forward, it only punishes those that could have given a message to a destination but did not. To prove that encounters took place we assume the presence of encounter tickets [13]. Nodes use this cryptographic mechanism to prove that they exchanged messages, by obtaining a signed receipt of message exchange.

While nodes do not need synchronised clocks, which can be difficult in an opportunistic network, they must agree on the relative ordering of encounters. When nodes encounter one another they exchange the time they believe the current encounter is taking place at; nodes can thus determine the time when the encounters in the foreign history took place relative to their own opinion of the correct time. Nodes can then use this information despite potential differences in the perceived time on the nodes. A similar mechanism to IRONMAN could be used at the clock synchronisation layer to detect lying about clock times. We leave this, however, to future work.

IV. EVALUATION

We evaluate IRONMAN's performance through trace-driven simulation of a simple message-passing application. As the performance of an opportunistic network may vary depending on the connectivity patterns of the nodes, we use three real-world traces in our analysis:

1) Our "SASSY" connectivity trace, available in the CRAWDAD data archive [17]. 24 individuals carried T-mote sensor nodes for 3 months [18]. We use the motes' ZigBee radios to detect co-location and collected the participants' *Facebook* "friends" as SRSNs.
2) The MIT Reality Mining (RM) trace [3]. 99 individuals carried mobile phones using Bluetooth to detect co-location. We use users' phone contact lists as SRSNs.
3) The HOPE dataset [19] of the movements of RFID tags carried by 767 attendees at the Hackers On Planet Earth conference. Participants registered their interest in topics and specific sessions before attending the conference, which we use as an SRSN, and derive encounters from the RFID readings. As the dataset is dense, and as most of the timetable is taken up by hour-long talks, we merge all contacts between pairs of nodes within one hour.

We believe that these traces capture a wide variety of possible network scenarios. To avoid the effects of warm-up or cool-down, where the recent/impending start/end of the trace

may affect node behaviour, we divide the traces into segments and use the most central segment. As the SASSY trace lasts over two months we split it into two 30-day segments and a 20 day segment and use the second segment. The nine-month RM trace is divided into three 30-day periods, from the beginning, middle and end of the trace respectively, and we use the middle segment. For the HOPE trace we use the second day of three.

Table I shows the different properties of the overall traces and the SRSNs. For the RM and HOPE traces the number of edges, clustering coefficient and graph density are higher in the encounter data than in the SRSN data. In other words, there are longer paths in the SRSN data, which might make them useful for building a trust mechanism. Nodes who are further away in the SRSN network might be less trusted, despite their frequent proximity in the trace network, reflecting the idea of familiar strangers [20]: nodes who you encounter frequently yet you do not know well.

A. Routing protocols

We analyse IRONMAN running over two different routing protocols:

1) Epidemic [2]: Nodes forward messages to any encountered nodes that do not already have a copy.
2) Spray-and-Wait [21]: Messages have a finite number of copies. Nodes give 50% of their copies of a message to an encountered if it does not already have a copy. Once nodes have only one copy left they only give the message to the destination. We treat the number of message copies as the number of nodes in the network, following [22].

We simulate performance at five different levels of selfishness following [23]: 0%, 50%, and 100%. Given Xu et al.'s finding [24] that the altruism of high-degree nodes is most important for mitigating the impact of selfishness, we choose the highest degree nodes to be selfish, so as to maximise the impact on our simulations.

B. Incentive mechanisms

We compare IRONMAN against two existing incentive mechanisms, and modifications of these two mechanisms to use SRSN information:

- IRONMAN: the mechanism outlined in Section III. We use a value of 100 as the default local rating for SRSN nodes, 50 for unknown nodes and 50 for the trust threshold. We use 50 as the behavioural constant.
- YSS: the peer-to-peer reputation mechanism developed by Yu, Singh and Sycara [12]. For the QoS parameter required to measure nodes' behaviour, we use the proportion of messages exchanged altruistically, detected using the same approach described in Section III-A. Where possible we used the thresholds outlined in their paper: the default opinion of nodes that are not known is 0.5, which is the same value as the trust threshold. We use their exponential approach to weighting opinions and credibility of opinions.

- RELICS+S: A modified version of RELICS [16], as representative of the state-of-the-art in incentive mechanisms for opportunistic networks. We attempt to use parameters as described in their paper: 0.8 for the desired delivery ratio threshold, 0.373mAh as the initial energy level, one hour as the energy epoch, and 14.30mAh for the increase in energy allowed during each epoch. Nodes are given a starting rank of two, allowing them to send two messages before being required to forward on behalf of other nodes. Estimated delivery probabilities are 1.0 if a node is the source of a message, and 0.5 otherwise. As RELICS, unlike the other mechanisms, uses delivery receipts, we simulate these, but assume that receipts have no forwarding cost, to maximise the potential performance of RELICS. As RELICS does not actively detect selfishness, we modified the mechanism to treat nodes whose rating is below the initial value of two as selfish, we call this RELICS+S.
- YSS+SRSN: Here we bootstrap the mechanism so that nodes give members of their social network an opinion of 1.0 (complete trust). We leave the default trust level for other nodes as 0.5 and we change the trust mechanism so that nodes only take into account the opinions of members of their social network.
- As a control, we also consider performance when no incentive mechanism is in place.

When a node is detected as selfish it ceases to behave selfishly. The node may continue to be punished, however, as other nodes need to detect its altruistic behaviour before trusting it again.

C. Scenarios and metrics

We perform simulations under two different scenarios, to highlight different features of the mechanisms:

1) a scenario with no resource constraints: nodes have infinite buffer space and energy and messages have an infinite time-to-live (TTL) value. There will be no false positives of selfishness generated in this scenario, as nodes will not drop messages due to full buffers.
2) finite buffers, energy and TTL, as listed in Table II.

We simulate ten runs per scenario, per incentive mechanism, per trace. We calculate the delivery ratio (number of messages delivered over messages sent) to see the difference in performance across incentive mechanisms.

15% of each trace is used as a warm-up period, where no messages are created or sent, but nodes may build up reputation information. All message senders and destinations are picked from an exponential distribution. For the SASSY and RM traces, we exponentially distribute the message creation times throughout the day. As the HOPE trace only lasts one day, we uniformly distribute the message creation times throughout the day, to prevent messages from being created with no time left in the trace for them to be sent.

As nodes in a real deployment would have memory limits, we restrict the size of the history of recent encounters for

IRONMAN and both YSS mechanisms to the most recent 1000 entries, in all scenarios.

To study the performance of the incentive mechanisms we consider the negative impact of selfishness on the network, using the following metrics:

1) *Detection Time*. The time that it takes a mechanism to correctly detect selfish behaviour in a node. This is the average time between a node choosing to behave selfishly, and the time that a node is detected as selfish. A mechanism with a low detection time will minimise the impact selfish behaviour has on the network.

2) *Detection Accuracy*. The proportion of selfish nodes that were correctly detected as selfish by a mechanism. An ideal mechanism will have a low Detection Time and high Detection Accuracy.

3) *Selfishness Cost*. The proportion of forwarded messages (medium accesses) that were generated as a result of a node creating a message while it was selfish. In some respects this can be seen as the "goodput" of a network with selfish nodes; a mechanism with a low Selfishness Cost is effectively maximising the use of the network by cooperative nodes.

We do not consider measuring the reputation changes as a performance metric as in [12], as measuring impact of the incentive mechanism on the performance of the network is sufficient to demonstrate the merit of the mechanism.

TABLE II
SIMULATION ROUTING PARAMETERS

Parameter	Value (SASSY/RM/HOPE)
TTL of messages	10 days / 2 days / 2 hours
Message frequency	1 per node per day
Simulation length	30 days / 30 days / 1 day
Message size (MB)	1
Buffer size (MB)	2000
Loss per second (mAh)	1.9×10^{-6}
Time to send bundle (s)	34
Max energy (mAh)	1200
Energy per send (mAh)	0.4
Charge time (h)	8

V. RESULTS

We now examine the performance of the incentive mechanisms in our simulations.

A. Infinite buffer, energy and TTL scenario

Figures 3(a)–3(c) show network performance (in terms of delivery ratio) across the three traces. It can be seen that IRONMAN performs the best of the evaluated mechanisms.

All the mechanisms have quite high detection times, due to intermediate nodes infrequently encountering destination nodes, but IRONMAN has a higher detection accuracy and lower detection time than all the other mechanisms in the SASSY and RM traces (Figures 4(a) and 4(b)). In the much denser HOPE trace, even though the detection accuracy is lower than the YSS-based mechanisms (Figure 4(c)), the resulting delivery ratio (Figure 3(c)) is higher because both YSS-based mechanisms discard more messages from selfish nodes. Indeed, in this trace (Figure 3(c)), IRONMAN is not only the sole mechanism that performs better than having no detection mechanism at all, but it performs almost as well with 100% of nodes acting selfish as 0%.

In addition to the fastest and most accurate detection of selfishness, IRONMAN has a lower or equivalent selfishness cost than the other mechanisms (Figures 5(a)–5(c)). In other words, IRONMAN is successful at ensuring that the network is predominantly used by cooperating nodes.

When the YSS mechanism is modified to use social network information, it performs the same, or better, than the original mechanism (Figures 3(a)–3(c)). Figures 4(a)–4(c) show that while YSS+SRSN has a slower detection time than YSS in two of the traces, it has a higher delivery ratio, as it does not drop as many messages from nodes perceived as selfish.

Note that to save space, we have omitted the results for Spray-and-Wait routing, but the relative performance of the mechanisms is the same as when using epidemic routing.

### B. Resource-constrained scenario

As one might expect, when we consider the effects of energy, buffer and TTL, we see that network performance drops. In the RM and HOPE traces (Figures 6(b)–6(c)), IRONMAN has the highest delivery ratios, while in the SASSY trace (Figure 6(a)), all mechanisms perform similarly. IRONMAN, however, detects a larger proportion of selfish nodes (Figure 7(a)), has a lower detection time and a lower selfishness cost (Figure 8(a)).

The relative detection time is not consistent across the traces, however. YSS and YSS+SRSN have a lower detection time and higher detection accuracy in the HOPE trace (Figure 7(c)) than IRONMAN; the density of the HOPE trace means the low detection time of YSS and YSS+SRSN (a result of the exponential weightings of ratings/opinions) causes both mechanisms to detect a higher proportion of nodes than both IRONMAN and RELICS+S, and results in a lower detection time. As in the infinite scenario, however, IRONMAN still has a higher delivery ratio (Figure 6(c)), despite the lower accuracy and detection time.
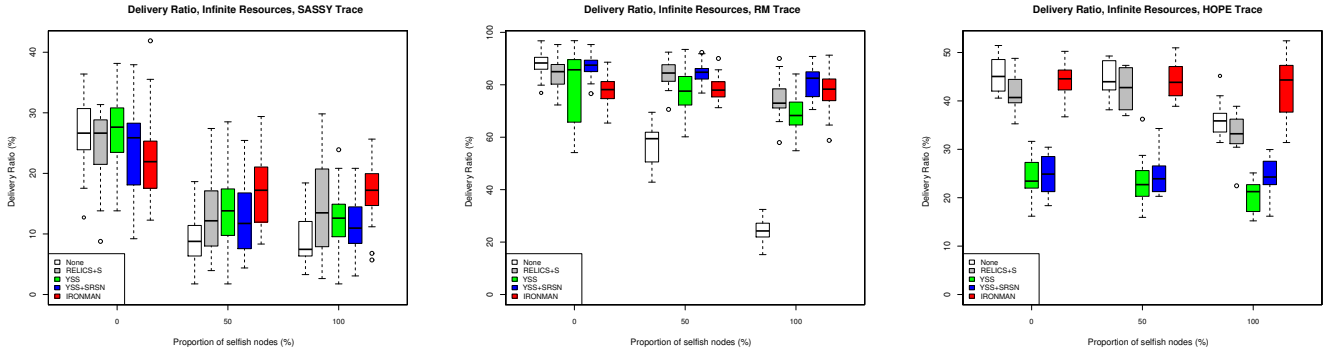
(a) All the mechanisms perform similarly. IRON-MAN and RELICS+S are the best performing when 100% of the nodes are selfish; however, IRONMAN has less variance.

(b) All the mechanisms perform similarly, with YSS performing slightly better than the others at high levels of selfishness.

(c) IRONMAN performs equivalent to having no selfish nodes in the network. Other mechanisms detect selfishness but do not allow enough forwarding: YSS and YSS+SRSN drop messages from nodes they detect as selfish, and RELICS+S's energy monitor allows too little forwarding.

Fig. 3. Incentive mechanism performance in the three traces under epidemic routing, with infinite buffer, energy and TTL.
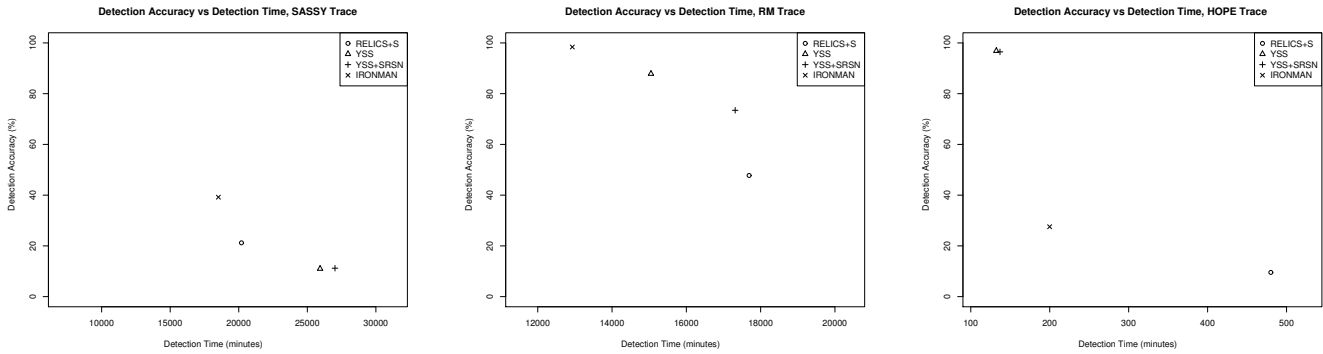
(a) IRONMAN performs the best, with the highest accuracy in the lowest time.

(b) IRONMAN again performs best. YSS performs better than YSS+SRSN as it does not trust as many nodes implicitly.

(c) YSS+SRSN and YSS perform best, with low detection time and high accuracy. In spite of this, IRONMAN still has the highest delivery ratio (Figure 6(c)).

Fig. 4. Detection Accuracy against detection time, when 100% of nodes are selfish. Infinite buffer, energy and TTL.

In the HOPE trace, the delivery ratios of YSS and YSS+SRSN do not change as selfishness in the network increases, as these mechanisms are able to detect selfishness and drop messages from those selfish nodes, thereby reducing the overall performance of the network. YSS and YSS+SRSN have similar Selfishness Cost results to IRONMAN however, as YSS and YSS+SRSN do not allow nodes that have become altruistic to forward many messages. If a group of nodes all believe each other are selfish, they will drop the messages created by the other nodes in the group. As the only way to build up a good reputation is to forward messages for other nodes, if nodes drop all incoming messages they can not build up enough reputation to have their own messages forwarded. The delivery ratio therefore remains low, and the Selfishness Cost remains high, as is the case for YSS and YSS+SRSN in the HOPE trace.

Again we see that RELICS+S does not perform as well as IRONMAN, with a lower delivery ratio (Figures 6(a)–6(c)). This is because the energy monitor does not allow for nodes

to forward sufficient messages. RELICS+S has a low detection accuracy in all traces (Figures 7(a)–7(c)), and a high detection time in all but the SASSY trace. This is because RELICS+S does not detect selfishness well enough, a problem exacerbated by the reduced forwarding opportunities. Figures 8(a)–8(c) show that IRONMAN has the lowest selfishness cost in all traces; IRONMAN is again the best at reducing the overall impact of selfish nodes on the network.

Overall we see that IRONMAN can perform as well as having no selfishness in the network, and SRSN-based mechanisms are always the best (or equivalent to) the best performing mechanism in the network. The exception is RELICS+S, which continues to perform badly, because the time to adjust to the correct energy level causes nodes to miss out on forwarding opportunities.

Note that again for Spray-and-Wait, the relative performance of the mechanisms is the same as for epidemic routing.
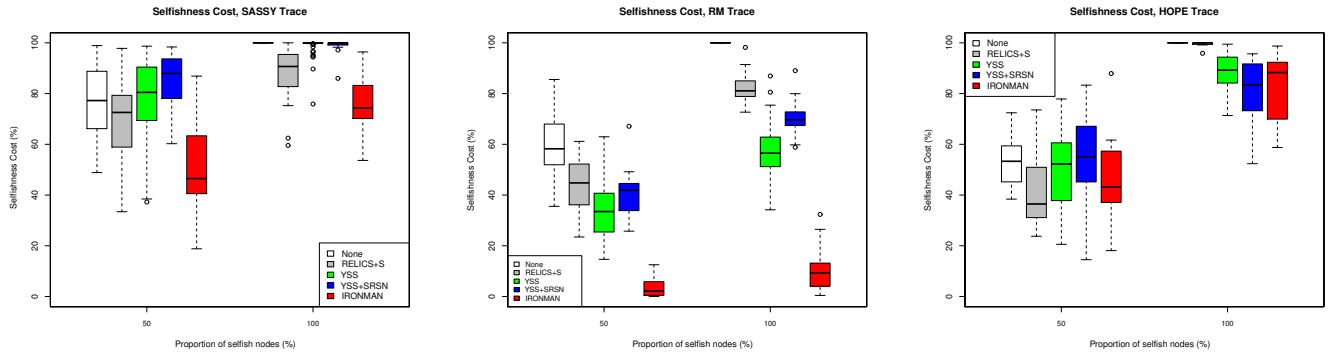
(a) IRONMAN performs best, as its low detection time ensures that more nodes are incentivised away from selfishness before sending messages.

(b) IRONMAN performs very well, ensuring almost all medium accesses are from altruistic nodes.

(c) All mechanisms perform similarly apart from when all nodes in the network are selfish. The energy model in RELICS penalises altruistic nodes.

Fig. 5. Selfishness Cost under epidemic routing and infinite buffer, energy and message TTLs.
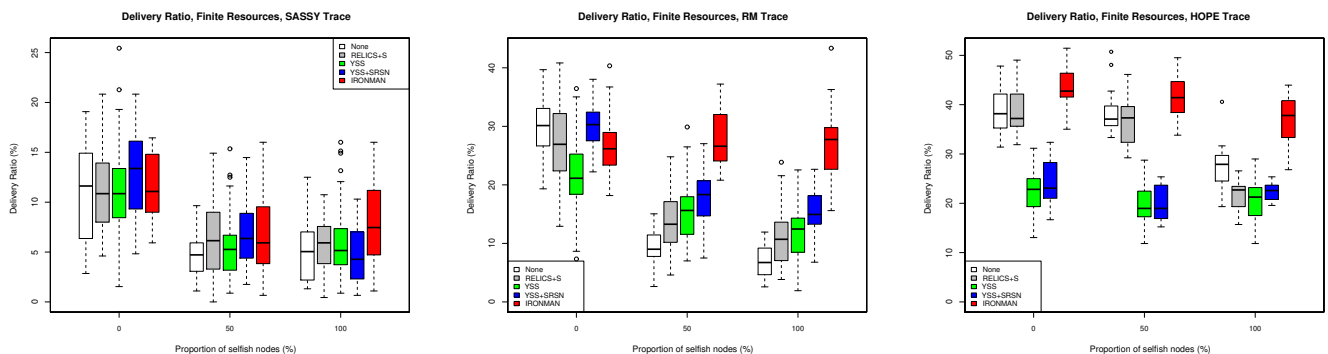
(a) All mechanisms perform similarly, with IRONMAN performing slightly better at 100% selfishness.

(b) IRONMAN performs far better than the other mechanisms, performing as well as having no selfish nodes in the network.

(c) The level of selfishness does not affect normal delivery, however almost all the mechanisms apart from IRONMAN do not perform well at 100% selfishness.

Fig. 6. Incentive mechanism performance under epidemic routing, with finite buffer, energy and TTL.

VI. CONCLUSIONS AND FUTURE WORK

We have introduced IRONMAN, an incentive mechanism for opportunistic networks that uses pre-existing social-network information to bootstrap trust relationships. Unlike existing mechanisms, IRONMAN does not require an oracle or infrastructure network, nor delivery receipts. We have demonstrated that IRONMAN outperforms existing incentive mechanisms, with accurate detection of selfish nodes in a timely manner, and improved delivery performance in the presence of selfishness. As a result, IRONMAN is able to maximise the proportion of the network that is used by cooperating nodes. We have also shown that social-network information can be used to improve existing incentive mechanisms in a similar manner. We believe that this use of social-network information will prove a fruitful topic for researchers in this and similar areas. For instance, is it possible to use social-network information to improve incentive mechanisms for peer-to-peer or ad hoc networks?

In future work we intend to explore the interaction between the application-layer social-network information that we exploit for our incentive mechanism, and the use of this information in the application itself. Many opportunistic network applications might themselves involve social networks, for instance, mobile social networks, crowdsourcing, or participatory sensing. Might it be useful to expose trust relationships from the routing layer to the application layer, or vice versa? Or could application-layer detection of misbehaving nodes, such as anomalous crowdsourced data, be used to inform routing decisions? Such further study will require both routing protocol development and application deployment.

We intend to analyse the theoretical reason behind IRONMAN's performance. We also wish to refine our models of social network behaviour. We currently assume that members of the same social network will be more likely to trust each other. But if behaviour is contagious across a social network, as proposed by Fowler and Christakis [25], then perhaps selfish behaviour might also propagate, leading to new incentive challenges.

REFERENCES

[1] L. Pelusi *et al.*, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, pp. 134–141, Nov. 2006.
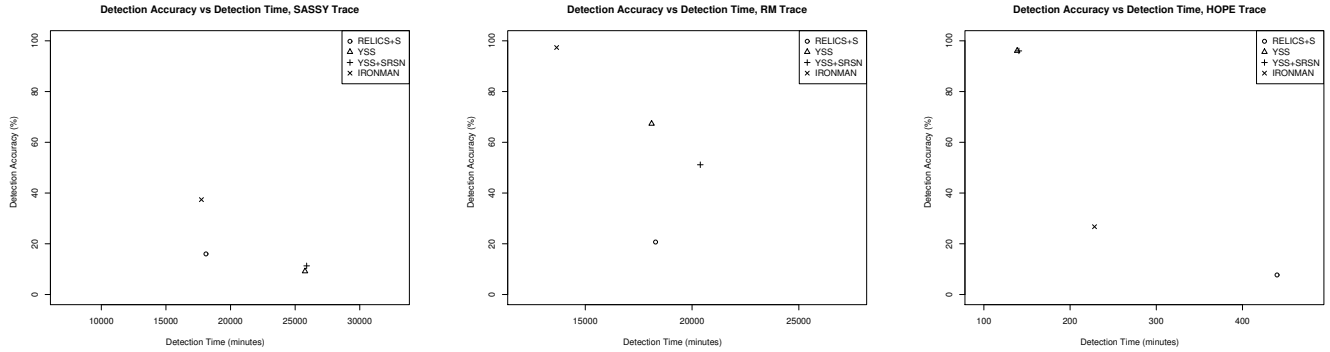
(a) IRONMAN provides the highest accuracy in the lowest time.

(b) IRONMAN again has the highest accuracy in the quickest time.

(c) YSS and YSS+SRSN perform best in the dense HOPE trace.

Fig. 7. Detection Accuracy against detection time, when 100% of nodes are selfish. Finite TTL, buffer and energy.
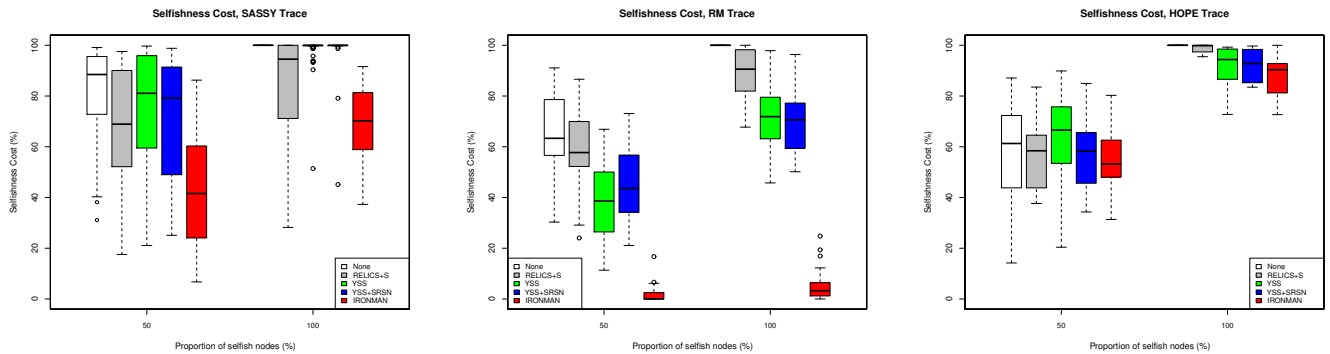
(a) IRONMAN performs well, with the lowest selfishness cost.

(b) IRONMAN greatly outperforms the other mechanisms, with both YSS mechanisms performing similarly, followed by RELICS.

(c) All mechanisms perform similarly, apart from when all nodes are selfish, when RELICS performs almost as badly as having no detection mechanism.

Fig. 8. Selfishness Cost under epidemic routing and finite buffer, energy and TTL.

[2] A. Vahdat *et al.*, "Epidemic Routing for Partially-Connected Ad Hoc Networks," CS-200006, Duke University, Tech. Rep., Apr. 2000.

[3] N. Eagle *et al.*, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, pp. 255–268, May 2006.

[4] S. Buchegger *et al.*, "Performance analysis of the CONFIDANT protocol," in *Proc. MobiHoc '02*, Jun. 2002, pp. 226–236.

[5] R. Mahajan *et al.*, "Sustaining cooperation in multi-hop wireless networks," in *Proc. NSDI '05*, May 2005, pp. 231–244.

[6] S. Marti *et al.*, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. MobiCom '00*, Aug. 2000, pp. 255–265.

[7] T. Anantvalee *et al.*, "Reputation-Based System for Encouraging the Cooperation of Nodes in Mobile Ad Hoc Networks," in *Proc. ICC '07*, Jun. 2007, pp. 3383–3388.

[8] Z. Liu *et al.*, "P2P trading in social networks: the value of staying connected," in *Proc. INFOCOM '10*, Mar. 2010, pp. 2489–2497.

[9] J. Kangasharju *et al.*, "Incentives for Opportunistic Networks," in *Proc. First IEEE International Workshop on Opportunistic Networking*, Mar. 2008, pp. 1684–1689.

[10] W. Wang *et al.*, "Low-cost truthful multicast in selfish and rational wireless ad hoc networks," in *Proc. IEEE MASS 2004*. IEEE, Oct. 2004, pp. 534–536.

[11] N. R. Suri *et al.*, "Broadcast in ad hoc wireless networks with selfish nodes: A bayesian incentive compatibility approach," in *Proc. COMSWARE '07*. IEEE, Jan. 2007, pp. 1–9.

[12] B. Yu *et al.*, "Developing trust in large-scale peer-to-peer systems," in *Proc. First IEEE Symposium on Multi-Agent Security and Survivability*, Dec. 2004.

[13] F. Li *et al.*, "Thwarting Blackhole Attacks in Disruption-Tolerant Networks using Encounter Tickets," in *Proc. INFOCOM '09*, Apr. 2009, pp. 2428–2436.

[14] R. Lu *et al.*, "Pi: A practical incentive protocol for delay tolerant networks," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 1483–1493, Apr. 2010.

[15] N. Li *et al.*, "RADON: reputation-assisted data forwarding in opportunistic networks," in *Proc. Second International Workshop on Mobile Opportunistic Networking*, Feb. 2010, pp. 8–14.

[16] M. Y. Uddin *et al.*, "RELICS: In-network realization of incentives to combat selfishness in DTNs," in *Proc. ICNP*, Oct. 2010, pp. 203–212.

[17] G. Bigwood *et al.*, "CRAWDAD data set st_andrews/sassy (v. 2011-06-03)," Downloaded from http://crawdad.org/st_andrews/sassy, Jun. 2011.

[18] ——, "Exploiting self-reported social networks for routing in ubiquitous computing environments," in *Proc. IEEE SAUCE*, Oct. 2008.

[19] Aestetix *et al.*, "CRAWDAD data set hope/amd (v. 2008-08-07)," Downloaded from http://crawdad.org/hope/amd, Aug. 2008.

[20] E. Yoneki *et al.*, "Visualizing community detection in opportunistic networks," in *Proc. 2nd ACM workshop on Challenged networks*, Sep. 2007, pp. 93–96.

[21] T. Spyropoulos *et al.*, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM workshop on Delay-tolerant networking*, Aug. 2005, pp. 252–259.

[22] A. Panagakis *et al.*, "On the effects of cooperation in DTNs," in *Proc. COMSWARE '07*, Jan. 2007.

[23] G. Resta *et al.*, "The effects of node cooperation level on routing performance in delay tolerant networks," in *Proc. SECON '09*, Jun. 2009, pp. 413–421.

[24] K. Xu *et al.*, "Impact of altruism on opportunistic communications," in *Proc. ICUFN '09*, Jun. 2009, pp. 153–158.

[25] J. H. Fowler *et al.*, "Cooperative behavior cascades in human social networks," *Proc. National Academy of Sciences*, vol. 107, pp. 5334–5338, Mar. 2010.