ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology Ecology Science and Technology (IJARBEST)
Vol. I, Special Issue I, August 2015 in association with VEL TECH HIGH TECH DR. RANGARAJAN DR. SAKUNTHALA ENGINEERING COLLEGE, CHENNAI
National Conference on Recent Technologies for Sustainable Development 2015 [RECHZIG'15] - 28th August 2015

EDUCING SELFISHNESS IN MOBILE ADHOC NETWORKS

1Mr.P.Karthikeyan1 ,[2]P.Vetriselvi, [3]S.Mythili,[4]Sornalatha
[1] Asst. Prof. Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College.
pkarthiekeyan@gmail.com

ABSTRACT

In this paper, Mobile Ad hoc Routing Protocols allows nodes with wireless adapters to communicate with one another without any pre-existing network infrastructure. Educe selfishness of the participants in ad hoc network can affect its overall functioning and sketches a protocol that discourages this kind of behavior. We protect our social mobile networks against message droppers to improve performance by reducing number of replicas and storage requirements. Goals are to maximize message delivery rate, minimize me ssage latency, good strategy could be to forward only to the highest quality nodes.

Index Terms—pocket switched networks, social mobility, selfishness, forwarding protocols, coalitions

1 INTRODUCTION

In the last few years, the diffusion of mobile personal devices exploded. Smartphones are used by people—not only technology gee ks—to communicate, to use applications once run only by desktops, and to organize their life. Typically, these devices can communicate with each other over short distances by using wireless technologies such as bluetooth. In this way, a new kind of network emerges where nodes are carried by people and links appear and disappear as people move and get in contact. These networks, also known as Pocket Switched Networks (PSN)can be key technology to provide innovative services to the users without the need of any fixed infrastructure. Pocket Switched Networks are usually disconnected, are character-ized by social-based mobility and heterogeneous contact rate. Examples of such networks include people at work places, students on university campuses, and citizens in metropolitan areas. The problem of designing efficient forwarding protocols for PSNs has attracted the attention of many researchers. In forwarding protocols, messages are routed from source to destination thanks to intermediate relays. One fundamental and natural question, especially in this setting, is why nodes should accept to use their own energy and bandwidth just to carry other people's messages. Indeed, the protocols in the literature break down immediately if you do not assume that all the nodes cooperate in an altruistic manner. We show this phenomenon, which is intuitive indeed, by a few experiments on Epidemic Forwarding and Delegation Forwarding, two important protocols in the literature.

In this paper, we introduce Epidemic Forwarding and Delegation Forwarding, which are, to the best of our knowledge, the first protocols for packet forwarding in a social mobile setting that leverage on the social aspects of the network to tolerate selfish behavior where random pair-wise exchanges of messages among mobile hosts ensure eventual message deliver y. The goals of Epidemic Routing are to:

i) maximize message delivery rate,
ii) Minimize message latency, and
iii) minimize the total resources consumed in message delivery In delegation forwarding every node is associated with a forwarding quality that may depend on the destination of the message.

This helps us showing formally that no rational node has any incentive to deviate. In other words, our two protocols are strategy proof, i.e., the strategies of following the protocols are Nash Equilibria. However, for simplicity we will use the words protocol and strategy (to follow that protocol) interchange-ably.

Lastly, we check the performance of Epidemic Forwardingand Delegation Forwarding. Quite surprisingly, we discover that some of the mechanisms that we introduce to make these protocols Nash equilibria are also useful to control the number of replicas in the network and push the messages quickly and cheaply far from the community where they have been generated. As a result, Epidemic Forwarding and Delegation Forwarding, besides providing robust-ness in a network where every node is selfish, have nearly the same delay and success rate of their original alter egos, and have a considerably lower cost in terms of number of replicas. Moreover, we also perform m a detailed study of the memory load required.

To the best of our knowledge, this is the first time such a detailed study is performed. We measure the cost generated by each protocol by computing the average storage requirements to forward one message. Our results show that our protocols, aside providing tolerance to selfish behavior, require considerably less storage than their vanilla alter -egos in almost all cases, even including the delay tolerant.

2 RELATED WORKS

A lot of work has been done in building efficient forwarding protocols for Pocket Switched Networks. Many of the protocols in the literature use in sophisticated ways the properties of human mobility .All of them rely on the altruistic cooperation among the nodes. The problem of building mechanism and protocols that can tolerate selfish behavior is an important and modern issue in the design of networking protocols and distributed systems. See, as an important example, the work in .Earlier work has been done to mitigate the impact of selfish behavior in mobile ad hoc networks as well. The solutions can be classified into two main approaches: reputation-based schemes and credit based schemes. In the former schemes, nodes collectively detect misbehaving members and propagate declarations of misbehavior throughout the network. Eventually, other nodes will avoid routes through selfish members. In credit-based approaches, nodes pay and get paid for providing service to others. All these solutions assume the use of public key cryptography for authentication of messages. Regard-less of the performance of these schemes on ad hoc networks, none of them is designed for social mobile networks. Indeed, no previous work is neither designed with a social mobile scenario in mind, nor exploits the social nature of the network or the properties of the movement that such social nature generates.

A barter-based cooperation system to increase message deliver r ate in opportunistic networks. That altruistic static nodes scattered on the network area generate messages downloadable by interested network members in physical proximity. When two nodes meet, they exchange the list of the messages in their buffers and each node decides to download from the other node only the messages of its interest. Then, the nodes start downloading messages till they move out each other's communication range. Though it introduces a novel technique of stimulation of cooperation, their work is oriented to a gossip-like service, where messages are created from special nodes and many other nodes are interested in downloading them, which is a natural incentive for the distribution.

A recent work presents a routing mechanism built upon the willingness (declared by each individual) to forward other individuals' messages. They show that, when forwarding algorithms that use multiple paths are considered, social mobile networks are robust to different distributions of altruism of nodes. To the best of our knowledge their work is the first study aimed to explore altruistic/selfish behavior in these types of networks and encourages for further work in this direction.

3 THE SYSTEM MODEL

3.1 System and Node Properties

In our system model, every node is selfish. This is a realistic scenario, if people can get the same level of service without using part of their battery or part of their wireless uptime or memory without any consequence, they will. And as soon as the first user finds a way to get more while paying less, and publishes the patch of the system software, everybody will download the patch and use it. So, it is reasonable to assume that, if some of the nodes deviate selfishly, after a while everybody will.

We assume that there are no byzantine nodes in the network. We also assume that selfish nodes do not collude. All the nodes in the system are interested in receiving and sending messages, in other words, all the nodes are interested in staying in the system. Nodes are loosely time synchronized. Loose time synchronization is very easyto get, if a precision in the order of the second is enough, like in our protocol. We assume that every control message of our protocols is labeled with a time stamp, though it does not appear in the protocols to keep the presentation clean. The clock is used to check the time-outs, and the time stamp is used when reporting misbehavior to the authority.

Lastly, nodes are capable of making use of public key cryptography—this capability will be used to sign messages and to make sender to destination encryption. Therefore, we assume that ever y node has a public key and the corresponding private key.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology Ecology Science and Technology (IJARBEST)
Vol. I, Special Issue I, August 2015 in association with VEL TECH HIGH TECH DR. RANGARAJAN DR. SAKUNTHALA ENGINEERING COLLEGE, CHENNAI
National Conference on Recent Technologies for Sustainable Development 2015 [RECHZIG'15] - 28th August 2015

3.2 The System Authority and Key Revocation

In our system nodes that join and leave are handled by a central authority. The authority handles new nodes joining the network in a standard way: It identifies the new node and it signs the new node's certificate (or the master public key is handed out to the node in case of an identity-based public key system). More authorities can coexist, as long as they exchange information on nodes that enter and exit the system in real time.

To communicate with the nodes, we assume that the authoritycan use the cellular infrastructure or wireless technology like, e.g., GSM. This technology is very expensive compared with Bluetooth communication used by our forwarding protocols. However, it is used very sparingly. The cellular network can also have some delay, usually due to nodes that are temporarily out of coverage—this is not an issue, even if a revocation due to misbehavior is late, it does not lose its power as a deterrent. Moreover, nodes can pretend to have run out of battery or to be out of coverage just to prevent communication with the authority. These problems can be dealt with quite easily (for example, by forcing the nodes to keep all the cryptographic proofs of their behavior until they are up and with cellular network coverage). However, in the following, we will assume that the cellular network covers the whole network and that nodes are always up and running. Node failures are dealt with as if node switches of the device for a certain amount of time. It is considered to be a legitimate behavior in the system, even though, as we will prove, no node will chose to do so deliberatively for its quality of service will drop during such time.

It is known that public key cryptography is more expensive than symmetric cryptography. However, modern cryptography techniques, like those based on elliptic curves, provide short signatures and cheaper and cheaper computation, which is shown tobe adequate even for sensors. The same is true for identity-based cryptosystems. In addition to this, the delay tolerant nature of the PSNs gives nodes the time to generate and verify signatures.Moreover, in our study we are addressing a network of smartphones or PDAs, which are not-so-small devices.

4. EPIDEMIC FORWARDING

In Epidemic Forwarding, every contact is used as an opportunity to forward messages. When node A meets node B, and A has a message that B does not have, the message is relayed to node. B. Epidemic forwarding with unlimited buffer is often used as a benchmark, it is easy to see that it is impossible to get smaller delay, or higher success rate. However, the overhead in terms of number of copies of the same message is very high. Put simply, many of the for warding protocols in the literature on Pocket Switched Networks have the goal of reducing drastically the overhead without affecting much the delay and the success rate of Epidemic Forwarding.

However, Epidemic Forwarding does not tolerate a scenario in which users can make selfish choices. Indeed, selfish nodes simply drop ever y message the y receive (except those destined to

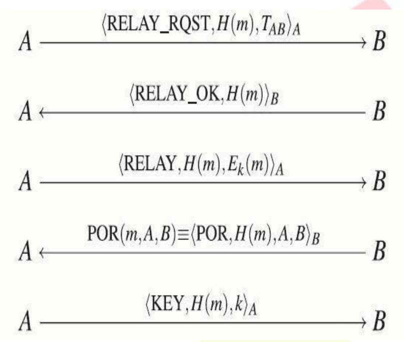
$A \xrightarrow{\langle \text{RELAY\_RQST}, H(m), T_{AB}\rangle_A} B$
$A \xleftarrow{\langle \text{RELAY\_OK}, H(m)\rangle_B} B$
$A \xrightarrow{\langle \text{RELAY}, H(m), E_k(m)\rangle_A} B$
$A \xleftarrow{\text{POR}(m,A,B) \equiv \langle \text{POR}, H(m), A, B\rangle_B} B$
$A \xrightarrow{\langle \text{KEY}, H(m), k\rangle_A} B$

Fig. 1. Protocol of the relay phase (in case node B does not have the message)

We will call message droppers the nodes that implement this simple form of deviation. In this section, we show how to build a version of Epidemic Forwarding, called Epidemic Forwarding that works in a system where every node is selfish. In this way, we protect the network against message droppers and against any other rational deviation. Most of the ideas and techniques that we develop in this section will be used in the more sophisticated.

Epidemic Forwarding consists of three phases: Message generation, relay, and test. The idea behind each phase is as follows: 1) during message generation the message is modified so that a relay candidate has no interest in not accepting it; 2) the relay phase forces nodes to collect the so-called proof of relay to show to previous relays (or source), during the test phase, 3) that they have correctly behaved with the message—this is to make it impossible to relays to drop messages. The details of each phase will be given in the remaining of the section.

4.1 G2G Epidemic Forwarding: The Relay Phase

Once the message is generated, the sender S tries to relay it tothe first two nodes it meets. When node S meets node B, node S starts a session with the possible relay by negotiating a cryptographic session key with node B. This is easily and locally done by using the certificates of the two nodes, signed by the trusted authority. In this way, both identities are authenticated. From this point on, every communication during the session is encrypted with a symmetric algorithm like AES and the session key (to keep the notation clean, this encryption is not shown in the protocols). Node S starts the relay phase by asking node B if it has already handled a message with hash HðmÞ (see Fig. 1, where the role of S is described as done by node Astep 1). Token T $_{AB}$ is a cryptographic proof that node A has completed with the testnode B.

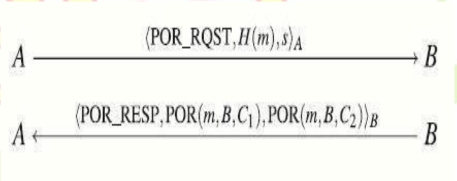
$A \xrightarrow{\langle \text{POR\_RQST}, H(m), s\rangle_A} B$
$A \xleftarrow{\langle \text{POR\_RESP}, \text{POR}(m,B,C_1), \text{POR}(m,B,C_2)\rangle_B} B$

Fig. 2. Protocol of the test phase

In the formal proof, the token will be key to show that also intermediate relays—and not just the sender—has an interest to perform the test phase. More details on how the token T $_{AB}$ is being computed will be given in the description of the test phase (Session 4.2). However, in case this is the first

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology Ecology Science and Technology (IJARBEST)
Vol. I, Special Issue I, August 2015 in association with VEL TECH HIGH TECH DR. RANGARAJAN DR. SAKUNTHALA ENGINEERING COLLEGE, CHENNAI
National Conference on Recent Technologies for Sustainable Development 2015 [RECHZIG'15] - 28th August 2015

interaction, the token can be empty. In case node B has never seen this message, the relay phase goes on (step 2), otherwise node B informs S that it should not be chosen as a relay. Note that node B would not lie, since it still does not know the content of the message, its destination, and, in particular, if node B itself is the destination. In other words, if B deviates and executes a modified version of the protocol in which it declines offers of being a relay without knowing the destination of the message, it won't receive any message, against its own interest.

Node S generates a random key k, and sends message m to B, encrypted with key k (step 3). Then, node B sends a proof of relay to node S which in turn, lastly, sends key k to B, who now knows whether it is the destination of the message or just a relay. Note that the relay phase is only started for messages that have not expired yet.

4.2 G2G Epidemic Forwarding: The Test Phase

Once it realizes that it is a relay for message m, node B follows the same protocol as done by node A, the previous relay. That is, find two other nodes and relay the message to these two nodes by executing the relay phase as shown in Fig. 1. By doing so, it can collect two proofs of relay (PORs) that it will be asked to show, when meeting node A again, during the test phase. If node B is not able either to show the two proofs or to prove to have the message still in its memory (as detailed at Step 2 in Fig. 2), then node A can send by the cellular network

a proof of misbehavior (PoM) to the authority. The proof of misbehavior consists of the proof of relay hPOR; HðmÞ; A; Bi $_B$, signed by node B. The authority, in turn, will revoke node B, if node B is not in the position to prove that A is wrong by showing the two proofs of relay or by proving to have the message still in its memory to the authority (the protocol between the authority and node B is simple and similar to the test phase in Fig. 2). It is important to realize, however, that under our assumptions misbehavior never happens. Indeed, all the nodes are rational—they will not deviate from the protocol since it is against their own interest as we prove more formally later in this paper.

In particular, nodes have no interest in sending fake proofs of misbehavior against other nodes to the authority since it is expensive and it is not going to have any effect.

Only when two proofs are collected the message can be discarded from B's memory. After a time -out _, B can stop looking for relays and can discard ever y information regarding the message. In turn, node A can discard the token T $_{AB}$, in case a test phase between the two nodes is executed. Time -out _ plays the role of the message time to leave (TTL) in Epidemic Forwarding. Therefore, it should be chosen in such a way that the success rate is high enough. Our experiments show that the delay of G2G

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology Ecology Science and Technology (IJARBEST)
Vol. I, Special Issue I, August 2015 in association with VEL TECH HIGH TECH DR. RANGARAJAN DR. SAKUNTHALA ENGINEERING COLLEGE, CHENNAI
National Conference on Recent Technologies for Sustainable Development 2015 [RECHZIG'15] - 28th August 2015

Epidemic Forwarding is very close to the delay of Epidemic Forwarding, and so _ can be chosen as in

its original alter ago without affecting the success rate.

The test phase is started by node A (see Fig. 2), when meeting node B before time-out _. During the test phase, node A challenges node B: Either it has two proofs of rela y, or it still stores the message. In case node B has two proofs of relay, it can reply with the two proofs. The challenge is a simple cryptographic protocol in which node A generates a random seed s and asks node B to send back the value of a hard to compute puzzle of message m and seed s.. By choosing a cost-function that is hard enough to compute, node B is encouraged to relay the message and get the two proofs of relay in all cases in which the probability of meeting node A again is not negligible (below a fixed and small probability). Note that B does not know the seed beforehand; it must be storing the message unless it has found two relays. So, while it is a legitimate part of the strategy to keep the message and not to relay it, it is rational to relay it unless the meetings with the previous relay are very infrequent. In this way,we can ma ke the event that B chooses to store the message rare in such a way that success probability is virtually unaffected. Note thatin many cases A cannot check whether B has correctly computed the puzzle: Node A may have already forwarded m to 2 relays (one is B itself) and thus legitimately dropped the message afterwards.

However, if A is the source, A is interested in storing the message and in checking that B has behaved correctly by verifying the puzzle result.

Since B does not know whether A is the source of m or not, node B has to behave otherwise it can be removed from the system.

Lastly, if A does not start the test phase as required by the protocol, node B is forced to keep proofs of relay in his buffer till time-out _ expires, although it could get rid of them and free the memory if A started the test phase. Thus, the token $T_{A B}$ will not be sent to A, and without it, node B will punish A and not relay any message whose destination is node A until time-out _ expires. If A does not start the test phase this is very important to get node A perform the test phase. In addition, it is not possible for B to fool A by forging any of the two proofs, since they are signed by the two relays. In any case, node A will store the proof received by node B as token T_{AB} until expiration of time-out.

4.4 Coalitions

Consistent with a very large part of the literature, in this paper we assume that nodes do not collude. This is a common assumption, coordination has a cost and in many practical settings people just do not trust each other enough to form a coalition. However, if a few nodes have a strong will to cheat the system, they could deviate in a coordinated manner from our protocols and do better. While this attack is out of the scope of this paper, we would like to point out a few countermeasures that can be worked out to mitigate its effect.

Assume that a set C of nodes, the coalition, deviate to a protocol in which the nodes mark in some secret way the body of messages coming from member s of the coalition, exchange fake proofs of relay when handling messages outside of the coalition, never execute the test-phase within the coalition, and drop messages that are not originated or destined to me mbers of the coalition. Of course, they stick to the protocol when in session with users outside the coalition. To stop this one and similar attacks, we can put in place two mechanism:

Random checks of conformity and rewarding traitors.

Random checks of conformity. The sender of a message, with small probability p, sends the proof of relays got from first relay B on the paths toward destination to the authority. The authority, at a random time before the time-out, checks whether node B has collected two proofs of relay or it is still storing the me ssage. Then, the authority can follow the message at the next relays and check whether the message has not disappeared. The sender has an interest to perform this check, and node B does not know if the previous relay is the sender or not. This check can be ver y costly since the authority uses the expensive cellular network infrastructure.

Howe ver, since the authority can impose a very stiff penalty, like the eviction from the system, this mechanism can be a good deterrent even with an

extremely small probability p, in such a way that the cost can be reasonably low.

Rewarding traitors . A simple observation is that any member ofthe coalition can prove that another node is a member as well. Suppose that node A 2 C, without actually exchanging message, gets a fake proof of relay from node B 2 C. Now, node A has a proof that can nail node B. If node A gives away the fake proof of relay signed by node B to the authority, then the authority can ask node B for a proof of storing the message (or two proofs of relays) and node B will not be able to respond. If the authority rewards node A in such a way that the benefit of betraying is larger than the benefit of being part of the coalition, this attack can be prevented efficiently.

These mechanisms, among others, can be used to extend our protocols in such a way to mitigate the possible presence of coalitions or to limit the possible harm they can make, including the protocols described in the next section. However, the full development of these mechanisms is out of the scope of this paper. Therefore, we stick to the classical assumption that nodes do not collude and proceed with more sophisticated, social-aware for warding protocols Delegation Forwarding builds upon all the techniques that we have developed for Epidemic Forwarding. First, in Delegation Forwarding the quality of the message is changed only when forwarded Delegation Forwarding consists of four phases:

Message generation, relay, test by the sender, and test by the destination. Message generation is just like message generation and in Epidemic Forwarding. Thus, in the next sections, we will describe only the phases that are substantially different from Epidemic Forwarding. As in Epidemic, the relay and the test phases are based

on the idea of making nodes collect proofs of relay and to check relays about their behavior with the message. The test by destination phase is to protect the system against cheaters: very low forwarding qualities declared by nodes are embedded in the message by the source, and, when eventually the destination is reached, are double-checked (forwarding qualities are symmetric) by the destination. In the remaining, we give details on the phases.

5.1 Delegation Forwarding: The Relay Phase and the Test by the Destination Phase

Fig. 3 shows the protocol of the relay phase. Just like Epidemic Forwarding, node A has an interest to start this phase, since it has to collect the proof of rela y for the message. In step 1, node A asks B what is its forwarding quality to destination D (denoted by q $_{B\ D}$). Note that forwarding quality q $_{B\ D}$must be provable. That is, it must be computed based on signed interactions. This is why nodes A and B exchange signed messages carrying q $_{AB}$ and q $_{B\ A}$at every session.

Note that, if q $_{AB}$ ¼6 q $_{B\ A}$, then one of the

$$A \xrightarrow{\langle FQ\_RQST, H(m), D', T_{AB}, q_{AB} \rangle_A} B \qquad (1)$$

$$A \xleftarrow{\langle FQ\_RESP, B, D', q_{BD'}, q_{BA} \rangle_B} B \qquad (2)$$

$$A \xrightarrow{\langle RELAY, H(m), q_m, E_k(m) \rangle_A} B \qquad (3)$$

$$A \xleftarrow{\langle POR, H(m), A, B, D', q_m, q_{BD'} \rangle_B} B \qquad (4)$$

$$A \xrightarrow{\langle KEY, H(m), k \rangle_A} B \qquad (5)$$

Fig. 3. G2G Delegation Forwarding: Protocol of the relay phase.

two players is cheating, and the loyal player can prove it based on the quality exchanged in the previous session. Again, in the request node A includes a token that certifies that the last interaction has completed with the test phase. Node B replies with its forwarding quality (we will see later why B has no interest in lying). When the destination of m is different from B, D ^{0} is the actual destination D;when the destination of m is B, D ^{0} is chosen as a random node different from B.

5.2 Delegation Forwarding: The Test Phase

The test by the sender is executed by the relay node as in Epidemic

Forwarding. Assume that node B has received the message from the node A. When B gets in contact with A again, node B is tested and, just like in Epidemic Forwarding, S. In this way, it is guaranteed that it is not rational to become a message dropper. More than that, this phase is also important to check that B is not a cheater, that is it

has not reduced the message quality q_m to get rid of the message quickly. Indeed, A can check whether

$^{q}B\ D^{1/4\ q}\ m1\ ^{<\ q}\ C1D\ ^{1/4\ q}\ m2\ ^{<\ q}\ C\ 2\ D\ ^{:}$

The second equality in this equation is true since the quality of the messages is changed only when forwarded Since we are sure that nodes do not lie, than we know that $q_{B\ D}$, q_{C1D} , and q_{C2D} are sound. Therefore, also q and conse-quently node B has not selfishly modified the forwarding quality of the message to convince B to takeit. Similarly, we also know that $q_{m\ 2}$ ¼ q_{C1D} and so node B has not cheated with node C_2 as well. Note that it is not possible for B to forge fake proofs or fake forwarding quality declarations of another node C. Indeed, as in G2G Epidemic, proofs and for warding qualities are signed. Again (as in G2G Epidemic), node A will store the proofs received by node B as token T_{AB} until expiration of time-out _. Lastly, if A does not start the test phase as required by the protocol, node B will not relay any message whose destination is node A until time-out _ expires.

To summarize, by using the techniques developed for Epidemic Forwarding and specific techniques for Delegation Forwarding, we can get the following result: Delegation Forwarding. is a Nash equilibrium.

5 DELEGATION FORWARDING

Delegation Forwarding is a class of protocols that have been shown to perform very well. In Delegation Forwarding, every node is associated with a for warding quality that may depend on the destination of the message at stake. When a

message is generated, it is associated with the forwarding quality of the sender. Then, the message is forwarded from node to node, creating a new replica of the message at each step, according to the following protocol: When a relay node A gets in contact with a possible further relay B, node A checks whether the forwarding quality of B is higher than the for warding quality of the message. If this is the case, node A creates a replica of the message, labels both messages with the forwarding quality of node B, and forwards one of the two replicas to B. Otherwise, the message is not forwarded.

Delegation forwarding, in many of its flavors, has been shown to reduce considerably the cost of forwarding (that is, the number of replicas), without reducing considerably success rate and delay. However, just like Epidemic Forwarding, it is far from being Nash equilibrium. A selfish node can easily send messages and receive messages without taking care of relaying any other message. It is also easy to see that it is not enough to translate all the techniques used in Epidemic Forwarding in order to get a version of Delegation Forwarding that is Nash equilibrium.

Simply speaking, the techniques we developed to build Epidemic Forwarding can be used to stop message droppers in Delegation Forwarding, but are not enough to make it a Nash Equilibrium. Indeed, selfish nodes have many other rational ways to deviate in these more sophisticated protocols. First, nodes can lie on their for-warding quality. The y can claim that their quality is zero and get their

messages served without participating actively. We will call these nodes liars. Not only that, selfish nodes can change the forwarding quality of the message to zero, in such a way to get rid of the message soon—they would be able to relay it to the first two nodes they meet. We will call these nodes cheaters. Of course, cheaters are less vicious than liars, in our setting. However, we will show how to build a version of Delegation Forwarding that is a Nash equilibrium. Just like what we did with Epidemic Forwarding, our approach is not to add patches against liars and cheaters or incentives

for altruistic nodes, our approach is to design a protocol such that, step by step, it can formally be shown that every rational player in the protocol cannot but follow the protocol truthfully. In this way, we protect our system against liars, cheaters, and any other possible way to deviate rationally.

Delegation Destination Frequency. Node A for wards message m to node B if node B has contacted m ^{0} s destination more frequently than any other node that the copy of the message m c arried by A has seen so far.

Delegation Destination Last Contact. Node A forwards message m to node B if node B has contacted m ^{0} s destination more recently than any other node that the copy of the message m c arried by A has seen so far.

In the above definitions the forwarding quality q $_{A}$ $_{B}$ is, respectively, the number of encounters between node A and node B, and the time of the last encounter between node A and node B. Since encounters are committed by both nodes with a

commonly agreed time, it is easy to see that q $_{AB}$ ¼ q $_{BA}$for ever y pair of nodes A and B.

6. Selfishness and Selfishness with Outsiders

In social environments, it is natural to consider two different ways of being selfish. The first is just selfishness— nodes that can deviate from the protocol with the goal of maximizing their personal interest. The second is selfishness with outsiders—nodes that can deviate from the protocol for their personal interest only when this does not damage people from the same community. This notion is natural since it comes from our personal experience: Some people can tend to be truthful with those they care about, and selfish with outsiders. Formally, it is just vanilla selfishness with a different objective function. However, it is useful to define it as an independent notion. To implement self-ishness with outsiders, we use the k-clique algorithm for community detection on each data trace. Nodes that are selfish with outsiders deviate from the protocol only in sessions with nodes from other communities.

7 CONCLUSIONS

In this paper, we have presented Epidemic

For warding and Delegation Forwarding, the first protocols for message forwarding that work under the assumption that all the nodes in the network are selfish. We formally show that the protocols are Nash equilibria Quite surpris-ingly, protocols also outperforms their alter egos in terms of cost, while being almost as good in terms of

success rate and delay.

REFERENCES

[1] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, Pocket Switched Networks and Human Mobility in Conference Environments, Proc. ACM SIGCOMM Worksho p Delay-Tolerant Networking (WDTN '05), 2005.

[2] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms, Proc. IEEE INFOCOM '06, 2006.

[3] A. Vahdat and D. Becker, Epidemic Routing for Partially Connected Ad Hoc Networks, Technical Report CS-200006, Duke Univ., 2000.

[4] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, Delegation Forwarding, Proc. ACM MobiHoc '08, 2008.

[5] P. Hui, J. Crowcroft, and E. Yoneki, Bubble Rap: Social-Based Forwarding in Delay Tolerant Networks, Proc. ACM MobiHoc '08, 2008.