Give2Get: An Approach of Forwarding Data in Social Mobile Wireless Network of Selfish Individuals

Cheepuri Aditya [#1], Dr.M.V.Rama Sundari [*2], Dr.S.Maruthu Perumal [*3]

[#1] IInd M.TECH (CSE) Student, [*2] Associate Professor, [*3] Professor & HOD
Department of CSE
Godavari Institute of Engineering and Technology (GIET),
Rajahmundry, AP, India.

Abstract

In social based mobility network comprises of selfish individuals that are not willing to forward the packets but wants to forward their own messages. In this paper we present two forwarding protocols for mobile wireless networks of selfish individuals. We assume that all the nodes are selfish and show formally that both protocols are Nash Equilibria that is, no individual has an interest to deviate. Extensive simulations with real traces show that our protocols introduce an extremely small overhead in terms of delay, while the techniques we introduce to force faithful behavior have the positive side-effect to improve performance by reducing the number of message considerably (more than 20%). We test our protocols also in the presence of a natural variation of the notion of selfishness—nodes that are selfish with outsiders and faithful with people from the same community. Even in this case, our protocols are shown to be very efficient in detecting possible misbehavior.

Keywords:

Wireless networks, sensor networks, Delay tolerant networks, pocket switched networks, social mobility, selfishness, forwarding protocols

1. Introduction

Nowadays people walk around carrying all sorts of devices such as cell phones, PDAs, laptops, etc. Typically, these devices are able to communicate with each other in short distances by using communication technologies such as blue-tooth. These networks, also known as Pocket Switched Networks (**PSN** [1],[2]), can be a key technology to provide innovative services to the users without the need of any fixed infrastructure. **Pocket Switched Networks** fall in the class of the Delay Tolerant Networks (DTNs). In DTNs, messages can multi-hop from source to destination by using the forwarding opportunities given by the contacts between the nodes. These networks are usually disconnected, are characterized by social-based mobility and heterogeneous contact rate. Examples of such networks include people in working places, students in university campuses, and citizens in metropolitan areas.

Internet users can access all the network applications if there exists internet connectivity within the device. When a user needs to transmit data to other needs a reliable connection between the users .That means the other user needs to be in the wireless range or has corresponding wireless capabilities. In effect all users need end-to end connectivity for reliable data transfer. If there exists no continuous connectivity between the users the data delivery rate will decrease considerably. In reality the networking architecture is unstable and the data is transferred within a stressed environment. The network is subjected to frequent and long lasting disconnection which interrupts the data transfer and increases the error rate. So here deals with a network that is based on sociality called Pocket Switched Networking (PSN).

The problem of designing a forwarding protocol for Pocket Switched Networks has attracted

the attention of many researchers. In most cases, the protocols in the literature break down immediately if you assume that all the nodes in the network are selfish. We show this phenomenon, which is intuitive indeed, by a few experiments on **Epidemic Forwarding** [3] and **Delegation Forwarding** [4], two important protocols in the literature. Epidemic Forwarding is often used as a benchmark, since it is easy to see that, by generating as many replicas as possible in the network, it has optimal delay and success rate. Delegation Forwarding, on the other hand, is one of the best protocols in the literature with its excellent trade-off of cost, delay, and success rate.

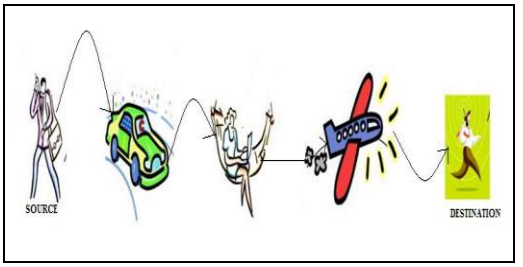

Fig. 1. Pocket switched network

In this paper, we introduce Give2Get Epidemic Forwarding and Give2Get Delegation Forwarding, which are, to the best of our knowledge, the first protocols for packet forwarding in a social setting such as the Pocket Switched Network that cope with the social aspects of the network to tolerate selfish behavior. We reach this goal by showing formally that no rational node has any incentive to deviate. In other words, our two protocols are Nash equilibria. In the paper we describe our methodology and the main steps, the mechanisms, and the idea that we have used to build the complete proof.

Lastly, we perform a large set of experiments to check performance of G2G Epidemic Forwarding and G2G Delegation Forwarding. Quite surprisingly, we discover that some of the mechanisms that we introduced 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, G2G Epidemic Forwarding and G2G Delegation Forwarding, besides providing robustness in a network where every node is selfish, have nearly the same delay and success rate of their original alter ego, and a considerably lower cost in terms of number of replicas (around 20% less).

2. Related Work

A lot of work has been done in building efficient forwarding protocols for Pocket Switched Networks. Many of these protocols use in clever and sophisticated ways the properties of human mobility [4]–[6]. All of them rely on the altruistic cooperation among nodes, which, in this setting where nodes are independent individuals cannot be given as granted. Therefore, 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 [7], [8].Previous work has been done in studying techniques of selfish mitigation for mobile ad-hoc networks.

The solutions can be classified in two main approaches: reputation based schemes [9]–[12] and credit based schemes [13]–[15]. In the former, nodes collectively detect misbehaving members and propagate declaration of misbehaving throughout the network. Eventually this propagation leads to other nodes avoiding routes through selfish members. In credit based approaches nodes pay and get paid for providing service to others. Digital cash system is implemented in order to encourage correct behavior among nodes. In [16], a combination of the two schemes is presented. All these solutions assume the use of public key cryptography for authentication of messages. Regardless of the performance of these schemes on ad-hoc networks, none of them is designed for social mobile networks.

Recently in [17] the authors introduce a barter-based cooperation system that aims to increase message delivery in opportunistic networks. The authors assume that altruistic static nodes scattered on the network area generate messages down-loadable from interested network members that pass by. 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

from the subset of the messages to which it is interested. Then the nodes start downloading one message per node at time slot, till they move out each other's communication range. The game-theoretical model developed helps the authors prove that the approach foster cooperation among the nodes. They support their findings with extensive simulations done with the restricted random waypoint model RRW model and the Simulation of Urban Mobility SUMO [18]. Though it introduces a novel technique of cooperation stimulation, their work is oriented to a gossip-like service within the network, where messages are created from special nodes and have no specific destination. Moreover, the setting is different and the solution does not consider social aspects of pocket switched networks.

In a very recent work the authors build a routing mechanism based on the willingness (declared by each individual) to forward other individuals' messages. In [19], the authors study for the first time the impact of different distributions of altruism on the throughput and delay of mobile social communication system. 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. System Model

In our system model, every node is selfish. This is a realistic scenario, if people can get the same level of service without consuming 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 (or the same) by paying less, and publish 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 will 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 easy to 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 timeouts, and the time-stamp is used when reporting misbehavior to the other network members.

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. 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 (a secure signature based on elliptic curves is just 160 bits long), and cheaper and cheaper computation, which is shown to be adequate even for sensors. Moreover, in our study we are addressing a network of smart-phones or PDAs, which are not so small devices. Modern smart-phones can run sophisticated applications, like decoders of streaming videos, 3D games, web browsers that can open SSL sessions, and others.

For these devices, a signature per message can be considered a relatively low overhead. Therefore, we assume that every node has a public key and the corresponding private key. The public key is signed by an authority that is trusted by every node in the system. Anyhow the authority is never used actively in the protocols, thus, as far as our protocols are concerned, it may remain off-line all the time. In the rest of this paper, we will use H() to denote a hash function, and $(m)_A$ to denote a message m signed by node A.

4. GIVE2GET Epidemic Forwarding

In Epidemic Forwarding [3], every contact is used as an opportunity to forward messages. If node A meets node B, and A has a message that B does not have, the message is relayed to node B. Epidemic forwarding 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 of

the network is very high. Put simply, many of the forwarding 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 would simply drop every message they receive (except those destined to themselves!). In this section, we will show how to build a version of epidemic forwarding, called Give2Get Epidemic Forwarding, that works in a system in which every node is selfish. We will see that G2G Epidemic Forwarding is Nash equilibrium, that is, no selfish node has a better choice than following the protocol truthfully. Most of the ideas and techniques that we develop in this section will be used in the more sophisticated protocols we introduce later in this paper.

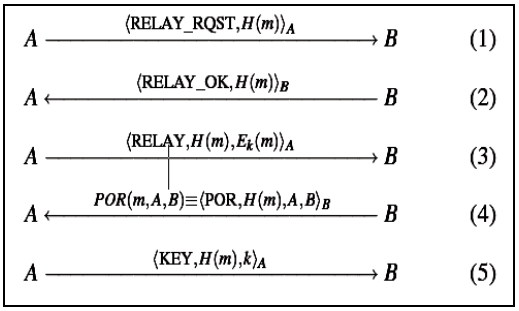

Fig. 2. Protocol of the relay phase (in case node B does not have the message).

G2G Epidemic Forwarding consists of three phases: **Message generation, relay, and test**. Message generation executes when one node has a message to send to some other node in the system. Suppose that node S has a message to send to node D. The message is built according to the following form:

$$m = \langle D, E_{PK_D}(S, msg\_id, \text{body})\rangle_S.$$

Key PKD is the public key of the destination D. Note that it is a precise design choice to hide the sender of the message to every possible relay except the destination. We will see later why it is important.

A. G2G Epidemic Forwarding

The relay phase Once the message is generated, the sender S tries to relay it to the first two (at least) nodes it meets. Assume that 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 a 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 H(m) (see Figure 2, where the role of S is described as done by node A step 1). 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 execute 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.

B. G2G Epidemic Forwarding: The test phase

Node B, once it realizes that it is a relay for message m, will follow the same protocol as done by node S—find two other nodes and relay the message to these two nodes by executing the relay phase as shown in Figure 2. By doing so, it can collect two proofs of relay that it will be asked to show, when meeting node S again, during the test phase. If node B is not able either to show the two proofs or to prove to have still the message in its memory, then node A can broadcast a proof of misbehavior (PoM) to the whole network that, in

turn, will remove node B if node B is not in the position to prove that A is wrong. The proof of misbehavior consists of the proof of relay <POR, H(m), A, $B>_B$, which is signed by node B.

Only when two proofs are collected the message can be discarded from B's memory. After a timeout Δ1, B can stop looking for relays, and after an additional timeout Δ2, node B can discard every information regarding the message. Timeout Δ1 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 Epidemic Forwarding is very close to the delay of Epidemic Forwarding, and so Δ1 can be chosen as in its original alter ago without affecting the success rate.

The difference Δ2−Δ1 bounds the time during which S can test B, and indicates how much longer B has to keep memory of the message. Thus, the shorter this difference, the better in terms of memory usage. On the other hand, timeout Δ2 should be chosen in such a way that, with non-negligible probability, nodes B meets node S again before Δ2 expires. We have to trade-off detection rate for efficiency. In our setting, here we can use the "good properties" of social networks: If S and B meet, then it is likely that they will meet again in the near future (within Δ2 in our case). Indeed, it has been shown in previous work [1], [2], [5] that in social mobile network nodes tend to form clusters of members that meet often in time. Our experiments in the following section fully support this claim. Simply by setting Δ2 = 2Δ1 the detection rate is very high (more than 90% of misbehaving nodes are detected). This result implicitly reveals that re-encounters between pairs of nodes happen soon enough with high probability. Note that during the interval (Δ1, Δ2) the nodes do not act as relays anymore. According to what happened before time Δ1 nodes keep trace of the message/PORs required in the test phase.

This might be the message itself (no relays or only one relay have been found till Δ1) or the two PORs (the message was relayed to 2 other nodes before Δ1). Note that the POR requires just the same overhead of a message signature. Thus, in the worst case, nodes keep a copy of the message for time Δ2 −Δ1 longer than in Epidemic Forwarding. On the other hand, speaking in terms of total replicas of messages generated in the network we have a gain in terms of cost. Indeed, differently from vanilla Epidemic, in G2G Epidemic nodes forward message copies to at most two other relays. While this is a design choice to make the protocol a Nash equilibrium, our experiments show that this reduces the number of replicas of some 20%.

The test phase is started by node S (see Figure 2, where, again, the role of S is described as done by node A), when meeting node B, after timeout Δ1 has expired. During the test phase, node S challenges node B: Either it has two proofs of relay, or it still stores the message. In case node B has two proofs of relay, it can replay with the two proofs. The challenge is a simple cryptographic protocol in which node S generates a random seed s and asks node B to send a keyed-Message Authentication Code HMAC on message m. The particular HMAC used in this protocol should be designed in such a way to be heavy to compute, since we want to incentive node B to relay the message and get the two proofs of relay. Since B does not know the seed beforehand, it must be storing the message unless it has found two relays. Note that it is not possible for B to fool S by forging any of the two proofs, since they are signed by the two relays. Note also that the test phase is started only by the source of the message, not by intermediate relays. This is very important to get a Nash equilibrium: only the sender has the interest of checking.As a positive side-effect, the heavy HMAC is virtually never executed if no node deviates from the protocol—it is extremely unlikely that the first two relays are not able to find two other nodes that have never seen the message.

5. GIVE2GET Delegation Forwarding

Delegation Forwarding [4] is a class of protocols that have been shown to perform very well. In Delegation Forwarding, every node is associated with a forwarding 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 forwarding quality of the message. If this is case, node A creates a replica of the message, label 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 a 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 G2G Epidemic Forwarding in order to get a version of Delegation Forwarding that is Nash equilibrium.

Simply speaking, the techniques we developed to build G2G Epidemic Forwarding can prevent message dropping by those who take the message. However, selfish nodes have many other rational ways to deviate in these more sophisticated protocols. First, nodes can lie on their forwarding quality. They can claim that their quality is zero, and nobody can do much about this, these nodes would 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 G2G 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 following the protocol truthfully. In this way, we protect our system against liars, cheaters, and any other possible way to deviate rationally. In this paper, we consider Delegation Destination Frequency and Delegation Destination Last Contact [4].

Delegation Destination Frequency Node A forwards message m to node B if node B has contacted m^1 s destination more frequently than any other node that the copy of the message m carried by A has seen so far.

Delegation Destination Last Contact Node A forwards message m to node B if node B has contacted m^1 s destination more recently than any other node that the copy of the message m carried by A has seen so far.

A. G2G Delegation Forwarding: The relay phase and the test by the destination phase

Figure 3 shows the protocol of the relay phase. Just like G2G Epidemic Forwarding, node A has an interest to start this phase, since it has to collect the proof of relay for the message. In step 8, node A asks B what is its forwarding quality to D^1. 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^1 is the actual destination D, when the destination of m is B, D^1 is chosen as a random node different from B. This mechanism has the goal of making it impossible to B to know whether it is the destination of the message or not before taking the message and giving the proof of relay. Therefore, just like in G2G Epidemic Forwarding, node B will follow all the relay protocol with the hope of being the actual destination of the message. Note that in G2G Delegation Forwarding the proof of relay contains much more information, including the forwarding quality towards D claimed by node B and the forwarding quality of the message at that point in time.

Note that, in our setting, we don't really need to introduce mechanisms to make this proof checkable by the authority, or by other network members: Node D has no interest in lying. However, simple techniques can be introduced to make it impossible for D to remove faithful nodes. For example, in case of Delegation Destination Last Contact, if the nodes exchange a signed message

(with a time-stamp, as usual) at every contact, this message would be a proof of misbehaving against B. Similar techniques can be introduce for Delegation Destination Frequency.

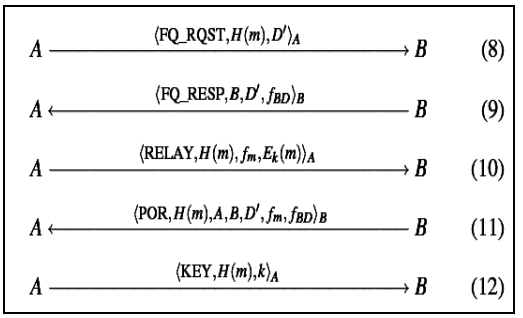

Fig. 3. G2G Delegation Forwarding: Protocol of the relay phase.

B. G2G Delegation Forwarding

The test by the sender is executed only by the sender of the message. Assume that node A is not the sender, and that it has received the message from the sender S. When A gets in contact with S again, after timeout Δ1 (defined as in G2G Epidemic Forwarding), node A is tested and, just like in G2G Epidemic Forwarding, it gives the two proofs <POR, H(m), A, B, D, f_m^1 , f_{BD} >B and <POR, H(m), A, C, D, f_m^2 , f c_D >B to node 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 A is not a cheater, that is it has not reduced fm to get rid of the message quickly. Indeed, S can check whether

$$f_{AD} = f_m^1 < f_{BD} = f_m^2 < f_{CD}.$$

6. Conclusion

In this paper we have presented G2G Epidemic Forwarding and G2G 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 G2G protocols are Nash equilibria. Quite surprisingly, G2G protocols also outperform their alter egos in terms of cost, while being almost as good in terms of success rate and delay.

7. References

[1] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference envi- ronments," in WDTN '05: Proc. of the ACM SIGCOMM workshop on Delay-tolerant networking. ACM Press, 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," in INFOCOM '06. Proc. of the 25th IEEE International Conference on Computer Communications, 2006.

[3] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep. CS-200006, 2000.

[4] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in MobiHoc '08: Proc. of the 9th ACM international symposium on Mobile ad hoc networking and computing. ACM, 2008.

[5] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in MobiHoc '08: Proc. of the 9th ACM international symposium on Mobile ad hoc networking and computing. ACM, 2008.

[6] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in MobiHoc '07: Proc. of the 8th ACM international symposium on Mobile ad hoc networking and computing. ACM, 2007.

[7] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J.-P. Martin, and C. Porth, "Bar fault tolerance for cooperative services," in In SOSP05 20th ACM Symposium on Operating Systems Principles. ACM, 2005.

[8] H. C. Li, , A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin, "Bar gossip," in Proc. of the 7th Symposium on Operating System Design and Implementation (OSDI '06), 2006.

[9] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in MobiCom '00: Proc. of the 6th annual international conference on Mobile computing and networking. New York, NY, USA: ACM, 2000.

[10] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol: Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks," in MobiHoc 02: Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing, June 2002.

[11] K. Balakrishnan, J. Deng, and V. Varshney, "Twoack: preventing selfishness in mobile ad hoc networks," in Wir. Comm. and Net. Conf., 2005 IEEE, vol. 4, March 2005.

[12] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security. Kluwer, B.V., 2002.

[13] L. Butty´an and J.-P. Hubaux, "Enforcing service availability in mobile ad-hoc wans," in MobiHoc '00: Proc. of the 1st ACM international symposium on Mobile ad hoc networking & computing. IEEE Press, 2000.

[14] J.-P. Hubaux, T. Gross, J.-Y. LeBoudec, and M. Vetterli, "Towards self-organized mobile ad hoc networks: The terminodes project," IEEE Communications Magazine, vol. 39, no. 1, 2001.

[15] M. Jakobsson, J.-P. Hubaux, and L. Butty´an, "A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks," in Proceedings of Financial Crypto 2003, January 2003.

[16] H. Miranda and L. Rodrigues, "Preventing selfishness in open mobile ad hoc networks," in Proc. of the Seventh CaberNet Radicals Workshop, October 2002.

[17] L. Butty´an, L. D´ora, M. F´elegyh´azi, and I. Vajda, "Barter trade improves message delivery in opportunistic networks," Ad Hoc Networks, vol. 8, no. 1, pp. 1–14, 2010.

[18] "SUMO-Simulation of Urban MObility," http://sumo.sourceforge.net/.

[19] P. Hui, K. Xu, V. Li, J. Crowcroft, V. Latora, and P. Lio, "Selshness, altruism and message spreading in mobile social networks," in Proc. Of First IEEE International Workshop on Network Science For Communication Networks (NetSciCom09), April 2009.

8. About the Authors

Cheepuri Aditya is currently pursuing his M.Tech (CSE) in Computer Science & Engineering Department, GIET, Rajahmundry. His area of interests includes Networks.

Dr.M.V.Rama Sundari is currently working as an Associate Professor in Department of IT, GIET, Rajahmundry. She was awarded with PhD in related field. Her research interests include Communications Networks.

Dr. S. Maruthu Perumal is currently working as a Head of the Department for Computer Science & Engineering Department, GIET, Rajahmundry. He is awarded with PhD in related field. His research interests include Image Processing, Data Mining & Warehousing, Networks and Security, Software Engineering.