Coupons: A Multilevel Incentive Scheme for Information Dissemination in Mobile Networks

Anargyros Garyfalos, *Student Member*, *IEEE*, and Kevin C. Almeroth, *Senior Member*, *IEEE*

Abstract—Integrating mobile computing and localized user interaction into the Internet requires more than simply overcoming pure routing challenges. Apart from issues such as intermittent connectivity and unreliable nodes, the human factor is becoming increasingly important. To date, user behavior has only been considered in terms of mobility patterns and incentives for encouraging cooperation. Moreover, the focus has always been on systems that aim to enforce fairness by either bringing an indirect equilibrium or by punishing "misbehaving" users. The goal of this paper is therefore to present `Coupons`. `Coupons` is an idea to *define*, *evaluate*, and *implement* algorithms for the emerging class of social network and information sharing applications. `Coupons` is a system designed to allow users to opportunistically interact and share data through a wireless medium, to overcome highly variable Internet access through adaptive localized interaction, and to incorporate an incentive scheme to make the system usable and encourage participation.

Index Terms—Coupons, wireless, incentive.

1 Introduction

We are evolving into a society with nearly constant access to the Internet and its vast wealth of information. In order to achieve this level of connectivity, evolution is taking place in two directions: evolution of devices and evolution of network access. With novel devices like Personal Digital Assistants (PDAs) and advanced cell phones, users can now operate through a range of options. On the networking side, the reach of the infrastructure is expanding along with the number and variety of data protocols. To this end, the focus is no longer limited to the choice between different access technologies but about being able to share content, either through the Internet or through direct interaction with different users. This is an evolution of true peer-to-peer applications toward the formation of social networking communities in wireless environments.

The key driver of this evolution is the desire for *data sharing* through *opportunistic contact*. Opportunistic contact is the ability to interact and communicate with people who are near the same geographical location. This proximity, and the specific circumstances under which it occurs, introduces a new range of application scenarios: plane passengers may interact through the filtering of specific profiles. Moreover, tourists may exchange information about places of interest or local restaurants. In these cases, content is no longer stored in a well-known source that can be periodically accessed. For this reason, communication is not targeted to a specific destination but to a *potential* number of nearby users. Although, in a certain sense, peer-to-peer has already introduced this model, we extend the concept by transferring it to the wireless domain and adding opportunistic contact.

Deploying such applications in a mobile environment creates a number of challenges. First, data must be shared in an *efficient* and *effective* manner. Users may move among access points and be directly connected, or they may move to a location from which they can only connect using other nodes as intermediaries. The network must thus maintain connectivity as best as possible even though nodes are moving and disconnecting frequently. While there are numerous solutions to many of the associated problems, the overall environment creates the second challenge. That is, nodes are expected to participate in relaying even though they have no incentive. Incentives will become increasingly important, as it may be the only way to share content in places with intermittent connectivity.

The goal of this paper is thus to investigate this evolving area of research. Our idea, called `Coupons`, is a first attempt to *define*, *evaluate*, and *implement* a set of algorithms, whose goals are threefold: 1) to allow users to opportunistically interact and share data through a wireless medium, 2) to overcome highly variable Internet access through adaptive localized interaction, and 3) to incorporate an incentive scheme to make the system usable and encourage participation.

We call our solution `Coupons` because one possible application is the dissemination of a *real* coupon. Coupons represent small discounts commonly used to attract shoppers to a particular store. In our scheme, coupons are shared as nodes come into contact with each other. Every node that participates in the forwarding adds a new *level* into the coupon, thus creating a *multilevel* incentive. If a coupon is eventually used, all users who participated in forwarding the coupon are given some form of credit. We emphasize, however, that the concept of `Coupons` can be applied to a whole range of new applications.

- *A. Garyfalos is with the Computing Department, Faculty of Science and Technology, InfoLab21, Lancaster University, LA1 4WA Lancaster, UK. E-mail: anargyros.x.garyfalos@jpmorgan.com.*
- *K.C. Almeroth is with the Department of Computer Science, University of California, Santa Barbara, Santa Barbara, CA 93106-5110. E-mail: almeroth@cs.ucsb.edu.*

Manuscript received 31 Jan. 2007; revised 19 Oct. 2007; accepted 6 Feb. 2008; published online 27 Feb. 2008.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0037-0107.
Digital Object Identifier no. 10.1109/TMC.2008.37.

In our previous work, we offered a limited version of the Coupons idea and performed a basic evaluation [8], [7]. The contribution of this paper, therefore, is to offer a thorough evaluation of the Coupons idea through three steps. First, we *define* a set of adaptive periodic broadcast algorithms. Second, we fully *evaluate* their performance, and finally, we *implement* the best of them as part of a protocol. One result from our research is the recognition of a particular new area of research. We believe Coupons is the basic platform on which a whole new range of peer-to-peer localized-sharing applications may be defined.

The remainder of this paper is organized as follows: Section 2 presents our Coupons scheme in more detail and gives an overview of related work. Section 3 provides a formal analysis of the system and defines the broadcast mechanisms. These mechanisms are then evaluated in a two-step process. Section 4 offers the initial evaluation with the goal of describing the baseline performance of the system through a set of simulations. Section 5 extends the evaluation model by considering malicious behavior and a testbed implementation. The paper is concluded in Section 6.

2 System Description

Coupons is a novel application with a specific aim: to exploit opportunistic contact in order to allow the sharing of content between as many users as possible *and* as efficiently as possible. Since we consider a network of mobile nodes, there are two main challenges. First, the nature of opportunistic contact does not only imply a constantly changing network topology, it also means that nodes should be able to detect these changes as quickly as possible in order to incorporate new users more efficiently. Second, nodes need to be encouraged to relay information to others through some sort of incentive. A solution to these challenges must also consider the need for users to exchange information even when not connected to the Internet.

The basic concept behind Coupons is to distribute a given piece of information through a mobile network. The characteristics of the mobile network can be very flexible as the underlying MAC protocol can include 3G, 802.11, or 802.16. The main issue is that the information is distributed using a controlled broadcast. This is related to the notion of *epidemic routing* in ad hoc networks [28], as information is passed between islands of nodes that come into transmission range. However, there are two notable differences. First, epidemic protocols attempt to *flood* a network with a piece of information. In Coupons, the aim is to discover a *set* of users that may be interested in a certain type of content. The difference is that the broadcast of the information should continue as a user moves, as this may be the only way to discover new users and pass data to them. Being able to adapt to various networking conditions is thus a key aspect in Coupons. The second difference is that common epidemic networks typically assume global participation *and* common behavior from all nodes. In Coupons, nodes are now independent users that must be encouraged to relay the content to others using a specified incentive. The incentive we propose is based on an ordered list of unique IDs appended to a message. The idea is that, once the information/coupon is used, users contained in the ordered list receive some sort of benefit. The benefit can be uniform for all users on the list or can be variable, e.g., the higher a user is on the list the more the reward value.

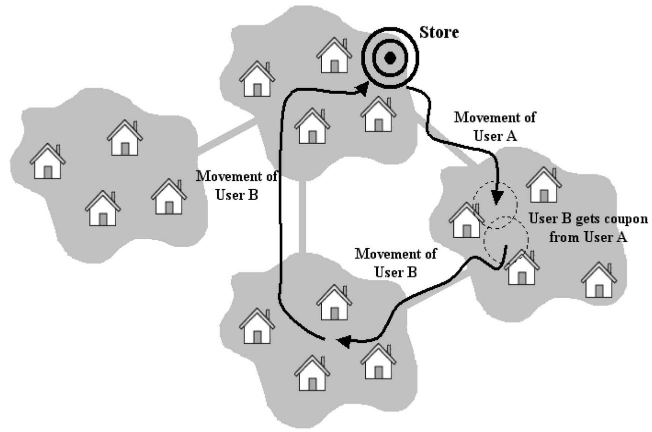


Fig. 1. Coupons operation.

As an example scenario, we could envision a true coupon where the information distributor is a store trying to advertise its products. A common mechanism for doing this is through discounts for merchandise. Normally, users would have little incentive to relay this kind of information, but if they could attach their ID to the coupon and receive benefit if someone used the coupon with their ID in the list, there would be an incentive.

Fig. 1 shows a typical scenario with a store that is interested in advertising a coupon. The different clusters of houses may be urban areas or even small towns. A store will periodically broadcast the coupon, and all nodes within the broadcast range will receive it. Some may decide to keep it and others to discard it. Some nodes may then also decide to rebroadcast it. Assuming User A is among those that keep the coupon and want to rebroadcast it, User A will start forwarding the coupon through periodic transmissions. Then, as User A moves throughout the neighbor cluster, the coupon is forwarded to local users who come into contact with User A such as User B. In the same manner, User B may decide to forward the coupon to other users in the same cluster and, possibly, other users in other clusters depending on where User B moves. If a user chooses to use a coupon, points will be awarded to all users in the ID list. For example, if User B goes to the store, User A would earn a discount.

Based, on the example in Fig. 1, we recognize that there are three fundamental elements or *roles* in the system:

- The **Application Provider** wants to spread the coupon to as many users as possible as fast as possible. Providers who use coupons will not particularly care how many network resources are consumed, they simply want their information to reach as far and wide as quickly as possible.
- The **Network Service Provider** desires to see the least possible number of redundant transmissions. This is particularly important in a system in which

many coupons are being advertised by many applications and users.
- The **Users** aim for a combination of the application and network provider objectives. Users may want to receive coupons but will not want to see their own resources consumed or the network congested unless they benefit directly.

Based on these roles, the range of applications could be further extended in two different dimensions. First, the incentive may have a more "indirect" nature. This is applicable in cases where the information is an advertisement for products or services for which a discount might be inappropriate, instead, free airline miles or other kinds of points can be awarded. Second, Coupons may even be applied in the area of public services. The information to be passed, for example, important police information, might use different incentives. In this case, the operation would have to be modified by either having an incentive such as free parking credits or even by forcing users to automatically forward urgent pieces of information in emergencies.

Overall, the motivation to develop Coupons is that it presents a platform on which a plethora of novel applications may be implemented. However, as Coupons is associated with wide-scale information dissemination, there are a number of key problems regarding the *efficiency* of the scheme. Despite the fact that existing basic flooding schemes offer ultimate simplicity and are effective in terms of spreading the information to a wide range of participants, Coupons demands the inclusion of mechanisms that provide participation incentives, security, and efficiency.

First, there is little incentive for nodes in mobile networks to forward information. Despite recent work to develop incentive schemes, both for ad hoc and peer-to-peer networks [19], [20], Coupons have an additional important distinction. It targets specific lightweight applications with a focus on minimizing complexity. Second, lightweight security mechanisms specifically and security in mobile networks more broadly are still open areas of research [25], [33]. In Coupons, we have developed a simple but effective security mechanism [8]. Third, in terms of efficiency, simple flooding is usually associated with *broadcast storms* [24]: a high rate of redundant messages, the increased probability of collisions, and increased congestion. Existing work has provided a number of different solutions using a broad range of heuristics, ranging from simple to very complex [24], [30], [22], [31], [23], [14], [2], [11], [3], [1]. Advanced schemes have been shown to be more efficient, but they are associated with increased deployment costs or challenging architectural requirements. On the other hand, the simpler schemes are known to suffer from relative inefficiency and lack of adaptability to highly variable network conditions. The remainder of this paper shows that Coupons avoids these concerns and, yet, maintains simple and uncoordinated deployment characteristics.

Finally, the idea of couponlike schemes has been mentioned in two places. Both iCLouds [10] and eNcentive [18] propose a similar coupon scheme with a coupon being an ID list, where nodes append their unique signatures. The authors of both papers give good motivation for using coupons, and they present a middleware architecture to enable this facility. In particular, iClouds focuses on the configuration of an application in order to support automatic filtering of incoming coupons. The actual transmission of a coupon is then based on a *pull model* through a three-way handshake; effectively, users query passing nodes for the desired piece of information. The work on this paper differs from iClouds in three ways. First, we offer a broader discussion on the nature of the incentive scheme and the impact of different award strategies. Second, we examine the impact of a more aggressive approach, where nodes attempt to "push" coupons to as many users as possible. This is a significant difference as it raises concerns over network performance. Investigating the extent of these concerns and quantifying the performance of Coupons is the third distinction of our work.

More generally, our work is related to two broad areas of research. The first is communication in disconnected ad hoc networks [21] and Delay Tolerant Networking (DTNs) [13]. Our work, while related, is fundamentally different in its assumptions and objectives. First, research in ad hoc networks almost always assumes an end-to-end path [5]. When ad hoc networking research relaxes this assumption, it looks more like the store-and-forward architectures in the DTN area. In both of these cases, *the intended recipient is always known*, and the focus is on how to reach that destination (or set of destinations) across a disconnected end-to-end path. In Coupons, communication occurs opportunistically between nodes that come into contact of each other. As such, these areas of related research offer context into the problem on which we focus but are not comparable quantitatively.

3 System Analysis

The key component of Coupons is to find when and how often a node should transmit. We call the algorithm that makes this decision the *broadcast frequency algorithm*. The main goal is then to find a solution that adapts to different environment conditions. In this section, we identify a set of such algorithms through a two-step process. First, we analyze system characteristics to identify the main parameters that affect performance. Then, we suggest a set of broadcast frequency algorithms that exploit these parameters.

3.1 System Characteristics

The *ideal* Coupons transmission scheme would be to spread the coupon as widely as possible and as quickly as possible but would keep the number of redundant transmissions to a minimum. Although these three properties seem to conflict, the specific nature of Coupons does not necessarily require trading one objective for another. Based on the dynamics of a mobile network, efficient asynchronous communication can be achieved without constant broadcasting but, instead, can be accomplished by intelligently pushing information on "appropriate occasions." An occasion is defined as "appropriate" based on two important parameters: node density and system saturation.

Node density is a measure of how many nodes are within the communication range of each node at any point

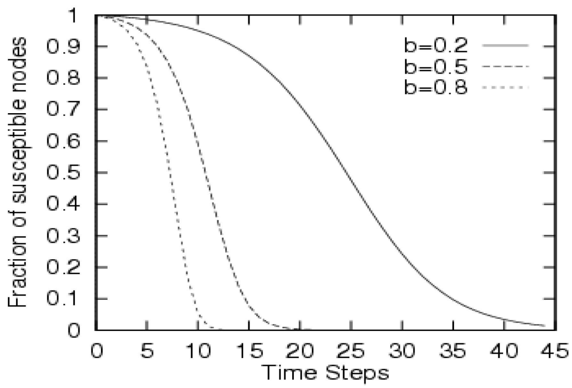


Fig. 2. Fraction of susceptible nodes over different node degrees.

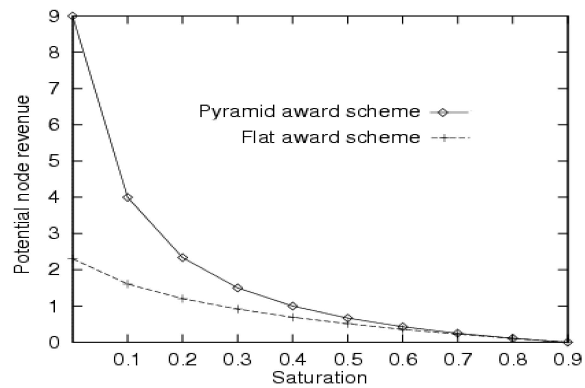


Fig. 3. Node incentive over saturation for different award schemes.

in time. Using our "epidemic routing" analogy again, node density is essentially a *node contact rate*. This rate has been traditionally approximated through the Susceptible, Infected, and Recovered (SIR) mathematical model [12]. SIR is a class of epidemiological models that compute the number of people *infected* with a contagious disease, $i(t)$, over those who are *susceptible* to infection, $s(t)$, at time t. In one of the simplest SIR models (Kermack-McKendrick), for a completely homogeneous and fixed population (i.e., no births or deaths), there are two coupled nonlinear differential equations:

$$\frac{ds}{dt} = -b * s(t) * i(t). \qquad (1)$$

$$\frac{di}{dt} = b * s(t) * i(t). \qquad (2)$$

These formulas show that the time-rate of change of susceptible nodes depends on the number already susceptible, the number of nodes already infected, and b. This is the rate of contact between susceptible and infected nodes. Fig. 2 shows a theoretical evaluation of these formulas and demonstrates how the fraction of susceptible nodes decreases for different values of b. The x-axis is an abstract reflection of time, whereas the y-axis shows the ratio of susceptible nodes over the complete population. Overall, Fig. 2 shows that the spreading efficiency of Coupons is expected to be heavily dependent on the population density of the underlying environment. This hypothesis is of particular importance since it implies that an efficient transmission mechanism should be able to adjust to environments with different node densities.

System saturation is defined as the percentage of interested users that have already received the coupon. Sensing the presence of uninfected nodes so that they can be infected is particularly important in Coupons. Nodes would like to use their resources wisely and, so, only try to broadcast if there are nearby uninfected nodes. In Coupons, *award effectiveness* by a node is achieved by being one of the very first nodes to be infected. This is fundamentally different than standard epidemic schemes, where the aim is to simply spread the information as efficiently as possible.

In order to understand the importance of this feature, we first model the potential incentive of each node. To this end, we modify a proposed model for monetary flow of multilevel marketing schemes [26] to define the incentive of a node participating in Coupons. This modified model is now

$$v_i(s) = \int_s^1 \frac{\pi(s)}{s} ds, \qquad (3)$$

where s is the saturation of the system $(0 \leq s \leq 1)$, and π_s is the number of points to be awarded by the store. The formula is evaluated by assuming that a node passes the coupon to all remaining susceptible nodes given that saturation is s.

Fig. 3 is a log graph showing the incentive awarded as "potential node revenue" (y-axis) over the saturation level (x-axis). The two lines represent two different award schemes, one being a typical *pyramid* award scheme and the other a *flat* scheme. In the first case, the store gives higher rewards to users whose IDs are higher in the list. Then, assuming an unlimited number of coupon levels, as the saturation increases, so do the chances of receiving a coupon, where the first slots have already been taken. Therefore, we use a linearly decreasing function such as $\pi(s) = \frac{1}{s}$ and thus obtain $v_i = \frac{1}{s} - 1$. In the *flat* scheme, all users receive equal points irrespective of their position in the ID list. If we model $\pi(s) = 1$, we obtain $v_i = -ln(s)$. Overall, Fig. 3 shows that the nodes that get the coupon early have a much better chance of receiving more points. While this is not a *fair* situation, our goal is not to create an equal opportunity for all users to benefit but rather to distribute the information as quickly and efficiently as possible.

We have thus presented the relevance of the two most important system parameters. In terms of node degree, retransmission should be moderate in densely populated environments and more intense in sparse environments. In terms of saturation, nodes should aim to be among the very first that introduce the coupon to a new area. The potential gain of a transmission becomes much lower as more users see the coupon. An efficient broadcast frequency algorithm should not only adhere to these characteristics but should also be adaptive as conditions change. In Section 3.2, we proceed to the second stage of our analysis and define a set of such adaptive algorithms.

3.2 Broadcast Frequency Algorithms

Although both node degree and system saturation could be derived through some sophisticated protocol and then used to calculate a rate of coupon broadcast, the implementation of such a scheme would introduce significant complexity and be counter to our goal of simplicity. For this reason, an implementation of an efficient `Coupon` scheme can only rely on two possible sources of information: monitoring received coupons over a specific time interval and collecting feedback information through acknowledgment messages.

Starting with the simplest of the two information sources, counting the rate of received coupons is a good estimate of a system's current saturation. The more coupons that are heard within a short time interval, the higher the probability that many nodes are already infected. We take this result as a strong suggestion for a node to limit its own broadcasts. However, not all cases are covered since the absence of transmissions does not necessarily imply that the system has not been saturated. It may well be the case where all nodes have the coupon and are no longer interested in forwarding it (e.g., when maximum coupon levels have been reached). Alternatively, it may be that no nodes are interested in the coupon at all. The distinction between such cases is fundamental.

By introducing the use of acknowledgments, a node could better understand both the level of saturation and the interest of surrounding nodes. We note that, in `Coupons`, we use the notion of an acknowledgment message in a fundamentally different way when compared to standard protocol practice. An acknowledgment not only denotes correct reception of a broadcast but also, more importantly, identifies a receiving end user as being interested in the coupon. Such interest may be specified either manually or through preconfigured user profiles. As an alternative, we could use a three-way handshake. First, the source broadcasts an advertisement for a coupon, instead of the coupon itself. Then, interested users would respond with a positive acknowledgment that would trigger transmission of the coupon itself. The trade-off is between the overhead of just transmitting the coupon (as compared to a short announcement message) versus the overhead of a three-way communication (as compared to the exchange of only two messages).

Based on the two sources of information, we now define three basic retransmission mechanisms. The first one is a static approach where the behavior remains stable based on a simple probability function, i.e., it uses no available information. The other two attempt to follow a more adaptive methodology based on, first, the level of traffic and, second, the reception of acknowledgment messages. The details of each of the three schemes are outlined below:

- **Probability-based scheme.** In this broadcast frequency algorithm, the behavior of nodes remains static throughout the duration of operation. It is similar to blind flooding apart from one important difference: each node retransmits based on a *predetermined* probability. Previous studies have shown the potential of this scheme but with an important assumption that the selected probability suits the density of the population [27]. This assumption works because, for dense environments, a low probability can achieve satisfactory spreading efficiency while saving network resources. In sparse environments, however, the probability must be higher to handle the reduced contact rate. We evaluate two different rates, a low (10 percent) rate and a relatively high (25 percent) rate.
- **Traffic-based scheme.** This broadcast frequency algorithm is the first attempt to propose a scheme that adapts the broadcast rate based on the density and saturation of the system. After a node receives a coupon for the first time, it attempts to retransmit at a constant rate. However, the reception of duplicate coupons broadcast by other nodes results in a sleep period during which no broadcasts are made. As soon as the node wakes up, it starts to retransmit as normal. Any reception of a coupon during the sleep period resets the sleep timer.
- **ACK-based scheme.** The final scheme is an effort to extend the capability of the broadcast frequency algorithm to detect the surrounding saturation level. For this reason, we extend the functionality of the *Traffic-Based* scheme by introducing the use of acknowledgments. Each node now enters a sleep period, not only when a duplicate coupon is received, but also through the absence of an acknowledgment message over a specified period. The period may either be implemented as a timer or through a simple transmission counter. Each broadcast increases the counter by one. If the counter reaches a specified threshold and an acknowledgment has not been received, the node sleeps. If at any time during the nonsleep period an acknowledgment is received, the counter is reset, and operation proceeds as usual.

The remainder of this paper examines the performance of the three algorithms in a two-step process; first, we offer a baseline evaluation through a set of simulations. This set of results is followed by simulations that model specific, highly dynamic behavior, as well as an implementation and its evaluation in a testbed.

4 Simulation-Based Evaluation

In this section, we perform the first step of our evaluation process. Our goal is to understand the performance of `Coupons` in order to highlight both the potential and any shortcomings of our broadcast frequency algorithms. To this end, we first explain the details of the simulation model. We then describe the first set of tests using a basic set of parameters. We conclude this section by introducing an extended set of parameters such as different participation levels and the impact of varying transmission environments.

4.1 Simulation Environment

We have performed our simulations using the GloMoSim simulator [32]. We chose to use GloMoSim because, for our evaluation needs, we only needed to focus on the exchange of messages at the application layer, functionality is accurately supported in GloMoSim. Our coupon schemes were then implemented on top of a simple flooding protocol where nodes broadcast the coupon to their immediate surrounding neighbors. These broadcasts were set to occur at large time intervals of 30 seconds. The reason for such long delays was to emulate a real life scenario where batter consumption is a primary issue. Initially, the coupon is transmitted by a single stationary node that takes

TABLE 1
Simulation Parameters

Parameter	Value Range	Nominal Value
Duration	4 hrs	4 hrs
Number of nodes	4...400	100
Node degree	0.2...9	0.65, 3, 9
Mobility model	Random Waypoint, ManhattanGrid, Cluster-Based	Random Waypoint
Coupon levels	1...10	5
Node behavior	Probability-Based, Traffic-Based, ACK-Based	N/A

the role of the "store." The details of our simulation parameters are described below,[1] along with a summary in Table 1:

- **Duration.** This is length of the simulation.
- **Number of nodes.** This is number of nodes in the network.
- **Node degree**. We use node degree as an abstraction to represent average node density. Node degree is determined by a combination of three parameters: number of nodes, radio range, and grid size.
- **Mobility model**. We run our simulations using a number of models. However, the results were essentially the same, therefore, in this paper, we only report the results for the *Random Waypoint* model. Different models gave slightly different results, but this was mainly due to the different average node density that was produced from these models.
- **Coupon levels.** The maximum number of ID signatures each coupon can carry. We assume that a node will broadcast a received coupon only if there is at least one free slot in the ID list.

Based on these parameters, we implemented the three broadcast frequency algorithms as follows:

- **Probability-based scheme.** Nodes retransmit with a random delay between 20 and 40 seconds. As we assume that the probability remains unchanged for the duration of the experiments, we used a range of values. The sections below show the results of two representative examples for both a low (10 percent) and a higher (25 percent) probability. These values have been selected because they show the best results for dense and sparse environments, respectively. In practice, as the results show, an incorrect retransmission probability can be catastrophic and either cause congestion collapse or result in almost no nodes receiving the coupon.
- **Traffic-based scheme.** When not in a sleep state, nodes transmit with a random delay between 20 and 40 seconds. The sleep period is randomly determined between 4 and 6 minutes. Randomness in the sleep period is used to avoid node synchronization and reduce the risk of starvation.
- **ACK-based scheme.** Nodes transmit up to a certain number of times, denoted by a simple counter. If no ACK messages are received, nodes enter a sleep period similar to the *Traffic-Based* scheme. If, however, an ACK message is received, the nodes reinitialize the counter and start to transmit again. In our simulations, the counter limit has been set to 10 messages.

Based on this setup, we performed our simulations according to the following metrics:

- **Network efficiency.** Measures the total number of times the same coupon is delivered to nodes. This metric is preferred to simply counting the number of broadcasts, since the latter may give misleading results. For example, if we assume that a node transmits x times, then it should make a difference if the node is in a deserted area, where it causes no interference to others, or in the middle of a city. Measuring the number of duplicate receptions distinguishes these two cases.
- **Spreading efficiency.** Measures the percentage of nodes that receive the coupon at least once during the simulation period. We note that if we were to run a simulation indefinitely, 100 percent spreading would eventually be achieved. For this reason, we also consider how quickly nodes receive the coupon the first time. The results presented are for the cumulative percentage of nodes that have received the coupon and so show both values.

4.2 Basic Environment Evaluation

In our attempt to describe the main characteristics of our broadcast frequency algorithms, we first explain their behavior in an environment that uses the nominal values given in Table 1. During our evaluation, we further considered a broad range of node populations, from 20 to 400. However, as the results intuitively follow the same principles, for the sake of clarity, we only present the results corresponding to a nominal value of 100 nodes. Finally, we evaluated two variations of the *Probability-Based* scheme. We used both a low (10 percent) and a relatively high (25 percent) probability in order to evaluate the importance of choosing the appropriate value.

Network efficiency is expressed as the number of coupons received by all 100 nodes over the duration of the experiments (4 hours). Fig. 4 shows the representative results for a range of node degrees with the x-axis counting 10-minute intervals. From our results, we make two important observations. First, we verified our analysis that showed the significance of node degree. Using the *Probability-Based (25 percent)* scheme as an example, we can see how the load increases from one received coupon per 10 minute period, shown in Fig. 4a, to almost six coupons per 10 minute period in Fig. 4c.

The second observation refers to the main characteristics of the algorithms. The *Traffic-Based* scheme is associated with the highest load, especially for high node degrees. For the *Probability-Based* scheme, we observe the importance of operating the system with the appropriate transmit probability. Going from 10 percent to 25 percent increases the

1. We tested numerous parameters across wide ranges of values, only to discover that many of the traditional parameters had a minimal impact on our system and metrics. For reasons of space and to avoid redundant and duplicative results, we include only the most illuminative results.

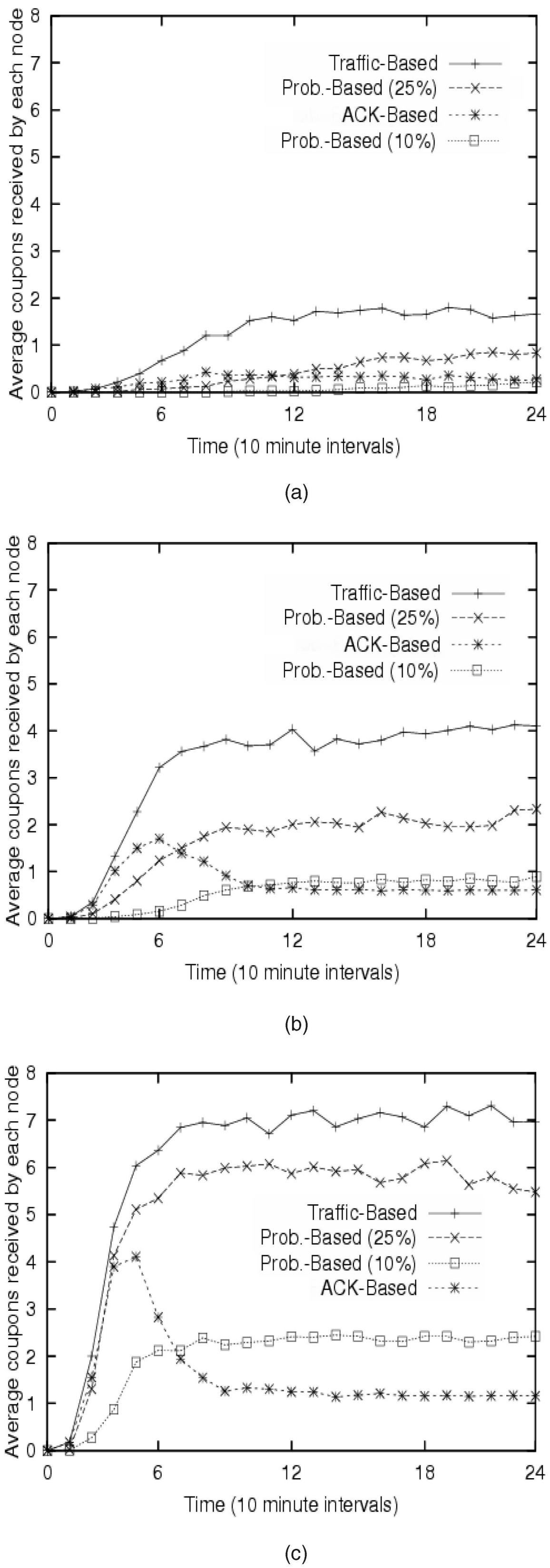


Fig. 4. Network efficiency. (a) $\text{Node Degree} = 0.65$. (b) $\text{Node Degree} = 3$. (c) $\text{Node Degree} = 9$.

load from two to more than six messages received in every 10 minute interval $(\text{node degree} = 9)$.

In general, suing a nonadaptive scheme like the *Probability-Based* scheme will result in satisfactory results in only a small set of homogeneous environments. The scheme simply lacks the capability to dynamically adapt to different environmental conditions. This adaptive capability, however, is evident in the *ACK-Based* scheme. Despite the short peak during the first stages of the experiments, the *ACK-Based* scheme offers the lightest load, since, even for highly dense environments, the load stabilizes at just over one received message every 10 minutes. The drop from the initial peak is caused by the saturation of the system and is detected through ACK messages.

Spreading efficiency measures the percentage of nodes that have received the coupon over a simulation period of 4 hours. Fig. 5 gives the representative results for the same three node degree values. For example, a value of 90 percent at Point 12 indicates that 2 hours into the simulation, the specific scheme had spread the coupon to 90 percent of the total node population.

The results in Fig. 5 confirm two major points. First, we verify the importance of the node degree to the spreading rate. The second point relates to using static probabilities in the *Probability-Based* scheme. Although the results indicate that network efficiency was good some of the time, the system generally underperforms in sparse environments and transmits too many coupons in dense environments.

The *ACK-Based* and *Traffic-Based* schemes demonstrated effective adaptation in all of our simulations. The *Traffic-Based* scheme generally performed better in sparse environments. This is because the counter may have been too low to account that the *ACK-Based* scheme was not responsive enough for the few interested listeners. Another important observation was that for high-density populations, increased traffic caused ACK messages to be lost. This behavior was compensated, however, by the fact that, in such environments, the first few broadcasts were effective in quickly spreading the coupon.

4.3 Extended Environment Evaluation

So far, we have performed our evaluation under two basic assumptions. First, we assumed that once a node receives a coupon, it immediately starts broadcasting. Second, all nodes can move across the entire grid without limitation. However, this is not always a realistic model. To this end, we now introduce extended parameters to address both limitations.

For the first assumption, either through manual operation or through other more advanced methods, users will be able to specify profiles stating their desired behavior. For instance, a profile might dictate not to retransmit if the node's battery is low. Alternatively, a user might be in the middle of a call and prefer to not use the bandwidth for coupons. More importantly, the user may specify that he/she is not interested in certain types of coupon or feels the incentive is not justified based on the content of the coupon.

In order to accommodate these aspects, we extended the simulation model of *susceptible* and *infected* nodes by introducing the *active* state. In the new model, once a user receives the coupon for the first time, it becomes *infected*. The difference now, however, is that retransmission takes

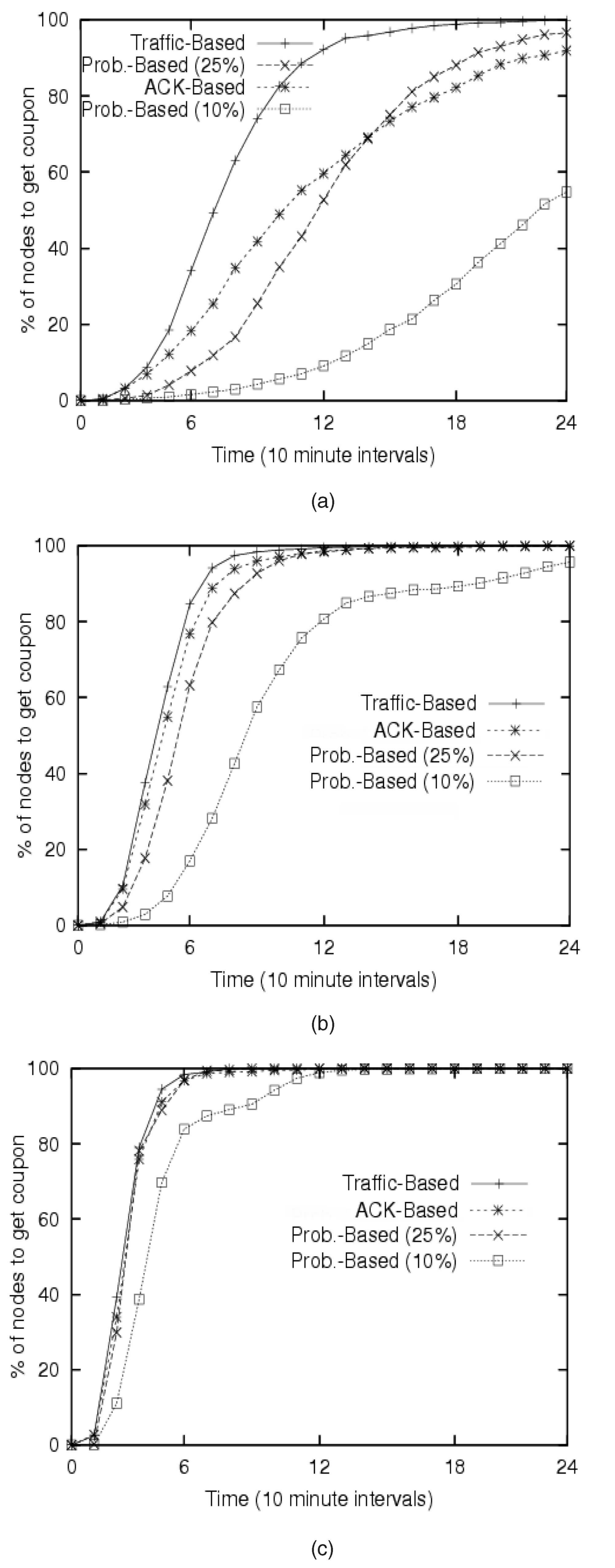


Fig. 5. Spreading efficiency. (a) Node Degree $= 0.65$. (b) Node Degree $= 3$. (c) Node Degree $= 9$.

place only if the node enters the *active* state. In simulation terms, this new state transitions happens with a probability, p. In addition, this transition takes place with a random delay. The inspiration for this model comes from the term "user vigilance" describing a person's resilience to a virus [29].

Figs. 6 and 7 show the results for network and spreading efficiency using a range of p from 10 percent to 100 percent. The values on the y-axis show the average number of times a coupon is received *throughout* the simulation (Fig. 6), and the percentage of nodes that receive the coupon *by the end* of the simulations (Fig. 7).

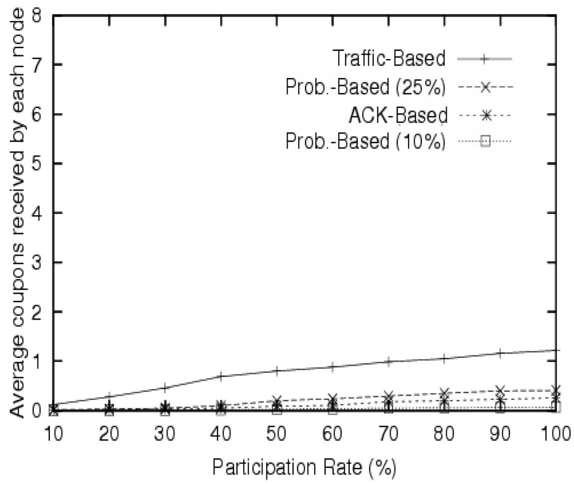
Overall, our results indicated that for a range of p values, there are two important relationships. First, low values of p affect spreading efficiency only in sparse environments. Second, when we consider dense environments, the *ACK-Based* scheme offers the most stable network behavior. This result occurs because, throughout the different participation levels, the overall network load was restricted to a specific range, irrespective of the number of interested users. The only disadvantage occurred in sparse environments where spreading took slightly more time compared to the other schemes (Fig. 7a). However, this trade-off in time versus spreading is likely only to be relevant if the application is such that very rapid spreading is required.

For the second assumption, earlier work on "landscape ecology" [15] shows that spatial characteristics play an important role in virus spreading, a role not reflected in the SIR model. Therefore, the ideal scheme should accommodate scenarios in which nodes move between different "infection" areas, e.g., the clusters and movement pattern shown in Fig. 1.

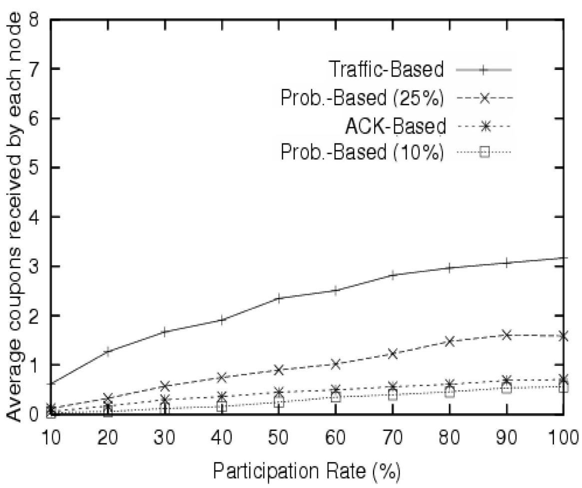
In simulation terms, nodes no longer move freely in all directions of a single terrain. This time, nodes are grouped into five areas, with each group being allocated a specific territory in the terrain. All territories were distinct and far enough apart so that any transmission taking place in one did not reach the others. These terrain areas were connected using *star*, *linked-list*, and *mesh* topologies [4]. At the start of each simulation, nodes move only within the specified geographical limits of their area. Movement to an adjacent territory only occurred according to a straightforward algorithm. At a regular time interval, each node decided whether to change areas or not based on a specified probability, $p_{migrate}$. The aim was then to examine how the system works when users are not free to move anywhere in the grid but are grouped into different areas.

Overall, our results show that the different algorithms still have the same basic characteristics. Apart from an expected delay in the spreading of a coupon, due to the limited movement of nodes between clusters, the introduction of a coupon to a new area follows the same principles of detecting the presence of new nodes and infecting them. The *ACK-Based* scheme limited load on every occasion with the only disadvantage being a slightly higher relative delay in the detection of new susceptible nodes.

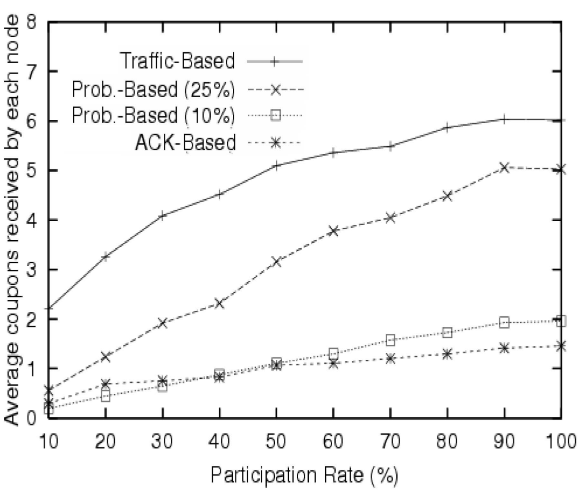
As a final observation, an overall disadvantage of both the *Traffic-Based* and *ACK-Based* schemes is that they need to store previously received coupons. The main reason for this overhead is to identify when a duplicate coupon has been received. However, with a potentially large number of different coupons, there is an obvious limitation on how many coupons a node will be able to store. Limiting the number of stored coupons could thus have an impact on



(a)



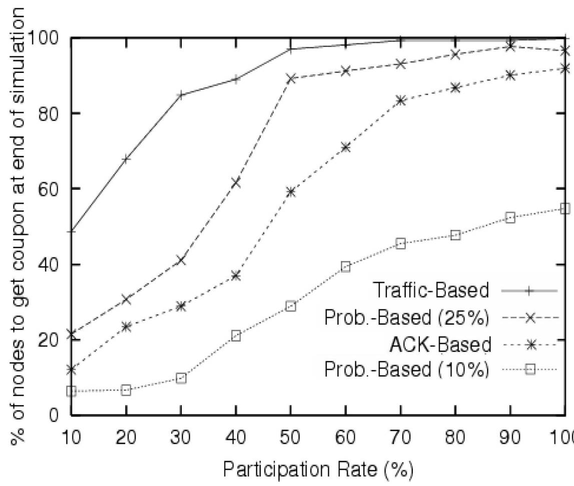
(b)



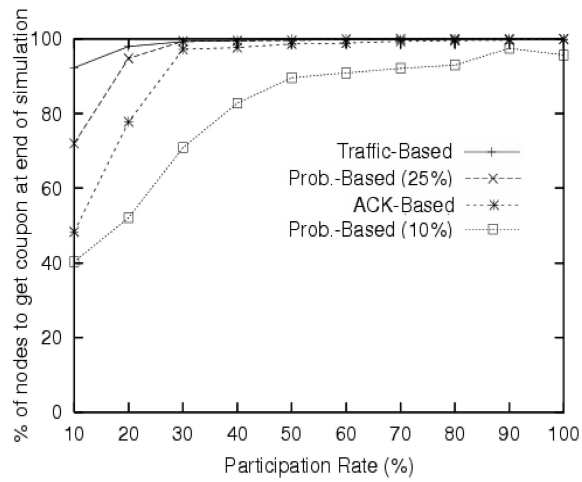
(c)

Fig. 6. Network efficiency for various participation levels. (a) $\text{Node Degree} = 0.65$. (b) $\text{Node Degree} = 3$. (c) $\text{Node Degree} = 9$.

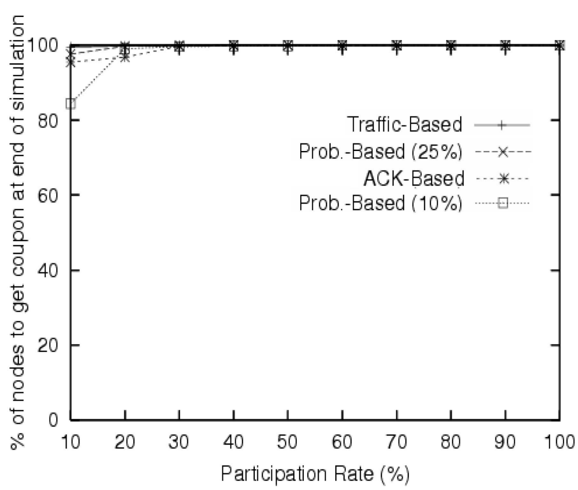
overall performance since nodes will falsely assume that a duplicate coupon is fresh when, in fact, it has been previously received.



(a)



(b)



(c)

Fig. 7. Spreading efficiency for various participation levels. (a) $\text{Node Degree} = 0.65$. (b) $\text{Node Degree} = 3$. (c) $\text{Node Degree} = 9$.

Our simulations showed two points. First, even a small buffer is enough to control a local burst of broadcasts. This is due to the characteristics of a burst since they occur in dense environments and within a small time period. Second, if a coupon is received at a later point in time and duplication cannot be detected, a new small burst will occur. This result might actually be an advantage since the node may have moved to a new area with few infected nodes.

In summary, our initial evaluation has shown two basic properties of `Coupons`. First, it is an efficient for distributing a piece of information based on the power of epidemic-style spreading. Second, the negative effects of the broadcast storm problem can be controlled using the *ACK-Based* approach. Although, for different subcases, the performance can be matched by the *Probability-Based* scheme, this only occurs if the optimal probability value is known, and node density remains constant.

5 Extended Evaluation and Implementation

Using `Coupons` in a real-world environment means that additional factors need to be considered. To this end, we extended our evaluation in two different ways. First, we modified our simulations to address the prospect of malicious behavior. Second, we designed an instance of the `Coupon` protocol, implemented it, and deployed it in a small proof-of-concept testbed.

5.1 Considering Malicious Behavior

Our previous evaluation has ignored an important aspect of human behavior: the possibility that a set of users cheat and override the defined behavior. Cheating can happen in order to either gain an unfair advantage or to deliberately cause extensive damage to the system. In this section, we investigate all of these cases.

One kind of cheating is the attempt to deviate from the defined mechanism in order to obtain more credit. In this respect, the most extreme example would be for nodes to override the default mechanism and to broadcast intensively all of the time. For the sake of simplicity, we define this behavior as a new broadcast frequency algorithm and name it *Cheating Based*. In order to examine the robustness of each scheme against such a behavior, we run two sets of simulations. In the first set, there is no cheating, all of the nodes conform to the prescribed algorithm behavior. In the second set, we assume that the first x percent deviate and use the *Cheating-Based* algorithm. Each of the two sets of simulations were performed with three characteristics. First, we assumed that at the end of each simulation run, every node would use the coupon, assuming it had one. Second, by using a simple example of the *pyramid* award scheme, we gave $\frac{100}{i}$ points to each node whose ID was referenced at the $i\text{th}$ position in the coupon. For example, the node whose ID was in the first coupon slot would receive 100 points, in the second, 50 points, and so on. A maximum of five levels were used in all experiments. Finally, we measured the gain of the cheating nodes by comparing the points they earned through cheating (the second experiment) to those they earned as normal users (the first experiment).

Our results show two main points. First, they verified our analysis, since for all three mechanisms, the majority of the points were awarded to the users who managed to be among the very first to get the coupon, irrespective of whether they had cheated or not. Second, apart from the very first few nodes, the benefit from deviating was either unclear or, in some cases, even negative.

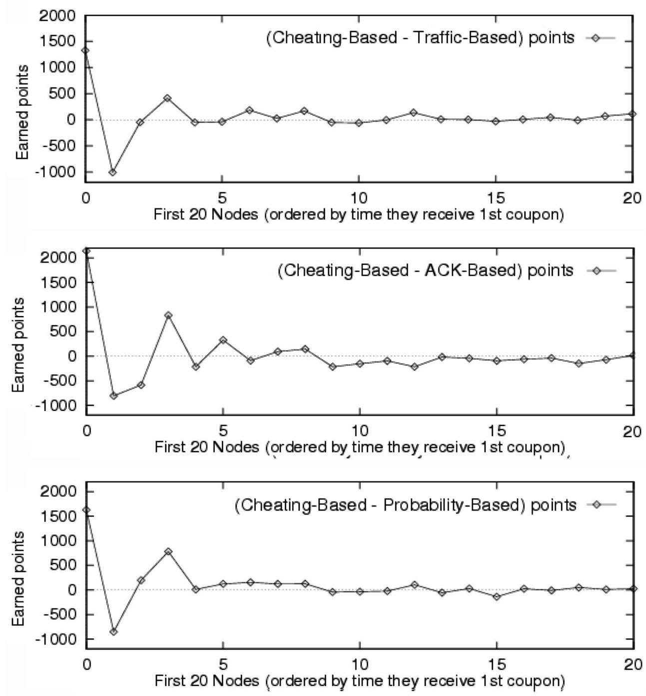


Fig. 8. Difference between 10 percent and 0 percent cheating users for each of the three broadcast schemes (*Traffic-*, *ACK-*, and *Probability-Based*) from top to bottom.

As a representative example, Fig. 8 shows the robustness of each of the three schemes for a fairly dense environment $(\text{node degree} = 3)$. In each of the graphs, we order the nodes based on the time when they first received the coupon. Fig. 8 shows only the first 20 nodes for the sake of clarity. The values on the $y\text{-axis}$ are calculated by subtracting the points earned when there is no cheating from the points earned when the first 10 percent cheated. As an example, the value of 1,200 on the top graph indicates that the first node earned 1,200 *more* points by cheating. Overall, Fig. 8 shows that apart from the first node, most subsequent nodes do no better and, in some cases, even experience a loss of points. Given the complexity associated with overriding the defined behavior, in addition to the operational cost of constant transmission, we regard our results as a strong indication that users would have little incentive to cheat. The *Cheating-Based* scheme has an important advantage over the other algorithms only in sparse environments. However, in such cases, network overhead due to constant broadcasts was very low due to the lack of nearby users. Finally, similar results were obtained as the percentage of cheaters was increased beyond 10 percent.

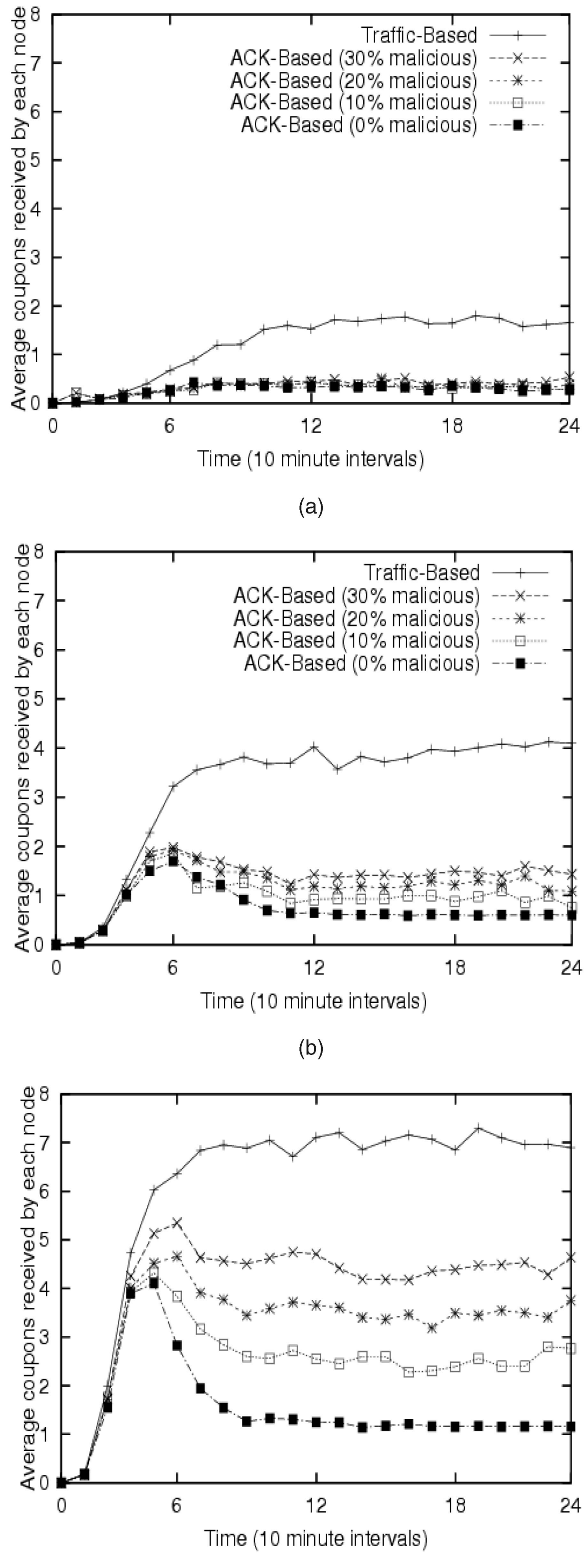
In the second type of maliciousness, we assume that a number of users aim to unbalance the system by exploiting a weakness of the algorithms. To this end, we focused our investigation on the *ACK-Based* scheme for two reasons: first, because it offers the most promising performance compared to the other alternatives and, second, because of its dependence on ACK messages. We measured system performance assuming that a subset of users send ACK messages irrespective of whether they had seen the coupon before or not. The aim would be to deceive the surrounding nodes in order to raise the number of redundant broadcasts. Our results, shown in Fig. 9, show the high resiliency of our algorithm, since, in the worst case (100 percent malicious users), the results match those of the *Traffic-Based* scheme. Although we consider this level of network load prohibitive, a 100 percent rate of malicious nodes an extreme case. More moderate rates, from 10 percent to 30 percent, would harm the system only in highly dense environments. In this case, although the load is restricted within certain limits, less than five coupons received every 10 minutes for 30 percent malicious users, certain precautions might be required. One enhancement would be to keep a record of old acknowledgments along with the sender identities. Then, an acknowledgment received at a later stage from the same node would be discarded. Alternatively, the sleep period could be extended.

Overall, Coupons is a robust scheme against different types of malicious behavior. This feature is due to the nature of the application since it contains two inherent protection mechanisms. These are, first, the randomness of the opportunistic contact and, second, the dynamics of epidemic-style spreading. Both characteristics make it difficult for any user to control and subsequently exploit any potential weaknesses. The best a user can do to exploit the system is to be among the very first users to introduce the coupon to a new area. However, this lack of fairness is not a weakness but, rather, simply an efficient system.

5.2 Considering a Testbed Implementation

In this part, we extend our evaluation framework by designing and implementing a protocol for Coupons and then deploying it in a small testbed. The motivation for this step was to study the interaction between two and three nodes for two main reasons, first, to evaluate the accuracy of our simulations. As simulations have been shown to incorrectly model key properties (e.g., physical layer properties [16], [9]), we used our implementation to verify the accuracy of our simulation results. Second, we evaluated whether the behavior of the algorithms matched real-world behavior and whether there were any new issues raised when attempting to actually implement the protocol. Overall, our implementation and testing was a useful exercise. Not only did we validate the behavior of all three algorithms, but also through our testing, we identified several extensions that, while it is difficult to accurately simulate, have been shown to offer further performance improvements. We did not proceed with the deployment of a large-scale testbed as we determined that the understanding of our system would not be significantly enhanced.

We implemented all three broadcast mechanisms on a 802.11b testbed. Each node, a laptop running the Linux-Fedora 4 operating system, was equipped with a Cisco Aironet 350 PCMCIA card and operated at the nominal bit rate of 11 megabits per second (Mbps). Experiments were then performed in two stages: first, we tested basic node interactions with an emphasis on understanding the impact of wireless medium unpredictability. To this end, we used three nodes within an office environment and located them within a range of 1 to 40 meters (the indoor range for 11 Mbps). In the second stage, the focus was shifted to the



(c)

Fig. 9. Effect of malicious behavior on network efficiency for the *ACK-Based* scheme. (a) $\text{Node Degree} = 0.65$. (b) $\text{Node Degree} = 3$. (c) $\text{Node Degree} = 9$.

robustness of the system under more dynamic conditions. This second set of tests was performed by running multiple user instances on each node. Movement was then emulated by randomly starting and terminating each instance. Based on these tests, we made a number of observations.

Our first observation was that the wireless medium had little impact on the overall system behavior. In particular, packet loss did not affect the general algorithm operation. Second, when a user entered an area of uninfected nodes and transmitted a coupon for the first time, all surrounding users would attempt to immediately reply with an ACK. This ACK-implosion was addressed by providing a simple extension to the *ACK-Based* scheme. When a node receives a coupon for the first time, it waits for a small random period. If during that interval the node hears another ACK, it refrains from transmitting its own acknowledgment. The final observation revealed that specifying the best sleep interval between successive coupon transmissions is a challenging task. A long sleep period results in missed coupon infection opportunities, while a short period causes increased network load. This observation highlighted the need for a more effective sampling of surrounding network conditions.

To monitor network traffic based on these observations, we modified our solutions. If a node is able to *sense* the surrounding traffic, then it can adapt its behavior correspondingly. Three different types of monitoring can be performed:

- **Ad hoc specific protocols.** The operation of routing protocols [5] could provide instant notification of a new neighbor. However, this option suffers from two limitations. Not only would we assume that all nodes would adopt the same routing protocol, but we would also have to augment the protocol to report routing events to the `Coupons` module.
- **Generic IP protocols.** Monitoring of generic IP protocols such as ARP, DHCP, and ICMP would identify the MAC addresses, or equivalent, of surrounding nodes. Consequently, a node could keep a local cache and transmit a coupon once a new MAC address had been detected. Such a solution is not restricted by any protocol assumptions since there is no dependency on a single routing protocol. The disadvantage, however, is power consumption due to the required network monitoring.
- **Radio layer interference.** The final option is to detect interference at the radio layer [17]. However, this solution not only suffers from high-power consumption, but also from the fact that identification of different nodes is now a much harder task due to potential interference [17], [6].

For reasons of general applicability, we implemented the *Sniffing-Simple* scheme by linking the `Coupons` code to the Ethereal network monitoring tool. Each node then monitored ARP messages, kept a cache of the recently recorded MAC addresses, and broadcast a coupon as soon as a new one was detected. This exercise revealed that with a simple extension, nodes could almost instantly detect the presence of new neighbors. We completed our evaluation by implementing *Sniffing-Simple* in the simulator. The operation of generic IP protocols was emulated by having each node randomly broadcast its MAC address. These broadcasts were repeated over a random interval between 20 and 120 seconds, as this is the average frequency that was monitored in the testbed.

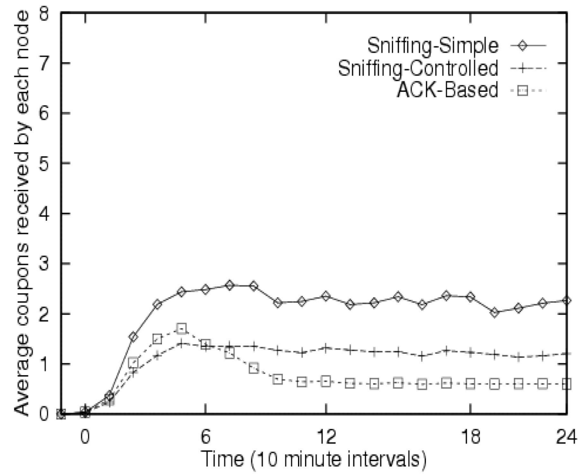


Fig. 10. Network efficiency for *ACK-Based* and *sniffing* schemes $(\text{Node Degree} = 3)$.

Overall, *Sniffing-Simple* gave better spreading efficiency in sparse environments since passing nodes were more successfully detected. In dense environments, however, *Sniffing-Simple* gave higher network load (Fig. 10). The reason is that, in a clustered environment, a node will almost constantly see new users and subsequently keep broadcasting. This issue can be addressed by limiting broadcasting when duplicate coupons are received. This is also shown in Fig. 10, where *Sniffing-Controlled* enters a 5-minute sleep period every time a duplicate coupon is received.

In summary, the two *sniffing* profiles and the *ACK-Based* algorithm are similarly efficient. Therefore, the choice of algorithm will ultimately depend on other factors, e.g., the characteristics of the device. If power consumption is a major issue, as is the case with phones and PDAs, network monitoring would be an unreasonable option. If saving power is of little concern, then *sniffing* would be a better alternative.

6 Conclusions

The goal of this paper has been to investigate applications and services that take advantage of opportunistic interaction. In particular, we have focused on information sharing and cooperation. Our idea, called `Coupons`, is specifically tailored to provide content sharing by incentivizing users. Cooperation is stimulated by adopting a pyramidlike scheme, where users are awarded credits as they pass a received piece of information, a *coupon*, to others.

In our previous work, we only described the basic coupon idea and conducted some basic simulations to show that the idea compares favorably to flooding [8], [7]. In this paper, we first exploited the potential application scope, and then, we defined a new set of periodic broadcast algorithms. Moreover, we extended our experimental methodology by defining a set of parameters to reflect realistic deployment conditions. To this end, we presented a set of possible malicious behaviors and performed a testbed development. The latter revealed additional broadcast algorithms.

Our results show that Coupons is a simple and elegant solution to the problems associated with network flooding. The incentive scheme, coupled with simple enhancements such as a feedback-based backoff mechanism, provide a robust and adaptive solution that brings the principles of peer-to-peer into the wireless world.

REFERENCES

[1] A. Kermarrec and A. Ganesh, "Efficient and Adaptive Epidemic-Style Protocols for Reliable and Scalable Multicast," *IEEE Trans. Parallel and Distributed Systems*, vol. 17, no. 7, July 2006.

[2] A. Datta, S. Quarteroni, and K. Aberer, "Autonomous Gossiping: A Self-Organizing Epidemic Algorithm for Selective Information Dissemination in Wireless Mobile Ad-Hoc Networks," *Proc. Int'l Conf. Semantics of a Networked World (ICSNW '04)*, June 2004.

[3] A. Khelil, C. Becker, J. Tian, and K. Rothermel, "An Epidemic Model for Information Diffusion in Manets," *Proc. ACM Int'l Workshop Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM '02)*, Sept. 2002.

[4] M. Cantwell and R. Forman, "Landscape Graphs: Ecological Modeling with Graph Theory to Detect Configurations Common to Diverse Landscapes," *Landscape Ecology*, vol. 8, no. 4, Oct. 1993.

[5] E. Belding-Royer and C. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," *IEEE Personal Comm.*, vol. 6, no. 2, Apr. 1999.

[6] G. Anastasi et al., "Wi-Fi in Ad Hoc Mode: A Measurement Study," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '04)*, Mar. 2004.

[7] A. Garyfalos and K. Almeroth, "Coupon Based Incentive Systems and the Implications of Equilibrium Theory," *Proc. IEEE Conf. Electronic Commerce (CEC '04)*, July 2004.

[8] A. Garyfalos and K. Almeroth, "Coupons: Wide Scale Information Distribution for Wireless Ad Hoc Networks," *Proc. IEEE Global Telecommunications Conf. (GLOBECOM '04)*, Dec. 2004.

[9] H. Lundgren, E. Nordstrom, and C. Tschudin, "Coping with Communication Gray Zones in IEEE 802.11b Based Ad Hoc Networks," *Proc. ACM Int'l Workshop Wireless Mobile Multimedia (WoWMoM '02)*, Sept. 2002.

[10] A. Heinemann, J. Kangasharju, F. Lyardet, and M. Mhlhuser, "iClouds—Peer-to-Peer Information Sharing in Mobile Environments," *Proc. Int'l Conf. Parallel and Distributed Computing (Euro-Par '03)*, Aug. 2003.

[11] J. Luo, P. Eugster, and J. Hubaux, "Route Driven Gossip: Probabilistic Reliable Multicast in Ad Hoc Networks," *Proc. IEEE INFOCOM '03*, Apr. 2003.

[12] S. Jones, *Differential Equations and Math. Biology*. CRC Press, 2002.

[13] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," *Proc. ACM Conf. Special Interest Group on Data Comm. (SIGCOMM '03)*, Aug. 2003.

[14] M. Akdere et al., "A Comparison of Epidemic Algorithms in Wireless Sensor Networks," *Computer Comm.*, vol. 29, no. 13, Oct. 2006.

[15] M. Keeling et al., "Dynamics of the 2001 UK Foot and Mouth Epidemic: Stochastic Dispersal in a Heterogeneous Landscape," *Science Magazine*, vol. 294, no. 5543, Oct. 2001.

[16] M. Takai, J. Martin, and R. Bagrodia, "Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks," *Proc. ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, Oct. 2001.

[17] N. Golmie et al., "Interference Evaluation of Bluetooth and IEEE 802.11b Systems," *ACM Wireless Networks*, vol. 9, no. 3, May 2003.

[18] O. Ratsimor et al., "eNcentive: A Framework for Intelligent Marketing in Mobile Peer-to-Peer Environments," *Proc. Int'l Conf. Electronic Commerce (ICEC '03)*, Sept. 2003.

[19] P. Orbreiter and J. Nimis, "A Taxonomy of Incentive Patterns—The Design Space of Incentives for Cooperation," Technical Report 2003-9, Univ. of Karlsruhe, May 2003.

[20] P. Golle et al., "Incentives for Sharing in Peer-to-Peer Networks," *Proc. IEEE Conf. Electronic Commerce (CEC '01)*, Oct. 2001.

[21] Q. Li and D. Rus, "Communication in Disconnected Ad Hoc Networks Using Message Relay," *J. Parallel and Distributed Computing*, vol. 63, no. 1, Jan. 2003.

[22] R. Ramanathan et al., "Prioritized Epidemic Routing for Opportunistic Networks," *Proc. ACM Int'l Workshop Mobile Opportunistic Networking (MobiOpp '07)*, June 2007.

[23] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using Redundancy to Cope with Failures in a Delay Tolerant Network," *ACM Mobile Networks and Applications*, vol. 35, no. 10, Oct. 2005.

[24] S. Ni et al., "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Proc. ACM MobiCom '99*, Aug. 1999.

[25] S. Zhong et al., "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," *Proc. IEEE INFOCOM '03*, Apr. 2003.

[26] A. Schmidt, *On Incentive Systems Based on Multi-Level Marketing of Virtual Goods*, preprint, Fraunhofer Inst. for Secure Information Technology, Sept. 2004.

[27] D. Scott and A. Yasinsac, "Dynamic Probabilistic Retransmission in Ad Hoc Networks," *Proc. Int'l Conf. Wireless Networks (ICWN '04)*, June 2004.

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

[29] Y. Wang and C. Wang, "Modeling the Effects of Timing Parameters on Virus Propagation," *Proc. ACM Conf. Computer and Comm. Security (CCS '03)*, Oct. 2003.

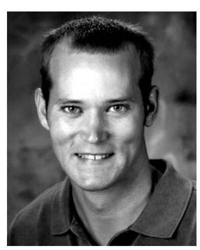
[30] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *Proc. ACM MobiHoc '02*, June 2002.

[31] X. Li, K. Moaveninejad, and O. Frieder, "Regional Gossip Routing for Wireless Ad Hoc Networks," *ACM Mobile Networks and Applications*, vol. 10, nos. 1-2, Feb. 2005.

[32] X. Zeng et al., "Glomosim: A Library for Parallel Simulation of Large-Scale Wireless Networks," *Proc. ACM Workshop Parallel and Distributed Simulation (PADS '98)*, May 1998.

[33] Z. Zhou and L. Haas, "Securing Ad Hoc Networks," *IEEE Network Magazine*, vol. 13, no. 6, Nov. 1999.

Anargyros Garyfalos is currently working toward the PhD degree in the Department of Computing, University of Lancaster, United Kingdom. He is currently an associate at JPMorgan. His main research interests include computer networks, multicast communication, mobile and wireless systems, and financial engineering. He has published a number of journal and conference proceedings. He is also a guest member in the Department of Computer Science, University of California, Santa Barbara. He is a student member of the IEEE.



Kevin C. Almeroth is currently a professor in the Department of Computer Science, University of California, Santa Barbara, where his main research interests include computer networks and protocols, wireless networking, multicast communication, large-scale multimedia systems, and mobile applications. He has published extensively with more than 150 journal and conference proceedings. He is also heavily engaged in stewardship activities for a variety of research outlets including journal editorial boards, conference steering committees, new workshops, and the IETF. He is a member of the ACM and a senior member of the IEEE.