Controlled Wi-Fi Sharing in Cities: A Decentralized Approach Relying on Indirect Reciprocity

Elias C. Efstathiou, Pantelis A. Frangoudis, and George C. Polyzos, *Member*, *IEEE*

Abstract—In densely populated cities, Wi-Fi networks—private or otherwise—are ubiquitous. We focus on the provision of citywide broadband communication capability to mobile users through private Wi-Fi networks that are in range but belong to others. We form a club that relies on indirect reciprocity: Members participate in the club and provide free Wi-Fi access to other members in order to enjoy the same benefit when they are away from their own Wi-Fi network. Our club scheme does not require registration with an authority and does not rely on centrally issued club identities: Members create their own identities (public-private key pairs) and receive signed digital receipts when they provide Wi-Fi service to other members. These receipts form a distributed receipt graph, parts of which are used as input to an indirect reciprocity algorithm that classifies club members according to their contribution. We show that our algorithm can sustain cooperation within the club and is robust to attacks by free-riders. We implement and evaluate our proposed club algorithms on commodity Wi-Fi routers and dual-mode cellular/Wi-Fi phones. Because we anticipate that Wi-Fi telephony will be a popular club application, we present and evaluate a secure and decentralized architecture for citywide voice (and multimedia) communications that is compatible with our club both from an architectural as well as an incentives perspective.

Index Terms—Wi-Fi, community networks, cooperation, decentralization, indirect reciprocity, Wi-Fi telephony.

1 Introduction

The low cost and ease of deployment of Wi-Fi networks, combined with the fact that Wi-Fi operates in unlicensed frequency bands, has made Wi-Fi the technology of choice for local-area wireless connectivity in residential, corporate, municipal, and campus settings. Usually, Wi-Fi networks are also connected to the Internet over fixed broadband links. Today, the cost of fixed broadband is low, access capacity has increased, and Wi-Fi signals pervade many cities. However, most private Wi-Fi networks are security-enabled, and when users are away from their base they must usually rely on the more expensive cellular network for voice and data communications. In this paper, we focus on the provision of Internet access to mobile users through private Wi-Fi networks that are in range but belong to others. We present a *club scheme* that encourages owners of Wi-Fi networks to provide free Wi-Fi access to other club members that are in range, in order to enjoy the same benefit when they themselves are away from their base. With dual-mode cellular/Wi-Fi phones now available from many manufacturers, we propose that such a scheme can complement cellular networks in cities where Wi-Fi density is high. As mobile multimedia traffic is increasing, this scheme benefits both casual users (by lowering their cellular phone bills) and, in the long term, cellular operators (by allowing them to save cell capacity and charge a premium for other value-added ubiquitous connectivity services with quality guarantees). Here, in addition to focusing on casual usage, we mainly consider low mobility scenarios and do not concentrate on the real-time handoff of Wi-Fi connections from one Wi-Fi access point to another.

1.1 Model and Assumptions

We will assume that Wi-Fi networks are connected to fixed Internet access links and that Internet service provider contracts permit the not-for-profit sharing of Internet connections with nearby mobile users over Wi-Fi. Sharing friendly operators exist today [4], [18].

Why should individuals share their Wi-Fi networks with nearby mobile users, especially when there maybe direct and indirect costs involved? Pure altruism is one answer. However, we assume that there are not enough altruists to allow the formation of Wi-Fi sharing communities that can rival cellular networks in citywide coverage. Instead of altruism, we build on *indirect reciprocity*. The idea is that an individual participates in our club and provides free Wi-Fi access to club members in order to enjoy the same benefit when mobile; mobile users who do not share their own Wi-Fi networks are excluded from this club, which provides them with an incentive to share.

To increase its chances of adoption, a distinctive characteristic of our club scheme is that it is fully decentralized. Club members do not register with a central authority or another trusted third party, and club members do not have to know or trust other club members. In addition, members create their own club identifiers (IDs) and we assume that members can use an unlimited number of these IDs. This assumption makes the problem of encouraging cooperation more difficult compared to similar problems where exactly one unique and long-lived ID per node is assumed.

- *The authors are with the Mobile Multimedia Laboratory, Department of Informatics, Athens University of Economics and Business, 47A Evelpidon & Lefkados Str., Athens 113 62, Greece.*
E-mail: {efstath, pfrag, polyzos}@aueb.gr.

Manuscript received 12 July 2009; revised 24 Nov. 2009; accepted 15 Dec. 2009; published online 28 Apr. 2010.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-07-0283.
Digital Object Identifier no. 10.1109/TMC.2010.79.

Furthermore, we assume that members can modify the modules that implement the club algorithms at will; that is, we do not rely on tamperproof hardware or software to make the system work.

All member IDs are unique public-private key pairs. We assume that it is impossible to forge digital signatures and that private signing keys remain private. We do not require the existence of a Public Key Infrastructure (PKI) and we do not assume that club members know the ID of any other member through some other means.

Finally, we assume that members, if and when they are granted access to a foreign Wi-Fi network, will securely tunnel all their traffic to a trusted Internet gateway (which can be hosted on their own Wi-Fi router at home, as in our reference implementation), so we are not concerned with the security of the wireless link or with Wi-Fi routers that attempt to eavesdrop on the traffic they relay.

1.2 Overview of Our Sharing Scheme

Our club sharing scheme works as follows: Each club member owns and manages a Wi-Fi network connected to a fixed Internet access link. The ideal result of our scheme is to encourage club members to match their consumption with at least an equal amount of contribution. By *consumption*, we refer to the volume of Internet traffic a mobile club member relays through foreign Wi-Fi networks, and by *contribution*, we refer to the volume of traffic a member's own Wi-Fi network relays for others.

Free-riding members, who contribute much less than they consume should find it hard to obtain service of good quality. And only short-term history is relevant: Members should continuously contribute in order to be able to continuously consume.

Members sign digital *receipts* when they obtain service from another member. These receipts form a logical *receipt graph*, which is physically distributed. Parts of this graph are used as input to an *indirect reciprocity algorithm* that identifies contributing members by running heuristics on the available graph. Receipts are disseminated within the club using a *gossiping algorithm*. The club Wi-Fi routers that execute the reciprocity and gossiping algorithms do not need to exchange any information over the Internet, so the risks of network security attacks as well as other implementation complexities are lowered: Gossiping only takes place between a Wi-Fi router and a foreign member who is locally connected to it over Wi-Fi.

1.3 Organization of the Paper

This paper is organized as follows: Section 2 presents the main club entities. Section 3 describes the three club algorithms. Algorithm performance is evaluated via simulations in Section 4. Section 5 presents our reference implementation on the Linksys WRT54GS Wi-Fi router, along with measurements of its performance. Section 6 presents a prototype club application, secure Wi-Fi telephony, along with an experimental evaluation. Related work is presented in Section 7. We discuss additional issues in Section 8, where we also conclude the paper.

2 System Entities

In this section, we present the main club entities: members, receipts, and the receipt graph.

2.1 Members

We consider citywide Wi-Fi sharing clubs comprising thousands of members, each with a Wi-Fi network that provides coverage to specific publicly accessible areas. Each member generates a member identifier, which is a unique (with high probability) public key whose corresponding private key is kept secret by the member. We do not require a Public Key Infrastructure, and member public keys remain uncertified. Members will present their public keys when they request service.

2.2 Receipts and Receipt Graph

A club receipt is evidence that Wi-Fi service was provided. Receipts are generated according to a *receipt generation protocol* (see Section 2.2.1) every time a member uses the Wi-Fi network of another member, and are sent over Wi-Fi to the Wi-Fi router that is providing service. Receipts consist of:

1. The public key of the contributing member.
2. The public key of the consuming member.
3. A *time stamp*, which notes the start time of the Wi-Fi session.
4. A *weight*, which notes the volume of traffic the Wi-Fi router relayed for the consuming member during the session.
5. The consuming member's *digital signature*, which is a hash of the four fields above, asymmetrically encrypted with the consuming member's private key. One can *verify* this digital signature using information on the receipt itself: the consuming member's public key.

Receipts form a logical receipt graph. The vertices of this graph are member IDs (public keys) and the weighted directed edges point *from* a consuming member ID *to* a contributing member ID. An edge's weight is equal to the volume of traffic the source consumed from the destination; that is, the weight of an edge is equal to the sum of the weights of the corresponding receipts, and the direction of the edge signifies an "owes to" relation.

2.2.1 Receipt Generation Protocol

During a Wi-Fi session, Wi-Fi routers periodically request receipts from all foreign members currently connected, in order to account for the traffic relayed up to that point. The request period is a local parameter. Requests contain the public key of the contributing member. Consuming members are required to produce a signed receipt (if they do not, their session is stopped), which must contain the weight the contributor is currently measuring. Wi-Fi routers only store the last receipt in such a series, per Wi-Fi session. (By requesting receipts periodically, the contributing member attempts to minimize the risk of a consumer *not* signing a receipt after he obtains service.) A useful side effect is that, by receiving a (verifiable) signed receipt, the contributing Wi-Fi router is certain that the consumer's session has not been hijacked because we assume no one else possesses the private signing key that corresponds to the consuming member's public key.

Because the receipts contain a time stamp that indicates the session start time, there needs to be loose time synchronization between the consuming member and the contributing Wi-Fi router so that the member is sure he is not signing a receipt for some future or past point in time.

New receipts are then stored in a local receipt repository (in our reference implementation, this repository is hosted on the Wi-Fi router). If the repository is full, the receipt with the oldest time stamp is deleted.

3 Algorithms

In this section, we present the three club algorithms. These are: 1) the indirect reciprocity algorithm, 2) the gossiping algorithm, and 3) the club entry algorithm.

3.1 Indirect Reciprocity Algorithm

The indirect reciprocity algorithm guides the contribution decisions of club members who adopt it and use it on their Wi-Fi routers. For the following analysis, we disregard that member IDs are implemented using public keys: We only need to remember that each member ID in a club is unique. Note also that each receipt can be uniquely identified from the following 3-tuple: {*contributing member ID*, *consuming member ID*, *time stamp*}.

A set of receipts defines a logical receipt graph **G** with the following characteristics:

1. The vertices in **G** represent member IDs.
2. **G** is a *directed* graph. A directed edge $C \rightarrow P$ exists in **G** if the source, Member C (Consumer), has obtained service at least once from the destination, Member P (Provider).
3. **G** is a *weighted* graph. The weight of the $C \rightarrow P$ edge is equal to the sum of the weights of the corresponding receipts. By *corresponding receipts*, we refer to all the receipts issued in the system that show C as the consumer and P as the contributor.

For the following analysis, when stating that "Member P cooperates with Member C" we mean that Member P provides Wi-Fi service to Member C. The result of this cooperative action will be the eventual generation of a new $C \rightarrow P$ receipt, with weight equal to the volume of traffic that P relayed for C during the session. This receipt then becomes part of the system, which results either in the creation of a new $C \rightarrow P$ edge if none existed before or in the increase of the weight of an existing $C \rightarrow P$ edge.

The idea behind the indirect reciprocity algorithm is to use the risk of exclusion as an incentive to encourage cooperation, and realize this by cooperating only with known cooperators. The problem then becomes how to distinguish cooperators from noncooperators.

We introduce a metric, called *Indirect Normalized Debt* (IND). Its values range from 0 to 1, inclusive. Consider a prospective consumer, Member C, that requests service from a prospective contributor, Member P. P computes the IND to C by examining **G**. The closer IND is to 1, the more P "owes" to C according to IND. The closer IND is to 0, the less P owes to C. IND is the product of two factors, r_1 and r_2, which we present below.

3.1.1 Maximum Flow and Factor r_1

We now examine how free-riding attackers can tamper with **G**. Because IDs are free, attackers can engage in egregious *false trading*: They can create an arbitrary number of fake graph edges with arbitrary weights, which do not correspond to real Wi-Fi sessions. To do that, attackers must be in control of an edge's source vertex; that is, they must possess the private signing key that corresponds to the public key, which represents the ID of that vertex. They can control the vertex if they created it themselves, and they are able to create a vertex because each attacker can generate an arbitrary number of IDs. These IDs can exist in parallel with his real ID or the attacker may possess no real ID at all. (By *real ID*, we refer to the ID, if any, that is associated with a member's Wi-Fi router.) On the other hand, attackers cannot create edges starting from vertices they do not control, and they cannot change the weight or delete an existing edge.

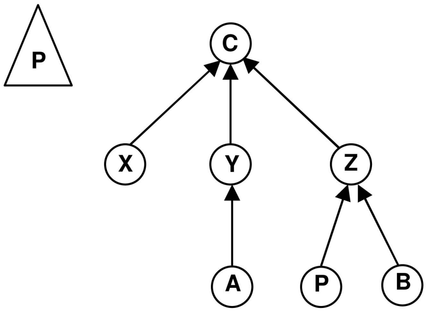


Fig. 1. Member C visits the Wi-Fi router of Member P and requests service. P sees C as the root of a tree that also contains P as consumer. P, therefore, "owes" to C (here, indirectly via Z). If P did not detect himself anywhere in the tree, the entire tree could have been fabricated by C (because we assume identities X, Y, Z, A, and B are unknown to P and could have easily been created by C).

Our indirect reciprocity algorithm uses *maximum flow* (from now on referred to as *maxflow*), a graph algorithm that measures the amount of flow that can pass from a source vertex to a destination vertex of a weighted graph. This is inspired by the work of Feldman et al. who analyzed the properties of a maxflow-based decision function in a similar context [20].

Let us show an example of how a heuristic based on maxflow can assist in detecting free-riders. In Fig. 1, a prospective contributor, P, must decide whether or not to cooperate with a prospective consumer, C. According to the graph that is available to P, C appears at the root of a tree of receipts, all of which directly or indirectly point to C, signifying that C contributed service in the past and is a good contributor that deserves to be served. But is he?

From P's perspective, there is no guarantee that such a tree of receipts is not completely fabricated by C. However, if P can detect himself somewhere in the tree in the role of consumer, he can be sure that at least one of the receipts in this tree is real—his own outgoing receipt (assuming such a receipt passes digital signature verification successfully). P can then be sure that he owes, directly or indirectly, to C.

How *much* does P owe to C? To counter attacks by C that attempt to present larger weights for some of the receipts in the tree, P's debt to C must be bounded by the only receipt that P can trust to be real: his own. Similarly, any bottleneck that appears on the path from P to C is taken into account: If P owes 1,000 Kbytes to Z and Z owes 10 Kbytes to C, then P owes 10 Kbytes to C.

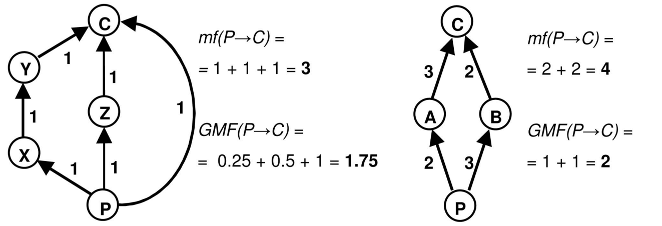


Fig. 2. In the two weighted graphs above we execute both maxflow and GMF from P to C. After isolating the component flows that make up the maxflow result, GMF discounts each component flow exponentially according to the length of the path that the flow traverses. Thus, GMF is a measure of the "directness" of debt. In the example above, the leftmost path $\mathrm{P} \rightarrow \mathrm{X} \rightarrow \mathrm{Y} \rightarrow \mathrm{C}$, which has a flow value of 1, is multiplied by $2^{(1-\mathrm{k})}$ with $k = 3$.

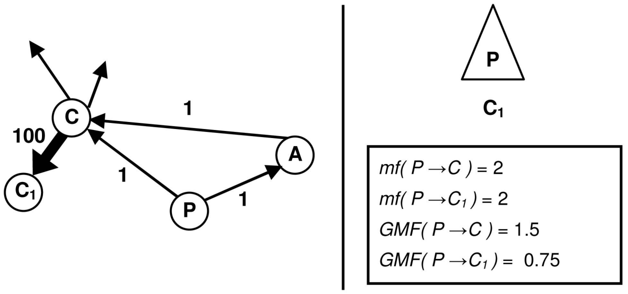


Fig. 3. If Member C appears as C_1, forwarding his "contribution reputation" one hop away to identity C_1 and consequently erasing all potential debts (his "consumption reputation"), the GMF result from P to C_1 will be half the GMF result C would have achieved had he appeared simply as C.

A way to aggregate these paths of debt is to use the maxflow algorithm: The maxflow from P to C indicates the *total* indirect and direct debt that P owes to C [20]. Under maxflow, a member C that fabricates receipts cannot appear to P as contributor, no matter how much false trading he engages in. If none of C's IDs provided service, there is no flow from P to any of C's IDs.

We now present the first factor, r_1, of the Indirect Normalized Debt metric:

$$r_1 = \min\left(\frac{mf(P \rightarrow C)}{mf(C \rightarrow P)}, 1\right), \tag{1}$$

where $mf(P \rightarrow C)$ is the result of maxflow from P to C. This factor is an indicator of how much more P owes to C, compared to what C owes to P (directly and indirectly) [20]. In (1), if the denominator equals 0, $r_1 = 1$ for positive numerators and 0 otherwise.

The above analysis depends on the assumption that the *owes to* relation is transitive. This, in general, is a matter of definition and depends on the application. Using Fig. 1 as an example and assuming unit weights on the graph edges, we note that as soon as P serves C and C issues a new $C \rightarrow P$ receipt, a circle of debt ($P \rightarrow Z \rightarrow C \rightarrow P$) is closed: Every node that is part of this circle has contributed and consumed exactly once (as part of this circle) and the debt for all can be considered settled even if settlement happened indirectly. We know that in a self-sufficient community, the total amount of consumption is matched by at least an equal amount of production. Permitting only two-way and multiway exchanges of equal value is one way to make sure this is true. The intuition, therefore, behind the use of maxflow is to detect (and close) potential circles of debt, allowing for a self-sufficient community. Simulations (Section 4) show that a cooperation decision based on maxflow, if adopted by club members, increases the club's overall satisfaction levels, and is also beneficial to the (self-interested) member who adopts it.

3.1.2 Generalized Maximum Flow

We now introduce another heuristic on the receipt graph, which we call GMF. GMF measures the "directness" of debt on **G** and is inspired by the class of *generalized maximum flow* algorithms [36]. To calculate GMF from a source vertex to a target vertex, we first execute a standard maxflow algorithm in order to identify all the component flows that contribute to the result of maxflow. The result of GMF is, in general, different from the result of a maxflow for a given pair of vertices: In contrast to maxflow, GMF *discounts* the value of flow as we move away from the source vertex. More specifically, for each component flow that contributes to the result of maxflow, GMF multiplies the value of flow by $2^{(1-k)}$, where k is equal to the hop count, measured in vertices, that we traverse to get from the source vertex to the target vertex following that flow. If the result of GMF is equal to the equivalent maxflow then all debt from P to C is direct: C is only one hop away from P in **G**. For two examples, see Fig. 2.

GMF counters the following attack: Assume we only relied on the r_1 factor presented previously. Let C be a consuming member; assume, however, that instead of consuming using identity C, each time C consumes he uses a fresh identity $\mathrm{C}_i, i > 0$, which C never reuses. In itself this would not benefit C because $mf(X \rightarrow C_i) = 0$ for all X and for all i (because there can be no receipts pointing to a newly created ID). Therefore, P would owe nothing to C_i according to maxflow.

However, if C fakes an interaction with this new ID by creating a new fake $C \rightarrow C_i$ edge with weight w, then, if

$$w \geq mf(P \rightarrow C),$$

we have

$$mf(P \rightarrow C_i) = mf(P \rightarrow C),$$

that is, by the definition of maxflow, all the flow from P that reaches C also reaches C_i if the weight of the $C \rightarrow C_i$ edge is enough to carry it. And, in addition, $mf(C_i \rightarrow P) = 0$ because ID C_i is fresh and has no debt (outgoing receipts).

This way, ID C_i earns all the "contribution reputation" of ID C and none of its debts: If P owed even the smallest positive indirect or direct debt to C, r_1 would equal 1.

With GMF, however, the GMF to such a member would be consistently less than what it would have been had the member not conducted this attack (see Fig. 3).

3.1.3 Factor r_2 and Final Algorithm

The input to the indirect reciprocity algorithm is **G** and the ID of the prospective consumer, Member C. The algorithm is executed by the prospective contributor, Member P, whenever a prospective consumer requests service. The output of the algorithm is a value between 0 and 1,

TABLE 1
Indirect Reciprocity Algorithm

```
Step 1: Prospective consumer, Member C, visits prospective contributor, Member P.
Step 2: Member C presents his member ID to P.
Step 3: P computes $IND_{P \to C}$ (on the receipt graph available).
    Step 3.1: P computes factor $r_1$.
    Step 3.2: P computes $GMF(P \to C)$ and updates $gmf_{avg}$.
    Step 3.3: P computes factor $r_2$ using the results from 3.2.
    Step 3.4: P computes $IND_{P \to C}$ using the results above.
Step 4: P translates the $IND_{P \to C}$ metric to contribution decision. □
```

inclusive, which we call the Indirect Normalized Debt of P to C and denote $IND_{P \to C}$.

To compute $IND_{P \to C}$, P computes its two factors. We have already introduced factor r_1 in (1). The second factor, r_2, is defined as

$$r_2 = \frac{GMF(P \to C)}{gmf_{avg}}, \tag{2}$$

and if $gmf_{avg} = 0, r_2 = 1$ for positive GMF results and 0 otherwise. gmf_{avg} is the average GMF that Member P observes in the club. It is updated according to the formula below each time P computes a new GMF result, GMF_{new}:

$$gmf_{avg} \leftarrow gmf_{avg}a + GMF_{new}(1 - a), \tag{3}$$

where a is a discounting constant (we use $a = 0.75$). gmf_{avg} is computed locally by each Wi-Fi router. At the start of the Wi-Fi router's lifetime, it is set to 0. Finally, we define $IND_{P \to C}$ as

$$IND_{P \to C} = \min(1, r_1 \cdot r_2). \tag{4}$$

It is then up to the prospective contributor, Member P, to translate this metric to a level of service to offer to the prospective consumer, Member C. The intuition (and guideline) is that the benefit C receives should be analogous to $IND_{P \to C}$, and 0 if $IND_{P \to C} = 0$. Two straightforward technical methods to lower C's benefit for small INDs, compared to the ideal, include: 1) P's Wi-Fi router limiting C's Internet bandwidth or 2) P causing large artificial time delays before it grants Internet access to C. Both of these methods offer a continuous set of possible values for P to choose from (depending on the value of the IND).

The complete algorithm is summarized in Table 1.

3.2 Gossiping Algorithm

So far, we used the receipt graph as input but we did not specify where the graph is physically stored. Our club is fully decentralized with no authority to store its history. Also, the Wi-Fi routers of the members do not communicate with each other, and receipt repositories (hosted on the Wi-Fi routers in our reference implementation) have a maximum number of receipts they can store.

If we did not introduce additional functionality, the following two things would hold true: 1) The receipt repository of a Member P would only contain receipts that showed P as the contributor; 2) the value of IND computed on such a partial view of the receipt graph would always equal 0 because without *outgoing* edges from P, maxflow and GMF return 0.

TABLE 2
Gossiping Algorithm

```
Step 1: Prospective consumer, Member C, obtains latest receipts from his home Wi-Fi router.
    Step 1.1: Receipts are placed in his mobile repository, replacing older ones.
Step 2: C visits prospective contributor, Member P.
Step 3: C presents all receipts from his mobile repository to P.
Step 4: P merges these receipts with his own receipts.
    Step 4.1: Receipts are placed in P's receipt repository, replacing older ones.
Step 5: P uses the available receipts (including those from Step 4) as input to the indirect reciprocity algorithm. □
```

We introduce *gossiping* to disseminate receipts in the club in a practical and incentive-compatible way and allow members to have a less-biased view of the graph. We require that prospective consumers carry with them, in *mobile* repositories, a part of their receipt repository. More specifically, clients periodically request to be *updated* with the latest receipts from their home Wi-Fi router. The Wi-Fi router presents them the most recently acquired receipts from its receipt repository. Because a receipt can be as small as 130 bytes (see Section 5), a phone-based client can easily download and store thousands of receipts.

The second phase of gossiping involves the mobile client presenting receipts from its mobile repository to prospective contributors when the client visits them to request service. Assume Member C is requesting service from Member P. C has a clear incentive to show receipts from C's repository to P. These receipts, originating from C's Wi-Fi router, include receipts earned by C that show C as the contributor: If P were to consider these receipts as additional input to the indirect reciprocity algorithm, this can only increase $IND_{P \to C}$.

Receipts are then further disseminated throughout the club via the following procedure: P takes the receipts that C presented and *merges* them in his own receipt repository. Again, the standard rule concerning receipt replacement applies: If a new receipt is inserted in the repository when the repository is full, the oldest receipt is removed. In practice, this means that P would never include in his repository a receipt with a time stamp that is older than the oldest receipt in the repository—which, effectively, defines a *time horizon* for P, and encourages C to carry "fresh" receipts and, therefore, to also keep his Wi-Fi router in sharing mode in order to earn fresh receipts that point *to* C.

P's Wi-Fi router would then also update Member P with the newest receipts from P's repository. Some of these receipts would have arrived via visitors and the gossiping procedure presented above. Member P, in turn, would present these receipts to the Wi-Fi routers that he visits, disseminating them further in the system. The algorithm is summarized in Table 2.

3.3 Club Entry Algorithm

New club members must first contribute to the club before they can consume. This is because the indirect reciprocity algorithm searches for direct or indirect debt from a prospective contributor to a prospective consumer. If the consumer is new and has never contributed before, he would be no different from a free-rider (he is owed nothing)

TABLE 3
Club Entry Algorithm

```
Step 1: New club member, Member N, sets up a Wi-Fi router,
        chooses a patience value, sets $k = 0$.
Step 2: Member N provides and requests service.
    Step 2.1: As contributor, N provides service to anyone
              who requests it.
        Step 2.1.1: N stores the newly earned receipt in
                    his receipt repository.
    Step 2.2: As consumer, N requests service.
        Step 2.2.1: If service is granted, N issues a receipt
                    and increases $k$ by 1.
        Step 2.2.2: If $k < patience$ go to 2, else go to 3.
Step 3: N exits club entry phase.
    Step 3.1: As contributor, N provides service guided by
              the indirect reciprocity algorithm. □
```

according to the IND metric computed by the prospective contributor. Similarly, if the new member attempted to use the indirect reciprocity algorithm to guide his contribution decisions, he would find that all members appear as free-riders to him: IND to anyone is zero because the new member has no outgoing receipts yet, either (he owes nothing).

To break this deadlock, the *club entry* algorithm requires that, to join the club, a new member N starts contributing *without* executing the indirect reciprocity algorithm at first. (Member N can, however, use gossiping; that is, N's Wi-Fi router will conduct merging of receipts when a consumer requests service from N.) In parallel, we assume that the new member will start trying to consume service from the club. In the beginning, he will be unsuccessful: There will be no incoming receipts of the form $X \rightarrow N$ to show, and no such receipts will be stored by the prospective contributors either.

However, as soon as a receipt of the form $X \rightarrow N$ is earned by the new Member N, the probability that N can obtain service from the club becomes positive. In the receipt example above, if Member X, who consumed service from N and issued the $X \rightarrow N$ receipt, was also a good contributor, others that owed directly or indirectly to X will now also owe (indirectly) to N.

We specify a club entry heuristic: Each new member has a parameter called *patience*. As a new Member N attempts to consume from the club, at some point he will eventually be offered service and will issue a receipt (assuming of course he has started to contribute to the club using his Wi-Fi router).

As soon as Member N issues a number of receipts equal to the *patience* parameter, Member N leaves the club entry phase and starts to use the indirect reciprocity algorithm properly to guide his decisions. See also Table 3.

The intuition is that after a number of successful consumptions, N deduces that he has become a known club contributor and that he can now select the ones he serves according to the result of the indirect reciprocity algorithm. If he tried to be selective earlier, he would hurt his own standing. On the other hand, if he never started to use the indirect reciprocity algorithm, he would continue to incur unneeded costs by potentially helping free-riders who offer no useful receipts (see also Section 4.4).

4 Simulations

In this section, we present simulations that evaluate the performance of our algorithms and show their robustness to attacks by free-riders. Time in our simulations is measured in *rounds*. During a round, club members are randomly matched and each member gets one chance to consume. The number of these *matches* per round is equal to the number of available members. This number changes: At Round 1, we start with one member and at the end of each round a new member joins the club. This models club growth and continues up to a maximum number of n members. For simplicity, we assume that Wi-Fi sessions result in a new receipt with unit weight. Also, if, in a match, the prospective contributor decides to provide service, his *score* is reduced by 1, which represents the (we assume fixed) cost of providing Wi-Fi service. At the same time, the score of the consumer is increased by a number from 0 to $b_{\max}$; this number is a linear function of the IND (see Section 3) as measured by the contributor. This represents the (we assume variable) benefit of obtaining club service, and $b_{\max}$ represents the maximum benefit one can obtain from a Wi-Fi session. We, therefore, assume that contributors have the technical means to lower the benefit that consumers will obtain (at no extra cost to contributors) by reducing the quality of the service in various ways (see also Section 3.1.3) based on the result of IND, and also by denying service if IND is 0, in which case contributor cost is 0 and not 1.

The above are inspired by similar evaluation frameworks in sociology and biology that are used to study cooperation among self-interested agents [19], [20].

From now on, we will use the term *strategy* to describe the algorithm that a member uses to guide his contribution decisions in the matches where he is the prospective contributor. One available strategy is RECI (Reciprocity), under which members adopt the algorithms we presented in Section 3. The *rating* of a strategy (following [20]) is the average of the running averages of scores per round of the strategy's *followers*, with each term weighted according to how many rounds a member has been using the strategy (so that veterans of a particular strategy carry more weight than amateurs). In experiments with *evolutionary learning*, members switch strategies with a probability proportional to the difference between the rating of the new strategy and the rating of their old strategy. This models real-world communities, where members learn (through various means external to the system) of better strategies that may benefit them more. There is also a small (configurable) chance that members *mutate*, that is, switch to another strategy for no reason, which helps introduce all available strategies to the club irrespective of the original strategy mixture.

The remaining simulation parameters are the sizes of the *receipt repository* and the *mobile repository*, measured in receipts. We do not model the update phase of the gossiping algorithm in detail: Rather, we assume that the two repositories of a member are always synchronized (if the mobile repository is smaller, it contains only the newest receipts of the receipt repository). Finally, we simulate the club entry algorithm as specified in Section 3.3.

Important simulation outputs are a strategy's rating and the *Social Welfare* (SW), which is the sum of the scores of all club members. The *optimal SW* is the SW that would be generated in a club if, in all matches and in all rounds, the prospective contributor contributed $b_{\max}$ benefit to the prospective consumer, at a cost of 1 to himself. In that case, SW would increase by $b_{\max} - 1$ after every match. The *fraction of optimal SW* at a specific simulation round is the

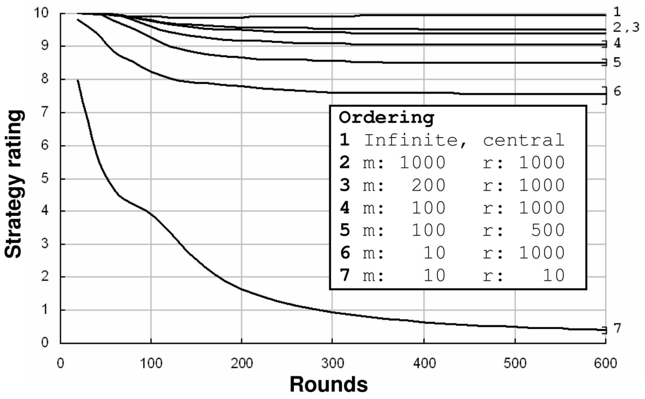


Fig. 4. RECI rating (in a community of 100 RECI followers) as rounds progress, corresponding to seven different scenarios regarding the amount of available information in the system.

ratio of the SW until that round, divided by the optimal SW that could have been generated until that round.

Our goal as designers is to maximize SW while assuming that each member is only interested in maximizing his own individual score.

4.1 Gossiping versus an Idealized Central Storage: Effect on Cooperation Levels

In this experiment, we wanted to see how the amount of available information affects cooperation levels. We experimented with various repository sizes. They are labeled in Fig. 4 "m: 10 r: 10," "m: 10 r: 1,000," and so on, with the two numbers representing, respectively, the sizes of the mobile repository and the receipt repository on the Wi-Fi router (measured in receipts). For comparison purposes, we also introduce an idealized system, where a central repository with infinite capacity stores every receipt. We label this case "infinite, central."

Here, we do not allow mutation or evolutionary learning and the club consists only of followers of the RECI strategy. The total eventual number of members is 100 (and every member will have joined by Round 100).

In Fig. 4, we plot the rating of the RECI strategy as rounds progress. At the end of each line, we also plot a range corresponding to the 95 percent confidence interval calculated on our set of measurements. (Note that for the top three lines the confidence intervals are too small to see.)

We see that the average rating of the RECI strategy has practically stabilized by Round 600 for the simulations represented by the top five lines. We have $b_{\max} = 11$ in this experiment, so the optimal RECI rating would be 10. In the "infinite, central" case, RECI reaches an average rating of 9.94 by Round 600. We can see that even with "infinite, central" there is still efficiency loss compared to the optimal. This results from the way the indirect reciprocity algorithm computes Indirect Normalized Debt by relying on maxflow ratios that are rarely exactly equal to 1, and by having a bias against ratios that are higher than 1. In fact, the only situation in which we could obtain a RECI rating equal to the optimal rating is if we had infinite memory and only two members in the club that took turns in consuming and providing service to each other. The loss of efficiency is small, however.

We see that even small (compared to the total number of members and receipts that are being generated) mobile repositories enable cooperation: For mobile repositories with only 10 receipts, RECI rating is 7.55 at Round 600.

The bottom (seventh) line shows what happens when the receipt repositories are not large enough to support the number of members in the club: RECI average rating drops to 0.41 at Round 600. This cooperation "collapse" happens because there is not enough information in the repositories to allow one RECI follower to identify another RECI follower as a good cooperator by examining the receipt graph. Most INDs computed equal 0, that is, most prospective consumers are mistaken for free-riders. Therefore, the prospective contributor does not cooperate, no Wi-Fi session occurs, and no new receipt is generated.

We see, from this experiment, that with simple gossiping, we can attain levels of cooperation that are very close to an idealized mode of operation with full information.

4.2 Random Waypoint Mobility: Effect on Cooperation Levels

In this experiment, we wanted to test another mobility model for the consumers in addition to the random uniform matching that we use in all the other experiments. In the *random waypoint* mobility model, members move in a more realistic way. Our random waypoint model is the same as the random waypoint model proposed in the literature [22], with a pause time equal to 0. More specifically, we arrange 100 Wi-Fi routers (corresponding to 100 members) in a rectangular area (side: 10 km). The rectangular area is divided into 100 equal squares and the Wi-Fi routers are placed at the center of each square.

Members move in the following manner: In the beginning of the simulation, they are randomly placed anywhere inside the rectangular area. Each member then selects: 1) a destination in the rectangular area, and 2) a speed chosen uniformly at random from the range $[u_{\min}, u_{\max}]$. Each member then proceeds to his destination using his chosen speed. Once the member arrives at the destination, he requests service from the Wi-Fi router nearest to his current position. We assume that the coverage in the rectangular area is total, so there is always one such router. Then, independently of the result of his previous request, the member picks another random destination and speed, and starts moving toward it, repeating the process.

In Fig. 5, we plot the average rating of the RECI strategy as rounds progress (a *round* in this mobility model is still a set of matches equal to the number of members: in this case, 100). We use $u_{\min} = 1$ m/s and two different maximum speeds: 2 and 20 m/s. Although for both scenarios, the ratings stabilize around a value, these values are lower than those achieved under the previous mobility model for the same repository sizes (in this experiment, the size of the mobile repository equals 200 and the size of the receipt repository equals 1,000; for this scenario, we obtained the value of 9.40 (Line 3) in Fig. 4).

In addition, the rating for the high maximum speed is lower than that for the low maximum speed. This is explained as follows: With the random waypoint model, the consumption request rate is not uniform any more. Depending on the speeds and distances chosen, a member that reaches his

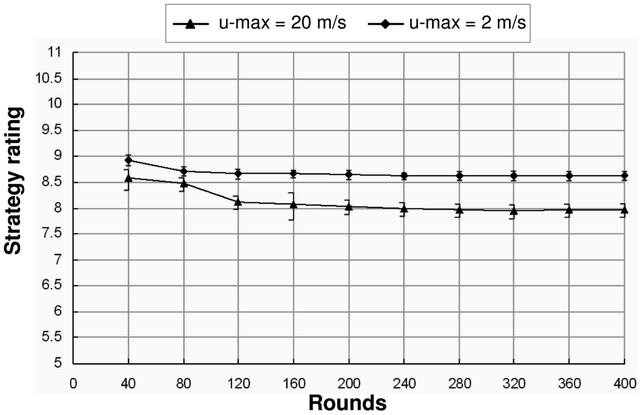


Fig. 5. The rating of the RECI strategy for two different maximum speeds of the random waypoint mobility model.

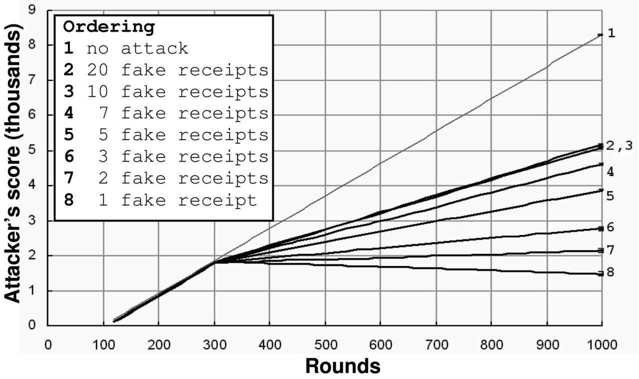


Fig. 6. Erase debt. To obtain a higher IND, a member attempts to "erase" all his debt by using a fresh ID (see attack description in Section 3.1.2). The GMF heuristic detects this attack.

destination first will request service earlier than a member who is still moving toward his own destination. In this way, members are no longer homogeneous in their consumption and contribution rates. A member has a chance to contribute whenever he receives a visitor in his area. However, after providing service, some time may pass without any visitors; at the same time, the same member, acting as a consumer, is moving. If the time he takes to arrive to a destination is small, he may make one or more consumption requests before he gets a chance to contribute. This asymmetry becomes evident in the receipt graph, and the effect is intensified when the maximum available speed is high. However, cooperation in the club is sustained.

4.3 Erase Debt and the Need for GMF

In this experiment, we wanted to see if a specific adversarial strategy, *erase debt*, benefits the attacker who follows it. We already discussed this attack in Section 3.1.2 where we introduced the need for the r_2 factor and the GMF heuristic. As before, $b_{\max} = 11, n = 100$. The receipt repositories contain (up to) 1,000 receipts and the mobile repositories (up to) 200.

Here, Member 100 (the attacker) joins a club of RECI followers at Round 100, himself following RECI. It can be seen in Fig. 6 that his score increases linearly—as is to be expected (Line 1 in Fig. 6). At Round 300, however, Member 100 decides to switch to the erase debt strategy in an attempt to cheat the r_1 factor and erase his debts and gain maximum benefit. In practice, if Member 100 were able to do just that, this means that he could keep his "contribution reputation" and consume as much as he wanted and erase the consequences of his consumption in the manner we described in Section 3.1.2. Such a strategy could be very successful, it could invite more followers, and it could lead the club to cooperation collapse. But can it?

To perform the attack, Member 100 pushes all his "good" reputation to a new ID by connecting this new ID to his old one using a number of (unit weight) receipts, which is a parameter of our experiment. (The equivalent, in a more realistic setting, is for Member 100 to connect his old ID to his new ID by issuing one receipt with large weight, enough to carry the flow from C to C_i—see also Section 3.1.2).

The average maxflow result in the club of this experiment is slightly less than 10, so 10 fake (unit weight) receipts are usually enough to carry all of the flow from C to C_i without losses. However, in Fig. 6, we see that if he performs the attack using one receipt only, Member 100 starts *losing* score as the GMF heuristic detects the attack.

For the remaining lines (going from 2 to 20 fake receipts), we see that the attacker does slightly better; however, in *all* cases, the attacker does worse compared to simply not performing the attack (Line 1).

4.4 Withstanding Invasion by Free-Riding Strategies in Larger Clubs

In this experiment, we wanted to see if RECI can outperform altruistic and free-riding strategies in larger clubs while simultaneously keeping SW high. We define a universe of four possible strategies: RECI, ALLC, ALLD, and RAND. ALLC is a strategy that stipulates "cooperate with everyone" while ALLD stipulates "cooperate with no one." (ALLC and ALLD are standard names, inspired by The Prisoner's Dilemma game and its two moves: *Cooperate* and *Defect* [7]). RAND stipulates "cooperate with 50 percent probability." Therefore, ALLC, ALLD, and RAND do not rely on the receipt graph for their decisions. The club starts out with 100 percent RECI followers but the three new strategies are introduced in it because there is a nonzero mutation probability for club members. Evolutionary learning is in effect, which will reward strategies that perform (individually) well in the short-term, by increasing their followers (which can hurt the club overall though).

We wanted to test larger clubs ($n = 1{,}000$) but continue to keep mobile repositories relatively small at 200 receipts (6,000 receipts for the repositories on the Wi-Fi router).

In the experiment (Fig. 7), the club grows to 1,000 members. We see that the fraction of optimal SW is 0.82 at Round 10,000, and has practically stabilized. That is, even though evolutionary learning is in effect, we see that the three new strategies cannot outperform RECI and cannot win over RECI followers. ALLD performs the worse: ALLD followers never cooperate (keeping their costs at 0) and RECI followers detect this using the indirect reciprocity algorithm and deny them service. The only way for ALLD followers to obtain service and increase their score is to visit a RECI follower during his club entry phase, or to visit an ALLC follower.

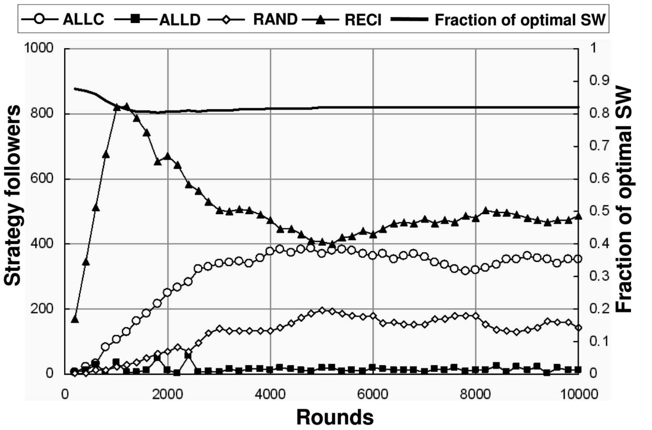


Fig. 7. A club where four strategies (RECI, ALLC, ALLD, and RAND) coexist, with the cooperative strategies RECI and ALLC outperforming the rest (and ALLC "riding on RECI's coattails," relying on RECI's policing of the club against free-riding strategies ALLD and RAND).

ALLC followers, in turn, cooperate with everyone: This means RECI followers reward them by cooperating with them. However, ALLC followers also cooperate with ALLD followers, something that RECI followers avoid. Therefore, ALLC costs are higher and ALLC rating is consistently lower than RECI's rating. RAND followers, as expected, perform somewhere in between ALLC and ALLD followers.

It is interesting to note that, eventually, ALLC followers perform very near RECI followers and so the number of ALLC followers in the club is near the number of RECI followers. This is because, effectively, RECI followers "protect" ALLC followers by driving ALLD followers close to extinction because of RECI's refusal to cooperate with ALLD. The more RECI followers mutated to ALLC and did *not* quickly evolve back to RECI (because of the relatively small difference in rating), the more these new altruists would indirectly invite the ALLD strategy to return. With additional free-riders, RECI followers would again start to outperform the altruists, which would lead to an increase in RECI followers and a reduction to both ALLC and ALLD followers, a process that can indefinitely maintain the dynamic cooperative equilibrium we observe here. (The simulation here continues for 10,000 rounds, which is 10 times longer, with a 10 times larger club, than our previous experiments.)

TABLE 4
Platform Specifications

Characteristic	Athlon XP 2800	Linksys WRT54GS
CPU speed	2.08 GHz	200 MHz
CPU type	AMD Athlon XP 2800	Broadcom MIPS32
RAM	512 MB	32 MB
Storage	60 GB HD	8 MB Flash (read only) 32 KB NVRAM
Operating system	Linux kernel 2.4.18	Linux kernel 2.4.18
Cryptographic library	OpenSSL 0.9.8 beta 5	OpenSSL 0.9.8 beta 5
Compiler	gcc 3.2	gcc 3.2
Compiler optimizations	-O3	-O3 –mcpu=r4600 -mips2

TABLE 5
Receipt Generation Protocol

Message	Description	Direction
CONN	Wi-Fi session initiation request	Client → Wi-Fi router
CACK	Wi-Fi session initiation response	Wi-Fi router → Client
RREQ	Receipt request	Wi-Fi router → Client
RCPT	Receipt	Client → Wi-Fi router

5 Implementation

5.1 Platforms

We implemented the club algorithms and protocols on the Linux-based Linksys WRT54GS Wi-Fi router [1] by including our modules in its firmware. We also implemented the client side of the protocols as a .NET application designed for phones running Windows Mobile.

We conducted experiments to test the performance of the software running on the Linksys Wi-Fi router. For comparison, we also ran the software on an AMD Athlon XP 2800 laptop. Table 4 shows the platform specifications.

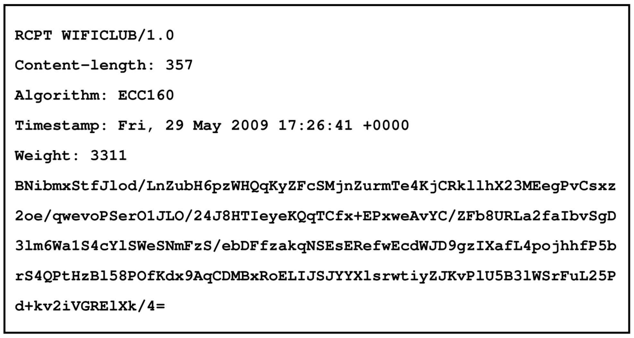
5.2 Protocol Messages

The club receipt generation protocol comprises four text-based messages and operates on top of TCP/IP. A Base64 encoder is used to convert binary data to a text-based wire format. Table 5 shows the protocol messages, the entities that exchange them and a description.

Fig. 8 shows the format of an RCPT message that contains a receipt (in its Base64 wire format) signed using the Elliptic Curve Digital Signature Algorithm (ECDSA).

5.3 Network Access Control and Accounting

The Wi-Fi router software uses the Linux *netfilter/iptables* packet filtering framework. We built our module for controlling network access and measuring the volume of traffic relayed per client on top of netfilter using the *libiptc* library. When Wi-Fi clients associate with the Wi-Fi router, they are assigned dynamic IP addresses from an address



```
RCPT WIFICLUB/1.0
Content-length: 357
Algorithm: ECC160
Timestamp: Fri, 29 May 2009 17:26:41 +0000
Weight: 3311
BNibmxStfJlod/LnZubH6pzWHQqKyZFcSMjnZurmTe4KjCRkllhX23MEegPvCsxz
2oe/qwevoPSerO1JLO/24J8HTIeyeKQqTCfx+EPxweAvYC/ZFb8URLa2faIbvSgD
3lm6Wa1S4cYlSWeSNmFzS/ebDFfzakqNSEsERefwEcdWJD9gzIXafL4pojhhfP5b
rS4QPtHzBl58POfKdx9AqCDMBxRoELIJSJYYXlsrwtiyZJKvPlU5B3lWSrFuL25P
d+kv2iVGRElXk/4=
```

Fig. 8. RCPT message: It contains the receipt encoded in its Base64 wire format. The *time stamp* and *weight* receipt fields can be found in human-readable form also.

TABLE 6
Maximum Flow Performance

	Athlon XP 2800		Linksys WRT54GS	
Number of receipts	100 members	1000 members	100 Members	1000 members
1000	0.43 ms	0.23 ms	12.64 ms	3.75 ms
10,000	5.88 ms	12.72 ms	59.27 ms	134.04 ms

pool via DHCP and are denied Internet access until the Wi-Fi router runs the indirect reciprocity algorithm. When a Wi-Fi session ends (an RREQ request times out) the allocated IP address is again blocked and measurement of traffic for or from that address is stopped.

5.4 Receipt Repository and Indirect Reciprocity Algorithm Implementation

We implemented the receipt repository on the Wi-Fi router. For the execution of the indirect reciprocity algorithm, the Wi-Fi router needs to calculate maximum flows. For this purpose, a FIFO variant of the *push-relabel* maximum flow algorithm [25] was implemented. Its $O(V^3)$ worst-case running time is long, we, therefore, used the *global relabeling heuristic* [11], [25], which yielded dramatic performance improvements. We measured this performance for various graph instances. In our experiments, we created random directed graphs comprising 1,000 and 10,000 receipts (edges), and 100 and 1,000 members (vertices). Table 6 shows the pure CPU time spent on executing the algorithm (measured with the Linux *times* function). Each reported value is the average time spent on the execution of the maximum flow algorithm for 20 random source-destination pairs of the same graph.

6 Applications

The proliferation of dual-mode cellular/Wi-Fi phones leads us to expect that one popular club application will be Wi-Fi telephony. We expect that club members would choose to make Wi-Fi calls if possible and we evaluate a secure and decentralized architecture to achieve this.

In our proposed architecture, when a mobile user uses the Wi-Fi router of another club member, he first sets up a secure VPN connection with his home Wi-Fi router and tunnels all his Internet traffic there, protecting it from eavesdropping by the visited Wi-Fi router and local wireless users. The reason we included VPN server functionality on top of our club Wi-Fi router was to negate the need for extra equipment other than the club basics.

Fig. 9 shows a basic call scenario. Suppose that mobile members M1 and M2 wish to establish a bidirectional voice session. We assume M1 and M2 have already established Wi-Fi sessions with visited Wi-Fi routers V1 and V2, respectively, and are tunneling all their traffic to their home Wi-Fi routers, H1 and H2, respectively. Note that here we consider only low mobility scenarios and we do not concentrate on the issue of connection handoffs.

In order to initiate the call, M1 must somehow signal M2. Because of our decentralized club architecture, there is no obvious rendezvous point (like a central SIP registrar). Also, like in a standard cellular call, we require that M1 only knows M2's cellular phone number. In our approach to signaling, to signal M2, the Windows Mobile software running on M1's phone "silently" sends a cellular SMS text message to M2's cellular phone number.

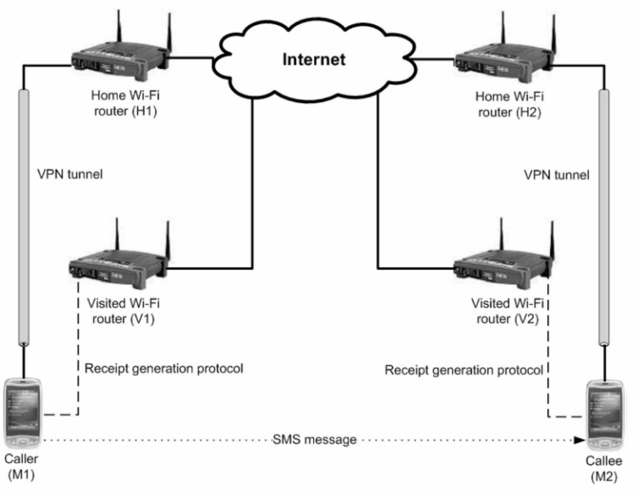


Fig. 9. Basic call scenario. In order for the two calling parties to be able to rendezvous without any additional information about each other, the phone client of the caller (M1) signals M2 by sending to the phone number of M2 a cellular text message (SMS) that contains the current IP address of M1's home Wi-Fi router (H1). This is enough for M2 to locate M1 over the Internet.

The message contains H1's public IP address (which we assume M1 knows either because it is static or because dynamic DNS is used). This is all M2 needs to know to respond over the Internet with his voice stream and his own home's (H2) public IP address (and thus, the voice stream for the other direction can also start). M2's packets are first tunneled to H2, from there directed to H1, and finally tunneled to M1.

6.1 Experimental Evaluation Goals

Here, we estimate the overhead of our architecture and how the additional overhead of security affects Wi-Fi calls. More specifically, we measure the number of voice calls of acceptable quality that a simple club Wi-Fi router like the Linksys WRT54GS can sustain when IPsec is used to secure all communication between the mobile client and his home VPN gateway, which is hosted on his home Wi-Fi router. Thus, several Wi-Fi router functions must run in parallel, consuming router resources. These are:

1. Relaying local wireless traffic from visiting mobile clients to the Internet and performing NAT.
2. Running the club receipt generation protocol for each visitor and verifying all receipts received.
3. Acting as home VPN gateway for its owner who is currently visiting other Wi-Fi routers.

Here, we study the combined effect on call quality.

6.2 Call Quality Assessment Methodology

In our experiments, we emulated voice conversations by setting up bidirectional UDP flows between two laptops. We implemented our own traffic generators, sending 50 packets per second, each with 20 bytes of audio payload and a 12-byte RTP header. This traffic pattern corresponds to the G.729

codec, which is used by many available dual-mode cellular/Wi-Fi phones. The 20 bytes of packet payload contain 20 ms of voice. Each laptop was connected to a separate Wi-Fi router and each voice call lasted 90 seconds. We assumed that at the receiver end there is a dejitter buffer, which introduces a 60 ms delay in the play out process.

We initiated parallel calls between the two laptops and collected delay and loss information for each packet at the receiver end for one direction. To estimate user-perceived voice quality, we used the evaluation method proposed in [12]. This method reduces ITU-T's E-model [24] to transport level metrics that are directly measurable. Using the proposed methodology, we derived a score representing the subjective quality of a voice call based only on network delay, jitter, and packet loss measurements. For the codec configuration described above, the so-called R-score [12] is given by the following formula:

$$\begin{aligned} R = {} & 94.2 - 0.024 \cdot (d_{network} + 85) \\ & - 0.11 \cdot (d_{network} - 92.3) \cdot H(d_{network} - 92.3) - 11 \\ & - 40 \ln[1 + 10 \cdot (e_{network} + (1 - e_{network}) \cdot e_{dejitter})], \end{aligned}$$

where $d_{network}$ is the end-to-end delay, $e_{network}$ represents network loss, $e_{dejitter}$ represents loss in the dejitter buffer, and $H(x) = 1$ if $x > 0$, or 0 otherwise. The *R-score* is then mapped to a subjective Mean Opinion Score (MOS) through [12]:

$$MOS = \begin{cases} 1, & if\ R < 0, \\ 4.5, & if\ R > 100, \\ 1 + 0.035 \cdot R + 7 \cdot 10^{-6} & if\ 0 < R < 100. \\ \quad \cdot R \cdot (R - 60) \cdot (100 - R), & \end{cases}$$

For a call of acceptable quality, average MOS should be over 3.60 (R-score greater than 70).

6.3 Testbed Setup

6.3.1 System Software and Equipment

Our testbed was composed of PCs running a custom-made Linux distribution that we created (based on Knoppix), running kernel version 2.6.8. This distribution included necessary measurement tools, club client protocol implementation, and the Openswan [2] IPsec implementation (all downloadable from our project website [3]).

Two Linksys Wi-Fi routers were used, connected to each other using a 3Com Ethernet switch. We used two Fujitsu Siemens laptops equipped with Intel PRO Wireless 2200 802.11b/g cards. Each of the two laptops was connected to a separate Linksys router using Wi-Fi, operating in DCF mode, with RTS/CTS and fragmentation disabled. Wi-Fi data rate was fixed at 11 Mbps. In our measurements, we calculated one end-to-end delay for each packet, based on packet time stamps. Synchronization between sender and receiver was achieved via NTP.

6.3.2 Receipt Generation Protocol Parameters

The main overhead of the receipt generation protocol is the CPU-intensive signature verification operations. We emulated multiple wireless sessions by performing receipt verifications on the Wi-Fi routers at regular intervals. We assume that calls and Wi-Fi sessions have a one-to-one correspondence and that Wi-Fi routers requested a receipt from each visitor every five seconds. More calls mean more Wi-Fi sessions and, thus, more receipt verifications.

TABLE 7
Maximum Number of Voice Calls of Acceptable Quality

Scenario	Number of calls
Plain	7
Verifications (RSA 1024)	7
Verifications (ECDSA 160)	6
VPN + Verifications (RSA 1024)	5
VPN + Verifications (ECDSA 160)	2

6.3.3 VPN Parameters

The Linksys Wi-Fi router also operated as a VPN gateway, using the Openswan IPsec implementation. The L2TP protocol was used for implementing tunnels and IPsec's ESP [27] was used to secure them [33]. IPsec operated in transport mode using the AES-CBC algorithm (128-bit keys) for data encryption. Preshared keys were used for authentication.

6.4 Results

Table 7 shows the maximum number of simultaneous voice calls of acceptable quality routed through the Linksys Wi-Fi routers. When no IPsec tunnels exist and no receipt verifications occur, seven simultaneous calls can be sustained in our scenario. On the other hand, the cost of using VPNs to secure communication proves to be high, especially when combined with the use of ECDSA for verifying the signatures on club receipts. In this case, due to the fact that ECDSA verifications are very CPU-intensive for the Linksys Wi-Fi router, additional delay is imposed in the routing of voice packets that are coincident with receipt verification events. Jitter that cannot be properly handled by the dejitter buffer is introduced, which leads to a MOS decrease.

A MOS value of 3.60 is the minimum average MOS value that a call should score to be considered acceptable. Table 7 implies that there is a threshold in the number of simultaneous voice sessions after which voice quality dramatically degrades.

In the future, it will be worthwhile to explore more lightweight security mechanisms for protecting streaming media such as voice (for example, Secure RTP) and use TLS or a similar protocol to protect signaling information. Here, we relied on L2TP/IPsec as a general tunneling mechanism mainly because of the L2TP/IPsec client that comes built-in with Windows Mobile devices. In general, however, there are many other security options.

We, therefore, see that a basic Wi-Fi router can simultaneously support all the club protocols and algorithms, act as home gateway for its owner, and relay several quality voice calls over Wi-Fi, all with strong security.

7 Related Work

Catch [30], CONFIDANT [8], and Core [31] are examples of incentive schemes for multihop wireless networks that, unlike our scheme, assume nodes have exactly one ID and, also unlike our scheme, assume nodes are willing to cooperate to perform distributed accounting tasks. Sprite [39], like our scheme, is receipt-based; however, it requires a central Credit Clearance Service that determines the real-world charge and credit to each node. The Nuglet cooperation approach [9], unlike our model, requires that ad hoc network nodes have tamper-resistant score-keeping modules, manufactured by a limited number of trusted manufacturers that cross-certify each other.

A general formal model for cooperation in static multi-hop wireless networks based on game theory and graph theory is presented in [21]. Simulation results show that, in practice, the conditions for cooperation without some form of incentives are virtually never satisfied and that cooperation needs to be encouraged.

Our position that Wi-Fi networks can be shared in a controlled manner was first presented in [15]. Our indirect reciprocity algorithm is a generalization of the algorithm we presented in [17] and an extension to the maxflow decision function presented by Feldman et al. [20] (itself inspired by older work on authentication metrics [29], [34]). In [16], we evaluated a version of the reciprocity algorithm under a different system model: A club member there was modeled as a *group* of individuals who pool their Wi-Fi resources to provide service and who consume in the name of the group (see also Section 8).

The context studied by [20] is similar to ours: There are no identity-certifying authorities and IDs are free. A follow-on work of [20] is [19], where the effect of free IDs on the cooperation strategies studied by Axelrod (*Tit-for-Tat*) [7] and Nowak and Sigmund (*Image*) [32] is analyzed. Their results confirm the seminal result by Friedman and Resnick [23] that free IDs come at a cost.

Sybil attack [14] is the name given to the attack that involves the creation of multiple identities per real entity; this is a fundamental problem in open, self-organized electronically mediated communities without identity-certifying authorities. Sybil attacks can invalidate any number of system assumptions by making collusion-based attacks much easier.

The maxflow decision function that we extend belongs to a class of cooperation strategies that rely on multiway exchanges, themselves a generalization of two-way exchanges but also a special (cyclical) case of indirect reciprocity [26]. Multiway exchanges have been adopted as an incentive scheme for file sharing [5], storage sharing [13], and in our own previous work [17].

Earlier work on partially observable distributed graphs includes [6] and [28]: [6] does not focus on incentive issues, while [28] addresses the incentives of nodes. In [28], they store accounting information only if it is in their interest, however, unlike our scheme, [28] also assumes that nodes are simply willing to share their stored accounting data with others over the Internet.

Micropayment-based incentive techniques for peer-to-peer systems include PPay [38] and Karma [37]. PPay requires an authority (a bank) to generate currency and to check for double spending. Karma requires other peers to keep track of a peer's account balance, assuming that distributed accounting is incentive-compatible. Also, Karma is susceptible to the Sybil attack: A peer can repeatedly join the system and obtain new start-up funds each time. The cryptographic puzzle that new entrants must solve only limits the *rate* of new ID generation.

A work in [35] is with the same objective as ours (fueling Wi-Fi deployment and use) that also focuses on quality-of-service (QoS). There, "wireless ISPs" have multilateral roaming contracts and register with a central authority that maintains reputation records, which are updated with QoS reports submitted by roaming users.

A well-known existing Wi-Fi sharing scheme is the FON system (http://www.fon.com). FON is a company that has assembled a worldwide community of users that share their Wi-Fi networks with others. FON's goal is to provide not just citywide but global Wi-Fi roaming capabilities through FON-affiliated Wi-Fi networks. Their system supports a "free sharing" extension: All FON users who own a FON Wi-Fi router can use other FON Wi-Fi routers for free. Although not all details of the FON scheme are available, it is apparent that, contrary to our scheme, the FON scheme relies on a central authority that issues user identities and guides cooperation decisions.

8 Discussion and Conclusion

As it stands, the controlled Wi-Fi sharing scheme that we propose attempts to balance a member's consumption with at least an equal amount of contribution. However, this may lower the overall value of the scheme for those potential members who live in areas of a city where there are not many visitors to serve. Indirect reciprocity favors symmetry between consumption and contribution, and homogeneity in the population. The more we depart from symmetry, the more our heuristics will fail to differentiate between a good cooperator and an attacker or free-rider, leading to a decrease of Social Welfare in the club.

For restoring symmetry, we see the following two possible extensions to the basic scheme as promising avenues for further work: First, the scheme could be extended to allow individuals to *team* with others in order to join the club as one group, whose *total* contribution would match its consumption [16]. Grouping individuals in teams would also make practical sense, for example, for families and roommates that share the same residence and collectively own only one Wi-Fi router. Teammates can also be geographically distributed, and pool their Wi-Fi routers, enabling the team to provide service in more than one location. With teams, we would need to replace the member identifiers with *team* identifiers. But also, instead of letting all teammates share the private signing key of the team, it would be interesting to explore specific ways a teammate can consume and contribute in the name of the team, but still maintain accountability within the team.

Second, we could extend the scheme by varying the weights on receipt requests depending on the relative scarcity of service: In areas with fewer visitors, the Wi-Fi routers could "charge" more (for example, 2 Kbytes for every 1 Kbyte relayed), making up for the fact there are fewer visitors in the area, and restoring a certain balance.

If our club scheme or a similar Wi-Fi sharing scheme is to be adopted, one of the first real-world problems that needs to be addressed is the legal/regulatory issue of sharing

Internet access links with strangers, for free. Most Internet Service Providers (ISPs) explicitly forbid this and there may be legitimate legal concerns against sharing from the point of view of individual subscribers. And although there are well-established technical means to share Wi-Fi securely and protect the computer networks of microoperators from wireless attackers, such ISP contractual issues pose a hurdle for the club's deployment.

Finally, if a free Wi-Fi sharing scheme is to become a substitute for the cellular system, the issue of fast handoffs of mobile users between Wi-Fi access points needs to be resolved. Initial results [10] on fast handoffs within community-based Wi-Fi installations are promising. Assuming an appropriate density of Wi-Fi networks and extrapolating from the capabilities of today's equipment, a system for fast handoffs between access points, which would also provide automatic reestablishment of secure user tunnels and multimedia streams, without affecting perceived quality of service, may become a reality soon.

Acknowledgments

The authors wish to thank Eleftherios Stefanis, Dimitrios Paraskevaidis, Vasileios Kemerlis, and Fotios Elianos for their invaluable help in implementing and evaluating the original Peer-to-Peer Wireless Network Confederation system and demonstrating it at international conferences. The authors also thank the anonymous reviewers whose comments helped improve the quality of the paper. This research was supported in part by the project "Mobile Multimedia Communications" (EP-1212-13), funded by the research program "Herakleitos—Fellowships for Research at the Athens University of Economics and Business."

References

[1] Linksys, http://www.linksysbycisco.com, 2010.

[2] Openswan, http://www.openswan.org, 2010.

[3] P2PWNC Project Website, http://mm.aueb.gr/research/p2pwnc, 2010.

[4] Speakeasy NetShare Service, http://www.speakeasy.net/netshare, 2010.

[5] K.G. Anagnostakis and M.B. Greenwald, "Exchange-Based Incentive Mechanisms for Peer-to-Peer File Sharing," *Proc. 24th Int'l Conf. Distributed Computing Systems (ICDCS '04),* 2004.

[6] S. Capkun, L. Buttyan, and J.P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks," *IEEE Trans. Mobile Computing,* vol. 2, no. 1, pp. 52-64, Jan.-Mar. 2003.

[7] R. Axelrod, *The Evolution of Cooperation.* Penguin Books, 1990 (first published by Basic Books, 1984).

[8] S. Buchegger and J.Y.L. Boudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation of Nodes: Fairness in Dynamic Ad-Hoc Networks)," *Proc. ACM MobiHoc,* 2002.

[9] L. Buttyan and J.P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *ACM/Kluwer Mobile Networks and Applications,* vol. 8, no. 5, pp. 579-592, 2003.

[10] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A Measurement Study of Vehicular Internet Access Using in situ Wi-Fi Networks," *Proc. ACM MobiCom,* 2006.

[11] B.V. Cherkassky and A.V. Goldberg, "On Implementing the Push-Relabel Method for the Maximum Flow Problem," *Algorithmica,* vol. 19, no. 4, pp. 390-310, 1997.

[12] R.G. Cole and J.H. Rosenbluth, "Voice over IP Performance Monitoring," *ACM Computer Comm. Rev.,* vol. 31, no. 2, pp. 9-24, 2001.

[13] L.P. Cox and B.D. Noble, "Samsara: Honor among Thieves in Peer-to-Peer Storage," *Proc. 19th ACM Symp. Operating System Principles (SOSP '03),* 2003.

[14] J.R. Douceur, "The Sybil Attack," *Proc. First Int'l Workshop Peer-to-Peer Systems (IPTPS '02),* 2002.

[15] E.C. Efstathiou and G.C. Polyzos, "A Peer-to-Peer Approach to Wireless LAN Roaming," *Proc. ACM Int'l Workshop Wireless Mobile Applications and Services on WLAN Hotspots (WMASH),* 2003.

[16] E.C. Efstathiou, P.A. Frangoudis, and G.C. Polyzos, "Stimulating Participation in Wireless Community Networks," *Proc. IEEE INFOCOM,* 2006.

[17] E.C. Efstathiou and G.C. Polyzos, "Self-Organized Peering of Wireless LAN Hotspots," *European Trans. Telecomm.,* special issue on self-organization in mobile networking, vol. 16, no. 5, 2005.

[18] Electronic Frontier Foundation (EFF), "Wireless Friendly ISP List," http://www.eff.org/Infrastructure/Wireless_cellular_radio/wireless_friendly_isp_list.html, 2009.

[19] M. Feldman and J. Chuang, "The Evolution of Cooperation under Cheap Pseudonyms," *Proc. Seventh IEEE Conf. E-Commerce Technology (CEC '05),* 2005.

[20] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust Incentive Techniques for Peer-to-Peer Networks," *Proc. ACM Conf. Electronic Commerce (EC '04),* 2004.

[21] M. Felegyhazi, J.P. Hubaux, and L. Buttyan, "Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks," *IEEE Trans. Mobile Computing,* vol. 5, no. 5, pp. 463-476, May 2006.

[22] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Comm. and Mobile Computing,* special issue on mobile ad hoc networking: research, trends, and applications, vol. 2, no. 5, pp. 483-502, 2002.

[23] E. Friedman and P. Resnick, "The Social Cost of Cheap Pseudonyms," *J. Economics and Management Strategy,* vol. 10, no. 2, pp. 173-199, 1998.

[24] ITU-T Recommendation G.107, "The E-Model, A Computational Model for Use in Transmission Planning," 1998.

[25] A.V. Goldberg and R.E. Tarjan, "A New Approach to the Maximum-Flow Problem," *J. ACM,* vol. 35, no. 4, pp. 921-940, 1998.

[26] B. Greiner and M.V. Levati, "Indirect Reciprocity in Cyclical Networks—An Experimental Study," Discussion Papers on Strategic Interaction 2003-15, Max Planck Inst. of Economics, Strategic Interaction Group, http://ideas.repec.org/p/esi/discus/2003-15.html, 2003.

[27] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," IETF RFC 2406, 1998.

[28] S. Lee, R. Sherwood, and B. Bhattacharjee, "Cooperative Peer Groups in NICE," *Proc. IEEE INFOCOM,* 2003.

[29] R. Levien and A. Aiken, "Attack-Resistant Trust Metrics for Public Key Certification," *Proc. USENIX Security Symp.,* 1998.

[30] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Sustaining Cooperation in Multi-Hop Wireless Networks," *Proc. Second USENIX Symp. Networked System Design and Implementation,* 2005.

[31] P. Michiardi and R. Molva, "CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," *Proc. IFIP TC6/TC11 Sixth Joint Working Conf. Comm. and Multimedia Security,* 2002.

[32] M.A. Nowak and K. Sigmund, "Evolution of Indirect Reciprocity by Image Scoring," *Nature,* vol. 393, pp. 573-577, 1998.

[33] B. Patel, B. Aboba, W. Dixon, G. Zorn, and S. Booth, "Securing L2TP Using IPsec," IETF RFC 3193, 2001.

[34] M.K. Reiter and S.G. Stubblebine, "Authentication Metric Analysis and Design," *ACM Trans. Information and System Security,* vol. 2, no. 2, pp. 138-158, 1999.

[35] N.B. Salem, J.P. Hubaux, and M. Jakobsson, "Reputation-Based Wi-Fi Deployment," *ACM Mobile Computing and Comm. Rev.,* vol. 9, no. 3, pp. 69-81, 2005.

[36] É. Tardos and K.D. Wayne, "Simple Generalized Maximum Flow Algorithms," *Proc. Sixth Int'l Conf. Integer Programming and Combinatorial Optimization,* pp. 310-324, 1998.

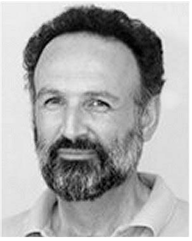
[37] V. Vishnumurthy, S. Chandrakumar, and E.G. Sirer, "KARMA: A Secure Economic Framework for P2P Resource Sharing," *Proc. First Workshop Economics of Peer-to-Peer Systems,* 2003.

[38] B. Yang and H. Garcia-Molina, "PPay: Micropayments for Peer-to-Peer Systems," *Proc. 10th ACM Conf. Computer and Comm. Security (CCS),* 2003.

[39] S. Zhong, J. Chen, and Y.R. Yang, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," *Proc. IEEE INFOCOM,* 2003.

Elias C. Efstathiou received the BSc degree in computer science and telecommunications from the University of Athens in 1999, the MSc degree in computer science from the University College London in 2001, and the PhD degree in computer science from the Athens University of Economics and Business in 2006. He is currently working on new product development at the Hellenic Telecommunications Organization (OTE SA). Previously, he was a visiting lecturer at AUEB, teaching MSc courses on information security and multimedia communications, while also working in industry as an IT consultant. In the past, he was a researcher at AUEB and a software engineer at INTRACOM SA, Department of New Technologies. He served his military service at the Hellenic National Defense General Staff, Cyber Defense Directorate. His current research interests include all-IP networks, fixed-mobile convergence, pricing telecommunications services, and game theory.

Pantelis A. Frangoudis received the BSc and MSc degrees in computer science from the Department of Informatics, Athens University of Economics and Business in 2003 and 2005, respectively. He is a PhD candidate at the same department. His current research interests include wireless networks, Internet multimedia, and network security.



George C. Polyzos received the diploma in electrical engineering from the National Technical University in Athens, Greece, in 1982, and the MASc degree in electrical engineering in 1985 and the PhD degree in computer science in 1989 from the University of Toronto. He has been a professor of computer science at the Athens University of Economics since 1999, leading the Mobile Multimedia Laboratory, and is currently the chair of the Division of Computer and Communications Systems. Previously, he was a professor of computer science and engineering at the University of California, San Diego (UCSD), where he was a codirector of the Computer Systems Laboratory, a member of the steering committee of the UCSD Center for Wireless Communications, and a senior fellow of the San Diego Supercomputer Center. His current research interests include mobile multimedia communications, ubiquitous computing, wireless networks, security, Internet protocols, and performance analysis of computer and communications systems. He is on the editorial board of *Wireless Communications and Mobile Computing* and has been a guest editor for the *IEEE Personal Communications*, *ACM/Kluwer Mobile Networks and Applications*, the *IEEE Journal on Selected Areas in Communications*, and *Computer Networks*. He has been on the program committees of many conferences and workshops and a reviewer for many scientific journals and research funding agencies (the US National Science Foundation (NSF), the California MICRO program, the European Commission, and the Greek Secretariat of Research and Technology). He is a member of the IEEE.