A Self-Managed Scheme for Free Citywide Wi-Fi

Elias C. Efstathiou and George C. Polyzos
Dept. of Computer Science, Athens University of Economics and Business
{efstath, polyzos}@aueb.gr

Abstract

We present a self-managed scheme that could fuel the deployment of free public wireless networks in cities; we call it Peer-to-Peer Wireless Network Confederation (P2PWNC). Unlike existing approaches, P2PWNC does not rely on central planning but on an ad hoc community of broadband Internet subscribers (the *peers*) with Wi-Fi access points (APs). These APs provide wireless access to peers that are away from home but within the range of another P2PWNC AP. In the P2PWNC scheme, wireless service is provided to those peers who consistently provide service to passerby peers, based on an algorithm that detects non-simultaneous multi-way peer-to-peer exchanges. This *indirect reciprocity* algorithm runs in isolation on every peer AP, resists Sybil attacks, and promotes cooperation without relying on trusted authorities, certified identities, or tamperproof modules. In this paper, we discuss P2PWNC's design and show preliminary results that support its feasibility.

1. Introduction

Can we build wireless cities using a self-managed solution? That is the question we attempt to answer in this paper. By *wireless city* we mean any metropolitan area covered by wireless access points that allow its inhabitants to use the Internet for free. By *self-managed solution* we refer to a fully self-organized distributed system that does not rely on trusted authorities at any stage in its lifetime, and which can be created spontaneously. Node identities in this distributed system are free and are created locally, i.e. without relying on certification authorities: this permits fast system growth. Tamperproof modules cannot be assumed because this implies authorities; system protocols must therefore be *incentive compatible* if *rational* peers are to follow them.

Incentive mechanisms that promote fair use and cooperation in fully self-organized distributed systems have been proposed [1], but there are indications [2, 3] that fundamental problems like the Sybil attack [4] (using multiple identities in order to easily launch collusion-based attacks) do not permit straightforward implementations when an authority is absent.

We present a practical scheme that is simple to implement; the scheme's main goal is to exclude egregious free riders that wish to consume without contributing. We call it the *Peer-to-Peer Wireless Network Confederation* (P2PWNC) scheme; under P2PWNC, broadband subscribers with Wi-Fi access points (APs) are given incentives to keep their APs connected to the Internet and open for sharing with passersby. P2PWNC achieves this by *excluding* non-contributors from accessing P2PWNC APs. At the same time, when *contributors* request service from other P2PWNC APs, the APs provide access in order to increase their owners' standing and enable them to receive service in the future.

Several questions arise: how to bootstrap cooperation; how to detect free riders; how to keep P2PWNC protocols simple, encouraging their adoption by wireless clients and APs, and fuelling P2PWNC deployment. We answer these questions in Sections 2 and 3, where we describe the salient features of P2PWNC and its decentralized operation. In Section 4 we show simulation results that support P2PWNC's feasibility. Finally, we present related work in Section 5 and conclude in Section 6.

2. P2PWNC Overview

2.1. Peer Model

We assume all peers are rational and selfish in the game theoretic sense. They never engage in transactions and revelation of information unless there is a

This research is supported 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," which is co-financed by the Ministry of National Education and Religious Affairs of Greece and the European Union, through the program "EPEAEK II."

benefit in doing so; they may change identities or use multiple identities; peers may also collude. Any accounting system we design should be compatible with this model and the fact that no trusted authorities exist. Therefore, distributed storage schemes, micro-payments, or reputation schemes that rely on peer cooperation, centralized or distributed brokers, and certification authorities, cannot be used.

2.2. System Entities and Terminology

We assume each **peer** operates exactly one Access Point (**AP**) that is attached to his broadband Internet connection. A peer *consumes* when accessing the Internet through an AP belonging to another peer. A peer *contributes* when the peer's AP provides access to another peer. Each peer generates an identity for himself, which is a unique public-private key pair. Initially, only the peer and his AP know the public key (PK), however the PK is no secret, and, as we shall see, the AP reveals it to anyone within range. On the other hand, peers keep their private keys secret and use them to sign **receipts** when consuming resources. Peers include their PKs in the receipts they sign; this way, their signature can be verified directly. Receipts contain: 1) the PK of the providing peer, 2) the PK of the consuming peer, 3) a timestamp noting the start time of the wireless session, 4) a weight noting the total amount of Internet traffic forwarded, and 5) the peer's signature, i.e. a hash of the above encrypted with the peer's private key. (We will assume that the weight is always equal to 1, meaning that transactions consist solely of unit contributions, i.e. all sessions are equivalent, irrespective of duration, quality, or amount of traffic forwarded.)

2.3. The NWAY Acceptance Algorithm

We now describe an algorithm called NWAY that peer APs use to decide if they should provide service to a requesting peer who is within range. We describe here a centralized version for ease of exposition; decentralized NWAY is presented in Section 3. For now, assume a Trusted Server (TS) that stores the community's history: every time a receipt is generated, TS keeps a copy forever. NWAY states: *when peer* C *requests service from peer* P*,* P *searches for a chain of receipts connecting it to* C*. If a chain exists, service is provided and a new receipt is created; if not,* C*'s request is denied.* (Identifiers are the PKs of entities.)

Imagine receipts to be the edges of a directed graph with peer identities the vertices; receipts point from the consuming peer to the providing peer. Every receipt has a unique label (the field combination {provider PK, consumer PK, timestamp} is unique), and we allow for multiple edges connecting two vertices in either direction because peers may interact repeatedly. A chain of receipts starting from *P* and ending at *C* indicates that *P* has directly or indirectly consumed from *C* in the past. If *P* now provides service to *C*, a new *C*→*P* receipt will be generated as a result and we say that a *non-simultaneous n-way exchange is completed*. For example, if TS contained a *P*→*C* receipt, and if *C* were to request service from *P*, *P* would offer service and complete a non-simultaneous 2-way exchange. If TS contained *P*→*X* and *X*→*C*, a 3-way exchange would be completed instead.

NWAY further states: *if* P *provides service,* P *must then discard all receipts in the discovered chain*, meaning that in *P*'s future decisions *P* must act as if TS did not contain the receipts that *P* has discarded.

Fairness and incentive analysis. P *cannot trust any receipt unless he signed it himself*. That is the rationale of NWAY, a consequence of inter-peer distrust and free identities. No matter how many receipts peer *C* produces to convince *P* that *C* is a good provider, *P* knows that the total cost of generating such receipts, *unless* P *has signed one of them*, might as well be zero. It would be trivial for *C* to generate peer identities and sign fake receipts that show *C* having provided service, or to achieve the equivalent in collusion with existing peers.

One might suggest that *P* can only be sure if he detects a potential 2-way exchange, i.e. a *P*→*C* receipt. However, *asymmetric* interactions could be common in P2PWNC: peer *A* could consume from peer *B* and have no chance to repay peer *B*. Searching for *generalized* (n-*way*) *exchanges* has more chances of success because *P* only requires to be connected to *C* through a *chain* of receipts. Every receipt in this chain can be verified using information contained in the chain itself, i.e. there is no need for a Public Key Infrastructure. *P* verifies its signature on the first receipt and uses the PK of the providing peer in that receipt to verify the next receipt, and so on. Even if all peer identities appearing in this chain (except *P*) are *C*'s aliases, *P* still knows that it owes one unit to *C*, irrespective of *C*'s "real" identity. If some receipts are the result of collusion, *P* still knows that it owes one unit to the colluding group. As long as *P* discards all receipts in the chain, *P* will never have to contribute more than it consumes, and no free riding peer will be able to consume from *P* unless that peer colludes with a peer from which *P* consumed indirectly or directly. Even then, the colluding group cannot achieve net gain because *P* will discard all receipts in the chain.

To analyze incentives in NWAY we look at the receipt graph again. Effectively, *C* is using a *tree* of

receipts rooted at *C* as proof of good standing. The tree may or may not contain *P* at one of its levels. Assuming complete inter-peer distrust, the only tree that, from *P*'s point of view, could not have been produced at zero cost is a tree that also contains *P* (see Fig. 5). Our results in Section 4 show that the probability of *P* detecting itself somewhere in *C*'s tree is high if *C* is a consistent contributor. Assuming that *P* always detects contributors reliably and *C* is such a contributor, *P* has an incentive to provide service to *C* because then *C*'s tree will become part of *P*'s tree, owing to the new $C \rightarrow P$ receipt that will connect them: *C* gives *P* the right to consume (once) from *C* and *from anywhere that C could consume from*.

Note here that NWAY requires that *P*'s AP should ignore discarded receipts irrespective of the identity of future requestors. This is again to guard against pseudo-spoofing; otherwise, a cheating *X* could generate multiple receipts of the form $\{X \rightarrow X_1, X \rightarrow X_2, X \rightarrow X_3, ...\}$ and exploit one contribution ($P \rightarrow X$) ad infinitum by assuming one of its X_n aliases.

3. Decentralized Operation

3.1. Receipt Repositories

Each peer AP maintains 4 receipt repositories: *incoming* (IR), *outgoing* (OR), *random* (RR), and *discarded* (DR). IR contains receipts where the peer was the provider; OR contains receipts that have been signed by the peer, i.e. where the peer was the consumer; RR contains receipts that encode transactions between other peers; DR contains receipts that this AP has discarded. The repositories hold up to s_{IR}, s_{OR}, s_{RR}, and s_{DR} entries.

3.2. Decentralized NWAY

We focus on a peer *C* who is requesting service from the AP belonging to peer *P*. In what follows, we will use the identifier *P* to mean *P*'s AP.

Step 1: Searching for a chain. Without access to all P2PWNC receipts, the best *P* can do is to search for a chain of receipts in *P*'s and *C*'s combined repositories. Specifically, combining OR_P, RR_P, RR_C, and IR_C would be enough, as the remaining repositories of *P* and *C* do not add information useful to this search. It is in the interest of *C* to carry up-to-date copies of IR_C and RR_C, and to show *P all* receipts therein because *C* cannot know which receipts *P* has already discarded. Requiring that *C* carry copies of RR_C and IR_C might sound burdensome, especially if *C* (when away from home) uses a lightweight device such as a WLAN-enabled mobile phone. However, a receipt only contains two public keys, a signature, a timestamp, and weight. If we allow 10 bytes for the timestamp and weight, and use Elliptic Curve Cryptography, this reduces to 2 x 20 + 1 x 40 + 10 = 90 bytes. We do not require that *C* has the most current version of IR_C and RR_C, only a recent one. To obtain this file, peers may opportunistically contact their APs over the Internet, e.g. at the end of their previous P2PWNC session or even using the cellular system. This is possible because we assumed the APs are always connected to the Internet. In our evaluation we show that repository sizes in the order of 100 are reasonable.

Step 2: Discarding receipts. If *P* detects a chain, *C* is admitted and *P* discards the receipts in the chain by storing their unique hashes in DR_P.

Step 3: Updating Time Horizon. If DR_P overflows, the entry corresponding to the receipt with the oldest timestamp is deleted. Because this way the AP will forget which receipts are discarded and should be ignored, it sets a Time Horizon (TH) variable equal to the timestamp of the receipt whose entry was just evicted from DR_P. APs consult their THs when searching for chains in order to make sure they are not considering discarded receipts, at the cost of potentially ignoring some non-discarded receipts.

Step 4: Consumer and provider store new receipt. If *C* is admitted, *P* stores in IR_P the new $C \rightarrow P$ receipt; *C* must then send the same receipt to his AP to be stored in OR_C. This can happen opportunistically, e.g., whenever *C* receives updates of IR_C and RR_C, *C* can also send any unreported outgoing receipts.

Step 5: Gossip. *P* updates RR_P with the random (from *P*'s point of view) receipts that *C* presented in IR_C and RR_C. We assume that the more recent a receipt is the more valuable it is, and this is the replacement rule we use when RR (and the other repositories) overflow. The intuition here is that the more recent a receipt is, the less time it had to circulate through the system and therefore the smaller the probability of it having being discarded by other peers. Peers are therefore encouraged to contribute continually in order to refresh their repositories with receipts that are newer than most time horizons.

Bootstrap. Peer APs cannot always follow NWAY. In the beginning of his lifetime, peer *P* does not have outgoing receipts. However, NWAY requires that *P* must search for *n*-way exchanges that, by definition, include an outgoing receipt. To avoid this deadlock we define p_{NWAY}, the probability that *P*'s AP obeys the NWAY admission rule. In the beginning of *P*'s lifetime, $p_{NWAY} = 0$ and *P*'s AP accepts visitors without requiring that a chain is detected in their combined receipt repositories. p_{NWAY} is then opportunistically updated using an additive increase, multiplicative decrease rule: every time *P* is successfully admitted,

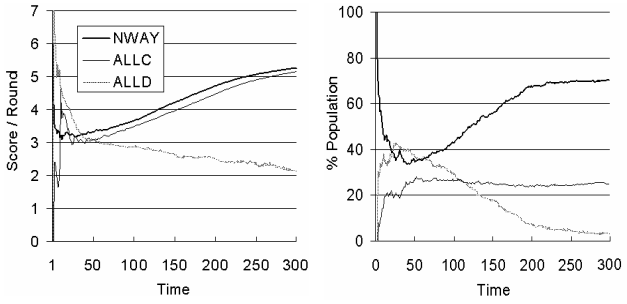
p_{NWAY} increases. If a foreign AP actively denies access to P, p_{NWAY} is decreased. The intuition behind p_{NWAY} is this: P is sure of its good standing when he is being accepted by other peers; P's failure to login if he is already "well-known" must be a rare event but it *could* happen as a result of several factors; one factor is the exclusion mistakes P's AP made because of its limited view of the receipt graph, mistakes that denied it perfectly valuable new receipts. These mistakes in turn caused other peers to mistake P for a free rider when P found himself without the necessary receipts. P's AP can mitigate this by occasionally becoming less strict.

4. Evaluation

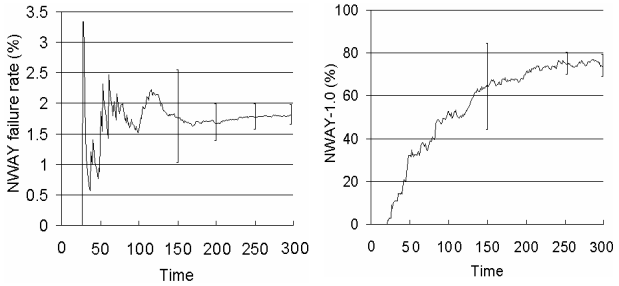
We base our simulations on the evolutionary framework of [1]. We assume that peers are randomly paired for games. In every *round*, each peer gets one chance to contribute (and lose $c = 1$ points if he does so), and one chance to consume (and gain $b = 7$ points if the other peer cooperates). (The values for these two parameters are taken from [1]. They indicate the relative benefit and cost of obtaining and supplying one "unit" of wireless access. Results are qualitatively the same for a wide range of b/c ratios.) We also simulate the growth of P2PWNC: in the beginning of time there are only 2 peers and at the end of each round a new peer joins. Peers never leave the system.

Every peer starts by following a *strategy*, but may change its strategy (evolve) at the end of each round with probability $p_{learn} = 0.05$. He will then pick the strategy (from the ones available) that currently has the highest *rating*, and will adopt it with probability proportional to the difference in rating between that strategy and its own. A strategy's rating is the average of the running averages of scores per round of its followers, with each term weighted according to how many rounds a peer has been using the strategy [1]. Our remaining parameters are the repository sizes (here, $s_{IR} = s_{RR} = s_{DR} = 100$ and $s_{OR} = 400$), and the additive increase and multiplicative decrease parameters of p_{NWAY} (0.05, 0.5 respectively). Time is measured in rounds as defined above.

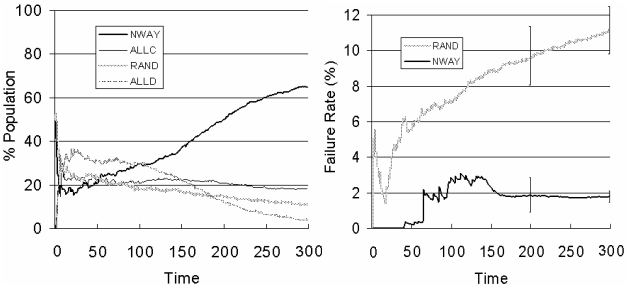
Experiment A. We let the NWAY strategy with the p_{NWAY} extension face ALLD (unconditional defectors) and ALLC (unconditional cooperators). Each new peer that joins the game initially follows one of these three strategies with equal probability (=1/3). Note that we define ALLD and ALLC as simple variants of NWAY: their followers also run the decentralized NWAY algorithm, but ALLD followers keep their APs disconnected and only attempt to consume, and ALLC followers cooperate irrespective of what the algorithm outputs. In Fig. 1 we see that the NWAY strategy is evolutionarily successful in "fighting" ALLD behavior, but not ALLC behavior. ALLC followers, in a sense, "free ride" on the "efforts" of NWAY to punish ALLD. NWAY performs better than ALLC though, but both strategies achieve scores near the maximum per round score of 6 (i.e. cooperation is established – unconditional defectors are persuaded to change their strategy). In Fig. 1b we see that as time progresses, a mixture of NWAY and ALLC is created, with NWAY in the majority. However, since ALLC followers face no real threat from ALLD followers (owing to the "efforts" of NWAY followers), ALLC persists also. In Fig. 2a we plot the average rate of NWAY *failures* (confidence interval 95%); failures occur when an NWAY follower is denied access by another NWAY follower that mistook him for a free rider. This is limited to less than 2% of an NWAY follower's requests. A peer's discarded receipts and advancing time horizon can annul another peer's contributions in the eyes of this peer, and cause these mistakes. Larger IR and RR repositories help as peer populations become large. (Repositories only need to grow very slowly relative to populations, owing to the birthday paradox, but we overlook this part of our argument for lack of space.)



Figs. 1a, 1b: NWAY against ALLC/ALLD (experiment A).



Figs. 2a, 2b: NWAY failure rate and percentage of NWAY followers having $p_{NWAY} = 1.0$ in the NWAY/ALLC/ALLD mix of experiment A.



Figs. 3a, 3b: NWAY against ALLC/ALLD/RAND (experiment B).

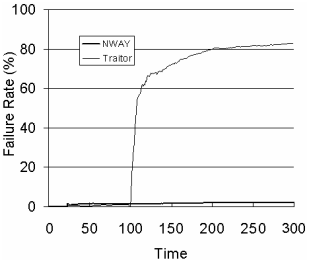
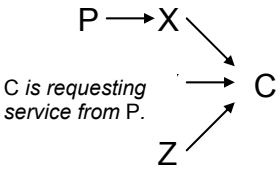


Fig. 4: Punishing treason.



This tree presents a potential 3-way exchange between them.

Fig. 5: Receipt tree rooted at *C*.

In Fig. 2b, we plot the percentage of NWAY followers that have $p_{NWAY} = 1$. We can see that even with the p_{NWAY} extensions, NWAY followers remain strict most of the time: some free riders persist but their level of service remains far from the 98% enjoyed by NWAY and ALLC followers.

Experiment B. We let the NWAY strategy face ALLD, ALLC, and RAND, a strategy that plays either ALLC or ALLD with equal probability (=0.5) when a peer requests service. Each peer that joins P2PWNC picks one of these four strategies with equal probability (=1/4). NWAY drives away its opponents (Fig. 3a). In Fig. 3b we see that extravagance has its cost in P2PWNC: RAND is a strategy that tries to consume twice as much as it contributes; its followers do not refresh their repositories quickly enough and the result is that their failure rates approach 11%

Experiment C. The list of competing strategies that NWAY needs to be robust against is infinite. As a last experiment, we let NWAY face a *traitor*, i.e. an NWAY follower that turned ALLD at round 100 and stopped providing. P2PWNC effectively detects this traitor in only a few rounds (Fig. 4), effectively giving the incentives to peers to contribute continually.

5. Related Work

Other proposals are also addressing the problem of fuelling wireless network deployment. The work in [5] presents a framework that motivates Wireless ISPs to provide access to each other's users by using a reputation mechanism that is maintained by a Trusted Central Authority. Major cities [6] are also considering their own centralized schemes, but no winning business model has yet emerged. Several P2P systems provide incentives for resource sharing through accounting: e.g., *PPay* [7] (micropayment scheme, requires a centralized broker); *Karma* [8] (DHT-based accounting, susceptible to the Sybil attack: the cryptographic puzzle that new entrants need to solve only limits the *rate* of identity generation); and the *Nuglets* approach [9] (cooperation in ad hoc networks, relies on tamperproof modules). The work most closely related to ours is on *n*-way exchange-based incentive mechanisms for file sharing [10].

6. Conclusions

The IEEE 802.11 WLAN protocols, for the first time in telecom history, allow individuals to provide telecom services to their peers. We propose a self-managed approach to citywide Wi-Fi, which provides appropriate cooperation incentives by excluding free riders. We see the P2PWNC scheme as a viable wireless alternative for urban areas that are already well served by (fixed) broadband: P2PWNC simply combines existing under-exploited Wireless LANs and unites them in a roaming federation. Our ongoing work includes the implementation of a P2PWNC AP on top of the Linux-based Linksys WRT54GS AP.

References

[1] Feldman, M., Lai, K., Stoica, I., and Chuang, J. Robust Incentive Techniques for Peer-to-Peer Networks. Proc. ACM EC'04.

[2] Huang, E., Crowcroft, J., and Wassel, I. Rethinking Incentives for Mobile Ad Hoc Networks. Proc. SIGCOMM PINS Workshop (2004).

[3] Mahajan, R., Rodrig, M., Wetherall, D., and Zahorjan, J. Experiences Applying Game Theory to System Design. Proc. SIGCOMM PINS Workshop (2004).

[4] Douceur, J. The Sybil Attack. Proc. IPTPS'02.

[5] Ben Salem, N., Hubaux, J.-P., and Jakobsson, M. Reputation-based Wi-Fi Deployment: Protocols and Security Analysis. Proc. ACM WMASH'04.

[6] Wireless Philadelphia. http://www.phila.gov/wireless/

[7] Yang, B., and Garcia-Molina, H. PPay: Micropayments for Peer-to-Peer Systems. Proc. ACM CCS'03.

[8]Vishnumurthy, V., Chandrakumar, S., Sirer, E.G. KARMA: A Secure Economic Framework for P2P Resource Sharing. Proc. p2pecon'03.

[9]Buttyan, L., and Hubaux, J.-P. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. In *ACM/Kluwer Mobile Networks and Applications*, 8(5), (2003).

[10]Anagnostakis, K. G., and Greenwald, M. B. Exchange-based Incentive Mechanisms for Peer-to-Peer File Sharing. Proc. ICDCS'04.