Reducing Management Complexity through Pure Exchange Economies: A Prototype System for Next Generation Wireless/Mobile Network Operators*

P. A. Frangoudis, E. C. Efstathiou, and G. C. Polyzos
Department of Computer Science
Athens University of Ecomomics and Business
Patision 76
Athens 10434
Greece
{pfrag, efstath, polyzos}@aueb.gr

Abstract

We present the design and implementation of a protocol for supporting Wireless LAN (WLAN) roaming federations. Unlike existing approaches, which incur significant management overhead, our protocol is designed specifically with reducing management complexity in mind. More specifically, the next generation operators that oversee our WLAN federations are not involved in WLAN provisioning decisions. Rather, the WLANs that participate in the federation form a pure exchange economy in which WLAN resources are constantly being traded. By simply enabling a reputation-based incentive mechanism, the operator allows cooperation to evolve within the federation, and can benefit indirectly from the process.

Keywords

Wireless LAN, roaming, peer-to-peer, next generation operators

1. Introduction

Today, mobile phones are being equipped with Wireless LAN (WLAN) transceivers that allow them to operate in unlicensed frequency bands in order to access data at broadband speeds [1, 2]. This trend is likely to increase as the price of WLAN chipsets continues to drop, and as the chipsets themselves become more power-

* 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."

efficient. Mobile Network Operators (MNOs) are already expanding their networks using inexpensive WLAN base stations in public venues such as hotels and airports. The limited range of a WLAN makes wide-area WLAN coverage costly – i.e., the *number* of base stations required to cover a given area offsets the low cost of WLAN base stations. This is one reason why MNOs prefer to partner with established public WLAN operators in, so called, *roaming federations*. Still, today, even successful public WLAN operators provide only limited area coverage, as they mainly concentrate on high-density public venues such as the ones mentioned above.

A new business model that can overcome this coverage limitation is emerging: certain next generation network operators [3, 4] seek to take advantage of the *residential* WLANs that are already deployed in cities; effectively, these network operators partner with households that operate home WLANs.

The number of residential WLANs in urban areas is increasing. Indeed, a U.S. company is already promoting a product [5] that relies on *beacon signals*, which are emitted by WLAN base stations, to provide positioning information. This service is directly competing with the Global Positioning System in 25 major U.S. cities.

Most residential WLANs are connected to the Internet through always-on broadband connections, and, oftentimes, a single WLAN can cover substantial area surrounding the owner's residence. Anyone within range of such a WLAN can, in principle, access the Internet and other public networks at high speed. In the new business model mentioned above, the operator's micro-partners are compensated for their contribution to the public in various ways. However, by treating each home WLAN owner as a separate roaming partner, the next generation operators face significant management overhead. This is probably one of the reasons this business model has enjoyed limited success so far. Joining such a roaming scheme can also be problematic for the average WLAN-operating household because it involves various economic, legal, and technical hurdles.

In this paper we propose a new management paradigm for next generation network operators that wish to federate with numerous smaller players. Our model relies on a *pure exchange economy* that naturally promotes cooperation within a loose federation of WLAN providers. The prototype system that we have developed demonstrates that operators can adopt our proposal with minimal financial risk, using technological components that are readily available.

2. Model Features

Two principles underlie our proposed federation scheme:

(1) *"Relaxed" accounting*: instead of accounting resource consumption and resource contribution in a precise manner, we allow cooperation between providers and consumers to evolve naturally by tying consumption to contribution in a simple *reciprocity* scheme. Our main focus is on discouraging egregious *free riding*, and not on supporting a completely accurate accounting and settlement system.

(2) *Peer symmetry and peer autonomy*. The federation unites providers of resources (i.e. owners of WLAN base stations) and consumers of resources. Similarly to electronically mediated *peer-to-peer* (P2P) communities, each micro-partner in the federation is assumed to be both a consumer and a provider of WLAN resources; a micro-partner may contribute lots of resources, or contribute nothing at all. Only a simple *receipt-based accounting scheme* exposes free riders, which can then be punished (by being excluded from the resource-sharing scheme) by the partners who *do* contribute resources.

Our proposed peer-to-peer design involves a *Trusted Central Authority* (TCA). In terms of our model (i.e. the next generation operator case), the micro-partners (the home WLAN owners) are the peers, and the operator assumes the role of the TCA. The TCA modules, however, are designed to be lightweight and with minimal functionality.

The assumption of our roaming federation scheme is that all peers subscribe to the TCA for standard cellular service, but only those peers who *provide* WLAN resources can also *consume* WLAN resources from other peers. The TCA acts as a trusted information repository and *WLAN service provision decisions are left to the peers* themselves (this is an example of the autonomy that WLAN-sharing peers enjoy when it comes to the WLAN service). Our thesis is that operators can provide wide-area broadband services (albeit without service guarantees) at a minimal cost, and increase their subscriber base in the process. Although operators will not be able to charge for WLAN service directly, they still control federation membership as well as the central repository of information, which serves as a foundation of a *reputation-based incentive mechanism* that guides the micro-partners in their contribution decisions.

3. System Architecture

In our design, the participating system entities are (1) the *roamers*, who consume WLAN resources; (2) the *providers*, who provide WLAN resources; and (3) the *Trusted Central Authority* (TCA), which maintains a repository of accounting information (in the form of digitally signed *receipts* – see below). Providers are organized in small *teams*. Teams may be thought as being the *peers* in a P2P community of teams, where the consumers of resources must also provide resources to others. When a member of a team is roaming and requests service from a WLAN operated by a member of a different team, she must first prove that the team she belongs to has provided (the required amount of) service to the entire community. Teams, in general, provide service by operating a number of WLAN base stations. A team can be as small as a household with a single WLAN base station, or perhaps be a group of neighbors who agree to pool their base stations in order to increase the WLAN area covered by their team, and, consequently, the frequency with which they can earn “cooperation points,” i.e. positive reputation as providers.

Teams are useful in this respect: the notion of teams helps include providers who *are* willing to contribute but rarely get the chance to serve (e.g. they live in the outskirts of a city). This must happen, however, at the expense of the more altruistic team members (their friends and neighbors) who are willing to provide more resources so that their (willing, but incapable) teammates may earn the right to consume.

Each team is simply identified by a cryptographic public/private key pair. The TCA is responsible for issuing these keys. Each team then generates key pairs for its members, and uses its secret key to sign *member certificates* for them. Member certificates are of the following form:

Member cert = {*Team public key*, *Member public key*, *Team signature*}

The TCA does not need to know the number *or* the real-world identities of team members. Its role is limited to recording the transactions between teams, but it makes no decisions concerning WLAN provision; these decisions are left to the peers themselves. The TCA simply provides relevant information to teams who request it.

A *transaction* is a session during which a team provides resources (here, WLAN access) to a member of another team. Accounting of sessions is done through the issuing of *receipts*, which roamers must sign. Receipts have the following format:

Receipt = {*Roamer member certificate*, *Providing team public key*, *Timestamp*, *Weight*, *Roamer signature*}

The receipt *timestamp* indicates the exact time a transaction started, and the receipt *weight* is the amount of traffic uploaded and downloaded by the roamer during the transaction. Roamers digitally sign receipts with their private key. This way, the authenticity of the receipt can be verified. During a session, the provider periodically asks the roamer for a receipt, which forces the roamer to acknowledge that she has consumed WLAN resources. If the roamer refuses to sign a valid receipt, the provider can terminate the session. At the end of a WLAN session, the provider forwards the receipt to the TCA. The TCA verifies the receipt and can also update a record of the amount of service each team has provided (taking into account the weight of the reported receipt). The TCA can mediate between teams, answering queries regarding the amount of service that a specific team has provided. The TCA therefore helps a querying team to decide (using a team-specific *decision function*) whether or not the current roaming visitor comes from a team with good standing and should thus be provided access (providing access raises the reputation of the providing team).

At this point we can sum up the basic system assumptions: First, we assume the

TCA is fully trusted by all teams. Second, we assume full intra-team trust (trust among members of the same team). Third, we assume that all digital certificates and receipts are protected by simple cryptographic primitives and we assume that it is computationally infeasible to break the encryption scheme and generate a fake signature without access to the relevant private key.

4. Prototype Implementation

In terms of our implemented prototype, relevant entities include the *devices of mobile users* (WLAN-enabled phones, PDAs, and laptops), the *WLAN base stations* operated by the teams, and the *TCA* (which can be hosted on a single server or be distributed for scalability and fault-tolerance reasons).

We have built a complete prototype following the design we described above. The TCA is hosted on a server running Linux. The TCA is responsible for cryptographic operations such as key generation and receipt verification. Also, and more importantly, it contains a repository, where team identities (key pairs) and traffic statistics per team are kept, including the associated stored receipts. Our TCA implementation is written in C and relies on the OpenSSL cryptographic library.

All our provider modules run on top of the Linux-based Linksys WRT54GS WLAN access point (currently retailing for less than $70). That is, all provider functions run on the WLAN base station itself. For each WLAN session, the base station maintains state information, which includes the session start time, the public key of the roamer, and the current traffic volume forwarded for the roamer. A kernel-level traffic-measuring module has been developed that can accurately measure roamer traffic. This module interoperates with the Linux traffic control module and with the Linux firewall to ensure that no unauthorized roamers are granted access. All our embedded software has been implemented in C. We again rely on the OpenSSL cryptographic library for the relevant functionality.

Our *federation protocol* (see Section 6) requires a series of message exchanges between the roamer and the providing team's base station before a session is allowed to start. Also, roamers may or may not make use of modules on their devices that measure forwarded traffic in order to avoid any possibility of being cheated by base stations that wish to overcharge (i.e., ask for receipts with more weight than what was actually offered). We have implemented two versions of the roamer software, one in C and one in Java, and we also plan to port it to smartphone platforms.

All cryptographic operations use either *RSA* or *Elliptic Curve Cryptography* (ECC) functions. The advantage of ECC over conventional RSA is that it provides the same level of security at considerably smaller key lengths. This is important for memory- and bandwidth-constrained devices such as the mobile phones and the WLAN base stations that represent the main components of our system.

5. The Receipt Repository

The repository of system receipts should provide support for several types of decision functions that teams may choose to use. Thus, there is a need for a versatile data structure that will be suitable for operations such as fast receipt lookup, receipt range queries, receipt insertion, receipt deletion, and various types of graph operations. The main data structure on which most decision functions rely is the *receipt graph* (where nodes represent teams and directed edges represent receipts that connect them).

To implement the above, a composite data structure made up of a *red black tree*, *hash tables*, and an *adjacency list* is used. In order to reduce memory overhead, there is only one instance of each receipt in memory, which is shared by all our data structures (via pointers).

Each receipt is a node of a red black tree. Red black trees are (almost) balanced binary search trees [6]. Their height is at most $2log(n + 1)$, where n is the number of the tree nodes (the receipts, in our case). The key of each node is the receipt's timestamp (see Section 3 for the format of receipts). Deletion, insertion and searching for a receipt all require logarithmic time on the number of nodes. This is important, since one of the criteria for receipt validity is their timestamp. In the TCA repository, old receipts are constantly being replaced by new ones. Therefore, locating the receipt with the oldest timestamp and replacing it with a new one has to happen fast. What is more, this data structure improves the efficiency of range query operations. For example, the receipt repository may be required to return the receipts whose timestamp is greater than a specified value. Theoretically, this requires linear time. In practice though, since receipts are ordered, and because there will be a tendency for clients to request only recent receipts, this operation is performed much faster using a red black tree.

To avoid storing duplicate receipts, there is a hash table with pointers to the receipts already stored. Each receipt is uniquely identified by the tuple <*consumer certificate*, *provider public key*, *timestamp*>. In the case where a new receipt is to be inserted, a fast lookup is performed using this hash table to ensure that the receipt with the same identifier is not already in the repository.

An important family of decision functions that provide the right incentives for cooperation require running the *maximum flow* (maxflow) algorithm on the receipt graph. To run maxflow efficiently, there is a need for an additional structure, since the ones introduced before are not suitable for graph operations. Since the set of receipts is dynamically updated, there are pointers from the red black tree nodes to the nodes of the adjacency list representing graph edges. This helps in synchronizing the two data structures in constant time every time a modification in the repository takes place. This data structure is a weighted directed graph. Its nodes represent teams and its edges represent service consumption. Namely, an edge from node A to node B indicates that B has provided service to A. The weight of the edge from A to B represents the sum of the weights of all receipts issued from team A to team B.

To calculate the maximum flow from a node to another, the FIFO variant of the push-relabel algorithm [7] is used. The time complexity of this method is $O(V^3)$, where V is the number of graph nodes. However, this has proved a poor upper bound for our application. We implemented a well-known heuristic, which yielded dramatically better results. According to this heuristic, a breadth-first search from the terminal node to the source is performed periodically in order to update the estimate of the distance to the terminal node that all other nodes have. This operation is rather time consuming, since its running time is $O(V+E)$. However, it is not performed frequently (in our implementation, it is carried out every V relabel operations).

Experiments on the efficiency of our implementation of the above data structures and the maximum flow algorithm have been carried out. Since our target platforms are embedded devices (WLAN base stations), the testbed for our measurements included the Linksys WRT54GS embedded Linux-based 802.11g access point mentioned above, and an AMD Athlon XP 2800 laptop. The exact specifications of the two platforms are shown below.

Table 1: Characteristics of the platforms used in the maxflow experiments

	Linksys WRT54GS	***AMD Athlon XP 2800***
CPU	*MIPS 200MHz*	*x86 2.08GHz*
Main memory	*32MB*	*512MB*
Operating system	*Linux 2.4 kernel*	*Linux 2.4 kernel*

The inputs for the maximum flow algorithm were randomly generated graphs and random provider-consumer pairs. The experiments were executed on both platforms with identical inputs. We measured wall time with the use of the *gettimeofday()* C function. During the execution of the algorithm, no network or I/O operations were taking place.

In the diagrams that follow (Figures 1 and 2), three curves have been plotted, for graphs of 100, 500 and 1000 nodes (teams) respectively. Each curve consists of 100 data points. The x-coordinate of a data point indicates the size of the receipt repository and its y-coordinate indicates the time in microseconds spent on an execution of the maximum flow algorithm. Values on the x-axis start from 100 receipts, going up to 10 000 receipts, with a step of 100 receipts.

The value of a data point's y-coordinate is the average of the execution time of the maximum flow algorithm on 20 random pairs of nodes (a node pair represents a consumer-provider pair) of the same graph. All the experiments were performed three times and each of the 20 values that yield a data point represents the average of the time spent on the three executions of the same experiment.

As one might expect, given that the Linksys WRT54GS processor is an order of magnitude slower, the execution of the maxflow algorithm is about an order of

magnitude faster on the PC. Despite this performance drawback, though, the Linksys platform seems powerful enough to be able to support a locally-cached receipt repository, in order to perform receipt operations on it without having to consult the TCA every time it needs to execute the decision function.

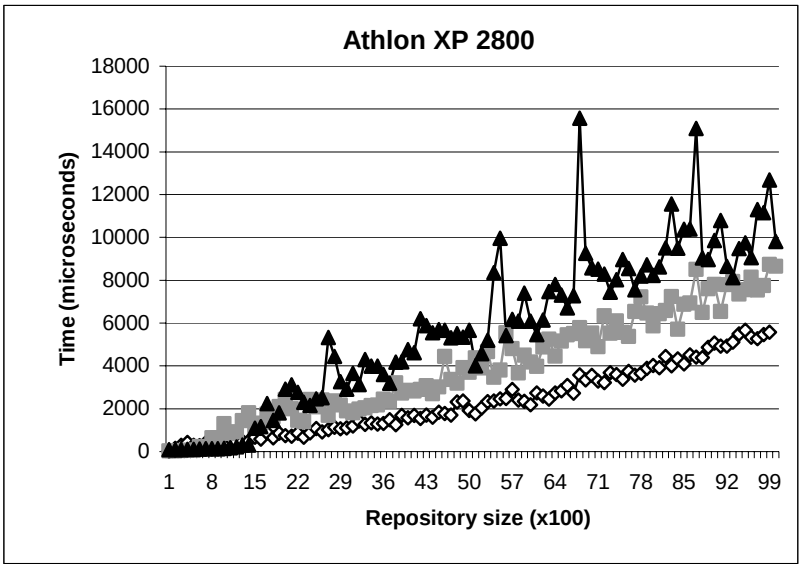

Figure 1: Maxflow performance on a typical PC

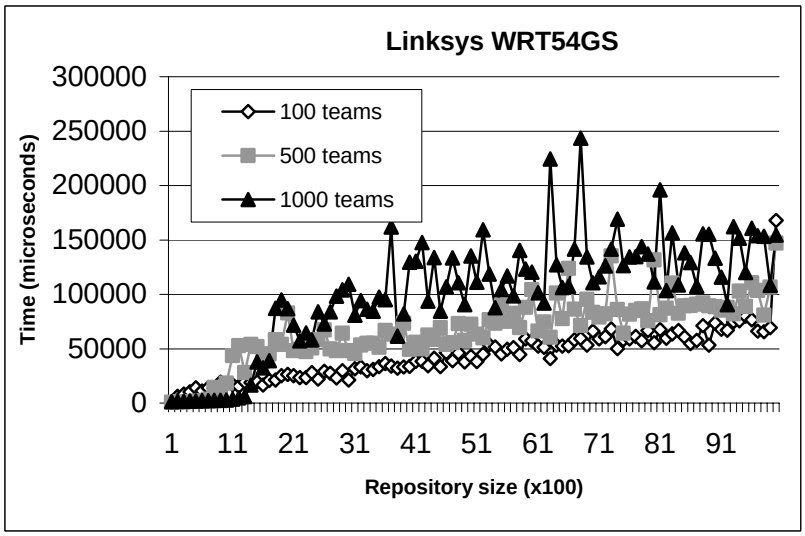

Figure 2: Maxflow performance on the Linksys WRT54GS

6. The Federation Protocol

What follows is a list of the messages involved in the federation protocol that we propose, as well as a short description of each one.

CONN Such a message initiates a session between a roamer and a visited access point. It is sent by the roamer and contains her certificate, so that the visited access point can verify the roamer is indeed a member of the team that issued the certificate.

CACK This message indicates that a roamer has been given access to a visited wireless network. Normally, it is the access point's response to a CONN message. The CACK response includes the timestamp of the particular session, i.e. the time that the session starts. All subsequent receipts that a client is to sent to the access point (as "payment" for service provided) during the session include this timestamp.

RREQ Periodically, the access point module requests that the roamer acknowledges that she has consumed a specific amount of resources during the session. This request is performed by means of the RREQ message. This message includes the traffic that the client has initiated during the session, measured in bytes, and the provider team's public key. The roamer must immediately reply with an RCPT message, which is described below.

RCPT An RCPT message is the basic unit of information concerning service provisioning in our system. It represents a transaction between a roamer and the access point of another (service providing) team. An RCPT message contains the roamer's certificate (consisting of the roamer's team public key, the roaming member's public key, and the team's signature), the service providing team's public key, the amount of traffic initiated by the mobile user during the session (the "weight" of the receipt, measured in bytes) and the session's timestamp. All the above are digitally signed by the roaming member using her private key. This signature is included in the RCPT message, thus acknowledging service consumption on behalf of the client. It should be noted that, during each session, many receipts will be generated. All these receipts have the same timestamp (equal to the time the session started) but have increasing weights, since they are the client responses to the periodic RREQ messages sent by the access point. Eventually, only the last receipt of a session is significant for the system, since it summarizes the amount of service the provider has offered to a roamer during the session in question.

QUER This message is used by an access point to inquire whether access should be granted to a visiting roaming user. It contains the public key of the provider team and the public key of the team to which the roamer belongs. QUER messages are in general sent to the TCA.

QRSP QRSP is the reply to QUER. It is issued after the TCA has processed the QUER message. Its header field "Action", which can take the values "Grant" or "Forbid" indicates the outcome of the decision function the TCA applied on behalf of the client.

7. Related Work and Conclusions

The economics of the peering model we presented in this paper have been analyzed before in [8, 9]. A comparison to existing roaming schemes appears in [10]. A *decentralized* version of the peering scheme that does not rely on a trusted authority appears in [11]. In conclusion, we have designed and prototyped a resource-sharing federation that is supervised by a single trusted authority, which acts solely as information repository and delegates all provisioning decisions to the resource-sharing peers themselves. Our scheme reduces management complexity by relying on a simple reciprocity-based incentive mechanism, which allows cooperation within the federation to evolve naturally, without the overhead associated with centralized decisions. This "best-effort" model may be appropriate for providing gcertain non-critical resources, such as, for example, WLAN-based broadband wireless access.

References

[1] Motorola CN620. http://www.motorola.com/wlan/solution_cn620.html

[2] Nokia 9500 Communicator. http://www.nokia.com/nokia/0,,54106,00.html

[3] Speakeasy WiFi NetShare Service. http://www.speakeasy.net/netshare/

[4] Linspot. http://www.linspot.com/businessmodel.html

[5] Skyhook Wireless Wi-Fi Positioning System. http://www.skyhookwireless.com

[6] L. J. Guibas and R. Sedgewick. "A Dichromatic Framework for Balanced Trees," Proc. of 19th Annual Symposium on Foundations of Computer Science, 1978.

[7] A. V. Goldberg and R. E. Tarjan, "A New Approach to the Maximum Flow Problem," *J. Assoc. Comput. Mach.*, vol. 35, 1988.

[8] P. Antoniadis, C. Courcoubetis, E. C. Efstathiou, G. C. Polyzos, and B. Strulo, "Peer-to-Peer Wireless LAN Consortia: Economic Modeling and Architecture," Proc. 3rd IEEE International Conference on Peer-to-Peer Computing, Linkoping, Sweden, Sept. 2003.

[9] C. Courcoubetis and R. Weber, "Asymptotics for Provisioning Problems of Peering Wireless LANs with a Large Number of Participants," Proc. of 2nd Workshop on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, Cambridge, UK, March 2004.

[10] E. C. Efstathiou and G. C. Polyzos, "A Peer-to-Peer Approach to Wireless LAN Roaming," Proc 1st ACM International Workshop on Wireless Mobile Applications and Services on Wireless LAN Hotspots, San Diego, CA, Sept. 2003.

[11] E. C. Efstathiou and G. C. Polyzos, "Self-Organized Peering of Wireless LAN Hotspots," *European Transactions on Telecommunications*, vol. 16, no. 5 (special issue on Self-Organization in Mobile Networking), Oct. 2005.