Experimental evaluation of community-based WLAN voice and data services

Pantelis A. Frangoudis[*] Vasileios P. Kemerlis[†] Dimitrios C. Paraskevaidis[*]
Elias C. Efstathiou[*] George C. Polyzos[*]
Mobile Multimedia Laboratory
Department of Informatics/Computer Science
Athens University of Economics and Business
47A Evelpidon Street, GR-11362 Athens, Greece
{pfrag, dcp, efstath, polyzos}@aueb.gr[*] vpk@cs.aueb.gr [†]

ABSTRACT

The purpose of this work is to experimentally evaluate the quality of voice and data services over community-based WLAN access networks. We use P2PWNC, the Wireless LAN roaming architecture that we have developed, as the basis for the provision of ubiquitous citywide WLAN access and set up an infrastructure for secure voice and data communications on top of it. Our scheme was designed with widely available, low-cost WLAN equipment in mind. Thus, we wish to estimate the performance penalty our proposed mechanisms incur for such devices and study their limitations as to the support for secure data and multimedia communications. In this work, we measure the maximum number of simultaneous voice calls that a P2PWNC-enabled access point can sustain, and the TCP throughput achieved by a single mobile station in the presence of simultaneous TCP flows by other P2PWNC users, when IPsec is employed to secure communications.

Keywords

Community Networks, Secure VoIP, Wi-Fi Networks, Performance Measurements

1. INTRODUCTION

The low cost and ease of deployment of Wi-Fi technology have assisted in its proliferation, making it the predominant standard for Wireless LANs. Wi-Fi-enabled network interface cards are standard equipment for laptops, PDAs and smart phones, while WLAN access points (APs) connected to fixed broadband lines are commonplace in homes, university campuses and corporate environments. In densely populated urban areas, WLAN signals are ubiquitous creating the ideal environment for the development of community based wireless Internet services.

As a result of the low cost of Wi-Fi equipment and the fact that the 802.11 standard operates in unlicensed spectrum, *Wireless Community Networks (WCNs)* [1, 8] have sprung in many cities. These networks are managed by individual volunteers who aim at interconnecting their nodes using wireless backbone links and provide free IP-based voice and data services to their members. Furthermore, there are municipality initiated and supported networks, such as [10], whose purpose is to offer wireless Internet access throughout cities.

The case is different for residential WLANs, which are typically connected to flat-rate DSL lines. Usually, they are not configured for open access and are secured by means of mechanisms such as WEP or WPA. Although wireless coverage in cities is significantly increased by such WLANs, access to them is restricted, even though these may be unused for many hours a day. In prior work [14], we have developed a scheme called *Peer to Peer Wireless Network Confederation (P2PWNC)*. This scheme aims at harnessing this underutilized resource and fueling ubiquitous wireless Internet access through residential WLAN sharing. A wireless community of users that share their Internet connections via WLAN with passers-by, with the anticipation of enjoying the same benefit for free when mobile, can thus be built on the private contributions of individual micro-providers.

In densely populated metropolitan areas with adequate wireless coverage, P2PWNC-based wireless communities can build an infrastructure for free Internet access and create a low cost alternative to GSM/3G services. The killer application on top of this scheme is envisioned to be free voice over IP, especially with the advent of WLAN-equipped smart phones. Performance of voice and data services over such an architecture, though, is an key issue, mainly due to the nature of wireless communications and the fact that our scheme is designed to operate on low-cost, off-the-shelf, embedded WLAN devices, which are resource constraint.

In this work, we experimentally evaluate the performance of community-based wireless communications, using P2PWNC as the basis for the provision of citywide wireless Internet access. Most important to us is to estimate the maximum number of simultaneous VoIP calls of acceptable qual-

Mobimedia'07 Month 8, 2007, Nafpaktos, Aitolokarnania, Greece
Copyright 2007 ICST 978-963-06-2670-5.

ity that a typical P2PWNC-enabled WLAN AP can sustain, by measuring how the P2PWNC protocol operation and the use of VPNs to secure communications affect voice quality. What is more, we present our results on the effects of the proposed architecture on the TCP throughput achieved.

Section 2 provides an overview of the P2PWNC scheme, with applications that can be built on top of it following in Section 3. Our experimental methodology and testbed are described in Sections 4 and 5 respectively and the results of our measurements are presented in Section 6. Before we conclude in Section 8, we present a short literature review (Section 7).

2. P2PWNC OVERVIEW

P2PWNC [14] aims at providing WLAN roaming services, featuring full anonymity for its users and not requiring any trusted central authority for controlling its operation. Thus, it is fully compatible with the anarchic and open nature of WCNs. P2PWNC participants use their wireless APs to offer Internet access to passers-by, while they consume Internet bandwidth from APs belonging to others when they themselves are mobile. Wireless bandwidth is shared in a peer-to-peer manner, similar to the way files are shared in popular file sharing schemes like Kazaa and Gnutella. P2PWNC operation is controlled by a simple rule of *reciprocity*; only users that contribute bandwidth to others can enjoy the same benefit when they themselves are mobile.

P2PWNC users are identified by simple public/private key pairs, which are not centrally stored or issued. Thus, there is no need for a centralized PKI. Service accounting is based on the exchange of digital receipts between *consumers* (mobile users who get Internet service) and *providers* (peers that provide Internet access through their APs). Each time service provision takes place, the consumer digitally signs a receipt containing his and the service provider's identity, thus acknowledging the service that has been consumed. Digital receipts are stored in *receipt repositories*, which form the system's history of transactions. Each time service is requested from an AP, the *reciprocity algorithm* [14] is invoked. Its input is the system's history of transactions and its output is the decision on whether the requesting party deserves to be granted service, according to the amount of service that he has provided to the P2PWNC community.

Communication among P2PWNC entities is carried out using a simple text-based protocol. Public key cryptography is used for digital receipt generation and verification, and for key and certificate management. Both the RSA and the Elliptic Curve Digital Signature Algorithm (ECDSA) are supported. The cryptographic operations that our system involves are CPU-expensive and the overhead imposed is more important, considering that P2PWNC is designed to operate on top of embedded devices such as PDAs and wireless APs.

3. APPLICATIONS

We expect that the primary application to be used on top of the P2PWNC architecture is Voice over IP (VoIP). Our vision is a situation where P2PWNC-enabled WCNs will provide almost complete coverage in metropolitan areas. If such an architecture were in place, people would be able to enjoy free Wi-Fi roaming around the city and place VoIP calls to other mobile users connected to P2PWNC hotspots. Thus, our architecture can provide a low-cost alternative to GSM/3G services in citywide areas.

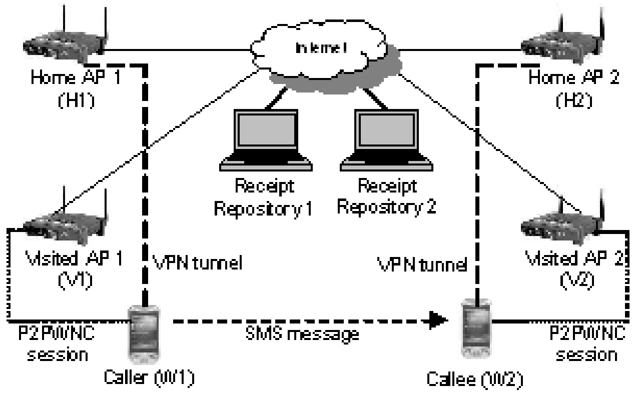

Figure 1: A P2PWNC-based secure multimedia call

Apart from its low cost, this alternative has inherent privacy enhancements. Given that P2PWNC does not entail central registration to a service provider and that users are free to switch identities at will, our scheme enhances user anonymity. However, even in such a case, a mechanism to secure VoIP calls from being intercepted by the untrusted APs that P2PWNC users associate with is necessary. In this section, we provide a brief description of how such a secure multimedia communications architecture should be built, considering the setup of a P2PWNC hotspot. Figure 1 shows the proposed architecture.

When a P2PWNC mobile user approaches an AP of another team, he set up a secure VPN connection with his home network and tunnel all his Internet traffic there, protecting it from interception by the untrusted visited AP. In order to spare users the need for extra equipment acting as a VPN gateway, we have built this functionality in the AP's firmware. It should be noted that this AP may serve other P2PWNC visitors at the same time. Suppose now that users W1 and W2 that belong to different P2PWNC teams wish to establish a voice call. W1 and W2 are assumed to have established P2PWNC sessions with APs V1 and V2 respectively (V1 and V2 belong to two other P2PWNC teams) and tunnel all their Internet traffic to their home gateways, H1 and H2 respectively.

In order to initiate the call, V1 must somehow discover V2. There are several ways to accomplish this. For example, dynamic DNS or a registrar dedicated to that purpose could be used. In our case, we have implemented user discovery using GSM SMS text messages. V1 can send an SMS to V2, informing him of his home gateway's (H1) IP address. Then, V2 responds with the voice stream, which is first tunneled to H2, then routed to H1, and, finally, tunneled to V1. Tunnels are implemented using L2TP/IPsec [23]. VPN tunneling imposes an important data and processing overhead, the effects of which on TCP throughput and voice quality are studied in this paper.

4. EXPERIMENTAL METHODOLOGY

Since our vision is to complement GSM/3G services with a P2PWNC-based alternative, we wish to estimate the capabilities of the proposed architecture as to the provision of

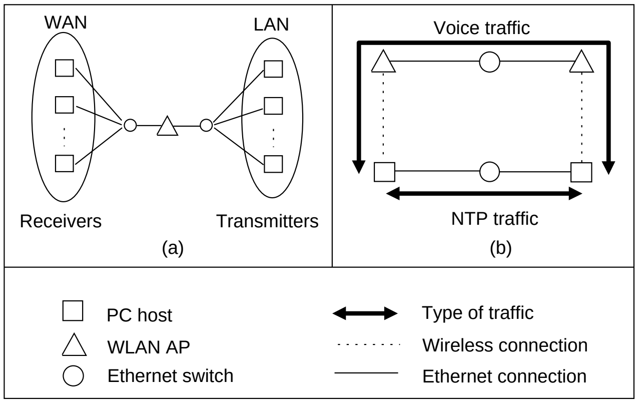

Figure 2: Testbed setup. (a) TCP measurements, (b) VoIP quality measurements.

secure voice and data services. In particular, we attempt to measure:

1. The number of voice calls of acceptable quality that a P2PWNC-based AP can sustain, especially when IPsec is employed to secure communication.

2. The TCP throughput degradation due to the various mechanisms that our architecture involves, when typical low-cost WLAN equipment is used. For this purpose, our reference testbed includes the Linksys WRT54GS [4] wireless router (see Section 5).

The requirements for a typical P2PWNC-enabled home WLAN AP are the following:

- Route mobile client's traffic using NAT [1].
- Operate the P2PWNC protocol.
- Act as the home VPN gateway for users that belong to the team that operates it and are currently visiting other (untrusted) P2PWNC APs.

Our experimental methodology and testbed setup have to consider the above requirements and study their combined effect on the performance of a P2PWNC-enabled AP.

4.1 TCP throughput measurements

Here we evaluate the impact of NAT routing, P2PWNC-related operations and VPN tunneling on the TCP throughput achieved by a single station connected to a Linksys box, in the presence of parallel sessions operated by other stations. For our TCP experiments, we carry traffic over the router's *wired* interface (ethernet 100BaseTX) instead of the wireless, to test the router's behavior under more severe traffic conditions.

Our experiments involved parallel bulk TCP transmissions of 100Mbytes. Each time, we measured the throughput achieved by a single reference host in the presence of additional transmitting-receiving pairs. The transmitters were connected to the Linksys WRT54GS router's local network (emulating local clients who are assigned private IP addresses required for the NAT routing), while the receivers were attached to the router's WAN interface (emulating some Internet hosts with public IP addresses). This setup is shown in Figure 2(a). Each set of experiments was repeated 5 times and the average throughput was calculated, for increasing number of parallel sessions (ranging from 1 to 7).

Six sets of experiments were carried out. First, as a reference, we measured the throughput achieved without the intervention of the Linksys box (i.e. over pure switched ethernet). Then, we connected the transmitting hosts to the router's LAN and tested the router's NAT performance. For the 3rd and 4th sets of experiments we initiated P2PWNC sessions, with the transmitting hosts acting as P2PWNC clients to the Linkys box, using either RSA or ECDSA receipts. Finally, L2TP/IPsec VPN tunnels were established between transmitters and the Linksys box to evaluate the combined effect of NAT, P2PWNC protocol and VPN operations (5th and 6th sets of experiments).

4.2 VoIP performance measurements

In our experiments, we emulated voice conversations by setting up bidirectional UDP flows between two laptop PCs. We implemented our own traffic generators, sending 50 packets per second with 20 bytes of audio payload each and 12 bytes for the RTP header. This traffic pattern corresponds to the G.729 codec, which is used by many available Wi-Fi VoIP phones. The 20 bytes of packet payload contain 20 msec of voice. Each host was connected to a different 802.11b WLAN AP and each voice call lasted for 90 seconds. We have assumed that at the receiver end there is a 120 msec dejitter buffer and that the first packet is played out when the buffer has reached half occupancy (equivalent to 3 packets). Thus, the dejitter buffer introduces a 60 msec delay in the playout process.

We initiated parallel VoIP calls between the two laptops and collected delay and loss information for each packet at the receiver end for one of the two call directions. While increasing the number of parallel calls, we also increased the number of receipt verifications per second on both Linksys boxes accordingly, to emulate P2PWNC operations (we assume that P2PWNC sessions and VoIP calls have a one-to-one relationship). Our results reflect the perceived voice quality for a single call in the presence of simultaneous calls. For the VPN experiments, we also set up VPN tunnels between the laptops and the APs each one was connected to.

To estimate user perceived voice quality, we have used the evaluation method proposed in [12]. This method reduces ITU-T's E-model [17] to transport level metrics which are directly measurable in our testbed.

Using the proposed methodology, we can derive a score that represents the subjective quality of a voice call based only on network delay, jitter and packet loss measurements. For the codec configuration described above, this score (*R-score*) is given by the following formula:

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

where:

[1] In a typical WLAN setting, wireless clients are given private IP addresses via DHCP and the AP uses NAT to route traffic from/to the clients. In our architecture, most APs are expected to be connected with a single DSL line and, thus, have only one public IP address

- $d_{network}$ is the end to end network delay
- $e_{network}$ represents network loss
- $e_{dejitter}$ represents loss in the dejitter buffer
- $H(x) = 1 \quad if \quad x > 0; \quad 0 \quad otherwise$

Furthermore, R-score is mapped to a subjective Mean Opinion Score (MOS) by the following set of equations:

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

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

5. TESTBED DESRCIPTION

5.1 System software and equipment

Our testbed is composed of PCs running a custom-made Linux distribution that we have created, based on the Knoppix [3] live CD, running kernel version 2.6.8. This distribution includes the measurement tools that we have developed, P2PWNC client software, and the Openswan [5] IPsec implementation. P2PWNC software and our measurement tools are available from our project's website [7].

Since we wish to test the capabilities of standard low-cost WLAN equipment, we have used the Linksys WRT54GS wireless router, with the OpenWRT [6] embedded Linux distribution (kernel version 2.4.29) on its firmware. Additional software, namely the P2PWNC access point module and VPN gateway software, was also included in the router's firmware. Each Linksys box is operated by a 200MHz MIPS processor, and is armed with 32Mb RAM and 8Mb flash memory. The two Linksys boxes were connected using a 3Com Ethernet switch

For the TCP experiments, we used 14 identical 3GHz Intel Pentium IV desktop PCs with 512Mb RAM, 7 of which were performing bulk TCP transfers to the other 7 over 100Mbit Ethernet, using SiS 100BaseTX ethernet cards.

For our voice measurements we used two Fujitsu Siemens laptops equipped with Intel PRO Wireless 2200 802.11b/g cards, as well as with Broadcom ethernet interfaces. Each of the 2 laptops was connected to a separate Linksys box using 802.11b, operating in DCF mode with the RTS/ CTS and fragmentation options disabled. Data rate was fixed at 11Mbps to disable the rate adaption mechanism, which imposes additional uncertainly in our experiments due to unfairness issues [16]. The laptops' wired interfaces were used for time synchronization (see Section 5.2). This experimental setup is shown in Figure 2(b).

5.2 Node synchronization

Nodes were synchronized using NTP and data transmissions were scheduled using Linux *crond* daemon. To estimate the quality of voice calls we needed accurate measurements of network delay. We achieved this by comparing the timestamps generated at the transmitter and the receiver end for each voice packet. In our voice experiments, packets were sent at fixed intervals of few milliseconds, so synchronization is considered fairly accurate. Nodes communicate with a local NTP server over their ethernet interfaces so that VoIP (wireless) traffic is isolated and the required accuracy level (of a few hundreds of microseconds) is achieved.

Table 1: Receipt operations CPU times

	Security level (RSA / ECC)	
Key size (bits)	**1024 / 160**	**2048 / 224**
Generation (msec)	300.6 / 20.3	1529.0 / 23.4
Verification (msec)	12.3 / 114.7	37.9 / 135.7

5.3 P2PWNC parameters

The main overhead of the P2PWNC protocol are the CPU-intensive receipt operations. We emulated P2PWNC sessions by performing receipt verifications at the APs at regular intervals. We assume that VoIP calls and P2PWNC sessions have a one-to-one relationship and that an AP requests a receipt from each visitor every 5 seconds. Therefore, in our experiments, more VoIP calls mean more parallel P2PWNC sessions, and thus, more frequent receipt verification requests.

As to the P2PWNC-specific cryptographic parameters, we have used both the RSA (with 1024 and 2048 bit keys), and the ECDSA algorithms for receipt generation/verification. Equivalent security to 1024 and 2048 bit RSA keys can be achieved with 160 and 224 ECC bit keys respectively [19]. As to the ECDSA-specific parameters, we have used the *secp160r1* and *secp224r1* verifiably random curves over the F_p finite field [25][26] to generate 160 and 224 bit keys.

5.4 VPN parameters

The Linksys WRT54GS wireless router operates also as a VPN gateway, with the Openswan IPsec implementation built into its firmware. The L2TP protocol was used for implementing tunnels and IPsec ESP (*Encapsulating Security Payload*) [18] was used to secure them [23]. IPsec operated in transport mode using the AES-CBC algorithm (128bit keys) for data encryption. Preshared keys were used for authentication.

6. RESULTS

In this section we present the results derived from our measurements. As a reference point, we have included the pure CPU times needed for a P2PWNC receipt generation and verification on the Linksys platform (Table 1), which also appeared in [14]. It should be noticed that receipt verifications (public key operations) are performed slower when the ECDSA algorithm is used instead of RSA. The opposite holds for receipt generations, which involve digital signing using the issuer's private key. However, fast digital signing is more important for the typically battery-powered mobile devices[2]. This, combined with the reduced space overhead due to smaller key/signature sizes makes ECDSA more favorable from the viewpoint of mobile devices.

Figure 3 shows the TCP throughput achieved by a single host in different scenarios, plotted against the number of simultaneous TCP flows. The "Reference" curve represents the average throughput of a single host when parallel

[2] P2PWNC receipts are generated by mobile nodes and verified by the service-providing AP

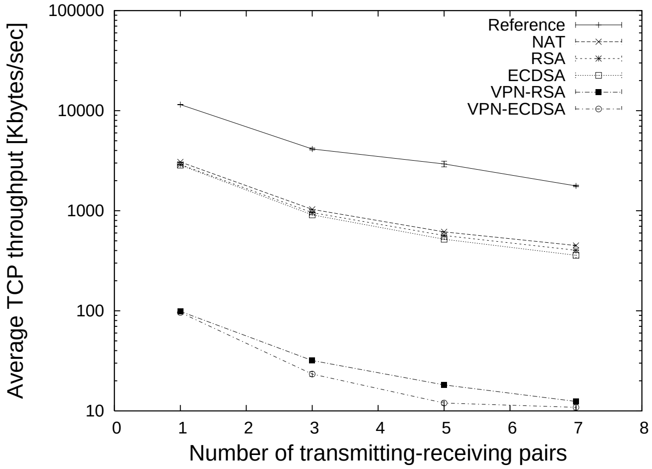

Figure 3: TCP throughput performance of a P2PWNC-enabled Linksys WRT54GS wireless router

TCP flows are present over pure switched Ethernet and is included for comparison. The "NAT" curve represents the scenario where all traffic is forwarded through the Linksys box, emulating the case where local clients communicate with some Internet hosts. NAT routing results in important throughput degradation. The most important overhead though, is imposed by VPN tunneling, as shown by the "VPN" curves. The overhead on TCP throughput is such that, for a high number of parallel sessions, it degrades to few kilobytes per second. The "RSA" and "ECDSA" curves imply that P2PWNC protocol overhead is minimal compared to that imposed by NAT and VPN tunneling, for the given P2PWNC configuration (see Section 5.3).

Table 2 shows the maximum number of simultaneous voice calls of acceptable quality in our scheme. When no security mechanism is in place and P2PWNC is not in use, in our wireless-to-wired-to-wireless scenario, 7 calls can be sustained. This is the maximum number of acceptable voice calls when no other architectural, security or protocol overhead is present. Again, the cost of using VPNs to secure communication proves to be high, especially combined with the use of ECDSA for P2PWNC receipts. In this case, due to the fact that ECDSA verifications are very CPU-expensive for the Linksys box, additional delay is imposed in routing voice packets coincident with a receipt verification event. Jitter that cannot be properly handled by the dejitter buffer may thus be introduced, leading to a MOS decrease.

The results of Table 2 derive from MOS calculations that are presented in Figure 4. The horizontal line (MOS value of 3.60) represents the minimum average MOS value that a call should score to be considered of acceptable quality. Figure 4 implies that there is a threshold in the number of simultaneous VoIP sessions, after which voice quality dramatically degrades[3].

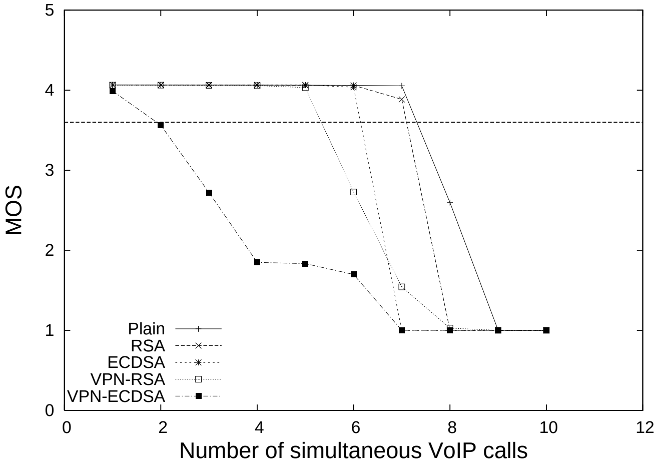

Figure 4: Average MOS as the number of simultaneous VoIP calls increases

Table 2: Maximum number of VoIP calls of acceptable quality

Scenario	Supported calls
Plain	7
P2PWNC (RSA 1024)	7
P2PWNC (ECDSA 160)	6
VPN - P2PWNC (RSA 1024)	5
VPN - P2PWNC (ECDSA 160)	2

Next, we quantify the per-packet space overhead of the applied tunneling mechanisms. In our voice scenario, the original IP packet (IP, UDP and RTP headers and voice payload, adding up to 60 bytes) is encapsulated in a PPP frame (4-byte header). The PPP frame is carried within an L2TP tunnel, thus an 8-byte L2TP header and an 8-byte UDP header are prepended to it. The resulting packet will be encrypted using the AES-CBC algorithm and encapsulated in an ESP header and trailer. The input data of the encryption algorithm also include the ESP "pad length" and "next header" fields (1 byte each) and their total length is 82 bytes. Before encryption, they are padded to become a multiple of the 16-byte AES-CBC block size, raising their size to 96 bytes. ESP packet contents also include the AES initialization vector (16 bytes), sequence number (4 bytes) and SPI index (4 bytes), as well as an integrity check value of 12 bytes (HMAC-MD5). Finally, the 132-byte packet is prepended with an IP header, adding up to a total of 152 bytes (compared to the 60 bytes of the unencrypted voice packet).

Note that in a practical P2PWNC-based scenario, NAT traversal would be used for IPsec communication, since users are typically in private LANs, and this would add to the space overhead (additional UDP encapsulation for traversing NAT).

Overall, it appears that security comes with a significant cost. Cryptographic operations related CPU overhead, especially for constraint devices such as embedded Linux-based WLAN APs and Wi-Fi enabled handheld devices, and the space overhead due to additional packet headers account for that.

[3] As shown in [15], adding a voice call causes the quality of all parallel calls to degrade. It was shown that for a wireless-to-wired scenario, given our codec settings, 14 calls can be supported. We experimentally verified this result using MOS calculations to compare it with our wireless-to-wired-to-wireless case.

7. RELATED WORK

The P2PWNC architecture, trust model, reciprocity algorithm and communication protocols were presented in [14]. Simulations were used to show the robustness of the reciprocity algorithm against sophisticated attacks, and some initial performance measurements followed. QoS extensions and their implementation details were presented in [13].

Similar motivation led the authors of [24] to propose a scheme for Wi-Fi roaming in which WISPs have multilateral roaming contracts and register with a central authority that maintains reputation records derived from QoS reports submitted by roamers. There are also some commercial solutions for WLAN sharing [2, 9], usually in a for-profit basis.

We have adopted the methodology proposed in [12] to monitor VoIP quality. This has also been used in [22] to assess VoIP performance in 802.11-based wireless mesh networks. In [15], the authors derived an upper bound on the number of VoIP calls that a 802.11 WLAN can support as a function of the VoIP codec used and the size of the audio payload, while in [21] experiments on the effects of IPsec on voice quality in 802.11 and Bluetooth cells are presented.

Voice over IPsec in wireline networks is experimentally studied in [11], where a header compression method called *cIPsec* is also proposed and evaluated. IPsec encryption and packetization overhead are studied via analysis and simulation in [27], while in [20] performance of IPsec and application layer security protocols are experimentally compared.

8. CONCLUSION

In this paper we used the P2PWNC peer-to-peer WLAN sharing infrastructure to build community-based voice and data services and evaluated their performance on typical off-the-shelf WLAN equipment. We estimated via experiments the TCP throughput degradation and the number of acceptable voice calls that a P2PWNC-enabled access point can admit, in an attempt to show that our architecture can be a viable secure low-cost alternative to GSM/3G services in metropolitan areas, where privately owned WLAN APs may provide adequate coverage. Our results indicate that the main overhead in implementing secure communications on top of this architecture is due to VPN tunneling and NAT routing, rather than P2PWNC-related operations.

9. ACKNOWLEDGEMENTS

This research is funded in part by the Greek GSRT under a project titled "E2E-Security over heterogeneous mobile/wireless networks."

10. REFERENCES

[1] Athens Wireless Metropolitan Network. http://www.awmn.net.

[2] FON. http://en.fon.com.

[3] KNOPPIX Linux Live CD. http://www.knoppix.org.

[4] Linksys. http://www.linksys.com.

[5] Openswan. http://www.openswan.org.

[6] OpenWRT Linux distribution. http://openwrt.org.

[7] The P2PWNC project website. http://mm.aueb.gr/research/p2pwnc/.

[8] Seattle Wireless. http://www.seattlewireless.net.

[9] Speakeasy WiFi NetShare service. http://www.speakeasy.net/netshare/.

[10] Wireless Philadelphia Executive Committee. http://www.phila.gov/wireless.

[11] R. Barbieri, D. Bruschi, and E. Rosti. Voice over IPsec: Analysis and solutions. In *Proc. 18th Annual Computer Security Applications Conference (ACSAC '02)*, page 261, 2002.

[12] R. G. Cole and J. H. Rosenbluth. Voice over IP performance monitoring. *Computer Communication Review*, 31(2):9–24, 2001.

[13] E. C. Efstathiou, F. A. Elianos, P. A. Frangoudis, V. P. Kemerlis, D. C. Paraskevaidis, E. C. Stefanis, and G. C. Polyzos. Public infrastructures for Internet access in metropolitan areas. In *Proc. AccessNets'06*, Athens, Greece, September 2006.

[14] E. C. Efstathiou, P. A. Frangoudis, and G. C. Polyzos. Stimulating participation in wireless community networks. In *Proc. IEEE INFOCOM*, Barcelona, Spain, April 2006.

[15] S. Garg and M. Kappes. Can I add a VoIP call? In *Proc. IEEE International Conference on Communications (ICC '03)*, volume 2, pages 779–783, 2003.

[16] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *Proc. IEEE INFOCOM*, San Francisco, CA, March 2003.

[17] ITU-T Recommendation G.107. The E-model, a computational model for use in transmission planning, December 1998.

[18] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP). RFC 2406, Nov. 1998.

[19] N. Koblitz, A. Menezes, and S. A. Vanstone. The state of elliptic curve cryptography. *Des. Codes Cryptography*, 19(2/3):173–193, 2000.

[20] S. Miltchev, S. Ioannidis, and A. Keromytis. A study of the relative costs of network security protocols. In *USENIX 2002 Annual Technical Conference*, pages 41–48, June 2002.

[21] A. Nascimento, A. Passito, E. Mota, E. Nascimento, and L. Carvalho. Can I add a secure VoIP call? In *Proc. WOWMOM '06*, pages 435–437, 2006.

[22] D. Niculescu, S. Ganguly, K. Kim, and R. Izmailov. Performance of VoIP in a 802.11 wireless mesh network. In *Proc. IEEE INFOCOM*, Barcelona, Spain, April 2006.

[23] B. Patel, B. Aboba, W. Dixon, G. Zorn, and S. Booth. Securing L2TP using IPsec. RFC 3193, Nov. 2001.

[24] N. B. Salem, J.-P. Hubaux, and M. Jakobsson. Reputation-based Wi-Fi deployment. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(3):69–81, 2005.

[25] Standards for Efficient Cryptography Group. *SEC1: Elliptic Curve Cryptography*, September 2000. Available at http://www.secg.org.

[26] Standards for Efficient Cryptography Group. *SEC2: Recommended Elliptic Curve Domain Parameters*, September 2000. Available at http://www.secg.org.

[27] C. Xenakis, N. Laoutaris, L. Merakos, and I. Stavrakakis. A generic characterization of the overheads imposed by IPsec and and associated cryptographic algorithms. *Computer Networks*, 50(17):3225–3241, 2006.