On the performance of secure user-centric VoIP communication

Pantelis A. Frangoudis a,⇑, George C. Polyzos b

a INRIA Rennes-Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes, France
b Mobile Multimedia Laboratory, Department of Informatics, School of Information Sciences and Technology, Athens University of Economics and Business, 11362 Athens, Greece

Abstract
Motivated by the increased Wi-Fi coverage in metropolitan areas and the emergence of user-centric wireless access schemes, we focus on the provision of secure, user-centric voice services and explore their potential performance-wise, by designing a VoIP communications scheme tailored to open-access wireless environments, but also with wider applicability, and experimenting with it to estimate its upper bounds on VoIP capacity, under constraints posed by user-centrism; operation at low-cost and on user-controlled equipment, minimal dependence on centralized entities, and tackling specific security challenges. We identify quality degradation factors and quantify their importance by simple analysis and experimentation, showing that typical user Wi-Fi equipment can sustain a satisfactory number of concurrent secure VoIP sessions with acceptable Quality of Experience and, at the same time, protection from malicious user activity can be offered to access providers, while a level of roaming privacy can be guaranteed.

1. Introduction

The traditional view of communications has recently been disrupted by the evident user empowerment in all aspects of the communication process. Traditionally, the operator-centric view dominated, where users had a passive role as service consumers. This view seems to change and the factors that have led to this shift are numerous. Hand-in-hand with the revolution in current Internet usage trends, where we witness a vast increase in the volume and popularity of user-generated content, a new communication paradigm, where users have a central role, has emerged. With the advent of low-cost, ubiquitous, and easy to install and configure wireless equipment, but most importantly, protocols that operate in unlicensed spectrum, users can effectively acquire the dual role of becoming service consumers and providers at the same time. This fact has the potential of bringing up new disruptive technologies, where users can enjoy low-cost wireless connectivity via the infrastructure provided by a heterogeneous crowd of micro-operators.

In this work, building on advances in user-centric wireless access [1,2], we focus on providing user-centric voice (and multimedia) communication services in such an environment. We assume an underlying access substrate where nomadic users enjoy community-based Internet access by potentially anonymous, and most probably untrusted, Wi-Fi hotspot operators and wish to set up end-to-end multimedia communication in a user-centric way, in the sense that critical operations such as signaling and security management are carried out on user-controlled software and equipment, with minimal dependence on centralized infrastructures. It should be noted that, although our work is positioned in the context of community-based Wi-Fi access, our approach to setting up end-to-end VoIP communication is more generic and could be applied to other wireless access infrastructures as well.
We face both security and performance challenges. Trust cannot be assumed among access providers and roaming users, thus data confidentiality cannot be guaranteed. At the same time, access providers need to be protected from malicious activities on behalf of roaming users, to which they may be held legally liable. On the other hand, performance of VoIP over user-centric wireless networks is a key issue, in part due to the unpredictable nature of wireless communications, where delay sensitive applications like Internet telephony are known to suffer. Poor signal conditions, but also contention for access to the medium and interference brought by spectrum scarcity, dense and anarchic Wi-Fi deployment, and poorly-configured wireless equipment, account for that. The need for security has also an adverse effect: important space and processing overhead is imposed by mechanisms applied to protect communications. These performance penalties are intensified by the limitations and challenges of a user-centric solution; minimal dependence on centralized infrastructure and operation on low-cost, resource-constrained, user-provided equipment, i.e., embedded WLAN devices and mobile terminals.

We approach the above challenges performance-wise, by designing and experimenting with a secure voice communications scheme which makes use of tunneling technologies, and evaluating service quality adopting a Quality-of-Experience (QoE)-oriented stance. Our purpose is to estimate the maximum number of simultaneous VoIP calls of acceptable quality – as a user would perceive it – that a typical home WLAN can sustain, by measuring how the use of security mechanisms and Wi-Fi operation affect voice quality. The contributions of this work are summarized below:

- We present solutions for VoIP communication services set up in purely user-centric fashion and operating fully on user equipment, designed for, but not limited to, a user-centric/community-based wireless access environment, and also discuss various design alternatives.
- We study the tradeoff between security and performance under the constraints posed by low-end user equipment. To the best of our knowledge, we are the first to provide a combined study of the overheads imposed by security mechanisms both at the processing and the protocol level under the premise that both call endpoints are connected to Wi-Fi links and cryptographic operations are carried out on low-cost, off-the-shelf user equipment, thus shedding light on the practical performance limits of secure user-centric VoIP communication.

This article is structured as follows: Section 2 provides a review of the state of the art in relevant research fields. Section 3 presents a user-centric secure VoIP service designed for user-centric wireless access, but also discusses design alternatives. In Section 4 we focus on the performance evaluation of this service; we use a simple analytic model to estimate upper bounds on VoIP capacity for our tunneling-based architecture (Section 4.2), describe our experimental methodology and testbed (Section 4.3) and present performance results (Section 4.4). In Section 5 we summarize and discuss our conclusions.

2. Related work

2.1. User-centric wireless access

User-provided wireless access schemes have recently received research and commercial attention. Based on the private contributions of individuals who operate Wi-Fi equipment, architectures and systems are being proposed with the aim of building resource sharing communities to achieve wide wireless coverage [1]. In this section we overview approaches in this direction, since our design assumes that such a scheme will be used for Wi-Fi access.

Large wireless communities such as guifi.net [3] and the Athens Wireless Metropolitan Network [4], counting thousands of participants, are nowadays a reality. Various communication services, and, notably, intra-community VoIP, are run on top of them, while at the same time they act as testbeds for research and experimentation [5]. Originally, research in the area focused on the technical aspects of building wireless community networks, but attention soon shifted towards socioeconomic and incentive aspects. A critical aspect of user-provided wireless networks is to design mechanisms to encourage contribution while limiting attacks by selfish users who aim at free riding. In our prior work [2], we proposed a fully decentralized scheme to this end. Our multimedia communications architecture was designed with this scheme in mind.

FON [6] has followed a different architectural approach. It acts as a mediator for the development of a Wi-Fi sharing community, centrally dealing with user authentication and accounting. The role of such mediators as community providers is modeled by Biczók et al. [7,8]. They analyze their interactions with users and ISPs in global-scale wireless community networks and explore the space of available parameters (e.g., roaming cost, ISP’s profit share) to determine the benefits of each player when joining the community. The authors present interesting results regarding the role of ISPs: Arguing that ISP endorsement is important for the global scaling of wireless communities, they find that depending on parameters set by the mediator, they will either fully support or abandon (i.e., prohibit Wi-Fi sharing) the community. This conclusion appears to be closely related to the terms of use adopted by ISPs regarding broadband connection sharing over Wi-Fi.

Two significant issues pertinent to wireless communities are studied by Manshaei et al. [9]. First, they study how initial community network coverage and user payoffs and fees affect the evolution of the community. Second, they focus on the competition between licensed wireless access providers and community-based ones, which is an important step towards answering whether wireless communities can be a viable alternative (or complement) to licensed cellular networks.

Ai et al. [10] focus on a critical usability aspect; their scheme only requires software updates at the client side and no firmware upgrades at the AP side. However, their design depends on a central server.

The legal aspects of user-provided wireless access should not be neglected. MacSithigh [11] discusses how the adoption of open wireless access is hindered by a
demonstrated on resource-constrained home Wi-Fi equipment; such scenarios have not received significant research attention, to the best of our knowledge.

Voice over IPsec in wireline networks is experimentally studied by Barbieri et al. [23], where a header compression method called cIPsec is also proposed and evaluated. IPsec encryption and packetization overhead are studied via analysis and simulation by Xenakis et al. [24], while Miltchev et al. [25] experimentally compare the performance of IPsec and application layer security protocols.

A work more closely related to ours is due to Nascimento et al. [26], who present experiments on the effects of IPsec the quality of VoIP over wireless networks. In particular, they measure the maximum number of simultaneous VoIP sessions that can be supported in a single wireless cell, comparing two different encryption algorithms (AES and 3DES) and two different wireless technologies (IEEE 802.11b and Bluetooth). Our work however focuses on a different architecture and configuration.

2.4. VoIP quality assessment

Numerous models for assessing the quality of voice services have been proposed. Jelassi et al. [27] provide a detailed review of relevant approaches.

Subjective methodologies [28] involve experiments with human subjects who rate voice (or conversational) quality. Objective methodologies such as PESQ (Perceptual Evaluation of Speech Quality) [29], on the other hand, attempt to estimate user-perceived quality by objective measurements, without involving human subjects: an original input signal is compared to a degraded output signal as a result of its transmission through a communication system. PESQ derives quality ratings using psychoacoustic fundamentals.

While fairly accurate, methodologies based on signal comparisons fail to capture the effects of various operating parameters (e.g., codec settings, dejitter buffer implementation) and phenomena in the end-to-end path (network delay, packet loss, delay variation). Parametric models have thus been proposed, aiming at estimating user-perceived voice quality and its dependence on such parameters. A widely used parametric model for conversational voice quality estimation is the ITU-T E-model [30]. It is based on modeling the results from a large number of subjective tests, on a wide range of transmission parameters. The output E-model is a scalar quality rating value known as the “Rating Factor, R”.

Since the E-model involves fairly complex expressions, Cole and Rosenbluth [31] have proposed a methodology to reduce the E-model to transport-level metrics and derive a simplified E-model approximation. We have adopted this methodology, since our purpose is to quantify the effects of various phenomena pertinent either to the network topology (i.e., the fact that we focus on a scenario where both call endpoints are attached to Wi-Fi links), network devices in the end-to-end path (resource-constrained residential Wi-Fi routers), or to security services we apply (high-overhead tunneling mechanisms) and correlate them with perceived voice quality; codec settings and other environmental factors were considered fixed or not studied in our work. In Appendix A, following the approach...
of Cole and Rosenbluth, we derive an updated approximation of the E-model, based on the most recent version of the standard.

2.5. Our prior work

In our prior work [2], we briefly reported on experimental results obtained for a user-centric, tunneling-based VoIP service, albeit in a different setup, i.e., with a different Wi-Fi standard (IEEE 802.11b) and different tunneling technologies (L2TP/IPsec), and our evaluation mainly focused on the performance effects of the Wi-Fi sharing protocol we had proposed. Here, we study more generic settings, providing design alternatives and a more detailed experimental analysis, with an in-depth look into the performance implications of various quality degradation factors, and backed up by a theoretical discussion on VoWLAN capacity.

3. Service architecture

3.1. Environment, design goals and threat model

We target an open-access wireless networking scenario, where users enjoy Internet connectivity over community Wi-Fi hotspots, expecting VoIP and mobile multimedia to emerge as popular applications and complement GSM/3G services in citywide areas, where Wi-Fi and cellular nowadays have comparable coverage. We design a peer-to-peer secure VoIP service tailored to such a networking environment, drawing our inspiration and some basic assumptions from our prior work [2], where we presented a decentralized community-based Wi-Fi sharing scheme based on service reciprocity. Our design is more generic, however, and could be applied independently of the underlying access solution: the same principles would apply for users attached to commercial Wi-Fi hotspots or Campus WLANs, since access is decoupled from other services, and one can consider that what differentiates access solutions from one another are the service negotiation and accounting procedures, with services (e.g., VoIP) being deployed independently on top of the access substrate.

Summarizing our design goals, we focused on (i) operation at low cost, (ii) purely on user-provided equipment, (iii) with minimal dependence on centralized infrastructures, (iv) offering secure communication and (v) privacy enhancements. We quantify the tradeoff among security, low cost and performance in Section 4.

We build on the basic assumption that between an access provider (visited AP) and a consumer (roaming user), no trust is assumed and no form of cooperation and interaction is expected apart from the basic Wi-Fi protocol exchange and routing user traffic to/from the Internet. We further assume that a user’s trust anchor is the equipment installed at his premises, i.e., his home Wi-Fi router. There is the risk for a visitor that the traffic forwarded by the AP on his behalf is intercepted, and, on the other hand, the visitor may engage in malicious acts, masking behind the provider’s home network.

3.2. A tunneling-based scheme

To tackle these threats, a design based on tunneling is a natural approach: after being granted wireless access, the user sets up a VPN tunnel to his home network and securely relays all his Internet traffic through it, protecting it from eavesdropping. For security purposes, it is reasonable to assume that a visited AP only forwards a user’s traffic from/to a specific IP address (or a restricted number thereof), which serves as the visitor’s trusted gateway. There are various technical means to achieve this: for instance, this address can be explicitly negotiated between the AP and the client after wireless access has been successfully established, or the AP can simply allow traffic only to/from a single destination across a user session, that of the first outgoing IP packet.

This approach mandates that the user operates a trusted VPN gateway. To lower infrastructure cost, offer a user-centric feel, and considering the capabilities of modern home Wi-Fi equipment, we have proposed that this VPN functionality be built into the firmware of the user’s home router. 1 Obviously, for improved performance, and if the user can afford it, he can operate the VPN gateway on separate equipment, and even outside his home network.

With this approach, confidentiality of the user’s traffic is ensured, while the service provider cannot be held liable for illegal activities carried out by the visitor, since potential attacks, download or distribution of illegal content, or other malicious behavior will appear to have been performed from the visitor’s home network (or the network where his VPN gateway is located).

3.3. Rendezvous and call setup

Given that visitors have set up secure tunnels to their home networks, we now show how a voice (or multimedia) call can be set up between two users accessing the Internet via foreign Wi-Fi hotspots. In order to spare users the need for extra equipment acting as a VPN gateway, we have built this functionality into the AP’s firmware. Our basic goal is to demonstrate that a secure multimedia session can be set up in a user-centric manner, with minimal dependence on centralized infrastructures. Fig. 1 shows the proposed scheme.

A major challenge one has to tackle is for the caller to discover where to initiate the call to. Call endpoints are reachable via their home VPN gateways. Thus, the caller needs to discover the IP address of the callee’s home. Public home IP addresses are typically assigned by the users’ ISPs and, more often than not, they are dynamically allocated from the ISP’s DHCP pool. Here we present a solution for peer discovery based on the exchange of GSM SMS text messages, based on the following assumptions:

- At any time, a user is aware of the IP address of his home VPN gateway, which he can communicate to its peer. This is necessary, not only for user discovery, but also to set up a tunnel in the first place.

1 It should be noted that, at the same time, this device may be offering Internet access to other stations, e.g., other wireless community members.
A user who wishes to initiate a call to the other end knows his peer’s GSM mobile phone number. This is a reasonable assumption, given that either the two users know each other from prior contact and have exchanged such information, or the caller is aware of the callee’s phone number via an external channel. In any case, this model follows the traditional GSM paradigm. Also, it is reasonable to assume that cellular phone owners have active subscriptions with GSM operators, and are thus reachable over GSM.

This is a point where we relax our decentralization assumption: we rely on the centralized GSM infrastructure, but only for reasons of service discovery, and exploit an external communication channel that is ubiquitous and already in use by peers.

Peer discovery works as follows: suppose that users U1 and U2 wish to establish a voice call. U1 and U2 are associated with APs V1 and V2 respectively and tunnel all their Internet traffic to their home gateways, H1 and H2 respectively. In order to initiate the call, U1 sends an SMS to U2, informing him of his home gateway’s (H1) IP address. The SMS can also convey other session parameters, such as the port on which his gateway can receive a media stream, codec parameters, and security-related information. (See Section 3.6 for a discussion on achieving end-to-end security.) Then, U2 responds directly with the voice stream, which is first tunneled to H2, then routed to H1, and, finally, tunneled to U1.

It should be noted that the voice application at the U1 side should be prepared to receive an incoming stream at the agreed upon port. Also, the appropriate state should have been set up at both home gateways so that incoming packets with the peer’s home gateway’s IP as the source are forwarded to the appropriate tunnel endpoint. (A home VPN gateway may be managing multiple tunnels to roaming stations at the same time.)

Various alternative technologies exist for implementing tunnels. In our measurement-based evaluation we have used OpenVPN [22], a popular and easy to deploy SSL/TLS-based solution. In our earlier experiments [2], though, we had applied an L2TP/IPsec-based scheme [32], which is more complex (protocol-wise) and with higher per-packet space overhead. VPN tunneling imposes an important data and processing overhead, the effects of which on voice quality are studied in Section 4.

3.4. Roaming privacy

One should note that our specific tunneling-based design offers users a level of location privacy. Since each call endpoint only discloses the IP address of his home network, it is not possible for the other end to infer the approximate location of his peer. This comes with a performance penalty, though, since each packet needs to be encrypted and decrypted twice. This overhead is quantified in Section 4.

3.5. Design alternatives

Rendezvous between two peers who wish to communicate can be carried out using various mechanisms and protocols and each has advantages and disadvantages in its own right. In any case, users need to discover each other’s service parameters, namely the IP address and port where they listen for incoming multimedia traffic and potentially negotiate security parameters to achieve an end-to-end encrypted channel. In our system, we have opted for call initiation based on the exchange of GSM SMS messages, but other options are possible.

3.5.1. Dynamic DNS

In a DNS-based alternative, instead of initiating a call using the callee’s mobile phone number, if we assume that a user’s home IP address can be resolved from a DNS name, a caller can directly stream his data to his peer’s home. The procedure then follows the same way. It should be noticed that since VPN gateways would normally reside in users’ home networks, typically accessible via a dynamic IP address, user equipment should dynamically update the name-IP address binding. Many dynamic DNS services exist [33] and most home network equipment have the capability to perform such updates built into their firmware.

Relying on DNS hides one vulnerability similar to when using GSM SMS text messages, since it is still implied that (i) users trust the Dynamic DNS name service, and (ii) name servers that peers use to resolve each other’s host names are trusted. These conditions have to do with how calls are set up: the caller tunnels a request to resolve the callee’s name to his home name server through his VPN gateway. If this name server is under the control of an adversary, it is possible that the callee’s name resolves to a host controlled by the adversary, who could then easily launch a man-in-the-middle attack. An end-to-end security scheme where users authenticate each other is necessary to combat such threats.

3.5.2. Using the session initiation protocol (SIP)

A multimedia call could also be established using the Session Initiation Protocol (SIP) [34]. Peer discovery, however, is again an important issue. Users are identified using SIP URLs, which mandate the existence of SIP registrars where users are listed and, typically, SIP proxies in each user’s domain. In the decentralized scenario we study, this
would mean that each peer should maintain its own SIP server or that some peers should host this service for the community and others should trust these peers and agree upon using their service. Typically, the caller communicates with a SIP proxy in its domain, which uses DNS to resolve a SIP proxy in the callee’s domain. DNS procedures can also be used to discover service parameters (such as the transport protocol to use or the port the other server listens to). Locating SIP servers is specified in RFC 3263 [35].

Applying a SIP-based solution has some advantages: first, it is a standards-compliant approach, with SIP being heavily used and tested. Also, many SIP phones or implementations of SIP user agent software are readily available. On the downside, the need for every peer to operate SIP servers increases management effort and home gateway complexity, while it is still necessary to rely on DNS.

3.5.3. Peer-to-peer SIP

The problem of locating call endpoints in a peer-to-peer manner is being studied within the P2PSIP IETF Working Group [36]. For reasons of scalability, resource limitations that do not allow for the installation of SIP servers, but also for reasons of trust and reliability, e.g., when an organization or individual does not want to rely on external centralized service infrastructure, a distributed alternative to SIP is being standardized. The core concept is to rely on a P2P network (a Distributed Hash Table (DHT) such as Chord [37]) built on the equipment of the users of the system to distribute the core functionality of SIP, such as user registration, proxying, and locating users. The REsource LOcation And Discovery (RELOAD) protocol has been specified by the Working Group for signaling in a P2PSIP network. Cirani et al. [38], on the other hand, have proposed a discovery architecture similar in nature, but without utilizing the RELOAD protocol. Again, instead of using SIP servers to locate call endpoints, a DHT is used as the IP address/port lookup structure.

Interestingly, the security aspects of P2PSIP architectures have recently received attention. For a thorough review of relevant issues, the reader is referred to the work of Tuceda et al. [39].

Applying the above approaches in a decentralized, user-centric open access scheme requires that either a network of special nodes form the DHT, or, some of the home gateways join it. This, in turn, implies the formation of a community where the incentives for participation are unclear: in order to build a reliable DHT on top of user Wi-Fi equipment, cooperation is necessary on the peers’ behalf, who would have to maintain a set of records and respond to lookup requests for multimedia calls they are not involved with. A mechanism to stimulate cooperation at the DHT level would thus be necessary.

3.6. End-to-end security and avoiding man-in-the-middle attacks

Providing end-to-end security in a peer-to-peer manner in the multimedia communications architecture that we have proposed is yet to be tackled. One would notice that although the two endpoints of the multimedia call have set up secure VPN connections with their trusted home gateways, the path between the two gateways is still unsecured.

If we assume that the IP address of the home gateway of the caller has been communicated to the callee via a GSM SMS, it is straightforward for the GSM operator to sniff on the multimedia call, performing a simple man-in-the-middle (MITM) attack: it can modify the contents of the SMS pointing to a gateway of its own. The callee responds with the multimedia stream that comes unsecured from the VPN gateway of the callee to the gateway that belongs to the GSM operator. Then, the operator forwards the traffic to the caller and none of the call endpoints is aware that their traffic is being intercepted.

To combat such an attack and achieve end-to-end secure and private peer-to-peer multimedia communication, the unprotected path of the call has to be secured. A simple means is that of conveying the caller’s public key to the callee during the call setup phase (GSM SMS exchange), and using it to exchange a shared key for traffic encryption. However, the callee has to verify that the public key of the caller has not been changed by an adversary (e.g., the GSM operator) performing a MITM attack. Thus, after communication has been set up and assuming a voice or video call, the two parties can use some form of vocal acknowledgment to verify that the public key exchanged is the appropriate. This way, an end-to-end secure VoIP or video call can be set up without resorting to trusted certification authorities.

Methods and systems that implement this type of in-band key exchange have been proposed in the literature. The ZRTP [40] protocol uses public key cryptography without resorting to a centralized PKI and, thus, without the need for certificates. During call setup, the two parties perform Diffie–Hellman key exchange in the media path and set up a shared secret. Then, a Short Authentication String (SAS) based on the shared secret is derived, which appears on the displays of the participants’ devices. To ensure that a man-in-the-middle attack is not being performed, both users should confirm the same SAS value.

All ZRTP messages are multiplexed with the media stream, making it independent of the signaling protocol used for call setup (e.g., SIP). The fact that it is purely peer-to-peer makes it suitable for application to our architecture. Vocal verification of a key exchanged between the two parties engaging in a phone conversation is a concept presented as early as 1996 [41].

One should note the performance implications of applying such end-to-end security schemes to our service design. While the two peers maintain VPN tunnels with their home networks to protect themselves from untrusted visited APs and suffering the overhead of tunneling, they also have to encrypt their data end-to-end before tunneling them home. Therefore, there is redundant use of security mechanisms. Along the mobile user-home gateway path data are encrypted twice: application data are first encrypted with the shared key established between users (call endpoints) when the media session was initiated. Then, data are encrypted again and tunneled home, where they are decapsulated, sent to the peer’s home gateway, encrypted and tunneled to the peer, where they are...
decapsulated. The resulting data are eventually delivered to the application, where they are decrypted using the shared key. It is important that future research efforts emphasize on removing redundancy across security layers, and optimizing the end-to-end media delivery path.

4. Performance evaluation

4.1. Motivation and approach

The inherent performance limitations of VoIP over WLAN, coupled with shifting demanding security operations to low-end user equipment, raise questions about the capacity of the proposed user-centric communication schemes. Depending on the scenario, there is a critical number of concurrent VoIP connections, beyond which user experience dramatically deteriorates for all connections, and communication collapses.\(^2\) We focus on identifying this threshold for different scenarios and study its dependence on various parameters. Such parameters, which affect voice quality, and, in turn, VoIP capacity, range from the operation of wireless communication technologies and the capabilities of user equipment, to codec configuration (e.g., codec type, packet payload size) and the application of security mechanisms.

Being able to identify capacity limits and the effects of quality degradation factors provides understanding about the capabilities and limits of user-centric VoIP communication, and is also important for network planning and management purposes. For example, in our design, a user-provided Wi-Fi router can serve as a trust anchor (VPN gateway) for multiple users, who can simultaneously tunnel their VoIP traffic through it. Based on the estimated VoIP capacity, the owner of this device can apply appropriate VPN admission control schemes to limit load and maintain satisfactory quality levels for the VoIP calls which are tunnelled through it. Also, he can opt for upgrading its equipment based on the target number of users that it is supposed to serve, or take any appropriate management decisions to provide QoE guarantees and improvements. Such mechanisms and actions are outside the scope of this article.

We selected to apply a mix of analysis and experimentation to evaluate the performance of our scheme: we first estimate upper bounds on the number of VoIP calls that can be supported in a community-based Wi-Fi access scheme using a simple throughput model, and then present experimental results, where we derive the same capacity metric carrying out user experience measurements for various settings, quantifying the effects of different quality degradation factors.

These two methodologies have complementary advantages: on the one hand, with a simple analytic model, we can study the effects of multiple parameters, getting a better understanding of how the underlying networking technologies are related with VoIP capacity, without resorting to time consuming testbed measurements. On the other hand, since the theoretical performance model cannot capture significant aspects, such as the effects of CPU-expensive VPN operations on user experience, and cannot directly provide delay and loss estimates based on which QoE is calculated, we perform testbed experiments, at the same time validating our theoretical results.

4.2. Upper bounds on VoWLAN capacity

4.2.1. Analysis

Prior work [16] reveals that even though the bitrate of a VoIP call may be small, the overhead imposed by the IEEE 802.11 PHY and MAC mechanisms and packet headers is such that an unexpectedly low number of concurrent VoIP sessions can be sustained by a Wi-Fi cell. In this section, we adapt the simple analytical model proposed by Garg and Kappes [16] for our scenario which involves two wireless last hops, to estimate an upper bound on the number of concurrent voice calls of acceptable quality. Here we focus on the effects of the PHY and MAC layers, as well as the packet overhead imposed by VPN mechanisms.

The IEEE 802.11 MAC protocol typically applies the Distributed Coordination Function (DCF), a CSMA/CA mechanism for arbitrating channel access. DCF dictates that for a station to transmit a packet, it should sense the medium idle for a specified time duration called DCF interframe space (DIFS). If so, the station enters a contention phase, where it senses the medium for a random number of slots. For each idle slot, it decrements a counter and transmission starts when the timer reduces to zero. If the medium is sensed busy at a contention slot, the station “freezes” its counter and needs to sense the medium idle again for a DIFS period before reentering the contention phase. The number of slots the medium should be sensed idle is drawn uniformly at random within a Contention Window (CW). It is possible that the backoff counters of two stations reach zero at the same slot. The stations will transmit and collision will occur. In a collision event, which is identified by the lack of an acknowledgment during a short period after the transmission, called the short interframe space (SIFS), stations retry transmission by drawing a new backoff counter value, doubling the size of the CW, after they have waited for a DIFS period. A retry limit is specified, after which a frame is considered dropped.

The above procedure imposes significant MAC-layer overhead, which is more evident when small packets are transmitted, as is typically the case for VoIP services. Also, for each transmission, there is some physical layer overhead: a preamble used for synchronization and the Physical Layer Convergence Protocol (PLCP) header transmitted at a low rate precede each frame. Two preamble types are specified. The short preamble is 72 bits and is transmitted at 1 Mbps, but is not compatible with legacy IEEE 802.11 systems, therefore the long preamble is typically used in IEEE 802.11b. When using the short preamble, the PLCP header (48 bits) is sent at 2 Mbps and the total PHY layer overhead adds up to 96 μs. In contrast, using the long preamble (144 bits), the header (48 bits) is transmitted at 1 Mbps and the total PHY layer overhead is 192 μs. In a Wi-Fi BSS operating in pure IEEE 802.11g mode

\(^2\) In the following sections, the term VoIP capacity denotes the maximum number of simultaneous VoIP connections that can be handled in a specific scenario.
(when no legacy IEEE 802.11b devices exist) the physical layer overhead adds up to 20 µs.

For IEEE 802.11b and IEEE 802.11g, fixed PHY, MAC and network (IP) layer overhead is presented in Table 1, where frame transmissions, except for the preamble and the PLCP header, are carried out at 11 and 54 Mbps, respectively. Depending on the higher-layer protocols used, overhead due to protocol headers increases.

A simple throughput model for our setup is given [16] by

\[
S = \frac{T_P}{T_P + T_{\text{Overhead}}} \times R,
\]

where \( T_P \) is the time required to transmit the audio payload at rate \( R \) and \( T_{\text{Overhead}} \) is the time overhead due to PHY, MAC and other higher-layer protocols. Our system handles constant bitrate, bidirectional flows. Let \( R_a \) denote the bitrate of a voice call. For G.729a, \( R_a = 16 \) kbps (8 kbps per direction). The total number of simultaneous voice sessions is thus given by

\[
n_{\text{max}} = \left\lfloor \frac{T_P \times R}{(T_P + T_{\text{Overhead}}) \times R_a} \right\rfloor,
\]

where

\[
T_{\text{Overhead}} = 2 \times (T_{\text{DIFS}} + T_{\text{DCF}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_H + 2) \\
\times T_{\text{PHY}} + T_P.
\]

\( T_H \) stands for the overhead imposed by MAC, IP, UDP, RTP and potential security related headers (OpenVPN or L2TP/IPsec). Note that \( T_{\text{Overhead}} \) is the time overhead for two wireless transmissions, therefore Eq. (3) also includes the “serialization” time \( T_P \) for retransmitting the packet payload over the second wireless hop.

The only component in the above equation that varies with the number of stations and traffic is the overhead imposed by the DCF mechanism. In IEEE 802.11b, and under various assumptions, Garg and Kappes [16] have approximated it as

\[
T_{\text{DCF}} = 8.5 \times T_{\text{SLOT}} + T_W \times P_c,
\]

where \( T_W = T_{\text{PHY}} + T_H + T_{\text{DIFS}} + T_{\text{SIFS}} + T_P \) is the time wasted for a packet that has suffered collision and \( P_c \) is the collision probability calculated as \( P_c = 0.03 \). In the following results, we make the following simplifying approximations to derive an upper bound on VoIP capacity:

- The time a station spends waiting for the medium to become idle is composed only of idle slots counted during the backoff phase. Namely, we ignore the case when a station freezes its backoff counter when sensing a transmission and the DIFS time that it has to sense the channel idle before restarting its backoff procedure. Since the transmission of a VoIP packet occurs rarely (e.g., once every 20 ms) and occupies little time (approximately 34 µs for a 20-byte audio packet, including IP/UDP/RTP protocol headers, using IEEE 802.11g), it is reasonable to assume that when operating below capacity, and given that VoIP traffic sources are not synchronized, the probability that a packet is generated during an ongoing transmission is low. With this approximation, we get \( T_{\text{DCF}} \approx \frac{8.5 \times 20}{34} \times T_{\text{SLOT}} = 7.5 \times T_{\text{SLOT}} \) for IEEE 802.11g.
- It should be noted that this approximation becomes less accurate as the size of packets increases (e.g., due to the space overhead imposed by security mechanisms).
- There are no collisions when operating below capacity.

We numerically evaluate Eq. (2) and compare the maximum number of simultaneous voice connections that can be achieved when using unencrypted VoIP streams with the case when they are secured with OpenVPN or L2TP/IPsec. In the latter cases, the total overhead due to protocol headers is shown in Table 1. (Details on the structure of a VPN-secured packet are discussed in Section 4.3.3.) We also compare the use of G.729a with G.711, in each case sending 20 ms of audio payload per packet (20 vs. 160 bytes). G.711 trades bandwidth requirements for slightly better audio quality. For example, for plain unencrypted G.729a (bitrate \( R_a = 16 \) kbps) over IEEE 802.11g \( (R = 54 \) Mbps), where \( T_H = 272 + 160 + 64 + 96 = 10.96 \) µs, \( T_{\text{PHY}} = 20 \) µs, \( T_{\text{DCF}} \approx 7.5 \times T_{\text{SLOT}} = 67.5 \) µs, \( T_{\text{SIFS}} = 10 \) µs, \( T_{\text{DIFS}} = 28 \) µs, \( T_{\text{ACK}} = 2 \) µs, and \( T_P = 2.96 \) µs (total overhead adding up to \( T_{\text{Overhead}} = 319.88 \) µs), the approximated VoIP capacity is \( n_{\text{max}} = \frac{T_P \times R}{(T_P + T_{\text{Overhead}}) \times R_a} = 30 \) calls. The respective number for G.711 is 27 calls. Our results are summarized in Table 2.

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed overheads for the transmission of an IP datagram over IEEE 802.11. A variable delay is also introduced due to the DCF mechanism.</td>
</tr>
<tr>
<td>Overhead</td>
</tr>
<tr>
<td>PHYS</td>
</tr>
<tr>
<td>Slot time (TSLOT)</td>
</tr>
<tr>
<td>MAC header</td>
</tr>
<tr>
<td>SIFS</td>
</tr>
<tr>
<td>DIFS</td>
</tr>
<tr>
<td>ACK</td>
</tr>
<tr>
<td>IP header</td>
</tr>
<tr>
<td>UDP header</td>
</tr>
<tr>
<td>RTP header</td>
</tr>
<tr>
<td>Security (OpenVPN)</td>
</tr>
<tr>
<td>Security (L2TP/IPsec)</td>
</tr>
</tbody>
</table>

\[ ^a T_{\text{DCF}} = T_{\text{SIFS}} + 2 \times T_{\text{SLOT}}. \]

\[ ^b \text{The space overhead due to tunneling varies, since padding is added to unencrypted data based on their size. The above values refer to the case for 60-byte unencrypted IP datagrams carrying 20 bytes of audio payload, a UDP and an RTP header.} \]
Table 2
Estimated maximum number of simultaneous VoIP calls in our wireless-to-wired-to-wireless scenario.

<table>
<thead>
<tr>
<th>Standard</th>
<th>G.729a</th>
<th>G.711</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.11b – Uncrypted</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>IEEE 802.11b – OpenVPN</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>IEEE 802.11b – L2TP/IPsec</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>IEEE 802.11g – Uncrypted</td>
<td>30</td>
<td>27</td>
</tr>
<tr>
<td>IEEE 802.11g – OpenVPN</td>
<td>28</td>
<td>25</td>
</tr>
<tr>
<td>IEEE 802.11g – L2TP/IPsec</td>
<td>28</td>
<td>25</td>
</tr>
</tbody>
</table>

\( a \) For IEEE 802.11b, we use the approximation of Garg and Kappes [16], where \( T_{DCF} = 8.5 \times T_{SLOT} + P_t \times T_W \).

\( b \) For IEEE 802.11g, we use \( T_{DCF} = \frac{C_{22}}{C_{21}} \times T_{SLOT} \).

4.2.2. Silence suppression

A technique that can significantly improve capacity at the expense of implementation complexity is silence suppression by means of Voice Activity Detection (VAD). In this case, packets are not transmitted if no voice activity is detected, thus reducing bandwidth demands. When the start of a silence period is detected, the transmitter signals the receiver with special packets so that the latter generates comfort noise. During a talk-spurt, packets are sent at a constant rate.

The performance benefits of using VAD cannot be easily quantified, because they heavily depend on the actual voice activity during a session. A simplifying assumption would be that VAD reduces required bandwidth by 50%. However, studies indicate that a voice activity detector creates silence and activity periods of different durations [42,43], thus the Voice Activity Factor (VAF), which determines the required bandwidth, may be less than 0.5.

ITU-T Recommendation P.59 [44] describes a method to generate artificial conversational speech based on a 4-state Markov model. It reports the temporal parameters of conversational speech, averaging the results of prior studies which where based on the analysis of speech samples (among which the popular work of Brady [42]). In particular, the reported average duration of a talk-spurt is 1.004 s, and the respective duration of a silence period is 1.587 s, yielding a VAF of \( \alpha = 0.38 \). Eq. (2) then becomes

\[
n_{\text{max}} = \frac{T_P \times R}{(T_P + T_{\text{Overhead}}) \times \alpha \times R_S}.
\]

As shown in Fig. 2, under the assumptions that (i) \( \alpha = 0.38 \), (ii) the G.729a codec is used, (iii) each packet carries 20 ms of audio payload, and (iv) IEEE 802.11g at 54 Mbps is used, applying VAD techniques improves VoIP capacity from 30 to 81 simultaneous calls. For the same settings, IEEE 802.11b (11 Mbps) capacity could improve from 7 to 18 calls.

4.2.3. The effects of the payload size

A further option to increase VoIP capacity is to increase the size of the audio payload, for instance by packing multiple samples in a single packet. However, increasing audio payload can arguably make quality worse, due to increased delay, loss and jitter. Most commercial implementations use small payload sizes (10–30 bytes). Fig. 2 shows the increase in voice capacity as the audio payload size increases in our scenario which involves two wireless hops. Note that while G.729a applies compression such that 10 ms of audio are encoded using 10 bytes, for G.711, 10 ms of audio expand to 80 bytes.

4.3. Experimental methodology

Our calculations in the previous section cannot capture the effects of cryptographic operations on VoIP capacity and involve simplifications about MAC layer behavior. To address these issues, but also to offer a user-centric view of how voice quality is perceived and study how transport level characteristics are associated with the achieved QoE, we performed a set of testbed experiments.

Again, we wish to estimate the VoIP capacity of our secure tunneling-based architecture when critical security functionality is built into typical home Wi-Fi devices.

The distinctive characteristics of our approach are that:

- Both call endpoints are assumed to be attached to (potentially community-operated) Wi-Fi APs.
- Users tunnel their traffic to trusted VPN gateways. In our case, these gateways are collocated with each user’s home Wi-Fi router (built into its firmware).

Thus, the requirements for a typical home WLAN AP are the following:

- Route mobile client’s traffic using NAT.

\[\text{In a typical WLAN setting, wireless clients are given private IP addresses via DHCP and the AP uses NAT to route traffic from/to the clients. In our architecture, most APs are expected to be connected with a single DSL line and, thus, have only one public IP address.}\]
Act as the home VPN gateway for its owner, while the latter is visiting other (untrusted) APs. Encryption mechanisms and packet expansion due to additional headers cause processing and data overhead.

Our experimental methodology and testbed setup have to consider the above requirements and study their combined effect on VoIP performance.

4.3.1. Testbed description
VoIP call endpoints are laptops running Linux 2.6.38. We have implemented a set of measurement tools to generate constant-bitrate bidirectional UDP streams to emulate VoIP traffic, and have selected OpenVPN to implement VPN tunnels, due to its configuration simplicity and popularity.

Since we wish to test the capabilities of standard low-cost WLAN equipment, we have used Linksys WRT54GL wireless routers running the Openwrt Kamikaze 8.09 distribution (Linux 2.4.35). Linksys WRT54GL APs are based on the Broadcom BCM5352 chipset, which includes a 200 MHz MIPS processor, 16 MB RAM and 4 MB of permanent storage (flash) where the firmware is also stored. A Broadcom IEEE 802.11b/g Wi-Fi adapter (controlled by a proprietary driver) is also included.

Transmission rate was fixed at 54 Mbps to disable rate adaptation, and the RTS/CTS mechanism was disabled. Our testbed operated in pure IEEE 802.11g mode (i.e., the physical layer overhead was 20 μs, $T_{\text{slot}} = 9 \mu$s and $CW_{\text{min}} = 15$). To avoid the effects of interference and contention for channel access, the APs operated in different non-overlapping channels (1 and 11) and we verified via site surveys that no other Wi-Fi networks were operating at our testbed location during our experiments.

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. Transmitters and receivers were synchronized using NTP: one of the two laptops was operating an NTP server and the gigabit Ethernet interfaces of the two hosts were connected back-to-back, so that VoIP (wireless) traffic was isolated and receivers were synchronized using NTP: one of the two hosts was connected back-to-back, so that VoIP (wireless) traffic was isolated and the required millisecond-level accuracy was achieved. In our experiments, packets were sent at fixed 20 ms intervals, so synchronization is considered fairly accurate.

Our experimental testbed setup is shown in Fig. 3.

4.3.2. VoIP quality assessment
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.729a codec, which is used by many available Wi-Fi VoIP phones. The 20 bytes of packet payload contain 20 ms of voice. Each host was connected to a different IEEE 802.11g WLAN AP and each voice call lasted for at least 120 s. We assume that at the receiver end there is a dejitter buffer to ensure smooth playout at the expense of a constant 60 ms delay.

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. Since our testbed is symmetric, the same results were observed for the opposite direction.

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 VoIP QoE, we used the evaluation methodology of Cole and Rosenbluth to reduce ITU-T's E-model to transport level metrics (see Section 2.4 and Appendix A) and derive a score that represents the subjective quality of a voice call based only on network delay, jitter and packet loss information, which are directly measurable in our testbed. For the codec configuration described above, this score ($R$-factor) is given by Eq. (A.1). For a call of acceptable quality, the $R$-factor value should be greater than 70.

4.3.3. Security parameters
The home wireless router operates also as a VPN gateway. In our prior work we experimented with an L2TP/IPsec solution, building the Openswan IPsec implementation into the router’s firmware. The L2TP protocol was used for implementing tunnels and IPsec ESP (Encapsulating Security Payload) was used to secure them. IPsec operated in transport mode using the AES-CBC algorithm (128bit keys) for data encryption. Preshared keys were used for authentication.

The above solution was complex protocol-wise, but also as far as configuration and maintenance were concerned. As to its space overhead, 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), which is carried within an L2TP tunnel, thus an 8-byte L2TP header and an 8-byte UDP header are prepended to it. The

---

5 Note that, at the same time, multiple tunnels with different endpoints may be active: more than one users may be using the same gateway at the same time, since a home AP may be shared by more than one (e.g., family members).

---

Preshared keys were used for authentication.

AES-CBC algorithm (128bit keys) for data encryption.

Preshared keys were used for authentication.

The above solution was complex protocol-wise, but also as far as configuration and maintenance were concerned. As to its space overhead, 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), which 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). Fig. 4 shows the format of a tunneled VoIP packet using L2TP/IPsec.

Note that, in practice, 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).

Instead of L2TP/IPsec, in this article we opted for an OpenVPN-based solution, maintaining the same security level (128-bit AES-CBC, but also certificate-based SSL/TLS session authentication and key exchange and HMAC-SHA1 packet authentication) and exploiting the popularity, configuration simplicity and wide availability of OpenVPN. In this case, the size of a 60-byte IP datagram expands to 145 bytes (compared to 152 bytes when using L2TP/IPsec). The structure of a packet tunneled over UDP using OpenVPN [48] and the above combination of encryption and authentication schemes is shown in Fig. 5. Note that the size of the data to encrypt is 64 bytes (audio payload plus the 4-byte ID field). Since this is a multiple of the AES-CBC block size, a 16-byte padding is necessary.

4.4. Results

In this section we present our measurement results. The figures presented here include end-to-end network delay, packet loss and dejitter buffer loss statistics for three types of experiments; (i) plain unencrypted VoIP transport, (ii) emulation of the space overhead introduced by OpenVPN tunneling, and (iii) VPN-secured VoIP transport.

For each experiment, we also calculate user-perceived VoIP QoE as a function of the above three quantities and for the codec settings we have assumed. Each data point is the mean of the measured values for 5 iterations of the same experiment, i.e., 5 VoIP calls under the same conditions, presented with 95% confidence intervals. For each iteration, the R-factor was calculated using the mean values for delay, network loss and dejitter buffer loss.

For comparison, we also mention our prior results obtained in an IEEE 802.11b testbed.

4.4.1. IEEE 802.11g performance

Our first experiment does not involve any security mechanisms. Instead, it measures transport level characteristics and quantifies user QoE for an unencrypted end-to-end VoIP session. The major quality degradation components are due to IEEE 802.11 MAC and PHY layer mechanisms. This experiment serves as a baseline case for the measurements to follow. We find that 30 concurrent VoIP sessions can be sustained, with mean R-score >75 (Fig. 9). Beyond that point, network delay reaches more than 100 ms (Fig. 6), which, combined with the intrinsic encoder delay (25 ms) and the playout delay introduced by the dejitter buffer, but also with a noticeable network (Fig. 7) and dejitter buffer (Fig. 8) loss makes call quality unacceptable. Results of this experiment are shown as circle points in the figures.

The results of this experiment exactly match the upper bound derived using a simple analytic model in Section 4.2 (see Table 2).

4.4.2. VPN space overhead

In this experiment we quantify QoE degradation due to the space overhead imposed by VPN mechanisms. We do not carry out any cryptographic operations. On the contrary, we transmit packets of fixed size, equal to the size of 60-byte VoIP datagrams tunneled using OpenVPN.
As discussed in Section 4.3.3, the size of each such IP datagram is 145 bytes. There is a noticeable drop in VoIP capacity in this case: Space overhead accounts for a 30% decrease in VoIP capacity, which is not fully captured by our analytic model. Instead of the measured maximum number of 21 concurrent high-quality VoIP sessions (With 21 ongoing calls, R-score is still above the quality threshold, as shown in Fig. 9), we had estimated that 28 calls could be sustained. This is because our analysis underestimated the delay imposed by the IEEE 802.11 DCF. As packet size increases, the probability that a node senses the medium busy while being in its backoff stage increases. This causes its backoff counter to freeze, moving $T_{DCF}$ beyond the optimistic $\frac{CW_{min}}{2} \times T_{SLOT}$ value that we had assumed.

4.4.3. Combined VPN space and processing overhead

Our 3rd experiment focuses on the combined effect of VPN space and processing overhead. Note that we have assumed that each call endpoint operates a separate VPN gateway at its premises, building this functionality in the Wi-Fi router’s firmware. Other options which would offer performance enhancements are possible (e.g., dedicating a more powerful device to this purpose or negotiating an end-to-end secure tunnel, which would involve a single encryption and decryption operation per packet), but our solution is more generic and spares the need for separate VPN equipment.

This time, performance is severely impacted, but a reasonable number of parallel high-quality VoIP sessions is
still possible to support. In particular, 8 such calls can be simultaneously supported, maintaining very high quality (mean R-score approximately 80), as Fig. 9 indicates. However, adding a single secure VoIP call causes all sessions to collapse: one-way network delay increases to more than 220 ms, while 5–10% of the transmitted packets are lost. Dejitter buffer loss was small in all our experiments. Packet loss obviously does not occur due to congested wireless links, since for the same traffic pattern without performing VPN-related cryptographic operations (see Section 4.4.2), loss was negligible. Rather, increased processing requirements at the APs incur queuing delays, which cause buffer overflows and thus packet loss.

As a final note, the AP processor is much slower than that of the laptops we used in our experiments. Tests we carried out using the OpenSSL speed utility revealed that the cryptographic throughput of encrypting 64-byte blocks using the AES-CBC algorithm (128 bits) on a Linksys WRT54GL was 2 MBps, compared to 90 MBps on an Intel i3 quad-core CPU (making use of a single core for the speed test, though) which we used in our experiments. The laptop was capable of encrypting/decrypting approximately 1.4 M packets/s, whereas 9 concurrent VoIP sessions generated and received 900 packets/s. Therefore, for 9 concurrent VoIP sessions, the AP seems to be the bottleneck, even though in our testbed the laptop was handling the same amount of traffic to encrypt/decrypt.

4.4.4. Comparison with IEEE 802.11b

In our prior work [2] we presented results from a similar set of experiments, albeit using IEEE 802.11b, but also L2TP/IPsec for implementing tunnels. When no security mechanism was in place, in our wireless-to-wired-to-wireless scenario, 7 calls could be sustained. This is in line with our theoretical upper bound of Section 4.2. We also experimentally verified the result of Garg and Kappes [16] that 14 calls can be admitted given our codec settings in a single-hop wireless-to-wired scenario, using QoE metrics. The use of L2TP/IPsec further reduced the number of sustainable VoIP sessions to 5. As a final note, Nascimento et al. [26] presented experiments showing that, in a different setting, where VoIP is encrypted end-to-end using AES-CBC and IPsec ESP in tunnel mode, an IEEE 802.11b WLAN can support up to 8 simultaneous G.711 calls (with 30 ms of audio per packet) with acceptable QoE. We validated this result analytically. Their experimental setting, however, does not correspond to our architecture design, and cannot appropriately capture the processing overhead due to the VPN operations performed on user Wi-Fi routers.

5. Conclusion

In this article, we have delved into performance aspects of secure, user-centric, voice communication over wireless access. To this end, we have presented a tunneling-based communications scheme, at the same time discussing various design alternatives. Our work has touched upon the tradeoffs posed by the needs for low-cost operation, security and performance, quantifying the overhead imposed by security mechanisms and operation on low-end user equipment.

We found that operating VPN gateways in typical off-the-shelf Wi-Fi equipment is the major quality degradation component: in our configuration, the number of high-quality VoIP sessions which can be simultaneously sustained drops from 30 to 8 when security mechanisms are put in effect, and this is mainly due to the increased delay and observed packet loss imposed by the CPU-intensive encryption and decryption operations at the APs. Our architecture is thus CPU-, rather than bandwidth-constrained, which is a promising result, as further improving VoIP capacity depends to a significant extent on increasing the processing power at VPN gateways, and using techniques such as silence suppression to reduce the load on their crypto engines. Furthermore, the observation that packet loss due to excessive jitter was small, especially in low traffic load scenarios, since we assumed a static dejitter buffer which added a fixed 60 ms delay, motivates the investigation of adaptive buffering techniques to reduce the introduced delay and improve user experience.

Overall, our results show that, performance-wise, even in the constrained environments that we focus, with user-centric Wi-Fi access solutions with adequate coverage in place, low-cost, secure, user-driven alternatives for mobile multimedia could be viable.

Acknowledgements

This work was carried out in part while P.A. Frangoudis was with the Mobile Multimedia Laboratory, Athens University of Economics and Business, and during the tenure of an ERCIM “Alain Bensoussan” Fellowship Programme at INRIA Rennes-Bretagne Atlantique. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007–2013) under Grant agreement No. 246016.

Appendix A. Updated expressions for approximating the E-model

The E-model, specified in ITU-T G.107 [30], is based on an algorithm with which the individual transmission parameters are transformed into different individual “impairment factors,” whose effect on user experience is additive. Its output, the R-factor, is given by

$$R = R_0 - I_s - I_d - I_{e_{-eff}} + A.$$  

$R_0$ is the basic signal-to-noise ratio at the receiver end, including noise sources such as circuit noise and room noise, and $I_s$ is the combination of impairments that occur simultaneously with the voice signal (e.g., low loudness, non-optimal sidetone, quantization noise). $I_d$ represents delay impairments and $I_{e_{-eff}}$ is the packet-loss dependent effective equipment impairment factor, which includes impairments caused by low bitrate codecs and packet loss. The E-model also includes an “advantage of access” factor $A$ to account for cases where user expectations compensate for impairments: For example, a user may accept some decrease in quality in exchange for mobile connectivity. We drop the advantage factor in this discussion.
ITU-T G.107 defines default values for all input parameters to the E-model algorithm, and recommends using these values for all parameters which are not varied during its application. Choosing default values for all parameters other than delay and equipment impairments, the rating factor $R$ is reduced to

$$ R = 93.2 - L_d - L_{e-eff}. $$

$L_{e-eff}$ is given by

$$ L_{e-eff} = L_e + (95 - L_e) \frac{P_{pl}}{BurstR} + B_{pl}, $$

where $L_e$ is the codec-specific equipment impairment factor, $BurstR$ is the packet loss burstiness, $P_{pl}$ is the packet loss percentage and $B_{pl}$ is a factor expressing the robustness of the codec to packet loss. Recommendation ITU-T G.113 [49] provides default values for $L_e$ and $B_{pl}$ for various codecs. For G.729a, and assuming random packet loss ($BurstR = 1$), $L_e = 11$ and $B_{pl} = 19$. Thus, the above equation becomes

$$ L_{e-eff} = 11 + 84 \frac{P_{pl}}{B_{pl}}. $$

To derive a simplified expression of $L_d$, Cole and Rosenberg [31] used the default values for all other parameters than network and codec-induced delay in the analytic expression for $L_d$ and produced a set of points, on which they fit a simpler curve. We reproduced their approach using the recent version of the E-model. We calculated the codec to network and codec-induced delay in the analytic expression of the delay impairment factor $d$ for G.729a, with each IP packet carrying non-bursty random packet loss:

$$ R = 81.863 - 84 \frac{P_{pl}}{P_{pl} + 19} - 0.024d - (0.097d - 16.467)/H(d - 169.4). $$

where $d$ is the total delay in the mouth-to-ear path and $H(x) = 1$ if $x > 0$; 0 otherwise.

Combining the expressions of $L_{e-eff}$ and $L_d$, we derive an expression of the $R$-factor for the G.729a codec and assuming (non-bursty) random packet loss:

$$ R = 81.863 - 84 \frac{P_{pl}}{P_{pl} + 19} - 0.024d - (0.097d - 16.467)/H(d - 169.4). $$

Note that $P_{pl}$ includes packet losses in the network path, but also packets that are discarded due to excessive jitter, i.e.,

$$ P_{pl} = 100 \times (e_{mk} + (1 - e_{mk})e_{dj}), $$

where $e_{mk}$ is the network packet loss ratio and $e_{dj}$ is the ratio of packets discarded at the dejitter buffer. Furthermore, the delay components are network latency ($d_{mk}$), as well as the delays associated with the codec and dejitter buffer configuration. For G.729a, with each IP packet carrying 20 ms audio blocks, the algorithmic and packetization delay add up to 25 ms. If we also assume a 60 ms dejitter buffer, $d = d_{max} + 85$ (in ms).

**References**


Pantelis A. Frangoudis received his PhD (2012) in Computer Science from the Department of Informatics, Athens University of Economics and Business. He holds a BSc (2003) and an MSc (2005) in Computer Science from the same department. Currently, he is a post-doctoral researcher at INRIA Rennes-Bretagne Atlantique. His research interests include wireless networking, Internet multimedia and network security.

George C. Polyzos, Computer Science Professor at AUEB, founded and is leading the Mobile Multimedia Laboratory. He was CSE Professor at UCSD, Computer Systems Laboratory co-director, Center for Wireless Communications Steering Committee member, and San Diego Supercomputer Center Senior Fellow. Obtained his EE Diploma (NTUA), and MA (EE) and PhD (CS) from the University of Toronto. His current research interests include Internet architecture and protocols, mobile multimedia communications, wireless networks, ubiquitous computing, and network security.