A Short History of VoIP Services

Dorgham Sisalem[1], Jiri Kuthan[1], and Jörg Ott[2]

[1] Tekelec Germany
Berlin, Germany
{Dorgham.sisalem,Jiri.Kuthan}@tekelec.com
[2] Aalto University
Espoo, Finland
jorg.ott@aalto.fi

Abstract. While starting as an experimental research topic in the early seventies VoIP went through different stages before becoming a commodity service competing with the circuit switched telephony and in some cases even replacing it. In this chapter we give a brief overview of the major developments in the area of voice over IP (VoIP) and look at the major milestones and competing standards. We further give a short look into the latest developments and recent applications and deployment scenarios.

1 Introduction

The discussion about the various aspects of advanced services such as service creation, service platforms and service interfaces have occupied a large share of the research and standardization work done in the area of telecommunications. However, even with the increasing usage of smart phones the plain telephony service based on the selling of minutes still generates more than half of the revenues of telecom operators [1].

Up to the early nineties the voice service used to be the only revenue generating service of telecommunication operators. Since the introduction of telephony services at the end of the nineteenth century most of the innovations in the telecommunication sector were targeted at the operators and not the customers. Hence, the most revolutionary innovations such as the move from manual switching to mechanical switching or the move from analog to digital had nearly no effects on the services used by the subscribers. The service itself did not change, only the comfort of using it, the price and availability have improved.

While the introduction of intelligent networks (IN) and ISDN have surely improved the quality of the telephony services and added a number of additional useful features the first real revolution in telecommunication networks as perceived by the subscribers was the move from fixed to mobile networks in the late eighties. However, even in this case, the service was still plain telephony.

The nineties saw the advent of two major developments. With the rise of the Internet operators started offering dial-up access. Thereby, the phone plug was no longer just the source of calls but also the access point to music, video and chat services and the world wide web. The introduction of the Short Message Service

E. Bertin et al. (Eds.): Telecommunication Services Evolution, LNCS 7768, pp. 90–110, 2013.

(SMS) extended the telephony service of mobile operators with a simple service for exchanging text messages.

The nineties have also seen the first attempts to introduce telephony services on top of the Internet. While mostly a commercial failure these early Voice over Internet Protocol (VoIP) services were the first steps for the introduction of real-time communication to the Internet and the transformation of the Internet into an all encompassing communication platform.

In this chapter we will be looking at the different stages of the development of the VoIP technology over the last thirty years and its effects on the telecommunication market.

2 Pre-VoIP: Voice over Packet Networks

The first papers discussing the possibility of transmitting information using packet switched networks were published in the early sixties [2]. Already at this early stage of the development of packet switched networks the authors were considering the possibility of transmitting voice over packet switched networks [3].

These early considerations were first put into practice with the ARPA (Advanced Research Projects Agency) funded research project under the name of Network Secure Communications (NSC) in the beginning of the seventies. The goal of the NSC project was "to develop and demonstrate the feasibility of secure, high-quality, low-bandwidth, real-time, full-duplex (two-way) digital voice communications over packet-switched computer communications networks" [4].

As part of the NCS project the Network Voice Protocol (NVP) was designed and implemented. NVP specified a control and a data transport protocol. The control part of NVP enabled the establishment and termination of two and multi-party voice sessions and negotiation of capabilities. The data protocol enabled the transport of voice packets between the end systems.

The NCS project resulted in the development of a low bandwidth voice compression algorithm, namely LPC (Linear Predictive Coding) [5] as well an implementation of the NVP protocol and the first demonstration of voice over a packet switched network between different sites connected to the ARPA network.

The NCS project resulted in an innovative communication system supporting a user interface, multi-party communication, voicemail and floor control. However, as only well funded universities and research labs could afford the needed hardware and network links these results can only be seen as a proof of concept showing the feasibility of using packets switched networks for voice communication.

It is probably worth noting that while NVP can be seen as the predecessor of modern VoIP protocols, NVP did not run on IP as the specifications of the Internet Protocol [6] and the move from the then used NCP [7] (Network Control Protocol) to IP did not take place till the beginning of the eighties.

3 First Steps: Proprietary Solutions

In the seventies and eighties most of the work related to VoIP was confined to universities and research labs. There, researches investigated different possibilities for exchanging audio and video data over packetized networks with a high quality of service (QoS). This involved research on compression schemes, scheduling and queuing algorithms, congestion control mechanisms and protocols and operating systems for real-time communication.

It wasn't until the mid nineties that the VoIP technology was allowed to leave the research labs. The 1995 released Internet Phone application by Vocaltec [8] was probably the first VoIP client targeted for commercial use. Based on a proprietary signaling protocol and a proprietary compression technique the Internet Phone application enabled two users using the same application and having similar sound cards to turn their PCs into phones. The Internet Phone offered voice-mail and text chat. To enable the users to communicate with other users the vendor maintained a global directory, which listed other users of the Internet Phone application. Further, one could directly call another user using the email address or IP address of that user –if known.

The voice quality provided by the Internet Phone was lower than that of traditional phones. This was a result of the low bandwidth compression technique used, losses in the overloaded Internet and delays caused when processing the voice at the PCs. However, with the high costs of long distance calls the idea of free calls lured a fair number of users.

Besides the commercial success of the Internet Phone itself, this venture into the VoIP market had two important contributions. On the one side, end users started to become aware that there are other options for making phone calls than what is offered by telecom operators. This awareness paved the way for other companies to roll out VoIP solutions based on standardized protocols and new business models. On the other hand, the technology behind the Internet Phone contributed to a great extent to the development of the H.323 suite of standards, see Sec. 5.

Vocaltec was also the first company to demonstrate a VoIP to PSTN gateway and thereby launch the PC to PSTN services as well as the VoIP trunking business, see Sec. 6.1.

4 Turn of Millennium: Protocol Wars

By the end of the century, telecommunication industry was working hard on standardized solutions. Several competing standards for VoIP communication protocols began to claim their place on the planet. In most of the standardization debates passionate technological and religious arguments played a role. However, it was eventually the market forces that gave conclusive answers.

The battle over centralized control was one of the very first on the VoIP battlefield. Proponents of the telco-leaned paradigm advocated "dumb" telephones controlled by "smart" network elements, frequently referred to in marketing speak as "softswitch". The outcome of this approach was the twin "**master-slave**" protocols MGCP and Megaco/H.248 [9]. These protocols allow network components to control telephones in a centralized manner. The network elements control, when a telephone starts

ringing, propagate notifications on answered calls, tear down established calls, and so on. Opponents argued that innovation advances faster in the end-devices and would be impeded by a strict control protocol. Instead they offered the "**end-to-end**" vision based on smart end-devices. Such devices can set up media-rich sessions between each other with the help of application-unaware infrastructure. Eventually, MGCP and Megaco gained noticeable adoption only in the PSTN realm as protocol for decomposed PSTN gateways. Most native IP end-devices followed, however, one of the decentralized end-to-end protocol designs.

The decentralized protocols, ITU-T's H.323 and IETF's SIP, battled bitterly against each other. ITU-T, the telecom standardization body, started off and published the H.323 standard in November 1996. The protocol family defined in this standard largely borrowed from the ISDN protocols for sake of seamless PSTN interoperability. The Internet community in the IETF accepted the challenge and published a competing standard called Session Initiation Protocol (SIP) in March 1999. This protocol mimics Web's client-server HTTP protocol. The most visible and indisputable difference between SIP and H.323 is encoding. H.323 messages are encoded in a binary form whereas SIP messages are textual and human-readable. SIP advocates also maintained that SIP was built with greater extensibility in mind. This argument certainly affected decision-making process of 3GPP, the mobile phone standardization body. As result, in 2000 3GPP adopted SIP for use in all-IP mobile networks. We believe that the most important argument came from the market few years later. It was the ISPs and ASPs who started the mass consumer VoIP services in 2004. As the SIP "language" was easier to understand for ISPs used to deal with HTTP, SIP eventually prevailed in most deployments.

At the same time, conflict between architectural purists and deployment pragmatics caused years of delay. The conflict's origin had been hard-wired in the IP protocol decades ago: too short IP addressing space. 32 bits were simply too short to match with the dramatic Internet growth. Market's answer was Network Address Translators (NAT) that allow multiple devices to share a single IP address. Purists condemned the NATs as evil because they violate transparency of the Internet and have indeed numerous side-effects. One of them is that servers behind NATs are hard-to-reach, a problem affecting every VoIP telephone behind a NAT. That's because such a phone acts as server when it listens for incoming calls and voice. By then purists were hopeful that NATs would disappear with the arrival of IPv6. Pragmatists were concerned about slow IPv6 adoption rate and were trying to find protocols to get around NATs, such as Midcom, UPnP, STUN and TURN.

Market began to be impatient and delivered two answers before tha standardization efforts were concluded. One of them is the notion of a "Session Border Controller (SBC)", a network box that handles the "NAT problem" for both end-devices and other network equipment. The SBCs mediate both signaling and media in a proprietary way which allows VoIP to traverse NATs. Market availability of the SBCs more or less drove the NAT traversal standardization debate in obsolescence. The other market's answer was skype: skype architects didn't bother with debates about IPv6 and created a proprietary peer-to-peer protocol that can traverse NATs and firewalls. We believe it was this capability which made VoIP largely usable for consumers. As a result, skype sky-rocketed on consumer market in years when the standardization bodies were still trying to find an "architecturally correct" answer.

Next to these major battles, numerous other ones took place in standardization bodies and related to addressing (Email-like versus telephone numbers), Internet-ready codecs, encryption protocols (zRTP versus DTLS), QoS control (tied versus loosely-coupled), integration with messaging (SIP versus jabber), and others. In most of these matters practicability for service providers and especially PSTN backwards compatibility often determined the outcome. As a result most VoIP users are reachable today by a telephone number and the most interoperable codec remains the proven but wasteful G.711.

5 Telecom VoIP Standard: H.323

The origins of H.323 date back to 1994, when (at that time) Study Group 15 of the ITU-T decided to extend their perspective on multimedia communication (especially video telephony and conferencing) to include local area networks. SG 15 had, at this point, developed the Recommendations for video telephony over ISDN (H.320), combining 1 – 30 ISDN B channels to create a multimedia pipe of up to 2 Mbit/s and running a bit-oriented multiplexing protocol on top to differentiate between audio, video, data, and control channels. SG15 had already expanded their scope to native support for ATM networks, officially termed "B-ISDN" in H.321 (primarily driven by institutions from Japan) and was looking at video communication over modem connections with H.324. An extension of this is known as H.324M, the low-overhead equivalent for multimedia over circuit-switched cellular networks, which found its application for video calls in early (pre-IMS) releases of UMTS. A parallel (low-profile) activity was also defining how to run video communications over "local area networks with guaranteed quality of service", i.e., isoEthernet (IEEE 802.9), which led to H.322 but remained without practical relevance.

There are several myths about the ITU-T and some of these may hold true to some extent: two prominent ones state 1) that the work progress is slow (because of bureaucracy and long meeting cycles) and 2) that the work is driven by the telecom operators. Interestingly, none of those held for the group designing H.323: Concerning 2), the group was dominated by equipment vendors who, coming from H.320, wanted to build interoperable products also for IP-based local area networks. With a few exceptions, telcos played mostly just an observing role in the beginning. As for 1), the companies were eager (if not required) to move quickly to get their products into the market. This resulted in a tremendous effort put into H.323 and yielded the completion of the first functionally complete specification in about 10 months (just a bit more than a year including the formal voting process, 1996). The strong efforts continued to a functionally enhanced and partly optimized H.323v2 (1998) and subsequent extended revisions H.323v3 (1999) and H.323v4 (2000). With especially video conferencing products being shipped—and with the Session Initiation Protocol (SIP) establishing itself as a (supposedly) more promising solution for telephony—the effort reduced and the group went more into a maintenance mode. At this point, the specification was essentially complete and only rather minor functional enhancements took place, the most notable one being the work on NAT and firewall traversal.

5.1 H.323 Series of Recommendations

The original goal of H.323 was extending H.32x-based multimedia communication to endpoints across (local area) IP networks—so that gateway considerations played an important role. Recall that, at that time around 1994, 56k modems were about to be standardized and the non-academic wide-area Internet was essentially unusable for multimedia communication. Thus, it was not the creation of a new multimedia communication architecture that guided the design but gatewaying and legacy interoperability considerations—paired with the need for a certain "cultural compatibility" to obtain support in the video conferencing industry and acceptance in the ITU-T. One specific consequence of this was that NAT traversal was not an issue: local endpoints would connect to their gateway and then be routed via the telephone network to the target site, where they would be again gatewayed to their final destination. The idea of using H.323 for Internet telephony evolved only over time.

Typically, an H.32x series required multiple specifications, as does H.323:

- the systems framework (H.323) that provides the overview and defines the interactions of the diverse components comprising one endpoint;
- the signaling protocol (H.225.0) that defines the interaction between endpoints (for ISDN and PSTN, H.221 and H.223 would also do the channel multiplexing);
- the media control channel (H.245) for capability negotiation, setup/teardown of media channels, and further control operations; and
- specifications for multiparty conferencing support using central *multipoint control units* (MCUs) (H.239, H.241).

When the H.323 system specification was started, this did not happen in a vacuum:

- H.245 was incorporated as an elaborate—actually: too elaborate, given that many features were generally not used in the end—capability and media control channel since all other systems specifications (except for H.320) were using it as well and mappings to H.320 were already in place.
- With gatewaying to ISDN in mind, Q.931 was chosen as the basis for the call signaling channel. While the protocol state machine could be considered ok for the purpose, the protocol messages required substantial extensions. H.225.0 evolved as the equivalent call signaling protocol borrowing heavily from Q.931.
- The basics for multiparty conferencing operation were adapted and enhanced from the basic conferencing functionality defined for H.320.
- Mapping between the different H.32x systems was defined in H.246.

Moreover, SG15 also standardized audio (ITU-T G.7xx series of Recommendations) and video codecs (H.261, H.263, and H.264). These codecs could be used without changes. Nevertheless, over time, the growing relevance of packet-based communication led to a shift in codec standardization, away from mere bit error tolerance to considering packet boundaries and especially packet losses in the coding process and data representations. The IETF Real-time Transport Protocol (RTP) was chosen as the media transport and so were the established payload formats—followed

by a fruitful interaction between ITU-T codec people and the IETF to define appropriate ones for new codecs.

One important difference to all other H.32x series Recommendations was the need for an address resolution mechanism: E.164 phone numbers (as used for H.320 and H.324) would need to be mapped to IP addresses—and a similar approach could be taken for aliases or H.323 URIs for users. Therefore, a *Registration, Admission, and Status* (RAS) channel was added to H.225.0. Endpoints would use H.225.0 to register themselves with an entity called *Gatekeeper* and to ask the Gatekeeper to resolve addresses. When performing address resolution, the Gatekeeper could also police or prioritize call requests, for example for local resource management.

The above specifications were complemented over time by a number of additional one as H.323 grew: for security (H.235 series), *supplementary services* for telephony (H.450 series), media gateway control (H.248), service creation, call reporting, intra/inter-domain gatekeeper communication, and robustness (diverse annexes), broadcast-style conferences (H.332), an extensibility framework (H.460), and directory services (H.350 series), among others. Data communication may be added, e.g., using the T.120 series of Recommendations.

Except for the Real Time Transport Protocol (RTP) used for the transport of media data and the basic PDU formats inherited from Q.931, all H.323 specifications use the *packed encoding rules* (PER) of ASN.1 (X.68x and X.69x) for protocol encoding.

5.2 System Overview

Figure 1 provides an exemplary overview of H.323 system components and their interactions across two domains. Endpoints (EP), gateways (GW), multipoint control units (MCU), and media servers (MS) are the nodes originating and terminating H.323 calls, i.e., call control channels and media streams.

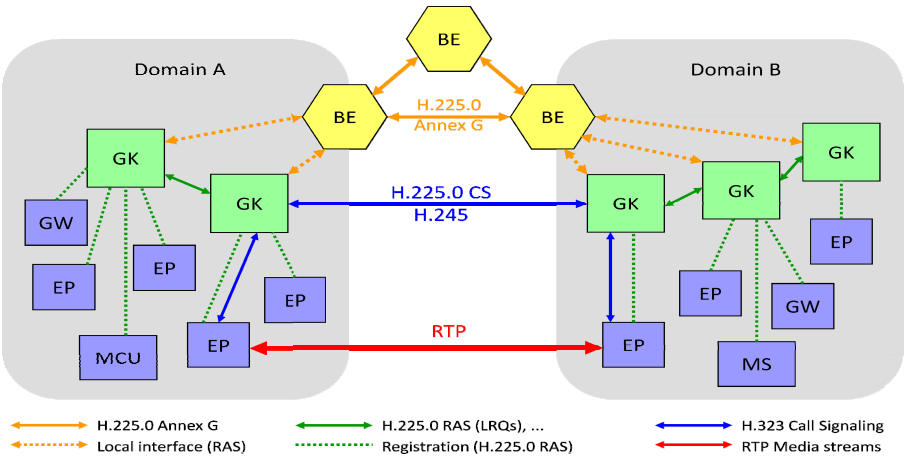

Fig. 1. H.323 system overview

In order to become reachable they register with their local gatekeepers that are also responsible for address resolution and call admission control. Since multiple gatekeepers may exist per domain, they may cooperate to resolve local addresses using an intra-domain inter-gatekeeper protocol. Interaction with other domains (e.g., to exchange call routing information) may happen via dedicated Border Elements (BE) that perform (similar to border routers in IP) policy-based information exchange, but they do not participate in the actual per-call signaling. In practice, albeit implemented, very few people actually use BEs; instead, one rather would find peering relationships directly between gatekeepers with all necessary policies implemented in those.

Media always flows directly end-to-end while the call signaling and capability negotiation channels may or may not through a gatekeeper, depending on the chosen call model.

5.3 Call Models

H.323 defines three different call models: 1) In the *direct call model*, the two H.323 endpoints establish direct TCP connections for call signaling and H.245 (one each). The gatekeeper is (usually) only involved in the beginning and at the end of a call for address resolution and to obtain and release resources. Alternatively, a *gatekeeper-routed call model* can be used, in which 2) the call signaling channel runs through the gatekeeper(s) and the H.245 channels directly end-to-end (this model was not specified and left for further study) or 3) both call signaling and H.245 are routed via the gatekeeper. The calling party's gatekeeper dictates the local model upon address resolution where it either returns its own address or that of the remote peer; it may also decide to target the remote gatekeeper with the call setup (rather than directly the remote endpoint if call routing information demands so). The called party's gatekeeper can take a similar decision and order its endpoint to redirect the call signaling to itself to enforce the gatekeeper-routed call model also on this side. Since the gatekeeper is optional, there is arguably another model: 4) Without a gatekeeper involved there is no RAS channel and endpoint will interact directly, using external mechanisms (e.g., DNS-based) for address resolution; this is, however, mostly limited to environments where devices have fixed addresses so that there is no need for highly dynamic addresses resolution, e.g., in some distance education setups.

Being the ones fully defined, only 1) and 3) were commonly used, but variants of 2) appear to be used in practice when decomposing MCUs into Media Processors (MPs) for media switching/mixing and Media Controllers (MCs) for handling the signaling; in such a case, H.225.0 would be terminated at the MC whereas the H.245 control channel go to the MPs. Figure 1 shows the call signaling and H.245 connections found in model 3).

5.4 End-to-End Design

All video communication Recommendations of the H.32x series treated the network as a bit pipe (as ISDN channel, ATM virtual circuit, or a modem connection):

obviously, since – coming from the telephony domain – the end user would be expected to know the number to call and the network would do the rest. All media multiplexing, control signaling (naturally besides channel setup and teardown), and all media themselves would run in-band end-to-end.

H.323 did not deviate from this concept of assuming a dumb network: assuming IP connectivity underneath, it borrowed the relevant IETF work for transport (UDP, TCP, RTP) to have communication happen end-to-end. And with a packet-based network, H.323 did no longer require to provide its own multiplexing scheme as H.320 and H.324 did. The only infrastructure element (a host from a network perspective) that H.323 relies upon is the gatekeeper. But, of course, the gatekeeper as well as MCUs, media servers, and gateways could be run by operators as well as by enterprises or, ultimately, by cloud service providers.

5.5 Functional Evolution

As noted above, H.323 started out as a system for extending video conferencing into LANs. However, once the basic system architecture and the IP-based signaling standards were in place, the strongest influence came from vendors and service providers interested in Voice-over-IP and video and multiparty communication got out of focus for quite a while [10]. What followed was a rush for adding all kinds of features mimicking telephony services in the IP world and making those "work" within the confines of the architectural framework devised for the original design goals, with the limitations of telephony signaling, and with the burden of extensive multimedia capabilities.

It was this phase—maybe as a result of the success of the early development and the take-up of industry interest—during which the fairly clean architectural design of H.323 got lost. A flood of contributions suggested manifold features often independently that needed to be bolted on mostly one at a time, so that it turned out impossible to maintain the architectural integrity and principles of the specifications and develop a clear structure for extending the system and at the same time keep up with the strong industry demand [11]. The result was a series of extensions in fairly rapid succession, leading to several revisions of the base specification and the development of numerous additional ones (as annexes and separate specifications). It was probably this phase during which H.323 lost quite a bit of its appeal becoming way too complex for the supposedly low-cost telephony world as the specifications grew quickly in size and number.

After 2000, the specifications stabilized and operational and maintenance features (robustness mechanisms, a MIB, NAT traversal, etc.) were added. While H.323 was leading the development, numerous of its features (most prominently probably user registration) were also adopted by SIP. At the later stages the development could possibly be characterized by an inverse "me too" strategy, in which H.323 received suggestions for features developed for SIP before, including presence and instant messaging, among others. But only few, such as the (natural) use of URIs, were actually adopted [12].

5.6 H.323 in Retrospect

As noted above, H.323 has "earned" a reputation as being (too) complex and adoption especially by academia (and thereby future engineers) was quite limited – it took quite a while before open source projects (such as openh323 or opengatekeeper) took up. There are probably many reasons, but four—in the authors' opinion—important ones include: 1) The semi-closedness of the design process and limited access to specifications might have brought an advantage to the (paying) ITU-T members involved in the design, but hurt the specification adoption in the long run. 2) The extensive use of complex data notation for exchange formats and their binary encoding made implementation and manual debugging extremely hard; the de-facto need to buy expensive tools to even start development is a non-started for university and open source projects. 3) The tool-based design also simplified the notation for complex ideas where the connection between a specification and the resulting implementation complexity was lost so that modesty in protocol design was not encouraged. This holds for encodings but also for the implications for protocol state machines. 4) Finally, H.323 had too many cooks dragging the specification to do too many things at the same time, harming architectural integrity and thereby contributing to further extensions getting more and more complex [13].

H.323 has also earned the reputation as being telco technology—which is closer to a fairy tale (or counter-marketing) than reality. Nevertheless, H.323 has clearly been designed coming from a telecom background and clearly missed the opportunity of introducing a paradigm shift in communication that capabilities of which were surely inherent in IP.

To sum up, H.323 has been a tremendous commercial success in the video conferencing market. It solidified the customer base, aggregated the vendors and made the market for IP based video conferencing. While the path was difficult, and many of the decisions were flawed, H.323 provided a platform that provided interoperable video communications for the last decade.

Finally, it is worth noticing that the development of H.323 and experience gained with it surely accelerated the paradigm shift towards IP-based multimedia and the rapid development of SIP (competition is healthy, after all).

In the long run, the specific technology may become immaterial at least for endpoints as we are moving towards a "webby" model in which the end point (i.e., a web browser or equivalent) just downloads the code to interact with a remote peer when engaging on an interaction. This paradigm shift is aggressively pursued by RTC web [14].

6 Internet VoIP Standard: Session Initiation Protocol (SIP)

By the mid nineties the Internet had established itself as a consumer product. The number of users buying PCs and subscribing with an ISP for a dial-up access was increasing exponentially. While mostly used for the exchange of Email, text chatting and distribution of information VoIP services based on proprietary solutions as well as H.323 started to gain some popularity.

While there is no organization that is formally responsible for the Internet as such the Internet Engineering Task Force (IETF) is playing the role of the standards

organization of the Internet. The IETF has among others produced the needed specifications for the transport and routing of packets in the Internet as well as the protocols for Email, address resolutions and all other kinds of applications and services running on top of the Internet.

At this stage the IETF has already produced different protocols needed for enabling VoIP. The Real-Time Transport Protocol (RTP), see RFC 1889 [15], enabled the exchange of audio and video data. The Session Description Protocol, see RFC 2327 [16], enabled the description of multimedia data. With the Session Announcement Protocol (SAP), see RFC 2974 [17], it was even possible to distribute the necessary information to watch a certain publicly broadcasted audio and video session. Further, the first applications, mostly open source, for the sending and reception of real-time audio and video data were available.

Those days, the procedure for establishing a VoIP call between two users based on the IETF standards would look as follows: The caller starts his audio and video applications at a certain IP address and port. The caller then either calls the callee over the Phone or sends him an Email to inform him about the IP and port address as well as the audio and video compression types. The callee then starts his own audio and video applications and informs the caller about his IP and port number. While this approach was acceptable for a couple of researches wanting to talk over a long distance or for demonstrating some research on Quality of Service of media compression this was clearly not acceptable for the average Internet user.

The Session Initiation Protocol (SIP), see RFC 3261 [18], was the attempt of the IETF community to provide a signaling protocol that will not only enable phone calls but can be also used for initiating any kind of communication session. Hence, SIP can be used for VoIP just as well as for setting up a gaming session or a control session to a coffee machine.

In general a SIP-based VoIP service consists of user agents (UA), proxies and registrar servers. The UA can be the VoIP application used by the user, e.g., the VoIP phone or software application, a VoIP gateway which enables VoIP users to communicate with users in the public switched network (PSTN) or an application server, e.g., multi-party conferencing server or a voicemail server.

The registrar server maintains a location database that binds the users' VoIP addresses to their current IP addresses.

The proxy provides the routing logic of the VoIP service. When a proxy receives a SIP request from a user agent or another proxy it also conducts service specific logic, such as checking the user's profile and whether the user is allowed to use the requested services. The proxy then either forwards the request to another proxy or to another user agent or rejects the request by sending a negative response.

With regard to the SIP messages we distinguish between requests and responses. The INVITE request used to establish a session between two users is a session initiating request. The BYE sent for terminating this session would be an in-dialog request. Responses can either be final or provisional. Final responses can indicate that a request was successfully received and processed by the destination. Alternatively, a final response can indicate that the request could not be processed by the destination or by some proxy in between or that the session could not be established for some reason. Provisional responses indicate that the session establishment is in progress, e.g., the destination phone is ringing but the user has not picked up the phone yet.

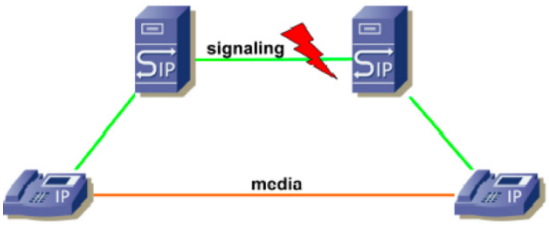

Fig. 2. SIP trapezoid model

As illustrated in Figure 2 the actual topology of a server-mediated call between two SIP phones has the SIP trapezoid in its heart. On the remote sides of the trapezoid, there are the SIP telephones belonging to their respective call participants. Each phone is registered with its SIP server. The registration happens when a phone is turned on and re-registers periodically later to prove it remains reachable. When a caller later decides to dial his peer, his telephone sends a SIP INVITE request through his SIP server. His SIP server looks up the IP address of the server responsible for the destination domain using DNS and forwards the request there. The destination server eventually relays the request to its final destination, the telephone of the previously registered called party.

The initial vision of SIP foresaw a world in which Network Address Translators (NAT) and firewalls were not used, users were more or less trusted and all logic for any type of service was supposed to be located at the end devices. This simplified view of the world led to simple specification and easy implementation. However, with a world full of NATs, firewalls, untrusted devices and subscribers used to some set of supplementary services, this meant that SIP still had to go through endless discussions and a long standardization path before a deployable version was finally available. What started in the mid nineties as a simple solution for session establishment is still a continuing process today and has led to a set of specifications that describe session establishment, NAT traversal, transport of DTMF tones, various addressing schemes, security and application of SIP to various other services such as messaging.

First commercial deployments of SIP-based VoIP services started appearing at the beginning of this century. Internet Service Providers (ISPs) started offering VoIP services as an additional service to Email and messaging on top of their broadband access lines. Unlike the VoIP deployment 10 years before, the availability of moderately priced broadband access lines and the seamless integration of SIP into DSL and cable access devices enabled the ISPs to rapidly increase the number of SIP-based VoIP users from a couple of thousands at the beginning of the century to millions today [19].

6.1 Trunking and SIP-I

SIP was originally designed with an end-to-end VoIP model in mind with the caller and/or the callee being connected to the Internet. While this model is popular with ISPs offering their customers broadband Internet access, large telecom operators have been more reluctant to replace their classical PSTN-based service with a VoIP service. With the classical telephony minutes still making up the largest part of the

operators' revenues there is no clear need for replacing one voice service with another one –even if it was based on SIP. However, IP based technology still has different advantages for large operators; namely in reducing the costs of the backbone. With telephony switches reaching their end of life operators started replacing their SS7 based infrastructure with an IP based one.

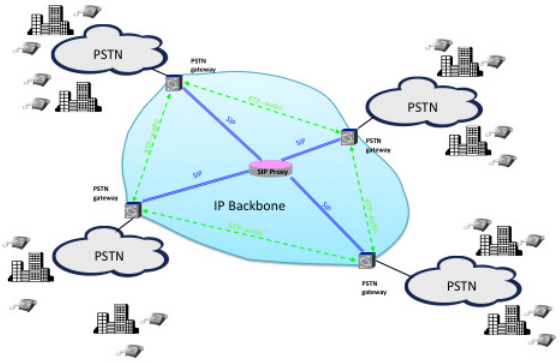

Fig. 3. SIP trunking scenario

In the SIP trunking scenario the original PSTN signals are attached to SIP messages and preserved across the Internet haul. The SIP-based tunneling is known as "SIP-T", RFC 3372 [20], its ITU-originating version as "SIP-I" [21]. Choice of bandwidth-saving codec is important, since with trunking volume bandwidth cost is considerable.

SIP trunking has been from historical and volume point of views the most successful use case so far. With SIP trunking, see Figure 3, the actual calls are both originated and terminated in the PSTN. Only in the middle of the path the call traverses the Internet (or sometimes a private IP network) through SIP-to-PSTN gateways. Call participants remain frequently unaware of the fact that they are using SIP. A particular reason for success of this scenario is twofold. First, it is economically viable: with large traffic volume the VoIP saving can return the investment quite quickly. Secondly, it is easy to integrate. In many cases it takes PSTN gateways from a single vendor, all configured very similarly and interoperating with each other smoothly. The different gateways can be connected with each other in a full-mesh topology or a star topology as depicted in Figure 3.

Proprietary solutions for trunking began to be in operation from about 1995 on. Competing solutions to SIP-based trunking are based on transporting of PSTN signaling using the Sigtran, see RFC 2719 [22] or BICC [23] protocols, remain, however, out of scope here.

6.2 IP Multimedia Subsystem (IMS)

A great push for SIP was its endorsement by the mobile-phone standardization body, 3GPP, as part of its strategy for deploying Internet in mobile networks in the beginning of the millennium.

The Third Generation Partnership Project (3GPP) is a collaboration agreement that was established in December 1998 between a number of telecommunications standards bodies; namely ARIB, CCSA, ETSI, ATIS, TTA, and TTC. Mainly looking at the needs and requirements of mobile operators, the 3GPP first specified the IP Multimedia Subsystem (IMS) as a service architecture combining the Internet's IP technology and wireless and mobility services of current mobile telephony networks. Through the work of the TISPAN, the IMS architecture was extended to include fixed networks as well. The Telecoms and Internet converged Services and Protocols for Advanced Networks (TISPAN) is a standardization body of ETSI, specializing in fixed networks and Internet convergence and was formed in 2003.

IMS [24] builds on Internet Engineering Task Force (IETF) protocols like Session Initiation Protocol (SIP) and Session Description Protocol (SDP). However, for a SIP based solution to replace the current mobile and fixed telecommunication infrastructure it needs to offer the same capabilities; namely secure and efficient access to high quality multimedia services regardless of the user's location. The IMS specifications are, hence, mainly based on the IETF SIP specifications but add some new architectural and functions extensions:

- Functional distribution: The IETF SIP specifications mainly foresee a SIP proxy for the routing of SIP messages. The IMS specifications define different instances of so called Call Session Control Functions (CSCF):
    - P-CSCF (Proxy-CSCF): The P-CSCF is the first point of contact between the IMS terminal and the IMS network. All the requests initiated by the IMS terminal or destined to the IMS terminal traverse the P-CSCF.
    - *I-CSCF (Interrogating-CSCF):* The I-CSCF retrieves user location information and routes the SIP request to the appropriate destination, typically an S-CSCF.
    - *S-CSCF (Serving-CSCF):* The S-CSCF maintains a binding between the user location and the user's SIP address of record (also known as Public User Identity). Like the I-CSCF, the S-CSCF also implements a Diameter interface to the HSS.
    - *HSS (The Home Subscriber Server)*: contains all the user related subscription data required to handle multimedia sessions.
- QoS control: One of the major differences between VoIP and traditional telephony services is the decoupling of the media and signaling paths. On the one hand, this decoupling allows for the establishment of new business models in which a service provider can offer VoIP services without having to own the physical network itself. On the other hand, this implies that the provider will not be able to support any kind of traffic prioritization or resource reservation that would be needed to offer VoIP services with a predictable quality of service level. In the IMS the session establishment process is coupled tightly with the reservation of resources required for achieving the desired QoS level [25]. Further, certain IMS SIP components have an additional interface that allows them to control and communicate with the underlying physical infrastructure.

- Roaming support: The IMS introduces the concept of home and foreign service providers in a similar manner to the current mobile telephony system. A home service provider maintains a contractual relation with the user as well as various user related information required for authenticating the user and offering him certain services. A foreign provider is the provider offering access to the IMS services in geographical locations not covered by the home provider. In order to enable a user to roam to geographical locations not covered by his own provider and still get access to IMS services in a simple and transparent way, roaming agreements between the home and foreign providers are established. These agreements govern whether a user is allowed to access IMS services in a foreign location and the costs of such access.
- Security: The native security mechanisms of SIP enable the service provider to authenticate the users using HTTP Digest, see RFC 2617 [26]. In case the user wants to authenticate the components of the service provider then the Transport Layer Security (TLS), see RFC 5246 [27] should be used. In order to support roaming, the security model in IMS requires also the establishment of a trust relation between the user and the foreign service provider as well as a trust relation between the foreign provider and the home provider. IMS supports similar authentication mechanisms to those used in current mobile networks as well as digest-based authentication. Further, with the extension of IMS to support fixed networks, additional security mechanisms were specified for IMS that reflect the specific needs and characteristics of these networks [28].
- Network-Centric Call Control: Current mobile telecommunication networks provide different capabilities that enable the operators to terminate a user's active communication session when the pre-paid account of a user becomes empty or terminate his subscription if he did not pay his bill for some time. To offer similar capabilities, the SIP components used in an IMS network maintain sufficient dialog and registration information so as to be able to terminate a running session by sending a BYE request to the caller and callee.

While the IMS was initially designed for mobile operators it was first deployed by fixed-line operators. With a profitable business of selling telephony minutes the incentive to replace one technology that provides telephony services with another one was not high. This is especially the case if the new technology is even less efficient in utilizing the limited frequency spectrum and requires all subscribers to either install new applications or even buy a new mobile phone. Fixed operators are on the other hand facing stiff competition from service providers offering bundled packages of high speed Internet access and telephony services. With sufficient access bandwidth and the VoIP clients already integrated into the access devices, IMS offers fixed-line operators a natural solution that reduces the costs and enables a better positioning of the operators.

The advent of the Long Term Evolution (LTE) technology and the all IP Enhanced Packet Core (EPC) networks [29], is changing this. With the increased importance of mobile data services and the availability of high bandwidth wireless networks the number of users moving to more powerful smart phones in increasing together with

the interest in VoIP and IMS. IMS is now being considered as the appropriate solution for providing Voice over LTE (VoLTE) services.

7 Reality Beyond Standards: SKYPE

SIP services are based on a client-server model with the servers being operated by a service provider. Hence, similar to PSTN networks, the provider operates a centralized infrastructure that is responsible for user authentication, routing of the signaling traffic and providing additional services. Skype is based on a more distributed architecture based on an overlay peer-to-peer (P2P) network, similar to its file sharing predecessor KaZaa [30]. There are three main components in the Skype network [31]:

- The Skype login server (LS) is one of the few central components of the network. Every user is authenticated through the login server to gain access to the network.
- A Skype client (SC) provides all user functionality to access the network, that is login, initiating and receiving calls, instant messages and file transfer.
- Super Node (SN). A super-node is an SC that is well connected to the Internet and provides additional functionalities to other SN and SC. A super node performs routing tasks such as forwarding requests to appropriate destinations and answering to queries from other SCs or SNs. The SN can also forward login requests in case the login server is not directly reachable from an SC. Additionally, the SN provides media proxying capabilities for other SCs that have only restricted internet access, be it through Network Address Translation (NAT) or restricted firewalls.

To log in to the network, an SC tries to contact one or more Super-Nodes (SN). The code of the clients already contains a list of possible Super-Nodes that are provided by Skype itself. These bootstrap SNs are contacted upon first launch of the client to gather an updated and more extensive list of currently available SN.

Except for some dedicated operations like authentication, user list storage or Skype-to-PSTN connectivity, there are no further central servers in the Skype network. All other operations, e.g. user searches or message forwarding are performed in a decentralized way by the super-nodes.

Skype is arguably the most successful VoIP service. Skype has taken a different approach than most of other VoIP players. Skype invented its own protocol, which is highly proprietary. It is secured against common security threats as well as reverse engineering.

Probably the main reason for the success of Skype is the approach it has taken for rolling out its VoIP service. In the case of SIP and H.323 a lot of energy went into the specification of the signaling protocols. Aspects of deployability and user interface were only considered in the second round. Skype rolled out a complete service with an easy to install and use application, low bandwidth and high quality voice encoding and a highly flexible firewall and NAT traversal solution.

The proprietary mode in which Skype has gotten traction is, however, its greatest weakness too. Many technological companies are reluctant to support closed walled-

garden environments and prefer open standard instead. It remains to be seen how the success is split between open SIP and proprietary Skype over the coming years. Further with less than 2 USD average spending per subscriber [32] the business plan of Skype might not be attractive for operators used to much higher margins.

8 Converging VoIP and Web: WEBRTC

While first designed as the interface to display information provided by web servers, web browsers are now used as the access to social networks, the interface to online games and for exchanging emails and messages as well as streaming audio and video content. Thereby web browsers have become the main access interface to the Internet and have actually become synonymous with the Internet itself for a large portion of the Internet users.

Up until recently the communication capabilities of web applications were limited to either text-based communication such as messaging or email or non-real-time audio and video, e.g., streaming. The combination of real time services such as a voice call or a video conference with a web application was only possible using either a separate application or proprietary plug-ins that lack open specifications, interoperability and are often limited to certain platforms [33]. Using a separate application would mean leaving the browser and launching a new application. Thereby there can be no real integration of the content presented by the browser and the real time content. Solutions based on plugins provide tighter integration between the real-time content and the provider's web pages. However, plugins such as Flash are proprietary and do not work in all environments. In particular Flash does not work over IOS used for iPhones for example. Another issue with the Flash technology is its centralized model. A Flash plugin that was downloaded from domain X can only communicate with a server in domain X. This means that an application provider that is offering a number of applications in the form of Flash pluging will have to deal with all the signaling and media traffic generated by the plugin. This restriction was introduced to prevent a malicious application from sending traffic to some destination and hence attacking that destination.

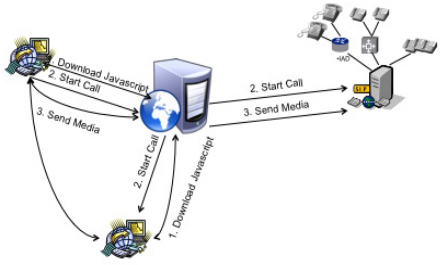

Fig. 4. High Level WEBRTC Framework

The major standardization groups responsible for the advancement of the Internet protocols and applications have launched the HTML5 and real-time web (WebRTC) initiatives to complement web applications with real time media features. New browser capabilities are being defined in HTML5 [34] for video conferencing and peer-to-peer communication. New working groups have been created in W3C and IETF to define elements of real-time communication in the browser[35,36,37]. The specified WebRTC framework is based on the following main parts:

- Browser API: To provide application developers with the ability to send and receive audio and video streams directly from a browser, browsers must be enhanced with capabilities for controlling the local audio and video devices at the computing device at which the browser is running. These capabilities will then be exposed to the application developers through a well-defined application programming interface (API).
- Web application: The typical mode of running a web application is for the user to download a Javascript from a web server. This script runs then locally at the user's system but interacts with the web server for executing the application logic. The web server can instruct the Javascript to conduct certain actions and the script can send feedback information to the web server.
- Web server: The server provides the Javascripts for the users and executes the application logic.

With such a framework a web telephony application would be developed as a Javascript that is provided at a web server, see **Figure** 4. A user wishing to use this application downloads the script. When making a call the Javascript then informs the web server about the call destination and the web server contacts the final destination. Once the callee answers the call the web server forwards the response to a Javascript running at the caller's system. The Javascript now instructs the browser to use the local audio and video devices to exchange audio and video content with the callee.

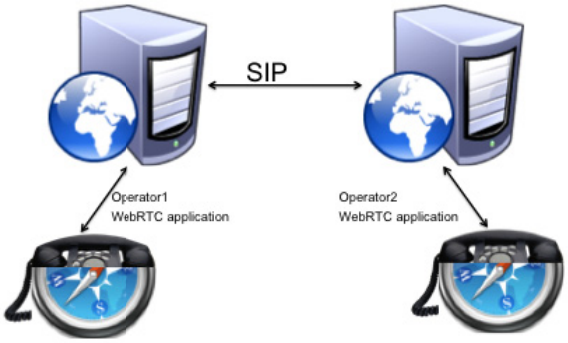

Fig. 5. WebRTC Trapazoid

In order to ensure that the type of applications that can benefit from the integration of real-time services with the browser is only limited by the imagination of the developers, the WebRTC framework is only defining the API to be provided by the browser as well minimal security requirements needed to avoid the misuse of WebRTC applications for initiating denial of service attacks.

In order to avoid the restriction of a centralized model that is used with the Flash technology, the WebRTC framework indicates that a browser can send data to a host other than the one from which the application was downloaded if that host consents to receiving the data.

To enable browsers using different application providers to communicate with each other (e.g. a user logged in to Facebook wants to call someone that is logged in to linkedin) a so called RTC trapezoid, see **Figure** 5, can be used. In this case the two providers use a widely used VoIP signaling protocol in between such as the Session Initiation Protocol [38] to federate between them. However, each of their respective browser-based clients signals to its server using proprietary application protocols built on top of HTTP and Websockets.

WebRTC technology should not be mistaken for yet another telephony service. Dedicated applications and devices based on Skype and SIP will continue to be the preferred way for making phone calls. WebRTC will, however, turn telephony to become one of the many features offered by a web application instead of being a dedicated service.

9 Summary

It is obvious that there is no clear winner in the VoIP arena. While not becoming the next PSTN, H.323 continues to exist, especially in video-oriented installations. SIP dominates the trunking deployments and is the first choice for ISPs and ASPs. Skype uses its proprietary protocols for on-net calls and SIP to reach PSTN, and is reportedly the largest provider of cross-border voice communications [39]. Latest efforts concentrate on a more tied integration of web services. Noticeable examples include integration of skype with facebook, and standardization of VoIP embedded in web browsers known as WebRTC.

The different VoIP standards continue to exist next to each other as well as next to PSTN technology and it is our belief that this will be the case for some time to come.

References

1. http://www.ctia.org/advocacy/research/index.cfm/AID/10323
2. Kleinrock, L.: Information Flow in Large Communication Nets. RLE Quarterly Progress Report (July 1961)
3. Baran, P.: On Distributed Communications Networks. IEEE Trans. Comm. Systems (March 1964)
4. Cohen, D.: Specifications for the Network Voice Protocol (NVP), RFC741(1977)

5. Rabiner, L.R., Schafer, R.W.: Digital Processing of Speech Signals. Prentice-Hall(SignalProcessingSeries) (1978)
6. Postel, J.: Internet Protocol. RFC791 (1981)
7. Crocker, S.: Protocol Notes. RFC 36 (1970)
8. http://www.vocaltec.com
9. International Telecommunication Union. Gateway control protocol: Version 1. ITU-T Recommendation H.248.1 (March 2002)
10. Interestingly, a similar observation could be made during the evolution of SIP in the late 1990s and early 2000s
11. SIP also shared this fate: a flurry of proposals came about just after the initial revision of SIP became RFC in February 1999 and strong arguments were needed to maintain some architectural integrity—only to be given up later with the demand for operator-administered middlebox control
12. H.323 did not adopt presence and instant messaging, though, leaving those to XMPP
13. In the end, quite a few of the specifications published did not matter in the real world because they were not implemented—yielding another similarity to SIP
14. https://datatracker.ietf.org/wg/rtcweb/charter/
15. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications (RFC1889). IETF (1996)
16. Handley, M., van Jacobson: SDP: Session Description Protocol (RFC 2327). IETF (1998)
17. Handley, M., Perkins, C., Whelan, E.: Session Announcement Protocol (RFC2974). IETF (2000)
18. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol (RFC 3261). IETF (2002)
19. http://www.ilocus.com/content/report/global-voip-market-2010-11th-annual-update
20. Vemuri, A., Peterson, J.: Session Initiation Protocol for Telephones (SIP-T): Context and Architectures (RFC3372). IETF (2002)
21. Interworking between Session Initiation Protocol (SIP) and bearer independent call control protocol (BICC) or ISDN user part (ISUP). ITU-T Rec. 1912.5 (2004)
22. Ong, L., Garcia, M., Schwarzbauer, H., Coene, L., Lin, H., Juhasz, I., Holdrege, M., Sharp, C.: Framework Architecture for Signaling Transport (RFC2719). IETF (1999)
23. Bearer Independent Call Control Protocol (BICC). ITU-T Rec. 1902 (2003)
24. 3GPP, TSG SSA, IP Multimedia Subsystem (IMS) – Stage 2, TS 23.228
25. 3GPP, TSG SSA, End-to-end quality of sevice (QoS) signaling flows. TS29.208
26. Franks, J., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., Stewart, L.: HTTP Authentication: Basic and Digest Access Authentication (RFC2617). IETF(1999)
27. Dierks, T., Rescorla, E.: The Transport Layer Security (TLS) Protocol. Version 1.2 (RFC5246). IETF (2008)
28. Sisalem, D., Floroiu, J., Kuthan, J., Abend, U., Schulzrinne, H.: SIP Security (2009)
29. Lescuyer, P., Lucidarme, T.: Evolved Packet System (Eps): The LTE and SAE Evolution of 3G UMTS. Wiley Publishing (2008)
30. Ross, K.W., Liang, J., Kumar, R.: The kazaa overlay: A measurement study. Computer Networks 49, 6 (2005)
31. Ehlert, S., Petgang, S., Magedanz, T.: Analysis and signature of Skype VoIP session traffic. In: 4th IASTED International (2006)
32. http://www.sec.gov/Archives/edgar/data/1498209/000119312511056174/ds1a.htm

33. http://download.macromedia.com/pub/labs/flashplayer10/flashplayer10_rtmfp_faq_070208.pdf
34. Hickson, I.: HTML5. Web hypertext application technology working group, http://whatwg.org/html
35. Web real-time communications working group charter. W3C (December 2010), http://www.w3.org/2010/12/webrtc-charter.html
36. RTC-Web IETF working charter proposal (March 2011), http://rtcweb.alvestrand.com/ietf-activity
37. Rosenberg, J., et al.: An architectural framework for browser based real-time communications. IETF Internet draft. Work in progress (February 2011)
38. Rosenberg, J., et al.: SIP: Session Initiation protocol. IETF RFC 3261 (June 2002)
39. Financial Times, Skype's changing traffic growth (May 10, 2011), http://www.ft.com/cms/s/2/e858ad1c-7b1f-11e0-9b06-00144feabdc0.html#axzz1uyxE2aNp