IMS-centric Evaluation of IPv4/IPv6 Transition Methods in 3G UMTS Systems

László Bokor, Zoltán Kanizsai, Gábor Jeney
Budapest University of Technology and Economics-Department of Telecommunications (BME-HT)
Mobile Communication and Computing Laboratory – Mobile Innovation Center
Magyar Tudósok krt. 2, H-1117, Budapest, Hungary
{goodzi, kzoltan, jeneyg}@mcl.hu

Abstract - **The Internet Protocol is facing version change nowadays, IPv4 (the old version of IP) will be replaced by IPv6 (the new version of IP) in the near future. This transition strongly affects also wireless and mobile architectures due to widespread application of IP-based mobile networking architectures, the continuously increasing number of mobile Internet users, and the emerging convergence of different communication services driven by the IP Multimedia Subsystem (IMS). However both IPv6 and IMS are deeply covered in the existing literature as self-possessed researches, the challenge of provisioning IPv6-based IMS services over 3rd Generation (3G) Universal Mobile Telecommunication System (UMTS) networks as well as related problems and performance issues were not considered so far. In this work, we try to fill this gap and raise attention on the current questions and challenges of the transition from IPv4 to IPv6 in all-IP 3G and beyond multimedia systems. We analyze eight state of the art methods providing IPv6 support in existing mobile telecommunication architectures and evaluate their impacts on the network and service/application performance. In order to achieve this, we designed and implemented a real-life 3G UMTS-IMS testbed, and compared the characteristics of the selected transition techniques with native IPv6 and IPv4 scenarios from an IMS-centric point of view. Our results expose the main benefits and drawbacks of the evaluated technologies and their actually available implementations.**

Keywords - IPv4, IPv6, L2TP, OpenVPN, 6to4, ISATAP, Teredo, NAT-PT, IMS, all-IP, 3G UMTS, performance evaluation, real-life testbed, measurements

I. INTRODUCTION

IPv6 is the new version of the Internet Protocol and expected to be introduced for the wide audience in the next few years. IPv6 comes with a huge amount of improvements compared to IPv4; however it keeps the conceptual basics. For instance, IPv6 has built-in functionality for mobility management, while IPv4 has only an extension for this purpose (and it is usually not implemented). Thus, for mobile networks we believe that the appearance of IPv6 will extend provisioning systems, therefore evaluation of novel and advanced services over IPv6 is essential [1].

IPv6 was built on the same fundaments as IPv4: both represent a best effort service over a packet switched network [2]. Since IPv6 cannot be a global replacement of IPv4 (they are not compatible), it is expected that IPv6 and IPv4 will live together for approximately twenty years. In the short run, devices and networks will be dual stack, having both IPv4 and IPv6 supported. Later, some terminals and network segments might appear to support IPv6 only, and finally IPv4 will be regarded obsolete. Obviously, it must be a very long process. Thus, it is very important to see how IPv6 behaves compared to IPv4 in mobile networks. This article wants to discover some performance metrics of IPv6 in a mobile environment.

As the world tends to apply IP as the sole networking protocol, the role of mobile operators may turn simply into internet service providers. There are three facts, which should not be forgotten: 1) mobile services yields more income than Internet services, 2) mobile networks/services are centralized compared to some distributed internet services (e.g., P2P), and finally 3) distributed services are difficult to charge. Mobile service providers do not want to take a loser position in the next version of mobile networks, so a new centralized entity has been defined: the IP Multimedia Subsystem (IMS) [3]. IMS plays a central role in the network: it provides multimedia services to users, so users must use the IMS to have these services available/operational. Thus, IMS keeps being the centralized entity of mobile networks, where charging can be solved easily. IMS assures the future of mobile operators: using the IMS as an efficient instrument in the work of combining the new all-IP multimedia features with the benefits of IPv6, mobility support and multihoming, it becomes possible to provide an almost unlimited range of advanced, interactive multimedia services even for future scenarios.

One of the core aspects of the IP Multimedia Subsystem is the convergence on Internet protocols such becoming the main delivery platform for multimedia services throughout every kind of possible access networks. The technical background of this convergence is built upon two protocols, namely IP (v4 and v6) for data transport and Session Initiation Protocol (SIP) 0 for the negotiation and management of sessions. Since all users in an IMS enabled network must experience the performance metrics of key IMS operations, in this paper all the measurements are connected with basic IMS signaling and media delivery parameters.

This paper is organized as follows. Section II is the introduction part and this is the longest section of this paper. First an overview of 3G UMTS and IMS is given in Section II-A. Then, Section II-B details the specifics of IPv6 UMTS access: eight possible access methods (native IPv6, L2TP, OpenVPN over UDP/TCP, 6to4, ISATAP, Teredo and NAT-PT) are described in separate subsections. Section III introduces the performance metrics, which are used for comparison in the measurements. Section IV shows the testbed where all the experiments have been done. There has not been any simulation, only real measurements with physical hardware have been applied. Section V describes the measured results. Finally, Section VI concludes the paper and shows some possible future work.

II. Background

In this section we first introduce the basics of IMS and 3G UMTS architectures, then we present the existing most well-known and most-widespread protocols to set up and maintain IPv6 connection for end users in all-IP 3G (and beyond) systems. Performance characteristic of IMS over IPv6-capable 3G networks will be analyzed using these methods as they provide IPv6-based connection for IMS applications in next generation mobile telecommunication systems.

A. Overview of 3G UMTS and IMS

The major innovation presented by the 3rd generation mobile networks during the pending evolution of mobile telecommunication architectures was the introduction of the Wideband Code Division Multiple Access (WCDMA) technology on the air interface and the all-IP paradigm of the core. As a result, significantly higher bandwidth became available compared to 2nd Generation (2G) Global System for Mobile telecommunications (GSM) and 2G+ Global Packet Radio Subsystem (GPRS) and Enhanced Data rates for GSM Evolution (EDGE) networks and also converged service provision became possible. The 3rd Generation Partnership Project (3GPP) 3G UMTS architecture can be divided into three main domains: Circuit Switched (CS), Packet Switched (PS) and Registration domain. In the next generation converged IP services, the most important one of the above listed items is the PS domain. The Packet Switched domain relies on the basics that were set in the GPRS principles but it uses the IP protocol in a more sophisticated way. In the core network the most important entities for the PS access are the RNC, SGSN and the GGSN [5]. The RNC (Radio Network Controller) manages the available radio resources by assigning appropriate radio bearer to user to maintain optimum performance. The SGSN (Serving GPRS Support Node) is responsible for routing and mobility management while also taking part in the authentication process. The GGSN (Gateway GPRS Support Node) provides the connections towards any exterior IPv4 and/or IPv6 network as seen in Fig. 1.

When a subscriber wants to access PS services it needs to request a PDP (Packet Data Protocol) context that enables the subscriber to access the service based on the information stored in the HSS (Home Subscriber Service). The PDP context defines the APN (Access Point Name) where the user belongs to, which determines the IP address and QoS properties for that PDP context. In case the connection is successfully set up the traffic between the SGSN and the GGSN is transmitted in GTP (GPRS Tunneling Protocol) tunnels. Theses tunnels are used to differentiate the user traffic belonging to a PDP context until it reaches the GGSN.

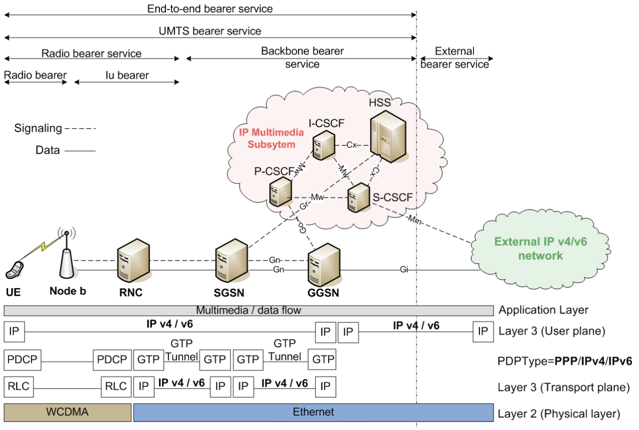

Figure 1. Overview of the all-IP 3G UMTS/IMS architecture

As the UMTS penetration has reached a critical level, research on advanced service provisioning standards was emerged to overcome the shortcomings of the already existing solutions. The core concept of IP Multimedia Subsystem (IMS) is to provide a comprehensive service provisioning framework for delivering IP multimedia to mobile users. As Fig. 1 shows, the IMS – one of the most important structural elements of all-IP systems in 3G and beyond – is an organic and integrated part of the 3G UMTS core network and also depends on the PS domain [6]. Initially designed for mobile networks by 3GPP, IMS has since evolved to also incorporate Next Generation Networks and the associated Fixed/Mobile Convergence vision: the Evolved Packet System (EPS). IMS enhances the basic IP connectivity of UMTS, provides flexible multimedia session management, media processing and control, and generally defines overlay architecture on the top of the packet switched core network incorporating the key converged, service and application oriented networks of the future [3]. All the above functions to control the multimedia sessions are implemented via different types of Call Session Control Functions (CSCF) using the Session Initiation Protocol (SIP) as a basis:

- Serving-CSCF: the session controller of the UE in the home network (like a GGSN).
- Proxy-CSCF: the local contact point of the UE in the visited network (like an SGSN).
- Interrogating-CSCF: the router of sessions in case of multiple S-CSCFs.

The flexibility of the above main architectural elements made IMS to be the common standard for next generation fixed (ETSI/TISPAN), cable (PacketCable) and mobile (3GPP, 3GPP2) networks, supporting the efficient delivery of multimedia data over any kind of access technologies (wired, wireless and mobile), and of course allowing operators and service providers to control the deployment, management and charging procedures of such convergent services.

B. IPv6 deployment and transition in 3G networks

The integration of the next generation Internet Protocol in today telecommunication architectures is a work in progress: IPv6 is still at its early stage of deployment. While rock-solid implementations are available for the most of the core network entities, solutions for mobile users to connect to IPv6 networks are sparse in the access (or the "last mile") segments of the networks. Besides 3G and beyond cellular networks these access architectures also include xDSL connection, cable connection and different heterogeneous access environments. Nowadays such systems usually provide the user with an IPv4-only connectivity, implying several drawbacks and incorporating different and often restrictive policies like limitation of possible mobility scenarios, restriction in the number of simultaneously active users, the application of private IPv4 addresses and NAT technologies, different firewall rules, etc. These disadvantages and drawbacks should be eliminated by serving the users with native IPv6 connection over existing access network technologies or by applying special solutions in order to provide IPv6 connectivity over IPv4. Only with doing this can IPv6 play its roles as the basis of new peer-to-peer services requiring advanced IP reachability and as the enabler of future innovation in converged mobile and wireless systems.

However, the transition from IPv4 to IPv6 and the regarding deployment questions are quite complex and affect many layers in the 3G UMTS/IMS architecture.

Considering the networking layer it is obvious that the IPv6 reachability is one of the fundamental needs in this context. The IPv6 network connectivity of a 3G user equipment (UE) can be provided either natively (i.e., by introducing native IPv6 in the user plane) or by applying one of the existing transition technologies on IPv4 (i.e., dual stack IPv4/IPv6 support in the affected network elements, tunneling, or IPv4-IPv6 protocol translators). In the first phase of the v4-v6 network transition several IPv6 islands will be interconnected by the IPv4 Internet using tunneling mechanisms. IPv4 only or dual stack UEs will use mainly IPv4 services and the rare IPv6 services provided to the users in this phase will be reached by tunneling (e.g., 6to4 [7] and ISATAP [8]) or protocol translation (e.g., Network Address Translation – Protocol Translation, NAT-PT [9] and Transport Relay Translator [10]). In the second phase we presume that IPv6 will be widely deployed over the Internet and numerous services will be based on the next generation IP protocol. However the deployment of IPv6 will be global in this phase, the IPv4 reachability will still be needed as the IPv6 Internet will not have full connectivity: several services will still exist only on IPv4 requiring dual stack implementations for efficient networking support. In the last phase IPv6 will achieve the dominant position. Due to global IPv6 connectivity all services will work on the IPv6 platform thus no dual stack functionality or other transition technique will be needed in the 3G and beyond architectures: native IPv6 will simplify the network architecture and will make possible to assign a unique, globally routable address to each and every user equipment in the network.

In the signaling layer two main aspects can be classified from the UE's point of view: IMS (i.e., SIP) signaling and Domain Name System (DNS) resolution. No issues of IMS signaling emerges in cases when the IMS callee (in the IMS context, callee refers to the called party) and caller are communicating over the same version of IP and the IMS itself supports the same version of Internet Protocol in the user plane. However, when an IPv6 user tries to call an IPv4 user (or reverse), translation in SIP and session negotiation (SDP) is to be applied by using application-aware translators as NAT-PT interworking with IMS Application Layer Gateways (ALGs) and special proxy servers acting as Back-To-Back User Agent (B2BUA). Regarding the DNS resolution the root of the problem lies on the fact that IMS procedures (e.g., registration, call-setup) strongly rely on the DNS database, as corresponding A/AAAA records provide the mappings of domain names of IMS entities to their IP addresses. In order to support this, a DNS ALG must be applied [11] or the DNS database has to be extended and modified with the appropriate AAAA entries and IPv6 mechanisms [12].

In the media layer questions similar to the signaling layer are to be answered as the main challenge here is also to handle v4/v6 heterogeneous situations (i.e., when an IPv6 user calls an IPv4 user or vice versa).

In this paper we focus on the state of the art user plane transition techniques (L2TP, OpenVPN UDP, OpenVPN TCP, 6to4, ISATAP, Teredo, NAT-PT) both able to efficiently deal with IPv6 provision over existing IPv4 3G UMTS architectures. The performance characteristics of IMS signaling and media transport will be compared over the above techniques using Native IPv6 and IPv4 scenarios as the basis of our comparison.

1) Native IPv4 / IPv6 3G UMTS

When a mobile user wishes to use packet switched (e.g., Internet) services in a 3G UMTS architecture, it must first attach to the network and then activate a PDP context. The UE receives its IP address during the activation of the PDP context and then it will be able to start the packet switched data communication. After the UE has been attached to the SGSN and it has been successfully authorized (i.e., the UE's identity has been checked and granted to access PS services), it must activate a PDP context (with appropriate IPv4 or IPv6 address) for commencing packet data communication. This is usually performed on application request (e.g., by starting a web-browser on the SmartPhone), but in some cases users may choose to be on-line for the whole time thus the packet data connection is established during or right after the boot sequence (e.g., registration into IMS). Users must specify on the UE the network service access point (i.e., the APN) of the Packed Data Network (PDN) they want to connect to and the PDP type (i.e., IPv4, IPv6, etc.) of the PDN they want to use. At the beginning of the PDP context activation, the UE puts the above parameters in an *Activate PDP Context Request* message and sends it to the SGSN. The SGSN uses the APN parameter to identify the corresponding GGSN for the requested PDN and makes it aware of the UE by the exchange of the *Create PDP Context Request* and *Create PDP Context Response* messages. As a result, a two way point-to-point tunnel is established between the SGSN and the GGSN: activating a PDP address sets up a GTP association between the UE's current SGSN and the GGSN that anchors the PDP address. A special data record is created regarding the associations maintained between the SGSN and GGSN. This record is called as PDP context and describes the main parameters of the connection (e.g., network type, and address type, APN, Quality of Service, billing information, etc.). After creating or updating the PDP context, the SGSN sends an *Activate PDP Context Accept* message to the UE in order to inform the mobile about the assigned PDP address and other context-related information. After finishing the PDP context activation procedure, the UE starts the appropriate v4/v6 address setup or allocation mechanism (e.g., DHCPv4/v6, IPv6 stateless autoconfiguration, etc.) depending on the requested PDP type and the received PDP address value. As a result, a native IPv6 or IPv4 connectivity will be produced where the GGSN plays the role of the default gateway for the UE. (More details on the IPv4/IPv6 address allocation mechanisms in 3G UMTS architectures can be found in [2], [13].)

2) Layer Two Tunneling Protocol

Layer Two Tunneling Protocol (L2TP) of RFC3931 [14] was designed to provide a dynamic and effective mechanism for tunneling Layer 2 "circuits" across datagram-oriented communication systems (like IP networks). L2TP was originally defined in RFC 2661 as a standard scheme for tunneling Point-to-Point Protocol (PPP) [15] sessions over IP. It was also designed to terminate these PPP sessions in a defined concentration point (i.e., L2TP Access Concentrator) of the network. Since the release of the first version of the protocol, L2TP has been adopted for tunneling a number of other layer two protocols like Ethernet, Frame Relay and Asynchronous Transfer Mode (ATM). L2TP merges the functionality of two former proprietary tunneling methods for PPP, which are Cisco's L2F (Layer 2 Forwarding) and Microsoft's PPTP (Point to Point Tunneling Protocol) and operates in the Session Layer of the OSI reference model. The latest version of the protocol also incorporates advanced security features (L2TP/IPSec VPN protection), improved encapsulation and the possibility to carry extended circuit status attributes (to communicate finer-grained error states).

L2TP operates in two sublayers namely the control sublayer and the data sublayer. The control sublayer provides the reliability through packet numbering and acknowledgment system, while the data sublayer ensures the data transmission and detects any message loss using a sequence number. During its operation, L2TP simulates a specific data link layer and inserts every single data packet into a PPP frame before adding the L2TP encapsulation. Then the entire L2TP packet (including the payload and the L2TP header) is sent in a simple IP or in a UDP datagram. When L2TP operates directly over IP, L2TP packets cannot take advantage of the UDP checksum for checking packet integrity, which is important especially in case of L2TP control messages. Therefore L2TP usually applies UDP, in which messages will be transmitted using any IP network based on any data link connection between the two endpoints of the L2TP tunnel. These two endpoints are called the LAC (L2TP Access Concentrator) and the LNS (L2TP Network Server). The LAC plays the role of the initiator of the tunnel establishment while the LNS is the server continuously waiting for new tunnel requests. Every established L2TP tunnel between the peers is bidirectional and transmits higher-level protocols. In order to support this, an L2TP session (i.e., call) is established within the tunnel for each higher-level protocol such as PPP. Sessions inside an existing tunnel can be initiated either by the LAC or the LNS, and the traffic of each session is isolated by the L2TP. Note that this feature makes it possible to set up multiple virtual networks across a single tunnel.

L2TP is often used as a tunneling mechanism in xDSL and Cable architectures as a solution for selling/reselling endpoint connectivity: an L2TP tunnel sits between the user and the ISP the connection is to be sold/resold to, so the selling/reselling ISP will not appear as dealing with the transport functionalities.

In 3G UMTS architectures L2TP could be an effective way to provide IPv6 access over existing IPv4 technologies: the IPv4 PDP type traffic containing encapsulated IPv6 packets from the UE is processed at the GGSN, where the IPv4 sessions are terminated, then the GGSN transports this traffic over L2TP and then routes over Gi to their IPv6 destination.

3) Virtual Private Networks

Virtual Private Networking (VPN) is another method to provide IPv6 connection on an existing IPv4 only 3G UMTS architecture. In general, a Virtual Private Network is a special computer network that is implemented as supplemental software layer (i.e., overlay) on the top of an existing network infrastructure aiming to create an exclusive interconnection of communicating nodes or to provide a secure access to a private network by extending it into an insecure or shared/public architecture (like the Internet). Such overlay structures can be built by using different tunneling methods and by encrypting, decrypting and authenticating traffic inside the tunnels. OpenVPN is a well known and widespread VPN implementation also based on tunneling [16]. OpenVPN creates the secure tunnels using SSL (Secure Sockets Layer), which is a commonly-used protocol for securing Internet transactions in the application layer (HTTPS protocol also uses SSL for securing Internet transactions on the web). This protocol is one of the industrial standards for establishing VPNs, robust, quite easy to implement/manage by administrators and learn/understand by users.

The implementation of OpenVPN is based on the OpenSSL library, which realizes encryption, authentication and certification features for the secure tunnel and manages the SSL connection over TLS (Transport Layer Security) protocol to transmit data [17]. OpenVPN tunnels can be established between a client and a server and can run both over UDP and TCP. During the operation IP packets that need to be sent in the tunnel are encrypted and encapsulated in a UDP or TCP message. Then this packet can be transmitted using any IP network based on any layer 2 connection. The fact that OpenVPN is implemented as a user-space daemon rather than a kernel module or a complex extension to the IP layer makes the method portable, easily deployable and configurable.

As OpenVPN is a cost-effective and lightweight alternative to other VPN technologies, it is commonly applied at Small and Medium Enterprises (SMEs) and also well targeted in the enterprise markets.

In order to provide IPv6 connectivity for UEs in an IPv4 3G UMTS network, a secure OpenVPN tunnel can be used, which encapsulates the IPv6 traffic and relays it to the UE through the IPv4 networking segment. In such a scenario the dual stack UE operates in IPv4 mode (it opens an IPv4 type PDP context and receives an IPv4 address from the GGSN), but also an OpenVPN tunnel is spanned over this IPv4-only connection between the UE and the tunnel server. This tunnel is the gate to a VPN, which basically extends the IPv6 connectivity into the IPv4 3G UMTS network.

4) 6to4

In RFC3056 [7] authors specify a scheme for IPv6 sites to communicate with each other over an existing IPv4 network without explicitly given tunnel endpoint information. 6to4 does not use IPv4-compatible IPv6 addresses (where the prefix ::/96 is separated for IPv4-compatible addresses, and the rightmost 32 bits of the IPv6 address stand for the IPv4 address of the destination) but it has a proper IPv6 address format that includes the IPv4 address of the tunnel endpoint in the prefix such allowing automatic tunnel setup. In 6to4 the transport IPv4 network behaves as a unicast point-to-point link, and the 6to4 domain segments communicate via 6to4 routers (i.e., 6to4 gateways): IPv6 packets are encapsulated and decapsulated here requiring at least one globally unique IPv4 unicast address. Only the gateways need to be 6to4 compatible, therefore no other changes have to be made to the IPv6 nodes inside the 6to4 network. The prefix for the 6to4 protocol assigned by the IANA organization is `2002::/16` providing 6to4 addresses in the `2002:IPv4Adrr::/48` structure. It is important to notice, that if a host in a 6to4 network wants to exchange packets with a host in another 6to4 network, no tunnel configuration is needed: the tunnel entry point can take the IPv4 address of the tunnel exit point from the IPv6 address of the destination. Besides the above, a 6to4 relay router is needed for a successful communication with an IPv6 node in a remote IPv6 network. The relay router is a router configured for 6to4 operation and also IPv6 connection. The relay router connects 6to4 networks to the native IPv6 network as the `2002::/16` prefix is announced into the native IPv6 network by such relays.

As an extension to the basic standard, RFC3068 [18] specifies a 6to4 relay router anycast address in order to optimize the configuration of 6to4 gateways, which require a default route towards a 6to4 relay router on the IPv6 Internet.

The application of the 6to4 technique in mobile telecommunication architectures is twofold. On one hand sites offering IPv6 mobile access can be connected with each other and with the IPv6 world through IPv4 using 6to4. Here the operation of the transition technique is transparent to the IPv6 mobile UEs: they only have to be configured with at least one 6to4 IPv6 address in the `2002:IPv4Adrr:SubnetID::/64` format. On the other hand 6to4 tunnels can be spanned right between a 6to4-compatible dual-stack UE and the 6to4 relay router over the IPv4-only 3G UMTS network, such providing encapsulation-based IPv6 support while still using IPv4 PDP contexts. In this case the 6to4 relay router resides inside the operator network on the v4/v6 domain boundary.

5) ISATAP

The Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) is specified in RFC4214 [8] aiming to provide IPv6 connectivity for dual-stack hosts over an IPv4-based networking infrastructure. This technique also uses the existing IPv4 network as one large link-layer architecture and allows the dual-stack hosts to automatically create tunnels and exchange data between themselves. ISATAP can be used regardless of whether the hosts have global or private IPv4 addresses. Addresses of this automatic tunneling mechanism embed an IPv4 address in the EUI-64 interface identifier in the following format:

```
64bitPrefix:16bitControl:5EFE:IPv4address.
```

ISATAP interfaces form ISATAP interface identifiers using their IPv4 addresses and apply them to produce the ISATAP link-local addresses in order to make the technique able to perform standard IPv6 neighbor discovery mechanisms. Using this method, IPv6 nodes inside an IPv4 intranet can communicate with each other. If hosts want to communicate with IPv6 hosts outside the intranet (e.g., 6Bone hosts), a border router must be configured, which can be an ISATAP router or even a 6to4 gateway. An important issue of this method is that all hosts in an ISATAP network need to support the ISATAP protocol.

ISATAP (together with 6to4) are considered as two really promising and already popular transition technologies evaluated and assessed within several real-life testbed experiments and projects like [19], [20] and [21]. In IPv4-only 3G UMTS architectures ISATAP can be used as an automatic tunneling solution for dual-stack UEs that are multiple IPv4 hops away from the IPv6 network. Mobile terminals can build tunnels between each other and exchange IPv6 traffic using their link-local addresses: the packets are transmitted via ISATAP tunnels with endpoints that are derived from the interface ID segment of the link-local addresses. For outside (or offlink) IPv6 traffic UEs have a default route, pointing to the ISATAP address of the ISATAP router.

6) Teredo

Teredo is specified in RFC4380 [22] as an IPv6 transition technology providing address assignment and automatic host-to-host tunneling for unicast IPv6 traffic in cases when IPv6/IPv4 nodes are placed behind IPv4 network address translators (NATs). Comparing Teredo with 6to4 and ISATAP we can summary that 6to4 makes IPv6 available over an IPv4 network using public IPv4 addresses, ISATAP helps deployment of IPv6 nodes within a site regardless of whether it applies private or public IPv4 addresses, and Teredo makes IPv6 available to nodes through any number of NAT layers using UDP-based tunneling. The Teredo architecture consists of a Teredo server (a well-known host, which is used for initial configuration of a Teredo tunnel helping clients to access IPv6 networks), several Teredo clients (running on an IPv4/IPv6 dual-stack terminal in an IPv4 network behind a NAT) and Teredo relays (the remote end of a Teredo tunnel forwarding IPv6 traffic between a Teredo client and a host in the IPv6 network). The technique introduces a special prefix called Teredo Service Prefix (`2001:0000::/32`), which is announced by the Teredo relays to the outside world using conventional IPv6 routing mechanisms. Based on this prefix each Teredo client assigns a public IPv6 address that is constructed as follows:

```
2001:0000:ServerIPv4:Flags:UDPport:ClientIPv4.
```

A significant part of RFC4380 deals with how Teredo identifies the specific type of NAT deployed in the actual network and defines mechanisms for handling these various NAT types.

During the protocol's basic communication procedure first the Teredo client inside the IPv4-only domain starts the determination of the Teredo relay serving the IPv6-only host by sending out an *IPv6 Echo Request* message via the Teredo server. This request is forwarded to the IPv6-only host, which answers it with an *IPv6 Echo Reply* message destined to the Teredo client's address and routed to the to the nearest Teredo relay. The Teredo relay tunnels the reply message to the client that now determines the relay IPv4 address and starts sending packets to the IPv6-only host via the relay. The Teredo relay decapsulates the IPv6 packet and forwards it to the IPv6-only Host.

Based on the above operation Teredo solves numerous problems of IPv4-IPv6 transition. However, the current version of the standard does not work with symmetric NATs. In order to support Teredo for symmetric NAT traversal, authors of [23] proposed SymTeredo, which imposes minor modifications on the Teredo relay and the Teredo client components but also keeps compatibility with the standard protocol.

3G operators can rely on Teredo's efficient and NAT friendly IPv4-IPv6 transition toolset by introducing the components of the Teredo architecture in the UMTS network. However, as Teredo can only provide a single IPv6 address per tunnel endpoint, it is not possible to use a single Teredo tunnel to make connection with multiple nodes (contrary to 6to4), such creating significant tunneling overhead on the air interface in several common scenarios. The application of Teredo –similarly to the majority of the above schemes– still not transparent: it requires additional UE configuration and installation of supplementary software modules (i.e., Teredo implementation) on the UE. Nevertheless, the big number of Teredo implementations that are already available for the widest scale of operating systems (Linux, *BSD, Mac OS X, Windows XP SP2/Server 2003/Vista and Windows 7) may assume that popular UE platforms will introduce Teredo functionality.

7) NAT-PT

RFC2766 [24] introduces the Network Address Translation - Protocol Translation (NAT-PT) transition scheme, which uses a pool of public IPv4 addresses for dynamic assignment to IPv6 hosts, and employs a stateful IPv4/IPv6 header translation on a special network device located at the boundary of the IPv4 and the IPv6 networks. This NAT device translates IPv6 packets into analogous IPv4 packets and vice versa, and such routes between an IPv6 network and an IPv4 network. NAT-PT reserves the pool of IPv4 addresses and translates the fields for IP Source addresses, IP, TCP, UDP, and ICMP header checksums. Note that in order to achieve this behavior, NAT-based v4/v6 transition schemes usually apply IPv4/IPv6 header translation rules specified in RFC2765 (Stateless IP/ICMP Translation) [25].

An extension of NAT-PT is Network Address Port Translation - Protocol Translation (NAPT-PT), which further extends the original idea: in order to allow numerous IPv6 hosts to share one single IPv4 address for multiplexing multiple sessions on one address, transport identifiers (such as TCP and UDP port numbers) are also translated in this technique.

The main benefit of NAT-PT and NAPT-PT is that no changes are required to end hosts because all the translation procedures are executed at the separate NAT device in the network. However the mechanisms defined in RFC2766 seem to be convenient in several transition scenarios, serious issues exist with the standard. For example, NAT-based schemes cannot take full advantage of the enhancements offered by IPv6, and it is really hard to maintain the big number of Application Level Gateways (ALG) needed in NAT devices to keep the widest scale of applications working correctly through the gateway. The raised problems are summarized in RFC4966 [26] together with the conclusion that technical and operational difficulties resulting from these issues make it undesirable to recommend the usage of RFC2766 as a general purpose transition mechanism. However, the transparent nature of NAT-PT/NAPT-PT (i.e., the fact that clients don't need to be modified for benefitting from the method's IPv4-IPv6 transition services) makes suitable the technique for application in mobile telecommunication systems.

In 3G UMTS networks NAT-PT or NAPT-PT can be deployed by installing a NAT device and the appropriate ALGs at the boundary of the IPv4/IPv6 network segments. Configuration and modification on UEs is not required, only the suitable DNS server settings must be provided for the terminals.

III. Performance Metrics

The main motivation of our work was to compare native IPv4 3G UMTS network performance with different IPv4-IPv6 transition methods (including the native IPv6 communication itself), using essential parameters of IMS operations as performance metrics. These measured parameters, which substantially affect the network performance in IMS based multimedia-centric user scenarios are the following: the round-trip time, the IMS registration time, the call setup time, and the downlink RTP delay.

A. Round-trip Time

The round-trip time (RTT) is the time elapsed while a transmitted packet arrives back from the recipient, if the packet is forwarded back immediately. This parameter is useful to examine the minimum response delay between two communicating nodes.

We used the `ping` application with 64byte packets to measure the round trip delay between the UE (sender) in the 3G network and the CN (recipient) in the outside PDN. The results of RTT measurements are corresponding as the main performance metrics of the examined architectures in the four scenarios.

B. IMS Registration Time

Registration is one of the most important procedures in next generation IP multimedia systems since this mechanism makes possible to initiate sessions between users in the network and to receive data from media and application servers.

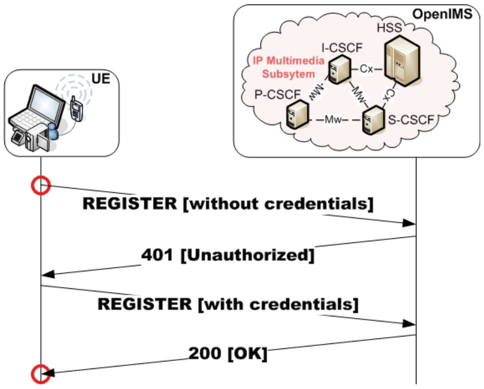

Figure 2. IMS Register Flow

To measure the time required to register a user inside the IMS in a 3G UMTS architecture we used `SIPp`, which is a traffic generator tool for the Session Initiation Protocol (SIP) [27]. The simplified schematics of the message exchange of an IMS registration procedure is shown in Fig. 2. The registration starts with a REGISTER message sent by the UE. A 401 UNAUTHORIZED message reaches the UE after the IMS processed the initial REGISTER in order to challenge the UE to send the required authentication information. After that an extended REGISTER message is transmitted on the same path as the first one. This message now contains all the required data to authenticate the UE. The IMS indicates the successful registration with a 200 OK message. (Further details on the IMS registration procedure can be found in [28].)

Appropriate `SIPp` scripts were executed on the UE in order to manage the REGISTER procedure and to control the flow of synthetically produced SIP packets between the UE and the IMS system. IPv4 or IPv6 addresses of IMS entities (e.g., P-CSCF) were provided by the DNS server.

In this context we considered the registration time as the elapsed time between sending the first REGISTER message and receiving the 200 OK message in the UE side (see the red markings in Fig. 2).

C. Call Setup Time

Right after a successful registration, IMS subscribers of a 3G UMTS system can initiate IMS calls to other subscribers or media providers. An outline of the IMS call setup flow is depicted in Fig. 3. (The detailed flowchart can be found in [28].)

The caller UE starts the call setup procedure by sending an INVITE message to the P-CSCF with the CN's user name and the SDP descriptors in it. This message is forwarded to the CN by several IMS mechanisms leading the CN to reply by sending a 183 SESSION IN PROGRESS message containing SDP descriptors. Also some informal messages are exchanged (100 TRYING, 180 RINGING) during the procedure, and finally a 200 OK arrives back to UE, which means that the callee (i.e., the CN) accepted the call. This fact is acknowledged by an ACK message, which is sent by the UE to the CN (and the S-CSCF) through the P-CSCF. When the CN receives the ACK message, the call setup is finished and the Real-time Protocol (RTP) [29] datagram exchange starts between the communicating peers. This metric can also be measured using `SIPp` on UE and CN entities in order to generate and manage IMS signaling in the context.

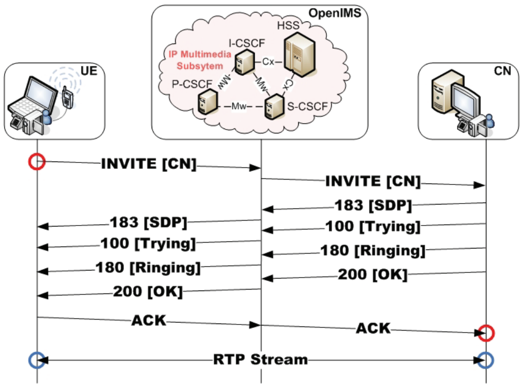

Figure 3. IMS Call Setup and RTP delivery

As because the call setup time is the elapsed time between the first INVITE message (sent by the UE) and the ACK message (arrived at the CN) (see red markings in Fig. 3), there is a strong need to synchronize the clocks of the two nodes to get precise results. This time synchronization can be achieved by NTP (Network Time Protocol). In order to avoid NTP inaccuracy and undesirable drifting, we introduced a dedicated "shadow" network for the NTP signaling between a local NTP Server and the UE/CN nodes. Based on this scheme we achieved an approximated accuracy of ±30µs, which offers sufficient error margin for the measurements presented in the article.

D. One-way RTP Delay

When the call setup is finished, RTP packets are starting to be exchanged between the two communicating peers. Because of the nature of services, the RTP data flow is often unidirectional, usually in downlink direction (e.g., in case of a video or audio streaming). Therefore we measured the downlink, one-way RTP delay as the most significant performance metric of the media plane.

In our scenarios the CN played the role of the media server and the UE was the subscriber to an audio streaming service, which provided a 192kbps Constant Bit Rate (CBR) audio source.

As in the case of the previous metric, here also a time interval between events occurring on two different nodes (see the blue markings in Fig. 3) have to be examined, so the dedicated NTP network must be introduced here too for accurate measurements. RTP packets can be captured by packet analyzers (e.g., `tshark` [30]), and the time stamps of the sent and received packets can be used for calculating the RTP delay.

IV. Measurement Architecture and Scenarios

In this section we introduce our testbed and the scenarios used to compare the main performance metrics of IMS operations over different IPv6 provision techniques in 3G UMTS networks. In the first subsection our native IPv4/IPv6 3G UMTS network is described in details, followed by the eight measurement scenarios: native IPv4, native IPv6, L2TP IPv6, OpenVPN IPv6, 6to4, ISATAP, Teredo, and NAT-PT respectively.

A. Overview of the Testbed

In order to provide a testbed for advanced IPv6 mobility and multihoming researches and analyzing IPv6 deployment and v4-v6 cohabitation/transition issues in next generation multimedia-centric communication systems, we designed and implemented a native IPv6 UMTS/IMS architecture based on the existing hardware elements of Mobile Innovation Centre (MIK) located in Budapest, Hungary [31]. However almost all the relating hardware and software components were presented in MIK, one important item was missing: the laboratory did not possess any dedicated Gateway GPRS Support Node (GGSN) device for supporting native IPv6 UMTS access. Thus one of the main tasks during the implementation of our UMTS/IMS testbed was to design and develop a GGSN, prepared to be integratable with the other UMTS elements and adequate to handle also IPv6 type PDP (Packet Data Protocol) contexts besides IPv4. In order to achieve this, we used a software GGSN implementation called OpenGGSN [32] as a basis of our work. Our GPL licensed and publicly available OpenGGSN modification (OpenGGSN 0.84_v6_05 [33]) uses the same GTP library and the main architecture as version 0.84, but extends the original edition with the missing IPv6 routines and some other related components for setting up, maintain and tear down contexts of native IPv6 UMTS communication.

The integration of our IPv4/IPv6-compatible (i.e., dual-stack) GGSN software into the UMTS/IMS testbed architecture for providing also native IPv6 packet exchange was a six-step procedure. First, we had to create a new, IPv6-compatible APN in the SGSN, than we had to enable also IPv6 PDP contexts for the SIM cards of our devices in the Home Subscriber Server (HUAWEI HSS 9820). After that we compiled, configured and started all the required OpenGGSN 0.84_v6_05 components on a SunFire X4200 (powered by AMD Opteron^{TM} processors, 4GB RAM, and running Ubuntu 7.04 Feisty Fawn operating system with kernel 2.6.23). As the 4th step we deployed an open-source software IMS implementation called Fraunhofer OpenIMS [34], which realizes all the functional entities (HSS and all CSCFs) of IMS architecture and supports both IPv4 and IPv6. We used version 604 of OpenIMS with a Debian 5 (Lenny) operating system and kernel 2.6.26 on a SunFire X4150 server comprising 2.83GHz Intel^{TM} Dual Quad-Core Xeon E5440 processors and 8GB RAM. Step No. 5 was the configuration of end terminals, while the last step was setting up the appropriate IPv6 routing entries in the routers of the testbed in order to provide outside IPv6 PDN (i.e., GEANT) connection to the mobiles. Fig. 5 shows all the details of the native IPv6 UMTS/IMS architecture we used for our native IPv6 experiences, while Fig. 4 presents the details of the native IPv4 3G UMTS testbed. Note, that these two figures represent one, integrated, dual-stack tested system basically under the same architecture (with the same OpenGGSN 0.84_v6_05): using our OpenGGSN modification both IPv4 and IPv6 PDP contexts can be handled such creating a highly configurable all-IP 3G testing environment making possible to observe, measure and even modify every kind of IP-level function, traffic or operation.

The core UMTS infrastructure in our laboratory consists of one Node B and one RNC linked to the SGSN, which is connected to the GGSN and the HSS using standard interfaces. As Fig. 4 and 5 show, the SGSN and the GGSN are still communicating over IPv4 (i.e., the GTP tunnels are set up on IPv4), but this fact has no effect on the UE's context: either native IPv6 or native IPv4 UMTS connection can be provided, the mode of communication between the GSN nodes (i.e., the transport plane) does not have any impact on the type of user plane communication. The GGSN is connected to the outside (v4 or v6) network through its Gi interface.

For accessing this UMTS/IMS architecture, a dual-stack UE has been constructed from conventional hardware building blocks and equipped with the appropriate software components. UE's hardware is based on an ASUS V6800VA notebook with a Nokia N95 8Gb SmartPhone as an IPv6-compatible, dual-stack 3G modem for UMTS connectivity. The UE's operating system is a Ubuntu 8.04 LTS equipped with IPv6-capable Point-to-Point Protocol daemon (pppd v2.4.4) and SIPp v3.1 for managing the synthetic IMS signaling and media traffic. The CN is a Fujitsu Siemens Scienic SE PC with 3GHz Intel^{TM} Pentium 4 processor, 2GB RAM, double Ethernet LAN adapter and the same software components as on the UE.

B. Measurement scenarios

In the previous subsection we presented the general structure of our dual-stack 3G UMTS/IMS architecture. In order to implement different measurement scenarios we applied several modifications and added some new entities for dealing with scenario-specific functions. These modifications and architectural changes are described in the following paragraphs.

1) Native IPv4

The testbed setup for the native IPv4 scenario is shown in Fig. 4. The UE uses the Nokia N95 8Gb smart phone as 3G wireless interface and connects through the 3G PS/IMS domain to the wired Correspondent Node (CN), which will be the communication partner of the UE during the measurements. An important node is not presented by Fig. 4 although it has a significant task not only here but also in the further scenarios: the Network Time Protocol (NTP) Server providing time synchronization for nodes under measurement is connected to the UE and the CN by a wired "shadow" network. The NTP server itself is a desktop PC running Ubuntu 8.04 LTS with NTP v4.2.4p4.

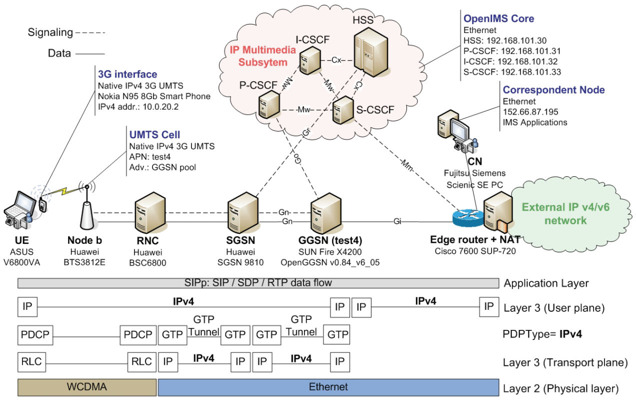

Figure 4. Native IPv4 3G UMTS/IMS testbed architecture

As introduced in the previous sections, the Packet Data Protocol (PDP) context offers a packet data connection over, which the UE and the network can exchange IP packets. In this scenario an IPv4 PDP context is used to build up native IPv4 user plane communication sessions between the UE and the PDN (i.e., the IPv4 Internet). The GGSN provides `10.0.20.2` address from its pool to the UE for IPv4 PS communication. Due to this and the limited number of available IPv4 addresses we also turn on NAT functions in our edge router for assuring outside communication of User Equipments.

The OpenIMS and the related DNS entries for the HSS and the CSCF sub-entities were configured to be reachable with IPv4 addresses. The used APN was *test4*, which identifies the IPv4 PDN in our testbed and the OpenGGSN software is responsible to implement its functions.

2) Native IPv6

The native IPv6 3G UMTS network is basically the same as the IPv4 version. The main difference is the usage of IPv6 PDP contexts for the UE in order to establish and maintain native IPv6 user plane communication (Fig. 5). It can be achieved by specifying IPv6 for the type of PDP context to be created. The UE's IPv6 compatible 3G modem interface can easily be instructed to do this using an appropriate AT command (that is `AT+CGDCONT=1,"IPV6","test6",,0,0` in our testbed setup). As it can be seen, the requisited APN was also modified from *test4* to *test6* (belonging to the IPv6 PDN). Thanks to this, the UE is aware of that an IPv6 PDP context is to be created and will send and *Activate PDP Context Request* message with `PDP type=IPv6` towards the SGSN. The SGSN sends a *Create PDP Context Request* message to the GGSN, which answers it with a *Create PDP Context Reply* containing an IPv6 address in the `PDP address` field of the message. This address will be passed to the UE in an *Activate PDP Context Reply* by the SGSN. The UE extracts the interface identifier part from the received IPv6 address, creates its IPv6 link-local address (`fe80::1234:1234:1234:1234`) and sends an IPv6 *Router Solicitation* message to the GGSN. The GGSN replies with a *Router Advertisement* containing an appropriate IPv6 networking prefix (`2001:738:2001:20a9::/64`). Using this advertisement and the previously get link-local identifier, the UE is able to generate its global IPv6 unicast address for the IPv6 communication. Note, that our native IPv6 3G UMTS testbed only supports the above mechanism (i.e., the IPv6 stateless address autoconfiguration) and no DHCPv6 is supported at the moment.

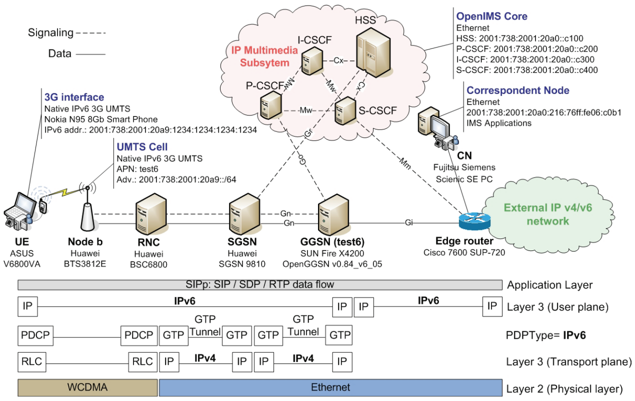

Figure 5. Native IPv6 3G UMTS/IMS testbed architecture

All the procedures shortly introduced above are taking part from the standard operations of a native IPv6 UMTS system, thus the implementation of these functions was mandatory for our OpenGGSN 0.84_v6_05 [33] implementation. However, we took advantages of some simplification possibilities during the design of our dual-stack GGSN software in order to reduce the development time and the requested human resources. These simplifications are mainly connected to the address allocation procedures and the QoS-related functions. More details on our OpenGGSN development and on IPv6 PDP context management in 3G and beyond architectures can be found in [33] and [13], respectively.

After the successful IPv6 context activation and address configuration, the UE is able to natively communicate with the IPv6 IMS domain, with other network entities or nodes in the IPv6 Internet (e.g., the IPv6 CN). Thanks to the tremendous number of available addresses and the nature of IPv6 in general, there is no need to apply NAT for outside communication in this scenario.

The OpenIMS and the DNS entries for the HSS and CSCFs must be configured to use IPv6 addresses. It is not shown but the NTP server still provides time synchronization service over the dedicated "shadow" network for UE and CN nodes.

3) *L2TP IPv6*

The Layer-2 Tunneling Protocol (L2TP) [14] scenario is built upon the native IPv4 scenario (Fig. 6). After initializing the native IPv4 3G UMTS user plane communication, the UE – configured as an L2TP Access Concentrator (LAC) in the `l2tp.conf` – searches for an L2TP Network Server (LNS) and sets up an unsecured L2TP tunnel over IPv4 in order to transport IPv6 packets on it. It means that on our Linux-based UE a novel Point-to-Point interface (`ppp1`) will be created besides the PPP interface used by the 3G UMTS connection (i.e., `ppp0`). The Router Advertisement Daemon (radvd-1.6) [35] running on the LNS will send periodic Router Advertisements through the PPP tunnel towards the UE, which will be able to configure its global address (`2001:738:2001:20a9:2c4b:931f:144e:1478/64`) with stateless autoconfiguration. The LNS in this scenario is the SunFire X4200 server, which also acts as the *test4* GGSN for the IPv4 PDN and runs Roaring Penguin v0.4 user-space implementation of L2TP such as the UE [36].

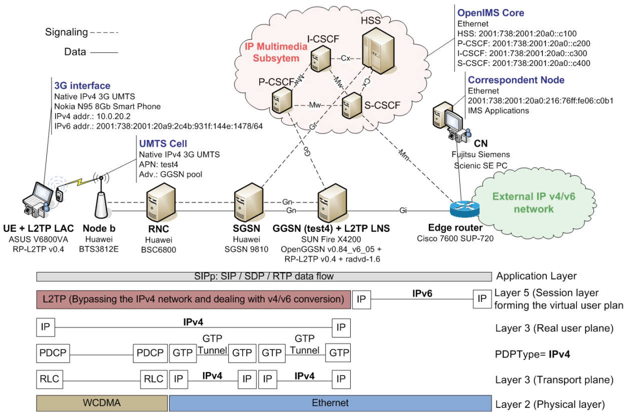

Figure 6. L2TP IPv6 3G UMTS/IMS testbed architecture

As shown in Fig. 6 the L2TP tunnel spanned in the session layer forms a virtual user plane where L2 data frames (Ethernet in our case) are accepted and forwarded. The L2TP tunnel uses UDP datagram to send the L2TP header and the payload to the two endpoints (LAC, LNS). The IPv6 packets are encased into this type of UDP packets and sent through the tunnel as IPv4 packets. Accordingly, the CN, the IMS and the DNS need to be reachable on IPv6 for the measurements.

The "shadow" network for NTP is used again in this scenario in order to synchronize the UE/CN nodes.

4) *OpenVPN IPv6*

This scenario uses OpenVPN [16] to create an encrypted point-to-point tunnel between the UE and the gateway towards the IPv6 PDN and supports IPv6 communication over a built IPv4 3G UMTS user plane based on both TCP and UDP transport protocols. The scenario topology is almost the same as the previous one, but here OpenVPN (v2.1_rc11 both on the UE and the GGSN) is used to create an application level IPv6 on IPv4 tunnel (Fig. 7).

After setting up our own Certificate Authority (CA) and generating certificates and keys for the OpenVPN server running on the same host as the GGSN and for the OpenVPN client of the UE, we created both the server and client configuration files (`openvpn.conf`). Here we specified the transport protocol (UDP or TCP) and the device (`tun0`) to be used, and edited the *ca*, *cert*, and *key* parameters. The upcoming step of constructing this measurement scenario was the startup of the VPN over the built IPv4 3G UMTS connection by running `openvpn` both on the UE and GGSN nodes. The assembled VPN connectivity makes possible to send `radvd` *Router Advertisements* from the GGSN to the UE over the `tun0` interface. Eventually this enables the UE to configure its IPv6 address for global communication (`2001:738:2001:20a9:d42d:d4ff:fe28:cb6b/64`).

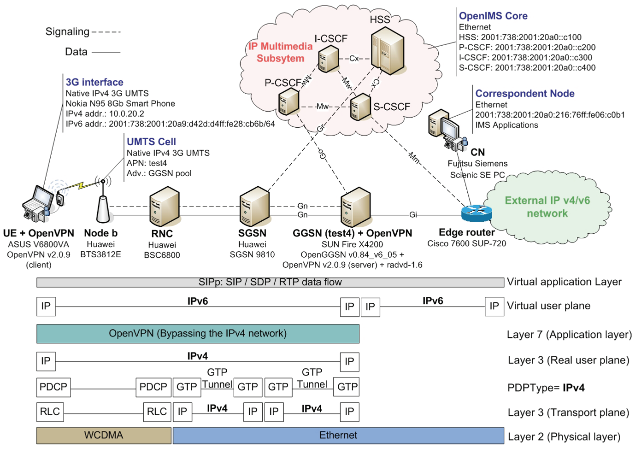

Figure 7. OpenVPN IPv6 3G UMTS/IMS testbed architecture

We measured the performance of the key IMS operations over both TCP and UDP based OpenVPN tunnels. The measurements were supported by NTP using the same dedicated time synchronizer network as in the above scenarios. Since it is also an IPv6 scenario, the CN, the DNS and the IMS needs to comprise IPv6 reachability for the measurements.

5) *6to4*

In this v4-v6 transition scenario the dual-stack UE was also a 6to4 router: it was configured to support the use of a 6to4 tunnel interface and to forward 6to4-addressed traffic between itself and a 6to4 relay over the IPv4 3G UMTS connection. Since 6to4 routers require additional configuration and processing logic for encapsulation and decapsulation, the operation of such 6to4 compatible UE cannot be transparent. We used the Linux kernel implementation of the 6to4 protocol and our setup was based on the descriptions and guidance of [37].

The UE's 6to4 prefix was `2002:0A00:1402::/48` derived from the `2002::/16` IPv6 prefix and the IPv4 address `10.0.20.2` acquired during its pure IPv4-type PDP context activation. We assigned the suffix `::1` to this entity, such creating the IPv6 address of the UE, which equals with the IPv6 address of the 6to4 tunnel spanned between the UE's and the GGSN's IPv4 address.

As the GGSN is the IPv6/IPv4 entity that must forward 6to4-addressed traffic between 6to4 routers (i.e., UEs) inside the 3G UMTS network and IPv6 hosts on the IPv6 Internet, it also applies 6to4 relay functions.

The created 6to4 tunnel maintained by the 6to4 router and relay (i.e., the UE and the GGSN respectively) provides the virtual user plane making able the UE to perform IPv6 communication with the IMS core and the CN (Fig. 8).

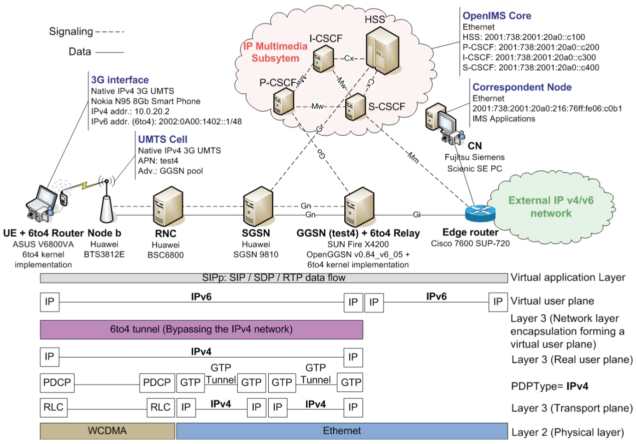
Signaling
Data
IP Multimedia Subsytem
HSS
I-CSCF
P-CSCF
S-CSCF
Cx
Mw
Gi
Go
Mm
OpenIMS Core
Ethernet
HSS: 2001:738:2001:20a0::c100
P-CSCF: 2001:738:2001:20a0::c200
I-CSCF: 2001:738:2001:20a0::c300
S-CSCF: 2001:738:2001:20a0::c400
Correspondent Node
Ethernet
2001:738:2001:20a0:216:76ff:fe06:c0b1
IMS Applications
CN
Fujitsu Siemens
Scienic SE PC
3G interface
Native IPv4 3G UMTS
Nokia N95 8Gb Smart Phone
IPv4 addr.: 10.0.20.2
IPv6 addr. (6to4): 2002:0A00:1402::1/48
UMTS Cell
Native IPv4 3G UMTS
APN: test4
Adv.: GGSN pool
External IP v4/v6 network
Gn
Gi
UE + 6to4 Router
ASUS V6800VA
6to4 kernel implementation
Node b
Huawei
BTS3812E
RNC
Huawei
BSC6800
SGSN
Huawei
SGSN 9810
GGSN (test4) + 6to4 Relay
SUN Fire X4200
OpenGGSN v0.84_v6_05 +
6to4 kernel implementation
Edge router
Cisco 7600 SUP-720
SIPp: SIP / SDP / RTP data flow
Virtual application Layer
IP
IPv6
Virtual user plane
6to4 tunnel (Bypassing the IPv4 network)
Layer 3 (Network layer encapsulation forming a virtual user plane)
IPv4
Layer 3 (Real user plane)
PDCP
GTP
GTP Tunnel
PDPType= IPv4
RLC
Layer 3 (Transport plane)
WCDMA
Ethernet
Layer 2 (Physical layer)

Figure 8. Application of 6to4 in our 3G UMTS/IMS testbed

The dedicated “shadow” NTP network for UE and CN time synchronization was implicitly applied also in this measurement scenario.

6) ISATAP

The ISATAP-based v4-v6 transition scheme was built upon the Linux in-kernel ISATAP support firstly introduced in kernel version 2.6.25. In order to make this implementation work, the GPLv2 licensed isatapd-0.9.6 [38] was installed on the client side (UE) and a static ISATAP tunnel device with `radvd` [35] support was configured on the ISATAP router.

The `isatapd` module on UE creates and maintains ISATAP tunnels by taking care of the following tasks:

- Constructing ISATAP tunnel device(s) based on IPv4 interface(s)
- Periodically querying and adding router addresses to the potential ISATAP router list
- Periodically sending router solicitation messages to potential ISATAP routers to get on-demand router advertisements for maintaining IPv6 connectivity
- Receiving and parsing incoming router advertisements in order to adjust the router solicitation interval
- Detecting link changes and maintaining ISATAP tunnel(s)

The configuration of the ISATAP router was performed on the GGSN with Linux command line tools: we had to statically set up an ISATAP tunnel device, then configure an ISATAP compatible address for it and start `radvd` on it for on demand advertisements.

Applying the above steps in our testbed the dual-stack, ISATAP compatible UE with only IPv4 PDP context in the 3G UMTS network was able to construct its ISATAP address (`2001:738:2001:20a9::5efe:0a00:1402/64`) and to bypass the IPv4-only segment by connecting to the ISATAP router using the `isatapd` module and mechanisms introduced above.

The prepared measurement architecture for the ISATAP scenario can be seen on Fig. 9 (note that the separated NTP network is not shown here).

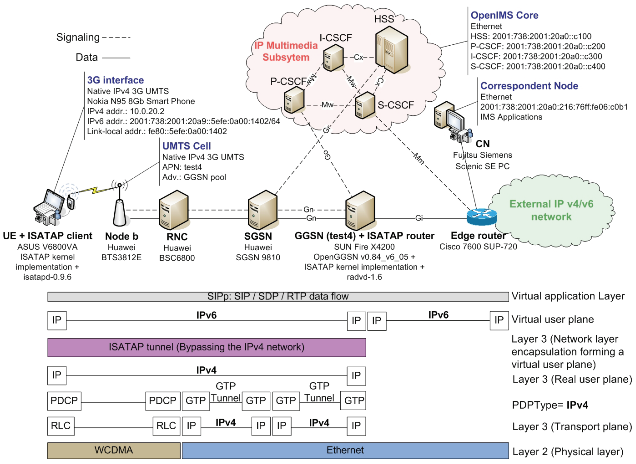
Signaling
Data
IP Multimedia Subsytem
HSS
I-CSCF
P-CSCF
S-CSCF
Cx
Mw
Gi
Go
Mm
OpenIMS Core
Ethernet
HSS: 2001:738:2001:20a0::c100
P-CSCF: 2001:738:2001:20a0::c200
I-CSCF: 2001:738:2001:20a0::c300
S-CSCF: 2001:738:2001:20a0::c400
Correspondent Node
Ethernet
2001:738:2001:20a0:216:76ff:fe06:c0b1
IMS Applications
CN
Fujitsu Siemens
Scienic SE PC
3G interface
Native IPv4 3G UMTS
Nokia N95 8Gb Smart Phone
IPv4 addr.: 10.0.20.2
IPv6 addr.: 2001:738:2001:20a9::5efe:0a00:1402/64
Link-local addr.: fe80::5efe:0a00:1402
UMTS Cell
Native IPv4 3G UMTS
APN: test4
Adv.: GGSN pool
External IP v4/v6 network
Gn
Gi
UE + ISATAP client
ASUS V6800VA
ISATAP kernel implementation +
isatapd-0.9.6
Node b
Huawei
BTS3812E
RNC
Huawei
BSC6800
SGSN
Huawei
SGSN 9810
GGSN (test4) + ISATAP router
SUN Fire X4200
OpenGGSN v0.84_v6_05 +
ISATAP kernel implementation +
radvd-1.6
Edge router
Cisco 7600 SUP-720
SIPp: SIP / SDP / RTP data flow
Virtual application Layer
IP
IPv6
Virtual user plane
ISATAP tunnel (Bypassing the IPv4 network)
Layer 3 (Network layer encapsulation forming a virtual user plane)
IPv4
Layer 3 (Real user plane)
PDCP
GTP
GTP Tunnel
PDPType= IPv4
RLC
Layer 3 (Transport plane)
WCDMA
Ethernet
Layer 2 (Physical layer)

Figure 9. ISATAP-based v4-v6 transition in our 3G UMTS/IMS testbed

7) Teredo

This scenario uses Miredo [39] to provide Teredo client/server/relay functions in our 3G UMTS testbed. Miredo is an open-source Teredo IPv6 tunneling software for Linux and *BSD operating systems. It requires TUNTAP driver (`CONFIG_TUN`) and IPv6 stack support in the kernel, and realizes functional implementations of all components of the Teredo standard (client, relay and server). We installed miredo-1.1.3 on both the UE (with Teredo client functions) and the GGSN (for Teredo server and relay operations). See Fig. 10 for the detailed scheme of our Teredo-extended 3G UMTS testbed architecture.

The installation of Miredo on the UE was performed from binary package. As client mode is the default Miredo behavior, we added only the `ServerAddress` directive in the UE’s `miredo.conf`. According to the Miredo implementation the UE first authenticates with the Teredo server (using the information given in `ServerAddress`), and if successful, it sets up the Teredo tunneling interface with the public Teredo address (`2001:0000:9842:578d:100e:598a:0a00:1402`) and the default IPv6 route constructed/calculated by the implementation. Hereafter, this virtual networking interface will be used to reach the IPv6 Internet and other Teredo clients.

As the Teredo server needs two subsequent IPv4 addresses for operation (it waits for UDP IPv4 packets on port 3544 on both addresses), we set up an additional public IPv4 address on the GGSN’s Gi interface besides the “normal” IPv4 address and the IPv6 connectivity. The `miredo-server.conf` was used to specify the primary and the secondary IPv4 addresses of the Teredo server while on the IPv6 side no special setting was needed.

Miredo makes possible to run Teredo server (i.e., `miredo-server`) and Teredo relay (i.e., `miredo`) instances on the same host. Therefore the relay role was also played by the GGSN and `miredo.conf` was used for specifying the relay type. We applied `RelayType restricted` for our measurements. The relay took care of adding required Teredo IPv6 routing and addressing on the host. However, "non-Teredo" IPv6 addressing/routing requires manual configuration or usage of dynamic routing.

As in all of our measurement setups, a separated NTP network for UE and CN time synchronization was also applied here.

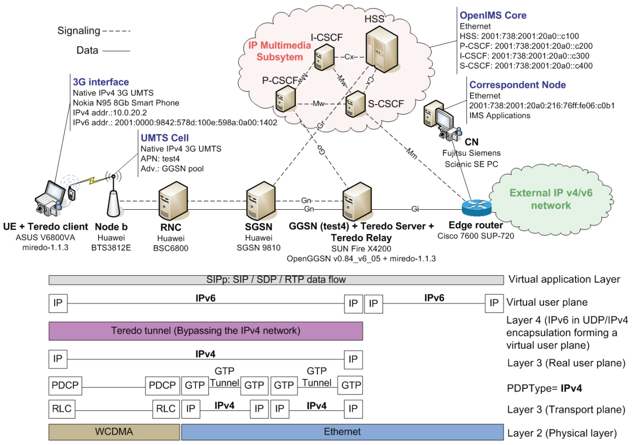

Figure 10. Architecture of Teredo tunneling in our 3G UMTS/IMS testbed

8) NAT-PT

This measurement scenario is to evaluate NAT-PT, which is the most widespread translation-based v4-v6 transition scheme transparently applicable for User Equipments. Our analysis and testbed setup was built upon the NAT-PT implementation called `naptd` [40]. The `naptd` software loosely implements RFC2766 [9] in user space, runs on GNU/Linux operating systems and makes possible to easily setup and configure Network Address Translation - Protocol Translation between IPv6 (as internal) and IPv4 (as external) networks. It was designed to effectively utilize available system resources such to run even on low-end hardware with only one network interface card installed. According to the recommendations, `naptd` uses Address Resolution Protocol (ARP) on the IPv4 and Neighbor Discovery (ND) on the IPv6 network segments while also participates in dynamic routing for both IPv4 and IPv6 if needed.

Usually, NAT-PT implementations cannot translate IP address and subsidiary information carried inside packet payloads. However, some protocols (e.g., DNS, FTP or SIP) require such intervention for proper translation between IP versions. This issue is also solved in `naptd` as different Application Level Gateways (ALGs) are implemented by loadable plugins of the main module.

We applied `naptd` version 0.4 (`naptd-0.4`) in our testbed with some minor modifications to the software's default usage scenario and ALG support: we made it possible to measure the translation use-case between internal IPv4 and external IPv6 networks, and introduced a simple way to provide SIP ALG operation for supporting IMS applications and services embodied by our `SIPp` scripts. This slightly modified naptd-0.4 architecture was installed and configured in our 3G UMTS testing environment (Fig. 11) by giving the roles of the NAT device and ALG functions to the GGSN (i.e., the boundary router situated between the IPv4 and IPv6 network segments). Besides the setup of the dedicated "shadow" NTP network, UEs and CNs did not require additional configuration or modification of their basic software environment in this scenario.

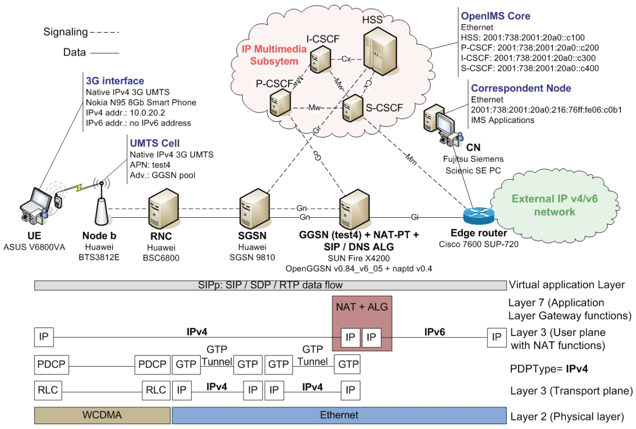

Figure 11. NAT-PT-based v4-v6 transition in our 3G UMTS/IMS testbed

V. PERFORMANCE RESULTS

This section presents the results of our efforts to evaluate the performance of key IMS operations over the above introduced eight scenarios of 3G UMTS access: native IPv4, native IPv6, L2TP, OpenVPN UDP/TCP, 6to4, ISATAP, Teredo, and NAT-PT. The outcomes are presented in boxplots (i.e., box-and-whisker diagrams) to depict the collected numerical data groups through their six-number summaries. The used six representatives are as follows: the lowest sample value (lower line), the lower quartile called Q1 (the lower edge of the box), the mid-quartile or median called Q2 (the delimiter of the two distinctive colors of the box), the upper or third-quartile called Q3 (the upper edge of the box), the largest sample value (the upper line), and the mean of the collected data (red colored rhombus). In our diagrams the Q1-Q2 interval is indicated by grey color and the Q2-Q3 interval is colored with light blue. The height of boxes (i.e., the interquartile range) represents the middle fifty percent of the measured data.

The test cycles for every performance parameter comprised a total of 1000 measured events in every scenario: 1000 RTTs, 1000 IMS Registrations, 1000 Call Setups, and 1000 RTP transmissions, respectively.

The main motivation behind our measurements was to compare native IPv6 3G UMTS network performance with native IPv4, tunneled IPv6 solutions and the most widespread translation-based solution, using key parameters of IMS operations as performance metrics. The comparison based on the access modes (native IPv4, native IPv6, L2TP, OpenVPN UDP/TCP, 6to4, ISATAP, Teredo, NAT-PT) of all the examined v4-v6 transition scenarios revealed an explicit order, which is noticeable almost in all cases. The analyzed key IMS performance metrics show that the fastest solutions are the native IPv4 and native IPv6 access modes as expected, and these are followed by the L2TP, 6to4, ISATAP, Teredo, OpenVPN UDP, NAT-PT and the OpenVPN TCP-based IPv6 solutions.

The measurements regarding the native scenarios (IPv4 and IPv6) reveal a slight advantage of IPv4, especially obvious if considering the IMS Registration Time. However, in some cases IPv6 outperforms IPv4 according to the mean values (e.g., Call Setup Time). Although this was sparse it must be mentioned that IPv6 was always very close to IPv4, and particularly observing the deviation we can say that IPv6 showed quite a well balanced performance. The advantage of native IPv4 at the most of the evaluated IMS metrics could be explained with the smaller address space, and it is also a significant fact in this matter, that IPv4 is a full-fledged protocol – it has been developed for nearly forty years – while the IPv6 protocol stack implemented in the present devices is yet likely to face with some performance issues due to its immature nature.

As also expected the tunneling-based access methods have the worst performance compared to the native solutions in all the scenarios and at all key IMS metrics. The explanation can be found in the general nature of tunneling mechanisms, in the characteristics of the used transport protocols and in the implementations. RP-L2TP uses only packet encapsulation over UDP without any encryption to transmit packets between the two end points of the tunnel, and that simplicity added to the session layer operation made possible to get close to the network (6to4, ISATAP) and transport-level (Teredo) tunneling schemes and to beat application-level (OpenVPN) tunneling solutions together with the also evaluated application-level translation-based method (NAT-PT). The operation of 6to4 and ISATAP requires a lower encapsulation overhead compared to L2TP, Teredo and OpenVPN that both apply UDP/IP or TCP/IP encapsulation. OpenVPN builds up tunnels in an encrypted way using the OpenSSL/TLS library by default thus the tunnel endpoints require more time and resources to process the packet encapsulations and decapsulations. In addition, if the solution uses TCP instead of UDP, the OpenVPN implementation expects acknowledgements after sent packets causing more delay and significant deviation among measurement data. It is generally noticeable that choosing more and more complex mechanisms will cause larger response time and thus worse performance. That is also the main reason of the outcomes of our NAT-PT measurements, which show that translation between different IP versions with application-layer gateway support can provide results only barely better than the most resource consuming OpenVPN TCP solution. However, NAT-PT does not require intervention in UE softwares, which could make the deployment of this transition scheme really fast.

Depending on the observations two main conclusions can be stated. The first one is that nowadays native IPv6 is almost as fast as native IPv4 and in some circumstances it can even outperform its predecessor, although yet IPv6 is an immature protocol and further improvements are expected in the near future. However, no serious deducible difference can be observed between the analyzed IPv4 and IPv6 protocol stacks.

According to the second statement we can say that it is highly recommended to use native IPv6 instead of tunneling protocols in 3G UMTS and beyond, because currently available tunneling methods are much slower and worst balanced than their native counterpart. However we cannot determine significant differences between L2TP, 6to4 and ISATAP, we can say that these above tunneling methods outperform Teredo, OpenVPN UDP/TCP and even NAT-PT in most of the measurement scenarios.

IPv6 will provide enough IP addresses for every piece of device also in an “Internet of Things” era, and the native accommodation of the next generation Internet Protocol also will remarkably simplify the network architecture of mobile and wireless communication systems. However, IPv6 in mobile and wireless networks can not appear in one night, IPv6 will not suddenly provide global coverage. This implies that tunneling-based, translation-centric or other kind of transition techniques need special care and explicit attention, despite the fact that they perform worse and show significant overhead compared to the native cases.

VI. Conclusion and Future Work

The research presented in this paper mainly concerned the questions and challenges of the transition from IPv4 to IPv6 in all-IP 3G and beyond multimedia systems, and their impacts on the performance of IMS services and applications. In order to quantify the effects of different methods providing IPv6 support/transition techniques in existing mobile telecommunication architectures, we designed and implemented a 3G UMTS testbed (including the IMS core) and compared the performance characteristics of several selected transition techniques (L2TP, OpenVPN UDP, OpenVPN TCP, 6to4, ISATAP, Teredo, NAT-PT) with native IPv4 and IPv6 scenarios using key IMS operations as performance metrics. Our results exposed the main benefits and drawbacks of the examined technologies based on their actually available implementations, and highlighted some strict limitations concerning the non-native IPv6 support so we must stress the need for further studies aiming to help and urge the process towards the global native IPv6 coverage.

As a part of our future work we are planning to extend the evaluation of heterogeneous scenarios (i.e., v4 caller communicates with a v6 callee and vice versa) using other translation-based transition mechanisms (e.g., BIS, BIA, TRT, SOCKS64), application layer gateways and proxies. We are also devoted to analyze some yet missed tunneling mechanisms (6over4, DSTM, Proto41, AYIYA/AICCU etc.). We also would like to extend our experimental approach with extensive and detailed overhead measurements of different IPv4/IPv6 transition techniques.

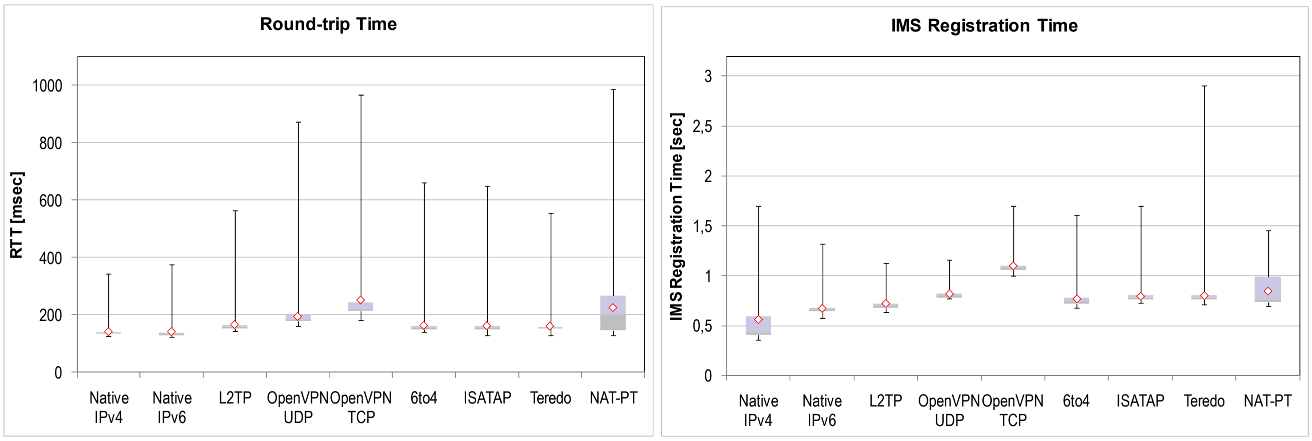

Figure 12. Round-trip Time and IMS Registration Time

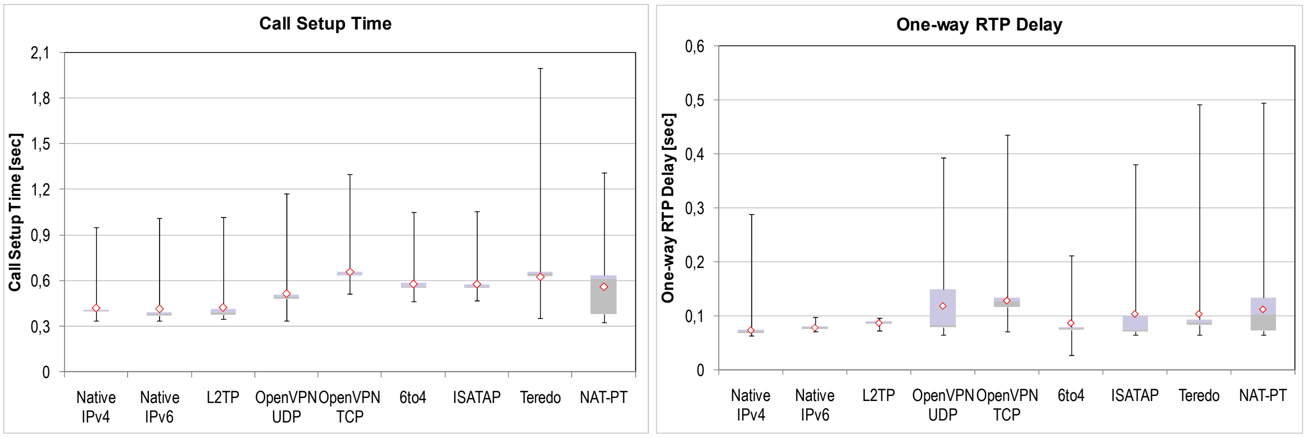

Figure 13. Call Setup Time and One-way RTP Delay

ACKNOWLEDGMENT

This work was supported by the Hungarian Government through the TÁMOP-4.2.1/B-09/1/KMR-2010-0002 project at the Budapest University of Technology and Economics together with the help of Mobile Innovation Centre Hungary and the IST-ANEMONE project, which was partly funded by the Sixth Framework Programme of the European Commission's Information Society Technology. The authors also would like to express their appreciation to László Madarassy and Szabolcs Kustos for their essential work on this research.

REFERENCES

[1] L. Bokor, Z. Kanizsai, and G. Jeney, "Performance Evaluation of Key IMS Operations over IPv6-capable 3G UMTS Networks", In proc. of the Ninth International Conference on Networks (ICN 2010), pp. 1-10, ISBN: 978-0-7695-3979-9, DOI: DOI:10.1109/ICN.2010.49, Les Menuires, France, April 2010.

[2] S. Deering and R. Hinden, "Internet Protocol version 6 (IPv6): Specifications," IETF RFC 2460, Dec. 1998.

[3] A. Cuevas, J.I. Moreno, P. Vidales, and H. Einsiedler, "The IMS Service Platform: A Solution for Next-Generation Network Operators to Be More than Bit Pipes", *IEEE Com.M.* V.44, N.8, pp.75-81, 2006.

[4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol", IETF RFC 3261, June 2002.

[5] 3GPP TS 23.060 : "General Packet Radio Service (GPRS); Service description", Stage 2, (Release 9), V9.2.0, September 2009.

[6] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS)", Stage 2, (Release 9), V9.2.0, December 2009.

[7] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", IETF RFC 3056, February 2001.

[8] F. Templin, T. Gleeson, M. Talwar, and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", IETF RFC 4214, October 2005.

[9] G. Tsirtsis and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", IETF RFC 2766, February 2000.

[10] J. Hagino and K. Yamamoto, "An IPv6-to-IPv4 Transport Relay Translator", IETF RFC 3142, June 2001.

[11] P. Srisuresh, G. Tsirtsis, P. Akkiraju, and A. Heffernan, "DNS extensions to Network Address Translators (DNS_ALG)", IETF RFC 2694, September 1999.

[12] S. Thomson, C. Huitema, V. Ksinant, and M. Souissi, "DNS Extensions to Support IP Version 6", IETF RFC 3596, October 2003.

[13] 3GPP TS 29.061: "Interworking between the Public Land Mobile Network (PLMN) supporting packet based services and Packet Data Networks (PDN)", (Release 9), V9.4.0, September 2010.

[14] J. Lau, Ed., M. Townsley, Ed., and I. Goyret, Ed., “Layer Two Tunneling Protocol - Version 3 ”, IETF RFC 3931, March 2005

[15] W. Simpson, Ed., “The Point-to-Point Protocol (PPP)”, IETF RFC 1661, July 1994

[16] OpenVPN Technologies, http://www.openvpn.net [Accessed: Jan. 13, 2011]

[17] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2”, IETF RFC 5246, August 2008.

[18] C. Huitema, “An Anycast Prefix for 6to4 Relay Routers”, IETF RFC 3068, June 2001.

[19] Z. Xiaodong, M. Yan, and Z. Yumei, “Research on the Next-Generation Internet Transition Technology”, In proc. of the Second International Symposium on Computational Intelligence and Design (ISCID '09), pp. 380-382, Changsha, China, 2009.

[20] S.D. Lee, M.K. Shin, and H.J. Kim, “The implementation of ISATAP router”, In proc of the 8th International Conference on Advanced Communication Technology (ICACT'06), pp. 1163, Phoenix Park, Republic of Korea, 2006.

[21] Y. Hei and K. Yamazaki, “Traffic analysis and worldwide operation of open 6to4 relays for IPv6 deployment”, In proc. of International Symposium on Applications and the Internet, pp. 265-268, 2004.

[22] C. Huitema, Teredo: “Tunneling IPv6 over UDP hrough Network Address Translations (NATs)”, IETF RFC 4380, February 2006.

[23] S.M. Huang, Q. Wu, and Y.B. Lin, “Enhancing Teredo IPv6 tunneling to traverse the symmetric NAT”, IEEE Communication Letters , 10 (5), 408-410, 2006.

[24] G. Tsirtsis and P. Srisuresh, “Network Address Translation - Protocol Translation (NAT-PT)”, IETF RFC 2766, February 2000.

[25] E. Nordmark, “Stateless IP/ICMP Translation Algorithm (SIIT)”, IETF RFC 2765, February 2000.

[26] C. Aoun and E. Davies, “Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status”, IETF RFC 4966, July 2007.

[27] SIPp test tool and traffic generator, http://sipp.sourceforge.net [Accessed: Jan. 13, 2011]

[28] 3GPP TS 24.228: “Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)”, Stage 3, (Release 5), V5.15.0, September 2006

[29] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, IETF RFC 3550, July 2003

[30] Tshark: The terminal oriented version of Wireshark network protocol analyzer, http://www.wireshark.org [Accessed: Jan. 13, 2011]

[31] Budapest University of Technology and Economics - Mobile Innovation Centre (MIK), http://www.mik.bme.hu [Accessed: Jan. 13, 2011]

[32] OpenGGSN on SourceForge: http://sourceforge.net/projects/ggsn [Accessed: Jan. 13, 2011]

[33] L. Bokor, Z. Kanizsai, and G. Jeney, “Setting up native IPv6 UMTS access with open-source GGSN implementation” [Online] Available: http://www.ist-anemone.eu/index.php/Setting_up_native_IPv6_UMTS_access_with_open-source_GGSN_implementation [Accessed: Jan. 13, 2011]

[34] Fraunhofer FOKUS NGNI, OpenIMSCore Project, Official website: http://www.openimscore.org [Accessed: Jan. 13, 2011]

[35] Linux IPv6 Router Advertisement Daemon (radvd), http://www.litech.org/radvd [Accessed: Jan. 13, 2011]

[36] Roaring Penguin L2TP, http://rp-l2tp.sourceforge.net [Accessed: Jan. 13, 2011]

[37] P. Bieringer, “Configuring 6to4 tunnels”, Linux IPv6 HOWTO on TLDP, http://tldp.org/HOWTO/Linux+IPv6-HOWTO/configuring-ipv6to4-tunnels.html [Accessed: Jan. 13, 2011]

[38] S. Hlusiak, “*isatapd*–ISATAP client for Linux”, http://www.saschahlusiak.de/linux/isatap.htm [Accessed: Jan. 13, 2011]

[39] R. Denis-Courmont, “Miredo : Teredo IPv6 tunneling for Linux and BSD”, http://www.remlab.net/miredo [Accessed: Jan. 13, 2011]

[40] L. Tomicki, “Network Address Translation, Protocol Translation IPv4/IPv6”, http://tomicki.net/naptd.php [Accessed: Jan. 13, 2011]