Telecommun Syst (2018) 68:393–407
https://doi.org/10.1007/s11235-017-0400-6

Assembling SIP-based VoLTE Call Data Records based on network monitoring

Tamás Tóthfalusi[1] · **Pál Varga**[2]

Published online: 1 November 2017

Abstract The Voice over LTE (VoLTE) service is under deployment at many operators. Although it has great promises—from high service flexibility to previously unseen QoS measures—so far it poses more challenges than expected. The serving architecture is diverse, and the flexible services mean variable use-cases with complex message sequence charts. In order to be able to control the procedures—during deployment as well as during the operation—we need to have proper network and service management in place. Such a management framework includes network and service monitoring tools that help to provide better visibility of the various procedures. For the VoLTE service, both the LTE Evolved Packet Core (EPC) and the IP Multimedia Subsystem (IMS) need to be monitored. Network-wide data capture and analysis for the EPC and the IMS require new processing methods. These would allow operators to correlate control and user plane information of various interfaces and protocols. There are many obstacles to overcome here, including ciphered control messages and global identifiers hidden by temporary ones. This paper presents a system for SIP Call Data Record assembling, shows what kind of key parameters have to be extracted in order to enable expert analysis. After going into the details of messages, transactions, and dialogues, we present Call Data Record assembling methods for various scenarios. Furthermore, we present methods on how to gather cross-correlated data on specific fault management use-cases, particularly for unsuccessful voice calls.

Keywords VoLTE · Network monitoring · CDR generation · IMS · SIP

1 Introduction

Wireless data traffic is increasing exponentially worldwide [8]. Supporting and managing this growth of traffic on the signaling links poses a great challenge to the operators. Fault management—especially the detection and the root cause analysis of failures—has become very complex, and requires deep telecommunication knowledge. The main challenge here is to understand the 2G, 3G, and 4G mobile core—and not only their architectures, but their interworking as well. There are key identifiers that get mapped from one to the other while the subscriber is wandering among these technologies; these should be tracked and handled properly. Furthermore, The IP Multimedia Subsystem (IMS), which is responsible for service control—that includes call control, among others—adds another factor of complexity. IMS has a quite different type of organizational philosophy when compared to those of the mobile cores [2].

VoLTE is a new service—it uses the LTE infrastructure, as its name suggests. Beside this, it utilizes IMS, a set of entities where telecommunications operators handle the IP-based call control function, and various other control functions for IP-based services. Moreover, the VoLTE service has to interact with the 2G and 3G infrastructure, since callers and callees are often attached to those, while simultaneously being attached to 4G—and this can complicate the call tracing.

✉ Pál Varga
pvarga@tmit.bme.hu

Tamás Tóthfalusi
tothfalusi@aitia.ai

[1] Telecommunication Division, AITIA International, Inc., 48-50. Czetz J. Str., Budapest 1039, Hungary

[2] Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, 2. Magyar Tudósok körútja, Budapest 1117, Hungary

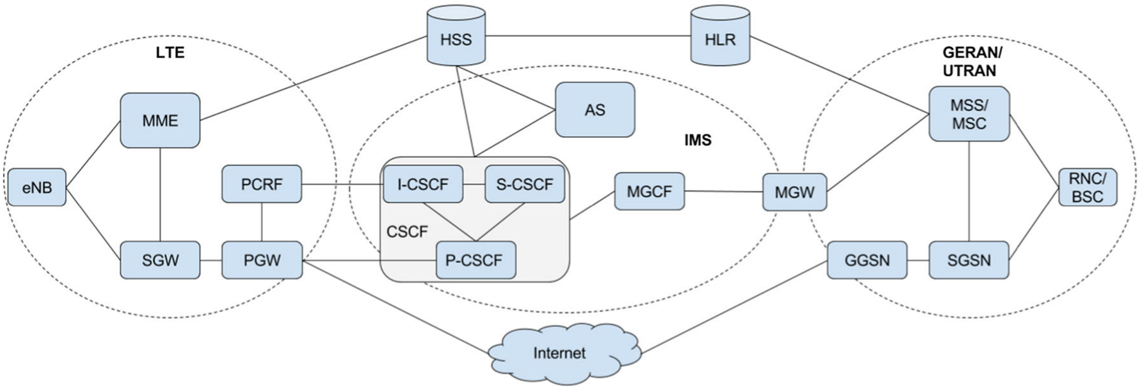


Fig. 1 Core network architecture serving Voice over LTE—including 2G, 3G, 4G and IMS components

From the network and service management point of view, it is important to gather and analyze the control traffic. In order to see a proper big picture with all the details, the control traffic captured on various interfaces has to be correlated—i.e. identifiers appearing on one interface help to assemble session data records of an other interface [16]. In relation to VoLTE session and call establishment, the key information-exchange points—links between the functional nodes—are depicted in Fig. 1. A brief summary of each nodes' function can be found in Sect. 2.

Passive monitoring is supposed to be lossless: when the links are tapped, and the probes receive data in a non-intrusive manner, they cannot ask for resending anything. What they missed seeing, they have lost capturing. Based on the monitoring data, engineers can support performance management, network optimization, as well as failure detection, which is one of the most important tasks for operations and maintenance. This paper discusses the requirements and the functions of a VoLTE monitoring system, which is under deployment. Furthermore, the paper presents some practical use-cases on call tracing, as well.

The current paper is organized in the following way. Section 2 provides a technical recap for the main material of the paper, covering the LTE and IMS architecture, and some monitoring background for SIP (Session Initiation Protocol) specialities. Section 3 details IMS-related monitoring considerations: which links to monitor, and what information is expected to be gathered there. Furthermore, this chapter presents the assembly mechanism for monitoring-based CDR (Call Detail Record) generation. Section 4 presents call scenarios that are either typical, or need special attention in relation to monitoring and CDR assembly. Section 5 provides suggestions for CDR assembling: summarizes the requirements gathered, based on the various call scenarios, and describes how to collect and correlate user-related identifiers in order to cover the requirements. Section 6 concludes the paper.

2 Technical background and related work

2.1 LTE architecture

The LTE Evolved Packet Core (EPC) [20] comprises merely packet switched network elements; it only supports the legacy circuit switched functions through IP-based packet transfer.

- *MME (Mobility Management Entity)* The MME lies on the border of the EPC and EUTRAN (Evolved Universal Mobile Telecommunications System Terrestrial Radio Access Network)—and is mainly responsible for mobility-management. Its main functions include controlling handovers between eNodeBs (Evolved NodeB, Base Station), SGW (Serving Gateways) or MMEs, connecting to HSS (Home Subscriber Server), user identification, authentication and controlling the roaming functions. MME registers and handles the User Equipment (UE) in his own area, registers where the UE is located, either exactly at eNodeB level (if communication is active), or within a Tracking Area (TA), which is designated to a group of eNodeBs (in case of passive UE, no active connection is needed).
- *SGW (Serving Gateway)* The SGW is responsible for user traffic stream handling, and controlling the allocation of resource capacities, the changes or deletion of sessions and finishing IP connections. From the eNodeB point of view, it is a fixed anchor node through which the core network elements can be accessed. Furthermore, the SGW controls the User Plane tunnels with GPRS Tunneling Protocol (GTP) [5], although this can also be guided by the MME or the PGW—depending on the process rules.
- *PGW (Packet Data Network Gateway)* PGW can be seen as an edge node of the EPC, since it ensures the connections to external data networks (e.g., the Internet or a private corporate network) and handles the UE's data traffic that is entering or leaving the core network. Besides, it

offers interfaces for further functions such as QoS control or billing.

- *HSS (Home Subscriber Server)* HSS takes the roles of HLR/AuC (Home Location Register/Authentication Centre) in the LTE network. It can be seen as the ultimate data storage that contains the subscribers' service-related data. The HSS stores the subscribers' profile, containing the enabled services and accesses (e.g. allowed roaming services to external networks). It is mainly responsible for access management and authentication, and it also registers the subscribers' position within the network. The HSS cooperates with the MME in all UE-related change events that are administered by the EPC.
- *PCRF (Policy and Charging Rules Function)* Based on its predefined policies and QoS-related rules, the PCRF sends control information to the PGW. This set of information is called "Policy Control and Charging rules", and it is exchanged between the PCRF and the PGW when a new bearer is set up, e.g. in case a new UE activates new PDP to the network or a new UE requires a data plane bearer with a different QoS policy.

2.2 IMS architecture

The IMS (IP Multimedia Subsystem) is a global architecture, constructed as a standardized platform [2] for the different telecommunication networks. The basic idea is producing a common, Internet based architecture to grant communication between different telecommunication networking technologies (e.g. 2G, 3G, 4G, fixed phone, Internet). Figure 1 depicts the architecture of an IMS domain.

Since these different technologies use various call control protocols in their various interfaces, IMS introduced a homogeneous usage of call control, through SIP (Session Initiation Protocol [25,26]). This also means that technology-dependent call control protocol messages and parameters need to be translated to SIP. Such protocols are, e.g. ISUP (ISDN User Part) [13] between fixed telephone exchanges, or BSSAP (Base Station System Application Part) [1] for the 2G access network. There are standardized protocol converter functions defined in order to handle these multi-protocol dependencies. As an example, MGCF (Media Gateway Control Function [3]) operates at the edge of the IMS domain to cover this functionality, among others.

The main benefits of introducing IMS are:

- homogenized handling of multimedia services within the control plane,
- providing QoS support for real-time multimedia services,
- enabling IP-based multimedia services for mobile users.

The central IMS modules—that will be introduced in the upcoming paragraphs—often serve millions of users, which opens major networking challenges to solve. In order to reduce the network load, the traffic can be distributed by using prefilter modules. However, it should be noted that the centralized functions obstruct scalability, hence they could lead to architectural drawbacks.

IMS is essentially a control system [21], which enables for the registered users to find other users and application servers—in order to build up multimedia connections (e.g. conference call, instant messaging). The architecture manages and maintains merely the control messages; the media traffic is routed independently of the control traffic between the end users (e.g. RTP protocol based user traffic).

The central element of the architecture is the CSCF (Call Session Control Function). It is functionally split into three submodules:

- P-CSCF (Proxy Call/Session Control Function),
- S-CSCF (Serving Call/Session Control Function), and
- I-CSCF (Interrogating Call/Session Control Function).

Figure 2 represents the differentiated CSCF-functions within a domain. These submodules serve routing and management tasks, and operate together as the main CSCF network node.

The P-CSCF part allows a secure entry point for the end users. It often uses IPSec to grant secrecy for IMS-related traffic.

The main task of the S-CSCF module is to build up and manage connections between the IMS-registered users. It operates as the central element of the signaling plane, and handles the registration and user services. During the user authentication procedure, the S-CSCF requests user related information from the AuC (Authentication Center) through the HSS (Home Subscriber Server), which are key elements of 3G and 4G mobile architectures. After the successful registration, it handles the user-control messages and acts as an anchor point for the registered users.

The Interrogating part—so called I-CSCF—is located at the edge of the IMS domain, and acts as a forwarding point for remote servers. In general, it supports the communication from and towards other IMS domains [21].

The central control plane grants access independency for the end users, through protocol converter components and gateway components towards the Internet world. The MGCF (Media Gateway Control Function) module supports ISUP - SIP signaling protocol conversion. It enables the connection from 2G/3G mobile networks. The MSAN (Multi Service Access Node) network element is for fixed phones; it allows communication through the IP-based systems. VoIP and VoLTE phones, or other devices with SIP protocol capabilities are connected through IP-clouds to the IMS domain, using a GGSN, a PDN GW (Packet Data Network Gateway), or just a router, depending on the media access.

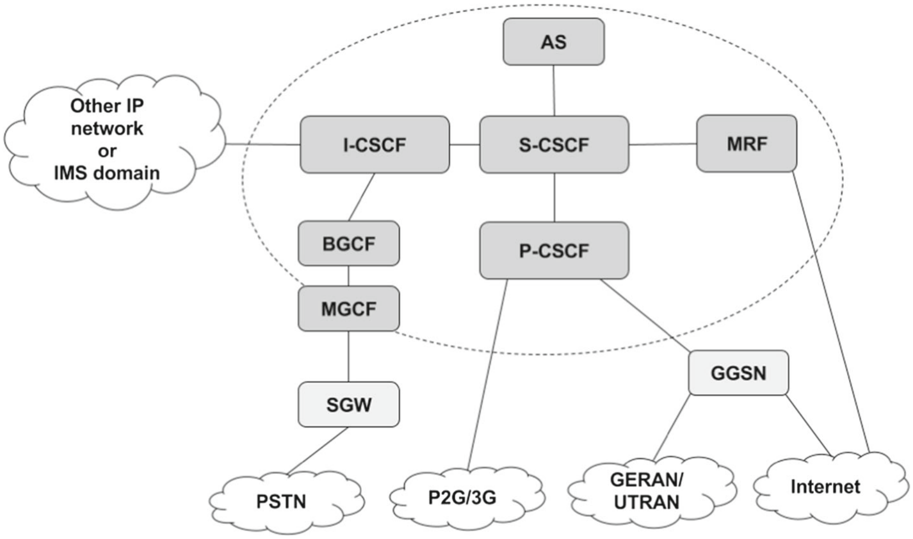


Fig. 2 IMS architecture, with the differentiated CSCF submodules

Beside CSCF, IMS contains further modules (e.g. Application Server, Media Resource Function), which are differentiated based on their services. In this paper we concentrate only on those main nodes, which play key roles in a session's lifetime.

The AS (Application Server) is directly connected to the S-CSCF submodule. It is necessary to allow telephony and other multimedia services for the registered users. The module contains the management logic itself to provide and execute multimedia services (e.g. videoconferencing, gaming, file sharing).

The MRF (Media Resource Function) is responsible for media stream control. It can be separated to MRFP (Media Resource Function Processor) and MRFC (Media Resource Function Control) submodules. MRFP supports media manipulation (e.g. playback function, playing tones), conferencing, DTMF user activity. MRFC is the controller part, which manages the resources of MRFP.

2.3 Understanding SIP from the monitoring viewpoint

The main signaling protocol of the IMS is SIP (Session Initiation Protocol). It uses IPv4 or IPv6 in the network layer, and UDP, TCP or SCTP in the transport layer. Since it is a text-based protocol, the order of the fields can be varied—as well as the size of parameters. Explaining the tasks of each message and parameter field is not within the scope of this paper—we merely describe the essential parts that are needed for understanding CDR (call data record) generation algorithms.

Let us examine SIP protocol messages from the monitoring and protocol decoder viewpoint. Based on the best practices of protocol technology, we can differentiate two types of messages: request and response. The first line of each message has a mandatory format, which greatly supports the type-recognition process.

According to the SIP terminology, the first line of a Request message is called Request-Line, and its parameters are Method, Request-URI and SIP version. Parsing a Response message, the first line is called as Status-line, in which the mandatory parameters are: SIP version, Status code, and Reason phrase. The standard also defines the communication partners. The UAC (User Agent Client) generates the Request message, and the UAS (User Agent Server) generates the Response.

The basic request messages are Register, Invite, Bye, Cancel, Options, Ack, Subscribe, Notify, Message, and Prack [23,26]. Among these, the Invite message marks the start of each call—hence it is very important when generating SIP call records.

The Response messages are also human readable, and have two parts: a (numerical) Status code and a (textual) description (e.g. 200 OK). The first digit of the 3-digit Status code is the class code. The 1XX class is for the provisional responses (e.g. Trying, Ringing, Session Progress). The 2XX class means successful operation, and 3XX reports a redirection event. The 4XX, 5XX and 6XX classes are used to indicate error or failure events.

The SIP terminology differentiates three types of communication: transaction, dialog and session.

A SIP Transaction starts with a request message, and ends with a final response.

A SIP Dialog is a longer conversation between the end users, containing many transactions. A Dialog can be clearly identified by the Dialog-ID, which is concatenated from the value of the Call-ID field, the tag parameter of the From field, and the tag parameter of the To field. This triplet defines a SIP Dialog between the UAC and UAS. Based on the request message types, two Dialogs can be differentiated: the Invite-

and the Subscribe-dialog. The Subscribe-dialog is used to sign up for a service or listening an event.

The Invite-dialog sets up a multimedia (e.g. for audio, video, gaming) session, called as SIP Session. The Session is built-up successfully, when a three-way handshake (INVITE, 200 OK, ACK messages) at the beginning of the Dialog is successful. The messages within the Invite-dialog (e.g. INVITE, PRACK, 200 OK, 180 Ringing) often carries SDP (Session Description Protocol) protocol [10] to manage the setup procedure.

2.4 Related work

Tatai et al. [27] presented the flexible and economical traffic mass measurement system called SGA-7N, which was developed for GSM systems. The system has been growing in features since then, and scaling up very well, covering UMTS, and LTE systems, as well as the IN (Intelligent Network) and the IMS domains. The extended SGA-NETMON system [4] provided the technological basis of the network monitoring results presented in this paper.

Breda and Mendes [6] proposed a QoS monitoring and failure detection solution in voice communication. The algorithm is based on CDR records. The proposed system classifies the generated CDRs into predefined events, e.g. Carrier loss, Called party does not answer. The authors found that the QoS categorization is efficient on large event set.

Lutiis and Lombardo [18] proposed a monitoring tool that focuses on the security features for NGN (Next Generation Network) and IMS (IP Multimedia Subsystem). The authors describe that the main challenge for a monitoring system is to handle the ever growing subscriber base, and it must be scalable to work properly even during the peak hours. The proposed anomaly detection algorithm (SAD—SIP Anomaly Detection) is based on CDRs, and it calculates an E entropy value for 60 minutes time-windows.

Nassar et al. [19] proposed an intrusion detection method for SIP protocol based IMS interfaces. The work is based on anomaly and attack classification. Beside the SIP record profiling, they also examine the billing information and the server logs.

Raouyane et al. [22] aimed to verify the QoS in IMS networks using eTOM (enhanced Telecom Operations Map), as an IMS monitoring and Management system. The authors applied web services to demonstrate the distributed architecture. The work defines the required data collection points of the network, and calculates general KPIs (e.g. response time, availability).

Hoffstadt et al. [11] designed a SIP protocol based monitoring and analyzing system (STR—SIP Trace Recorder). The software is written in Java language and uses multiple CPU cores. The message flow can be differentiated into three steps: capturing, parsing and storing into database. During the automatic analysis phase the STR system is used to analyze SIP-based attacks in different scenarios. To evaluate the system, the authors generated attack signatures.

Zhang et al. [29] proposed a monitoring system for SIP protocol, which is able to deal with one Gbit/s speed traffic flow. The paper deeply analyzes the SIP session and call procedures. To test the capability of the work, the authors used their proprietary laboratory environment as a test-bed. The results show that the software-based implementation is able to deal with 6000 simultaneous SIP sessions.

Balcerzak et al. [5] proposed a monitoring model based on the registration procedure, in order to detect anomalies in IMS domains. Since the SIP Register messages take a great proportion (in their model: about 35%) of the total SIP signaling of the IMS, it could be an efficient base for first level problem diagnosis. The authors defined different KPIs based on registration-specific counters.

Hyun et al. [15] proposed a Deep Packet Inspection (DPI) method to classify VoLTE traffic, using the SIP User-Agent field. The classification method is able to achieve 3.8 Gbit/s processing speed to differentiate SIP and also RTP traffic. The authors used captured, real LTE network traffic for the evaluation phase. They further elaborated this work in [14].

Li et. al. [17] presented their security assessment methods and metrics on one of the first VoLTE deployments. They discovered several vulnerabilities in both the control and the data plane. During their analysis they had various interesting findings, including that the device OS and chipset fail to prohibit non-VoLTE apps from accessing and injecting packets into VoLTE control and data planes. Such vulnerabilities pose extra tasks on network failure management, as well.

3 IMS network monitoring and CDR assembling

3.1 Monitoring considerations for the IMS domain

In order to compile Call Data Records based on IMS monitoring, the primary task is to identify the monitoring points within the network. This can be done through understanding the IMS architecture, as well as its purpose within Voice over LTE call-control.

Among other tasks, the P-CSCF plays the security guards role for IMS: all its external connections are ciphered through IPSec. The call control messages arriving from the PGW (Packet Data Network Gateway) or GGSN (Gateway GPRS Support Node) enter the IMS through the P-CSSF—secured by IPSec. This makes the monitoring of this interface very complex—and still, it reveals very little about the IMSs behavior.

Since the P-CSCF is mostly responsible for security-related issues, faults during call control can be captured on the internal leg of P-CSCF, as well. It would make sense to

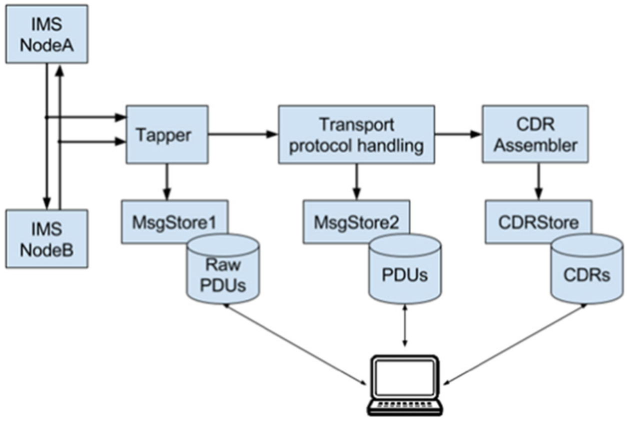


Fig. 3 Monitoring architecture with the stages of on-the-fly data processing

monitor this leg—in theory. However, in practice, IMS vendors often integrate the logical P/I/S-CSCF modules into one physical entity. This means that the internal IMS-interfaces are not available for any kind of monitoring. On the other hand, if any traffic between the P/I/S-CSCF entities is accessible, it is worth taking that/those interface(s) into account as extra monitoring point(s).

The Application Server and the MMTel (Multimedia Telephony Service) modules of IMS get call control information as well, since they have to maintain data related to charging, user authorization, and QoS-control. This way the external interfaces of these modules should also be considered as monitoring points.

Among its other features, IMS is there to support technology integration for 2G, 3G and 4G mobile call control. This is handled by various modules—such as the MGCF (Media Gateway Control Function)—, hence monitoring of their interfaces is also important.

Figure 3 depicts a multi-purpose, high-level monitoring architecture for control plane—in this case: IMS-traffic.

This monitoring architecture is based on a modular, pipelined solution. The elements of the pipeline have the following functions:

- *Tapper* This module is the first software component in the chain. It takes Ethernet frames from the hardware (i.e. network monitoring cards) and captures the traffic continously. It applies filter rules to drop the unnecessary Ethernet frames and to forward the remaining ones for further processing. The module optionally cuts the Ethernet header and VLAN tags, and forwards only the IP packet to the next node.
- *MsgStore1* The main function of the MsgStore1 module is to store the captured messages bit-by-bit. It uses a physical storage element (Raw PDUs), and stores the incoming data in an encrypted form.
- *Transport protocol handling* The IP-level fragmentation and the TCP and SCTP segmentation are handled here. This element outputs the assembled, application-level PDUs.
- *MsgStore2* This module stores the assembled, application-level PDUs—in this case: SIP messages—with a custom header that contains extracted information from the network and transport layers.
- *CDR Assembler* The assembler operates on the SIP messages, using an internal algorithm to prepare the Call Detail Records. The algorithm is based on the SIP protocol and on the functional behavior of the IMS.
- *CDRStore* This function is part of the CDR Assembler module. It stores the generated records on a physical storage (CDRs).

The information processing flow of the captured traffic is the following.

The Tapper module continuously captures the traffic of the previously described interfaces. It forwards these messages to a bit-by-bit storage (MsgStore1) module for raw PDUs (Protocol Data Units). Simultaneously, it forwards the messages to further modules, for on-the-fly processing.

In order to analyze the SIP protocol-content, the transport protocol (e.g. IPv4/IPv6, UDP, TCP, SCTP) encapsulation needs to be handled. The SIP content can be broken as a result of IP fragmentation, or TCP as well as SCTP segmentation. These needs to handled in order to obtain the original SIP content in one piece. It is rational to store the purified SIP content as well (MsgStore2), to ease the operator's work and further processing tasks.

The final stage of on-the-fly processing is the generation of Call Detail Records. The CDR Assembler takes the SIP messages as input, and uses its internal algorithms (that are going to be detailed in further sections) to generate the CDRs.

Operators can search, obtain and analyze these CDRs independently from the generation process. Beside the general requirement that CDRs must be searchable by various, user- and call-related identifiers, very often the actual messages that induced the CDR should be offered for observation. This feature is available in the architecture presented in Fig. 3, since the MsgStore1 module has saved the raw PDUs in the beginning of the processing chain. Finding root causes of faults related to IMS misbehavior can be eased by using the above described architecture for on-the-fly processing of protocol data

4 Complex SIP call scenarios in IMS

SIP transactions and dialogs can be differentiated based on their initial request message. Since some of these request types are initiated (and then closed) asynchronously from

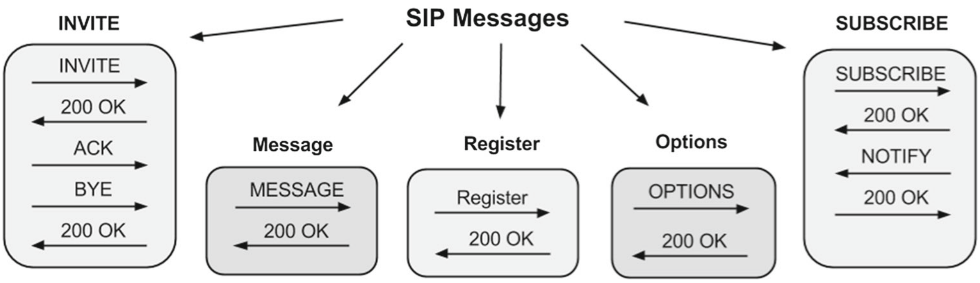


Fig. 4 Different SIP CDR types

each other, we should define different record types for each. There can be different Call Data Records built, based on these initial messages: INVITE, REGISTER, SUBSCRIBE, OPTIONS, MESSAGE, and INFO. Figure 4 depicts the CDR types with their basic, CDR-related messages.

Out of these, the INVITE CDR type is the one that contains the actual Call Detail Record, in the classic sense. In other words, an INVITE CDR is a SIP call dialog, which starts from the initial Invite message and finishes at the end of the call (with various ways). Since a call setup procedure contains more transactions, INVITE CDRs include Update [24], Cancel and often Info [9] transactions, as well. In the next subsections, we describe a few typical scenarios for SIP call setup, and show techniques on how to recognize the coherent messages within the IMS domain. Based on these scenarios, we introduce a general CDR collection algorithm in "The enhanced ruleset" subsection, to assemble SIP dialog messages into records.

Nevertheless, not all Invite messages mark a newly starting call. Some Invite messages do actually start the call, while others arrive during the same call, carrying feature modification information.

We differentiate the Invite messages in the call flow, based on the value of the To field tag parameter. If the To tag parameter does not exist, we consider it as an initial-Invite Request (starting a call), otherwise it is a Re-Invite (modifying parameters, e.g. QoS settings) [7].

Considering Invite-dialogs, they start with an initial-Invite request message, and finish when one of the following events occurs:

- One of the communication partners sends a Cancel request message, which causes 487-Response message to the Initial-Invite;
- Bye request and 200 OK Response pair;
- 3XX, 4XX, 5XX or 6XX response to the initial-Invite.

In later sections (Complex practical cases and considerations) we show exceptions, which Status code belongs to 4XX-class, and do not finalize a dialog.

4.1 The simple case

The trivial way to follow a SIP call's control through passive monitoring is by using the Dialog-ID. The Dialog-ID is a parameter derived from three field parameters of the SIP message: Call-ID, From tag, and To tag. The Call-ID is a unique ID within the IMS domain at a given time. Besides, the From and To tag parameters are minimum 32-bit wide truly random numbers. Altogether, this triplet (the Dialog-ID) ensures strong identification. The Dialog-ID clearly associates the INVITE- or SUBSCRIBE dialogs, and can be used to pair request-response messages of other transactions.

The preparation of the Dialog-ID is a two-step process. The initial message between two User Agents never contains a To tag. After receiving the Request message, the UAS generates the Dialog-ID and sends it back within the To field of the Response message. After this tag-exchange procedure, each user knows the full Dialog-ID, and uses it as a unique ID of the call. The IMS also records this ID for routing functions.

In the case of simple Request-Response transactions (e.g. OPTIONS, MESSAGE), which contain only two messages, every new Request triggers a new To tag generation process. In contrast, SIP dialogs always use one To tag during the dialog lifecycle. Similarly to the two-message transactions, the initial-Invite message does not contain a To tag. Actually, this is the property that is used to identify it as an initial-Invite.

According to RFC3261 [26] the UAS generates a To tag parameter (in its response), when the request does not contain it. This also means that the CDR assembling algorithm does not have any To tag to use (as part of the Dialog-ID) until the final Response of the initial-Invite arrives.

As an example of such a case, let us take a look at Fig. 5. This shows a SIP dialog ended by a Cancel transaction. During the call cancellation procedure, the Cancel request message must contain the same From and To field parameters, as the initial-Invite message did. As discussed earlier, the initial-Invite does not contain any To tags, hence this parameter will be empty in the Cancel message, as well. The reception of a Cancel request (with empty To tag) triggers a new tag-generation process—and the "200 OK (CANCEL)"

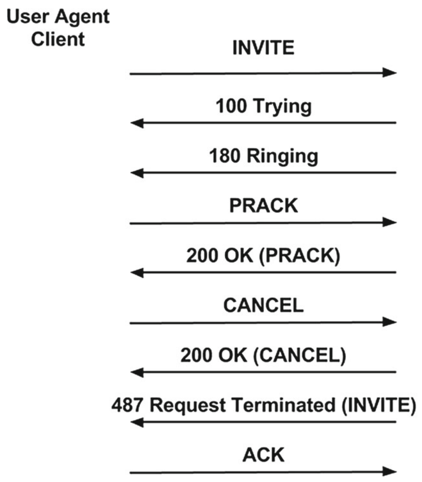


Fig. 5 Signaling messages of a cancelled INVITE dialog

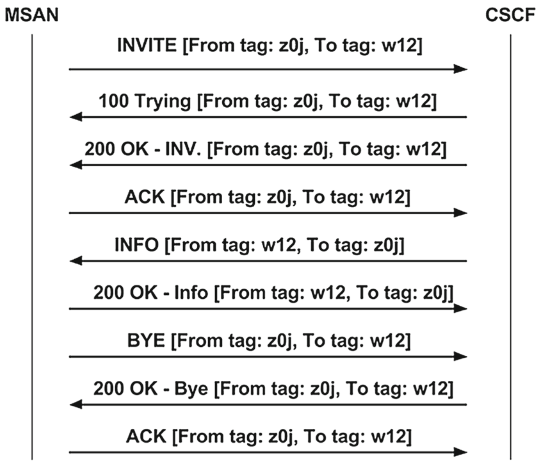

MSAN
CSCF
INVITE [From tag: z0j, To tag: w12]
100 Trying [From tag: z0j, To tag: w12]
200 OK - INV. [From tag: z0j, To tag: w12]
ACK [From tag: z0j, To tag: w12]
INFO [From tag: w12, To tag: z0j]
200 OK - Info [From tag: w12, To tag: z0j]
BYE [From tag: z0j, To tag: w12]
200 OK - Bye [From tag: z0j, To tag: w12]
ACK [From tag: z0j, To tag: w12]

Fig. 6 From and To field parameter values change their place in INVITE dialog messages

response hence contains a new To tag, different to the ones seen in previous dialog messages. As this simple example reveals, although the Dialog-ID is a strong key, its usage in general is cumbersome.

Until waiting for the final response message, the Call-ID and the From tag together could be a strong enough, unique CDR identifier. It should be noted that, based on our experiences in network-monitoring, Call-ID alone is also a strong key. Nevertheless, From and To tag parameters grant a safe and global CDR collection technique.

Furthermore, including endpoint IP addresses in the CDR identifier key is very beneficial; mostly for indexing purposes. Telecommunication operators often analyze the IMS domain's traffic on given signaling links, and they are interested in statistics between network nodes. That is the reason why IP addresses should be included (as the second part) in CDR identifier keys. Beside the Dialog-ID triplet, IP addresses support clear association between calls and network links.

Using this general key format, the CDR assembling algorithm should support the From and To field exchange procedure. If the UAS and UAC change the roles during the SIP dialog, the From and To fields could be exchanged, as well. Figure 6 depicts an Invite-dialog, in which the From and To field values change their place in the INFO Request-Response messages. The From tag of the 7th message is equal to the To tag of the previous messages. This suggests that field parameters should be handled with certain dynamic placing within the CDR keys.

4.2 Complex practical cases and considerations

4.2.1 Issues with CDR closing reasons

During the CDR Assembling procedure, opening a new record is clearly based on the initial message of a transaction or a dialog (i.e. initial-Invite message triggers a new CDR in the case of a SIP call). In contrast, it is a complex task to recognize the record-closing message. According to the RFC papers [12,26], transactions and dialogs have different closing methods.

The INVITE dialogs complete, if one of the following trigger event occurs:

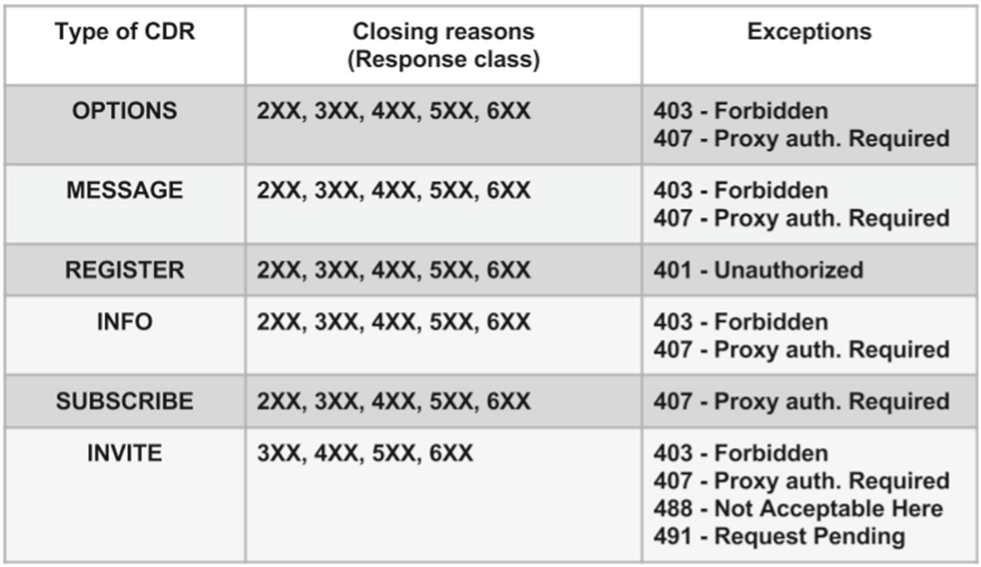
- "487 Request Terminated" message responds to the initial-Invite (triggered by CANCEL transaction),
- "408 Request Timeout" or "481 Call/Transaction does not exist" message responds to Re-Invite message,
- BYE transaction, if the BYE message contains valid From and To tag parameters,
- 3XX/4XX/5XX/6XX class type Response messages within the Invite-dialog.

The SUSCRIBE dialogs complete, if one of the following event occurs:

- The Expires parameter arrives, or the value of the Expires field is 0,
- The Response message to the Subscribe request arrives with 401, 405, 410, 480–485, 489, 501 or 604 status code,
- UAS does not receive periodic Subscribe messages within the "Expires" interval.

First of all, an IMS node does not initiate the closing of the Invite-dialog after sending a BYE message, until the "200 OK" response-message does not return. UAs often send periodic BYE (e.g. in 1, 2 or 4 s), until a response or a timeout event.

RFC papers [26,28] often refer to the 4XX Response messages as transaction- or dialog-closing messages. Our network monitoring experience reveals that there are excep-



Type of CDR	Closing reasons (Response class)	Exceptions
OPTIONS	2XX, 3XX, 4XX, 5XX, 6XX	403 - Forbidden 407 - Proxy auth. Required
MESSAGE	2XX, 3XX, 4XX, 5XX, 6XX	403 - Forbidden 407 - Proxy auth. Required
REGISTER	2XX, 3XX, 4XX, 5XX, 6XX	401 - Unauthorized
INFO	2XX, 3XX, 4XX, 5XX, 6XX	403 - Forbidden 407 - Proxy auth. Required
SUBSCRIBE	2XX, 3XX, 4XX, 5XX, 6XX	407 - Proxy auth. Required
INVITE	3XX, 4XX, 5XX, 6XX	403 - Forbidden 407 - Proxy auth. Required 488 - Not Acceptable Here 491 - Request Pending

Fig. 7 CDR Closing reasons and exceptions

tions of this, as shown in Fig. 7. Some 4XX responses do not mean the release of a call or registration process. The CDR assembling algorithm should handle these cases, in order to provide a proper overview about the core network.

Based on the original assumption about the status code classes, the "407 Proxy Authentication Required" message could even be a CDR closing event. Nevertheless, after examining precisely the new request after a 407 Response message, we found that the UAC does not generate new Call-ID and From tag parameter for the retransmitted initial-Invite. If the rule for creating (opening) a new CDR is the arrival of an initial-Invite message, the retransmitted request-response pair of initial-Invites also open a new CDR. This is not the desired behavior from CDR analysis point of view. We can overcome this issue by associating these messages with the Dialog-ID and using extra information about the 407 responses. In this way the retransmitted messages can be collected into the same CDR as the original request - (407) response pair.

Similarly, "491 Request Pending" also indicates a failed process, but does not mean the end of a SIP dialog. After receiving 491 Response messages, the UAC retransmits the Request message with the same Dialog-ID parameters.

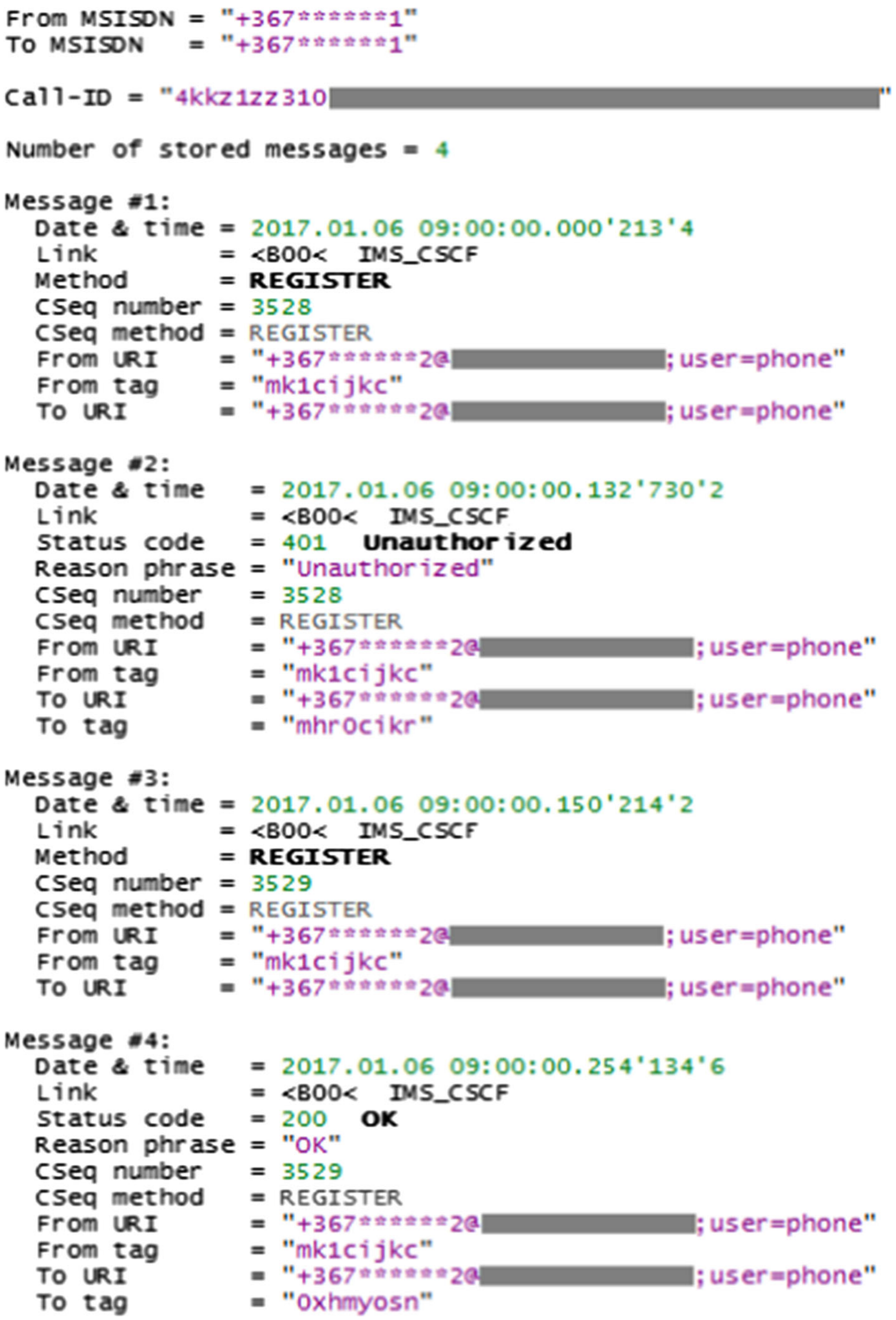
Furthermore, the "403 Forbidden" response has 4XX class type status code—but does not complete a dialog in reality, either. The code 403 merely means that the UAS received the previous request, but it refuses to process it. As part of our network monitoring analysis we found that during an Invite-dialog, the 403 response message does not end the call. It is because this failure code is related only to one request message, and not the whole dialog. However, in REGISTER transactions, after a 403 Response message the UAC can decide to finish the Register procedure, if it seems to be a periodic failure.

Last, but not least, the "401 Unauthorized" Response message falls into this exceptional category, as well. This message appears in REGISTER transactions, as an authentication challenge. Although the code itself belongs to an error-class, this is a normal operation when trying to establish a call. The 401 response message contains the RAND parameter for the user authentication process, and triggers a new Register request with the same Call-ID and From tag parameter. When adding the Register transactions into CDRs, the CDR assembling algorithm can handle the 401 status code as a normal part of the authentication procedure, and does not close the CDR. This is the correct behavior, since the upcoming request-response pair for the same call contains the same key parameters. Instead of two CDRs, the four messages could belong to the same CDR. An exaple of this is depicted by Fig. 8.

This idea has a drawback, too. We have experienced the following network behavior, when the registration process fails: the UAC sends periodic Register messages, and receives 401 responses. Considering this 401 Response as a non-closing event, the CDR could contain more than four messages, and the statistics do not represent this as network failure.

4.2.2 Scenario 2: Call recognition

Associating transactions and dialogs by taking their link-ID into account is a basic CDR-generation technique. It is based on the assumption that the communication between two network nodes are direct (through one link). In practice, this is not a proper assumption; although provides the



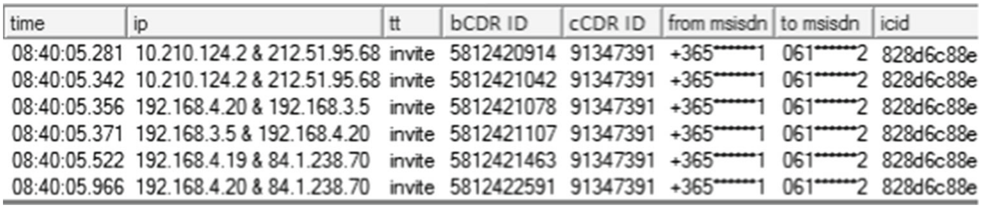
```
From MSISDN = "+367*******1"
To MSISDN   = "+367*******1"

Call-ID = "4kkz1zz310[redacted]"

Number of stored messages = 4

Message #1:
  Date & time = 2017.01.06 09:00:00.000'213'4
  Link        = <B00<  IMS_CSCF
  Method      = REGISTER
  CSeq number = 3528
  CSeq method = REGISTER
  From URI    = "+367*******2@[redacted];user=phone"
  From tag    = "mk1cijkc"
  To URI      = "+367*******2@[redacted];user=phone"

Message #2:
  Date & time   = 2017.01.06 09:00:00.132'730'2
  Link          = <B00<  IMS_CSCF
  Status code   = 401  Unauthorized
  Reason phrase = "Unauthorized"
  CSeq number   = 3528
  CSeq method   = REGISTER
  From URI      = "+367*******2@[redacted];user=phone"
  From tag      = "mk1cijkc"
  To URI        = "+367*******2@[redacted];user=phone"
  To tag        = "mhr0cikr"

Message #3:
  Date & time = 2017.01.06 09:00:00.150'214'2
  Link        = <B00<  IMS_CSCF
  Method      = REGISTER
  CSeq number = 3529
  CSeq method = REGISTER
  From URI    = "+367*******2@[redacted];user=phone"
  From tag    = "mk1cijkc"
  To URI      = "+367*******2@[redacted];user=phone"

Message #4:
  Date & time   = 2017.01.06 09:00:00.254'134'6
  Link          = <B00<  IMS_CSCF
  Status code   = 200  OK
  Reason phrase = "OK"
  CSeq number   = 3529
  CSeq method   = REGISTER
  From URI      = "+367*******2@[redacted];user=phone"
  From tag      = "mk1cijkc"
  To URI        = "+367*******2@[redacted];user=phone"
  To tag        = "0xhmyosn"
```

Fig. 8 Example content of a REGISTER-type CDR: associating many Register messages into one CDR

expected result in many cases. In reality the IMS domain has many functional nodes, and the SIP messages of a dialog are routed within the domain in many times. Furthermore, special nodes (e.g. AS, IN SDP (Intelligent network Service Data Point)) often regenerate some dialog-related parameters (e.g. Call-ID, From tag and To tag parameters). These nodes start a new SIP dialog, which continues the original call setup process. In order to properly identify a complete call setup procedure—and monitor the routing within a domain or between IMS domains—there are extra information needed beside the basic CDR keys (Dialog-ID and IP Addresses).

Let bCDR be the basic, signaling link based CDR, and let cCDR be the concatenated bCDRs, representing the whole SIP call. The cCDR includes all CDRs from the various links, over which the call control messages traversed.

In order to concatenate bCDRs into one cCDR, we need some higher level information, which allows to join the independent call-segments properly. A good candidate for this is the ICID (IMS Charging ID) parameter—which is also used by the IMS to follow the user activity. This is a unique value in the Invite-dialog messages, and it is used for gathering the billing information. The ICID value is located in



time	ip	tt	bCDR ID	cCDR ID	from msisdn	to msisdn	icid
08:40:05.281	10.210.124.2 & 212.51.95.68	invite	5812420914	91347391	+365*******1	061*******2	828d6c88e
08:40:05.342	10.210.124.2 & 212.51.95.68	invite	5812421042	91347391	+365*******1	061*******2	828d6c88e
08:40:05.356	192.168.4.20 & 192.168.3.5	invite	5812421078	91347391	+365*******1	061*******2	828d6c88e
08:40:05.371	192.168.3.5 & 192.168.4.20	invite	5812421107	91347391	+365*******1	061*******2	828d6c88e
08:40:05.522	192.168.4.19 & 84.1.238.70	invite	5812421463	91347391	+365*******1	061*******2	828d6c88e
08:40:05.966	192.168.4.20 & 84.1.238.70	invite	5812422591	91347391	+365*******1	061*******2	828d6c88e

Fig. 9 Example of a concatenated INVITE CDR from basic CDRs

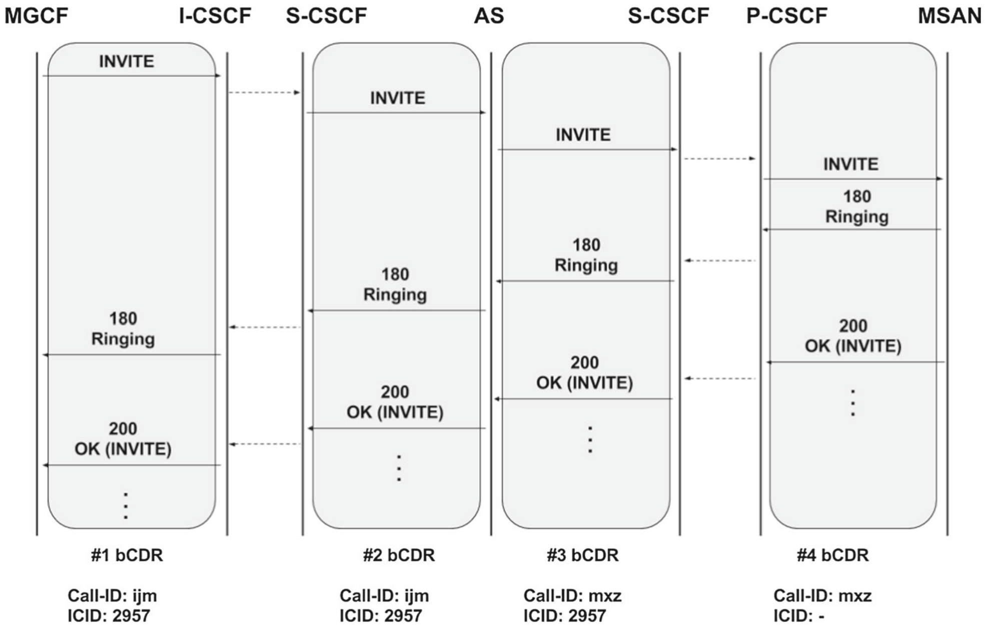


Fig. 10 ICID-, and Call-ID-based call recognition in IMS. Note, that the two S-CSCFs are the same equipment, but the bCDR's have different Call-IDs

the P-Charging-Vector field, as a parameter. Not every dialog message contains it, but it is mandatory in the initial-Invite request message. However, ICID appears only within the IMS domain, over the trusted IMS links. This means that this value is hidden outside the trusted domain by the edge node.

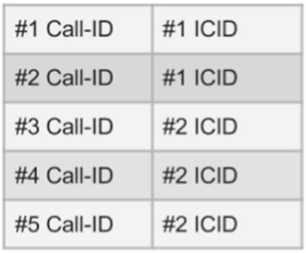
Since the Dialog-ID is same as in the trusted links, a well-maintained Dialog-ID—ICID database provides great support for cCDR generation. An example of such a concatenated CDR is depicted by Fig. 9.

Figure 10 represents a call setup procedure, segmented into bCDRs.

Figure 11 depicts an example for a very simple Call-ID - ICID database, which can be used to concatenate the basic bCDRs into cCDR. For the dialogs not containing any ICID—but have the same Call-ID as another dialog with ICID: we can group those, and assume the same ICID for all.

Let us demonstrate how this pairing works with the help of the Call-ID - ICID database—using the example of Fig. 11. By following the bCDRs shown in Fig. 10, we can see how they belong together.

Let #1 Call-ID be the key of the #1 and #2 bCDR, and #2 Call-ID be the key of the #3 and #4 bCDR. Note, that the ICID value is empty in #4 bCDR. Since #3 bCDR contains an ICID and a Call-ID as well, we can associate this ICID to #4 bCDR too, since they have the same Call-ID. This procedure can also be applied for #1 and #2 bCDRs.



#1 Call-ID	#1 ICID
#2 Call-ID	#1 ICID
#3 Call-ID	#2 ICID
#4 Call-ID	#2 ICID
#5 Call-ID	#2 ICID

Fig. 11 Basic Call-ID - ICID database for Call-CDR collection

4.2.3 *Scenario 3: CDR collection based on I-CSCF behaviors*

From the network operator's point of view, a Call Data Record shall contain the complete call, from the setup to the release procedure. There is a practical issue though, which

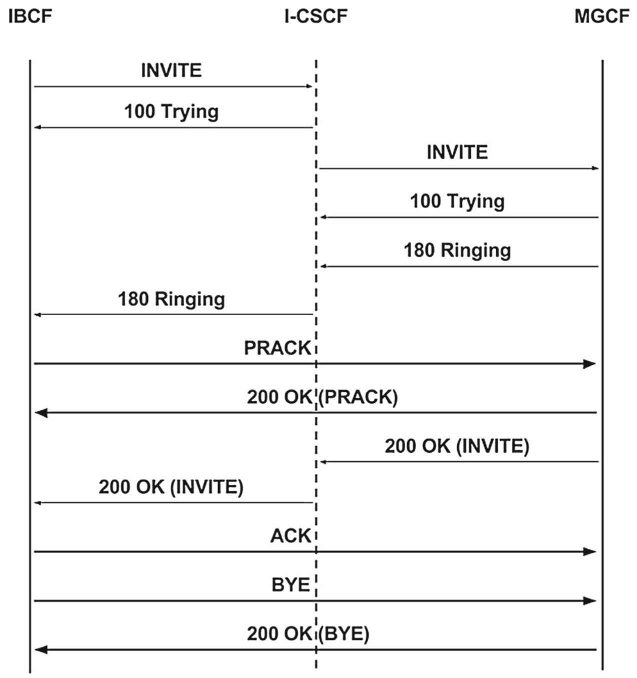


Fig. 12 I-CSCF operation after receiving a final response message

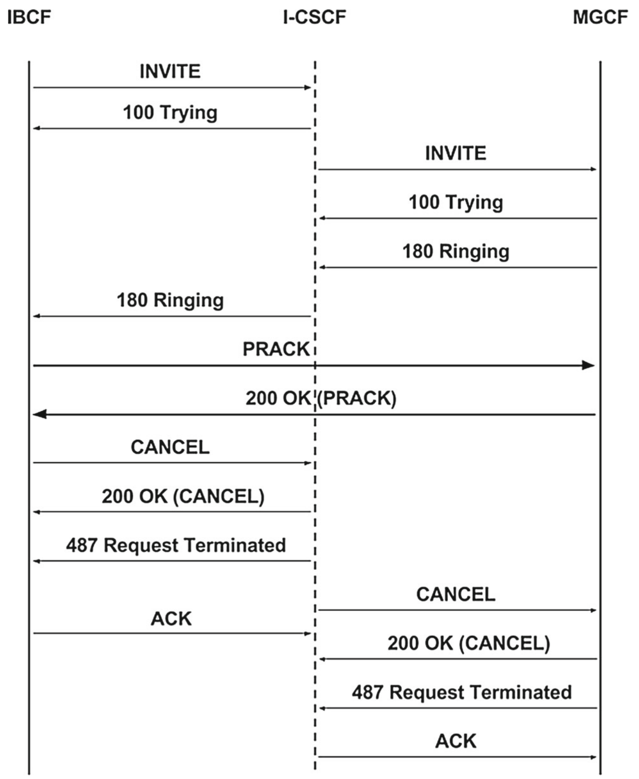


Fig. 13 I-CSCF message forwarding in a case of CANCEL Request. Note, that I-CSCF disappears from the call-chain, when 200 OK arrives

is not usually visible in the logical network architecture diagrams: an IMS domain is usually realized by more than one node—and more than one IP address—in the routing path. On the other hand, using "the" IP address in the CDR keys raises a question: "do we see the end of the dialog always at the same link, as the initial message"? To answer this question, we have to examine the functions of the IMS nodes. The problem space can be reduced to routing: how does the IMS look for a called user before eventually finds it?

The entry points to the IMS domain are the P-CSCF and I-CSCF modules. P-CSCF always forwards the incoming messages to the S-CSCF node. The link between P and S are often non-monitorable, if they reside within physically the same equipment. The I-part has a different task: it allows the communication setup between other domains. Unfortunately, after the UAC receives the final response to the initial-Invite message, the I-CSCF leaves the routing path, hence the edge nodes communicate directly with each other. The result of this property is that the IP addresses of the dialog's release messages are not the same as the IP addresses of the initial message.

Figure 12 depicts a typical IBCF - I-CSCF - MGCF communication flow. The initial-Invite message gets forwarded from the IBCF to the I-CSCF, but the call-completion messages (e.g. BYE, 200 OK (BYE)) are directly sent between IBCF - MGCF, using the same Call-ID and tag parameters, as the first message of the dialog.

When the final response to the initial-Invite message arrives with a failure code (e.g. a 487 Response, triggered by a Cancel request), it raises an other issue: collecting the special, provisional messages of the dialog. Using network monitoring it can be shown that those provisional messages (e.g. PRACK, ACK) are routed between IBCF and MGCF, so these are bypassing I-CSCF as well. Figure 13 depicts a Cancel procedure, in which the call release messages are routed at the same path as the initial message, except those special, provisional messages.

5 Conclusive suggestions for CDR assembling

5.1 About the proportion of exceptional call situations

In order to handle all sorts of exceptions, extra logic has to be added to the CDR assembling mechanism. A complex logic, which handles various (sometimes very peculiar) exceptions, does have a significant drawback. Experts must put in greater and greater efforts into figuring out solutions for those exceptions that have smaller and smaller probability to occur. While this gives a technically perfect result for the CDR assembler, it may not be economical, given the impact of the network monitoring system and its overall purpose. For the monitoring system, it is the operator's management decision whether further exceptions are worth covering, when e.g. over 99% of the cases is handled properly.

Based on a 5-day long monitoring period at a major Hungarian network operator, when over 60 million INVITE CDRs and over 530 million REGISTER transactions were captured, we got the following, rough proportions of exceptional cases.

- 20% of the final responses in the INVITE CDRs were “407 Proxy Authentication Required”;
- $< 0.8\%$ of the answers to Invite messages were “403 Forbidden”, $< 0.1\%$ were “488 Not Acceptable Here”, and $< 0.02\%$ were “491 Request Pending” responses;
- 10% of the bCDRs would have handled incorrectly without taking into account the behavior of I-SCSF described in Scenario 3;
- 1:6 is the proportion of the cases when the caller releases the call before it gets fully established (Cancel);
- 1:2 is the ratio of unsuccessful (4XX) and successful (2XX) response codes (see the distribution below);
- 1:4 is the ratio of unsuccessful and successful response codes when not counting the 487 response code (which should be accounted as normal for cancelled calls);
- 2:5 is the proportion of registrations requiring authentication out of all the registration requests.

The distribution of INVITE dialog response codes in this period were the following:

- 1XX: 66.0%,
- 2XX: 21.2%,
- 3XX: 0.13%,
- 4XX: 12.0%,
- 5XX: 0.45%,
- 6XX: 0.20%.

This 5-day long measurement showed that indeed, not covering the “exceptional” cases within CDR generation would significantly mislead the operators when analyzing the CDRs without such detailed knowledge of the complex and exceptional cases described in this paper.

5.2 Summarized requirements based on the call scenarios

The starting point of the CDR assembling algorithm is the dialog recognition process of the IMS nodes: Dialog-ID-based identification. Since the tag parameter of the To field is not fixed until the final response message, it could be skipped from the CDR keyset. Network monitoring results show that without the To tag parameter, the {Call-ID, From tag} pair is also unique, and the extension with IP addresses gives a strong identification key.

CDR closing reasons provide useful information for the operators while analyzing the CDRs. This information should be as precise as possible, otherwise it would be misleading. Hence, in the above paragraphs we proposed to use some extra logic that supports defining the CDR closing reasons more precisely. The monitoring system should take into account the above mentioned special features and exception-handling mechanisms of the IMS systems and the SIP protocol, otherwise operators would get false status and statistical data about their VoLTE service.

The record closing algorithm has to use extra logic for the CDR closing events for providing proper statistics. It is clear that not every 4XX-type response releases a Dialog or a Transaction, and the algorithm should handle these exceptions (e.g. 403-Forbidden, 408-Not Acceptable Here, 491-Request Pending). In a case of REGISTER-type records, the authentication procedure (e.g. 401-Unauthorized) could be also concatenated into the same CDR. It is important to distinguish cases of rejects during registration, and cases suggesting network error: this helps in proper classification of 4XX-type responses.

In order to show a higher level abstraction of the SIP calls, the link (IP address) related CDRs can be collected into one call record, using billing information as the global user-activity identifier (ICID). A Call-ID - ICID database is a strong assignment for higher level CDRs. In case the CDRs related to the same call would not get connected, statistics about current or overall number of call can be falsely interpreted, too.

The algorithm should support the I-CSCF behavior, and handle the half dialogs, or even just a PRACK transaction in a case of a CANCEL procedure. Using Call-ID - ICID database, the opening and closing of the insufficient records can be summarized into the normal-operation statistics, instead of indicating a failure event.

Although the above described scenarios are seemingly easy-to-cover by software, the distributed manner of user-related identifiers makes them harder to handle.

5.3 Collecting user-related identifiers: MSISDN, IMSI, IMPI

There are three important user-related identifiers that support the operator’s work during IMS administration tasks. These are

1. MSISDN—in short: the called/calling party number;
2. IMSI—the SIM card’s identifier, and
3. IMPI—a global, private identifier within IMS—which could also be a concatenation of MSISDN, IMSI, fixed equipment port numbers, and others.

IMS is using a text-based protocol for call setup (SIP), for which the order of the fields and parameters can be varied within a message. To collect user related information, the CDR-assembling algorithm has to search in many fields and parameters. To recognize the caller, the From field could be the starting point. However, in some cases the From field hides the user, and any user identifiers (such as MSISDN, IMSI or IMPI) can be included in the Contact, P-Asserted-Identity, P-Preferred-Identity or in Remote-Party-ID field, in which the format is also variable.

Recognizing the callee in a SIP dialog could be a quite complex task. First of all, in most cases the called-MSISDN is located in the Request line, To field or in P-Called-Party-ID field. Beside these fields, the called phone number can be sent digit-by-digit in an INFO transaction, within the INVITE-dialog. Call forwarding is also applicable in IMS: the History-Info field implies this event.

An Invite message often includes an SDP or XML layer. SDP is for the media session parameters, and it contains codec-related information. In contrast, the XML part may contain caller and callee MSISDNs (or IMSI or IMPI) and even ICID values. Furthermore, while ICID is a unique identifier within an IMS domain, it is possible to capture different calls with same ICIDs, when the border between IMS domains is monitored. To cover these scenarios as well, the globally unique IOI (Inter Operator Identifier) can be stored beside the ICID.

In some cases, a bCDR does not contain the MSISDN numbers, because the IMS hides the user information (e.g. anonymous caller). Using the ICID - Call-ID database, the missing user information could be derived from other bCDRs.

6 Conclusion

Although its first commercial deployments happened in 2014, Voice over LTE is still not yet a mature service, after 3 years. The majority of the LTE operators have not yet launched their service; the number of global deployments is just around one hundred. There is a reason behind this delay: it is hard to meet the expected quality of service when the infrastructure and the service logic reached such a high complexity.

Network and service monitoring is one of the key tools for providing feedback about quality. Regarding calls, monitoring-based CDR assembly is crucial for the operators in order to be able to trace what is working well—and what is not—with the VoLTE service. The current paper presents the challenges of CDR generation for the IMS, and provides a complete solution for these challenges. It shows what messages and parameters can define the CDRs and how, it suggests what interfaces to monitor, and shows what are the key parameters to use for cross-correlation.

Starting from simple use cases, and detailing more complex ones, this paper presents complete methods for SIP CDR assembly that works in current LTE scenarios. Furthermore, this paper shows methods to handle unusual call-situations, and—taking into account the complex and the unusual cases—it suggests further improvements to CDR generation.

One of the suggestions is to compose basic (link-level) CDRs into more general CDRs, to allow the operator to easily see the bigger picture, e.g. calls together with registrations. Another suggestion is that the monitoring system should be equipped with methods of handling and cross-correlating key parameters (IMSI, MSISDN, IMPI, ICID) on-the-fly. Such an up-to-date database with multi-parameter indices can also help in building further, monitoring-based services, as well.

Taking into account the methods and corrective actions for CDR-assembling lead to recognizing the really unusual call-situations and even to recognizing downgraded quality within the VoLTE service. If the specialties of the IMS architecture and the SIP protocol are handled by the monitoring system, its measurement result can be considered as a reference. Any significant deviation from that reference would indicate potential problems. When the operator trusts the result of the monitoring system, its personnel is kept interested in clearing out all the really erroneous situations that such a system points out.

References

1. 3GPP: Application of the Base Station System Application Part (BSSAP) on the E interface. TS 49.008, 3rd Generation Partnership Project (3GPP) (2015). http://www.3gpp.org/DynaReport/49008.htm.
2. 3GPP: IP Multimedia Subsystem (IMS). TS 23.228, 3rd Generation Partnership Project (3GPP) (2016). http://www.3gpp.org/DynaReport/23228.htm.
3. 3GPP: Media Gateway Control Function (MGCF) - IM Media Gateway (2016). http://www.3gpp.org/DynaReport/29332.htm.
4. AITIA: SGA-NETMON The GSM/GPRS/UMTS/LTE Network Monitoring System. White paper (2012). http://sga.aitia.ai/pdfs/SGA-NetMon.pdf.
5. Balcerzak, J., Tyszka, G., & Senecal, S. (2014). QoS monitoring model of registration procedure for IMS platform. In *16th international telecommunications network strategy and planning symposium*.
6. Breda, G. D., & de Mendes, L. S. (2006). QoS monitoring and failure detection. In *International telecommunications symposium*.
7. Camarillo, G., Holmberg, C., & Gao, Y. (2011). *Re-INVITE and target-refresh request handling in the session initiation protocol (SIP)*. RFC 6141.
8. CISCO: Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update. 2015–2020 white paper (2016). http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.
9. Donovan, S. (2000). *The SIP INFO method*. RFC 2976.

10. Handley, M., Jacobson, V., & Perkins, C. (2006). *SDP: Session description protocol*. RFC 4566.
11. Hoffstadt, D., Monhof, S. & Rathgeb, E. (2012). SIP trace recorder: Monitor and analysis tool for threats in SIP-based networks. In *8th international wireless communications and mobile computing conference*.
12. Holmberg, C. (2011). *Number portability parameters for the "tel" URI*. RFC 6228.
13. ITU-T: Specifications of Signalling System No. 7 - ISDN user part. Recommendation Q.763, International Telecommunication Union (1999).
14. Hyun, J., Li, J., Im, C. T., Yoo, J.-H., & Hong, J. W.-K. (2015). A high performance VoLTE traffic classification method using HTCondor. In *IFIP/IEEE International symposium on integrated network management (IM)*
15. Hyun, J., Li, J., Im, C. T., Yoo, J.-H., & Hong, J. W.-K. (2014). A VoLTE traffic classification method in LTE network. In *16th Asia-Pacific network operations and management symposium*.
16. Kozma, D., Soos, G., & Varga, P. (2016). Supporting LTE network and service management through session data record analysis. *Infocommunications Journal*, *71*(2), 11–16.
17. Li, C. Y., Tu, G. H., Peng, C., Yuan, Z., Li, Y., Lu, S., & Wang, X. (2015). Insecurity of voice solution VoLTE in LTE mobile networks. In *22nd ACM SIGSAC conference on computer and communications security*.
18. Lutiis, P. D., & Lombardo, D. (2009). An innovative way to analyze large ISP data for IMS security and monitoring. In *13th international conference on intelligence in next generation networks*.
19. Nassar, M., State, R., & Festor, O. (2010). A framework for monitoring SIP enterprise networks. In *4th international conference on network and system security*.
20. Olsson, M., Sultana, S., Rommer, S., Frid, L., & Mulligan, C. (2009). *SAE and the evolved packet core*. Oxford: Academic Press.
21. Poikselka, M., & Mayer, G. (2009). *The IMS: IP multimedia concepts and services*. Chichester: Wiley.
22. Raouyane, B., Bellafkih, M., Errais, M., Leghroudi, D., Ranc, D., & Ramdani, M. (2011). *eTOM business processes conception in NGN monitoring* (pp. 133–143). Berlin: Springer.
23. Roach, A. B. (2002). *Session initiation protocol (SIP)-specific event notification*. RFC 3265.
24. Rosenberg, J. (2002). *The session initiation protocol (SIP) UPDATE method*. RFC 3311.
25. Rosenberg, J., & Schulzrinne, H. (2002). *Session initiation protocol (SIP): Locating SIP servers*. RFC 3263.
26. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., et al. (2002). *SIP: Session initiation protocol*. RFC 3261.
27. Tatai, P., Marosi, G., & Osvath, L. (2001). A flexible approach to mobile telephone traffic mass measurement and analysis. In *18th IEEE instrumentation and measurement technology conference*.
28. Worley, D., Huelsemann, M., Jesske, R., & Alexeitsev, D. (2013). *Completion of calls for the session initiation protocol (SIP)*. RFC 6910.
29. Zhang, L., Zhang, Z., Cui, X. & Liu, D. (2013). Research on the monitoring and controlling model of SIP network. In *IEEE international conference on green computing and communications and IEEE internet of things and IEEE cyber, physical and social computing*.

Tamás Tóthfalusi is currently a Ph.D. student at the Budapest University of Technology and Economics (BME). He received the B.Sc. degree in Infocommunication Networks as part of the degree programme in Engineering Information Technology at the University of Debrecen in 2010, and the M.Sc. degree in Hardware Programming at the University of Debrecen in 2012. His research interests include hardware (FPGA) based acceleration of network processes, network management and measurement, VoLTE signaling and neural networks. He is a member of the SmartCom Lab at BME. He has been involved in industrial research and development projects in these topics.

Pál Varga is Associate Professor at the Department of Telecommunications and Media Informatics, BME, Hungary, where he got his M.Sc. (1997) and Ph.D. (2011) degrees. Besides, he is director in AITIA International Inc. Earlier he was working for Ericsson Hungary and Tecnomen Ireland, as software design engineer and system architect, respectively. His main research interest include communication systems, network monitoring, network performance measurements, root cause analysis, fault localisation, traffic classification, end to end QoS and SLA issues, as well as hardware acceleration, and Internet of Things. He has been involved in various industrial as well as European research and development projects in these topics.