OpenSIPS 2.3
From SIP-I Trunks to End Users

Răzvan Crainea

- 3 May 2017 -

AMSTERDAM
OPENSIPS
SUMMIT 2017

Outline

- Introduction
- OpenSIPS and SIP-I
- Examples
- Conclusions

Introduction

PSTN

- Public Switched Telephone Networks
- Aggregates all the circuit-switched telephone networks
- Based on Signalling System No. 7 (SS7)
    - Developed in 1975
    - **Call establishment and teardown**
    - Number translations and portability
    - Messaging (SMS)
    - Billing

ISUP

- ISDN User Protocol
  - Subsystem of SS7 used to establish telephone calls in PSTN
- Developed by the ITU-T group
- ISUP is a binary protocol

SIP and ISUP compatibility

Type of message	ISUP	SIP
Initiate call	IAM (Initial address)	INVITE
Call ringing	ACM (Address complete)	180/183 Ringing
Call answer	ANM (Answer message)	200 OK (for INVITE)
Terminate call	REL (Release)	BYE
Terminate Complete	RLC (Release complete)	200 OK (for BYE)

SIP-T

- Developed by IETF
  - RFC3372, RFC2976, RFC3204 and RFC3398
- Supported calls:
  - PSTN-PSTN over SIP
  - PSTN-SIP
  - SIP-PSTN calls
- Defines encapsulation and mappings
- Focuses on the interworking of basic calls
- Does not address extra services

SIP-I

- Developed by ITU-T
- TRQ.2815
  - ISUP and SIP
- Q.1912.5
  - 3GPPSIP and ISUP
  - SIP and ISUP
  - SIP-I and ISUP
- Focuses on interworking of basic calls
- Full support for ISUP supplementary services

Architecture

SIP-I = ISUP messages enveloped in SIP packages

Why SIP-I…

- … and not just plain SIP?
- SIP-I provides extra information that might/should not be part of the final SIP message
  - Ex: Caller ID, billing information
  - Caller/callee profile
- Standardizes the format this information is passed from one side to the other
  - Q.763 recommendations

OpenSIPS and SIP-I

Issues

- ISUP is binary
  - Fields cannot be manipulated with plain text operations
- ISUP message is attached to the SIP body
  - If SDP is also present, we need support for multiple SDP bodies
- ISUP protocol is quite complex
  - Various message types
  - Each type has its own mandatory parameters
  - Parameters have limited types and values
  - Their values are binary encoded

SIP-I Module

- Provides functions to parse ISUP binary messages
- Exports variables to read/modify/delete ISUP parameters
- Exports script functions to add ISUP body
- Defines default values for new ISUP message
    - Considers message type
    - Comply to the ITU-T Q.763 Requirement

Proxy Mode

- Relay SIP-I messages between SIP-I switches
- Use the ISUP information to route the message
- Update SIP headers based on ISUP indications
- Modify the ISUP body
    - Add/remove ISUP params
    - Modify params values

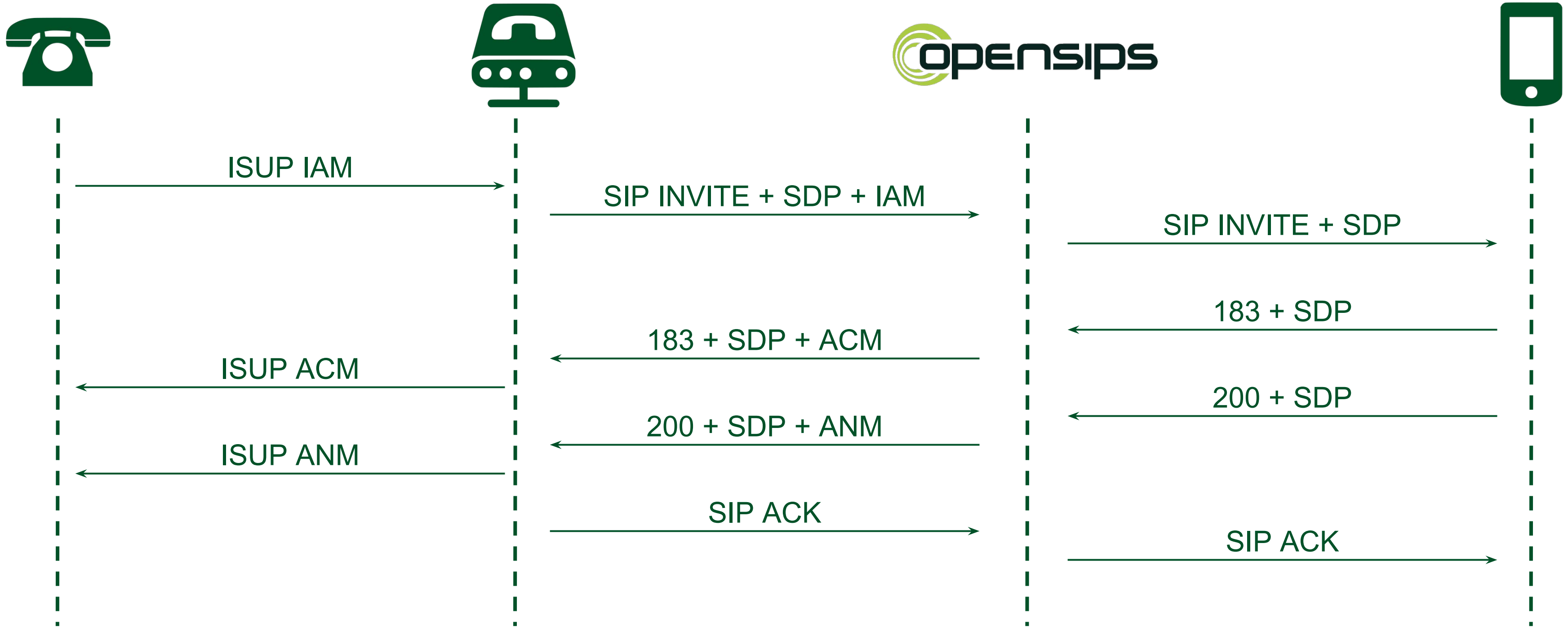
Gateway Mode

- SIP-I to SIP
    - Inspect ISUP message
    - Update SIP message according (Ex. change CID)
    - Drop ISUP payload

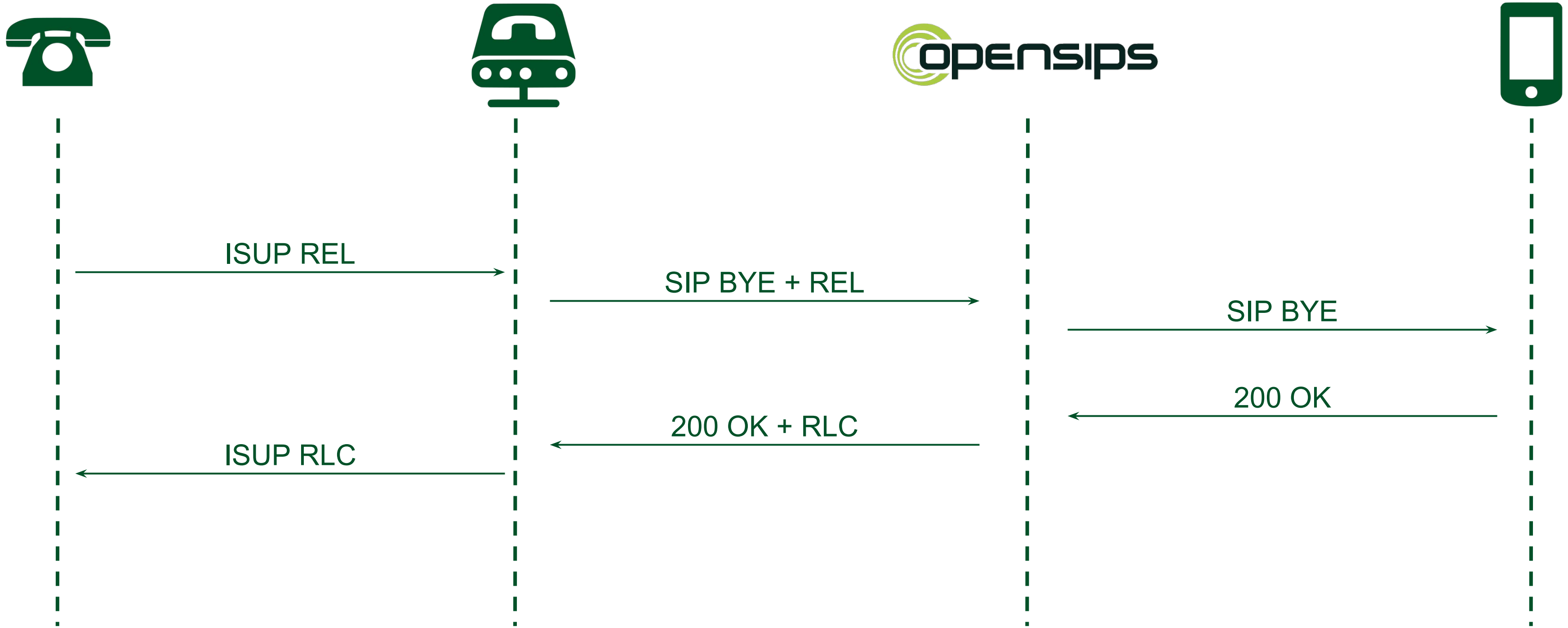
- SIP-I to SIP
    - Add ISUP body
    - Modify body according to the SIP message

Examples

SIP-I to SIP call establishment



SIP-I to SIP call termination



SIP-I Message

```
▸ Request-Line: INVITE sip:1234@[redacted]:5060 SIP/2.0
▾ Message Header
    Max-Forwards: 19
  ▸ P-Asserted-Identity: <sip:1234@[redacted]:5060;user=phone>
  ▸ Via: SIP/2.0/UDP [redacted]:5080;rport;branch=z9hG4bK1056725933
  ▸ From: <sip:1000@[redacted]:5080>;tag=665027175
  ▸ To: <sip:1234@[redacted]:5060>
    Call-ID: 1621534724@[redacted]:5080
  ▸ CSeq: 7 INVITE
    User-Agent: [redacted]
  ▸ Contact: <sip:1000@[redacted]:5080>
    Allow: ACK, INVITE, BYE, CANCEL, REGISTER, REFER, OPTIONS, INFO
    Content-Type: multipart/mixed;boundary=342386194_2648357551
    Content-Length: 440
▾ Message Body
  ▾ MIME Multipart Media Encapsulation, Type: multipart/mixed, Boundary: "342386194_2648357551"
      [Type: multipart/mixed]
      Preamble: 0d0a
      First boundary: --342386194_2648357551\r\n
    ▾ Encapsulated multipart part:  (application/sdp)
        Content-Type: application/sdp\r\n\r\n
      ▾ Session Description Protocol
          Session Description Protocol Version (v): 0
        ▸ Owner/Creator, Session Id (o): yate 1480701213 1480701213 IN IP4 [redacted]
          Session Name (s): SIP Call
        ▸ Connection Information (c): IN IP4 [redacted]
        ▸ Time Description, active time (t): 0 0
        ▸ Media Description, name and address (m): audio 30042 RTP/AVP 8 0 101
        ▸ Media Attribute (a): rtpmap:8 PCMA/8000
        ▸ Media Attribute (a): rtpmap:0 PCMU/8000
        ▸ Media Attribute (a): rtpmap:101 telephone-event/8000
      Boundary: \r\n--342386194_2648357551\r\n
    ▾ Encapsulated multipart part:  (application/isup)
        Content-Type: application/isup;version=itu-t92+\r\n
        Content-Disposition: signal;handling=optional\r\n\r\n
      ▾ ISDN User Part
          Message Type: Initial address (1)
        ▸ Nature of Connection Indicators: 0x0
        ▸ Forward Call Indicators: 0x6001
        ▸ Calling Party's category: 0xa (ordinary calling subscriber)
        ▸ Transmission medium requirement: 0 (speech)
        ▸ Called Party Number: 1234
          Pointer to start of optional part: 6
        ▸ Calling Party Number: 1000
          End of optional parameters (0)
      Last boundary: \r\n--342386194_2648357551--\r\n
```

ISUP Parameters Manipulation

```
# set the Numbering plan
$isup_param(Called Party Number | Numbering plan indicator) = 1;

# or set it using aliases
$isup_param(Called Party Number | Numbering plan indicator) = “ISDN”;

# check the value written
xlog(“Called Party Indicator: $isup_param(Called Party Number|Numbering plan
indicator)\n”);
# prints “Called Party Indicator: 1”

# check the expanded value
xlog(“Called Party Indicator: $isup_param_str(Called Party Number|Numbering plan
indicator)\n”);
# prints “Called Party Indicator: ISDN”
```

OpenSIPS Configuration - Initial Requests

```
if (has_totag() && is_method("INVITE")) {
     if (has_body("application/isup")) {
          xlog("Called number: $isup_param(Called party number)\n");
          remove_body_part("application/isup");
     } else {
          add_isup_part("Initial address");
          $isup_param(Called party number|Nature of address indicator) = 3;
          $isup_param(Called party number|Numbering plan indicator) = 1;
          $isup_param(Called party number|Address signal) = $rU;
          $isup_param(Calling party number|Nature of address indicator) = 3;
          $isup_param(Calling party number|Numbering plan indicator) = 1;
          $isup_param(Calling party number|Screening indicator) = 3;
          $isup_param(Calling party number|Address signal) = $fU;
     }
 }
```

OpenSIPS Configuration - Sequentials

```
if (has_totag() && loose_route()) {
    if (is_method(“BYE”)) {
        if (has_body(“application/isup”)) {
            xlog(“Called number: $isup_param(Called party number)\n”);
            remove_body_part("application/isup");
        } else {
            add_isup_part("Release");
            $isup_param(Cause indicators|Location) = 10;
            $isup_param(Cause indicators|Cause value) = 16
        }
    }
    t_relay();
}
```

Conclusions

Conclusions

- OpenSIPS SIP-I module parses binary ISUP messages
- Provides an easy and flexible way to add/remove ISUP body
- Facilitates ISUP message build
- Simple and easy to use interface
- Works both as a proxy and full SIP-I gateway

Take-Away Message

Starting with the new OpenSIPS 2.3 integrating PSTN trunks with has never been easier!

- Răzvan Crainea
  - OpenSIPS Project: www.opensips.org
  - Email: razvan@opensips.org