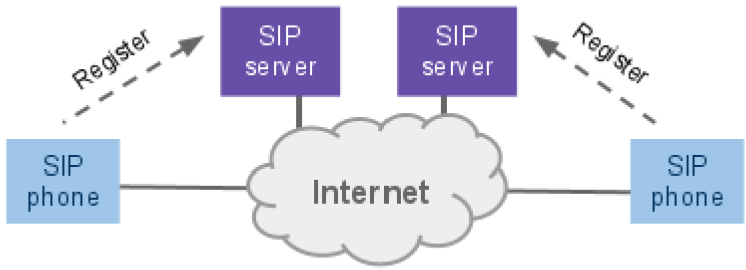
Android SIP Project Documentation

Group:

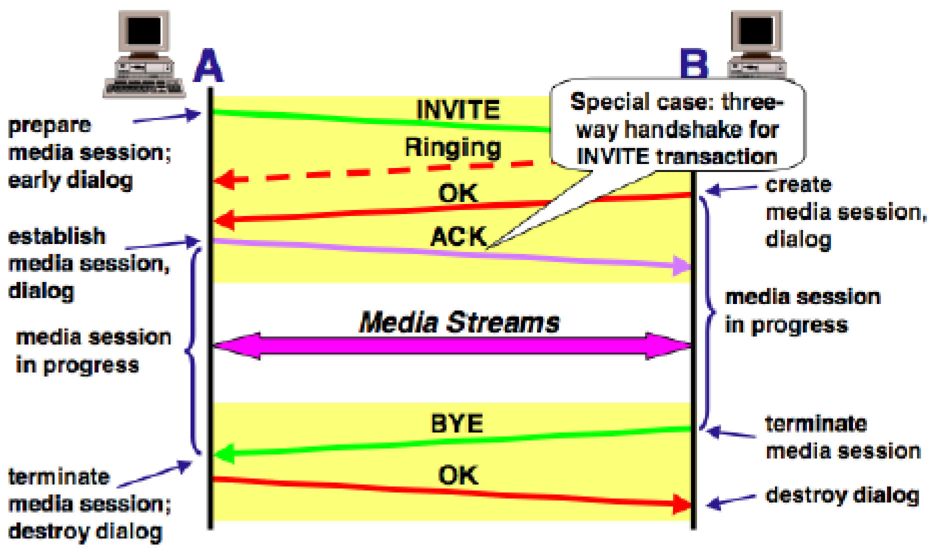
- Mamczarz Mateusz
- Łukasz Rojda
- Jakub Wełpa

Sip Architecture:



SIP

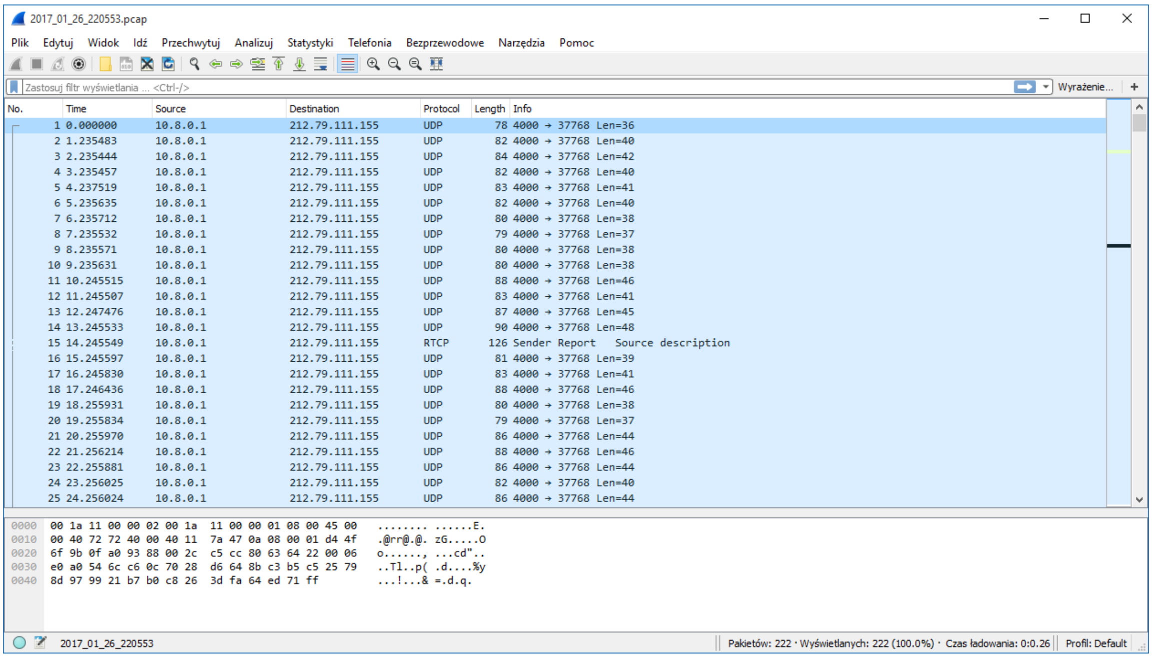
http://www.sipsocial.net/_/rsrc/1472867188161/sip-technology/sip-architecture/Centralized%20SIP.PNG



A basic SIP Session

Capture from Wireshark

USER A



2017_01_26_220553.pcap

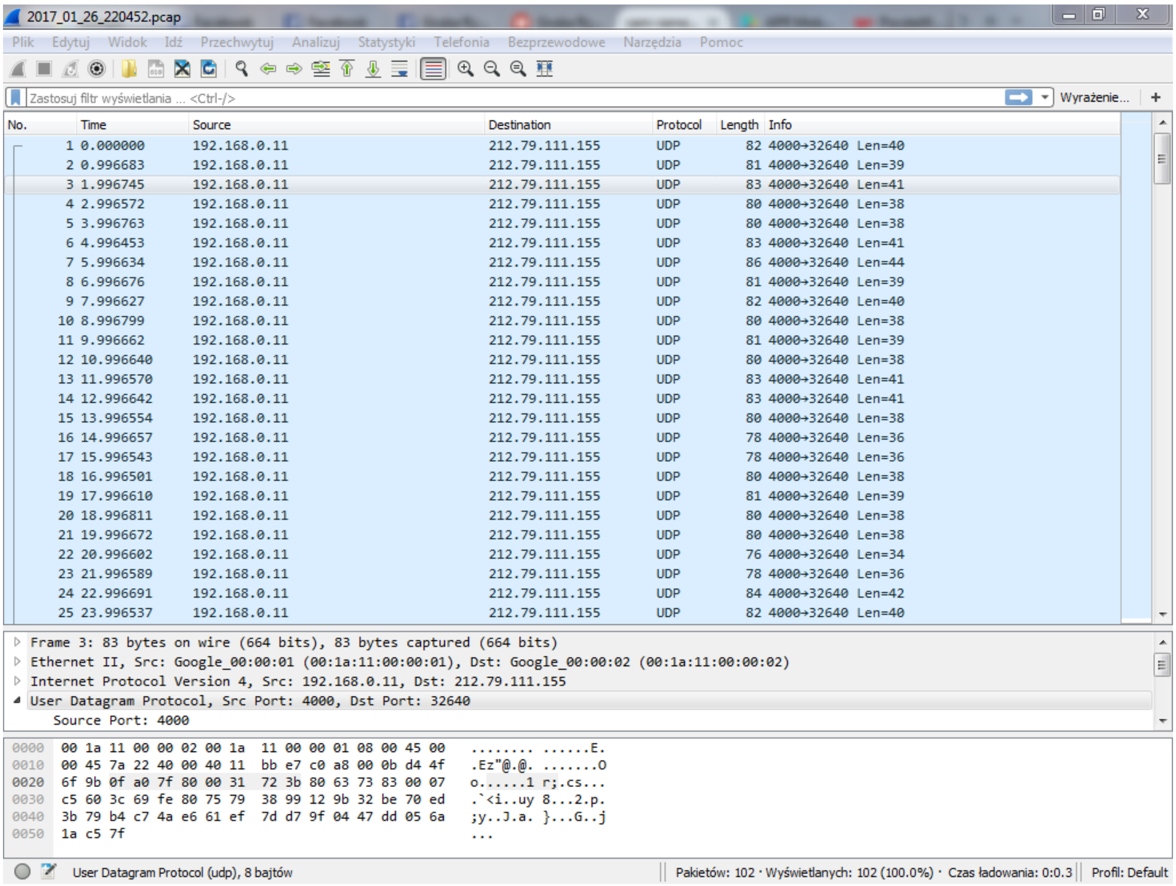
Plik Edytuj Widok Idź Przechwytuj Analizuj Statystyki Telefonia Bezprzewodowe Narzędzia Pomoc

Zastosuj filtr wyświetlania ... <Ctrl-/> Wyrażenie...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.8.0.1	212.79.111.155	UDP	78	4000 → 37768 Len=36
2	1.235483	10.8.0.1	212.79.111.155	UDP	82	4000 → 37768 Len=40
3	2.235444	10.8.0.1	212.79.111.155	UDP	84	4000 → 37768 Len=42
4	3.235457	10.8.0.1	212.79.111.155	UDP	82	4000 → 37768 Len=40
5	4.237519	10.8.0.1	212.79.111.155	UDP	83	4000 → 37768 Len=41
6	5.235635	10.8.0.1	212.79.111.155	UDP	82	4000 → 37768 Len=40
7	6.235712	10.8.0.1	212.79.111.155	UDP	80	4000 → 37768 Len=38
8	7.235532	10.8.0.1	212.79.111.155	UDP	79	4000 → 37768 Len=37
9	8.235571	10.8.0.1	212.79.111.155	UDP	80	4000 → 37768 Len=38
10	9.235631	10.8.0.1	212.79.111.155	UDP	80	4000 → 37768 Len=38
11	10.245515	10.8.0.1	212.79.111.155	UDP	88	4000 → 37768 Len=46
12	11.245507	10.8.0.1	212.79.111.155	UDP	83	4000 → 37768 Len=41
13	12.247476	10.8.0.1	212.79.111.155	UDP	87	4000 → 37768 Len=45
14	13.245533	10.8.0.1	212.79.111.155	UDP	90	4000 → 37768 Len=48
15	14.245549	10.8.0.1	212.79.111.155	RTCP	126	Sender Report Source description
16	15.245597	10.8.0.1	212.79.111.155	UDP	81	4000 → 37768 Len=39
17	16.245830	10.8.0.1	212.79.111.155	UDP	83	4000 → 37768 Len=41
18	17.246436	10.8.0.1	212.79.111.155	UDP	88	4000 → 37768 Len=46
19	18.255931	10.8.0.1	212.79.111.155	UDP	80	4000 → 37768 Len=38
20	19.255834	10.8.0.1	212.79.111.155	UDP	79	4000 → 37768 Len=37
21	20.255970	10.8.0.1	212.79.111.155	UDP	86	4000 → 37768 Len=44
22	21.256214	10.8.0.1	212.79.111.155	UDP	88	4000 → 37768 Len=46
23	22.255881	10.8.0.1	212.79.111.155	UDP	86	4000 → 37768 Len=44
24	23.256025	10.8.0.1	212.79.111.155	UDP	82	4000 → 37768 Len=40
25	24.256024	10.8.0.1	212.79.111.155	UDP	86	4000 → 37768 Len=44

```
0000  00 1a 11 00 00 02 00 1a  11 00 00 01 08 00 45 00   ........ ......E.
0010  00 40 72 72 40 00 40 11  7a 47 0a 08 00 01 d4 4f   .@rr@.@. zG.....O
0020  6f 9b 0f a0 93 88 00 2c  c5 cc 80 63 64 22 00 06   o......, ...cd"..
0030  e0 a0 54 6c c6 0c 70 28  d6 64 8b c3 b5 c5 25 79   ..Tl..p( .d....%y
0040  8d 97 99 21 b7 b0 c8 26  3d fa 64 ed 71 ff         ...!...& =.d.q.
```

2017_01_26_220553 | Pakietów: 222 · Wyświetlanych: 222 (100.0%) · Czas ładowania: 0:0.26 | Profil: Default



2017_01_26_220452.pcap

Plik Edytuj Widok Idź Przechwytuj Analizuj Statystyki Telefonia Bezprzewodowe Narzędzia Pomoc

Zastosuj filtr wyświetlania ... <Ctrl-/> Wyrażenie...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.11	212.79.111.155	UDP	82	4000→32640 Len=40
2	0.996683	192.168.0.11	212.79.111.155	UDP	81	4000→32640 Len=39
3	1.996745	192.168.0.11	212.79.111.155	UDP	83	4000→32640 Len=41
4	2.996572	192.168.0.11	212.79.111.155	UDP	80	4000→32640 Len=38
5	3.996763	192.168.0.11	212.79.111.155	UDP	80	4000→32640 Len=38
6	4.996453	192.168.0.11	212.79.111.155	UDP	83	4000→32640 Len=41
7	5.996634	192.168.0.11	212.79.111.155	UDP	86	4000→32640 Len=44
8	6.996676	192.168.0.11	212.79.111.155	UDP	81	4000→32640 Len=39
9	7.996627	192.168.0.11	212.79.111.155	UDP	82	4000→32640 Len=40
10	8.996799	192.168.0.11	212.79.111.155	UDP	80	4000→32640 Len=38
11	9.996662	192.168.0.11	212.79.111.155	UDP	81	4000→32640 Len=39
12	10.996640	192.168.0.11	212.79.111.155	UDP	80	4000→32640 Len=38
13	11.996570	192.168.0.11	212.79.111.155	UDP	83	4000→32640 Len=41
14	12.996642	192.168.0.11	212.79.111.155	UDP	83	4000→32640 Len=41
15	13.996554	192.168.0.11	212.79.111.155	UDP	80	4000→32640 Len=38
16	14.996657	192.168.0.11	212.79.111.155	UDP	78	4000→32640 Len=36
17	15.996543	192.168.0.11	212.79.111.155	UDP	78	4000→32640 Len=36
18	16.996501	192.168.0.11	212.79.111.155	UDP	80	4000→32640 Len=38
19	17.996610	192.168.0.11	212.79.111.155	UDP	81	4000→32640 Len=39
20	18.996811	192.168.0.11	212.79.111.155	UDP	80	4000→32640 Len=38
21	19.996672	192.168.0.11	212.79.111.155	UDP	80	4000→32640 Len=38
22	20.996602	192.168.0.11	212.79.111.155	UDP	76	4000→32640 Len=34
23	21.996589	192.168.0.11	212.79.111.155	UDP	78	4000→32640 Len=36
24	22.996691	192.168.0.11	212.79.111.155	UDP	84	4000→32640 Len=42
25	23.996537	192.168.0.11	212.79.111.155	UDP	82	4000→32640 Len=40

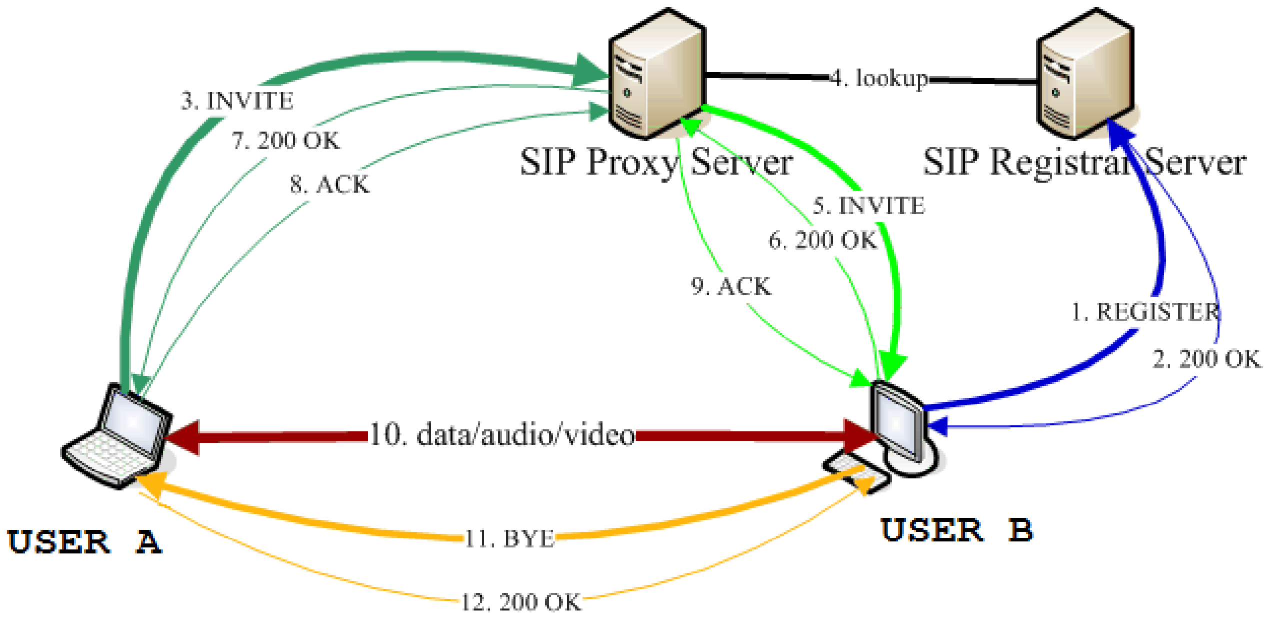
```
Frame 3: 83 bytes on wire (664 bits), 83 bytes captured (664 bits)
Ethernet II, Src: Google_00:00:01 (00:1a:11:00:00:01), Dst: Google_00:00:02 (00:1a:11:00:00:02)
Internet Protocol Version 4, Src: 192.168.0.11, Dst: 212.79.111.155
User Datagram Protocol, Src Port: 4000, Dst Port: 32640
    Source Port: 4000
```

```
0000  00 1a 11 00 00 02 00 1a  11 00 00 01 08 00 45 00   ........ ......E.
0010  00 45 7a 22 40 00 40 11  bb e7 c0 a8 00 0b d4 4f   .Ez"@.@. .......O
0020  6f 9b 0f a0 7f 80 00 31  72 3b 80 63 73 83 00 07   o......1 r;.cs...
0030  c5 60 3c 69 fe 80 75 79  38 99 12 9b 32 be 70 ed   .`<i..uy 8...2.p.
0040  3b 79 b4 c7 4a e6 61 ef  7d d7 9f 04 47 dd 05 6a   ;y..J.a. }...G..j
0050  1a c5 7f                                           ...
```

User Datagram Protocol (udp), 8 bajtów | Pakietów: 102 · Wyświetlanych: 102 (100.0%) · Czas ładowania: 0:0.3 | Profil: Default

Sequence diagram of the basic session

SIP-based services are SIP servers that offer services, such as message routing, to SIP endpoints, such as IP phones. The SIP registrar server and proxy server offer SIP registration and proxy services to help the SIP endpoints locate and communicate with each other.



1. User B registers itself to the registrar server by sending a REGISTER request.

2. The registrar server accepts the registration, which contains the User B's name address, by responding with a 200 OK status code.

3. User A requests to establish a communication session with User B by sending an INVITE request to the proxy server. The INVITE message's content typically contains the description of the communication session the User A wants to establish, such as media type, security, or IP address. The description is typically in Session Description Protocol (SDP) format.

4. The proxy server looks up the registrar server to find out the User B's current address. Note that lookup is an implementation issue not part of SIP.

5. The proxy server forwards the INVITE request from User A to User B based on its current address.
6. User B accepts the invitation by responding with a 200 OK status code. The 200 OK response to an INVITE request typically contains the description of the communication session that User B can establish with the User A.

7. The proxy server forwards a 200 OK response from User B to User A.

8. User A confirms the session establishment by sending an ACK message to the proxy server. The ACK message may contain the final agreement on the session.

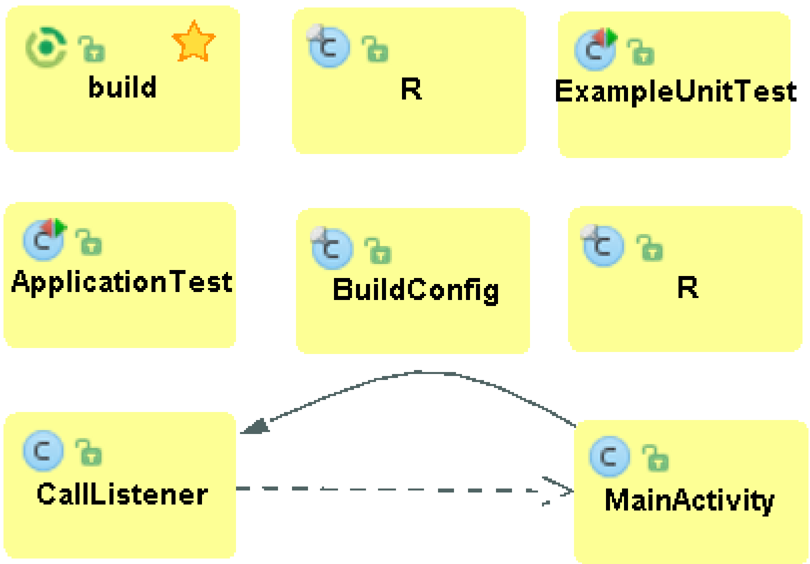
9. In turn, the proxy server forwards the ACK to the User B. Thus, the three-way handshake is completed via the proxy server, and a session is established.

10. Now the communication between User A and User B happens. The protocol used for communication may or may not be SIP. For example, instant messages can be transmitted over SIP. Voice conversations are typically transmitted over RTP.

11. Now, User B finishes the conversation and wishes to terminate the session by sending a BYE request.

12. User A responds with a 200 OK status code to accept session termination.

App Architecture



Final Functionalities:

- Create User friendly interface
- Possibility to connect any sip Address
- Possibility to call another sip User
- Possibility to send message to another sip User

Intermediary Steps

- implement the basic UI
- implement Connecting to the server
- implement Basic registration
- implement connection between devices
- implement session between devices
- implement end session between devices
- implement possibility to transmit data between devices

Library:

Application is using the basic Session Initial Protocol API which is provided by standard Android SDK.

Documentation for the API : https://developer.android.com/guide/topics/connectivity/sip.html

SIP API Classes and Interfaces

Here is a summary of the classes and one interface (`SipRegistrationListener`) that are included in the Android SIP API:

Class/Interface	Description
`SipAudioCall`	Handles an Internet audio call over SIP.
`SipAudioCall.Listener`	Listener for events relating to a SIP call, such as when a call is being received ("on ringing") or a call is outgoing ("on calling").
`SipErrorCode`	Defines error codes returned during SIP actions.
`SipManager`	Provides APIs for SIP tasks, such as initiating SIP connections, and provides access to related SIP services.
`SipProfile`	Defines a SIP profile, including a SIP account, domain and server

Class/Interface	Description
	information.
SipProfile.Builder	Helper class for creating a SipProfile.
SipSession	Represents a SIP session that is associated with a SIP dialog or a standalone transaction not within a dialog.
SipSession.Listener	Listener for events relating to a SIP session, such as when a session is being registered ("on registering") or a call is outgoing ("on calling").
SipSession.State	Defines SIP session states, such as "registering", "outgoing call", and "in call".
SipRegistrationListener	An interface that is a listener for SIP registration events.

Requirements

- You must have a mobile device that is running Android 2.3 or higher.
- SIP runs over a wireless data connection, so your device must have a data connection (with a mobile data service or Wi-Fi). This means that you can't test on AVD—you can only test on a physical device.
- Each participant in the application's communication session must have a SIP account. There are many different SIP providers that offer SIP accounts