

---

# **Praktikum Rechnernetze**

Protokoll zu Versuch 10 (VoIP) von Gruppe 1

Jakob Waibel, Daniel Hiller, Elia Wüstner, Felix Pojtinger

2021-12-21

## Inhaltsverzeichnis

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Einführung</b>                          | <b>3</b>  |
| 1.1      | Mitwirken . . . . .                        | 3         |
| 1.2      | Lizenz . . . . .                           | 3         |
| <b>2</b> | <b>STUN und Registrierung</b>              | <b>4</b>  |
| <b>3</b> | <b>Verbindungsaufbau und SDP-Protokoll</b> | <b>8</b>  |
| <b>4</b> | <b>RTP/RTCP</b>                            | <b>13</b> |
| <b>5</b> | <b>SIP-Byte</b>                            | <b>16</b> |

# 1 Einführung

## 1.1 Mitwirken

Diese Materialien basieren auf [Professor Kiefers “Praktikum Rechnernetze”-Vorlesung der HdM Stuttgart](#).

**Sie haben einen Fehler gefunden oder haben einen Verbesserungsvorschlag?** Bitte eröffnen Sie ein Issue auf GitHub ([github.com/pojntfx/uni-netpractice-notes](https://github.com/pojntfx/uni-netpractice-notes)):



**Abbildung 1:** QR-Code zum Quelltext auf GitHub

Wenn Ihnen die Materialien gefallen, würden wir uns über einen GitHub-Stern sehr freuen.

## 1.2 Lizenz

Dieses Dokument und der enthaltene Quelltext ist freie Kultur bzw. freie Software.



**Abbildung 2:** Badge der AGPL-3.0-Lizenz

Uni Network Practice Notes (c) 2021 Jakob Waibel, Daniel Hiller, Elia Wüstner, Felix Pojtinger

SPDX-License-Identifier: AGPL-3.0

## 2 STUN und Registrierung

**Bei der Konfiguration des sipgate-Accounts sind auch Angaben zum sogenannten STUN-Server erforderlich. Beschreiben Sie mit eigenen Worten Aufgaben und die Funktion eines STUN-Servers**

Die "Session Traversal Utilities for NAT" ist ein Standard, welcher dabei hilft, die öffentlichen Netzwerkadressen von Netzwerknodes herauszufinden, um eine Peer-to-peer Verbindung zwischen diesen Nodes hinter NAT herzustellen. Es kann außerdem festgestellt werden, welche Art von NAT verwendet wird (Full Cone, Restricted Cone...). Da die Belastung von STUN-Servern in der Realität vergleichsweise niedrig ist, können für viele Projekte öffentliche STUN-Server ausreichend sein.

**Welche IP-Adresse hat das REGISTER-Paket nach dem NAT-Vorgang (NAT ist wegen der privaten Adresse erforderlich)?**

Wie dem unteren Screenshot entnommen werden kann, hat das REGISTER-Paket nach dem NAT-Vorgang die IP-Adresse 194.49.221.7.

|             |                |                |         |   |
|-------------|----------------|----------------|---------|---|
| 1 0.000000  | 10.231.172.221 | 217.10.79.9    | SIP     | 710 Request: REGISTER sip:sipgate.de (1 binding)        |
| 2 0.010972  | 217.10.79.9    | 10.231.172.221 | SIP     | 521 Status: 401 Unauthorized                            |
| 3 0.011573  | 10.231.172.221 | 217.10.79.9    | SIP     | 904 Request: REGISTER sip:sipgate.de (1 binding)        |
| 4 0.021907  | 217.10.79.9    | 10.231.172.221 | SIP     | 585 Status: 200 OK (REGISTER) (1 binding)               |
| 5 31.298931 | 217.10.79.9    | 10.231.172.221 | SIP/SDP | 1359 Request: INVITE sip:2555428e0@10.231.172.221:49699 |

```

Request-Line: REGISTER sip:sipgate.de SIP/2.0
Message Header
  Via: SIP/2.0/UDP 194.49.221.7:22556;branch=z9hG4bK8041d10841bbe8118a33a1d115dda7c5;rport
  From: "PhonerLite" <sip:2555428e0@sipgate.de>;tag=3247336616
    SIP from display info: "PhonerLite"
    SIP from address: sip:2555428e0@sipgate.de
    SIP from tag: 3247336616
  To: "PhonerLite" <sip:2555428e0@sipgate.de>
    SIP to display info: "PhonerLite"
    SIP to address: sip:2555428e0@sipgate.de
  Call-ID: 8041D108-41BB-E811-8A30-A1D115DDA7C5@194.49.221.7
  [Generated Call-ID: 8041D108-41BB-E811-8A30-A1D115DDA7C5@194.49.221.7]
  CSeq: 1 REGISTER
  Contact: <sip:2555428e0@194.49.221.7:22556>;+sip.instance="urn:uuid:006D68F3-4421-E811-8759-B7021FA9EAB5"
    Contact URI: sip:2555428e0@194.49.221.7:22556
      Contact URI User Part: 2555428e0
      Contact URI Host Part: 194.49.221.7
      Contact URI Host Port: 22556
    Contact parameter: +sip.instance="urn:uuid:006D68F3-4421-E811-8759-B7021FA9EAB5"
  Allow: INVITE, ACK, BYE, CANCEL, INFO, MESSAGE, NOTIFY, OPTIONS, REFER, UPDATE, PRACK
  Max-Forwards: 70
  Allow-Events: org.3gpp.nwinitdereg
  User-Agent: SIPPER for PhonerLite
  Supported: replaces, from-change, gruu
  Expires: 900
  Content-Length: 0
  
```

**Abbildung 3:** Capture des Register-Pakets vor NAT

| No. | Time      | Source         | Destination    | Protocol | Length | Info   |
|-----|-----------|----------------|----------------|----------|--------|--|
| 1   | 0.000000  | 10.231.172.221 | 217.10.79.9    | SIP      | 710    | Request: REGISTER sip:sipgate.de (1 binding)       |
| 2   | 0.010972  | 217.10.79.9    | 10.231.172.221 | SIP      | 521    | Status: 401 Unauthorized                           |
| 3   | 0.011573  | 10.231.172.221 | 217.10.79.9    | SIP      | 904    | Request: REGISTER sip:sipgate.de (1 binding)       |
| 4   | 0.021907  | 217.10.79.9    | 10.231.172.221 | SIP      | 585    | Status: 200 OK (REGISTER) (1 binding)              |
| 5   | 31.298931 | 217.10.79.9    | 10.231.172.221 | SIP/SDP  | 1359   | Request: INVITE sip:2555428e0@10.231.172.221:49699 |

▶ Frame 3: 904 bytes on wire (7232 bits), 904 bytes captured (7232 bits) on interface \Device\NPF\_{60B079C4-4A54-4F24-9289-5222073BB27F}, in  
 ▶ Ethernet II, Src: Mettler-\_c8:d2:7f (00:e0:7c:c8:d2:7f), Dst: Tp-LinkT\_5e:ff:c6 (60:e3:27:5e:ff:c6)  
 ▶ Internet Protocol Version 4, Src: 10.231.172.221, Dst: 217.10.79.9  
 ▶ User Datagram Protocol, Src Port: 49699, Dst Port: 5060  
 ▼ Session Initiation Protocol (REGISTER)

▶ Request-Line: REGISTER sip:sipgate.de SIP/2.0  
 ▼ Message Header

▶ Via: SIP/2.0/UDP 10.231.172.221:49699;branch=z9hG4bK8041d10841bbe8118a34a1d115dda7c5;rport  
 ▶ From: "PhonerLite" <sip:2555428e0@sipgate.de>;tag=3247336616  
 ▶ To: "PhonerLite" <sip:2555428e0@sipgate.de>  
 Call-ID: 8041D108-41BB-E811-8A30-A1D115DDA7C5@10.231.172.221  
 [Generated Call-ID: 8041D108-41BB-E811-8A30-A1D115DDA7C5@10.231.172.221]  
 ▶ CSeq: 2 REGISTER  
 ▼ Contact: <sip:2555428e0@10.231.172.221:49699>;+sip.instance="urn:uuid:006D68F3-4421-E811-8759-B7021FA9EAB5">  
 ▼ Contact URI: sip:2555428e0@10.231.172.221:49699  
 Contact URI User Part: 2555428e0  
 Contact URI Host Part: 10.231.172.221  
 Contact URI Host Port: 49699  
 Contact parameter: +sip.instance="urn:uuid:006D68F3-4421-E811-8759-B7021FA9EAB5">\r\n  
 ▶ Authorization: Digest username="2555428e0", realm="sipgate.de", nonce="W60YGlujlu4RRkLKBpUQTDAFdxgDCaUq", uri="sip:sipgate.de", respo  
 Allow: INVITE, ACK, BYE, CANCEL, INFO, MESSAGE, NOTIFY, OPTIONS, REFER, UPDATE, PRACK  
 Max-Forwards: 70  
 Allow-Events: org.3gpp.nwinitdereg  
 User-Agent: SIPPER for PhonerLite  
 Supported: replaces, from-change, gruu  
 Expires: 900  
 Content-Length: 0

Abbildung 4: Capture des Register-Pakets nach NAT

### Erstellen und dokumentieren Sie den „FlowGraph“ des vorliegenden Pakets und erläutern Sie kurz den prinzipiellen Ablauf.

Im ersten Schritt registriert sich der Client beim Server in Form eines **Register**. Bei diesem teilt der Client dem Server seine Standort-Informationen mit. Wenn dies erfolgt ist, kann durch den **Invite** ein Anruf initialisiert werden. Es wird eine Verbindung zur Gegenseite hergestellt und anschließend auf deren Reaktion gewartet. Wenn die Gegenseite mit einem **Ack** reagiert, bestätigt sie uns die Verbindung. Nun kann die Kommunikation über einen RTP-Stream erfolgen. Um die Trennung der Verbindung zu realisieren, kann einer der beiden Teilnehmer ein **Bye** senden.

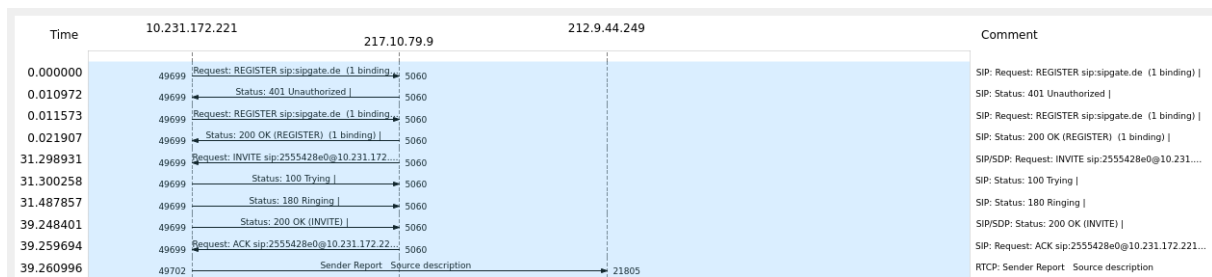


Abbildung 5: Verbindungsaufbau mit SIP



Abbildung 6: Kommunikation mit RTP



Abbildung 7: Verbindungsabbau mit SIP

## Nach diesem typischen Ablauf ist der UAC beim Provider registriert. Warum wird die Anfrage zur Registrierung zunächst abgewiesen?

Die Credentials des UAC werden mit Hilfe der in der Rejection (401 Unauthorized) vorhandenen Daten verschlüsselt. Sobald die Credentials verschlüsselt wurden, können sie an das SIP-Gateway geschickt werden.

## Worin unterscheiden sich die beiden REGISTER-Pakete?



## Abbildung 8: Vergleich beider SIP-Pakete (Contact & Authorization)

Dem linken (ersten) Paket fehlt die Authorisierung, das zweite besitzt diese im Realm `sipgate.de`

## Warum wird für die so wichtige Registrierung nicht TCP (garantiert die bitgetreue Zustellung) verwendet, sondern UDP?

Für die Registrierung wird UDP statt TCP verwendet, da das Session Initiation Protocol (SIP) selbst eine spezifizierte Nachrichtenabfolge hat und verbindungsorientiert ist. Das ermöglicht, dass das SIP fehlerhafte Protokollabläufe selbst feststellt und nicht zwingend TCP zur Fehlerkorrektur benötigt.

## Wie lange ist die Registrierung gültig?

Wie im folgenden Bild zu sehen ist, beträgt der "Expires" Wert 900 Sekunden. Dies entspricht 15 Minuten.

```

Session Initiation Protocol (REGISTER)
  Request-Line: REGISTER sip:sipgate.de SIP/2.0
  Message Header
    Via: SIP/2.0/UDP 10.231.172.221:49699;branch=z9hG4bK8041d10841bbe8118a34a1d115dda7c5;rport
    From: "PhonerLite" <sip:2555428e0@sipgate.de>;tag=3247336616
    To: "PhonerLite" <sip:2555428e0@sipgate.de>
    Call-ID: 8041d108-4188-E811-8A30-A1D115DDA7C5@10.231.172.221
    [Generated Call-ID: 8041d108-4188-E811-8A30-A1D115DDA7C5@10.231.172.221]
    CSeq: 2 REGISTER
    Contact: <sip:2555428e0@10.231.172.221:49699>;+sip.instance="urn:uuid:006068F3-4421-E811-8759-B7021FA9EAB5"
    Authorization: Digest username="2555428e0", realm="sipgate.de", nonce="W60YGluJlu4RRkLKBpUQTDAFdxgDCaUq", uri="sip:sipgate.de", response="29f298f70f57c059885a5e29765bed4"
    Allow: INVITE, ACK, BYE, CANCEL, INFO, MESSAGE, NOTIFY, OPTIONS, REFER, UPDATE, PRACK
    Max-Forwards: 70
    Allow-Events: org.3gpp.nwinitdereg
    User-Agent: SIPPER for PhonerLite
    Supported: replaces, from-change, gruu
    Expires: 900
    Content-Length: 0

```

**Abbildung 9:** Gültigkeitsdauer der Registrierung (900s)

**Die interne IP-Adresse des UA wird durch NAT in eine offizielle externe IP umgesetzt. Wie lautet die externe IP und zu welchem Unternehmen gehört diese IP?**

Die externe IP des UA lautet 194.49.221.0 und gehört zur “DFS Deutsche Flugsicherung GmbH”.

The screenshot shows the IPinfo.io website interface. At the top, there's a navigation bar with links for Products, Solutions, Why IPinfo?, Pricing, Resources, Docs, Login, and a Sign up button. Below the navigation bar, the main content area displays 'IP range details' for the IP range **194.49.221.0/24**, which is associated with **AS62434 - DFS Deutsche Flugsicherung GmbH**. On the left side, there's a sidebar with a search bar and three menu items: 'WHOIS Details' (highlighted with a blue bar and a right arrow), 'IP Addresses', and 'Hosted Domains'. The main content area on the right shows a 'Summary' table with the following information:

| Summary    |         |
|------------|---------|
| Country    | Germany |
| Domain     | dfs.de  |
| ASN        | AS62434 |
| Registry   | ripe    |
| Hosted IPs | 256     |
| ID         | DFS-NET |

**Abbildung 10:** Lookup-Ergebnisse zur IP (Deutsche Flugsicherung)

### 3 Verbindungsaufbau und SDP-Protokoll

#### Welche SIP\_Methods unterstützt der Anrufer?

Wie im Screenshot zu sehen unterstützt der Anrufer die SIP-Methoden **INVITE**, **ACK**, **CANCEL**, **OPTIONS**, **BYE**, **REFER**, **SUBSCRIBE**, **NOTIFY**, **INFO**, **PUBLISH** und **MESSAGE**.



```
▼ Session Initiation Protocol (INVITE)
  ▶ Request-Line: INVITE sip:2555428e0@10.231.172.221:49699 SIP/2.0
  ▼ Message Header
    ▶ Record-Route: <sip:217.10.79.9;lr;ftag=as1da87d54>
    ▶ Record-Route: <sip:172.20.40.6;lr>
    ▶ Record-Route: <sip:217.10.68.137;lr;ftag=as1da87d54>
    ▶ Via: SIP/2.0/UDP 217.10.79.9;branch=z9hG4bK620d.70720930871bcf1d63f6077496ee77cd.0
    ▶ Via: SIP/2.0/UDP 172.20.40.6;branch=z9hG4bK620d.458c80f8dc48e38afdc31b1c423a13c0.0
    ▶ Via: SIP/2.0/UDP 217.10.68.137;branch=z9hG4bK620d.e61e620768ab8026e3b97ca6f225b04f.0
    ▶ Via: SIP/2.0/UDP 217.10.77.115:5060;branch=z9hG4bK1f25f9bd
    ▶ Max-Forwards: 67
    ▶ From: "anonymous" <sip:anonymous@sipgate.de>;tag=as1da87d54
    ▶ To: <sip:2555428e0@sipgate.de>
    ▶ Contact: <sip:anonymous@217.10.77.115:5060>
    ▶ Call-ID: 5d0eca60468d2182243ab84b059ee901@sipgate.de
    ▶ [Generated Call-ID: 5d0eca60468d2182243ab84b059ee901@sipgate.de]
    ▶ CSeq: 103 INVITE
    ▶ Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE
    ▶ Supported: replaces
    ▶ Content-Type: application/sdp
    ▶ Content-Length: 415
  ▶ Message Body
```

**Abbildung 11:** Erlaubte SIP-Methoden

### Welche Bedeutung haben Trying und Ringing?

Die Response 100 **Trying** bedeutet, dass der next-hop Server die Anfrage erhalten hat und eine un-spezifizierte Handlung vorgenommen wird, um diesen Anruf zu ermöglichen. Nach **Trying** werden die **INVITE** Nachrichten gestoppt.

Die Response 180 **Ringing** bedeutet, dass der UA den **INVITE** erhalten hat und den Nutzer benachrichtigt.

### Welche Angabe bzgl. der Absender-Rufnummer erscheint auf dem Display des Empfängers?

In unserem Fall ist die Absender-Rufnummer “anonymous”, was auf eine versteckte Rufnummer hindeutet.

```

▼ Session Initiation Protocol (INVITE)
  ▶ Request-Line: INVITE sip:2555428e0@10.231.172.221:49699 SIP/2.0
  ▼ Message Header
    ▶ Record-Route: <sip:217.10.79.9;lr;ftag=as1da87d54>
    ▶ Record-Route: <sip:172.20.40.6;lr>
    ▶ Record-Route: <sip:217.10.68.137;lr;ftag=as1da87d54>
    ▶ Via: SIP/2.0/UDP 217.10.79.9;branch=z9hG4bK620d.70720930871bcf1d63f6077496ee77cd.0
    ▶ Via: SIP/2.0/UDP 172.20.40.6;branch=z9hG4bK620d.458c80f8dc48e38afdc31b1c423a13c0.0
    ▶ Via: SIP/2.0/UDP 217.10.68.137;branch=z9hG4bK620d.e61e620768ab8026e3b97ca6f225b04f.0
    ▶ Via: SIP/2.0/UDP 217.10.77.115:5060;branch=z9hG4bK1f25f9bd
    ▶ Max-Forwards: 67
    ▼ From: "anonymous" <sip:anonymous@sipgate.de>;tag=as1da87d54
      SIP from display info: "anonymous"
      ▶ SIP from address: sip:anonymous@sipgate.de
      SIP from tag: as1da87d54
    ▼ To: <sip:2555428e0@sipgate.de>
      ▶ SIP to address: sip:2555428e0@sipgate.de
    ▶ Contact: <sip:anonymous@217.10.77.115:5060>
      Call-ID: 5d0eca60468d2182243ab84b059ee901@sipgate.de
      [Generated Call-ID: 5d0eca60468d2182243ab84b059ee901@sipgate.de]
    ▶ CSeq: 103 INVITE
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE
      Supported: replaces
      Content-Type: application/sdp
      Content-Length: 415
  ▶ Message Body

```

**Abbildung 12:** Display-Info des SIP-Headers (“anonymous”)

**Der sehr lange “branch”-Wert ist eine Zufallszahl und identifiziert eindeutig eine SIP-Vermittlungsinstanz. Berechnen Sie die Wahrscheinlichkeit, dass zwei SIP-Geräte einen identischen Wert erwürfeln (es zählen nur die Angaben zwischen den beiden Punkten).**

Mittels folgendem JavaScript-Code wurden die Anzahl an Möglichkeiten berechnet:

```
1 Math.pow(  
2   16,  
3   "z9hG4bK620d.70720930871bcf1d63f6077496ee77cd.0".split(".")[1].length  
4 );
```

Wir kommen zur folgenden Anzahl an Möglichkeiten:

```
1 3.402823669209385e38;
```

Die Wahrscheinlichkeit, dass zweimal diesselbe Zahl berechnet wird, lässt sich also wie folgt berechnen:

```
1 1 /  
2   Math.pow(  
3     16,  
4     "z9hG4bK620d.70720930871bcf1d63f6077496ee77cd.0".split(".")[1].  
        length  
5   );
```

Wir kommen zur folgenden Wahrscheinlichkeit:

```
1 2.938735877055719e-39;
```

Die Wahrscheinlichkeit für eine Kollision ist, wie zu erwarten sehr klein.

**Beschreiben Sie Aufbau und Inhalt des Session Description Protokoll (SDP), insbesondere die verwendeten Portnummern und das Audio-Video-Profile AVP, das die erlaubten Codecs in einer priorisierten Reihenfolge angibt.**

In SDP werden Eigenschaften von Multimediadatenströmen aufgezeigt. SDP beinhaltet die Sitzungsbeschreibungen (Protokollversion (v), Session-ID (o)), die Zeitbeschreibung (Zeit, in der die Session schon aktiv ist (t),...) und die Medienbeschreibung (Medientyp, Port und Protokoll (m)). Wichtig bei den Medienbeschreibungen sind auch die Medienattribute (a), mit welchen weitere Einstellungen getroffen werden können.

Ein Beispiel des SDP-Teils einer SIP-Nachricht könnte wie folgt aussehen:

```
1 v=0
2 o=Jean 1234 1234 IN IP4 host.hdm.com
3 s=Video von 987654
4 c=IN IP4 host.hdm.com
5 t=0 0
6 m=audio 20000 RTP/AVP 97
7 a=rtpmap:97 iLBC/8000
8 a=fmtp:97 mode=30
9 m=video 20002 RTP/AVP 31
10 a=rtpmap:31 H261/90000
```

Das Audio-Video-Profile (AVP) spezifiziert die Formate für Audio- und Videostreams. Im obenstehenden Beispiel werden mit den IDs 97 und 31 iLBC und H.261 spezifiziert. Es können auch mehrere ID's hintereinander aufgeführt werden, was dann eine Liste mit Priorisierungen darstellt.

Die Portnummern hängen vom jeweiligen Datenstrom ab. Der eine Datenstrom, welcher in diesem Beispiel ein Audiostrom im Format iLBC (ID: 97) darstellt, läuft in diesem Fall auf Port 20000. Der andere Datenstrom, welcher einen Videodatenstrom im Format H.261 (ID:31) darstellt, läuft auf Port 20002. Dies konnte auch jeweils in der Beschreibung der Medientypen (m) in der Medienbeschreibung festgelegt werden.

**Welcher Sprach-Codec wird hier eingesetzt? Wie hoch ist die Bitrate dieses Codecs?**

Wie im folgenden zu sehen wird der Codec G.711 verwendet. Dieser Codec weist eine Bitrate von 64 kbit/s. Diese Bitrate resultiert aus den 8000 samples pro Sekunde mit jeweils 8 Bit.

[illegible]

**Abbildung 13:** Auszug des RTP-Captures (Codec G.711)

## 4 RTP/RTCP

**Dokumentieren Sie den RTP-Kommunikationsfluss anhand der IP-Adressen. Wer kommuniziert mit wem?**

Die beiden Teilnehmer kommunizieren durch den Server miteinander. Zur Veranschaulichung sprechen wir hier von Bob und Alice. Alice versendet ihr Paket an den Server, dieser leitet es dann an Bob weiter. Das gleiche gilt für die Pakete, welche Bob versendet. Dies ist zum Beispiel bei SRTP wichtig, da hierdurch die Streams unabhängig verschlüsselt werden.



Abbildung 14: Flow-Chart des Kommunikationsflusses

## Wieviel „Audio-Samples“ (Abtastproben) enthält ein Ethernet-Paket? In welchen zeitlichen Abständen werden die Pakete gesendet?

In digitaler Telefonie wird üblicherweise mit 8000Hz gearbeitet. Die Samplerate kann dann im Media-Attribut eingestellt werden.

## Welche Ethernet-Paketlänge wird übertragen? Warum fasst man nicht längere oder kürzere Zeiträume zusammen?

Es wird eine Paketlänge von insgesamt 214 Bytes übertragen. Von diesen 214 Bytes, stellen 54 Bytes Header von Ethernet (14 Byte), IPv4 (20 Byte), UDP (8 Byte) und RTP (12 Byte) dar. Somit bleibt eine Nutzlast von 160 Bytes. Längere Zeiträume würden zu höherer Latenz, kleine Zeiträume zu größerem Overhead durch den Header führen.

```

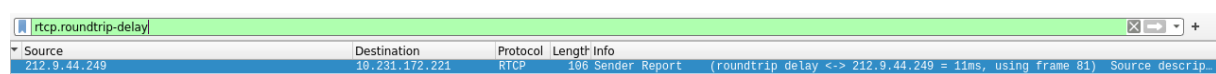
▼ Frame 391: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface \Device\NPF_{60B079C4-4A54-4F24-9289-5222073BB27F}, id 0
  ► Interface id: 0 (\Device\NPF_{60B079C4-4A54-4F24-9289-5222073BB27F})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 20, 2018 14:48:26.957950000 CEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1537447706.957950000 seconds
    [Time delta from previous captured frame: 0.005803000 seconds]
    [Time delta from previous displayed frame: 0.005803000 seconds]
    [Time since reference or first frame: 43.066884000 seconds]
    Frame Number: 391
    Frame Length: 214 bytes (1712 bits)
    Capture Length: 214 bytes (1712 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:rtp]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
  ▼ Ethernet II, Src: Tp-LinkT_5e:ff:c6 (60:e3:27:5e:ff:c6), Dst: Mettler-_c8:d2:7f (00:e0:7c:c8:d2:7f)
    ► Destination: Mettler-_c8:d2:7f (00:e0:7c:c8:d2:7f)
    ► Source: Tp-LinkT_5e:ff:c6 (60:e3:27:5e:ff:c6)
    Type: IPv4 (0x0800)

```

Abbildung 15: Länge des Ethernet-Frames (214 bytes)

## Wie groß ist die Verzögerungszeit über das Verbindungsnetz?

Um unser Ergebnis zu ermitteln, haben wir unter **Protocol Preferences** uns die **relative roundtrip calculation** anzeigen lassen und anschließend den Anzeigefilter **rtcp.roundtrip-delay** angewendet. Das Ergebnis war eine Verzögerungszeit von 11ms.



| Source       | Destination    | Protocol | Length | Info  |
|--------------|----------------|----------|--------|---|
| 212.9.44.249 | 10.231.172.221 | RTCP     | 106    | Sender Report (roundtrip delay <-> 212.9.44.249 = 11ms, using frame 81) Source descrip... |

Abbildung 16: Roundtrip-Delay eines RTP-Pakets, wie es von RTCP dargestellt wird

## Können Sie auch RTCP-Pakete erkennen? Wie häufig werden sie gesendet? Welchem Zweck dienen sie?

Es sind RTCP-Pakete in regelmäßigen Abständen zu finden. In unserem Fall beträgt das Zeitintervall zwischen von einer Node ausgehenden Nachrichten 10 Sekunden. Das ist insofern passend, da das minimale Zeitintervall zwischen RTCP-Paketen 5 Sekunden betragen sollte. RTCP dient dem Zweck Statistiken und Kontrollinformationen über RTP-Sessions bereitzustellen.

**Welche Portnummern werden für die RTP-Verbindung verwendet, welche für die zugehörigen RTCP-Kontrollkanäle (Wireshark: VoipCalls – SIPFlows - FlowSequence)**

RTCP verwendet den Port 49702 bei der einen Node und 21805 bei der anderen. Relativ dazu betragen die RTP Ports 49701 und 21804. Dies ist jeweils der um 1 verringerte RTCP Port.



**Abbildung 17:** Port-Nummern von RTP und RTCP

## 5 SIP-Byte

**Beschreiben Sie, wie der BYE-Method-Timer arbeitet?**

Der Mechanismus verwendet periodische Aktualisierungen, um die Sitzung aktiv zu halten. Dies wird durch re-INVITES oder UPDATES realisiert. Der Mechanismus ist abwärtskompatibel mit SIP, sodass er funktioniert, solange einer der beiden Teilnehmer eines Dialogs, ihn beherrscht. Es werden zwei neue Header-Felder (Session-Expires und Min-SE) und ein neuer Antwortcode (422) definiert. Session-Expires gibt die Dauer der Sitzung an, und Min-SE gibt den minimal zulässigen Wert für den Ablauf der Sitzung an. Der Antwortcode 422 zeigt an, dass die Dauer des Sitzungszeitraums zu gering war.



| No. | Time      | Source        | Destination   | Protocol | Length | Info                                     |
|-----|-----------|---------------|---------------|----------|--------|--|
| 1   | 0.000000  | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 2   | 0.499997  | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 3   | 1.500125  | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 4   | 3.501388  | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 5   | 7.503520  | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 6   | 11.503866 | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 7   | 15.504054 | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 8   | 19.504955 | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 9   | 23.505509 | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 10  | 27.505805 | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 11  | 31.506689 | 10.237.225.16 | 10.237.225.1  | SIP      | 370    | Request: BYE sip:30080@10.237.225.1:5060 |
| 12  | 67.993497 | 10.237.225.1  | 10.237.225.16 | SIP/SDP  | 760    | Request: INVITE sip:10.237.225.16:5060   |
| 13  | 68.000536 | 10.237.225.16 | 10.237.225.1  | SIP      | 287    | Status: 100 Trying                       |
| 14  | 68.009133 | 10.237.225.16 | 10.237.225.1  | SIP/SDP  | 806    | Status: 200 OK (INVITE)                  |
| 15  | 68.014841 | 10.237.225.1  | 10.237.225.16 | SIP      | 403    | Request: ACK sip:10.237.225.16:5060      |

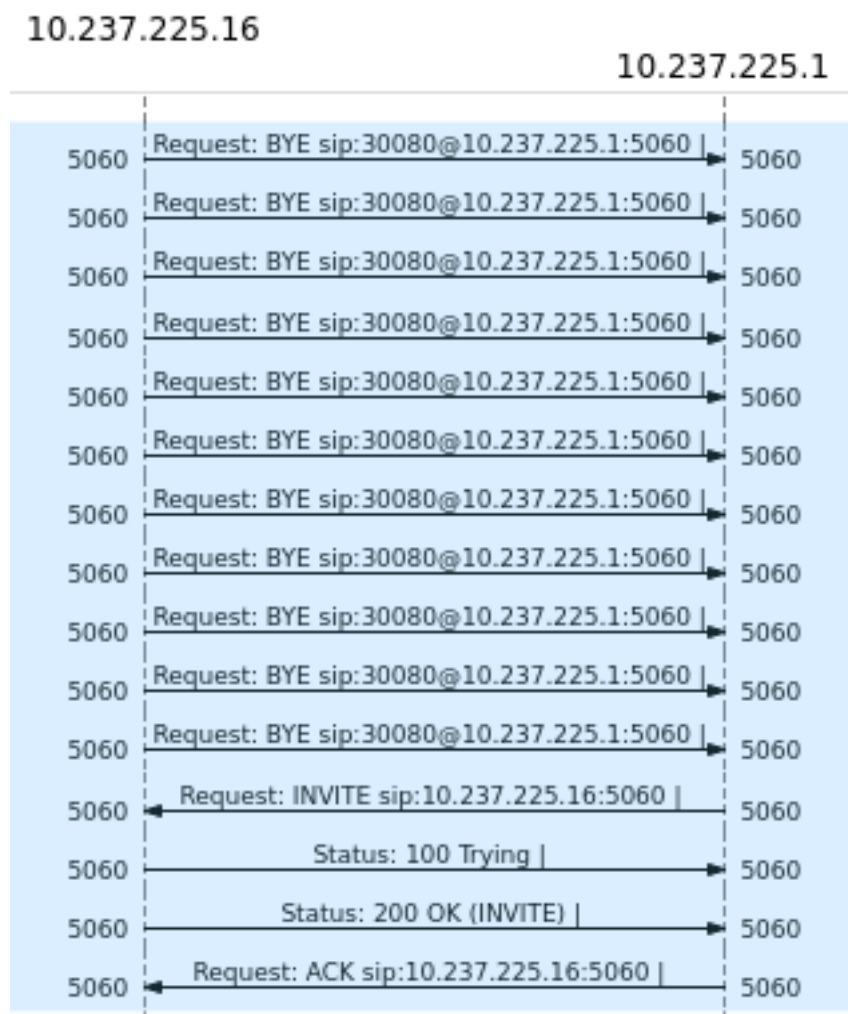
▶ Frame 12: 760 bytes on wire (6080 bits), 760 bytes captured (6080 bits) on interface \Device\NPF\_{60B079C4-4A54-4F24-9289-5222073BB27F}, id  
 ▶ Ethernet II, Src: Frequent\_01:17:f9 (00:01:bb:01:17:f9), Dst: Frequent\_01:72:dd (00:01:bb:01:72:dd)  
 ▶ Internet Protocol Version 4, Src: 10.237.225.1, Dst: 10.237.225.16  
 ▶ User Datagram Protocol, Src Port: 5060, Dst Port: 5060  
 ▶ Session Initiation Protocol (INVITE)

Request-Line: INVITE sip:10.237.225.16:5060 SIP/2.0  
 Method: INVITE  
 Request-URI: sip:10.237.225.16:5060  
 Request-URI Host Part: 10.237.225.16  
 Request-URI Host Port: 5060  
 [Resent Packet: False]

Message Header  
 Via: SIP/2.0/UDP 10.237.225.1:5060;rport;branch=z9hG4bK1118337311  
 From: <sip:30080@10.237.225.1:5060>;tag=158381371  
 To: <sip:10.237.225.16:5060>  
 Call-ID: 1912313353  
 [Generated Call-ID: 1912313353]  
 CSeq: 20 INVITE  
 Contact: <sip:30080@10.237.225.1:5060>  
 Content-Type: application/sdp  
 Allow: INVITE, ACK, BYE, CANCEL, INFO, UPDATE, REFER, NOTIFY, SUBSCRIBE, OPTIONS  
 Max-Forwards: 70  
 User-Agent: SipLib2-rel\_1\_509\_0  
 Subject: phone  
 Supported: timer, replaces  
 Session-Expires: 90  
 Min-SE: 90  
 Content-Length: 187

▶ Message Body

Abbildung 18: Re-INVITE mit Session Timer



**Abbildung 19:** Flowgraph mit Re-Tries für den BYE-Request

**Berechnen Sie die Bandbreite einer bidirektionalen VoIP-Verbindung (mit dem Codec G.711) mit den angegebenen Zahlenwerten. Gehen Sie dabei davon aus, dass alle 20 ms ein Sprachpaket abgegeben wird**

| Teil     | Größe    |
|----------|----------|
| FCS      | 4 Byte   |
| Payload  | 160 Byte |
| RTP      | 16 Byte  |
| UDP      | 8 Byte   |
| IP       | 20 Byte  |
| Ethernet | 14 Byte  |

- Alle 20ms ein Sprachpaket
- Pro Sekunde:  $\frac{1000ms}{20ms} = 50 \text{ Sprachpakete/s}$
- Wie groß ist jedes der Pakete?
- $4 \text{ Byte} + 160 \text{ Byte} + 16 \text{ Byte} + 8 \text{ Byte} + 20 \text{ Byte} + 14 \text{ Byte} = 222 \text{ Byte}$
- $50 \text{ Sprachpakete/s} \cdot 222 \text{ Bytes} = 11100 \text{ Bytes/s} = 88800 \text{ Bit/s} = 88 \text{ kBit/s}$
- Die Bandbreite einer VoIP-Verbindung beträgt mit dem G.711-Codec 88 kBit/s.