Hochschule Bochum
Campus Velbert/Heiligenhaus

Hochschule Bochum
Bochum University
of Applied Sciences
Campus
Velbert/Heiligenhaus

Dokumentation der Praktikumsversuche

von

Charles Biren, Stephan Rudat, Timon Baruch Scheffler, Marc Zentek

Eingereicht im: WS 2024/25

Prüfer: Herr Benedikt Wildenhain

Inhaltsverzeichnis

1 Versuch 1

1.1 Gruppenordner

Zu Beginn haben wir einen Gruppenordner mit einem ähnlichen Namen wie „IT-Infrastrukturen, Gruppe 5“ eingerichtet.

1.2 FusionPBX

FusionPBX ist ein Open-Source Projekt, das ein kostenloses, personalisierbares und flexibles Web-Interface für die VoIP-Plattform FreeSWITCH anbietet. FusionPBX läuft auf einer Vielzahl von Betriebssystemen und auf jeder ausreichend leistungsfähigen Hardware. Es beinhaltet eine GUI für unbegrenzte Erweiterungen, voicemail-to-email, music on hold, call parking, und viele weitere Fähigkeiten.[1]

1.3 Wireshark

Wireshark ist eine freie Software zur Analyse und grafischen Aufbereitung vom eigenen Netzwerkverkehr. Zu diesem Zweck ist Wireshark in vielen Industrien und Lehreinrichtungen der de facto Standard. Es bietet die Möglichkeit hunderte von Protokollen zu inspizieren, sowie einige zu entschlüsseln, eine Offline-Analyse der gesammelten Daten durchzuführen, Display-Filter auf den Datenstrom anzuwenden, VoIP-Analysen durchzuführen, Capture-Daten aus einer vielzahl von Formaten zu lesen, und vieles mehr.[2]

1.4 SIP

Das Session Initiation Protokoll (SIP) beschreibt ein Netzwerkprotokoll für den Aufbau, den Abbau und die Steuerung von Kommunikationssitzungen. Es wird am häufigsten in der IP-Telefonie eingesetzt. Hierzu sind sechs unterschiedliche Anfragen (REGISTER, INVITE, ACKNOWLEDGE, CANCEL, BYE, OPTIONS), sowie sechs Arten von Status-Codes (1xx (vorläufige Informationen), 2xx (Anfrage erfolgreich), 3xx (Umleitung), 4xx (Anfrage fehlgeschlagen), 5xx (Übertragung in einem Server fehlgeschlagen), 6xx (Transaktion fehlgeschlagen)) (Antworten) definiert.

[1] Crane (2024)
[2] Foundation (Nd)

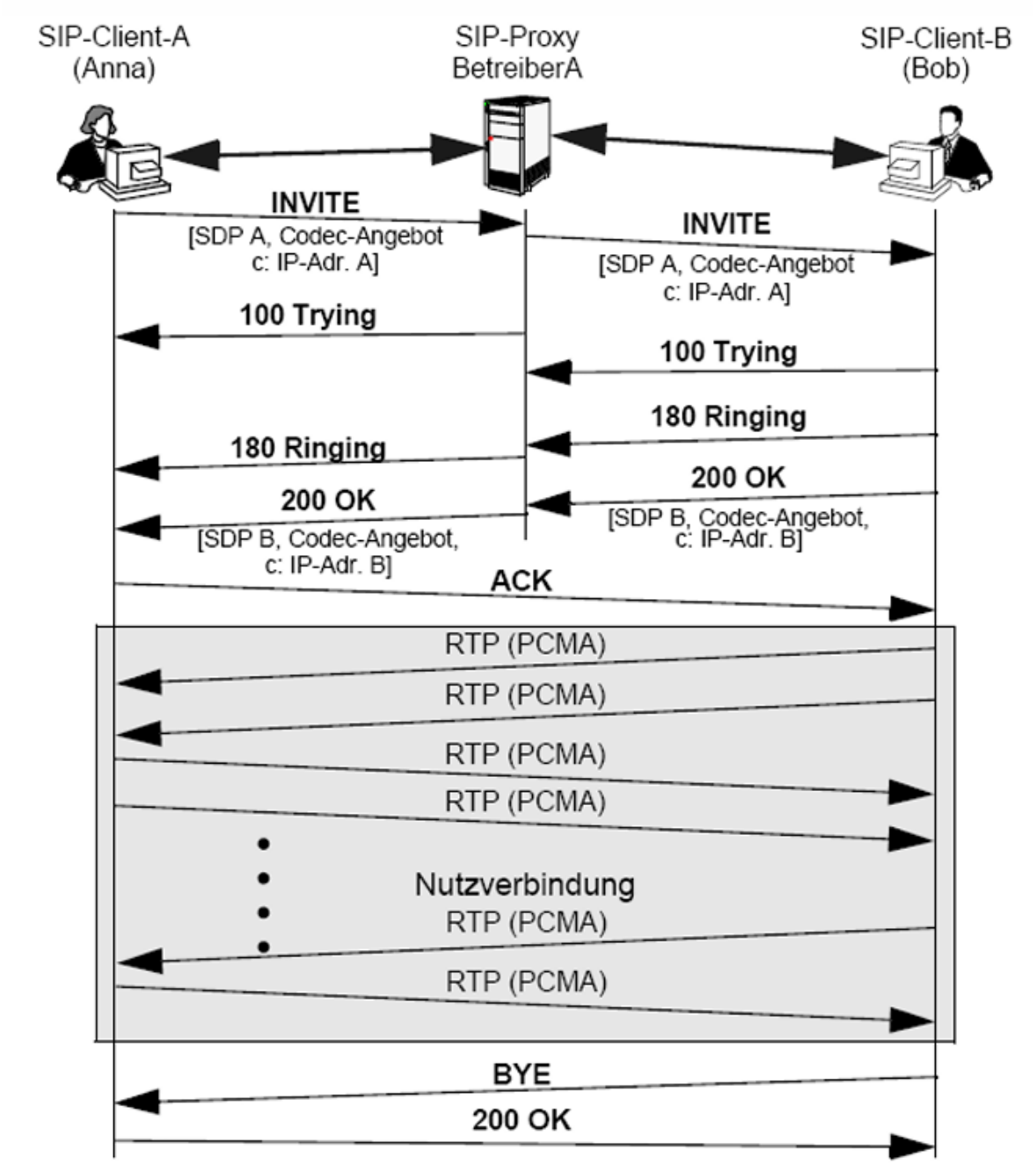

Bild 1.1: Ein beispielhafter Verbindungsauf- und -abbau in SIP

[3]

1.5 Versuch

Die erste Aufgabe umfasste die Änderung der IP-Adresse und die darauf folgende Einrichtung eines Administrationspassworts und SSH-Keys auf einem Router. Zuletzt sollten wir noch eine ESSID einrichten. Hierfür stand uns ein Router und ein eigener PC zur Verfügung. Im ersten Schritt haben wir über die IP-Adresse 192.168.1.1 auf den Router zugegriffen. Nach Eingabe des Passworts befanden wir uns auf OpenWrt. Von

[3] Rüsche (2024)

dort aus haben wir die IP-Adresse über "Network>Interface>Lan>edit"bearbeitet und auf 192.168.105.1/24 gesetzt. Über SSystem>Admnistration>Router Password"haben wir das Router-Passwort geändert. Unter SSH-Keys haben wir den gegebenen SSH-Key, beginnend bei ssh-rsa, eingetragen. Über "Network>Wireless"haben wir eine der inaktiven SSIDs gewählt und unter General Setup den ESSID Namen zu Gruppe5 geändert. Über Wireless Security haben wir die WPA2-PSK aktiviert und ein Passwort eingerichtet. Diese SSID arbeitet über Kanal 36 mit 5.180 GHz. Über diese Verbindung konnten wir den FreePBX-Server erreichen.

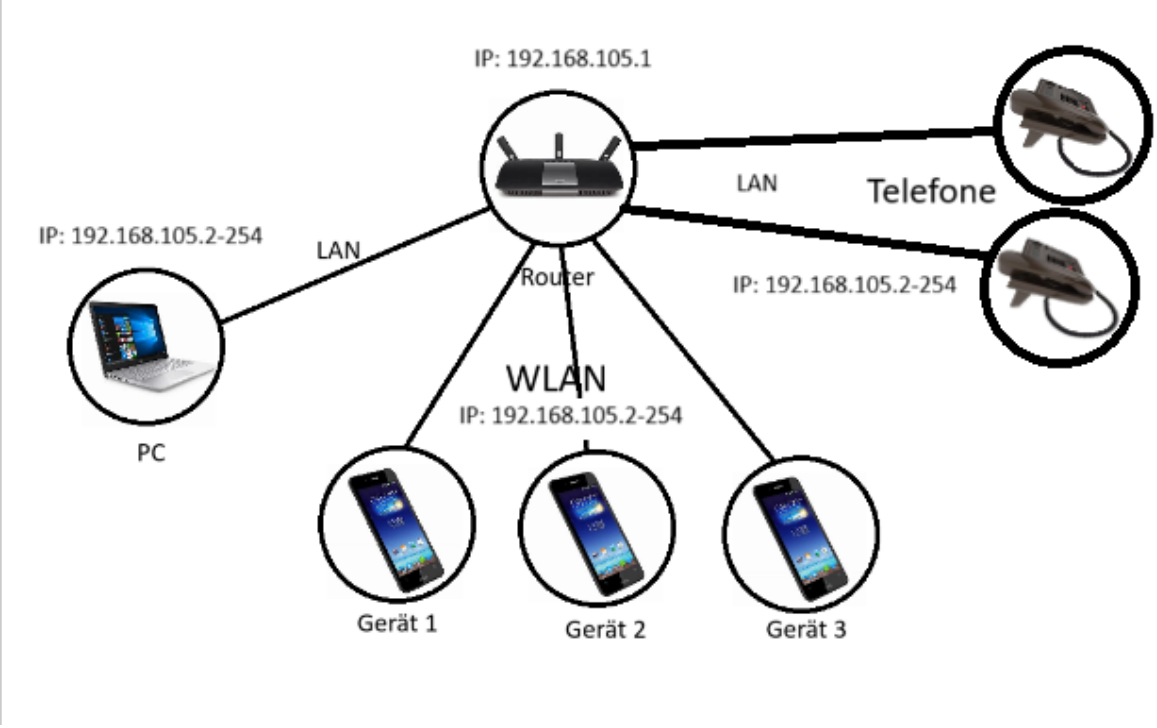

Bild 1.2: Der grobe Aufbau unserer physischen Infrastruktur

2 Versuch 2

2.1 Aufbau der geplanten Systemumgebung

2.1.1 Physische Netzwerkstruktur

- Geräte:
  - Telefon 1 (admin)
  - Telefon 2 (marc)
  - Smartphone (mit Linphone installiert) (Marc & Timon handy)
  - Router/Switch für IP-Verbindung
- Verkabelung:
  - Telefon 1 und Telefon 2 sind per Ethernet-Kabel mit dem Router verbunden.
  - Der Router verteilt an die IP-Adressen (DHCP) an die Telefone.
  - Notebook und Smartphone verbinden sich über das gleiche Netzwerk (ethernet für notebook/WLAN).

2.1.2 Logische Netzwerkstruktur

- Netzwerk:
  - DHCP: Aktiviert, IP-Adressen werden automatisch zugewiesen.
- IPV4-Adressen der Geräte:
  - Telefon 1: 192.168.105.232
  - Telefon 2: 192.168.105.207
  - Notebook: 192.168.105.243
  - Smartphone marc: 192.168.105.139
  - Smartphone timon : 192.168.105.166

2.2 Konfiguration der IP-Telefone

2.2.1 Telefon 1 (admin) und Telefon 2 (marc)

- Zugriff über das Web-Interface:
  - URL: 192.168.105.232, 192.168.105.207
  - Login-Daten: siehe PASSWORT.docx
  - Codecs: g722, pcmu, pcma, gsm, g723, g726-32, aal2-g726-32, g729, telephone-event
    - da wir mit einer ptime von 20 Millisekunden arbeiten ist das Verhalten von g723 in unserem Fall undefiniert[4]

[4]GmbH (2024)

  - RTP-Port-Bereich: 5000-5100

2.2.2 Notebook und Smartphone

- Linphone-Installation: Notebook
  - SIP-Konto: Benutzername: Notebook, Passwort: secure456
  - Server: 192.168.0.1

2.3 Sprachübertragungs-Tests

2.3.1 Zwischen Telefon 1 und Telefon 2

- **Testschritte:**
  1. Telefon 1 ruft Telefon 2 an.
  2. Telefon 2 nimmt den Anruf entgegen.
- **Ergebnis:** Klare Sprachqualität, keine merkbare Latenz.

2.3.2 Zwischen Telefon und Notebook/Smartphone

- **Testschritte:**
  1. Telefon 1 ruft das Notebook (Linphone) an.
  2. Notebook ruft Telefon 2 an.
  3. Smartphone ruft Telefon 1 an.
- **Ergebnis:** Erfolgreiche Sprachübertragung in allen Richtungen. Qualität abhängig von Netzwerkstärke. Für Notebook: rtc und rtcf verschlüsselungen der Telefone abgeschaltet.

2.4 Dokumentation der Systemparameter

2.4.1 Konfigurationsparameter

Gerät	IP-Adresse	Rufnummer	Benutzername	Passwort
Telefon 1	192.168.105.232	1	admin	pPCe*sgmxgkuVu54w?^e
Telefon 2	192.168.105.207	2	Marc	7muhe8JjpEHVH.zVCy?K
Notebook	192.168.105.243	4042	Marc-Laptop	Gruppe-5-Passwort
Handy (Marc)	192.168.105.139	404	Marc	Gruppe-5-Passwort
Handy (Timon)	192.168.105.166	405	Timon	Gruppe-5-Passwort

Tabelle 2.1: Konfigurationsparameter

2.4.2 Netzwerkdiagramm

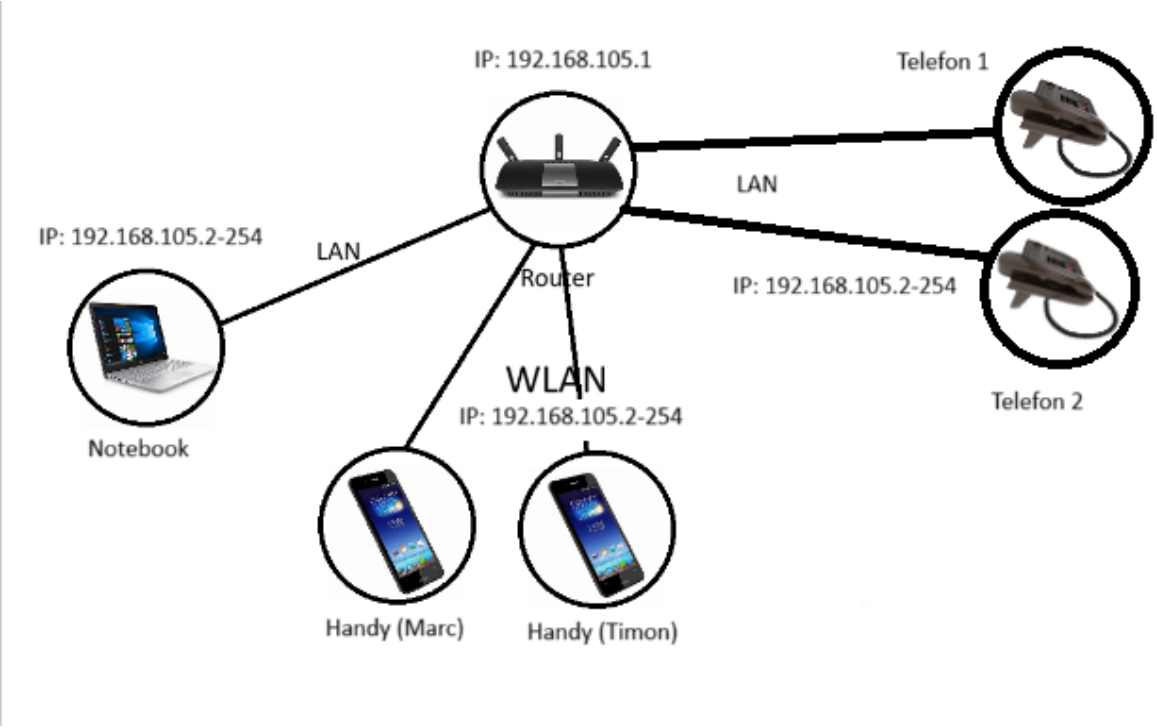

Bild 2.1: Die Netzwerkstruktur des zweiten Versuchs

2.5 Zusammenfassung

Die Systemumgebung wurde erfolgreich aufgebaut und getestet. Sprachübertragung funktionierte fehlerfrei zwischen allen Geräten. Sogar zwischen 2 Handys im gleichen Netz. Exakte Konfigurationen und Netzwerkinformationen wurden dokumentiert und im Gruppenordner gespeichert. Noch fehlende Parameter werden so bald wie möglich ergänzt.

2.6 Probleme

Anfangs gab es das Problem das wir die Router verwechselt haben mit Gruppe 1 und das der eine Router auch nicht resettet war. Dies hat die erste Stunde erhebliche Probleme gemacht da es bei keiner der Gruppen funktioniert hat. Nach einem Hardware RESET beider Geräte haben wir es endlich geschafft uns anzumelden an die klassische lokale Adresse. Zuerst hat soweit die Router Einstellung gut geklappt. Unsere gruppe blieb dann kurz hängen weil die zwei IP Telefone nicht funktionierten. Grund hierfür war, dass wir für jeden Anrufer einen „Account“ hinzufügen mussten. Gleiches Problem entstand bei den mobilen Telefonen, wurde aber recht schnell gefixt. Ein Problem welches wir am Ende hatten war bei der Verbindung zwischen dem Notebook und dem IP Telefon. RTC und RTCF Verschlüsslungen mussten abgeschaltet werden bei den Telefonen, sonst kam der Anruf vom Notebook nicht durch.

3 Versuch 3

3.1 Versuch

Um den Verbindungsaufbau und –abbau zu analysieren haben wir den Netzwerkverkehr sowohl aus Sender- als auch aus Empfängersicht aufgenommen und nach SIP gefiltert. Aus Empfängersicht stimmt der Verbindungsaufbau und –abbau mit der Theorie überein:
Der Proxy-Server leitet eine INVITE-Request an den Empfänger weiter. Dieser schickt zuerst einen 100 Trying-Status, dann einen 180 Ringing-Status und schließlich einen 200 Ok-Status zurück. Darauffolgend leitet der Proxy-Server eine ACK-Request weiter, woraufhin der Anruf selbst beginnt. Um den Anruf zu beenden schickt einer der Clients eine BYE-Request, welche von einem 200 Ok-Status beantwortet wird.
Aus Sendersicht stimmt der Verbindungsaufbau erst nach einer Authentifizierung gegenüber des Proxy-Servers mit der Theorie überein. Der Verbindungsabbau wiederum verläuft deckungsgleich mit dem aus Empfängersicht:
Der Sender schickt zu Beginn eine INVITE-Request an den Proxy-Server. Dieser antwortet mit einem 407 Proxy Authentication Required-Status. Auf diesen hin schickt der Sender eine ACK-Request und eine weitere INVITE-Request. Diese zweite INVITE-Request enthält, zusätzlich dem typischen Inhalt, einen Nonce-Authentifizierungs-Hash und in unserem Fall ebenfalls eine CNonce im SIP-Header. Dieser Austausch zwischen Sender und Proxy-Server entspricht nicht der Theorie, ist allerdings notwendig, um einen gewissen Grad an Sicherheit zu gewähren. Dieser Request folgen eine 100 Trying, eine 180 Ringing und eine 200 OK-Statusmeldung vom Proxy-Server. Nachdem der Sender noch eine ACK-Request abgesendet hat, beginnt der Anruf. Um den Anruf zu beenden schickt einer der Clients eine BYE-Request, welche von einem 200 Ok-Status beantwortet wird.
Bei einer Nonce[5], ausgeschrieben: Number Once, handelt es sich um eine einzigartige Nummer, die vom SIP-Server bei einer Anfrage generiert wird, und exakt ein einziges mal verwendet werden darf. Diese wird vom Sender benutzt, um, in unserem Fall mit dem MD5-Algorithmus, die Nutzerdaten in einen Hash-Code umzuwandeln. Zusätzlich generiert der Sender eine CNonce[6], ausgeschrieben: Client Number Once, und verhasht diesen Code noch einmal, bevor er den Hash-Code und die CNonce an den Server in der response sendet. Sobald der Server diese response empfängt, generiert dieser einen Hash-Code basierend auf seinen eigenen Daten und vergleicht die Hash-Codes.
Bei einem Hash-Code handelt es sich um einen Ausgabewert fester Länge einer nicht umkehrbaren Funktion. Im Fall der Kryptographie hat diese Funktion eine pseudozufällige Werteverteilung, sodass eine Ausgabe nur schwer auf eine Eingabe zurückgeführt werden kann.[7]

3.2 Probleme

Zu Beginn des Versuchs hatten wir einige Aufzeichnungen, in welchen keine SIP-Befehle zu finden waren. Dies lag daran, dass wir unsere Verbindung mit TLS verschlüsselt hatten. Um diese Verschlüsselung zu Umgehen haben wir für die Dauer des Versuchs unsere Linphone-Verbindung von TLS auf UDP umgeschaltet.

[5] Prokop (2015)
[6] Shekh-Yusef et al. (2015)
[7] Mohaisen (2023)

4 Versuch 4

4.1 Versuch

Für den Test der Verbindungsqualität haben wir Kanal 36 mit 5180 Mhz, Kanal 100 mit 5500 Mhz und eine Bandbreite von 80 MHz gewählt. Diese Werte lassen sich über die ESSID-Einstellungen des Routers anpassen. Hierbei hat Kanal 36 die niedrigste Geschwindigkeit, welche mit der Kanal-Nummer wächst. Zusätzlich gibt es eine automatische Kanal-Option. Außer 80 MHz gibt es noch Bandbreiten von 20 MHz und 40 MHz. Aus zeitlichen Gründen sind wir nur dazu gekommen die Verbindungsqualität auf Kanal 36 und 100 mit 80 MHz zu testen. Zum Messen ist eine Person beim Router geblieben und hat eine Aufzeichnung des Anrufs mit Wireshark vorgenommen. Eine andere Person ist mit einem Smartphone an unterschiedliche Orte im Gebäude gegangen und hat die Linphone-Statistik ausgelesen. Hierfür haben wir den Codec G.711-PCMU benutzt. Im Folgenden die Tabellen mit unseren Ergebnissen.

Kanal 36	In Raum 2-30	Vor Raum 2-30	Vor Raum 2-52	Vor Raum 2-48
Gesprächs-qualität	gut	gut	leichtes Rauschen	abgehackt, Abbruch nach 10s
Sender loss rate (max)	0%	0%	39.06%	1034%
Receiver loss rate (max)	0%	0%	0%	319%
Mean Jitter in ms	5.4	5.55	5.34	5.59

Tabelle 4.1: Die gemessenen Anrufparameter auf Kanal 36 (5180Mhz) bei einer Bandbreite von 80 Mhz

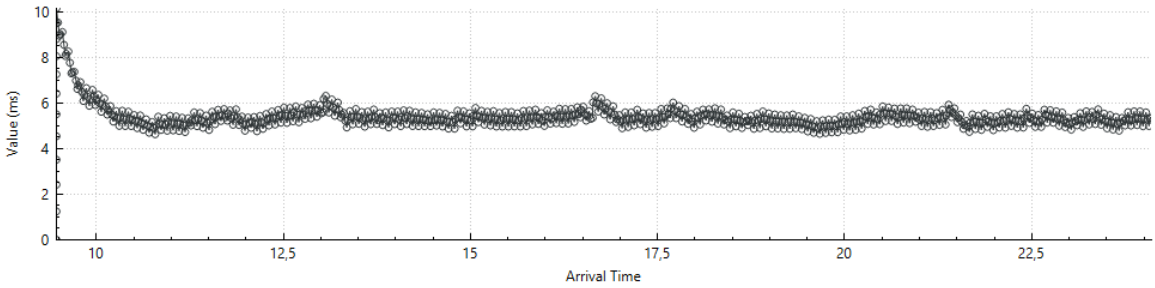

Bild 4.1: Der Verlauf des Jitter in 2-30 bei Nutzung von Kanal 36

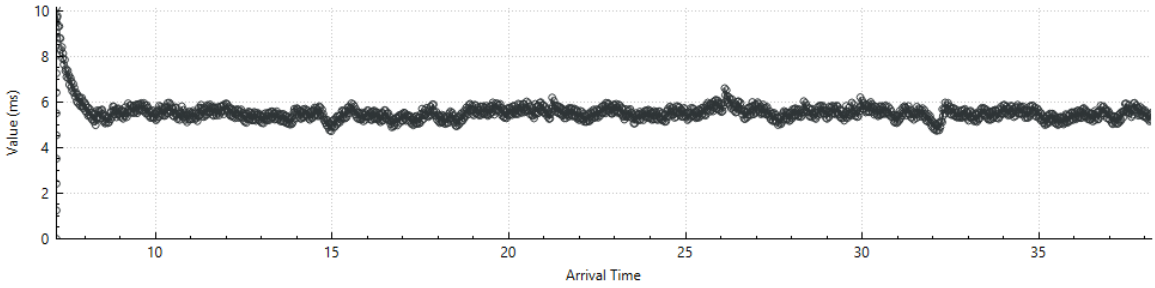

Bild 4.2: Der Verlauf des Jitter vor 2-30 bei Nutzung von Kanal 36

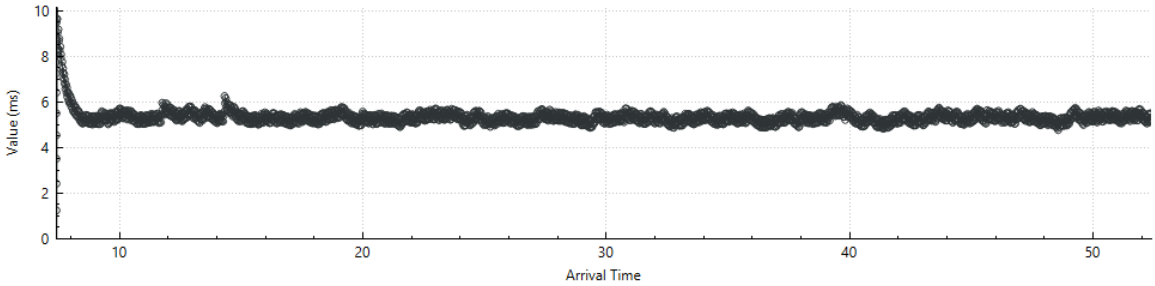

Bild 4.3: Der Verlauf des Jitter vor 2-52 bei Nutzung von Kanal 36

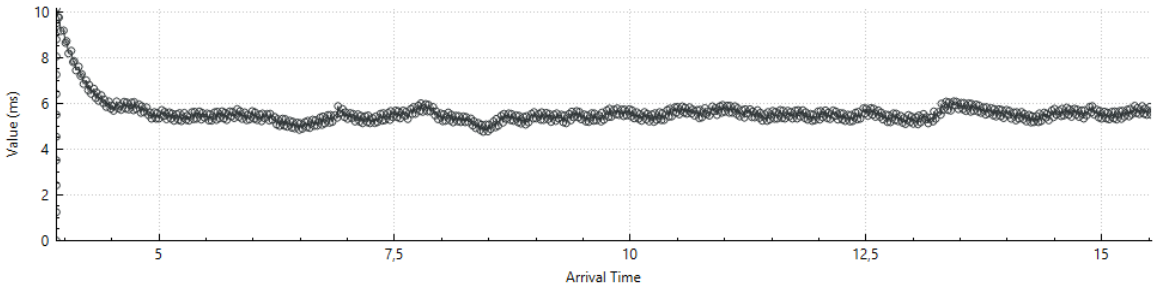

Bild 4.4: Der Verlauf des Jitter vor 2-48 bei Nutzung von Kanal 36

Kanal 100	In Raum 2-30	Vor Raum 2-30	Vor Raum 2-52	Vor Raum 2-48
Gesprächs-qualität	gut	gut	leichtes Rauschen	abgehackt, Abbruch nach 25s
Sender loss rate (max)	0%	39.06%	0%	3632%
Receiver loss rate (max)	0%	0%	0%	0%
Mean Jitter in ms	5.42	5.52	5.62	5.54

Tabelle 4.2: Die gemessenen Anrufparameter auf Kanal 100 (5500Mhz) bei einer Bandbreite von 80 Mhz

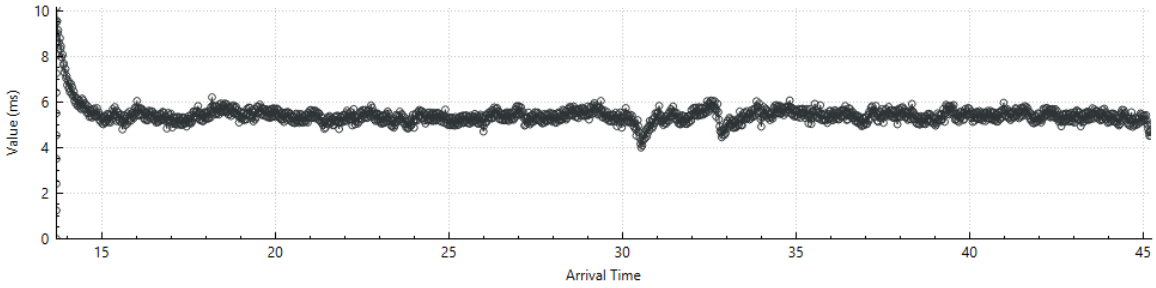

Bild 4.5: Der Verlauf des Jitter in 2-30 bei Nutzung von Kanal 100

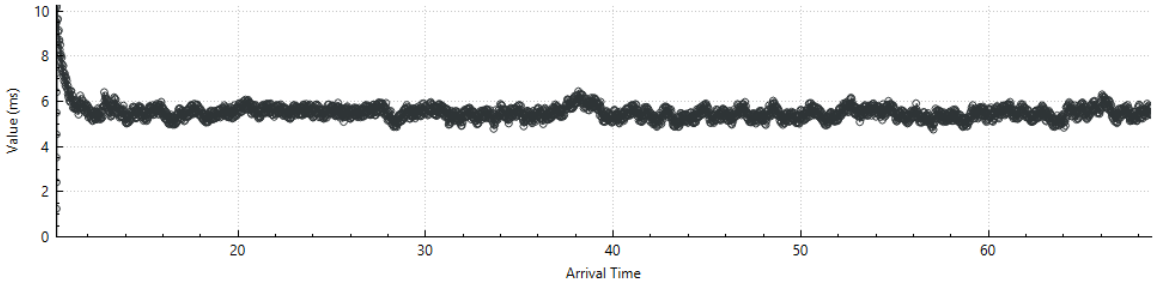

Bild 4.6: Der Verlauf des Jitter vor 2-30 bei Nutzung von Kanal 100

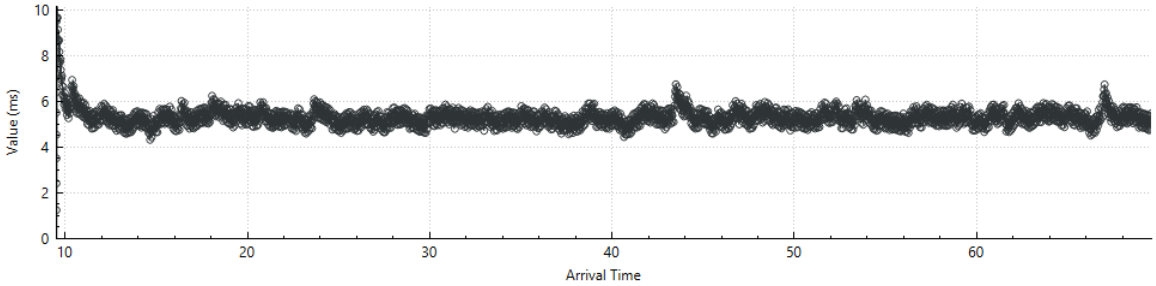

Bild 4.7: Der Verlauf des Jitter vor 2-52 bei Nutzung von Kanal 100

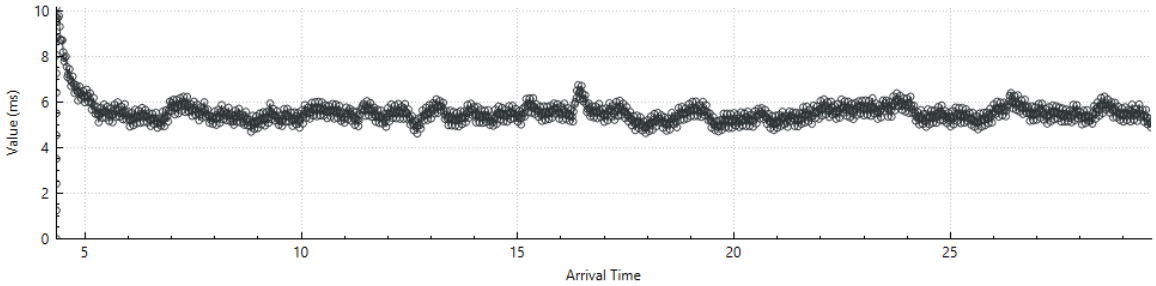

Bild 4.8: Der Verlauf des Jitter vor 2-48 bei Nutzung von Kanal 100

4.2 Probleme

Zu Beginn des Versuchs war uns nicht klar, was mit Prüfung der Qualität gemeint war. Daher haben wir in den ersten zwei Stunden eine Tabelle mit Gefühlswerten erstellt, be-

vor uns die Aufsicht auf die Möglichkeit, die Qualität mithilfe der Linphone-Statistiken auszulesen, aufmerksam gemacht hat. Aus diesem Grund ist uns zu Beginn des Versuchs viel Zeit verloren gegangen.
Bei ersten Tests ist uns aufgefallen, dass sowohl Linphone als auch Wireshark eine Verlustrate von 0% gezeigt haben. Da diese Werte keinen Sinn machen, besonders wenn man den Verbindungsabbruch vor Raum 2-48 bedenkt, haben wir uns letztendlich entschieden, die Verschlüsselung der Daten abzuschalten. Da Wireshark auch so noch die gleiche Verlustrate gezeigt hat, haben wir uns entschieden, die Verlustrate ohne Verschlüsselung aus Linphone und den durchschnittlichen Jitter aus Wireshark zu entnehmen.

Literatur- und Quellenverzeichnis

Crane, M. J. (2024). Welcome to FusionPBX Docs. https://docs.fusionpbx.com/en/latest/.

Foundation, W. (N.d.). About Wireshark. https://www.wireshark.org/about.html.

GmbH, S. T. (2024). Supported Codecs on Snom Desk Phones. https://service.snom.com/display/wiki/Supported+Codecs+on+Snom+Desk+Phones.

Mohaisen, M. (2023). Cryptographic Hash Functions: A Hands-on Introduction. https://medium.com/@mohaisen/cryptographic-hash-functions-a-hands-on-introduction-100c93a79f65.

Prokop, A. (2015). Understanding SIP Authentication. https://andrewjprokop.wordpress.com/2015/01/27/understanding-sip-authentication/.

Rüsche, S. F. (2024). Session Initiation Protocol (SIP). https://moodle.hs-bochum.de/pluginfile.php/921796/mod_resource/content/1/SIP_Information.pdf.

Shekh-Yusef, R., Ahrens, D., und Bremer, S. (2015). HTTP Digest Access Authentication. https://httpwg.org/specs/rfc7616.html.

Abbildungsverzeichnis

Tabellenverzeichnis