Implementation and Analysis of VoIP Services with NAT, Monitoring Capabilities and Interconnection

Suvendu Barai*
Communication Systems and Networks
Technische Hochschule Köln
Köln, Germany
suvendu.barai@smail.th-koeln.de

***Abstract*—This report presents the detailed implementation and analysis of a Voice over IP (VoIP) service provider network, following a structured approach across four key tasks. The first task focuses on building a robust VoIP service provider setup, ensuring efficient call routing and connectivity. The second task involves enabling and configuring comprehensive monitoring capabilities to maintain optimal network performance and troubleshoot issues. In the third task, a local area network (LAN) is transitioned to a remote site with Network Address Translation (NAT) to enhance network security and manageability. The final task examines the interconnection of multiple VoIP providers, analyzing the interoperability and performance metrics. Through these tasks, the report provides insights into the practical challenges and solutions in deploying and managing VoIP networks, highlighting the significance of NAT, monitoring, and inter-provider connectivity in ensuring reliable and high-quality VoIP services.**

***Index Terms*—Voice over IP (VoIP), Network Address Translation (NAT), Asterisk Server, Monitoring Capabilities, Peer-to-Peer VoIP, VoIP Service Provider, SIP (Session Initiation Protocol), RTP (Real-time Transport Protocol), LAN to Remote Site Transition, Interconnected VoIP Providers**

I. Introduction

The advent of Voice over IP (VoIP) technology has revolutionized communication by enabling voice calls to be made over Internet Protocol (IP) networks. VoIP offers numerous advantages over traditional telephony, including cost savings, flexibility, and the integration of various communication services. This report details the implementation and analysis of a VoIP service provider network, focusing on four critical tasks: building a VoIP service provider, enabling and configuring monitoring capabilities, transitioning a local area network (LAN) to a remote site with Network Address Translation (NAT), and creating and analyzing interconnected VoIP providers.

The first task involves setting up a VoIP service provider, which serves as the foundation for subsequent tasks. A robust VoIP infrastructure is crucial for ensuring reliable and high-quality voice communication. The implementation includes configuring the necessary hardware and software components, such as the Asterisk server, which is widely used for managing VoIP communications due to its flexibility and extensive feature set [1].

Monitoring capabilities are essential for maintaining the performance and reliability of VoIP services. The second task addresses the configuration of monitoring tools to track network performance, identify issues, and ensure the Quality of Service (QoS). Effective monitoring helps in proactive maintenance and rapid troubleshooting, thereby minimizing downtime and enhancing user satisfaction [2].

Transitioning LAN C to a remote site with NAT is the focus of the third task. NAT plays a critical role in IP address management and security, particularly when integrating remote sites into a broader network infrastructure. By translating private IP addresses to a public IP address, NAT allows for seamless communication while preserving the internal network's integrity and security [3], [4].

The final task examines the interconnection of multiple VoIP providers, which is increasingly important as organizations seek to integrate diverse communication systems. This task involves analyzing the interoperability and performance of interconnected VoIP networks, highlighting the challenges and solutions in achieving seamless communication across different providers [5], [6].

This report provides a comprehensive analysis of the practical challenges and solutions encountered in deploying and managing a VoIP network. The insights gained from these tasks underscore the importance of NAT, monitoring, and inter-provider connectivity in ensuring reliable and high-quality VoIP services.

II. Objectives

The objective of this report is to document the configuration, implementation, and analysis of a Next Generation Network (NGN) with VoIP capabilities through a series of structured milestones.

- **Basic VoIP Network Configuration and Operations (Milestone 1):**
  - Establish a fundamental VoIP network setup using Asterisk servers.
  - Configure and verify SIP-based VoIP communications within a local network (LAN-A and LAN-C).
  - Test and validate SIP and RTP traffic flow, ensuring proper registration and call setup.
- **Monitoring and Analyzing VoIP Traffic (Milestone 2):**
  - Enable and configure monitoring capabilities to capture and analyze VoIP traffic.
  - Implement SPAN port mirroring on switches to facilitate detailed traffic analysis using Wireshark.
  - Evaluate SIP registration processes, call setup, and codec performance through traffic captures.
- **VoIP Network with NAT and Peer-to-Peer Traffic (Milestone 3):**
  - Transform LAN-C into a remote site with NAT configuration to simulate real-world scenarios.
  - Enforce peer-to-peer VoIP RTP traffic routing through NAT configurations.
  - Analyze the effects of NAT on VoIP call signaling and media traffic, ensuring seamless communication despite NAT traversal challenges.
- **Interconnection of Multiple VoIP Service Providers (Milestone 4):**
  - Establish inter-connectivity between two separate VoIP service provider networks.
  - Configure SIP trunks to enable VoIP call routing between different networks.
  - Analyze SIP signaling, call setup processes, and transcoding mechanisms for interconnected VoIP services.
  - Ensure reliable and efficient communication between VoIP clients across interconnected networks.

III. Configuration Details

A. LAN A, B, C IP Addressing

- LAN A is configured with the subnet address 10.3.0.0/24.
- This subnet accommodates devices in LAN A, including Switch (S1), Router (R1), VoIP phone and the Asterisk server (PC-A). The outer interface of R1 (g0/0/0) which is configured with public IP address (139.6.19.132) and connected to Internet (DN.LAB access switch)
- LAN B uses the subnet address 10.3.1.0/24.
- The routers R1 and R2 in this LAN are connected via serial interfaces.
- LAN C is configured with the subnet address 10.3.2.0/24.
- This subnet hosts Switch (S2), Router (R2), VoIP phone, Monitoring PC (PC-B) and connects to external networks through R2.

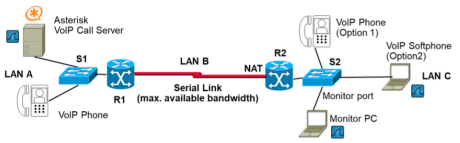


Fig. 1. Topology for Milestone 1,2,3

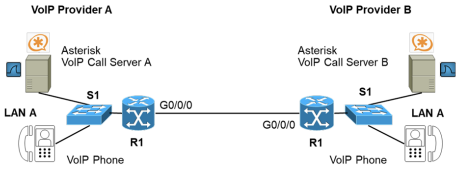


Fig. 2. Topology for Milestone 4

LAN A	Subnet Address	Subnet Mask
Default Gateway (R1)	10.3.0.1	255.255.255.0
Asterisk/Iperf Server (PC-A)	10.3.0.2	255.255.255.0
VoIP Phone	10.3.0.3	255.255.255.0
Outside Interface of R1 (g0/0/0)	139.6.19.132	255.255.255.224

TABLE I
IP Address Allocation for LAN A and Router R1

LAN B	Interface No.	IP Address	Subnet Mask
R1 Serial Interface	s0/1/0	10.3.1.1	255.255.255.0
R2 Serial Interface	s0/1/0	10.3.1.2	255.255.255.0

TABLE II
IP Address Allocation for LAN B

LAN C	IP Address	Subnet Mask
Default Gateway (R2)	10.3.2.1	255.255.255.0
Iperf Client (PC-B)	10.3.2.2	255.255.255.0
VoIP Phone	10.3.2.3	255.255.255.0

TABLE III
IP Address Allocation for LAN C

B. Network Topology and Device Initialization

- **Routers:**
  - **Router R1:**
    - Connects LAN A (10.1.0.0/24) and LAN C (10.3.2.0/24) via interfaces g0/0/0 and

s0/1/0, respectively.
    - Implements OSPF for dynamic routing between LAN A and LAN C.
  - **Router R2:**
    - Manages NAT (Network Address Translation) for LAN C (10.2.0.0/24) using interface s0/1/0.
    - Provides connectivity to external networks for LAN C.
- **Switches:**
  - **Switch S1:**
    - Interconnects devices in LAN A (10.1.0.0/24) via interfaces Fa0/1 and Fa0/2.
  - **Switch S2:**
    - Configured with a mirrored port (Fa0/24) for monitoring purposes.

C. Basic Switch and Router Settings

- **Router R1:**
  - Configured with NAT to translate LAN C (10.2.0.0/24) addresses for external communication.
  - OSPF routing protocol ensures dynamic routing across LAN A and LAN C.
- **Router R2:**
  - Implements NAT overload (PAT) for LAN C using ACL 1 and interface s0/1/0.
  - Interfaces g0/0/0 and s0/1/0 designated as NAT inside and outside interfaces, respectively.

D. Network Address Translation (NAT)

- **Router R2 Configuration:**
  - Applied NAT overload to LAN C subnet (10.2.0.0/24) using ACL 1.
  - Interface s0/1/0 designated as the NAT outside interface.

E. Extending the Network

- **LAN-X (10.23.0.0/24):**
  - Erasing NAPT translation in R1.
  - Established a new router named R23 to extend the network.
  - Connected all the LAN A routers of each team to a common switch.
  - The common switch was connected to R23.
  - Each team's LAN A router was connected to the common switch to facilitate inter-connectivity.
  - R23 was then connected to the uplink switch that provides internet access.
  - This setup enables inter-connectivity between all teams' Asterisk servers, allowing calls between each team's extensions.

IV. Approach and Methodology

This section outlines the step-by-step approach and methodology followed in each milestone, detailing the tasks performed and the objectives achieved. Each milestone progressively built upon the previous one to create a comprehensive VoIP network, facilitating thorough evaluation and troubleshooting. The methodology for each milestone is outlined below:

Milestone 1: Basic VoIP Network Configuration and Operations

1) **Network Setup:**
   - Configured two local area networks (LAN-A and LAN-C) with appropriate IP (subnet) addressing in Switches(S1, S2) and Routers (R1, R2). Both routers are connected via serial interfaces (s0/1/0) with V.35 cable and consists LAN-B network. The serial link bandwidth has also been configured.
   - Deployed Asterisk server in LAN-A (PC-A) after editing all the config files (pjsip, extensions) for both local users to handle SIP-based VoIP communications (proxy mode).
   - Configured VoIP clients in both LANs to register with the Asterisk server.
2) **Testing and Validation:**
   - Verified SIP registration of VoIP clients with the Asterisk server.
   - Conducted VoIP call tests between clients in LAN-A and LAN-C.
   - Used tools such as 'ping' and 'traceroute' to ensure network connectivity and diagnose any issues.

Milestone 2: Monitoring and Analyzing VoIP Traffic

1) **Enabling Monitoring:**
   - Configured SPAN (Switched Port Analyzer) ports on the network switches to mirror traffic to a monitoring port.
   - Connected a dedicated Monitor PC to the SPAN port for traffic capture.
2) **Traffic Capture and Analysis:**
   - Used Wireshark on the Monitor PC and to capture SIP and RTP traffic.
   - Initiated VoIP calls and captured traffic at various network points, including the Asterisk server and VoIP clients.
   - Analyzed the captured traffic to evaluate SIP registration processes, call setup, and media transmission.

```
[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

[3001]
type=endpoint
context=from-internal
allow=all
direct_media=no
auth=auth_3001
aors=aor_3001

[auth_3001]
type=auth
auth_type=userpass
password=password
username=3001

[aor_3001]
type=aor
max_contacts=1

[3002]
type=endpoint
context=from-internal
allow=all
direct_media=no
auth=auth_3002
aors=aor_3002

[auth_3002]
type=auth
auth_type=userpass
password=password
username=3002

[aor_3002]
type=aor
max_contacts=1
```

Fig. 3. PJSIP config for proxy mode

```
[phone1]
exten => 3001,1,Answer()
exten => 3001,n,Dial(PJSIP/3001,20)
exten => 3001,n,Hangup()

[phone2]
exten => 3002,1,Answer()
exten => 3002,n,Dial(PJSIP/3002,20)
exten => 3002,n,Hangup()

[from-internal]
include => phone1
include => phone2
```

Fig. 4. EXTENSION config

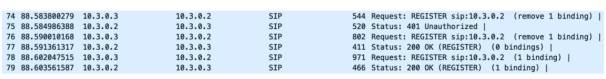


Fig. 5. Registration of caller party with server

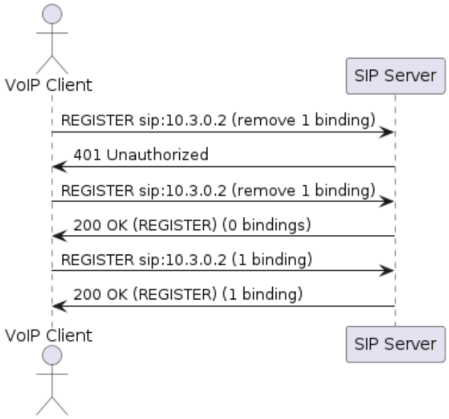


Fig. 6. MSC of caller party with server

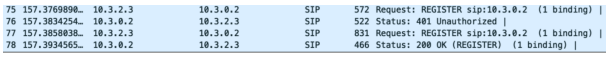


Fig. 7. Registration of called party with server

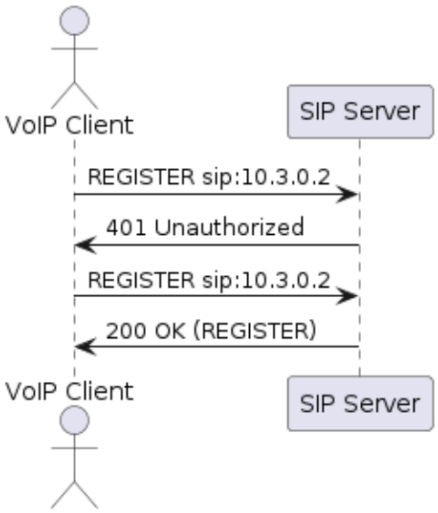


Fig. 8. MSC of called party with server

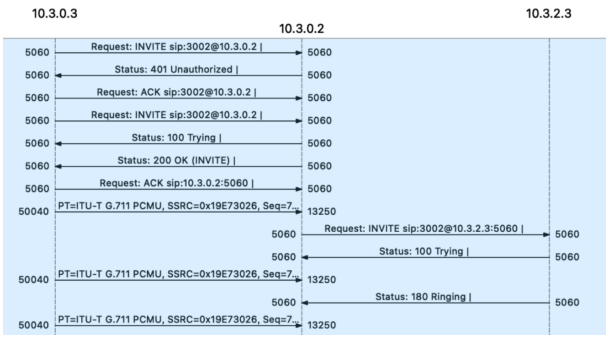


Fig. 9. VoIP call establishment between both parties through asterisk server (SIP phase)

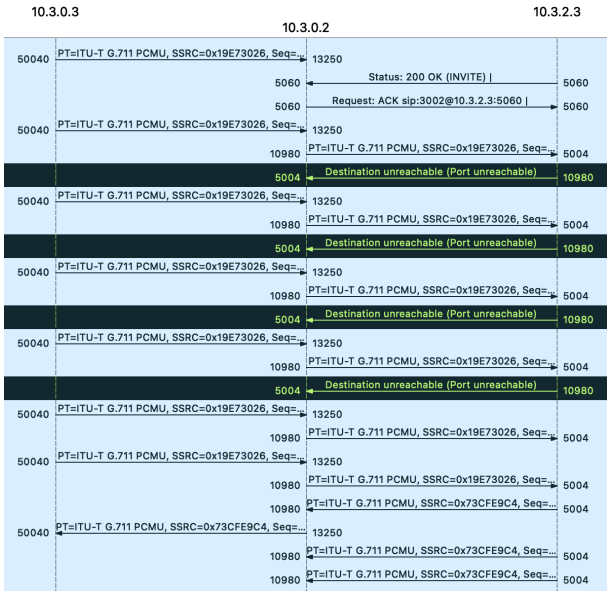


Fig. 10. VoIP call establishment between both parties through asterisk server (RTP phase)

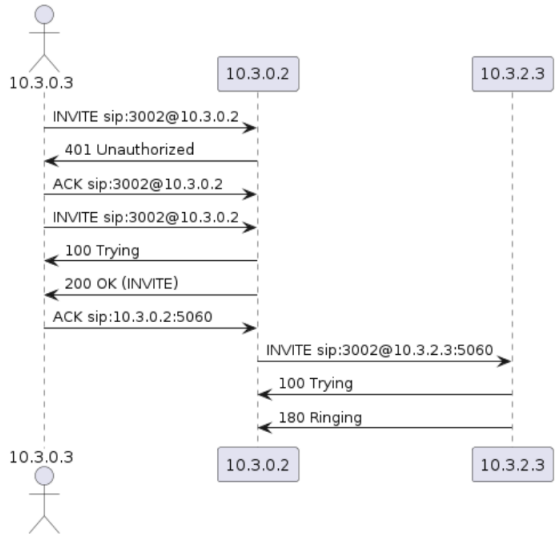


Fig. 11. MSC of call setup between both parties through proxy mode

- Evaluated registration process, call setup process in proxy mode, alanyzed signaling and media traffic, measured codec performance, throughput, call setup latency.

Milestone 3: VoIP Network with NAT and Peer-to-Peer Traffic

1) **Transforming LAN-C into a Remote Site:**
   - Configured Network Address Translation (NAT) on the router connecting LAN-C (R2) to simulate a remote site.
   - Removed static routes and implemented NAT with access control lists (ACL) to restrict traffic.

2) **Peer-to-Peer VoIP Traffic:**
   - Reconfigured the Asterisk server (editing pjsip.conf) to enable peer-to-peer media traffic.

```
[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

[3001]
type=endpoint
context=from-internal
allow=all
direct_media=yes
direct_media_method=invite
disable_direct_media_on_nat=no
auth=auth_3001
aors=aor_3001

[auth_3001]
type=auth
auth_type=userpass
password=password
username=3001

[aor_3001]
type=aor
max_contacts=1

[3002]
type=endpoint
context=from-internal
allow=all
direct_media=yes
direct_media_method=invite
disable_direct_media_on_nat=no
auth=auth_3002
aors=aor_3002

[auth_3002]
type=auth
auth_type=userpass
password=password
username=3002

[aor_3002]
type=aor
max_contacts=1
```

Fig. 12. PJSIP config for peer-to-peer mode

   - Captured and analyzed VoIP traffic to ensure proper NAT traversal and direct media path establishment.
   - Evaluated the impact of NAT on SIP signaling and RTP media traffic.

3) **Testing and Validation:**
   - Evaluated VoIP call functionality from LAN-A to LAN-C and vice versa in Peer-to-Peer mode after erasing NAT.
   - Listed NAT translations.

- Inspected SIP signaling and RTP media traffic for both directions.
- Checked debug messaged at the Asterisk terminal.
- Compared this peer-to-peer behavior with Asterisk Server in Proxy Mode.
- Ensured that calls could be established despite NAT, analyzing any issues with signaling or media path.

Milestone 4: Interconnection of Multiple VoIP Service Providers

1) **Network Interconnection:**
   - Partnered other teams to interconnect their LANs (10.1.0.0/24, 10.2.0.0/24) using a new LAN-X (10.23.0.0/24).
   - Configured router interfaces (IP: 10.23.2.0, subnet mask: 255.255.255.0) and established routing between the interconnected networks.
   - Verified connectivity between Asterisk servers in the interconnected networks.
2) **VoIP Interconnection via SIP Trunk:**
   - Configured Asterisk servers to operate in proxy mode.
   - Created a SIP trunk between the Asterisk servers of both teams to enable VoIP call routing between the networks.
   - Tested VoIP calls across the interconnected networks to ensure successful call routing.
3) **SIP Interconnection Analysis:**
   - Captured and analyzed SIP signaling traffic during inter-network VoIP calls.
   - Evaluated SIP registration, call setup processes, and measured call setup latency.
   - Tested and analyzed transcoding by using different codecs on VoIP clients in the interconnected networks.

V. Evaluation Results

A. Milestone 1

- **Objective:** Configure and verify the subnet addressing for all LANs and setup VoIP service.
- **Activities:** Assigned IP addresses within to all devices in LAN A, B, C. Installed and updated Asterisk software to use PC-A as UAS for VoIP communication between LAN-A and LAN-C. Configured local users (pjsip.conf) and VoIP call routing rules(extensions.conf) files within the server. Configured VoIP phones (clients) with the registrar server, user and password credentials.
- **Evaluation Metrics:**
  - Correctness of IP assignments
  - Network connectivity
  - Establishment of VoIP call
- **Results:**
  - **Summary:** All devices in LANs were successfully assigned IP addresses within the allocated subnets.
  - **Data Presentation:** 100% of devices showed correct IP assignments and successful pinging was possible.
  - **Analysis:** The subnet addressing was implemented correctly, with all devices achieving the desired network connectivity. A successful VoIP communication was established between clients in proxy mode.
  - **Challenges:** Configuring asterisk server with local users (pjsip.conf) and voip call routing (extensions.conf), configuring clients with proper user addressing and accessing web interfaces was thoroughly done. VoIP phones were successfully registered with server from the beginning but still couldn't call from one LAN to another initially.
    Database locked issue: Multiple instances of asterisk were running concurrently, which were identified through pid in asterisk terminal. This led to database locked error.
    Invalid URI issue: The client in LAN C (3002) was unreachable due to this error.
    Unnecessary instances were terminated to resolve the issue.

B. Milestone 2

- **Objective:** Enable and Configure Monitoring Capabilities.
- **Activities:** Configured monitor port on S2 to capture Ethernet frames. Validated traffic monitoring across VoIP network components. Analyzed SIP signaling (Registration: authentication mechanism, cryptographic algorithm and Call setup: codec selection mechanism, security) and media traffic using Wireshark. Measured call setup latency and data link layer throughput of VoIP codecs.
- **Evaluation Metrics:**
  - Effectiveness of traffic capture
  - Analysis of SIP signaling and media traffic
  - Measurement of data link layer throughput
- **Results:**
- **Task 1:** Configured SPAN port Fa0/24 on Switch S2 successfully. Monitor PC connected and captured traffic without issues.
- **Task 2:** Verified VoIP traffic capture at Asterisk server and Monitor PC for both signaling and media traffic. Addressed initial connectivity issues.
- **Task 3:**
  - **Part 1:** Captured SIP registration traffic accurately. Analyzed authentication mechanisms and SIP message flow.
  - **Part 2:** Inspected SDP (Session Description Protocol) in SIP, codec selection process and measured call setup latency.

'401 Unauthorized' response from the server, specifying the authentication method:

Client's response with REGISTER request with necessary credentials (username, password)

Fig. 13. Authentication mechanism and cryptographic algorithm (MD5)

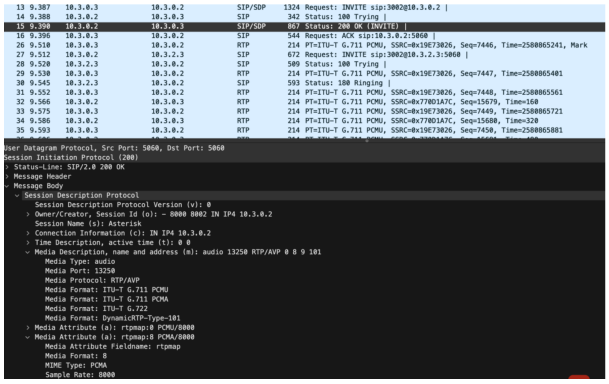


Fig. 14. Session Description Protocol (SDP) inspection

  - **Part 3:** Measured and compared data RTP user data throughput for both real data link layer and wireshark captured. Also addressed filtering challenges.
- **Challenges:** Port Mirroring allows to send a copy of network packets seen on one (or multiple) switch ports (or VLAN) to a monitoring port connected to a network analyzer or monitoring device (like Wireshark running on a Monitor PC). Carefully choosing ports to be mirrored, either ingress or egress traffic but not both to avoid duplicate packets catch.
  There was initial issues of IP packets flowing directly and not in proxy mode through the server and it was resolved shortly with proper pjsip configuration for proxy method.
  Why is the real Data Link Layer throughput even a little bit higher than the measured throughput of Wireshark capture? Calculating and explaining this question was a bit critical with the missing data in the Wireshark capture. Forgot to take FCS values which was later corrected.

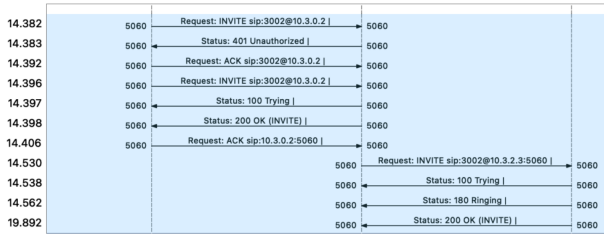
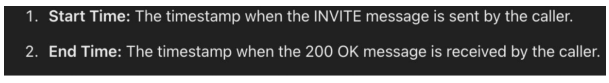


Fig. 15. Timestamp of VoIP call setup



Call Setup Latency = 19.892−14.382 = 5.510 seconds

Fig. 16. Latency calcuation

C. Milestone 3

- **Objective:** Configure LAN C as a remote site with NAT enabled to manage VoIP traffic.
- **Activities:** Deleted static IP route to LAN-C in Router R1. Configured NAT on Router R2 for LAN-C with ACL. Tested connectivity between routers and Asterisk server. Checked default NAT operation of Asterisk server. Reconfigured Asterisk server for peer-to-peer operations.

RFC 3581 Title	An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing
Description of Asterisk behavior in case NAT	By default, Asterisk does minimal NAT handling based on RFC 3581. This means it uses the `rport` parameter in SIP headers to ensure responses are sent back to the correct IP and port, accommodating for NAT traversal.

Fig. 17. Asterisk default NAT operation with RFC 3581

- **Evaluation Metrics:**
  - Configuration Deletion and NAT Setup
  - Ping tests
  - VoIP traffic analysis
  - Call success or failure analysis
  - SIP messages comparison between Peer-to-peer and proxy mode
- **Results:**
  - Static route to LAN-C deleted successfully on R1.
  - NAT configured on R2 with standard ACL for LAN-C IP addresses.
  - Ping tests from R2 to R1 and Asterisk server successful but not the other way around.
  - Signaling and media traffic captured for VoIP calls, with analysis showing successful (LAN C to A) and unsuccessful call scenarios (LAN A to C).
  - Asterisk server reconfigured for peer-to-peer operations, enhancing direct VoIP traffic routing.
- **Challenges:** Clearing NAT table was still not fast enough to make a call unsuccessful from LAN A to C. The automatic NAT entries were generatimg very quickly to establish connection between both LANs. Also tried creating another ACL (extended ip access list 100 (10 deny ip 10.3.2.0 0.0.0.255 any)) to deny the specific subnet for not making

Fig. 18. Unsuccessful call from VoIP phone in LAN-A to VoIP phone in LAN-C

communication. But finally it worked in the instructed way with erasing NAT translations and evaluated the impact of NAT.
Understanding relevant NATs in router R2 for establishing VoIP calls was a bit confusing initially, specifically outside local and global addresses.
Breaking down debug info and what they indicate regarding asterisk NAT handling was complicated. SIP Registration, SIP INVITE and Responses, RTP Media Streams, NAT Related Settings were interpreted with 'pjsip set logger on', core set verbose 5, core set debug 5 commands. Insights from debug messages include: Correct IP Address Handling (public IP address in the Contact header and not the private IP address behind the NAT), Port Mapping (correct RTP ports are being used and mapped, send and receive media through the NAT), Session Timers and Keep-Alives, Peer-to-Peer (Direct Media). Overall, how Asterisk handles NAT traversal for SIP signaling and RTP media.

D. Milestone 4

- **Objective:** Extend the network to allow interconnectivity between different teams' Asterisk servers for end-to-end VoIP communication.
- **Activities:**
  - Established a connection between all teams' LAN-A networks via a common switch linked to router R23.
  - Configured R23 to connect to the uplink switch, facilitating internet access and inter-team connectivity.
  - Reconfigured asterisk server to support proxy mode only.
  - Implemented SIP trunking to establish call service between servers.

```
[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

[3001]
type=endpoint
context=from-internal
allow=all
direct_media=no
auth=3001
aors=3001

[3001]
type=auth
auth_type=userpass
password=password
username=3001

[3001]
type=aor
max_contacts=1

[3002]
type=endpoint
context=from-internal
allow=all
direct_media=no
auth=3002
aors=3002

[3002]
type=auth
auth_type=userpass
password=password
username=3002

[3002]
type=aor
max_contacts=1

[trunk3]
type=identify
match=10.2.0.2
endpoint=trunk3

[trunk3]
type=endpoint
context=from-internal
allow=all
direct_media=no
auth=trunk-auth
outbound_auth=trunk-auth
aors=trunk3

[trunk-auth]
type=auth
auth_type=userpass
username=team2
password=trunk

[trunk3]
type=aor
max_contacts=10
contact=sip:10.2.0.2:5060

[trunk3]
type=registration
outbound_auth=trunk-auth
server_uri=sip:10.2.0.2
client_uri=sip:trunk3@10.3.0.2
```

Fig. 19. PJSIP config with trunking functionality in proxy mode

  - Examined SIP messages for mutual registration between servers.
  - Conducted VoIP call tests between different teams' extensions to ensure successful communication.
  - Analyzed signaling and media traffic for inter-team calls, confirming correct routing and handling of VoIP traffic.

```
[phone1]
exten => 3001,1,Answer()
exten => 3001,n,Dial(PJSIP/3001,20)
exten => 3001,n,Hangup()

[phone2]
exten => 3002,1,Answer()
exten => 3002,n,Dial(PJSIP/3002,20)
exten => 3002,n,Hangup()

[team2]
exten => _2XXX,1,Answer()
exten => _2XXX,n,Dial(PJSIP/${EXTEN}@trunk3,20)
exten => _2XXX,n,Hangup()

[team1]
exten => _1XXX,1,Answer()
exten => _1XXX,n,Dial(PJSIP/${EXTEN}@trunk1,20)
exten => _1XXX,n,Hangup()

[from-internal]
include => phone1
include => phone2
include => team2
include => team1
```

Fig. 20. EXTENSION config for call routing with other servers

  - Evaluated the impact of transcoding process with different audio codecs for endpoints.
- **Evaluation Metrics:**
  - Successful VoIP call tests between different teams' extensions.
  - Analysis of signaling and media traffic to ensure correct routing and handling of VoIP calls.
  - Resolution of issues related to NAT traversal, codec incompatibility, and registration errors.
- **Results:**
  - Router R23 configured and connected to the uplink switch, enabling internet access and inter-team connectivity.
  - VoIP call tests between different teams' extensions were successful, demonstrating proper inter-connectivity.
  - Signaling traffic for mutual server registration captured, media traffic analyzed, confirming correct routing and handling of VoIP traffic.
  - Transcoding process evaluated with RTP flows from one codec to others.
- **Challenges:** Interconnecting all the routers and achieving internet connectivity for LAN X was problematic initially but troubleshooting through traceroute command to understand in which hop the packet is unreachable was insightful.
  The configuration of pjsip file for identifying trunk 3 (10.2.0.2) and trunk 1 (10.1.0.2), verfication of endpoints (server) and authentication failure was skipped at first but after going through documentations, able to fix the issue which then solved the issue of finding the targeted servers.
  During evaluation of transcoding, there were codec negotiation issues which was flagged by Warning 6499 from terminal. Lost frames and quality considerations were relative during transcoding process and due to that there were not many apparently clear messages in the terminal for understanding codec conversion clearly.

```
NOTICE[43128][C-00000002]: translate.c:603 ast_translate: 18761 lost frame(s)
18762/0 (slin@8000)->(ulaw@8000)
```

Fig. 21. Notification of lost frames from terminal

The message 17800 lost frame(s) 21212/3411 (alaw@8000)-¿(ulaw@8000) indicates that during the transcoding process from alaw to ulaw, a significant number of frames were lost. Frames are units of audio data. Losing frames generally means that some audio information did not get transmitted or processed. But Why Frame Loss did Not degrade Call Quality in a general sense? When 10.3.0.2 called 10.1.0.2, there were issues in both audio and video functionalities although the pjsip config was same for both teams. But with allowing only a few matched codecs between both teams, this issue was resolved. but interesting is this issue didn't happen when 10.3.0.2 called 10.2.0.2. Possible cause of solution could be codecs mismatch which leads to less compatibility or reducing transcoding load.

VI. Discussion

A. MS1

Question1: Check, if your serial router interface is connected on the DCE or DTE connector of the cable by the command show controller serial¡x/y/z¿, where ¡x/y/z¿ is the interface to be tested. Is your R1 serial interface DCE or DTE? In a serial connection between routers, one end acts as Data Communications Equipment (DCE) and the other as Data Terminal Equipment (DTE). The DCE provides the clocking signal, while the DTE receives it.
show controllers serial 0/1/0
This command reveals whether the interface is DCE or DTE. For DCE, clock rate 64000 (max avaailable bitrate) and for DTE, no config is needed. This distinction ensures proper timing and synchronization for data transmission over the serial link.

Question2: Why is it important to disable DNS lookup on a router when configuring it? Disabling DNS lookup on a router prevents the router from trying to resolve mistyped or incorrect command entries as domain names, which can cause delays and unnecessary DNS queries, thereby simplifying and speeding up the command-line interface operations.

B. MS2

Question1: Authentication between caller and called party during Registration and Call setup process: why there is authentication only with the caller but not with the called party?
The key reason we see authentication with the caller but not with the called party during the call setup

is due to the way SIP handles registration and call initiation. The called party has already authenticated with the SIP server during its registration, so additional authentication during the call setup is not required. This streamlines the process and reduces unnecessary authentication overhead during call setups.

Question2: Which IP flow must be filtered in Wireshark, to select the VoIP user data (VoIP samples transmission without any signaling data IP flows) of one VoIP stream? Record with specific filter settings

To filter VoIP user data (RTP streams) without including any signaling data (such as SIP) in Wireshark, we need to use a display filter that specifically captures RTP packets. RTP (Real-Time Transport Protocol) is used for the actual media (audio/video) transmission in VoIP.

Filter settings:

udp or rtp: This filter tells Wireshark to display

Fig. 22. Different filter settings with I/O graph

only the packets that use the UDP protocol and are identified as RTP packets.

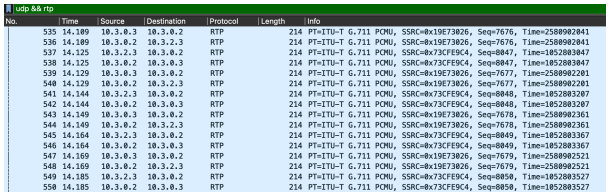


Fig. 23. Identifying VoIP user data

ip.src == 10.3.0.3 or ip.dst == 10.3.0.2 or udp or rtp: To be more specific and filter for a particular stream (RTP) based on IP addresses udp.srcport == 50040 or udp.dstport == 13250 or rtp: port based

Question3: From traffic capture, measure RTP user data throughput for PCMA (alaw) codec. Why is the data link layer throughput higher compared to the application layer throughput? Why is the real Data Link Layer throughput even a little bit higher than the measured throughput of Wireshark capture? Which data are missing in the Wireshark capture?

RTP User Data Throughput Calculation for PCMA (G.711 a-law) Codec:

Total RTP Packets: 576
Duration: 11.50 seconds
Payload Size per Packet: Typically, G.711 (PCMA) payload size is 160 bytes (20 ms of audio).
Total Bytes = RTP Packets * Payload Size
Total Bytes = 576 * 160 = 92160 bytes
Throughput (kbps) = (Total Bytes * 8) / Duration / 1000
Throughput (kbps) = (92160 * 8) / 11.50 / 1000 = 64.14 kbps

Data Link Layer Throughput Calculation:

Ethernet Frame Overhead:
Ethernet header: 14 bytes
IP header: 20 bytes
UDP header: 8 bytes
RTP header: 12 bytes
Frame Check Sequence: 4 bytes
Total Overhead per Packet: 58 bytes
—For each RTP packet:
Total Frame Size = Payload Size + Overhead
Total Frame Size = 160 + 58 = 218 bytes
Total Bytes (Data Link Layer) = RTP Packets * Total Frame Size
Total Bytes (Data Link Layer) = 576 * 218 = 125568 bytes
Data Link Layer Throughput (kbps) = (Total Bytes * 8) / Duration / 1000
Data Link Layer Throughput (kbps) = (125568 * 8) / 11.50 / 1000 = 87.35 kbps

Codec	Sampling Rate (Hz)	Sampling Throughput (kbps)	Captured Layer 2 Throughput (kbps)
PCMA	8000	87.35	64.14

** Why is the Data Link Layer Throughput Higher? Additional Headers: The data link layer throughput includes additional header data from Ethernet, IP, UDP, and RTP layers, FCS. These headers add extra bytes to each packet, increasing the total data transmitted compared to the application layer payload alone.

** Real Data Link Layer Throughput vs. Measured Throughput: Real Throughput Higher: The actual Data Link Layer throughput might be slightly higher than what Wireshark measures because Wireshark does not capture certain elements because of the instruction of the application itself, such as the Ethernet preamble (7 bytes), start frame delimiter (1 byte), the inter-frame gap (12 bytes), FCS (4 bytes) which are all part of the actual transmission but not visible in Wireshark captures.

C. MS3

Question1: How does Asterisk server by default handle NAT with RFC 3581?

This RFC introduces a way for SIP responses to be routed back to the source IP address and port 'rport' parameter from which the request was sent, even if the request was received on a different IP address or port.

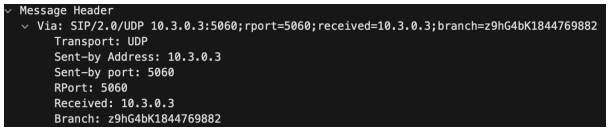


Fig. 24. NAT handling 'rport' with RFC 3581 from SIP header

Question2: Why the call from VoIP phone in LAN-A to VoIP phone in LAN-C is not successful after erasing NAT translations in R2? Provide reasons for each protocol.

VoIP call LAN-A to LAN-C		
Protocol	**Success (yes / no)**	**Description**
SIP signaling traffic	no	SIP protocol is working between caller and server but not between called party and server. The Asterisk server forwarded the INVITE to the VoIP phone in LAN-C. SIP TRYING/RINGING/OK: There are no response messages (TRYING, RINGING, OK) sent back to the server and subsequently to the VoIP phone in LAN-A, since there are no NAT to traverse the network correctly with translations. The Asterisk server and the VoIP clients depend on these translations to correctly route SIP messages between the internal and external networks. SIP ACK: Without the OK message, the ACK message cannot be sent, and the call setup process is not completed.
RTP media traffic	no	RTP media traffic relies on the SIP signaling to establish the media path. If the SIP signaling fails due to the lack of NAT translations, the RTP media stream cannot be established. Media Flow: Even if the initial SIP signaling succeeded by some means, the media flow fails because the NAT translations are required to properly route the RTP packets between the networks. Even if SIP signaling somehow succeeds, the media stream (audio) cannot be established. This results in a call where the participants cannot hear each other, effectively making the call unsuccessful. Here, caller sending the audio codecs in RTP traffic to server, but server is unable to reach the called party.

Question3: How the NAT entries are created automatically after erasing those with 'clear ip nat translation *' command?

The VoIP clients in both network continuously sending INVITE messages to server to initiate communication. When phone in LAN-C initiates a call, the SIP INVITE message is sent to the Asterisk server. As it traverses R2, NAT entries are created or updated. SIP responses like TRYING, RINGING, and OK, along with the ACK message, maintain these NAT entries. Finally, the RTP media stream establishes the media path, creating additional NAT entries for voice data. This automatic creation and maintenance of NAT entries ensure successful communication between both parties.

NAT table in R2 during VoIP call from LAN C to A:

SIP traffic from LAN-C to LAN-A				
P	**Inside Local Address:Port**	**Inside Global Address:Port**	**Outside Local Address:Port**	**Outside Global Address:Port**
UDP	10.3.2.3:5060	10.3.1.2:5062	10.3.0.2:5060	10.3.0.2:5060
TCP	10.3.2.3:54694	10.3.1.2:5064	23.201.240.195:80	23.201.240.195:80
TCP	10.3.2.3:34664	10.3.1.2:5068	2.18.64.13:80	2.18.64.13:80
TCP	10.3.2.3:46089	10.3.1.2:5063	108.177.15.188:5228	108.177.15.188:5228

Question4: Evaluate a VoIP traffic from LAN-C to LAN-A. Check signaling flows. Why the call is successful? What do you understand from debug messages of asterisk terminal?

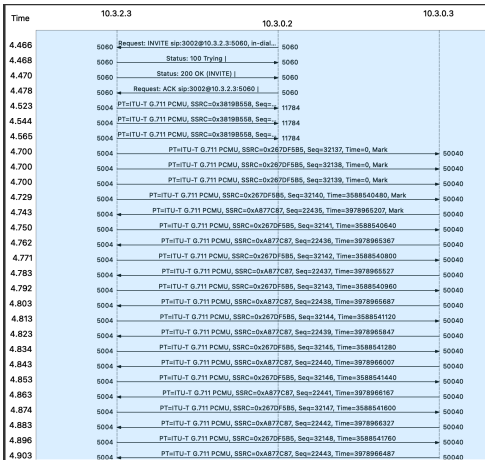


Fig. 25. VoIP traffic from LAN-C to LAN-A in peer-to-peer mode

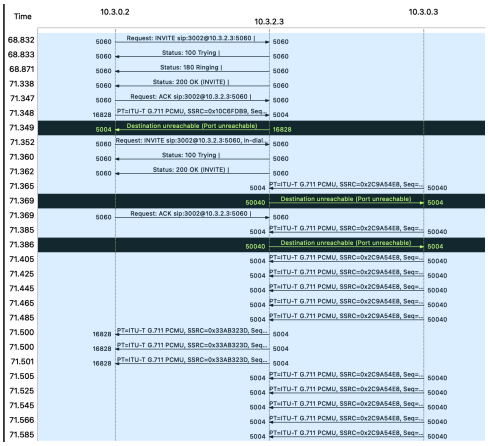


Fig. 26. VoIP traffic from LAN-A to LAN-C in peer-to-peer mode

VoIP call LAN-C to LAN-A		
Protocol	**Success (yes / no)**	**Description**
SIP signaling traffic	Yes	SIP is responsible for call setup, modification, and teardown. The static route in R1 is deleted which blocks routing from LAN A to LAN C, NAT configurations (setting NAT and NAPT interfaces) and ACLs (creating ACL to limit NAT translations) allow SIP messages to traverse correctly between LAN-C and LAN-A.
RTP media traffic	Yes	RTP carries the actual media (voice/video) of the call. NAT translates RTP ports correctly, enabling media stream between the two phones.

Initial call setup:
-- Executing [3002@from-internal:1] Answer("PJSIP/3001-00000012", "") in new stack
> 0x7f67c400c380 -- Strict RTP learning after remote address set to: 10.3.0.3:50040
> 0x7f67c400c380 -- Strict RTP switching to RTP target address 10.3.0.3:50040 as source

Dialing the Called Party:
-- Executing [3002@from-internal:2] Dial("PJSIP/3001-00000012", "PJSIP/3002,20") in new stack
-- Called PJSIP/3002
-- PJSIP/3001-00000012 requested media update control 26, passing it to PJSIP/3002-00000013

Ringing and Answer:
> 0x7f67c400c380 -- Strict RTP learning complete - Locking on source address 10.3.0.3:50040
-- PJSIP/3002-00000013 is ringing
> 0x7f67c4017b70 -- Strict RTP learning after remote address set to: 10.3.1.2:5064
-- PJSIP/3002-00000013 answered PJSIP/3001-00000012

Media Bridging:
-- Channel PJSIP/3002-00000013 joined 'simple_bridge' basic-bridge <a8c60f0f-8b0a-4085-a975-9b40d71ff87e>
-- Channel PJSIP/3001-00000012 joined 'simple_bridge' basic-bridge <a8c60f0f-8b0a-4085-a975-9b40d71ff87e>
> Bridge a8c60f0f-8b0a-4085-a975-9b40d71ff87e: switching from simple_bridge technology to native_rtp
> Remotely bridged 'PJSIP/3001-00000012' and 'PJSIP/3002-00000013' - media will flow directly between them

RTP Source Learning:
> 0x7f67c4017b70 -- Strict RTP learning after remote address set to: 10.3.1.2:5064
> 0x7f67c4017b70 -- Strict RTP switching to RTP target address 10.3.1.2:5064 as source

Call teardown:
-- Channel PJSIP/3001-00000012 left 'native_rtp' basic-bridge <a8c60f0f-8b0a-4085-a975-9b40d71ff87e>
-- Channel PJSIP/3002-00000013 left 'native_rtp' basic-bridge <a8c60f0f-8b0a-4085-a975-9b40d71ff87e>
== Spawn extension (from-internal, 3002, 2) exited non-zero on 'PJSIP/3001-00000012'

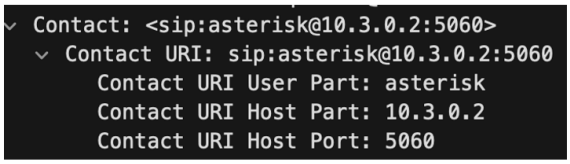
Fig. 27. Asterisk NAT debug information

Question5: Evaluate SIP call setup process behaviour with asterisk in peer-to-peer and proxy mode

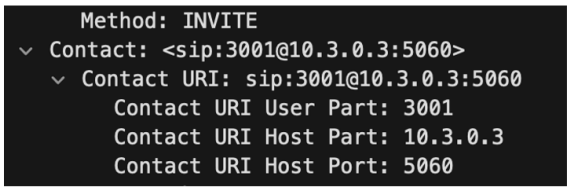
Question6: How can old NAT translations affect the connectivity of endpoints in a VoIP configuration, and how can this be diagnosed using Asterisk commands? Old NAT translations can prevent endpoints from connecting correctly because the router may route traffic based on outdated mappings. This issue can be diagnosed by using the command pjsip show endpoints in Asterisk, which displays the status and connection details of all SIP endpoints, helping to identify connectivity issues related to stale NAT entries.

Aspect	Peer-to-Peer	Proxy Mode
Contact Header	Direct IP addresses of the endpoints.	IP address of the SIP server.
SDP Media Information	Media (RTP) endpoints are the IP addresses of the caller and callee.	Media (RTP) endpoints are the IP address of the SIP server.
Call Setup Messages	INVITE and 200 OK messages will have endpoints' IP addresses directly.	INVITE and 200 OK messages will route through the SIP server, showing the server's IP in the signaling and media.

TABLE IV
IDENTIFYING DIFFERENCES BETWEEN PEER-TO-PEER AND PROXY MODE



Calling from LAN A to C in **proxy** mode



Calling from LAN A to C in **peer-to-peer** mode

Calling from LAN C to A in **peer-to-peer** mode

Fig. 28. Differences in Contact header of INVITE messages in proxy (server) vs peer-to-peer(caller/callee) mode

D. MS4

Question1: Record and analyze SIP messages for mutual registration between call servers during SIP registration process. which devices must register and authenticate with remote VoIP Service Provider.

Both asterisk servers must register and authenticate

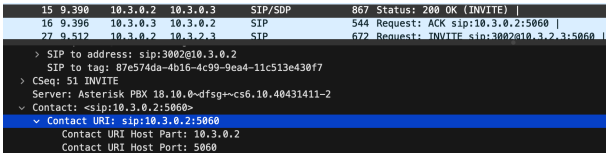


Fig. 29. Differences in Contact header of 200 OK messages in proxy (server) mode

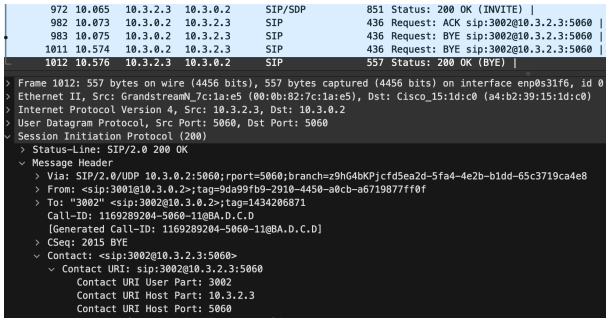


Fig. 30. Differences in Contact header of 200 OK messages in peer-to-peer mode

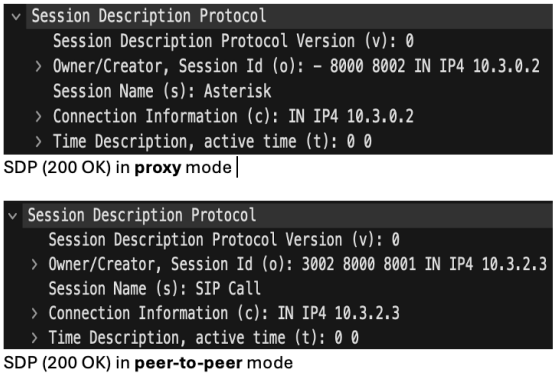


Fig. 31. Differences in 'c=connection information' parameter of SDP section in proxy (server) vs peer-to-peer (caller/callee) mode

with each other using credentials specified in the configuration (username and password) for handling calls for VoIP clients.

Question2: Illustrate SIP bindings between all the servers for an unified VoIP system

Question3: Record VoIP call setup for interconnected services with SIP messages. Which devices must authenticate? Measure call setup latency.

servers 10.3.0.2 and 10.2.0.2 require authentication for registration and call setup.

Call Setup Latency (calling from 10.3.0.2 to 10.2.0.2) = 21.658 21.633 = 0.025s = 25ms

Question3: Evaluate transcoding. Use one single, but different codec at VoIP client A and B with allowing all codecs in both servers.

Asterisk server handle calls between endpoints with different codec capabilities. When a codec mismatch

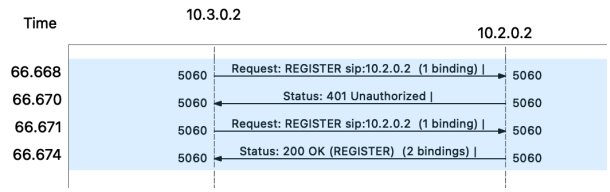


Fig. 32. SIP registration between team 2 and 3 servers

```
[trunk3]
type=identify
match=10.2.0.2
endpoint=trunk3

[trunk3]
type=endpoint
context=from-internal
allow=all
direct_media=no
auth=trunk-auth
outbound_auth=trunk-auth
aors=trunk3

[trunk-auth]
type=auth
auth_type=userpass
username=team2
password=trunk

[trunk3]
type=aor
max_contacts=10
contact=sip:10.2.0.2:5060

[trunk3]
type=registration
outbound_auth=trunk-auth
server_uri=sip:10.2.0.2
client_uri=sip:trunk3@10.3.0.2
```

Fig. 33. Registration and Authentication credentials

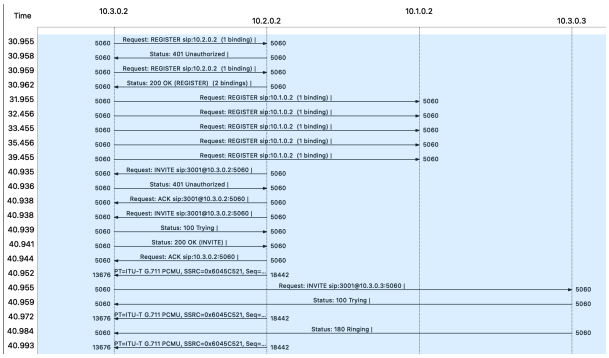


Fig. 34.

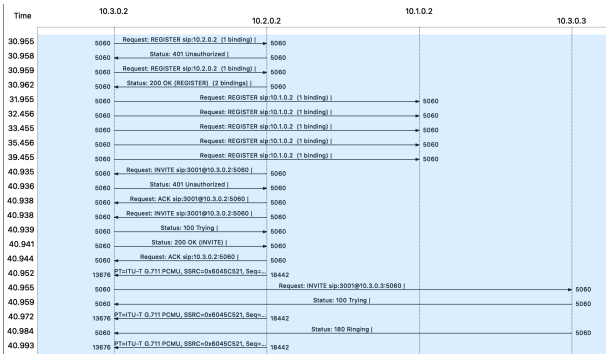


Fig. 35. SIP Registration and Call Setup Flow Between Servers

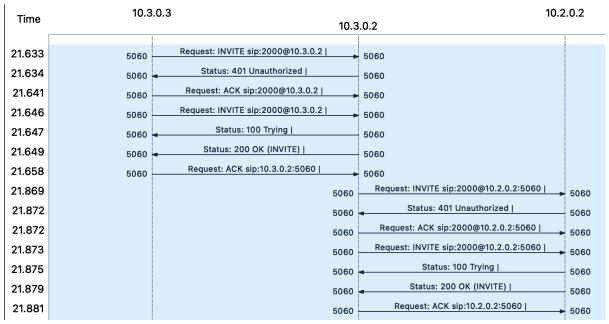


Fig. 36. SIP call setup messages when calling from team 3 to 2

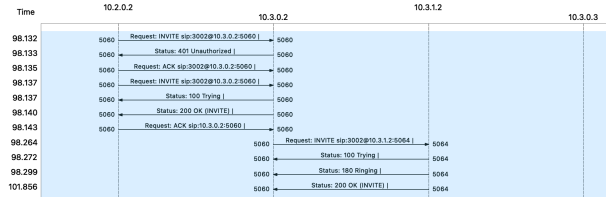


Fig. 37. SIP call setup messages when calling from team 2 to 3

occurs, Asterisk automatically handle the transcoding to ensure a successful end-to-end call. Here, Asterisk detects that 3002 supports only G.711 PCMU (ulaw) while 3001 supports only G.711 PCMA (alaw).

When endpoint 3002 (which only supports G.711 PCMU or ulaw) calls endpoint 3001 (which only supports G.711 PCMA or alaw), and the remote party offers ulaw codec which is not supported by 3001, Asterisk decodes the ulaw RTP stream from 3002's remote party and re-encode it into alaw for endpoint 3001, and vice versa.

Question4: Evaluate RTP flows in case of transcoding. Which device will be responsible to transcode samples?

Transcoding process: From G.711 PCMU to G.722

- 10.2.0.2 receives G.711 PCMU packets from 10.3.0.2
- It decodes these packets into raw audio.
- It re-encodes the raw audio into G.722 packets.
- These G.722 packets are then sent to 10.1.0.2

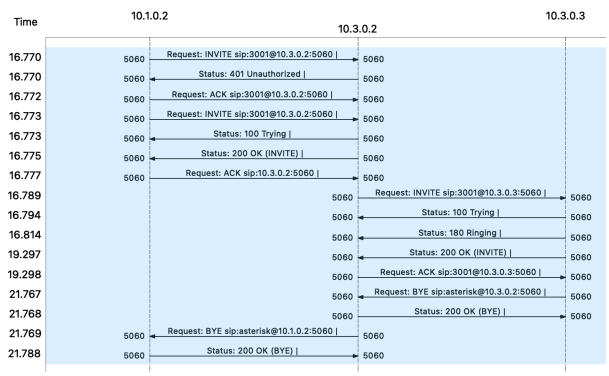


Fig. 38. SIP call setup messages when calling from team 1 to 3

```
[3001]
type=endpoint
context=from-internal
allow=!all,alaw
direct_media=no
auth=3001
aors=3001

[3001]
type=auth
auth_type=userpass
password=password
username=3001

[3001]
type=aor
max_contacts=1

[3002]
type=endpoint
context=from-internal
allow=!all,ulaw
direct_media=no
auth=3002
aors=3002
```

Fig. 39. PJSIP file with one but different codec for endpoints

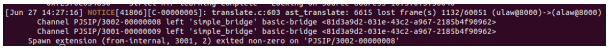


Fig. 40. Transcoding of codecs in Asterisk terminal

Transcoding responsibility The Asterisk servers are responsible for transcoding if the endpoints (VoIP clients) are using different codecs. When one endpoint uses Codec X and the other uses Codec Y, the Asterisk server transcode the media streams (RTP) between the two codecs.

Question5: Describe the transcoding process. What needs to be done to create a successful end-to-end call between phones with different codec capabilities?

- **Initial Negotiation:**

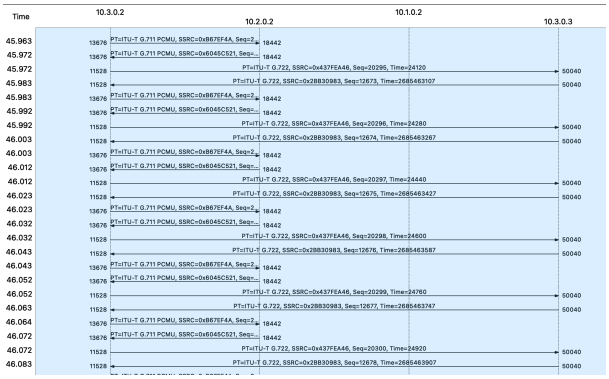


Fig. 41. Evaluation of RTP in transcoding process

  - When a call is initiated, SIP INVITE messages are exchanged between the endpoints through the Asterisk server at `10.2.0.2`.
  - These INVITE messages contain a list of supported codecs from the calling endpoint `10.3.0.2` and the callee endpoint `10.1.0.2`.
- **Codec Selection:**
  - Each endpoint responds with a list of acceptable codecs.
  - If the endpoints `10.3.0.2` and `10.1.0.2` do not have a common codec, the Asterisk server `10.2.0.2` identifies the need for transcoding.
- **Transcoding Setup:**
  - The Asterisk server `10.2.0.2` sets up two separate media streams: one between VoIP Client A (`10.3.0.2`) and itself using Codec X (e.g., G.711 PCMU), and another between VoIP Client B (`10.1.0.2`) and itself using Codec Y (e.g., G.722).
- **RTP Flow:**
  - RTP packets are sent from VoIP Client A (`10.3.0.2`) to the Asterisk server (`10.2.0.2`) using Codec X.
  - The Asterisk server decodes these packets into raw audio.
- **Conversion:**
  - The raw audio is then encoded into Codec Y by the Asterisk server (`10.2.0.2`).
- **Forwarding:**
  - The newly encoded RTP packets (now in Codec Y) are sent from the Asterisk server (`10.2.0.2`) to VoIP Client B (`10.1.0.2`).

Transcoding is CPU intensive and introduced slight delays and potentially reduced call quality. ulaw (G.711 μ-law) is typically used in regions where bandwidth is not a constraint and high call quality is desired. g729 is a compressed codec, suitable for conserving bandwidth, but it requires more processing power for encoding and decoding. Transcoding between ulaw and g729 introduce artifacts or degrade call quality compared to direct ulaw to ulaw or g729 to g729 calls.

Question6: Why Frame Loss Might Not Degrade Call Quality? Bandwidth and Network Capacity: Bandwidth refers to the maximum rate at which data can be transferred over a network. An 8 Mbps (megabits per second) connection is quite high for audio calls, which typically require much less bandwidth. For example, the ulaw and alaw codecs each use approximately 64 kbps (kilobits per second) for audio data. Even if we consider overhead, each call might use around 80 kbps. With 8 Mbps bandwidth, we can theoretically handle multiple simultaneous calls without significant bandwidth-related issues.
Techniques like Packet Loss Concealment, jitter buffers, and the inherent resilience of ulaw and alaw codecs help maintain audio quality even when some frames are lost during transcoding.
Question7: Can you analyze transcoding and other impportant parameters in a RTP stream from a capture?

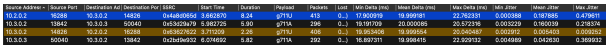


Fig. 42. Illustration of a RTP stream with parameters

Source Address	Destination Address	Codec
10.2.0.2	10.3.0.2	g711U (G.711 μ-law)
10.3.0.2	10.2.0.2	g711U (G.711 μ-law)
10.3.0.2	10.3.0.3	g711A (G.711 A-law)
10.3.0.3	10.3.0.2	g711A (G.711 A-law)

Question8: What do you think about A Man-in-the-Middle (MITM) attack on RTP streams is indeed a potential risk in VoIP communications? Unencrypted RTP Traffic: If RTP traffic is not encrypted, it is sent as plain text over the network. An attacker with access to the network can capture these RTP packets using tools like Wireshark.
Packet Sniffing: Using packet-sniffing tools, an attacker can intercept and capture the RTP streams. This captured data includes the audio packets, which can be reconstructed to retrieve the audio conversation.
Lack of Authentication: Without proper authentication mechanisms in place, attackers can impersonate one of the endpoints, intercepting and potentially altering the RTP stream.
Session Hijacking: An attacker can hijack an ongoing session by injecting malicious RTP packets, which can disrupt the conversation or insert unwanted audio.

Mitigation Measures:
Encryption: Using Secure RTP (SRTP) encrypts the RTP packets, making it significantly harder for attackers to interpret the captured data.
Authentication and Integrity: Implementing strong authentication mechanisms and ensuring

the integrity of the RTP stream can prevent impersonation and tampering.

Question9: Is it possible to achieve transcoding in Peer-to-Peer mode? Transcoding is generally not feasible in pure peer-to-peer mode because the media path does not pass through a central server that can perform the codec conversion. However, with a hybrid approach (where the initial call setup is in peer-to-peer mode, but if the need for transcoding is detected, the call can be re-routed through the Asterisk server (proxy mode) to handle the transcoding.) or advanced endpoints, it might be possible.

Question10: Analayze RTP streams of matched codecs and non-matched codecs audios. Is it possible to gather offsets and then further examine and compare the quality of calls to drag a conclusive comparison?

Understanding Offset and Peaks: Offset: In the context of RTP streams, offset refers to the difference between the expected arrival time of a packet and its actual arrival time. Measuring the offset can help diagnose network issues such as latency or jitter.

Peaks and Lows: Peaks: High points in the amplitude represent loud audio signals. Lows: Low points represent soft audio signals or silence.

parameters for examining audio quality:

- Amplitude Variations: Amplitude/Signal Strength: The vertical axis usually represents the amplitude of the audio signal, showing how loud or soft the signal is. Time: The horizontal axis represents time, indicating when each RTP packet was received or expected.
- Jitter Drops: Red Circles: These indicate jitter drops. Jitter occurs when there's variability in packet arrival times. High jitter can lead to drops or delay variation, which can degrade audio quality.
- Wrong Timestamps: Blue Diamonds: These indicate packets that have incorrect timestamps. Timestamps are crucial for synchronizing audio data. Wrong timestamps can lead to misalignment in audio playback.
- Inserted Silence: Yellow Triangles: These indicate periods where silence was inserted, possibly to mask lost or excessively delayed packets. Silence insertion helps to smooth out the audio but can result in noticeable gaps.

Transcodec call vs Matched codec call:

- Jitter Drops: Multiple instances of jitter drops are present, indicating variations in packet arrival times.
- Wrong Timestamps: There are several occurrences of wrong timestamps, which can cause synchronization issues.

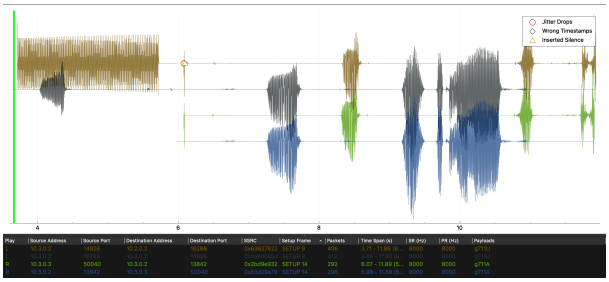


Fig. 43. Illustration of a RTP stream graph with transcoded

- Inserted Silence: Noticeable periods of inserted silence, resulting in gaps in the audio stream.
- Amplitude Variations: Significant variations in amplitude, indicating unstable audio quality.

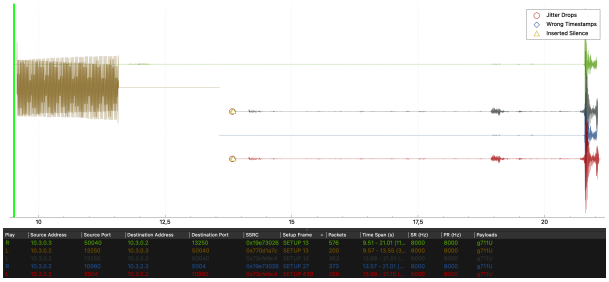


Fig. 44. Illustration of a RTP stream graph with matched codec in proxy mode

- Jitter Drops: Far fewer instances of jitter drops compared to the transcoded call.
- Wrong Timestamps: Minimal occurrences of wrong timestamps, suggesting better synchronization.
- Inserted Silence: Fewer periods of inserted silence, leading to more continuous audio.
- Amplitude Stability: More stable amplitude, indicating consistent audio quality.

Fig. 45. Illustration of a RTP stream graph with matched codec in peer-to-peer mode

- Jitter Drops: Few jitter drops, similar to the matched codec scenario in proxy mode.
- Wrong Timestamps: Minimal wrong timestamps.

– Inserted Silence: Some inserted silence, potentially due to direct communication.
– Amplitude Stability: More stable amplitude but with noticeable silence periods.

Conclusion of the comparisons

– Transcoding introduces significant issues such as jitter, synchronization problems, and audio gaps, degrading call quality.
– Matched Codecs (Proxy Mode) offer the best quality with minimal issues.
– Peer-to-Peer communication provides good quality but can have some inherent issues due to the lack of intermediary management.

Question11: How can you interpret a conference calling between three teams?

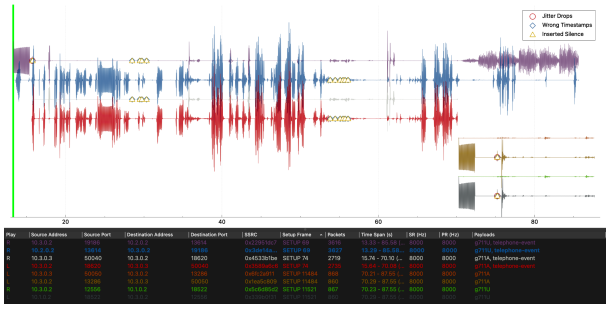


Fig. 46. Illustration of a RTP stream graph in conference style

– Jitter Drops: Significant jitter drops (red markers) are present, indicating packets arriving out of order or with variable delays. This is common in multi-party conference calls due to increased network complexity and potential congestion.
– Inserted Silence: Yellow markers show inserted silences, which can be due to missing or delayed packets. These silences can cause gaps in the audio, making it difficult to follow the conversation.
– Multiple Streams: The table shows multiple RTP streams with different source and destination addresses, ports, and codecs (G.711U and G.711A). This highlights the complexity of managing a conference call involving multiple endpoints.
– Payload Types: The presence of g711U and g711A payload types indicates that some form of transcoding took place between the endpoints, which can introduce additional latency and potential quality issues.

REFERENCES

[1] P. Anderson and M. Giles, *Mastering Asterisk: The Open Source Telephony Platform*. Packt Publishing, 2020.
[2] T. Szigeti, C. Hattingh, R. Barton, and K. Briley, *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs*. Cisco Press, 2019.
[3] K. Egevang and P. Francis, "The ip network address translator (nat)," RFC 1631, 1994. [Online]. Available: https://tools.ietf.org/html/rfc1631
[4] J. Yin, L. Zhang, and Y. Li, "Enhanced nat traversal techniques for real-time communication applications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 567–588, 2021.
[5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, and M. Handley, "Sip: Session initiation protocol," RFC 3261, 2002. [Online]. Available: https://tools.ietf.org/html/rfc3261
[6] S. Lee, J. Park, and H. Kim, "Interoperability challenges in interconnected voip networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 45–58, 2022.