# Design of PSTN-VoIP Gateway with inbuilt PBX & SIP extensions for wireless medium

PRIYESH WADHWA

Under the guidance of
**Prof. Sridhar Iyer**

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

July 16, 2007

# Table of contents

## Outline


1. Introduction
   - Motivation
   - Problem Statement
2. Gateway with Asterisk
   - Software Components
   - Hardware Components
   - Experiments
3. SigComp (Signaling Compression)
   - Implementation Details
   - Integration with Asterisk & Yate
   - Experiments and Results
4. Conclusion & Future Work

# Motivation

- The current setup for integrating PSTN-VoIP system requires PBX server installed on computer and a gateway.
- This solution is costly(approx. Rs.15000), high power consuming, and the setup is quite involved.
- Not suitable for rural environment.

- SIP being a text-based protocol, is engineered for high data rate links.
- On wireless links the packet drop probability of large message size is more.
- In wireless medium, the response time of Asterisk PBX server is much more than in wired medium(around 13 times).

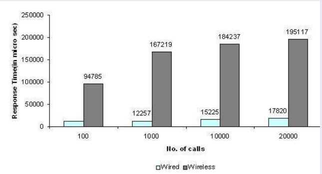



Figure: Response time of Asterisk in wireless and wired medium

# Problem Statement

## SBPVIS: Single Box PSTN-VoIP integrated system

Design a single box solution that integrates the functionality of the Asterisk PBX as well as the gateway. We aim to reduce the cost, power consumption and the intricacies of the system setup.

## SIP in wireless medium

Make SIP more efficient in wireless medium, and to implement these extended features in Asterisk server. Aim is to improve the Asterisk response time in wireless medium.

# Problem Statement

## SBPVIS: Single Box PSTN-VoIP integrated system

Design a single box solution that integrates the functionality of the Asterisk PBX as well as the gateway. We aim to reduce the cost, power consumption and the intricacies of the system setup.

## SIP in wireless medium

Make SIP more efficient in wireless medium, and to implement these extended features in Asterisk server. Aim is to improve the Asterisk response time in wireless medium.

# Software components

## Asterisk

- Open source software PBX system.
- Asterisk gives us connectivity for both PSTN and VoIP networks.
- Provides channels for communication on different hardwares, protocols(SIP), and codecs.

## Yate

- Yate is an open source soft phone which can be used as VoIP client.
- Yate provides many modules like 'callgen', and 'message sniffer' for measuring the performance of the PBX server.

# Hardware components

# Experiment 1: Sipura SPA3000 with Normal PC

Asterisk server is installed and configured on the computer system. SPA3000 is a gateway that enables PSTN-VoIP integration. SPA3000 needs to be configured to work along with Asterisk on the network.

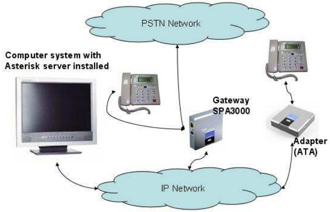



Figure: Conventional setup of Asterisk system

*Advantages*:

- This setup is easy to install.
- Sipura provides a nice web interface for its configuration.
- SPA3000 provides us the facility for fine tuning the system.

*Disadvantages*:

- This setup is the most expensive in terms of cost and power consumption.
- Asterisk server is installed on a computer system, causing wastage of computing resources.

# Experiment 2: Sipura SPA300 with Via motherboard

Aim: To improve the computational resource utilization and reduce the cost.

We replaced the processing unit with the Via motherboard. Via motherboard is inexpensive and consumes less power.

*Advantages*:

- Efficient usage of computational resources.
- Cost of the system is reduced by using Via motherboard(reduced to Rs.7030).

*Disadvantages*:

- The power consumption of the setup is still high.
- Cost of SPA3000 is still high, compared to Digium X100P card.

# Experiment 3: Digium X100P with Via motherboard

Aim: To reduce the cost of Gateway unit.

We replaced the SPA3000 with the Digium X100P PCI card. The Digium card provides the functionality of the gateway, however we cannot fine tune it like the SPA3000.

*Advantages*:

- This setup requires no extra effort to configure the gateway. Asterisk provides us the Zaptel drivers to communicate with Digium card. We just need to configure the zaptel.conf file to make the communication possible.

*Disadvantages*:

- The X100P card provides only the functionality of FXO and FXS ports. No fine tuning of the system is possible unlike SPA3000.

# Experiment 4: Normal Data modem with Via motherboard

Aim: To reduce the cost of PSTN-VoIP interface card.

We used the normal data modem instead of the Digium card. This requires some code modification in the Asterisk's Zaptel driver's code. The normal data modem provides us the FXO and FXS ports just like X100P after the code modification in Asterisk.

*Advantages*:

- Cost of the system is reduced by the use of data modem(Rs.6000).

*Disadvantages*:

- Code modification in Asterisk is required to make Asterisk work with the modem.
- Power consumption of the system is still high, because of the use of hard disk.

# Final Solution: Flash memory with Via motherboard

Aim: To find a replacement for hard disk.

We replaced the hard-disk of the system with a 40-pin flash IDE. Flash IDE is just like a hard disk that is connected to the motherboard on its 40-pin slot used to connect hard-disk data bus. We used AstLinux as our platform for the system.

*Advantages*:

- This setup makes efficient utilization of resource.
- The setup is low power consuming and less costly.

*Disadvantages*:

- The life time of the system is reduced because of the use of flash memory.
- Data retrieval/storage is slow in flash memory.
- We need to make code modifications in Linux and Asterisk to stop the logging.

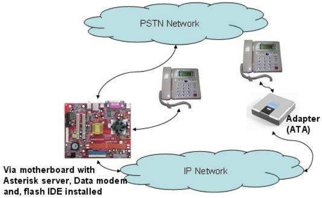



Figure: Improved setup of Asterisk system

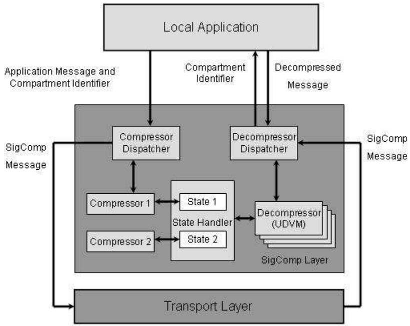
# Introduction

**SigComp** defines the mechanism to compress and decompress the SIP messages in end-to-end VoIP applications. Using SigComp we have obtained a compression ratio between 1:5 and 1:8. The important thing about SigComp is that it is totally independent of compression algorithm used.

## SigComp Architecture

The major components of SigComp are:

- Compressor Dispatcher
- Decompressor Dispatcher
- Compressor
- Decompressor
- State Handler



# Algorithm: Deflate Compression

- combination of the LZ77 algorithm and Huffman coding.
- replaces duplicate occurrence of strings in the input data with pointer[(distance, length)] to previous occurrence.
- distance, length, and literals are encoded using Huffman trees.

## Example

Deflate Compression

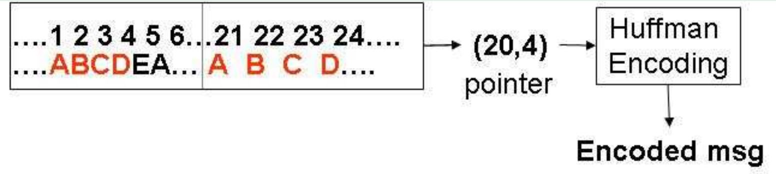
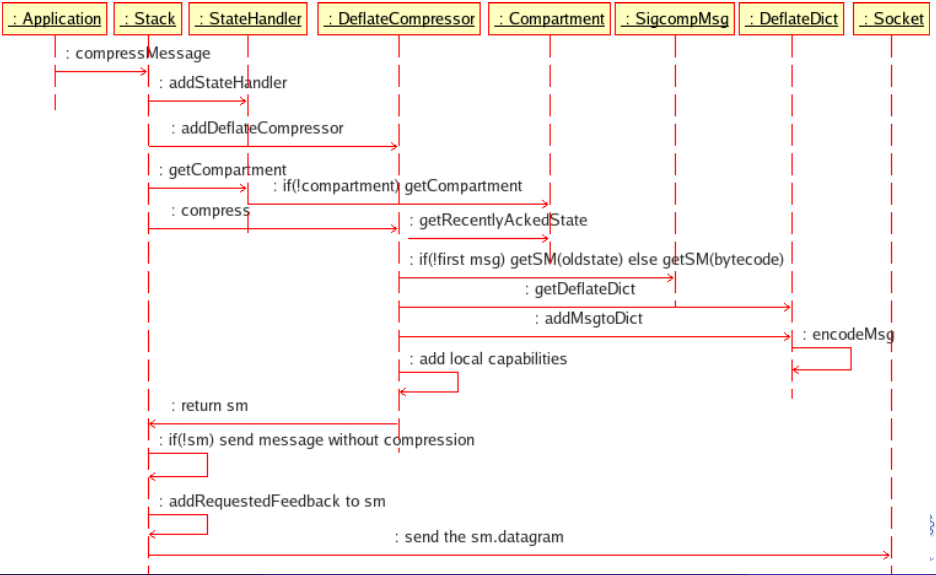
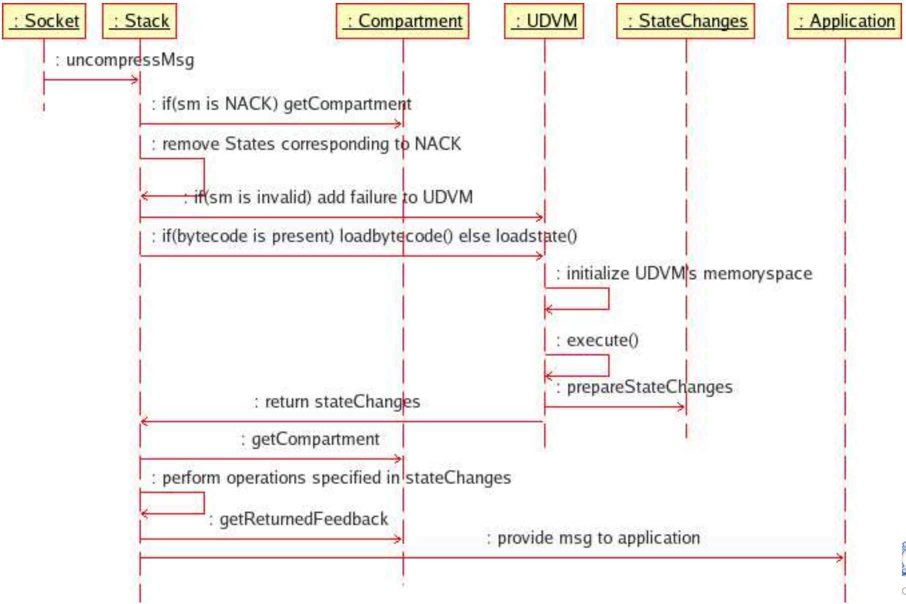



Figure: Deflate Compression

# Control Flow: Compression



# Control Flow: Decompression



# Integration with Asterisk & Yate

## Integration with Asterisk

- We have integrated SigComp in Asterisk's SIP channel.
- Made transmission & reception code thread-safe.
- Compression in: `__sip_transmit()` method and decompression in: `sipsock_read()` method.

## Integration with Yate

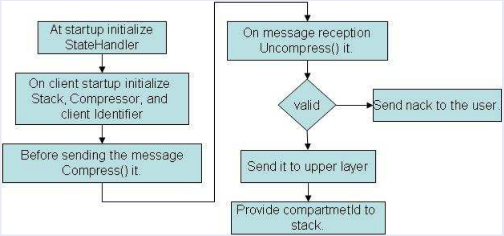



Figure: SigComp integration with Yate

# SigComp: Experiments and Results

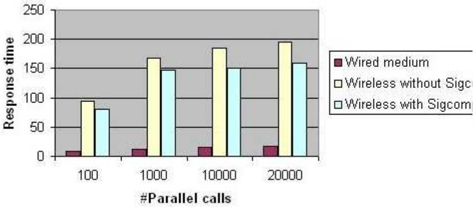



Figure: Improvement in Asterisk's response time with compression.

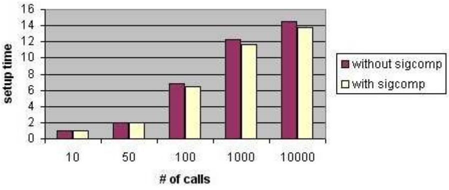



Figure: SIP to SIP connection improvement.

# SigComp: Experiments and Results

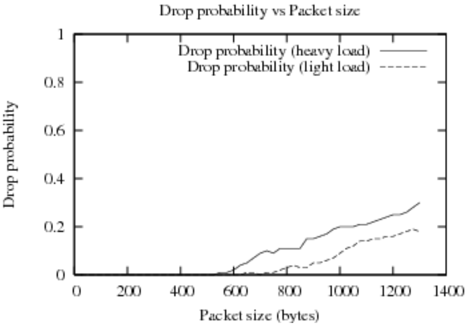



Figure: Packet drop probability vs Packet size for different bandwidths.

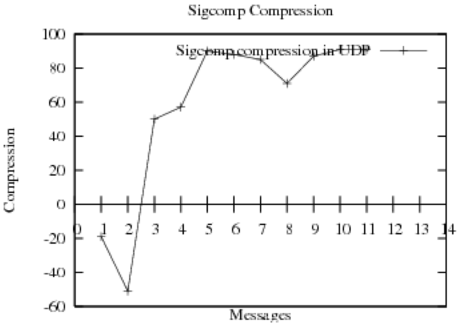



Figure: UDP: Deflate compression (Compression ratio vs Packet sequence number)

# SigComp: Pros and Cons

## Advantages

- Independent of compression algorithm.
- High compression ratio.

## Disadvantages

- Computational overhead.
- Initial message compression gives negative compression.

# Unfinished Idea

## Data storage on Edge Proxy

- Utilize repetition of the same content transmission in consecutive SIP message.
- Store the call profile information & reconstruct the message on the edge proxy.

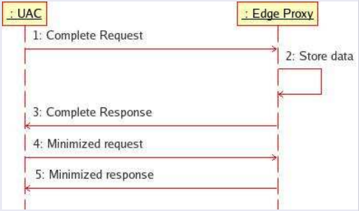



Figure: Data storage on edge proxy

# Conclusion and Future work

## Conclusion

- The single box solution is inexpensive, easy to install, robust and, is a low power consuming device. It is the best we can get out by using off-the-shelf components.
- Using SigComp we have achieved a compression of about 90% in the SIP messages.
- We have improved Asterisk's response time by about 10-15%. We have also reduced the session establishment time in direct SIP-to-SIP calls.

# Future work

## Future Work

- We have planned to integrate the SigComp implementation with the main branch of Asterisk's open source code repository.
- Implementation of data storage on Edge Proxy mechanism
- Hardware implementation of the single box solution
- More compression algorithms
- SigComp implementation for 3GPP2 IMS project

# Bibliography I


- J. V. Meggelen, J. Smith, and L. Madsen, *Asterisk The Future of Telephony*. O'Reilly, August 15, 2007.
- M. Spencer, M. Allison, and C. Rhodes, *The Asterisk Handbook*. Asterisk Documentation Team, 2003.
- R. Price, S. Manor, C. Bormann, T. Bremen, J. Christoffersson, H. Hannu, Z. Liu, and J. Rosenberg, *Signaling Compression (SigComp)*. RFC 3320, January 2003. [Online]. Available: `http://www.rfc-editor.org/rfc/rfc3320.txt`
- R. Balbinot, J. Guedes-Silveira, and P. Franco, *Desktop Voice Over IP Development with PSTN integration*, 2001.

# Thank You!