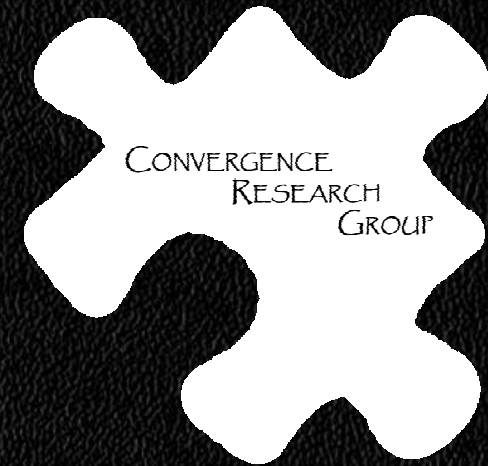


sponsored by



RHODES UNIVERSITY
Where leaders learn



Investigating Standardisation Progress of Peer-to-Peer Session Initiation Protocol and Evaluating Existing Systems

By: Erasmus Tyapa

Supervisor: Prof. Alfredo Terzoli and Mr. Mosiuoa Tsietsi

Outline

- Objectives
 - SIP in summary
 - P2PSIP
 - Distribution Model
 - Protocol Layering
 - DHT Choice
 - P2PSIP Tested Systems
 - Candidate systems for use
 - Conclusion
- } Key Issues for
standardisation

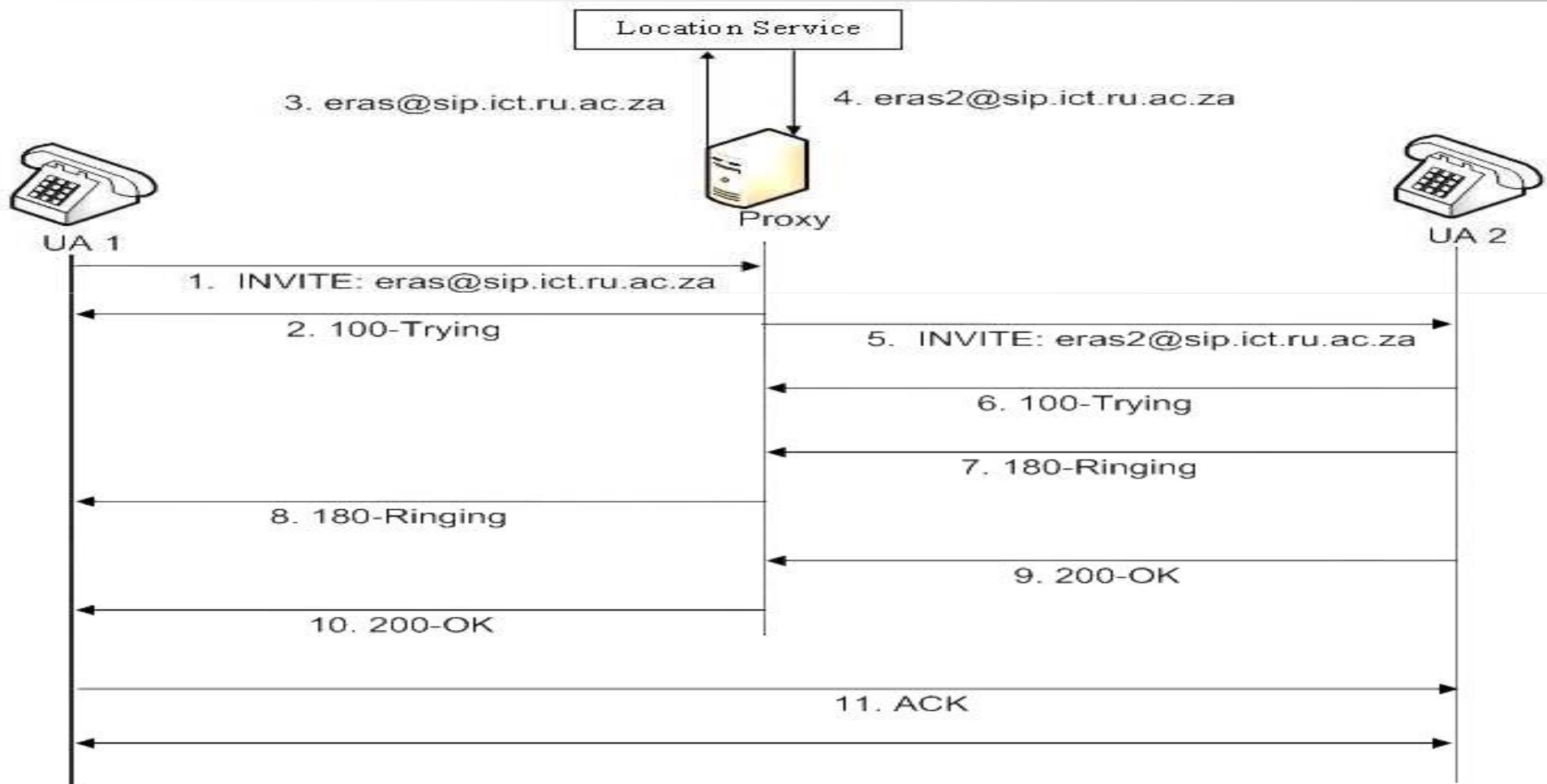


The objective of the project

- Investigate the standardisation progress of P2PSIP.
- Test P2PSIP implemented systems and identify which one is appropriate for further research use.

Session Initiation Protocol

A signalling protocol used to establish, modify and tear down session over IP network.



SIP Architecture

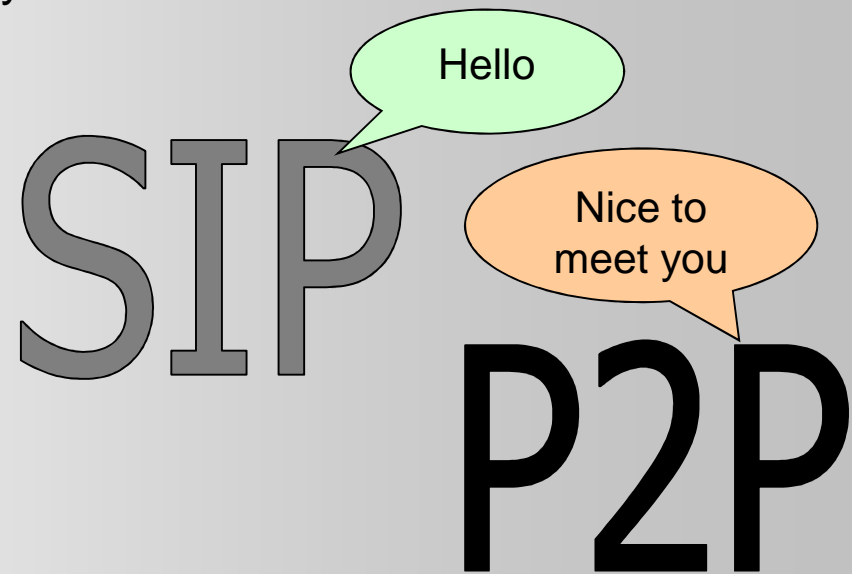
- Based on client-server and consists of user agents, proxies and registrars
 - Proxies and registrars are only a mechanism to find an AOR
 - Proxies only route SIP messages (exercise almost no control)
 - User agents have usually more intelligence than SIP proxies

- Problems with client-server architectural model
 - Proxies and registrars create single point of failure
 - Limited scalability
 - High cost for maintenance

**Overall suitability
for emergency
scenarios?**

Uniting P2P and SIP

- P2P uses direct client-to-client communication.
- To build a P2P network that uses SIP, user agents need lookup and routing functionalities. That is, a DHT overlay needs to be built in the user agents to provide them with **join**, **leave** and **lookup** capabilities.
- SIP can allow this with little changes BUT there is a need to agree on:
 - Distribution Model: “Roles” played by nodes
 - DHT algorithms to use
 - SIP-using-P2P vs. P2P-over-SIP



Distribution Model in P2PSIP

How nodes are distributed in order to carryout functions of central servers.

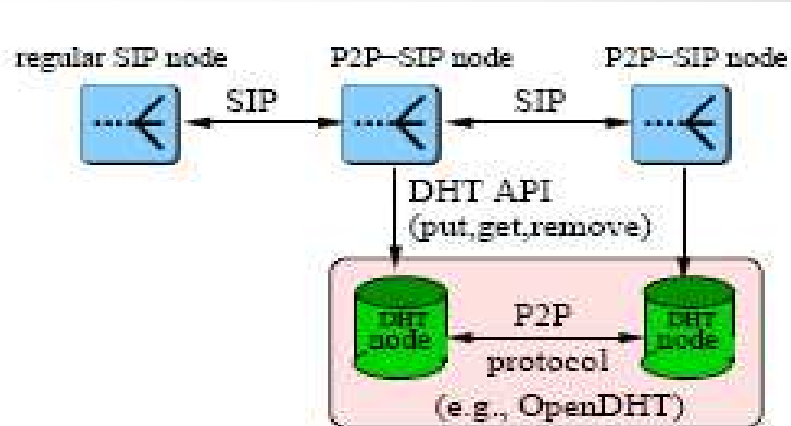
- Pure P2P model

- All peers are equal
- Any peer can carryout the functions of the removed servers

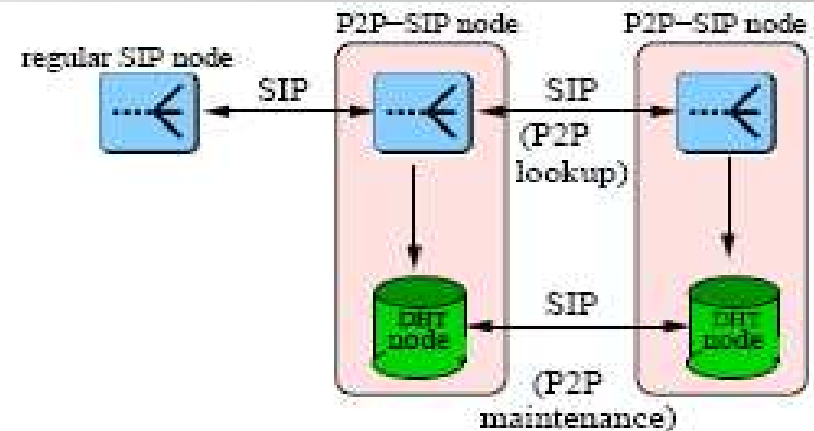
- Partial P2P model

- P2PSIP Peers: It is a node participating in a P2PSIP overlay that provides storage and routing services to other nodes in the same P2PSIP overlay.
- P2PSIP Clients: It is a node participating in a P2PSIP overlay that does not store resources, run the distributed database algorithm, and is not involved in routing messages to other peers or clients.

Protocol Layering



(a) SIP-using-P2P



(b) P2P-over-SIP

Architecture approach (from [Kundan PHD thesis])

SIP layer for registering users, resource lookup, establishing session. a P2P layer for maintaining a distributed network.

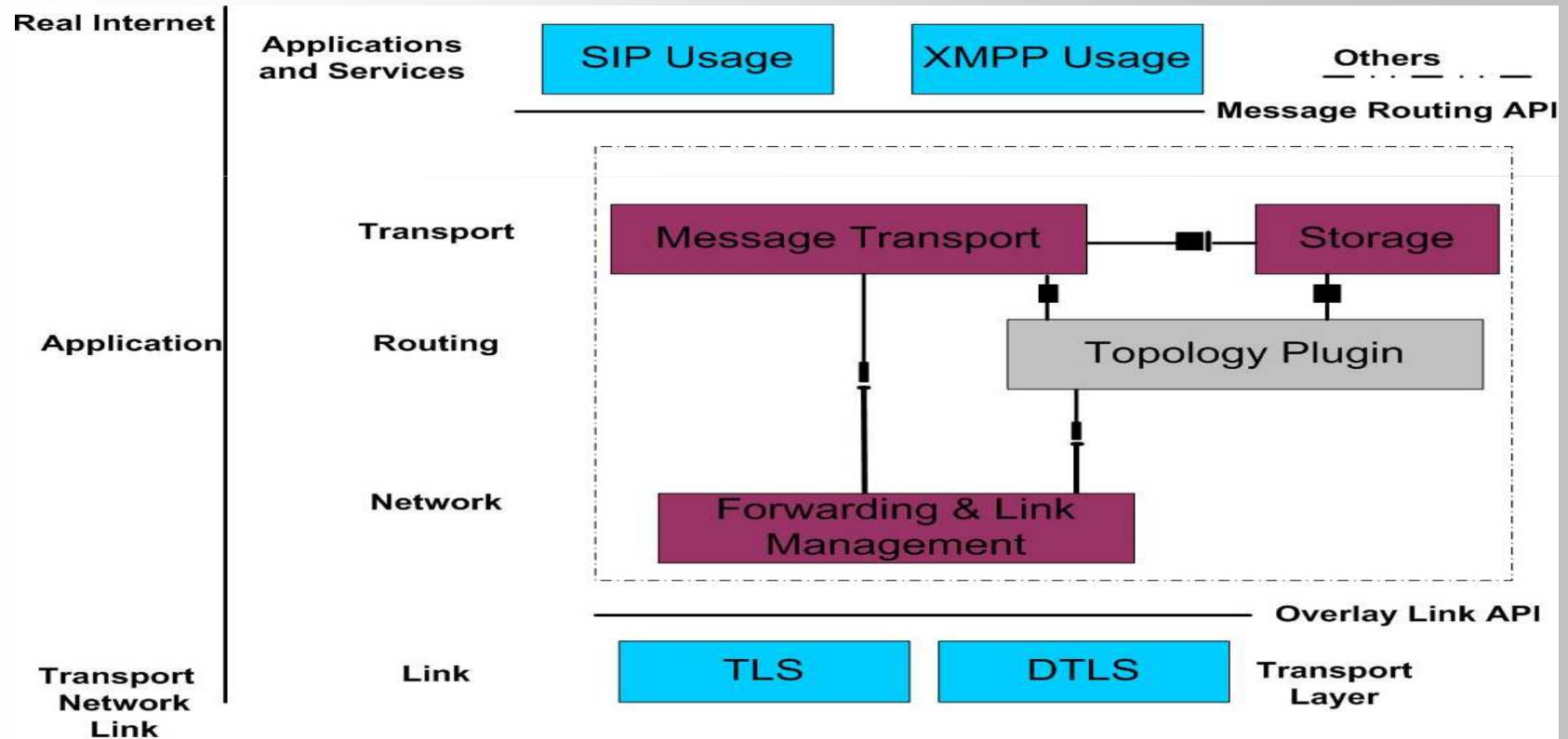
SIP messages are not used only for registering users, resource lookup, and establishing session but also for maintaining a P2P network. Overloading SIP causing high overhead, interoperable with other P2P applications

DHT Choice

- Distributed Hash Tables (DHTs) is a class of decentralized distributed system that provides a lookup service.
- Commonly used DHTs:
 - Pastry
 - CAN
 - Bamboo
 - Chord
 - Kademlia
- Single vs. Multiple DHTs
 - Is it mandatory to implement DHT, or do we just specify one to use for testing?
 - Best approach

RELOAD

RELOAD is a P2P version of SIP. Adapted by the P2PSIP Working Group as its starting point for the primary P2PSIP protocol.



Open Issues

- The P2PSIP IETF is working on improving the RELOAD protocol
- Security
 - Connection level: Connections between peers are secured with TLS or DTLS.
 - Message Level: Each RELOAD message must be signed.
 - Object level: Stored objects must be signed by the storing peer.
- NAT ISSUES
- RELOAD support of Video on Demand

P2PSIP Tested Systems

| | OverCord | 39 Peers | SIP2P | OpenVoIP |
|------------------|-----------------------|-----------------|-------------------------|----------------------------|
| Language | Java | Python | C++ | C++ |
| NAT support | No | Yes | No | Yes |
| Peers and Client | No | Yes | No | Yes |
| Complete | Yes | No | Yes | Yes |
| DHT | Bamboo and Open Chord | Kademlia | Kademlia | Kademlia, Bamboo and Chord |
| Hash Algorithm | None | SHA1 | None | SHA1, MD1 |
| Routing Methods | Recursive | Recursive | Recursive and Iterative | Recursive |

OverCord

- Java implemented systems, developed at Rhodes University as a part of Master's thesis.
- Jain SIP Applet Phone

```
### Finished overlay discovery and bootstrapping module ... ###
```

```
### Node instantiation ... ###
```

```
Using plugin: chordplugin.ChordPlugin
```

```
Starting: 146.231.123.76 as bootstrap peer...
```

```
Creating overlay: chord.ru.ac.za
```

```
Booting parameters: 146.231.123.76 0
```

```
Running node on 146.231.123.76 on port 3730
```

```
[main] log4 configured with 'log4j.properties'
```

```
***** Node options *****
```

1. Insert record
 2. Retrieve record
 3. Remove record
 4. Leave overlay
- Your value: [0]

```
--- Initialising ---
```

```
--- Looking for installed plugins ...
```

```
2 plugin(s) loaded. Names are:
```

```
chordplugin.ChordPlugin
```

```
bambooplugin.BambooPlugin
```

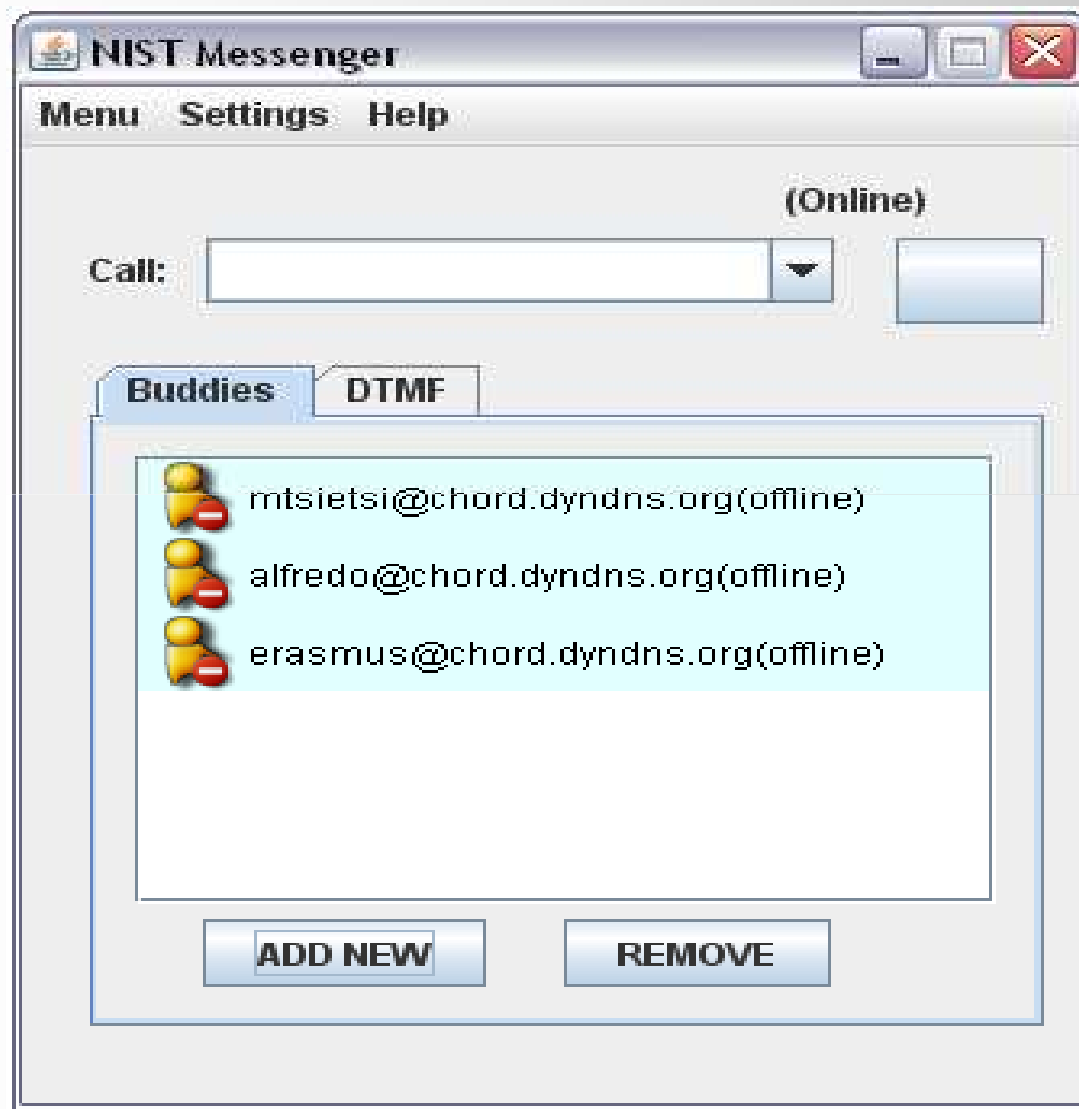
```
--- You can start an overlay with any of the following :
```

```
--- > > 1. chordplugin.ChordPlugin
```

```
--- > > 2. bambooplugin.BambooPlugin
```

```
--- Your choice:
```


Jain SIP P2P Applet Phone



SIP2P

- SIP2P is P2PSIP proxy Published at source forge (<http://sourceforge.net/projects/sip2p/>)
- Based on early proposal candidate (Peer-to-Peer Protocol) for P2PSIP.
- Running on Linux machine
- Using X-Lite running on any machine.
- Not incorporated into a user agent, the ideal way is to incorporate SIP2P into a user agent.
- X-Lite

The screenshot shows the 'SIP Account' configuration window with the following details:

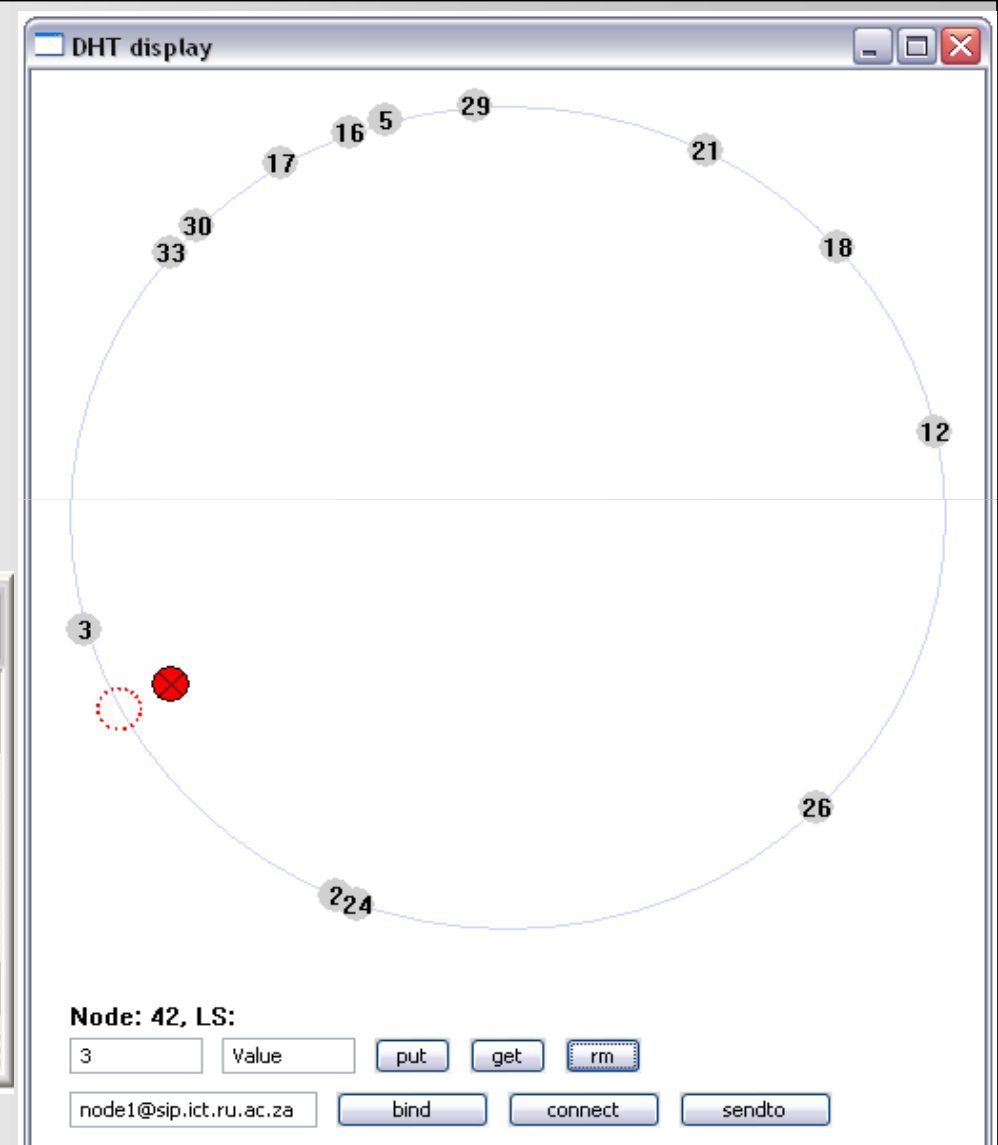
- Account name:** erasmus
- Protocol:** SIP
- User Details:**
 - User ID:** TyapaE
 - Domain:** 146.231.123.70
 - Password:** (empty)
 - Display name:** Tyapa
 - Authorization name:** (empty)
- Domain Proxy:**
 - ☒ Register with domain and receive calls
 - Send outbound via:
 - ☐ Domain
 - ☒ Proxy Address: 146.231.123.70
- Dial plan:** #1/a/a.T;match=1;prestrip=2;

Buttons: OK, Cancel

Candidate Systems for Use

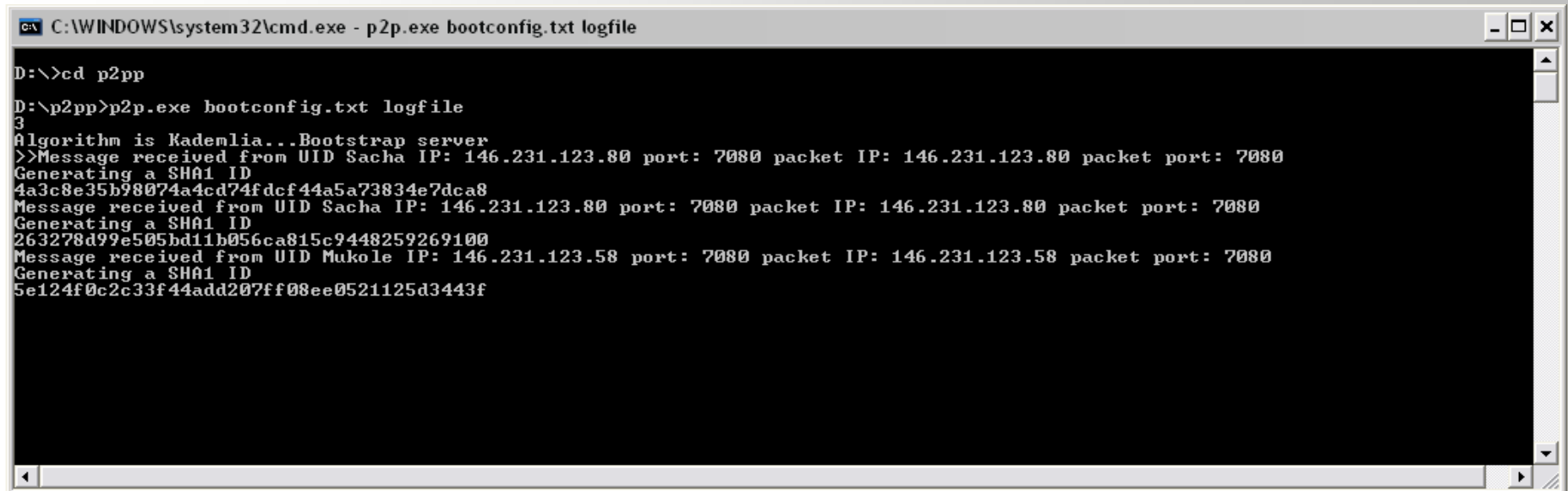
- 39 Peers (<http://39peers.net/>)
 - Incomplete
 - Work like an adaptor running locally
 - Implemented user agent class to allow user agents to route calls
 - Support NAT using STUN, TURN

```
C:\WINDOWS\system32\cmd.exe - python ap...  
C:\p2p-sip\src>python app\dhtgui.py  
put Key, Value  
connect node1@sip.ict.ru.ac.za  
sendto node1@sip.ict.ru.ac.za Value  
connect erasmus@sip.ict.ru.ac.za  
sendto erasmus@sip.ict.ru.ac.za Value  
-
```



Candidate Systems for use

- OpenVoIP (Columbia University)
 - Borrowed code to create our own P2P overlay.
 - Created a bootstrap server running on a local machine.
 - Users can join as client or as peers
 - Our system does not support NAT
 - Users use OpenWengo-P2P phone



```
C:\WINDOWS\system32\cmd.exe - p2p.exe bootconfig.txt logfile

D:\>cd p2pp
D:\p2pp>p2p.exe bootconfig.txt logfile
3
Algorithm is Kademlia...Bootstrap server
>>Message received from UID Sacha IP: 146.231.123.80 port: 7080 packet IP: 146.231.123.80 packet port: 7080
Generating a SHA1 ID
4a3c8e35b98074a4cd74fdcf44a5a73834e7dca8
Message received from UID Sacha IP: 146.231.123.80 port: 7080 packet IP: 146.231.123.80 packet port: 7080
Generating a SHA1 ID
263278d99e505bd11b056ca815c9448259269100
Message received from UID Mukole IP: 146.231.123.58 port: 7080 packet IP: 146.231.123.58 packet port: 7080
Generating a SHA1 ID
5e124f0c2c33f44add207ff08ee0521125d3443f
```

Conclusion

- P2PSIP has been put in practice in Industry (e.g. Damaka) and Academia (Columbia University).
- There is lack of implementations .
- Project under development and lack of technical documentation
- Lot of bugs
- Security and NAT issues need to be addressed.



■ Thank you for you attention

■ Questions?