Assignment-3
Dial-a-Song
Assignment Motivation

- Understanding SIP protocol (in a practical way)
  - Questions like: why certain headers are needed, purpose etc?

- Working with real world devices
  - Writing program that interact with real world devices

- If you have to design a protocol in future
  - it helps to have practical experience of working (code) with protocols
Assignment Overview

- A SIP Client calls a specific SIP user-id and gets a music stream to its terminal.
- Can be considered as a **Music-On-Demand** service (assignment does not require video support, but if supported could also be considered a Video-On-Demand Service).
- Using SIP for On-Demand service seems an interesting idea.

Components Involved:

- A SIP softphone
  - Example: kphone, X-lite - Can be freely downloaded
- A RTSP server (already UP and running at Netlab)
  - rtsp://130.233.154.184:8554/song1.wav
- The assignment program that interacts with both the above components
Assignment Details

- Must be able to handle requests from a chosen SIP client
  - Your program need to pose as a SIP server
    - Building a complete SIP server is **not** the task (neither possible in the given time)
    - Need to have functionality to successfully establish a voice call
      - i.e. functionality to process and respond REGISTER, INVITE, ACK and BYE messages
  - SIP Clients support both signalling over TCP and UDP
    - Choose one, that you prefer to implement (TCP or UDP)

- Interface with the Assignment-2
  - Getting the media stream from the RTSP server. (would use the Assignment-2 RTSP Client functionality)
  - But the media stream destination need to be specified as SIP client
Recap: Assignment-2 Overview

Hints:
1. DESCRIBE Response contains SDP description (SDP parameters carry media format details)
2. SETUP Request and Response carries address parameters
In Assignment-2, the media data was received by RTSP client. But in Assignment-3, the media data destination need to be redirected.

Hints: Media format in SIP response need to be based on what RTSP server provides.
Program Execution Flow

- SIP client calls your Assignment-3 program.
  - SIP user-id `sip://song1@address.com`

- Receive the call
  - Extracts the user-id. (song1)

- Initiates RTSP session with the media server (but the media destination need to be modified)
  - `rtsp://130.233.154.184:8554/song1.wav`

- After receiving PLAY response from RTSP Server, send 200 OK to the SIP client.

- Now the media is played at the SIP Client

- Also, take care of ending the session cleanly
SIP clients

- X-Lite
  - http://www.counterpath.net/x-lite.html
  - OS support: Windows, Linux and Mac

- kphone
  - Download from http://www.wirlab.net/kphone/kphone-4.2.tar.gz
  - Source code available

There are many other clients available, you are free to choose your SIP client.
Submission guidelines
(for both Assignment 2 and 3)

▷ All required source files
▷ A readMe file with details on compilation and execution instructions
▷ Brief comment about your assignment (approx. 1 page)
  ● Implementation issues faced
  ● Comments/Suggestions/Complaints, if any
  ● Extra features if any
  ● Anything that you would like to tell us
▷ There shall be a demo of the submitted assignments
  ● Demo date, yet to be finalized.
  ● Follow announcements in Noppa
  ● Deadline: One Final Deadline for both Assignment 2 and 3
    ▪ January 9th 2009 (No extension possible, So start early)