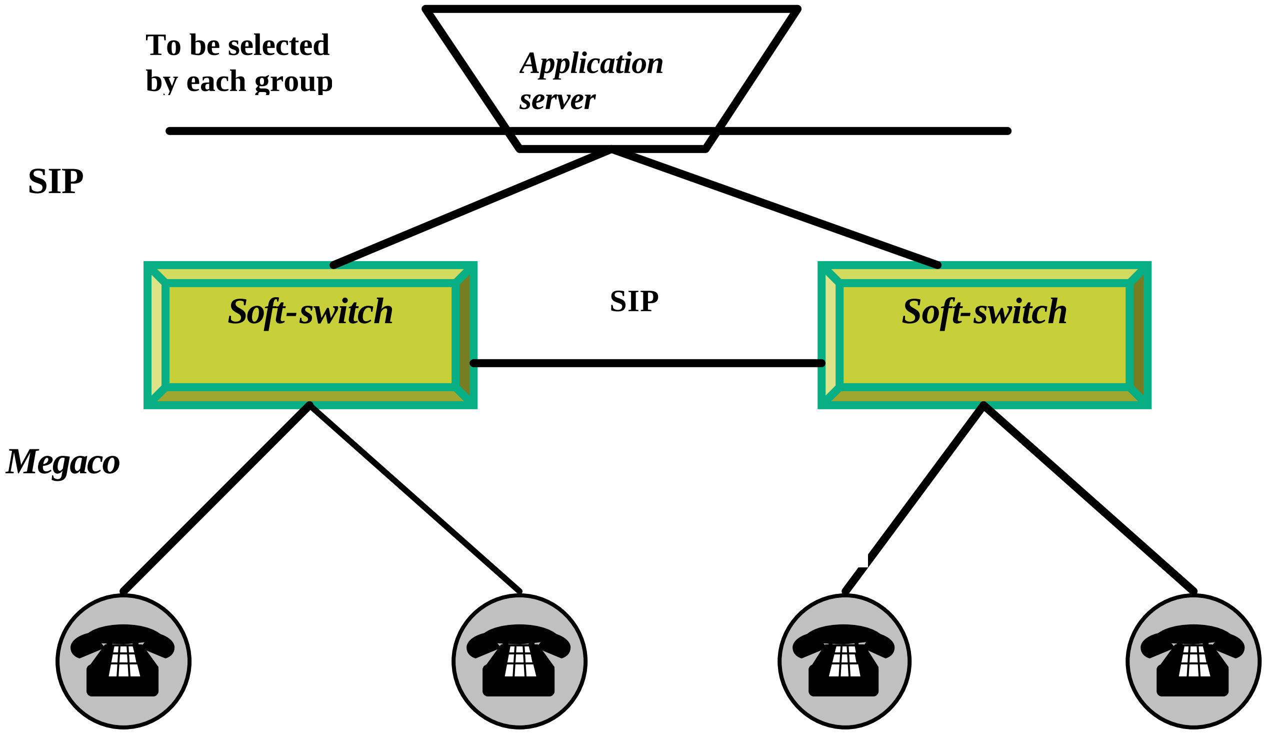
IP Phones, Soft-Switches and Value Added Services

1. Overview

The goal of this project is to specify, design, implement and demonstrate a simplified next generation telecommunications infrastructure. The infrastructure is made of IP phones, soft-switches, and an application server that contains software for value added services. The demonstration should include at least an application server containing at least a value added service, two soft-switches, and two IP phones connected to each soft-switch.

The paradigm to use for the application server (e.g. Parlay, SIP servlet, Web services) is to be decided by each group. The same applies to the specific value added services contained in the application server. The soft-switches use SIP for network to network signalling, and a simplified version of Megaco for network-user signalling. RTP is used for media handling and the phones are Megaco phones.



2. Components to be specified, designed and implemented during the project

2.1. IP Phones

The phones should be implemented as software running on desktops. This software should be able to handle the basic Megaco commands for call establishment / tearing down. It should also be able to notify the soft-switch when the user wants to establish a call.

2.2. *Simplified Megaco protocol*

A simplified version/profile of Megaco protocol should be specified and implemented. It should have as a minimum commands for establishing/tearing down calls, and notifying a soft-switch of the desire of the user t o establish a call.

2.3. *Soft-switch*

The soft-switch should implement as a minimum the following functions:

- Routing (e.g. the switch should be able to find out if the callee is connected to a different switch)
- Signalling (e.g. if the callee is connected to another switch, the switch should be able to request the other switch to establish the call)
- Generation / handling of basic Megaco commands for call establishment / tearing down and notifications.

2.4. *Application server*

The application server can be based on any of the paradigms studied in the course:

- Signalling protocol specific (e.g. SIP CGI, Servlet API)
- Signalling protocol neutral (e.g. Parlay)
- Emerging approaches (e.g. Web services)

3. Logistics

3.1. *Phases*

It is strongly recommended to handle the project in two phases. The first phase consists of implementing the network infrastructure:

- IP phones
- Megaco stack
- Soft-switches

All the related course material related to this phase will be covered by end January. This phase should therefore end by the end of February. Students have the options of showing a demo in early March.
The second phase consists of implementing the value added service infrastructure (i.e. application server + services). It should end at the end of the course.

3.2. *Teams*

Each team should consist of 2 to 4 students. Expectations on teams of more than 2 students are of course higher. While a simple value added service is acceptable for teams of 2 students, teams of 3 to 4 should provide an infrastructure with several services. The services should also be more sophisticated.

3.3 *Lab, programming languages and tools*

- The project will be hosted by the Concordia Institute for Information Systems Engineering (CIISE)' laboratory, located at 1425 Rene Levesque blvd. Users ids will be available starting from February 1st. The person to be contacted is Alex de Marco (Alex_demark@ece.concordia.ca)
- Any programming language can be used although Java is recommended. Ericsson can provide some Java tools (e.g. SIP stack, Parlay gateway). Tools listed at the URL below can also be used.

http://www.cs.columbia.edu/sip/implementations.html