Experiment 4:

Videoconference System: Setup and Protocol Analysis

References

- Andrew S. Tanenbaum: Computer Networks
- RFC0768: UDP - User Datagram Protocol
- RFC1889: RTP - A Transport Protocol for Real-Time Applications
- RFC2326: RTSP - Real Time Streaming Protocol
- http://www.cs.columbia.edu/~hgs/rtsp/faq.html.
- http://www.iec.org/online/tutorials/h323/: There is a detailed description about H.323 on this website.
- Documentation about the OpenH323 Gatekeeper (http://www.gnugk.org/)

Preparatory Questions

1. Why do video conferences and IP telephony have high network requirements?

2. What is the advantage of UDP over TCP for a video conference? For which part of the video conference is TCP used?

3. Explain how RTP and RTCP protocols work together.

4. Name different examples for using RTSP and H.323? What is the role of RTP in that matter?

5. Are video and audio sent in the same session? If separation is necessary, state why.

6. How are UDP, RTP, RTCP (RTP Control Protocol) and RTSP connected?

7. Can RTSP be used without an underlying streaming protocol to stream media? If not, which protocols can be used?

8. For what purposes does Q.931 used in IP-Telephone?

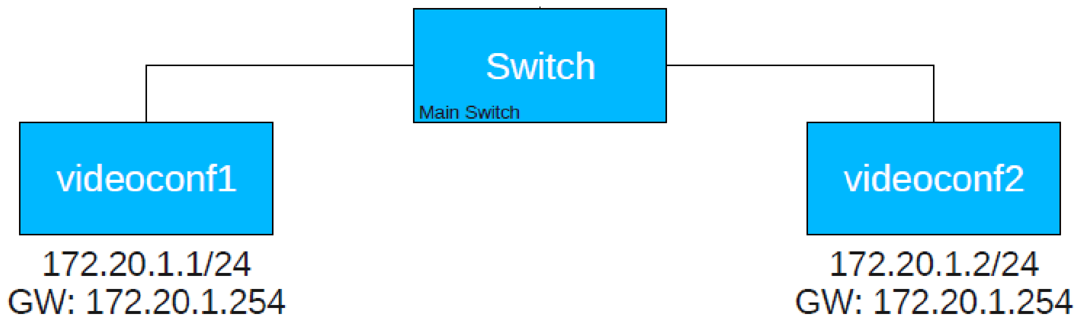
9. What are the tasks of the gatekeeper and gateway with VoIP?

10. How do VoIP connections differ with and without gatekeeper?

11. Is it possible to have more than one gatekeeper within one H323 zone?

Experiment Setup

The experiment consists of two virtual computers connected as the following figure, each of them connected to a virtual camera. These two computers are: [videoconf1, videoconf2]



To start this experiment:

1. Open a terminal (applications ➔ System Tools ➔ Terminal).
2. In the terminal, type "./run_exp4.sh", this will open the two Virtual Machines (VMs).

Experiment Procedure

Part 1: Peer to Peer Video Conferencing

1. Open the video conferencing software ekiga [Applications → Internet => Ekiga]
2. Open wireshark and start sniffing at both computers.
   - You can start wireshark by typing '*sudo wireshark &*'
   - Start capturing packets
   - Set the filter to 'sip or rtp'
3. Establish a connection between the two computers using ekiga.
   - On videoconf1 press view → Dialpad in Ekiga's menubar
   - In the lower box write '*sip:<ip_address_of_other_machine>*'
   - Click the call button (the green phone button).
   - Accept the call from videoconf2

4. Answer the following questions based on the sniffed packets and attach appropriate Snapshots:
    - How was the session initiated? What was the protocol used?
    - Did you find SDP packets? If so clarify over which protocol and what their function is.
    - Did you need to contact a server or was the whole session peer-to-peer?
    - What was the trasnport layer protocol of the SIP packets?
    - Over which application layer protocol where the video and audio packets transmitted?
    - What were the codecs used for the video and audio streams?
    - What happens if the receiver reject the call?

Part 2: Video Conferencing using a Gatekeeper

In this part, we will make a call between the two computers using their registered names (videoconf1 ➔ Alice, videoconf2 ➔ Bob).

1. Restart sniffing on wireshark and leave the filter as is.
2. Now make a call from Alice to Bob.
    - From videoconf1, press view → Contacts
    - Double click on Bob in the neighbours list to initiate the call.
3. Answer the following questions and attach snapshots:
    - How was the session initiated now?
    - Is there a new IP in the process? If yes, what is the role of the new host?
    - Based on your observations:
        - What are the differences between the current call and the previous one? Now what are the similarities?