

1 Integrated project - Let's talk

1.1 Challenge

Design and implement a WiFi-based multi-hop adhoc chat application between 4 wireless devices

1.2 Details

A normal chat application (whether it is based on text, voice or video) relies on a server to manage connections and to route data packets between and to clients. In this case, the server takes care of managing a chat session between client installed on device A and client installed on device B. If device A wants to communicate with device B, it needs to send a request using its client to the server and it is up to the server to deliver the message (or basically the packet) to the client installed on device B.

The adhoc chat application that you need to design and implement is different, as it does not rely on availability of an infrastructure or a server. In this case, the communication between device A and B is now serverless.

You do the integrated project in a group of 4 students.

You need to consider the following issues while designing your solution:

- You need to design and implement a multi-hop adhoc network between the 4 devices of your group using WiFi.
- You can use the multicast facility of the WiFi network to send packets to all other devices of your group which are within radio range of each other. For Java, use the Multicast Socket class, see:
<http://download.java.net/jdk7/archive/b123/docs/api/java/net/MulticastSocket.html>
Also see Section 1.3 for more details.
(Corresponding instructions for C++ for the EE students will be added later.)
- You will have to come up with some forwarding and routing mechanism to also exchange packets between nodes which can reach each other only via one or more intermediate nodes.
- Make sure your routing/forwarding algorithm is robust, so that even when the connectivity changes, packets cannot run around endlessly.
- Each device should broadcast its presence periodically.
- You need to consider link unreliability and the packet lost probability and ensure that messages are sent and received reliably. Think here about using acknowledgements of sequence numbers, for example.

- You need to decide what type of messages you support in your chat application. Think of short text messages, text files, audio files, or video files.
- You need to ensure the order of sent messages at the receiving device. For this, you may add time-stamp to each packet or a sequence number.
- Think how you can make the chat secure. You may consider for instance encryption.
- Think about the user interface.

1.3 Hints for getting started

You should boot your laptop from the USB given to you by the student assistants. The USB contains an UBUNTU Linux distribution. Copy the script 'adhoc_setup' to your home directory.

Execute the 'adhoc_setup' script with the following three parameters:

- wlanX: this parameter is the network interface name. Use iwconfig to find out all available network interfaces in your system. Replace wlanX with the name of your selected network interface.
- Y: Your group number. Each group will be given a unique group number from 1 to 20 by the student assistants.
- Z: Device number. This is the number of a device in one group. In a group of 4 students, Z will be 1..4 showing device 1, device 2, device 3, device 4.

For example, device 3 in group 13 should run the script like:

sudo bash adhoc_setup wlan1 13 3

After execution of the script, you should be able to ping each other in your group with ipv4 address: 192.168.5.Z (Z is the device number in your group).

1.4 Submission

Give a detailed description of the system design, identify all required components, their interfaces, requirements and the data flow. Include at least state charts of the overall system design in your description. From this description, the motivation for choices you make in the design should be clear. This description should be three pages long.

Before you start with implementation, you need to (i) make a planning including timeline, tasks and responsibilities of group members, and (ii) a detailed test plan. For each test case, write down the following:

- Which behavior do you expect of the system?

- What is the exact test setup? (make sure that the tests can be repeated based on your description!)
- How does the test scenario look like? What are the exact steps you should take during the test?
- What performance metrics do you consider for the system? what are the overall performance of the system?

The answer to the above questions including test results should be up to three pages long.

The adhoc chat application must be demonstrated (without a demonstration, no grade can be given).

The planning, design, test cases, performance evaluation metrics and results, and the implementation should be submitted in Blackboard.

1.5 Timeline

- 7 April 2015, 8:45: kick-off, with a short introduction to the project, requirements, and deadlines.
- 8 April 2015, 8:45–10:30: each group should show its planning including timeline, task division, and responsibilities.
- 13 April, 13:45: each group should show an initial demo of its system to the student assistants.
- 15 April, 13:45–17:30: final demo + borrel.
- 17 April, 15:45–17:30: demo of groups who failed on April 15.