CoE 4DN4 Advanced Internet Communications 2016
Lab 3

Objective:

In this lab you will implement an online group chatting system.

Work description:

Online chatting system is a typical peer-to-peer (P2P) communication system, where each peer can act as both a client and a server to exchange data. Once a P2P connection is established, communications are between the peers. You will implement group chatting, where several peers form a group first, and data sent from each member in the group are seen by all other group members. A system can include multiple groups each chatting on a different subject. For example, there can be three groups, one "programming" group, one "cooking" group, and one "travel" group. To make the work a bit easier, we assume that the system has a "default" group, and all system users are in the "default" group.

P2P discovery: Each member should discover its peers before chatting in a specific group (not the default group), e.g., the "cooking" group. To achieve this, the member can broadcast to the "default" group asking for response from members in a specific group (e.g., the "cooking" group), and all members in the group reply. This, however, can cause severe problems in large size systems that include a large number of members (why?). What we do here is to assume that when the system is initialized, each member stores information (IP, port number) of a list of peers in the system. The list may include a small number of peers, such as 3 or 4. When member **A** tries to discover peers for a specific group (not the "default" group), it only multicasts a "discovery" request to the peers in its list. Meanwhile, as a member receives a discovery request for a specific group (not the "default" group), if it wants to join the same group, it replies back to member **A** directly and forwards the request to all peers stored in its own list; and if it does not want to join the group, it simply forwards the request to all peers stored in its own list. The discovery command includes a TTL (time to live) field. Each time before a member forwards a discovery request, it reduces the TTL in the discovery request by 1. When the TTL is reduced to 0, the discovery command cannot be forwarded.

Once member **A** receives responses from one or multiple peers, it creates a multicast address and multicasts this address to all these peers, which can all join the same multicast address. All members in the group can then send messages to all other members in the group.

Requirements:

Demo:

- It is recommended that 3 or 4 students work in a group.
- Demonstrate to the TAs that 2 or 3 members can form a chatting group, and can

    - send and receive text messages within the group. That is, when one member sends a message, all other members receive the same message.
    - share files (such as music files) within the group. That is, when one member sends a file, all other members receive the same file.
- The protocol described above is mainly for forming a new chatting group (i.e., member **A** is the one that initializes the group chatting). Design a protocol and implement it so that member **A** can join an existing multicast group.
- Each group should schedule a demo session with the TAs between March 28, 2016 (Monday) and April 1, 2016 (Friday). Sign-up instructions will be available later.

Submission:

- Submission of lab reports is not required. However, each group should submit a zipped file of your codes to the Dropbox in Avenue to Learn.