CSE471: DATA COMMUNICATIONS AND COMPUTER NETWORKS

YEDITEPE UNIVERSITY

SPRING 2021

TERM PROJECT – DUE DATE MAY 28TH, 2021

As a term project, this semester you are expected to develop an encrypted chat application over IPv6. The application should use a peer-to-peer model to communicate with other nodes. All messages should be encrypted and sent to the local subnet and the relay between subnets should be handled by gateways nodes. However, your application should not overwhelm the network and bring on to a crash state. To prevent cycles and broadcast/multicast storms, you should implement some sort of control mechanism.

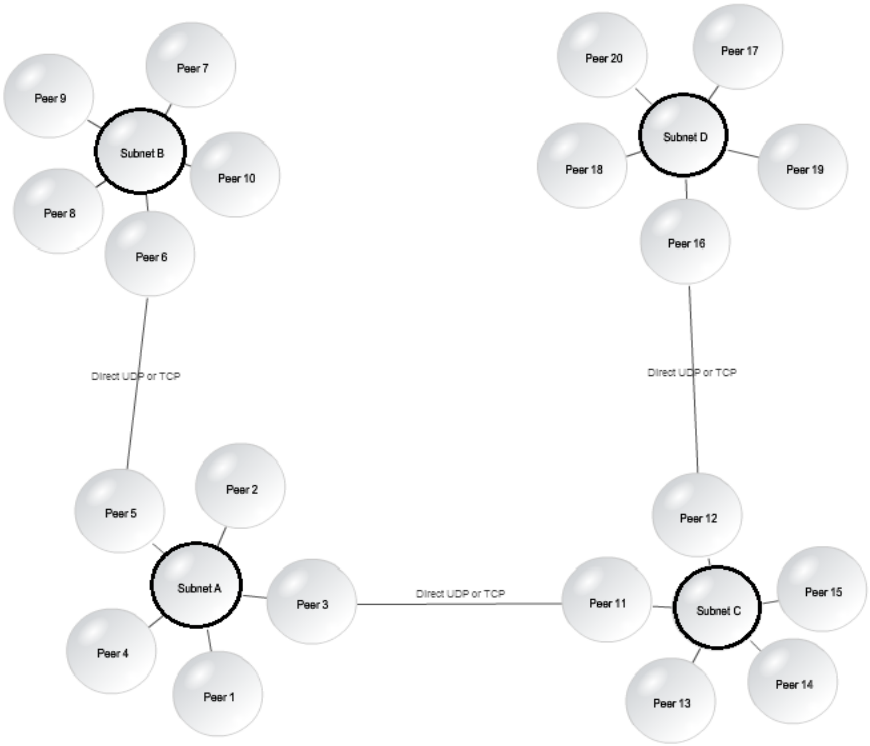

Figure 1: Chat System Overlay Network

Your system should eventually create a virtual overlay network (Figure 1) when all the protocols and algorithms are implemented. ZeroTier One (www.zerotier.com) is a platform to create a software-defined network (SDN) over existing network architecture. Utilizing this platform, multiple devices in the same virtual network can connect to remote hosts and clients using a very simple configuration. The platform can be configured to provide IPv4 as well as IPv6 support.

- In this proposed overlay chat network architecture, there should be two types of peers; (1) client peers that will act as chat applications and (2) gateway peers that will act as both chat applications and relays between two subnets. However, the application should allow the user to set manually the working mode according to his/her needs.

- A gateway node should be provided with a Gateway IP address list, which will contain all the IP addresses of nodes that act as gateway. This data can be stored on local file storage or can be retrieved from an online location. The gateway notes should be able to tunnel two IPv6 subnets via IPv4 infrastructure.

- For usability purposes, you are required to implement a graphical user interface (GUI) for your chat client – see Figure 2.

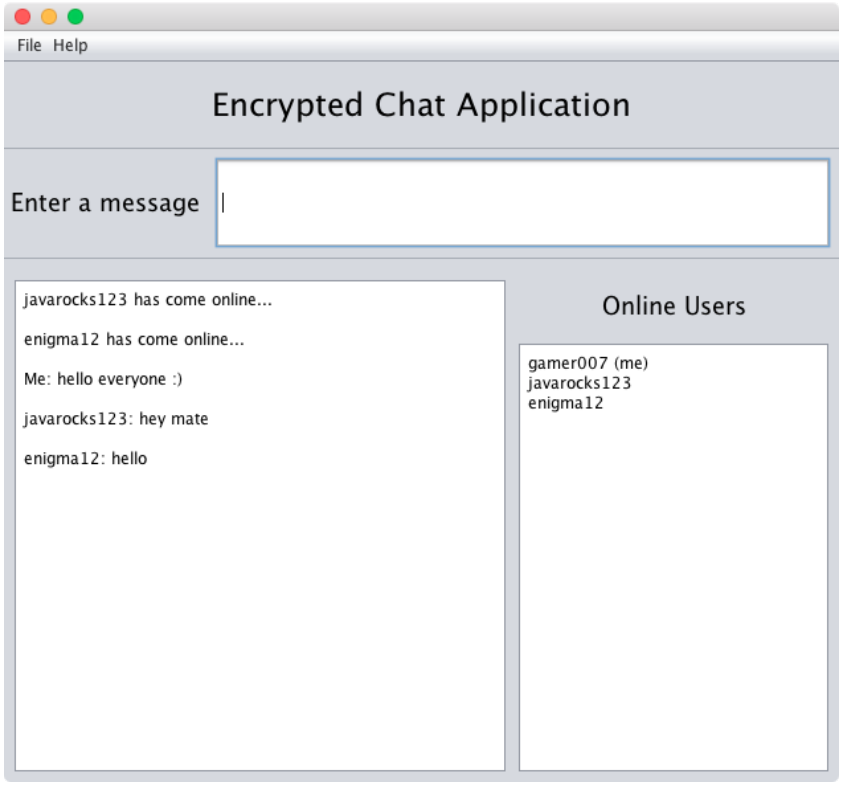

Figure 2: Chat Application Screenshot

- Before a user can connect to the Chat Network, s/he should create public and private keys using the *Generate Keys* menu item (*in File->Generate Keys*).
- After having created the keys, the user should be able to connect to the overlay chat network using *Connect to Network* menu item (*in File->Connect to network*). This command should trigger a pop up that requests a

Nickname from the user and consecutively multicasts the new user's identity (nickname and public key) to the local subnet.

- The user should be able to sign off from the Chat Network clicking the *Disconnect from network* menu item (*in File->Disconnect from network*). This action should trigger a special multicast *Quit* message.
- Users should be able to exit the application by pressing the Exit menu item (*in File -> Exit*).
- The Help menu should contain information about the developer of the application.

From the application logic point of view every node should satisfy the following requirements:

- Your chat network should be able to handle an arbitrary number of users and subnets.
- All messages sent should be using IPv6 global multicasting (`ff02::1`), with the exception of unicast communication between subnet gateway relay nodes.
- All the messages should be sent with RSA encryption.
- All the packages received from the local subnet by gateway peers should be relayed to the neighbouring subnets' gateway node using either direct UDP or TCP.
- All the nodes on the overlay network should keep track of active users and their public keys.
- Your application should implement a time interval mechanism to prevent cycles and multicast storms.
- High-level networking and user interface should be implemented using Java.
- [**BONUS**] Private chat between two users using the principles of the RSA algorithm.

Submit your project source code in a zip file, which has your student number as name, using COADSYS Exam Website (https://coadsysexam.yeditepe.edu.tr/) by the end of Friday, May 28th, 2021. All submitted source files will be checked for plagiarism among classmates and with any existing open source code available on the Internet. Furthermore, all students will be required to demonstrate their work for 15 minutes. DO NOT submit somebody else's work.