به نام خدا

تمرین کامپیوتری چهارم

استاد : دکتر یزدانی

اعضای گروه :

علی دارابی – 810100264

حسام رمضانیان – 810100248

توضیح کد های قسمت دوم:

1. **فایل Client.hpp و Client.cpp:**

Client::Client(const char* ip, uint16_t port) :

- یک سوکت ایجاد می‌کند. آدرس و پورت سرور را تنظیم می‌کند.

void Client::sendTo(int client_socket, const char* message) :

- داده‌های ورودی را به یک پکت تبدیل می‌کند و آن را از طریق سوکت ارسال می‌نماید. شناسه پکت را نیز افزایش می‌دهد.

void Client::retransmit(int packetId) :

- پکت با شناسه ورودی را از بافر دوباره ارسال می‌کند.

void Client::start() :

- به سرور متصل می‌شود، تبادل اولیه (دست دادن) را انجام می‌دهد، داده‌ها را ارسال می‌کند و در انتها ارتباط را قطع می‌نماید.

void Client::perform_handshake() :

- پیام‌های SYN، SYN-ACK و ACK را برای برقراری ارتباط با سرور ارسال و دریافت می‌کند.

void Client::send_data() :

- پیام‌های DEFAULT MESSAGE را تا تعداد مشخص ارسال می‌کند. تاییدیه‌های دریافتی را بررسی می‌کند و در صورت لزوم باز ارسال انجام می‌شود. همچنین کنترل ازدحام را مدیریت می‌نماید.

bool Client::isDuplicateAck(const char* ack) :

- تاییدیه دریافتی را بررسی می‌کند و مشخص می‌نماید که آیا یک تاییدیه تکراری (duplicate) است یا خیر.

2. فایل CongestionControl.hpp و CongestionControl.cpp:

CongestionControl::CongestionControl() :

- مقادیر اولیه حالت، اندازه پنجره و آستانه آغاز آهسته را تنظیم می‌کند.

void CongestionControl::onPacketAcked(bool isDuplicateAck, int numberOfLossPacket):

- بر اساس تاییدیه دریافتی و تعداد بسته‌های از دست رفته، اندازه پنجره و حالت کنترل ازدحام را به‌روز می‌کند.

void CongestionControl::onTimeout() :

- در صورت وقوع تایم‌اوت، آستانه آغاز آهسته و اندازه پنجره را به‌روز می‌کند و حالت را به آغاز آهسته تغییر می‌دهد.

uint32_t CongestionControl::getCwnd() const:

- اندازه پنجره را برمی‌گرداند.

State CongestionControl::getState() const :

- حالت فعلی کنترل ازدحام را برمی‌گرداند.

void CongestionControl::enterSlowStart() :

- حالت را به آغاز آهسته تغییر می‌دهد.

void CongestionControl::enterCongestionAvoidance() :

- حالت را به اجتناب از ازدحام تغییر می‌دهد.

void CongestionControl::enterFastRecovery() :

- حالت را به بازیابی سریع تغییر می‌دهد و آستانه آغاز آهسته و اندازه پنجره را به‌روز می‌کند.

void CongestionControl::exitFastRecovery() :

- از حالت بازیابی سریع خارج می‌شود و حالت را به اجتناب از ازدحام تغییر می‌دهد. اندازه پنجره را نیز به‌روز می‌کند.

3. فایل Packet.hpp و Packet.cpp:

Packet::Packet(int id, const char* data) :

- یک پکت جدید با شناسه و داده‌های ورودی ایجاد می‌کند.

const char* Packet::encode(const Packet& packet) :

- پکت ورودی را رمزگذاری می‌کند و یک رشته حاوی شناسه و داده‌های پکت را برمی‌گرداند.

Packet Packet::decode(const char* str):

- یک رشته رمزگذاری شده را از ورودی دریافت می‌کند و آن را رمزگشایی کرده و یک پکت حاوی شناسه و داده‌های استخراج شده را برمی‌گرداند.

void Packet::show() const :

- شناسه و داده‌های پکت را نمایش می‌دهد.

int Packet::getId() :

- شناسه پکت را برمی‌گرداند.

const char* Packet::getData() :

- داده‌های پکت را برمی‌گرداند.

4. فایل Server.hpp و Server.cpp:

Server::Server(const char* ip, uint16_t port) :

- یک سوکت سرور ایجاد می‌کند و آن را به آدرس و پورت ورودی متصل می‌نماید.

void Server::sendTo(int client_socket, int packet_id, const char* message) :

- یک پکت حاوی شناسه و داده‌های ورودی ایجاد می‌کند و آن را از طریق سوکت کلاینت ارسال می‌نماید.

void Server::start() :

- در یک حلقه نامحدود، منتظر درخواست‌های اتصال از کلاینت‌ها می‌ماند و برای هر درخواست جدید یک رشته جدید برای پردازش آن ایجاد می‌کند.

void Server::handle_client(int client_socket) :

- داده‌ها را از کلاینت دریافت می‌کند، پاسخ‌های مناسب را ارسال می‌نماید و رفتار خاصی در مقابل بسته‌های از دست رفته یا دریافت ناقص انجام می‌دهد.

5. **فایل main.cpp:**

- بر اساس ورودی کاربر، یک نمونه از کلاس Server یا Client را ایجاد می‌کند و متد start آن را فرا می‌خواند.