**Bài 1 FTP Client**

**• Bài 2. HTTP Downloader**

**• Bài 3. HTTP Server**

**• Bài 4. Chia sẻ tệp tin ngang hàng**

**• Bài 5. Game Server**

**• Bài 6. Chat Voice + Gửi File**

**• Bài 7. Image Streaming**

**• Bài 8. Multicast Video Streaming**

**• Bài 9. Apple HTTP Live Streaming Server**

**• Bài 10. GameShow**

**• Bài 11. UPNP**

**• Bài 12. Mã hóa đường truyền**

**• Bài 13. FTP Server**

**• Bài 14. TCP Redirector**

**• Bài 15. MS-RTSP Capture**

**• Bài 16. Peer 2 Peer Video Streaming**

**Bài 1 FTP Client**

• Viết chương trình FTP Client thực hiện được ít nhất các chức năng sau:

– Đăng nhập vào một máy chủ FTP.

– Hiện danh sách các tệp tin trên máy chủ.

– Tải một tệp tin được chọn, có thể tải bằng nhiều kết nối

song song.

– Upload được tệp tin lên thư mục được chọn.

– Đổi tên, xóa một tệp tin, thư mục được chọn

– Tạo một thư mục mới

– Ngôn ngữ C/C++/WinSock/MFC.

– Độ khó: 2.

– Tham khảo: FileZilla

**• Bài 2. HTTP Downloader**

• Viết chương trình tải tệp tin theo giao thức HTTP và

HTTPS thực hiện các chức năng sau

– Tải tệp tin ở một địa chỉ xác định.

– Cho phép nhập số luồng song song cần tải.

– Cho phép nhập các thông số về phạm vi của tệp tin muốn tải

(offset, length)

– Xử lý được việc tải file tự động từ các website chia sẻ thông

dụng:

• Mediafire.com

• Megashare.vn

• fshare.vn

• …

– Cho phép nhập proxy cho từng File, từng kết nối trong file.

– Ngôn ngữ C/C++

– Độ khó: 5

**Bài 2. HTTP server**

• Viết chương trình HTTP Server (HTTP/1.1) thực

hiện được các công việc sau.

– Xử lý được các lệnh: GET POST.

– Xử lý được cookie, session.

– Đáp ứng được tối đa 6400 kết nối.

– Xử lý được các vấn đề timeout, lọc kết nố

– Thống kê và ghi nhật ký.

– Xử lý được ngôn ngữ html, php (bằng cách triệu gọi

chương trình php.exe và lấy kết quả chuyển cho client)

– Không được sử dụng OpenSource

– Ngôn ngữ: C/C++

– Độ khó 6

**Bài 4. Chia sẻ tệp tin ngang hàng**

• Thiết kế một hệ thống quản lý và chia sẻ tệp tin ngang hàng P2P.

– Tự thiết kế giao thức giữa client-server

– Đảm bảo được tính công bằng trong hệ thống: upload nhiều => download nhiều

– Có cơ chế xác thực client và lưu nhật ký

– Ngôn ngữ C/C++/WinSock/MFC

– Không được sử dụng OpenSource

– Độ khó: 9

**Bài 5. Game Server**

• Xây dựng một GameServer trên nền TCP phục vụ các game Turn-Based (Caro, Chess, …)

– Tự thiết kế giao thức giữa client-server.

– Quản lý user, score… bằng cấu trúc dữ liệu tự tạo, hoặc

CSDL SQLite.

– Xây dựng game demo “Caro”.

– Đáp ứng tối đa 6400 user chơi cùng một lúc.

– Ngôn ngữ: C/C++

– Độ khó 7

**•• Bài 6 Xây dựng hệ thống chat voice + Gửi File**

– Ưu tiên thiết lập kết nối trực tiếp giữa các client

– Trong trường hợp không thể thiết lập kết nối trực tiếp

thì có thể trung gian qua server

– Giao thức tự thiết kế, codec tự chọn (mp3,wma,aac…).

Không truyền âm thanh chưa nén (wav,pcm).

– Mã hóa kênh truyền theo giải thuật tùy chọn.

– Ngôn ngữ C/C++/WinSock/MFC.

– Cho phép sử dụng OpenSource lame để nén và giải nén

MP3.

– Độ khó:8

**Bài 7. Image Streaming**

• Xây dựng hệ thống thu nhận hình ảnh từ webcam

qua internet.

– Xây dựng hai thành phần: Client + Server

– Client: Liên tục thu nhận hình ảnh từ Webcam và gửi về

Server. Cho phép thiết lập kích thước ảnh, tần suất gửi,

chất lượng nén…

– Server: Tổng hợp hình ảnh từ client và hiển thị ra màn

hình. Có thể điều khiển để thay đổi các tham số của

client từ xa.

– Ngôn ngữ: C/C++

– Thư viện: OpenCV.

– Độ khó: 10

**Bài 8. Multicast Video Streaming**

**•** Viết chương trình gửi dữ liệu multicast đến một nhóm các máy trong mạng

– Đầu vào là file video định dạng mkv hoặc mp4, với

codec là H264 và MP3

– Các client trong nhóm multicast có thể tự giải mã và

hiển thị video (dùng ffdshow, ffmpeg) hoặc sử dụng

chương trình thứ ba (Windows Media Player, Media

Player Classic) để hiển thị.

– Giao thức streamming tự định nghĩa hoặc sử dụng UDP.

– Ngôn C/C++. Được phép sử dụng thư viện ffmpeg.

– Độ khó: 10

Bài 7. IPv6

Tìm hiểu giao thức IPv6. Xây dựng ứng dụng thử

nghiệm chia sẻ tệp tin trong mạng IPv6.

– Độ khó: 0

**Bài 9. Apple HTTP Live Streaming Server**

t• Xây dựng server theo giao thức HTTP Live

Streaming

– Cho phép các client là iPhone, iPod, iPad kết nối đến và

hiển thị video thời gian thực.

– Đầu vào: nguồn video bất kỳ (MP4, MKV, AVI…),

Webcam.

– Đầu ra: Video theo chuẩn HTTP Live Streaming

(H264+AAC)

– Thư viện được phép dùng : ffmpeg

– Ngôn ngữ: C/C++

– Độ khó: 10

**Bài 10. Lịch chiếu phim**

• Xây dựng ứng dụng trên Desktop cho phép thu thập

lịch chiếu phim tại các rạp và các kênh truyền hình

cáp.

– Tự động hiển thị thông tin về: phim đang chiếu rạp, show

truyền hình đang chiếu tại thời điểm hiện tại.

– Các thông tin cần có về phim: Tên, Poster (Ảnh), Thể loại,

đánh giá, tóm tắt, trailer….(nguồn từ imdb.com).

– Các thông tin cần có về show truyền hình: tên show, giờ phát.

– Đầu vào: 24h.com.vn, vtcv.vn, sctv.vn, platinumcineplex,

egastar, national cinema centre….

– Cho phép đặt báo thức đến thời điểm một phim/show truyền

hình nào đó.

– Ngôn ngữ: C/C++/C#

– Độ khó: 5

**Bài 11. Remote Desktop**

• Viết chương trình điều khiển máy tính từ xa có các

chức năng sau:

– Quan sát được màn hình

– Quan sát được bàn phím và chuột

– Điều khiển được chuột và bàn phím nếu cần

– Giao thức tự định nghĩa.

– Ngôn ngữ: C/C++

– Độ khó: 10

– Gợi ý: Có thể sử dụng API SendInput để mô phỏng sự

kiện chuột/bàn phím ở máy nhận

**Bài 10. GameShow**

• Xây dựng một hệ thống chơi game show trực tuyến

thời gian thực.

– Hệ thống cho phép người chơi đăng ký chơi qua internet. Đến

đúng một khung giờ xác định trong ngày thì tất cả sẽ cùng

tham gia.

– Hệ thống sẽ tự gửi câu hỏi vào từng thời điểm thích hợp

– Mỗi câu hỏi có 1p để người chơi trả lời. Nếu trả lời sai, hoặc k

trả lời sẽ bị loại, tuy vậy vẫn được xem các câu hỏi diễn ra sau

đó.

– Người chơi có thể cá cược : 2X, 75%, 50%, 25%.

– Kết thúc cuộc chơi sẽ thông báo người thắng cuộc và phần

thưởng.

– Ngôn ngữ: C/C++

– Độ khó 5

**Bài 11. UPNP**

• Viết chương trình thử nghiệm giao thức UPnP để điều

khiển modem ADSL mở cổng NAT tự động.

• Cài đặt thuật toán UDP Hole Punching để tạo kết nối

giữa hai thiết bị nằm trong mạng NAT.

• Trong trường hợp không thể điều khiển modem hoặc

UDP Hole Punching thì tìm hiểu và cài đặt một kỹ thuật

NAT Traversal để kết nối hai client trong hai mạng

NAT qua internet (sử dụng một server trung gian để

mồi kết nối ).

• Ngôn ngữ: C/C++

• Không được sử dụng OpenSource

• Độ khó: 6

**• Bài 12. Mã hóa đường truyền**

• Viết chương trình chia sẻ tệp tin, thông điệp qua

kênh truyền mật SSL. Yêu cầu hỗ trợ các hình thức

xác thực sau:

– Mật khẩu.

– Chứng thực số được ký bởi một CA (tự tạo CA Root).

– Sử dụng thư viện OpenSSL hoặc một OpenSource tùy ý.

– Ngôn ngữ C/C++.

– Độ khó: 9

**Bài 15.WinPCap**

• Sử dụng thư viện WinPCAP để viết chương trình

– Bắt và hiển thị tất cả các ethernet frame đi ra và vào một

giao diện của máy tính (Địa chỉ MAC, IP, Cổng, ..)

– Xây dựng hình thức tấn công ARP Spoofing, hiển thị tất

cả các gói tin ra vào trong mạng cục bộ.

– Cấm không được chạy thử trong mạng của trường.

– Độ khó: 9

**Bài 15.Network Tamper**

• Sử dụng thư viện WinPCAP để viết chương trình

– Bắt và hiển thị tất cả các ethernet frame đi ra và vào một

giao diện của máy tính (Địa chỉ MAC, IP, Cổng, ..)

– Chỉnh sửa các thông tin của gói tin đi vào từ giao diện IA

và gửi ra ngoài theo giao diện IB (Yêu cầu máy test phải

có 2 giao diện mạng).

– Các luật có thể chỉnh sửa: Địa chỉ nguồn/đích, cổng

nguồn/đích.

– Cấm không được chạy thử trong mạng của trường.

– Độ khó: 8

**Bài 16. Gửi tin nhắn SMS**

• Viết chương trình gửi tin nhắn SMS miễn phí/có phí thông qua Portal của ba nhà cung cấp

VinaPhone, Vietel, MobiFone.

– Yêu cầu nhập số điện thoại và mật khẩu để đăng nhập vào Portal.

– Kiểm tra trạng thái tin nhắn, danh sách tin đã gửi…danh bạ nếu có

– Ngôn ngữ C/C++/C#

– Độ khó: 5.

– Được phép sử dụng libCURL

**Bài 13. FTP Server**

• Xây dựng ứng dụng FTP Server có thể thực hiện ít

nhất các chức năng sau:

– Cung cấp các lệnh FTP cơ bản

– Quản lý người dùng & các thư mục của người dùng

– Ngôn ngữ: C/C++.

– Đáp ứng được tối đa 6400 kết nối.

– Không được phép sử dụng OpenSource

– Độ khó: 6

**Bài 14. TCP Redirector**

• Xây dựng một cặp ứng dụng server và client có nhiệm vụ thiết lập một

kênh truyền ảo giữa hai điểm nhất định trên internet. Kênh truyền ảo phải

cung cấp các cơ chế sau:

– Nén dữ liệu trên đường truyền, giải thuật tùy chọn hoặc sử dụng OpenSource.

– Mã hóa dữ liệu trên đường truyền theo mật khẩu, hàm mã hóa tùy chọn.

– Một cặp client-server có thể mở nhiều kênh truyền.

– Một server có thể đáp ứng nhiều client.

– Server xác thực client thông qua mật khẩu.

– Hoạt động tương tự như giao thức SOCK.

– Ngôn ngữ C/C++ .

– Độ khó: 8.

– Có thể sử dụng libz để nén đường truyền.

Ví dụ: Server chạy ở máy 10.0.0.8, client chạy ở máy 10.0.0.2. Client không thể truy

nhập đến www.google.com:80 nhưng server thì có. Client sẽ tạo đợi kết nối ở

cổng 80, và chuyển tiếp tất cả dữ liệu nó nhận được đến server, server sẽ mở

kết nối và chuyển tiếp mọi dữ liệu từ client đến www.google.com:80

**Bài 15. MS-RTSP Capture**

• Xây dựng ứng dụng bắt các gói tin theo giao thức MS-RTSP (Microsoft).

– Đầu vào là một URL có dạng:rtsp://broadcast.manager.co.th/esan

– Đầu ra là một file asf (wmv, wma) có thể chơi được bằng Windows Media Player.

– Ngôn ngữ: C/C++

– Tham khảo: live555, rtspdump

– Độ khó: 10

**Bài 16. Peer 2 Peer Video Streaming**

• Tìm hiểu và sử dụng giao thức Bittorent Live để streaming một file video qua mạng ngang hàng.

– Cài đặt giao thức trên server và client.

– Đầu vào là một file video bất kỳ.

– File video sẽ được phát qua các nút trong mạng ngang hàng. Client có thể chuyển tiếp dữ liệu cho một player khac, VD: VLC

– Ngôn ngữ: C/C++

– Độ khó: 10

1. Viết chương trình truyền nhận file văn

bản đơn giản (text) giữa 2 máy tính trong

mạng LAN dùng cơ chế đa tiểu trình

(multi-threading).

2. Viết chương trình điều khiển máy tính

dùng công nghệ Bluetooth

3. Remote Shell: điều khiển máy tính từ xa -

- command line

3/04/2023

Chương 3: Sockets

2

4. Chương trình gửi/nhận mail dùng IMAP

5. Mô phỏng giao tiếp Web browser - DNS

Server - Web server

6. Chương trình Web browser

7. Image slider: trình chiếu ảnh qua mạng

8. Xây dựng chương trình CafeNet

9. Xây dựng chương trình theo dõi màn

hình của Client

3/04/2023

Chương 3: Sockets

3

10.Chương trình FTP Client

11.Chương trình Email Client

12.Sử dụng giao thức HTTP để viết chương

trình thi trắc nghiệm qua mạng LAN

13.Viết chương trình hỗ trợ thi (như đấu

trường 100) qua mạng LAN

14.Chương trình mô phỏng DHCP

15.Chương trình mô phỏng Proxy Server

3/04/2023

Chương 3: Sockets

4

16.Cài đặt các bài toán tính toán cơ bản sử

dụng kỹ thuật Remoting

17.Tính giai thừa bằng cơ chế đa tiểu trình

(multi-threading).

18.Chương trình truyền nhận tập tin có dung

lượng lớn qua mạng

19.Chương trình Chat có hỗ trợ gửi file

3/04/2023

Chương 3: Sockets

5

20.Chương trình Chat có hỗ trợ video

21.Cải tiến chương trình chat đã minh họa

trong chương 3, sao cho các người dùng

được phép chat riêng tư với nhau thông

qua "trạm trung chuyển" là Server.

22.Viết ứng dụng chơi cờ Caro qua mạng.

(Sử dụng giao thức UDP). Gợi ý: mỗi khi

người dùng đi thì sẽ gửi vị trí của ô vừa

đi cho ứng dụng kia (đối phương).

3/04/2023

Chương 3: Sockets

6

23.Xây dựng phần mềm giả lập hệ thống

Load Balancing.

24.Ứng dụng lập trình phân tán để xây dựng

hệ thống đặt vé tàu trực tuyến