**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**ĐỒ ÁN 01**

**Proxy server sử dụng socket**

**BỘ MÔN MẠNG MÁY TÍNH**

**Sinh viên thực hiện** :

* **NGUYỄN MINH TÂM**

**MSSV: 1712746**

* **VŨ PHẠM ĐỨC THẮNG**

**MSSV: 1712763**

**Lớp**: **CQ2017/1**

**MỤC LỤC**

[PHẦN I. TÓM TẮT MÔ TẢ YÊU CẦU ĐỒ ÁN…………………...3](#__RefHeading___Toc643_2491687699)

[PHẦN II. MÔ TẢ HƯỚNG GIẢI QUYẾT VẤN ĐỀ VÀ PHÂN CÔNG CÔNG VIỆC NHÓM……………………………………………..3](#__RefHeading___Toc645_2491687699)

[I. Tạo phần xương sống cho proxy (Nguyễn Minh Tâm)…………………………3](#__RefHeading___Toc647_2491687699)

[II. Blocking website (Vũ Phạm Đức Thắng)………………………………………….4](#__RefHeading___Toc649_2491687699)

[III. Caching website (Vũ Phạm Đức Thắng)………………………………………….4](#__RefHeading___Toc651_2491687699)

[PHẦN III. NHỮNG HÀM CHỨC NĂNG CHÍNH………………..5](#__RefHeading___Toc653_2491687699)

[I. Hàm main:………………………………………………………………………………..5](#__RefHeading___Toc655_2491687699)

[II. Trong hàm client\_listen:……………………………………………………………](#__RefHeading___Toc657_2491687699)..[5](#__RefHeading___Toc657_2491687699)

[III. Những hàm cache:…………………………………………………………………](#__RefHeading___Toc382_2771618234)...[6](#__RefHeading___Toc382_2771618234)

[IV. Những hàm phụ:……………………………………………………………………...7](#__RefHeading___Toc659_2491687699)

[PHẦN IV. Bắt gói tin Wireshark và giải thích:………………..7](#__RefHeading___Toc384_2771618234)

[PHẦN V. Vai trò proxy trên thực tế:……………………………….8](#__RefHeading___Toc386_2771618234)

# PHẦN I. TÓM TẮT MÔ TẢ YÊU CẦU ĐỒ ÁN

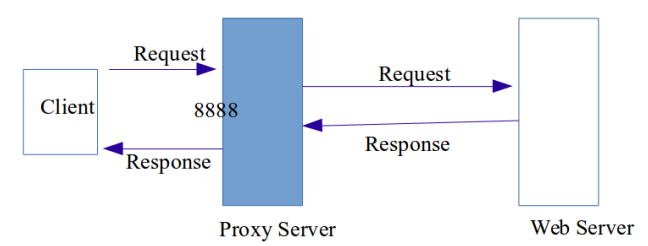
Chỉ sử dụng Socket do ngôn ngữ lập trình cung cấp để viết một chương trình Proxy Server sử dụng cho HTTP thoả mãn yêu cầu:

1. Có thể truy cập HTTP website (các method GET, POST và HTTP 1.0, 1.1), không cần HTTPs và chạy trên port 8888.
2. Chặn các website có trong file blacklist.conf, mỗi dòng một domain website.
3. Có chức năng caching (tự đưa ra cơ chế caching).

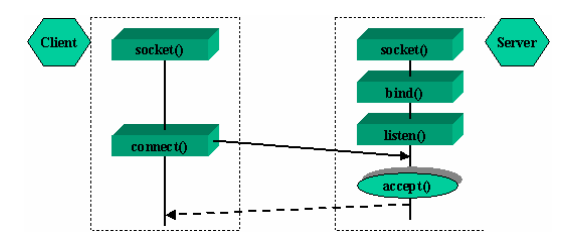
# PHẦN II. MÔ TẢ HƯỚNG GIẢI QUYẾT VẤN ĐỀ VÀ PHÂN CÔNG CÔNG VIỆC NHÓM

* Vì đồ án mang tính lý thuyết cao và khó để chuyển tiếp phần đang làm của người này sang người kia nên sẽ phân công theo yêu cầu (proxy, blocking, caching), mỗi tuần họp với nhau để trau đổi lý thuyết, tiến độ…

## Tạo phần xương sống cho proxy (Nguyễn Minh Tâm)



* Sẽ có một **Socket** làm proxy server và bind tới port 8888 localhost
* Lắng nghe từ các client (các request từ browser)
* Khi có một client kết nối (**Socket**) thì sẽ tạo một thread để làm đa luồng, truyền tham số là socket đó



**Với client là những request từ browser và server là proxy**

* Nhận request từ client, tìm host của web server qua hàm getaddrinfo
* Tạo một **Socket** connect đến web server và gửi request lên server
* Nhận response rồi gửi về client (browser)

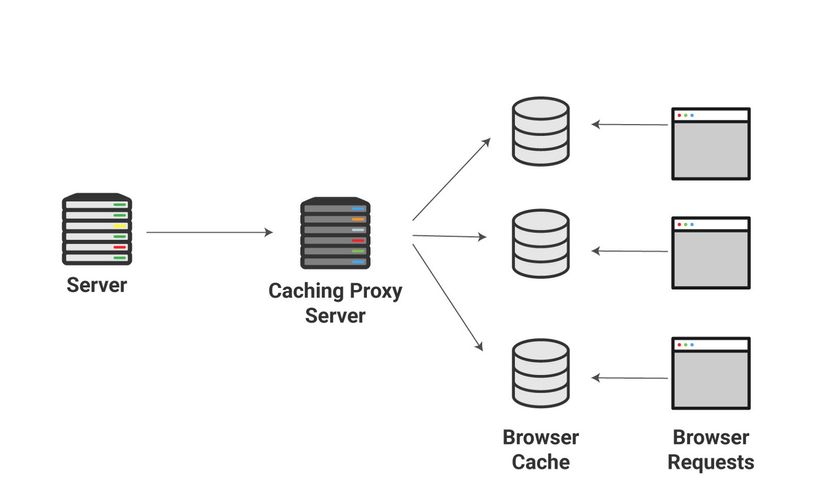
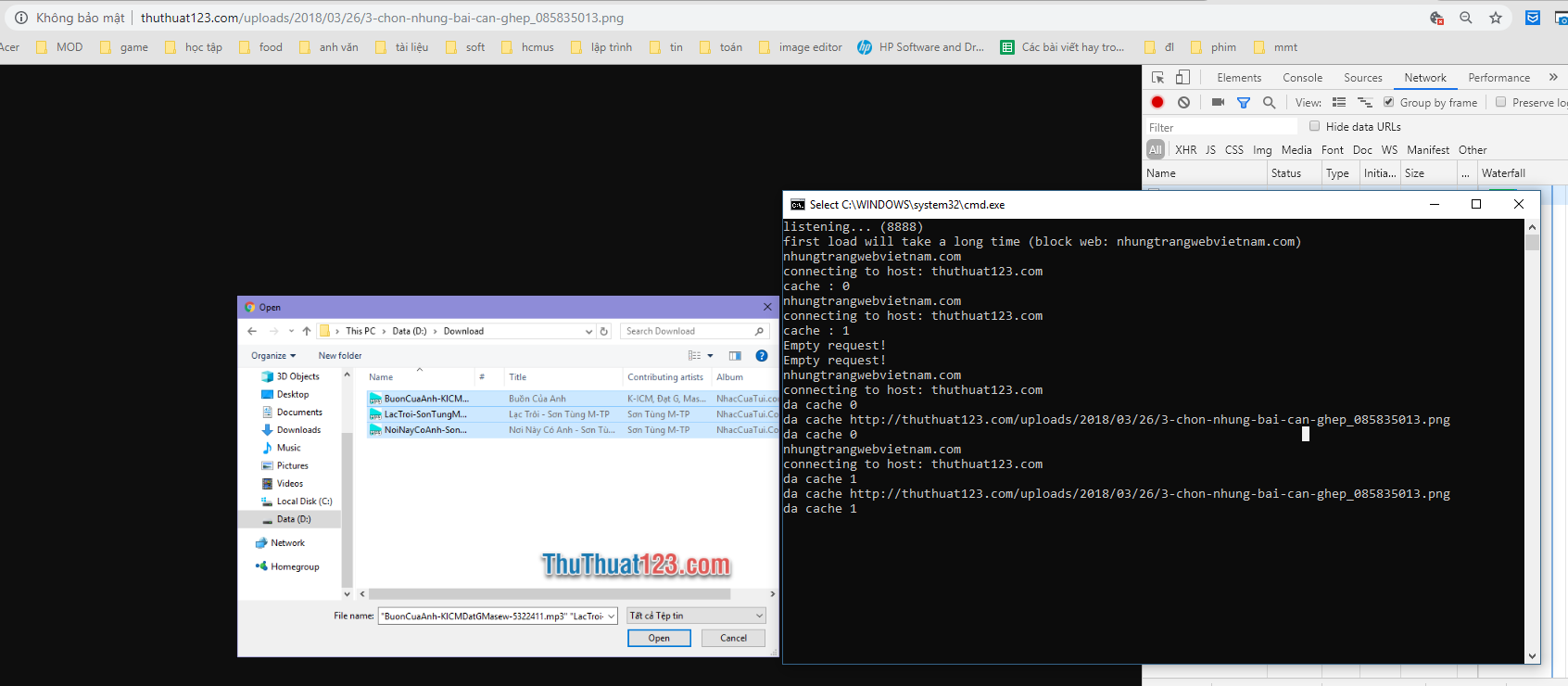
## Blocking website (Vũ Phạm Đức Thắng)

* Từng dòng của file blacklist.conf là mỗi website cần chặn
* Mỗi lần khi nhận request, kiểm tra domain nếu trong black list thì gửi về client chuỗi response sau:

#define block\_response "HTTP/1.1 403 Access denied\r\nConnection: close\r\n ContentType: text/html\r\n\r\n<html>\r\n<body>\r\Access denied 403\r\n</body>\r\n</html>\r\n\r\n"

## Caching website (Vũ Phạm Đức Thắng)

* Mục tiêu: Load trang web nhanh hơn ở những lần truy cập kế tiếp hoặc truy cập offline.
* Ý tưởng thực hiện: Lưu các response có thể cache (Response Header có chứa mã *200 OK* và chứa *Last-Modified* hoặc *Etag*) rồi tải lại ở lần truy cập tiếp theo
* Những component không được cache sẽ được kết nối tới server để lấy xuống lại
* Sử dụng hàm **char \* rebuild\_rq** để thêm phần “*If-Modified-Since*” (so sánh với “*Last-Modified*” của response header) và “*If-None-Match*” (so sánh với Etag), nếu trùng nhau thì trả về code *304 Not Modified*



# PHẦN III. NHỮNG HÀM CHỨC NĂNG CHÍNH

## Hàm main:

Ở hàm chính của chương trình ta tạo socket cho proxy, bind tới port 8888 localhost, listen những kết nối.

Khi có một kết nối thì tạo một luồn mới vào hàm client\_listen với tham số truyền vào là socket của client

## Trong hàm client\_listen:

DWORD WINAPI client\_listen(LPVOID lp);

Chức năng:

Khi nhận được một request, thêm một luồng mới sẽ thực hiện hàm này, thực hiện việc xử lý request từ browsers và response từ server

Trong hàm, nhận request từ browser, kiểm tra xem request có phải là request cần xử lý (hàm phụ right\_request)

Kiểm tra có phải là trang web cần block, nếu phải thì block website qua hàm:

bool send\_block\_response(SOCKET client);

Chức năng:

Gửi về client (browser) response 403, true nếu thành công, false nếu ngược lại)

Nếu không nằm trong black list thì tạo một socket để connect với host (lấy host bằng hàm phụ get\_host) qua hàm:

SOCKET createserverSocket(char\* pcAddress, char\* pcPort);

Chức năng:

Truyền vào địa chỉ host và port và trả về socket đã connect tới server (sử dụng hàm getaddrinfo)

Gửi request đến server qua hàm:

bool sendToSocket(const char\* buff, SOCKET sockfd, int buff\_length, string

name\_sock);

Chức năng:

Gửi đến socket data buf, truyền vào data cần gửi (buff), socket cần gửi đến (sockfd), chiều dài buf (buff\_length), tên socket (name\_sock, dùng để debug). Trả về true nếu thành công, false nếu ngược lại

Sau đó nhận request từ server (WebServer) rồi gửi lại cho client (browser) qua hàm:

bool revThenSendBack(SOCKET Clientfd, SOCKET Serverfd);

Chức năng:

Nhận từ Serverfd và gửi về Clientfd (trong hàm có sử dụng hàm sendToSocket)

## Những hàm cache:

Dùng struct CACHE để lưu các nội dung host, page, datetime (chứa last-modified) và etag của trang thành phần được cache. Khởi tạo tối đa 100 CACHE, nếu vượt quá thì lưu đè lại vị trí thứ 0.

Vector vector<vector<char\*>> DATA để chứa response của thành phần được cache, nếu trả về nhiều response thì các response sẽ được pushback vào sau. Vậy nên ta dùng vector 2 chiều.

Hàm int have\_cached(char \* request) trả về số thứ tự trong CACHE, nếu không tìm thấy thì trả về -1. Một trang đã được cache là trang đó có host và page ở cached trùng với host và page ở request.

Hàm int freeIndex() tìm kiếm vị trí còn trống trong 100 CACHE

Nếu một trang chưa được cache (have\_cached trả về -1) thì gửi request lên server và nhận response về server, đồng thời lưu những thông tin cần để cache thông qua chính hàm

bool recvThenSendBack.

Ngược lại, tải thông tin của trang đã được cache lên browser.

## Những hàm phụ:

bool right\_request(char\* request);

Xác định xem request của browser là cái cần xử lý.

Một request đúng yêu cầu:

* Có method là GET, POST
* Phải có chiều dài chuỗi lớn hơn 0
* Có chuỗi kết thúc request (“\r\n\r\n”)

char\* get\_host(char\* request);

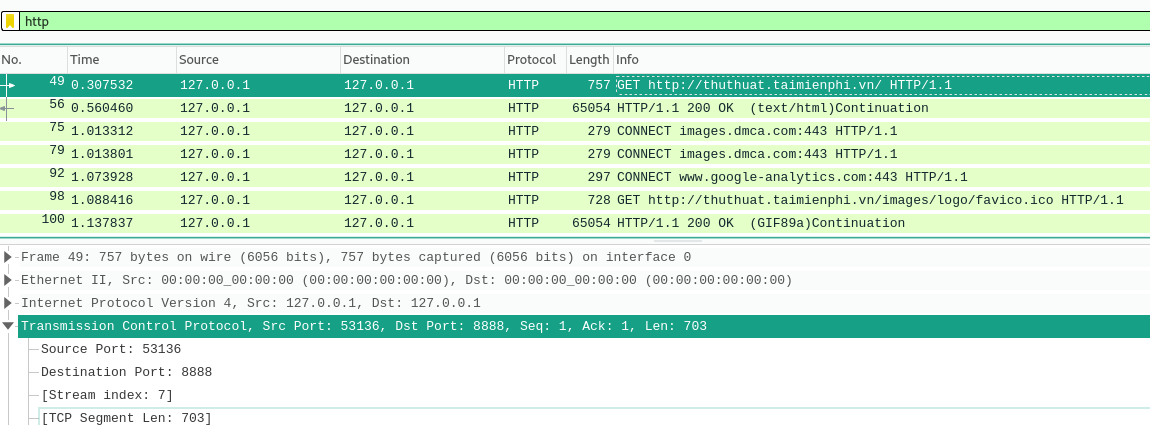
Lấy Host từ request

vector<char\*> read\_black\_list(string file\_name);

Lấy black list từ file (truyền file\_name là “blacklist.conf”)

# PHẦN IV. Bắt gói tin Wireshark và giải thích:

Trình duyệt gửi các request từ máy (có port thường 50000+) đến proxy (port 8888)



Sau đó từ proxy gửi lên webser, nhận về response, gửi về proxy (port 8888), rùi gửi đến client (port 50000+)

# PHẦN V. Vai trò proxy trên thực tế:

Có nhiều lý do tổ chức và cá nhân nên sử dụng máy chủ proxy.

* **Để kiểm soát việc sử dụng Internet của nhân viên và trẻ em:** Tổ chức và phụ huynh thiết lập máy chủ proxy để kiểm soát và giám sát nhân viên hoặc trẻ em sử dụng Internet. Hầu hết các tổ chức không muốn nhân viên của họ xem các trang web cụ thể trong thời gian làm việc. Họ có thể giám sát và ghi lại tất cả các yêu cầu web, do đó mặc dù không chặn trang web nhưng họ vẫn biết thời gian bạn dành cho những việc làm khác ngoài công việc.
* **Tiết kiệm băng thông và cải thiện tốc độ:** Các máy chủ proxy có thể lưu vào bộ nhớ cache (lưu một bản sao trang web cục bộ) các trang web hay truy cập. Do đó khi yêu cầu trang web, máy chủ proxy sẽ kiểm tra xem có bản sao mới nhất của trang web này hay không và sau đó sẽ gửi cho bạn bản sao đã lưu. Điều này có nghĩa là khi hàng trăm người truy cập vào trang web đó cùng một thời điểm từ cùng một máy chủ proxy, máy chủ này chỉ cần gửi một yêu cầu đến Quantrimang.com. Điều này giúp tiết kiệm băng thông của công ty và cải thiện hiệu suất mạng.
* **Bảo mật riêng tư**: Cá nhân và tổ chức cũng sử dụng máy chủ proxy để duyệt Internet riêng tư hơn. Một số máy chủ proxy sẽ thay đổi địa chỉ IP và thông tin nhận dạng khác. Điều này có nghĩa là máy chủ đích không biết ai thực sự đã thực hiện yêu cầu ban đầu, giúp giữ thông tin cá nhân và thói quen duyệt web của bạn riêng tư hơn.
* **Cải thiện bảo mật**: Bạn có thể cấu hình máy chủ proxy để mã hóa yêu cầu web để không ai có thể đọc được giao dịch của bạn. Ngoài ra, người dùng cũng có thể tránh các trang web độc hại thông qua máy chủ proxy. Các tổ chức có thể kết nối máy chủ proxy của họ với mạng riêng ảo (VPN), do đó người dùng từ xa có thể truy cập Internet thông qua proxy của công ty. VPN kết nối trực tiếp đến mạng công ty để có thể kiểm soát và xác minh người dùng của họ có quyền truy cập vào các tài nguyên họ cần (email, dữ liệu nội bộ) đồng thời cũng cung cấp kết nối an toàn cho người dùng để bảo vệ dữ liệu công ty.
* **Truy cập vào các tài nguyên bị chặn**: Máy chủ proxy cho phép người dùng phá vỡ các hạn chế nội dung do công ty hoặc một số tổ chức áp đặt. Nếu truy cập vào trang web bị chặn, bạn có thể đăng nhập vào máy chủ proxy ở nơi khác và xem từ đó. Máy chủ proxy khiến bạn giống như ở Mỹ nhưng thực ra bạn đang ở Việt Nam.

**Nguồn tham khảo:**

[**https://github.com/sameer2800/HTTP-PROXY**](https://github.com/sameer2800/HTTP-PROXY)

[**https://quantrimang.com/ly-thuyet-proxy-la-gi-117220**](https://quantrimang.com/ly-thuyet-proxy-la-gi-117220)

[**https://www.youtube.com/watch?v=PUPDGbnpSjw**](https://www.youtube.com/watch?v=PUPDGbnpSjw)

**và nguồn tài liệu thầy cung cấp**