

Chương 2: Nội dung

2.1. Nguyên lý của ứng dụng mạng

- 2.1.1. Các kiến trúc của ứng dụng mạng
- 2.1.2. Truyền thông giữa các tiến trình
- 2.1.3. Các dịch vụ giao vận

2.2. Web và HTTP

2.3. FTP

2.4. Thư điện tử

2.5. DNS (Domain Name Systems)

2.6. Ứng dụng Peer-to-peer

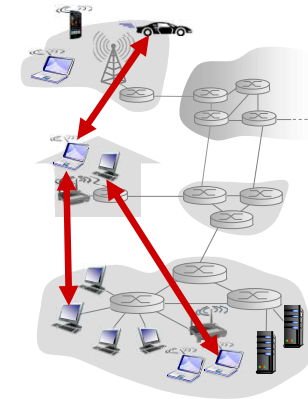
2.7. Video streaming và các mạng phân phối nội dung

2.8. Lập trình socket với UDP và TCP

Tăng ứng dụng 2-77

Kiến trúc mạng P2P thuần túy

- ❖ Không cần phải có server luôn hoạt động
- ❖ Các hệ thống cuối tùy ý kết nối trực tiếp
- ❖ Các peer (các nút mạng) yêu cầu dịch vụ từ các peer khác, cung cấp dịch vụ để đáp ứng lại các peer khác.
 - Khả năng tự mở rộng - các nút cung cấp khả năng dịch vụ mới và nhu cầu dịch vụ mới.
- ❖ Các peer không cần kết nối liên tục vào hệ thống mạng và có thể thay đổi địa chỉ IP.
 - Quản lý phức tạp



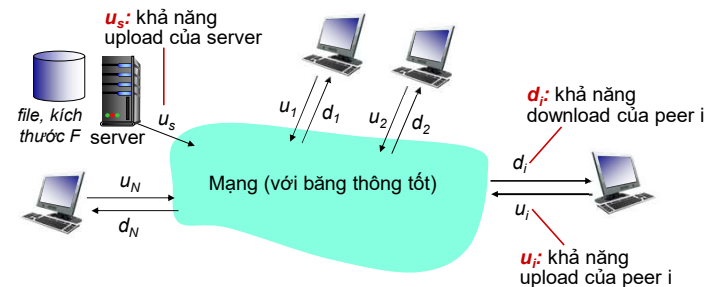
Ví dụ: Chia sẻ file (BitTorrent); Streaming (KanKan); VoIP (Skype)

Tăng ứng dụng 2-78

Chia sẻ file: client-server và P2P

Hỏi: Cần mất bao lâu để truyền (phân phối) file (có kích thước F) từ một server đến N peer?

- Khả năng upload/download (tải lên/tải về) của peer bị giới hạn bởi tài nguyên.



Tăng ứng dụng 2-79

Thời gian truyền file: client-server

- ❖ **Việc truyền của server:** phải gửi tuần tự (upload) N bản sao của file:

- Thời gian gửi một bản sao: F/u_s
- Thời gian gửi N bản sao: NF/u_s

- ❖ **Client:** Mỗi client phải download bản sao của file

- d_{\min} = tốc độ download nhỏ nhất của client
- Thời gian download (ứng với d_{\min}) của client: F/d_{\min}

Thời gian để phân phối F tới N client sử dụng cách tiếp cận client-server

$$D_{c-s} \geq \max\{NF/u_s, F/d_{\min}\}$$

Tăng tuyến tính theo N

Tăng ứng dụng 2-80

Thời gian truyền file: P2P

- ❖ **Việc truyền của server:** phải upload ít nhất một bản sao của file:

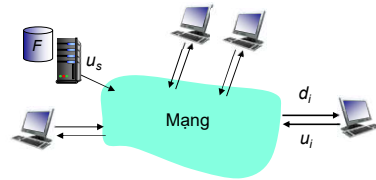
- Thời gian để gửi một bản: F/u_s

- ❖ **Client:** mỗi client phải download bản sao của file

- Thời gian download (ứng với d_{min}) của client: F/d_{min}

- ❖ **Các client:** phải download NF bits

- Tốc độ upload cao nhất (giới hạn) là $u_s + \sum u_i$



Thời gian để phân phối F

tới N client sử dụng
cách tiếp cận P2P

$$D_{P2P} \geq \max\{F/u_s, F/d_{min}, NF/(u_s + \sum u_i)\}$$

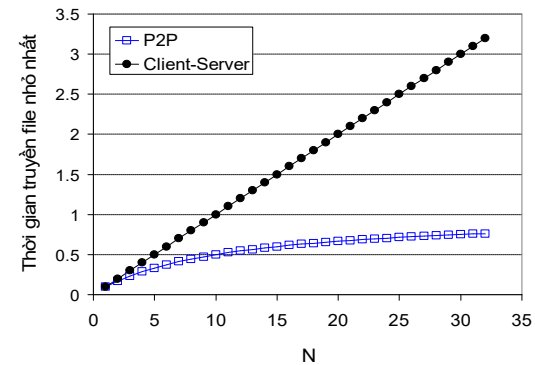
Tăng tuyến tính theo N ...

... nhưng để thực hiện việc này, mỗi peer phải có khả năng phục vụ

Tăng ứng dụng 2-81

Ví dụ so sánh client-server với P2P

Tốc độ upload của client = u , $F/u = 1$ giờ, $u_s = 10u$, $d_{min} \geq u_s$

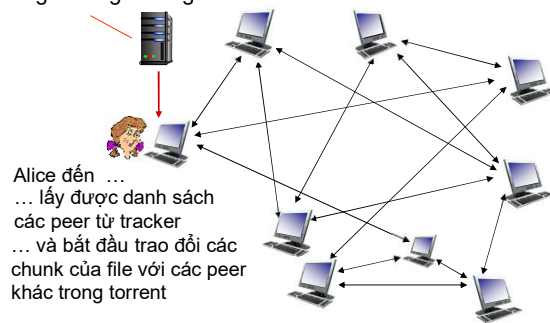


Truyền file P2P: BitTorrent

- ❖ File được chia thành các chunk (phần) có kích thước là 256Kb
- ❖ Các peer trong torrent gửi/nhận các chunk của file

tracker: kiểm tra các peer đang tham gia trong torrent

torrent: nhóm các peer trao đổi các chunk của một file

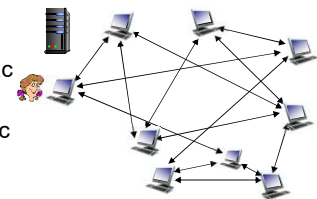


Tăng ứng dụng 2-83

Truyền file P2P: BitTorrent

- ❖ peer tham gia vào torrent:

- Không có các chunk, nhưng sẽ tích lũy chúng qua thời gian từ các peer khác
- Đăng ký với tracker để nhận được danh sách các peer, kết nối với tập con của các peer ("các hàng xóm")



- ❖ Trong khi download, peer sẽ upload các chunk tới các peer khác
- ❖ peer có thể thay đổi các peer mà nó sẽ trao đổi các chunk
- ❖ **churn:** các peer có thể đến hoặc đi
- ❖ Khi peer có được toàn bộ file, nó có thể (ích kỷ) rời đi hoặc (vị tha) ở lại trong torrent

Tăng ứng dụng 2-84

BitTorrent: yêu cầu lấy, gửi các chunk của file

Yêu cầu lấy chunk:

- ❖ Tại bất kỳ thời điểm nào, các peer khác nhau đều có các tập con các chunk khác nhau của file.
- ❖ Theo định kỳ, Alice sẽ hỏi mỗi peer về danh sách các chunk mà họ có
- ❖ Alice sẽ yêu cầu các chunk còn thiếu từ các peer, phần hiếm nhất trước

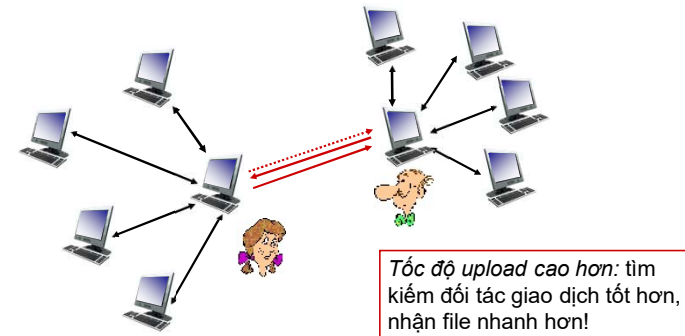
Gửi chunk: tit-for-tat

- ❖ Alice gửi các chunk tới 4 peer hiện đang gửi chunk của cô ấy với **tốc độ cao nhất**
 - Các peer khác đang bị chặn bởi Alice (không nhận được chunk từ cô ấy)
 - Đánh giá lại top 4 sau mỗi 10 giây
- ❖ Cứ mỗi 30 giây: chọn các peer khác một cách ngẫu nhiên, và bắt đầu gửi chunk
 - “Tối ưu hóa không chặn” peer này
 - Peer được chọn mới sẽ được gia nhập vào top 4

Tăng ứng dụng 2-85

BitTorrent: tit-for-tat

- (1) Alice “tối ưu hóa không chặn” Bob
- (2) Alice trở thành một nhà cung cấp trong top 4 của Bob;
- (3) Ngược lại, Bob trở thành một nhà cung cấp trong top 4 của Alice



Tăng ứng dụng 2-86

Chương 2: Nội dung

2.1. Nguyên lý của ứng dụng mạng

- 2.1.1. Các kiến trúc của ứng dụng mạng
- 2.1.2. Truyền thông giữa các tiến trình
- 2.1.3. Các dịch vụ giao vận

2.2. Web và HTTP

2.3. FTP

2.4. Thư điện tử

2.5. DNS (Domain Name Systems)

2.6. Ứng dụng Peer-to-peer

2.7. Video streaming và các mạng phân phối nội dung

2.8. Lập trình socket với UDP và TCP

Tăng ứng dụng 2-87

Video Streaming và CDNs: ngữ cảnh

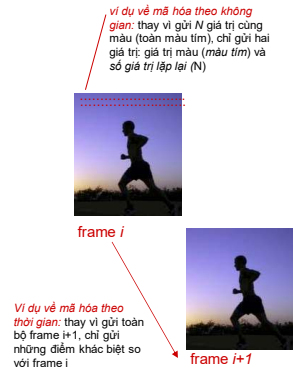
- Lưu lượng streaming video: tiêu tốn lượng lớn băng thông Internet
 - Netflix, YouTube, Amazon Prime: 80% lưu lượng truy nhập ISP khu vực dân cư (2020)
- **Khó khăn:** quy mô – làm thế nào để tiếp cận ~1B người dùng?
- **Khó khăn:** tính không đồng nhất
 - Người dùng khác nhau có những khả năng khác nhau (ví dụ, có dây hoặc di động; giàu băng thông hay nghèo băng thông)
- **Giải pháp:** cơ sở hạ tầng mức ứng dụng, được phân tán



Tăng ứng dụng 2-88

Đa phương tiện: video

- Video: chuỗi hình ảnh được hiển thị ở tốc độ không đổi
 - Ví dụ: 24 hình ảnh/giây
- Hình ảnh kỹ thuật số: mảng pixel
 - mỗi pixel được biểu thị bằng bit
- Mã hóa: sử dụng sự dư thừa **bên trong** và **giữa** các hình ảnh để giảm số bit được sử dụng để mã hóa hình ảnh
 - Không gian (bên trong ảnh)
 - Thời gian (từ một hình ảnh tới hình ảnh tiếp theo)



Tăng ứng dụng 2-89

Đa phương tiện: video

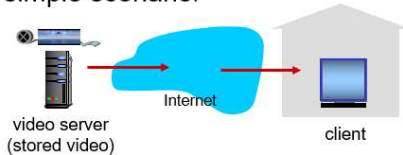
- CBR: (tốc độ bit không đổi):** tốc độ mã hóa video cố định
- VBR: (tốc độ bit thay đổi):** tốc độ mã hóa video thay đổi khi lượng mã hóa theo không gian, thời gian thay đổi
- Ví dụ:**
 - MPEG 1 (CD-ROM) 1.5 Mbps
 - MPEG2 (DVD) 3-6 Mbps
 - MPEG4 (thường được sử dụng trong Internet, 64Kbps – 12 Mbps)



Tăng ứng dụng 2-90

Streaming video được lưu trữ

simple scenario:

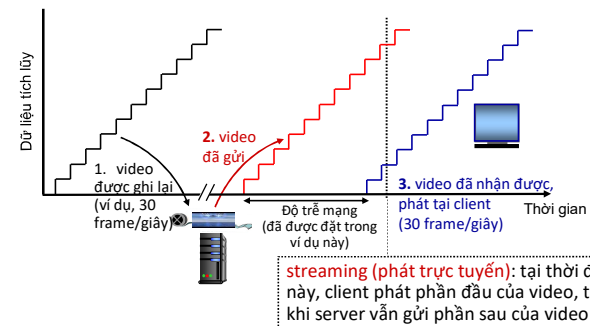


Những khó khăn chính:

- Băng thông từ server-đến-client sẽ **thay đổi** theo thời gian, với mức độ tắc nghẽn mạng thay đổi (trong nhà, mạng truy nhập, lỗi mạng, máy chủ video)
- Mất gói, trễ do tắc nghẽn sẽ làm chậm quá trình phát hoặc dẫn đến chất lượng video kém

Tăng ứng dụng 2-91

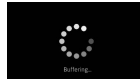
Streaming video được lưu trữ



Tăng ứng dụng 2-92

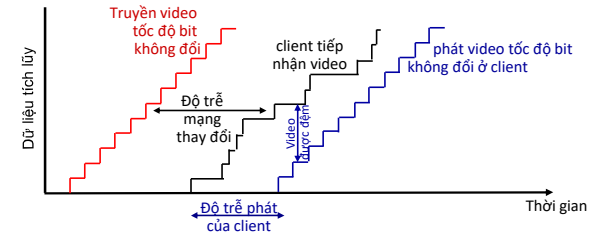
Streaming video được lưu trữ: khó khăn

- **Ràng buộc phát liên tục:** trong khi client phát video, thời gian phát phải khớp với thời gian ban đầu
 - ... nhưng **độ trễ của mạng có thể thay đổi** (jitter), do đó sẽ cần **bộ đệm phía client** để phù hợp với ràng buộc phát liên tục
- Những khó khăn khác:
 - Tương tác client: tạm dừng, tua nhanh, tua lại, chuyển qua video
 - Các gói tin video có thể bị mất, được truyền lại



Tăng ứng dụng 2-93

Streaming video được lưu trữ: đệm phát



- **Bộ đệm phía client và độ trễ phát:** bù cho độ trễ do mạng thêm vào, độ trễ jitter

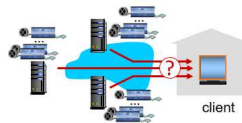
Tăng ứng dụng 2-94

Streaming đa phương tiện: DASH

Dynamic, Adaptive
Streaming over HTTP

server:

- Chia tệp video thành nhiều đoạn
- Mỗi đoạn được mã hóa ở nhiều tốc độ khác nhau
- Mã hóa tốc độ khác nhau được lưu trữ trong các tệp khác nhau
- Các tệp được sao chép trong các nút CDN khác nhau
- **Tệp kê khai:** cung cấp URL cho các đoạn khác nhau.



client:

- Ước tính định kỳ bằng thông từ server-đến-client
- Tự vấn kê khai, yêu cầu từng đoạn tại từng thời điểm
 - Chọn tốc độ mã hóa tối đa bền vững với băng thông hiện tại
 - Có thể chọn tốc độ mã hóa khác nhau tại các thời điểm khác nhau (tùy thuộc vào băng thông có sẵn tại thời điểm đó) và từ các server khác nhau

Tăng ứng dụng 2-95

Streaming đa phương tiện: DASH

- **“thông minh” tại client:** client xác định

- **Khi nào** cần yêu cầu đoạn (để không xảy ra tình trạng thiếu bộ đệm hoặc tràn bộ đệm)
- **Yêu cầu tốc độ mã hóa nào** (chất lượng cao hơn khi có nhiều băng thông hơn)
- **Nơi** yêu cầu đoạn (có thể yêu cầu từ URL server “gần” với client hoặc có băng thông khả dụng cao)



Streaming video = mã hóa + DASH + đệm phát

Tăng ứng dụng 2-96

Các mạng phân phối nội dung (CDNs)

Khó khăn: làm thế nào để truyền tải nội dung (được chọn lọc từ hàng triệu video) tới hàng trăm nghìn người dùng cùng lúc?

- **Lựa chọn 1:** “mega-server” đơn, lớn
 - Một điểm chịu lỗi
 - Điểm tắc nghẽn mạng
 - Đường đi dài (và có thể bị tắc nghẽn) tới các client ở xa

....quá đơn giản: giải pháp này **không mở rộng được quy mô**.

Tăng ứng dụng 2-97

Các mạng phân phối nội dung (CDNs)

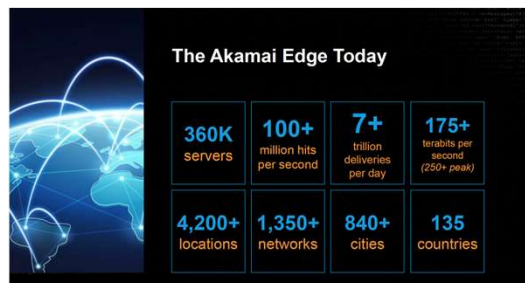
Khó khăn: làm thế nào để truyền tải nội dung (được chọn lọc từ hàng triệu video) tới hàng trăm nghìn người dùng cùng lúc?

- **Lựa chọn 2:** lưu trữ/phục vụ nhiều bản sao video tại nhiều trang web được phân phối theo địa lý
 - **Đi vào sâu (enter deep):** đẩy các server CDN vào sâu bên trong nhiều mạng truy nhập
 - Gần với người dùng
 - Akamai: 240,000 server được triển khai tại > 120 quốc gia (2015)
 - **Mang về nhà (bring home):** số lượng nhỏ hơn (khoảng 10) cụm lớn hơn trong các POP gần các mạng truy nhập
 - Được sử dụng bởi Limelight



Tăng ứng dụng 2-98

Akamai ngày nay:

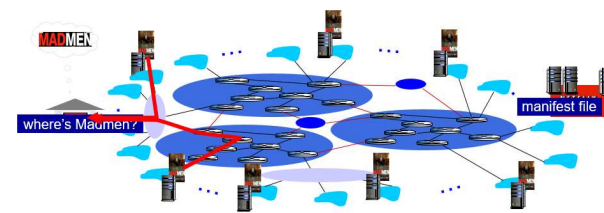


Nguồn: <https://networkingchannel.eu/living-on-the-edge-for-a-quarter-century-an-akamai-retrospective-downloads/>

Tăng ứng dụng 2-99

Netflix hoạt động như thế nào?

- Netflix: lưu trữ các bản sao nội dung (ví dụ: MADMEN) tại các nút OpenConnect CDN (trên toàn thế giới) của nó
- Thuê bao yêu cầu nội dung, nhà cung cấp dịch vụ trả về bảng kê khai
 - bằng cách sử dụng bảng kê khai, khách hàng sẽ truy xuất nội dung ở tốc độ có thể hỗ trợ cao nhất
 - có thể chọn tốc độ khác hoặc sao chép nếu đường dẫn mạng bị tắc nghẽn



Tăng ứng dụng 2-100

Các mạng phân phối nội dung (CDNs)



Khó khăn OTT: ứng phó với tình trạng tắc nghẽn Internet từ "biên"

- Nội dung nào cần đặt trong nút CDN nào?
- Từ nút CDN nào để truy xuất nội dung? Ở tốc độ nào?