Chương 3.1:
Internet và các ứng dụng đa phương tiện

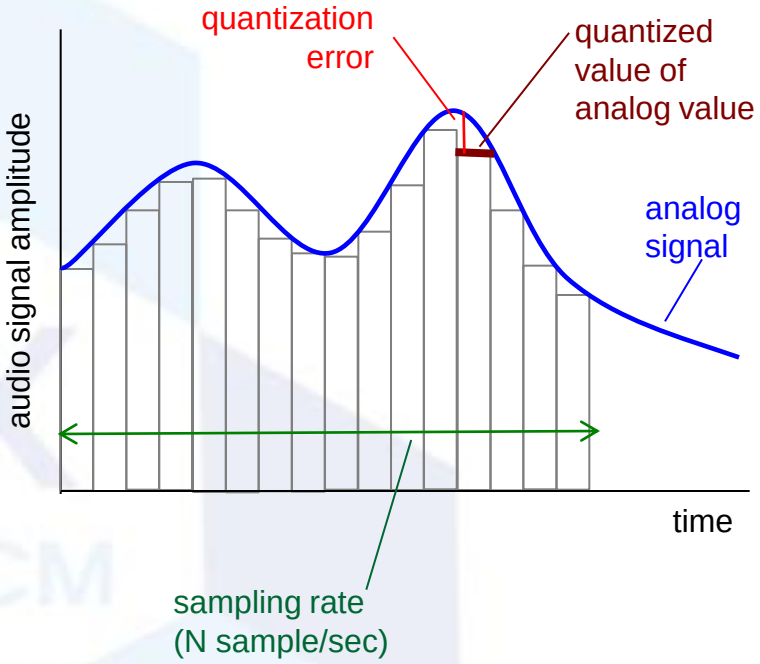
ThS. NGUYỄN CAO ĐẠT
E-mail:dat@hcmut.edu.vn

Nội dung

- Các ứng dụng đa phương tiện
- Truyền tải video/audio đã được lưu trữ
- Thoại trên IP (VoIP)

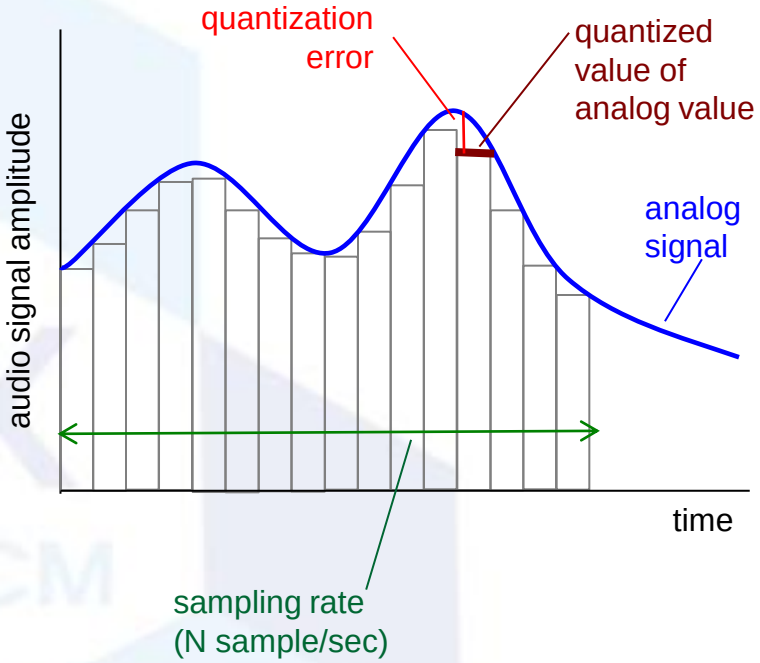
Các ứng dụng đa phương tiện

- **Audio**
  - Tín hiệu âm thanh tương tự được lấy mẫu ở tốc độ không đổi
    - Điện thoại: 8.000 mẫu/s
    - CD: 44.100 mẫu/s
  - Mỗi mẫu được lượng tử
    - Mỗi giá trị lượng tử biểu diễn bằng một chuỗi các bit
    - Ví dụ 8 bit, 2^8=256 giá trị lượng tử.



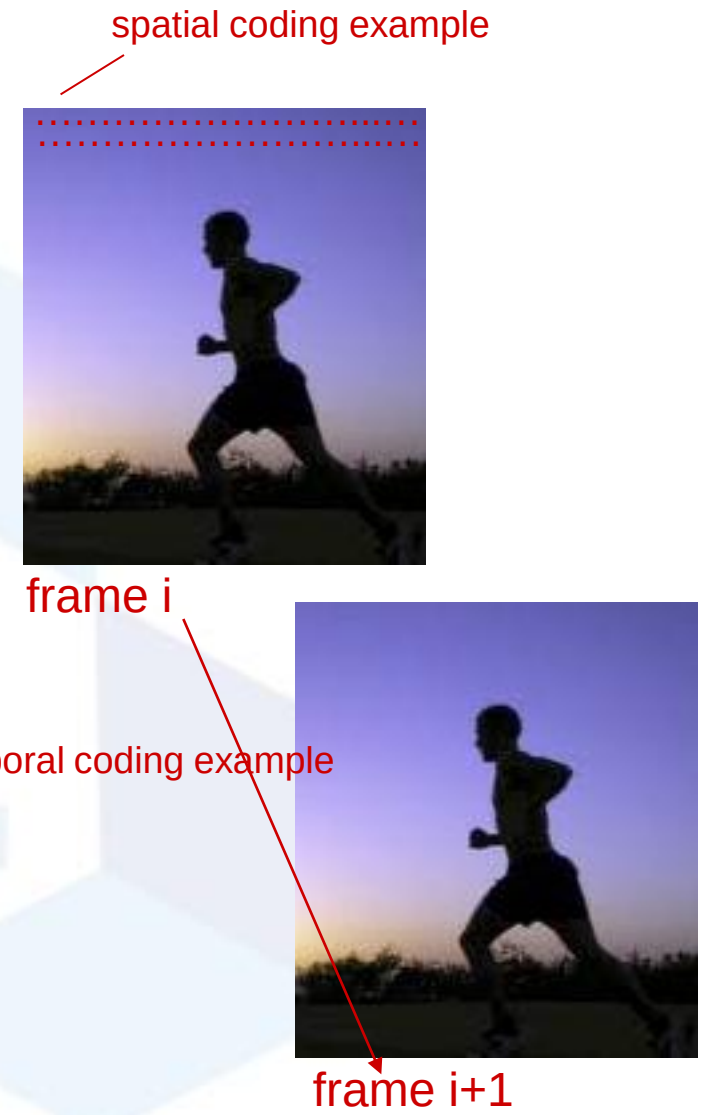
Các ứng dụng đa phương tiện

- **Audio**
  - Ví dụ 8.000 mẫu/s, 256 giá trị lượng tử, tốc độ là: 8000 x 8 = 64000 kbps
  - Bên nhận chuyển các bit trở lại tín hiệu tương tự
  - Có suy giảm chất lượng khi chuyển đổi
  - Một số tốc độ truyền âm thanh
    - CD: 1.411 Mbps
    - MP3: 96, 128, 160 kbps
    - Điện thoại trên Internet: 5,3 kbps hoặc cao hơn



Các ứng dụng đa phương tiện

- **Video**
  - Chuỗi các ảnh hiển thị ở tốc độ không đổi. Ví dụ 30 frame/s.
  - Ảnh số là mảng các điểm. Mỗi điểm được thể hiện bằng một chuỗi bit.
  - Mã hóa(coding): giảm kích thước ảnh bằng cách loại bỏ các dư thừa về không gian(spatial) bên trong từng ảnh, thời gian(temporal) khi chuyển từ ảnh này sang ảnh khác



Các ứng dụng đa phương tiện

- **Video**
  - CBR: (constant bit rate): Tốc độ mã hóa video cố định
  - VBR: (variable bit rate): Tốc độ mã hóa thay đổi
  - Tốc độ thông thường
    - MPEG 1 (CD-ROM): 1.5 Mbps
    - MPEG2 (DVD): 3-6 Mbps
    - MPEG4 (thường dùng trên Internet): < 1 Mbps

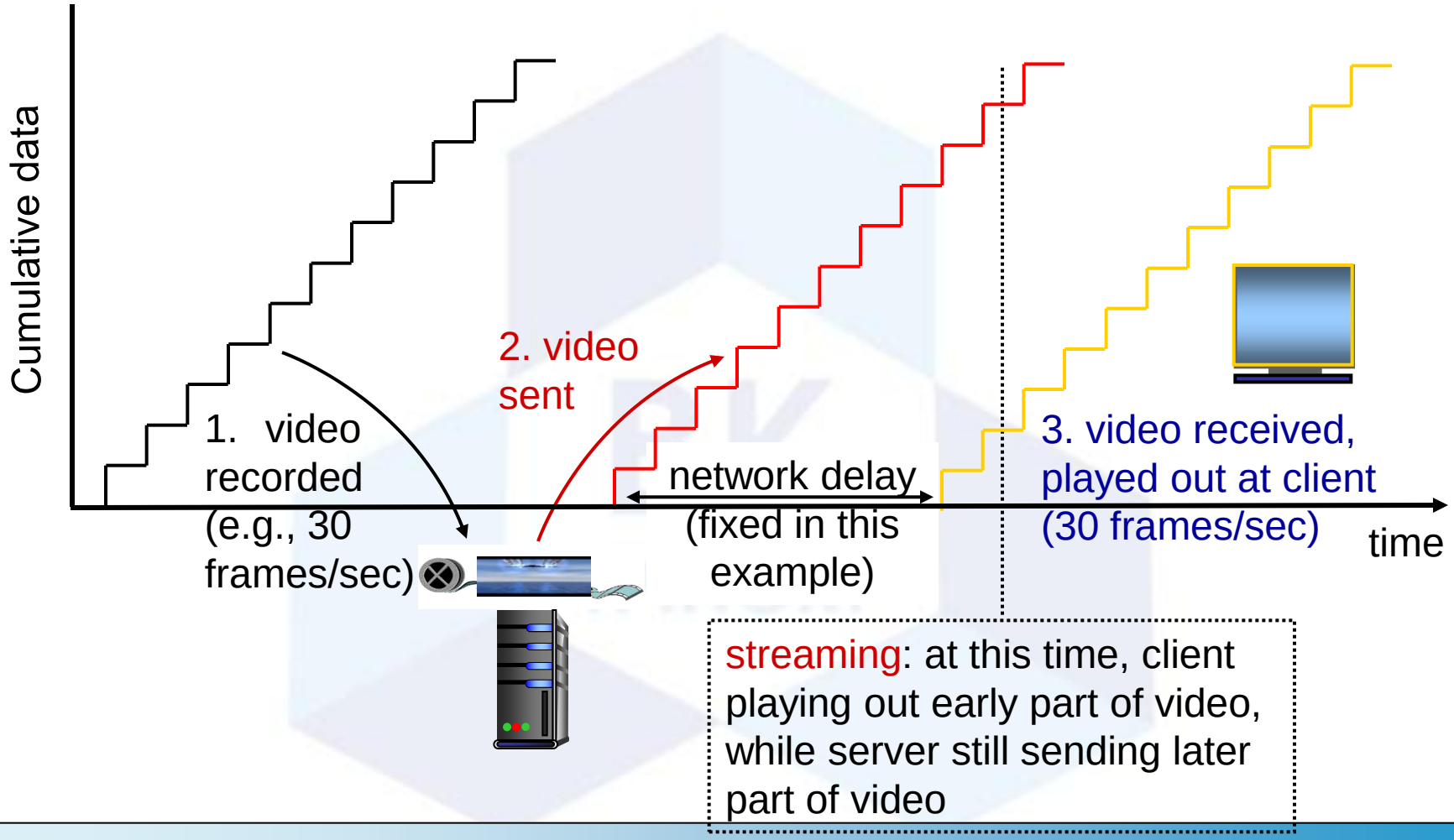
Các ứng dụng đa phương tiện

- **Phân loại**
  - **Truyền tải video/audio đã được lưu trữ**
    - *Streaming stored* audio, video
    - *Streaming:* Có thể bắt đầu phát trước khi toàn bộ tập tin tải về
    - *Stored :* Lưu trữ toàn bộ tại máy chủ, và cũng được lưu trữ/đệm tại client
    - Ví dụ: Youtube
  - **Trò chuyện (voice/video) trên IP**
    - *conversational* voice/video over IP
    - Tương tác giữa người và người
    - Ví dụ như Skype
  - ***Phát Audio/video trực tuyến***
    - *streaming live* audio, video

Nội dung

- Các ứng dụng đa phương tiện
- Truyền tải video/audio đã được lưu trữ
- Thoại trên IP (VoIP)

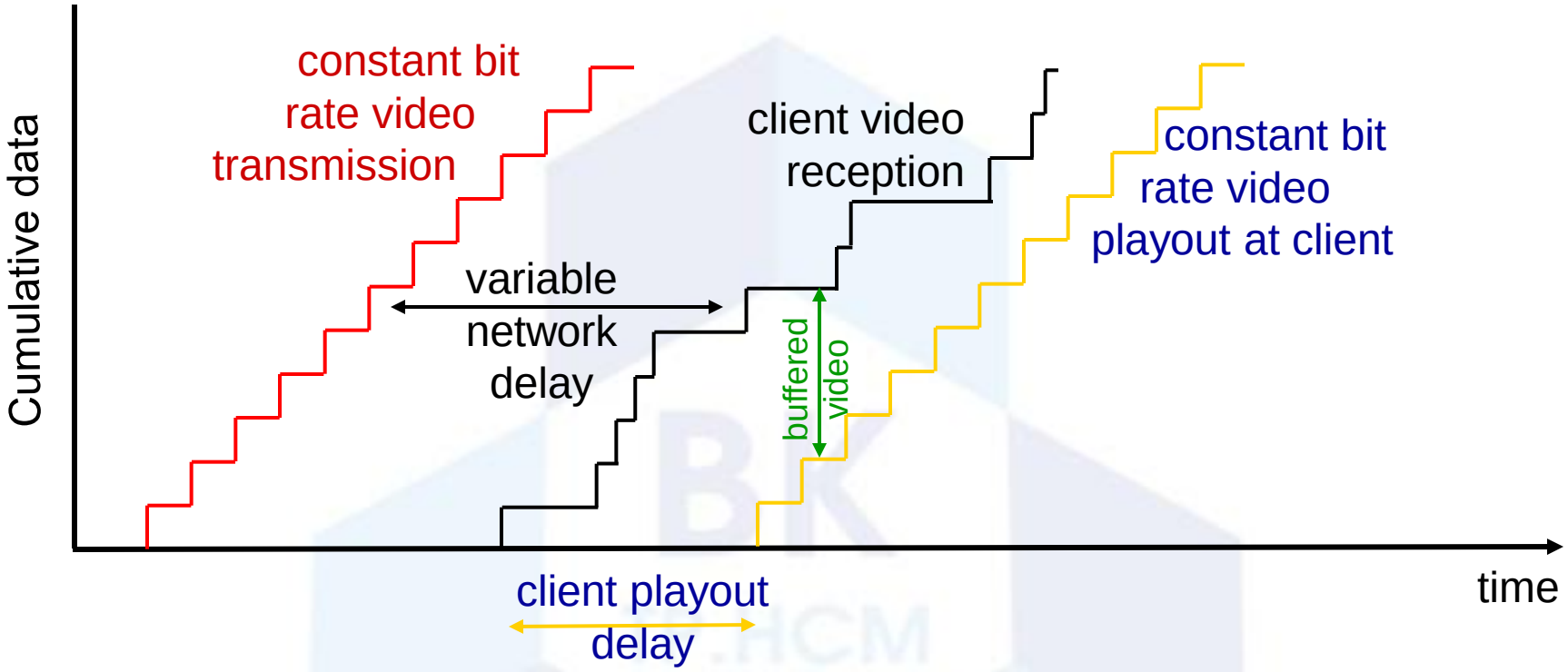
Truyền tải video/audio đã được lưu trữ



Truyền tải video/audio đã được lưu trữ

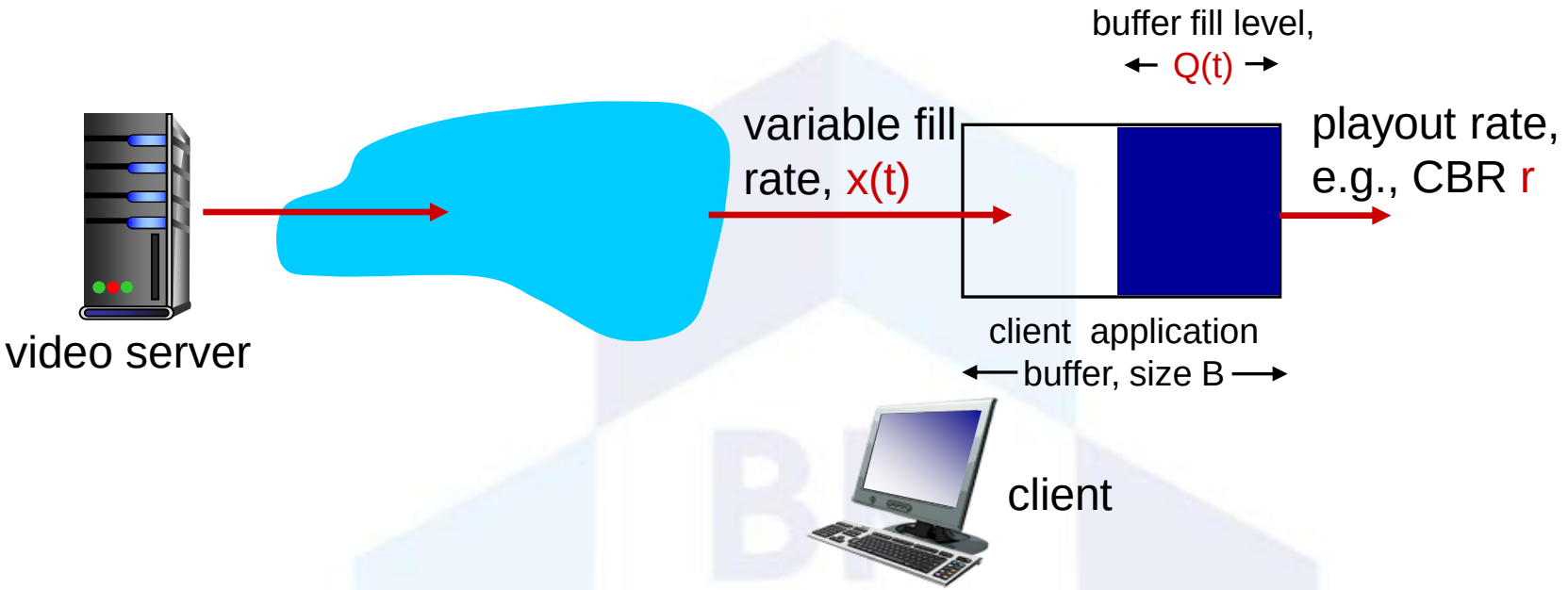
- **Các thách thức**
  - Liên tục phát sóng nhưng độ trễ mạng thay đổi vì vậy cần đệm ở phía client và phát trể
  - Tương tác: tạm dừng, tua đi, tua lại, nhảy đến một vị trí
  - Các gói có thể mất, truyền lại

Truyền tải video/audio đã được lưu trữ

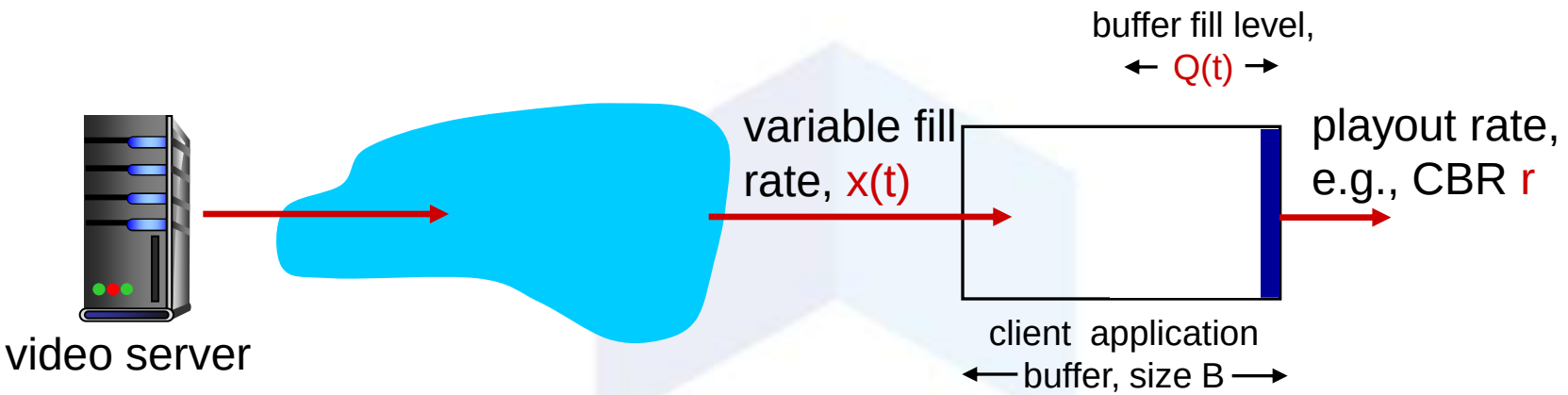


- *Đệm ở phía client và phát trể* nhằm bù đắp độ trể mạng

Đệm ở phía client và phát trể

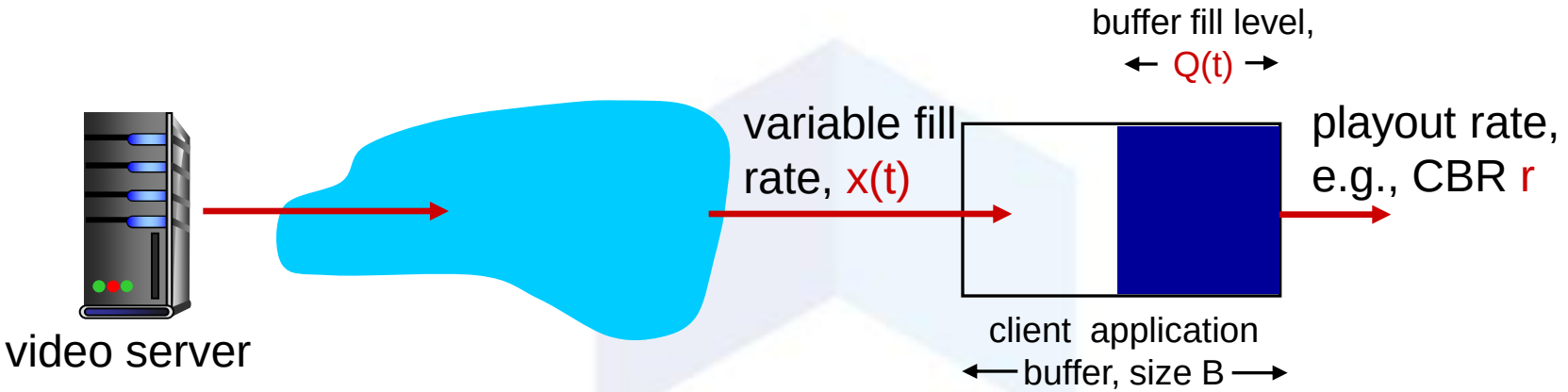


Đệm ở phía client và phát trể



1. Làm đầy bộ đệm cho đến khi bắt đầu phát tại t_p
2. Bắt đầu phát tại t_p,
3. Mức độ làm đầy bộ đệm thay đổi theo thời gian với tốc độ lấp đầy x(t) thay đổi và tốc độ phát r là hằng số

Đệm ở phía client và phát trể



- *Tốc độ lấp đầy trung bình là $\bar{x}$*
  - $\bar{x} < r$: Bộ đệm sẽ bị trống sau một thời gian
  - $\bar{x} > r$: Bộ đệm không bị trống nếu phát trể đủ lớn

Truyền tải video/audio đã được lưu trữ

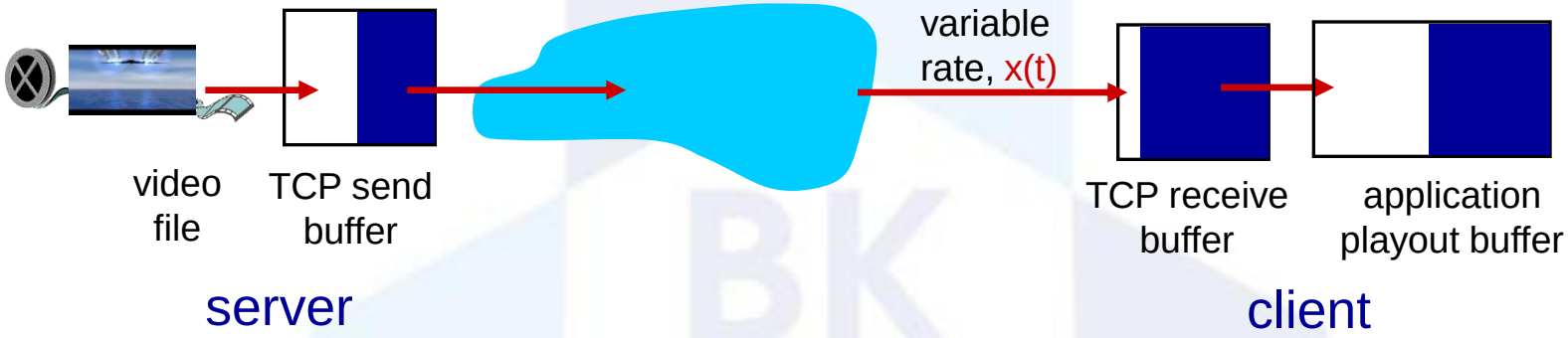
- **Phân loại**
  - Truyền tải dùng UDP
  - Truyền tải dùng HTTP
  - Truyền tải thích nghi

Truyền tải dùng UDP

- Server gửi với một tốc độ thích hợp
- Thông thường tốc độ gửi không đổi
- Tốc độ truyền tải có thể thấy rõ mức độ tắc nghẽn
- Trể phát ngắn(2-5 giây) để loại bỏ độ trể mạng
- Khắc phục lỗi ở mức ứng dụng và trong thời gian cho phép
- RTP [RFC 2326], RTSP [RFC 2326], : được trình bày trong chương 3.2
- UDP có thể không qua được bức tường lửa

Truyền tải dùng HTTP

- Lấy tập tin qua HTTP GET
- Gửi ở tốc độ tối đa có thể dưới TCP



- Tốc độ lấp đầy biến động do điều khiển tắc nghẽn và truyền lại

Truyền tải dùng HTTP

- Dùng *HTTP byte-range header* trong thông điệp yêu cầu HTTP GET
  - Tìm nạp trước nhằm giải quyết tốc độ truyền biến động
  - Kết thúc sớm hoặc định vị lại
- Dể dàng đi qua bức tường lửa

Truyền tải thích nghi

- *DASH: D*ynamic, *A*daptive *S*treaming over *H*TTP
- *server*
  - Mã hóa video với nhiều phiên bản khác nhau
  - Mỗi phiên bản có tốc độ tương ứng với chất lượng dịch vụ
  - Chia tập tin ở mỗi phiên bản thành nhiều khúc(chunk)
  - *Tập tin manifest :* cung cấp URLs cho các khúc khác nhau
- *client*
  - Yêu cầu một vài khúc của nhiều phiên bản khác nhau
  - Chọn khúc có tốc độ cao nhất phù hợp với thông lượng hiện hành
  - Tham vấn manifest để yêu cầu một khúc ở mỗi thời điểm

Truyền tải thích nghi

- "Thông minh" tại client:
  - Khi nào yêu cầu khúc mới(đệm trống nhiều hoặc tràn không xảy ra)
  - Tốc độ bit được yêu cầu(chất lượng cao hơn khi có thông lượng cao hơn)
  - Vị trí nơi yêu cầu lấy khúc mới(có thể yêu cầu từ máy chủ mà URL đó là "gần" với gần hoặc có thông lượng cao)

Mạng phân phối nội dung (CDN)

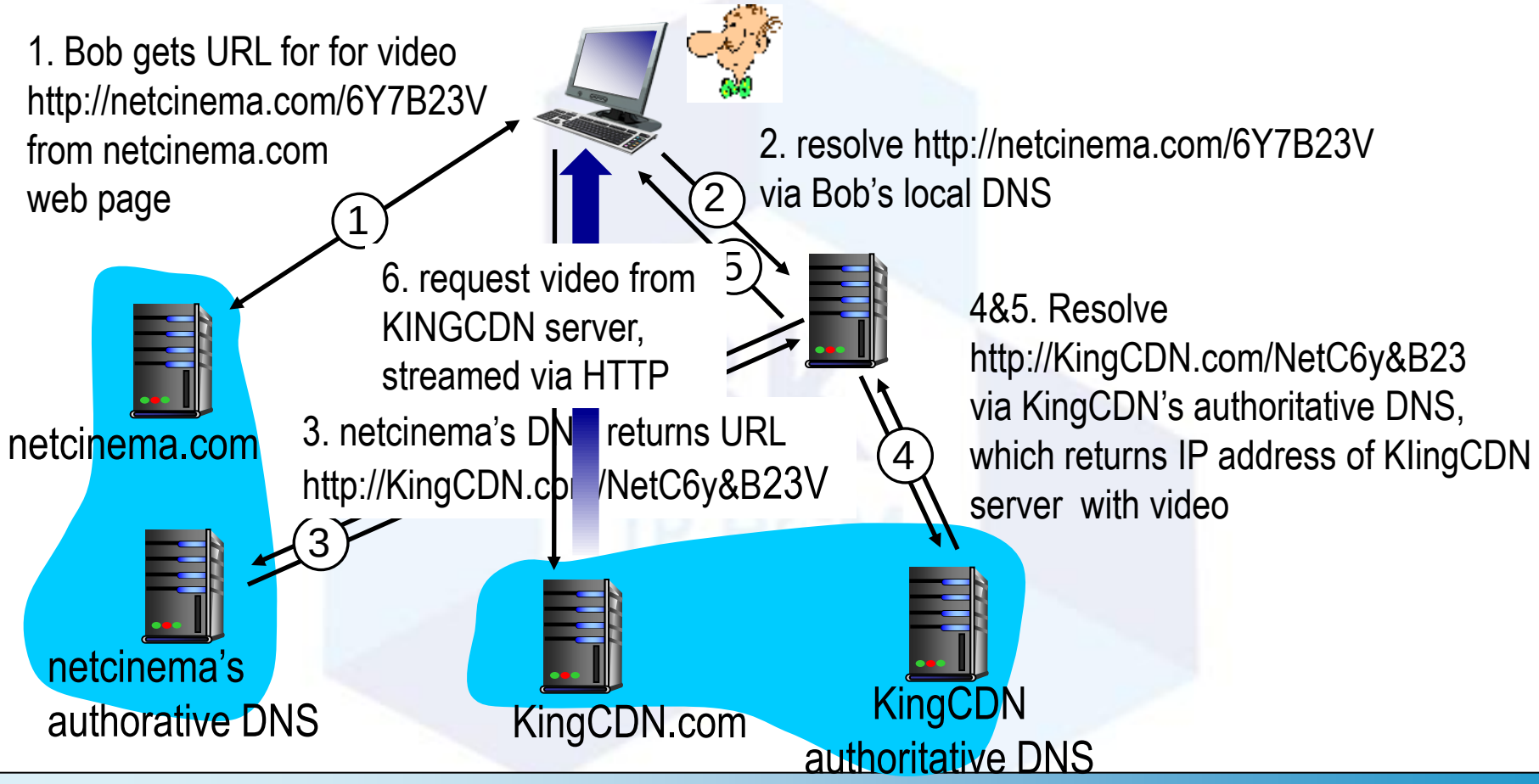
- Content distribution networks
- *Thách thức:* Làm thế nào truyền nội dung(được chọn từ hàng triệu video) đến hàng trăm ngàn người dùng đồng thời
- *Lựa chọn 1:* máy chủ đơn, "siêu máy chủ"
  - Điểm đơn sự cố
  - Điểm đơn tắc nghẽn mạng
  - Xa đối với với một số client
  - Nhiều bản sao của video được gửi ra
  - Giải pháp này không có tính khả mở

Mạng phân phối nội dung (CDN)

- *Lựa chọn 2:* Lưu trữ/phục vụ nhiều bản sao của các video tài nhiều site phân bố theo địa lý *(CDN)*
  - *Thâm nhập sâu:* Đưa các CDN server vào sâu trong nhiều mạng truy nhập
    - Gần với người dùng
    - Được dùng bởiAkamai với 1700 vị trí
  - *Gần nhà (bring home):* số lượng nhỏ hơn của các cụm máy tính lớn trong các POPs gần với (nhưng không phải trong) các mạng truy nhập
    - Được sử dụng bởi Limelight

Mạng phân phối nội dung (CDN)

Bob (client) yêu cầu video ở http://netcinema.com/6Y7B23V
và video được lưu trữ trong CDN ở http://KingCDN.com/NetC6y&B23V



Chiến lược lựa chọn cụm CDN

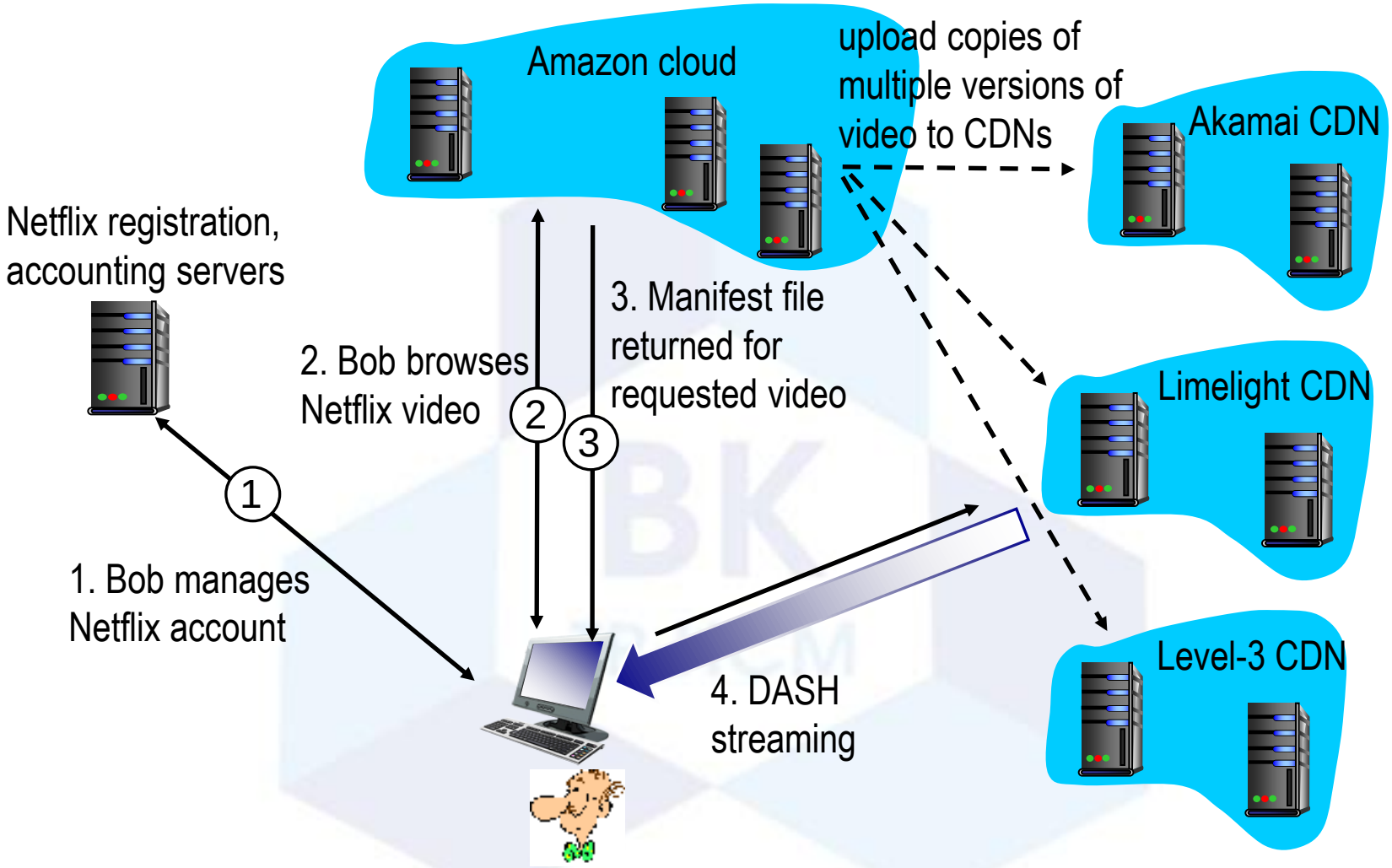
- *Thách thức:* Làm thế nào CDN DNS lựa chọn node CDN "tốt" để truyền video đến client
  - Lấy node CDN gần nhất về vật lý đối với client
  - Lấy node CDN với độ trể nhỏ nhất (số hop nhỏ nhất) đến client(các node CDN định kỳ phing đến các ISP truy cập, báo kết quả về cho CDN DNS)
  - IP anycast[RFC 1546].

- *Thay thế:* cho phép client quyết định
  - Cho client một danh sách các node CDN
  - Client ping các node và lấy nod "tốt nhất"

Tình huống xem xét - Netflix

- 30% lưu lượng truyền tải ở Mỹ vào năm 2011
- Cơ sở hạ tầng sở hữu rất ít, dùng các dịch vụ của các bên thứ 3:
  - Chỉ có các máy chủ đăng ký và thanh toán
  - Dùng dịch vụ Cloud của Amazon
    - Netflix cập nhật bản chỉnh đến Amazon cloud
    - Tạo nhiều phiên bản với chất lượng khác nhau trong cloud
    - Cập nhật các phiên bản từ cloud đến các CDN
    - Cloud chứa các trang web của Netflix
  - CDNs chứa nội dung là của Akamai, Limelight, Level-3

Tình huống xem xét - Netflix



Nội dung

- Các ứng dụng đa phương tiện
- Truyền tải video/audio đã được lưu trữ
- Thoại trên IP (VoIP)

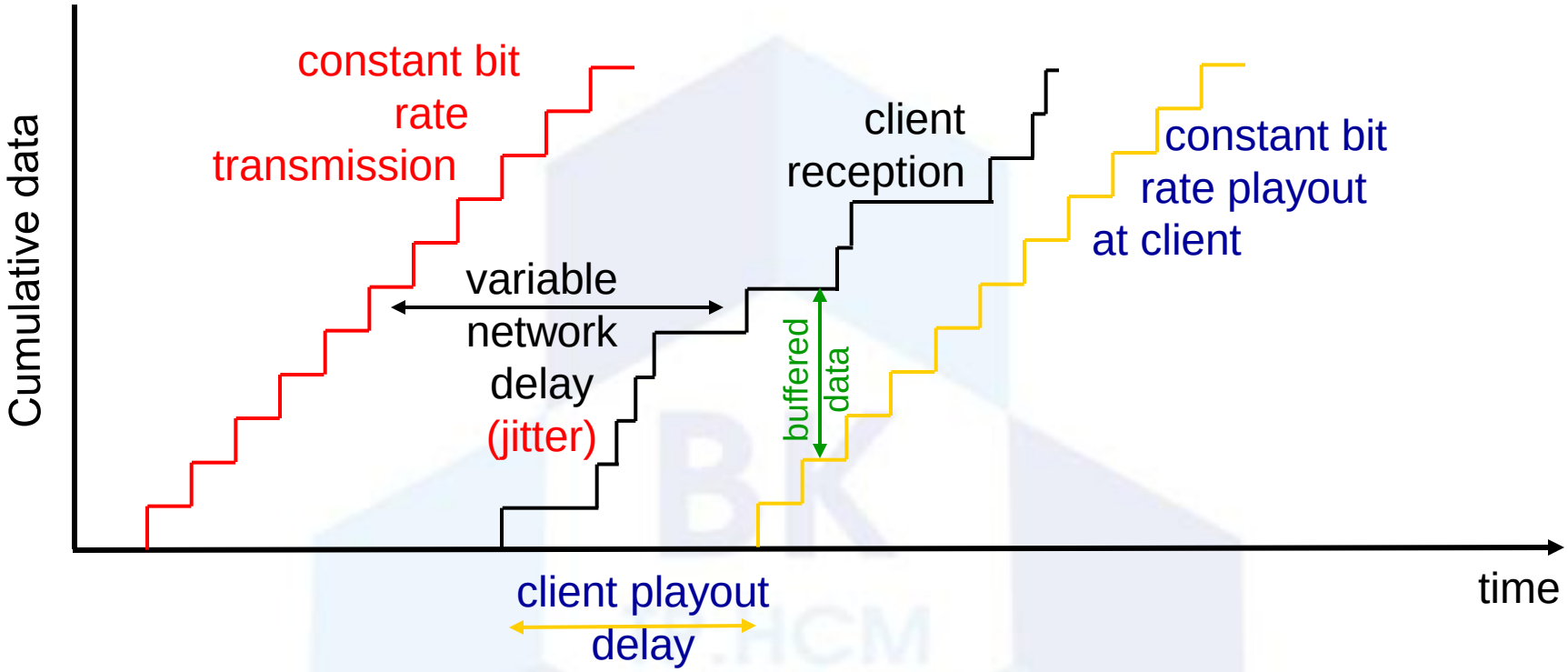
Thoại trên IP (VoIP)

- Giải quyết các hạn chế của dịch vụ IP
  - *Mất gói:* IP datagram mất do tắc nghẽn mạng, khả năng mất gói phụ thuộc vào cơ chế mã hóa, 1 – 10% là chấp nhận được
  - **Trể từ đầu cuối đến đầu cuối**
    - VoIP cần thiết phải duy trì "đàm thoại"
    - Quan tâm nhiều đến độ trể bao gồm cả trể ở mức ứng dụng và trể mạng (jitter)
    - < 150 msec: tốt ; > 400 msec: kém
- Khởi tạo phiên làm việc: làm thế nào người dùng thông báo địa chỉ IP, chỉ số cổng, giải thuật encode
- Các dịch vụ giá trị gia tăng: chuyển tiếp cuộc gọi, ghi âm, ..

Thoại trên IP (VoIP)

- **Cách thực hiện**
  - Xen kẻ giữa nói chuyện và im lặng
    - 64kbps trong khi nói chuyện
    - Gói tin chỉ được tạo trong khi nói chuyện
    - Lấy mẫu từng đoạn 20 msec với 64kbps: 160 bytes
  - Phần mào đầu ở tầng ứng dụng được thêm vào mỗi đoạn
  - Toàn bộ dữ liệu (chunk+header) đóng gói vào UDP hoặc phân đoạn TCP
  - Ứng dụng gởi phân đoạn đến socket mỗi 20 msec trong khi nói chuyện

Jitter



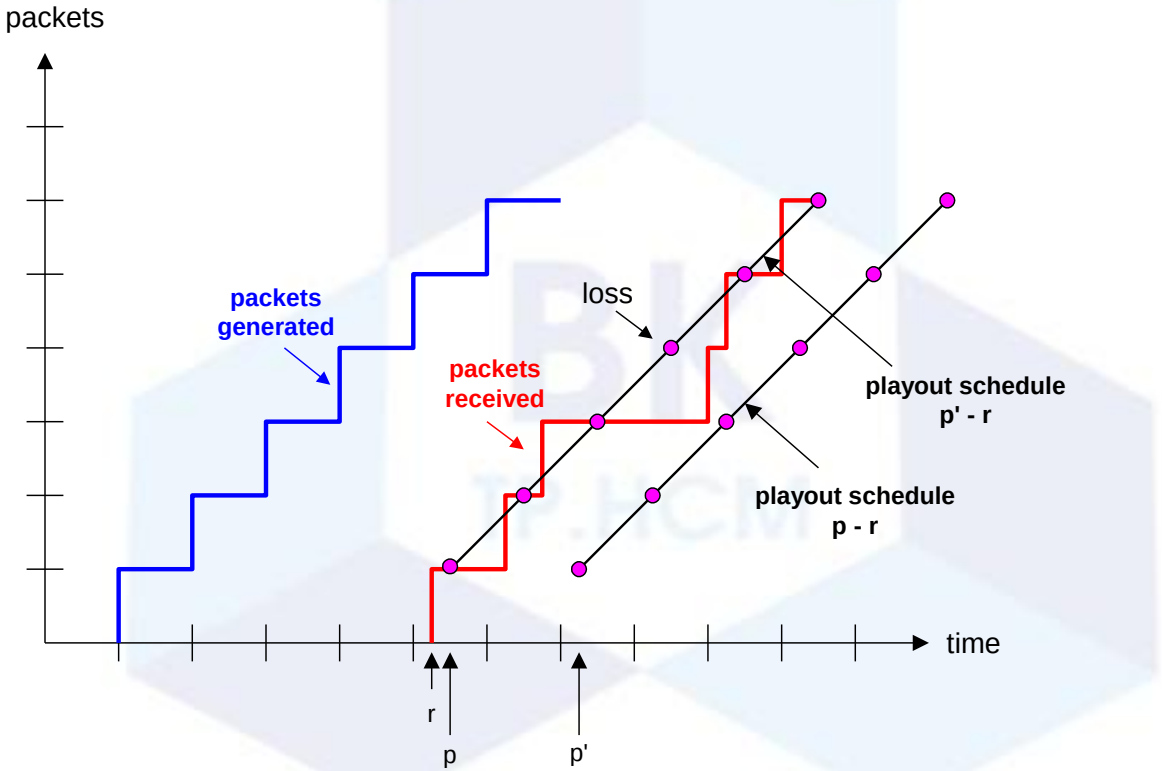
- Trễ đầu cuối đến đầu cuối của hai gói liên tiếp: sự khác biệt có thể được nhiều hơn hoặc ít hơn 20 ms (truyền thời gian khác nhau)

VoIP: phát trể cố định

- Bên nhận phát trể cố định q ms
  - Đoạn có nhãn thời gian là t sẽ đượcphátt ại t+q
  - Nếu đoạn có nhãn thời gian là t nhưng đến sau t+q -> dữ liệu đến quá trể ("mất" dữ liệu)
- Lựa chọn q
  - *q lớn:* ít "mất" dữ liệu
  - *q nhỏ:* tương tác tốt hơn nhưng "mất" dữ liệu

VoIP: phát trễ cố định

- Bên gửi tạo gói mỗi 20 ms trong khi nói chuyện
- Gói đầu tiên đến ở thời điểm r
- Lịch phát bắt đầu tại p: "mất" dữ liệu
- Lịch phát bắt đầu tại p' tốt hơn



VoIP: phát trể thích nghi

- *Mục tiêu:* phát trể thấp, "mất" dữ liệu thấp
- *Hướng tiếp cận:* điều chỉnh phát trể thích nghi
  - Ước tính trể mạng, điều chỉnh phát trể ở bắt đầu mỗi lúc có nói chuyện
  - Các khoảng thời gian im lăng được nén hoặc kéo dài
  - Các đoạn vẫn chunks still played out every 20 msec during talk spurt
- Trể gói ước tính thích nghi (tương tự TCP RTT)

$$d_i = (1-\alpha)d_{i-1} + \alpha\,(r_i - t_i)$$

delay estimate after ith packet

small constant, e.g. 0.1

time received - time sent (timestamp)

measured delay of ith packet

VoIP: phát trể thích nghi

- Ước tính độ lệch trung bình của sự chậm trể

$$v_i = (1-\beta)v_{i-1} + \beta\,|r_i - t_i - d_i|$$

- d_i, v_i được tính toán cho mỗi gói được nhận nhưng chỉ chỉ được dùng ở bắt đầu mỗi lúc có nói chuyện

- Gói đầu tiên trong mỗi lúc có nói chuyện sẽ có thời điểm phát:

$$\text{playout-time}_i = t_i + d_i + Kv_i$$

- Các gói còn lại trong lúc có nói chuyện được phát định kỳ

VoIP: phát trể thích nghi

- Làm thế nào để bên nhận xác định gói nào là gói đầu tiên trong mỗi lúc có nói chuyện ?
- Nếu không có mất gói bên nhận xem xét khác biệt giữa hai nhãn thời gian. Nếu khác biệt > 20 ms thì đó là gói đầu tiên trong mỗi lúc có nói chuyện
- Nếu có mất mát, bên nhận phải xem xét cả nhãn thời gian và chỉ số tuần tự. Nếu khác biệt > 20 ms và chỉ số tuần tự không có gián đoạn thì đó là gói đầu tiên trong mỗi lúc có nói chuyện

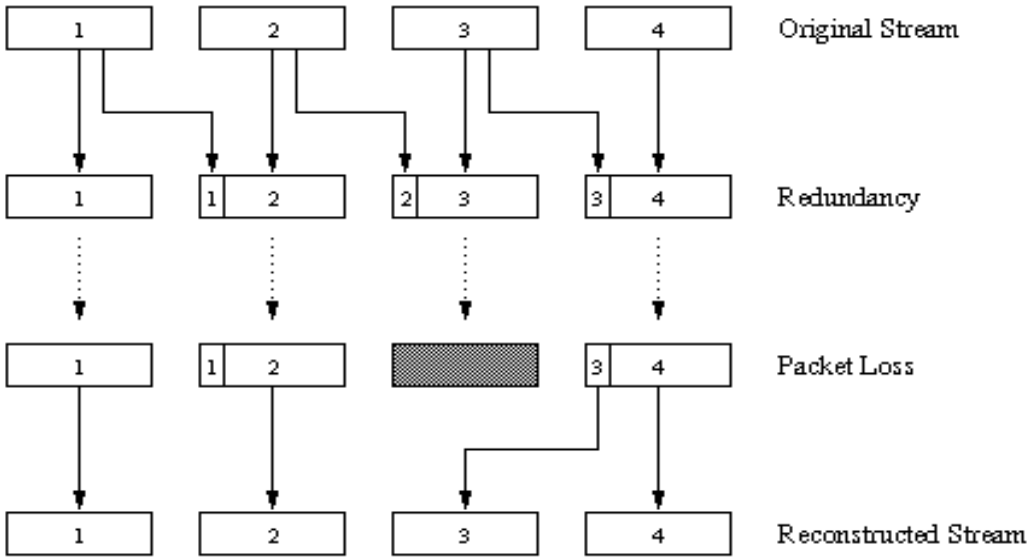
VoIP: Phục hồi do mất gói

- *Thách thức:* phục hồi do mất gói tin được chấp nhận nếu chậm trễ nhỏ giữa truyền dẫn ban đầu và phát sóng
  - Mỗi ACK/NAK ~ một RTT
- Thay thế: dùng sửa lỗi (*Forward Error Correction -FEC)*
  - Gửi đủ bit để phục hồi mà không phải truyền lại
- *FEC đơn giản*
- Với mỗi nhóm n đoạn(chunk), tạo ra đoạn dư thừa bằng cách XOR trên n đoạn ban đầu
- Gửi n + 1 khối, tăng chi phí là 1/n
- Có thể tái tạo lại cả nhóm n đoạn nếu bị mất nhiều nhất là một đoạn từ n + 1 đoạn

VoIP: Phục hồi do mất gói

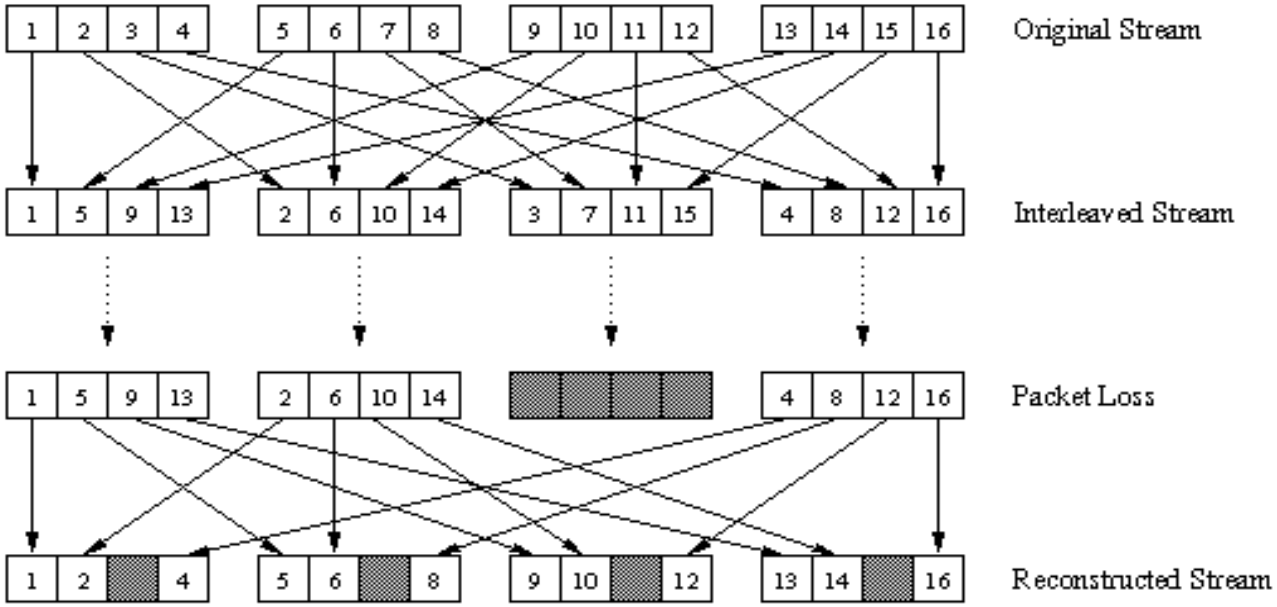
Một lược đồ FEC khác

- "piggyback dòng chất lượng thấp hơn"
- Gửi dòng chất lượng thấp hơn như thông tin dư thừa
- Ví dụ dòng PCM ở tốc độ 64 kbps và dòng dư thừa GSM ở 13 kbps



- Không liên tiếp mất, bên nhận có thể che giấu sự mất mát
- Có thể tổng quát hóa lên

VoIP: Phục hồi do mất gói



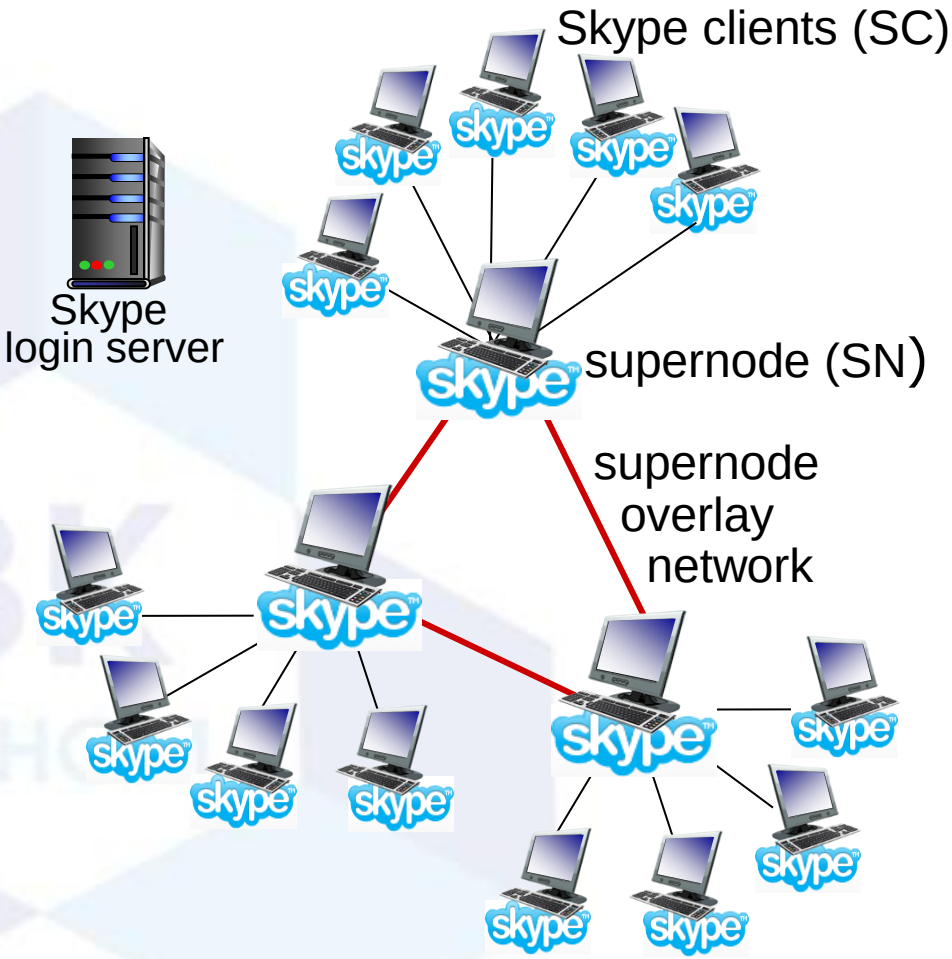
Đan xen để che giấu mất mát:

- Đoạn chia thành nhiều đơn vị nhỏ hơn (ví dụ mỗi đơn vị 5ms
- Gói chứa các đơn vị từ các đoạn khác nhau

- Nếu mất gói vẫn còn gần như đoạn gốc
- Không tăng chi phí do dư thừa nhưng tăng trễ phát

VoIP: Skype

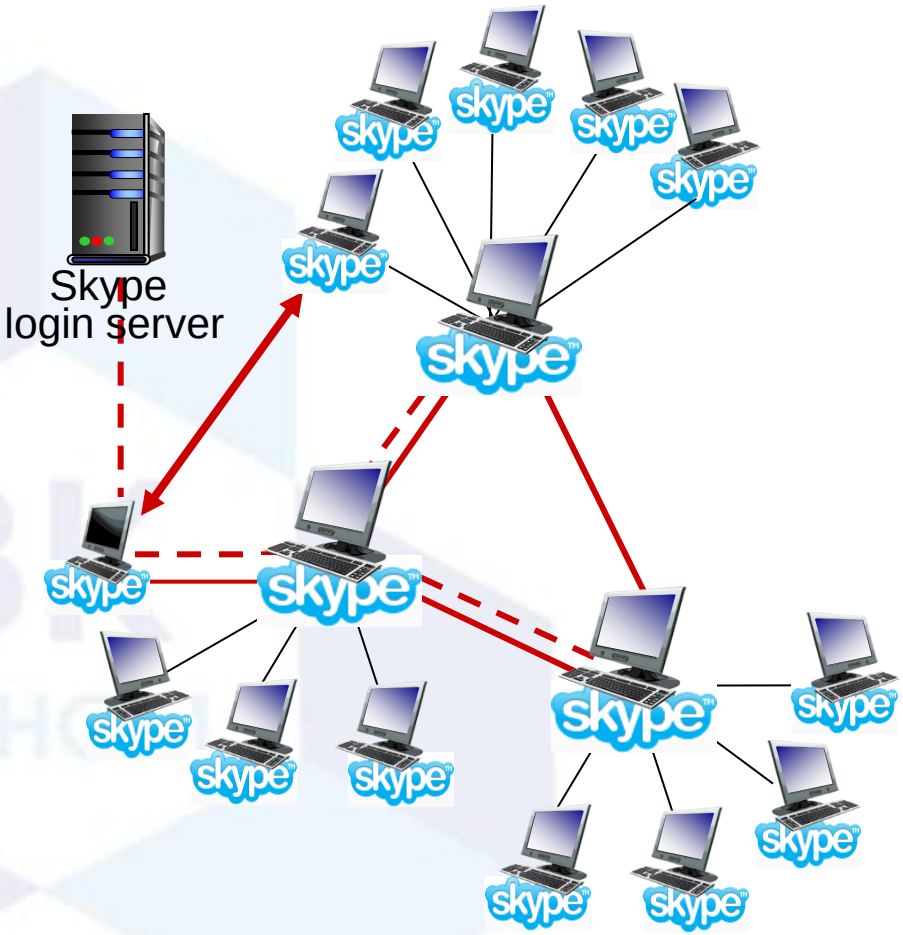
- Giao thức ứng dụng độc quyền
- Các thành phần P2P
  - clients: skype kết nối ngang hàng trực tiếp với nhau cho VoIP
  - super nodes (SN): Các skype ngang hàng có những chức năng đặc biệt
  - overlay network
  - login server



VoIP: Skype

Hoạt động ở skype client

1. Tham gia mạng skype bằng cách liên hệ SN (địa chỉ IP được lưu trữ) sử dụng giao thức TCP
2. login đến skype login server
3. Lấy địa chỉ IP của danh sách bạn bè từ SN hoặc SN overlay
4. Khởi tạo cuộc gọi trực tiếp đến bạn bè



VoIP: Skype

- *Vấn đề:* cả Alice, Bob phía sau "NAT"
- Giải pháp chuyển tiếp
  - Alice, Bob duy trì kết nối đến các SN
  - Alice báo hiệu với SN của cô để kết nối đến Bob
  - SN của Alice kết nối đến SN của Bob
  - SN của Bob kết nối đến Bob trên kết nối đã được khởi tạo giữa Bob và SN này