



TRƯỜNG ĐẠI HỌC VINH
VIỆN KỸ THUẬT VÀ CÔNG NGHỆ

BÀI GIẢNG

MẠNG MÁY TÍNH



TS. Lê Văn Minh

Bộ môn Hệ thống và Mạng máy tính

Viện Kỹ thuật và Công nghệ, Trường Đại học Vinh

Email: minhlv@vinhuni.edu.vn

Home page: <http://home.vinhuni.edu.vn/minhlv>

Chương 6. Mạng đa phương tiện



□ Chuẩn đầu ra:

- *G2.5. Giải thích nguyên lý hoạt động của các ứng dụng truyền thông đa phương tiện.*

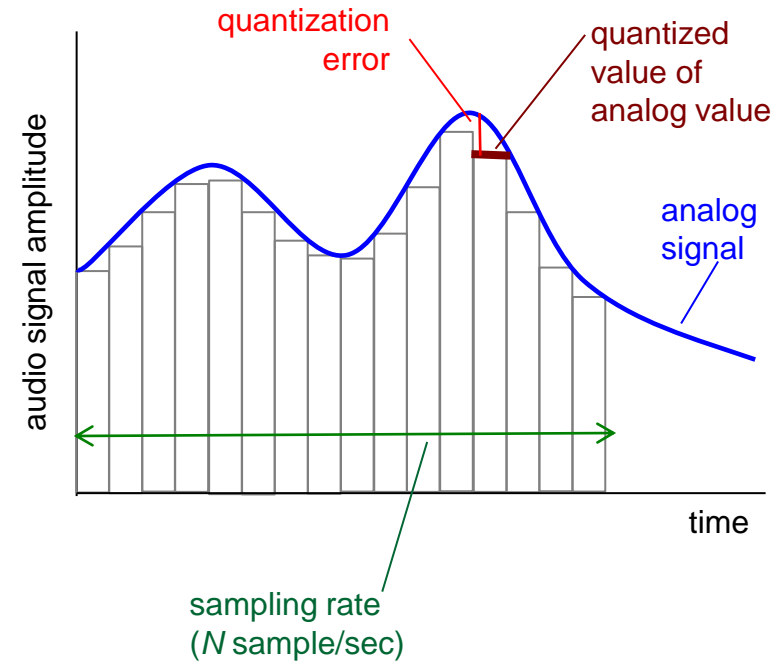
Chương 6. Mạng đa phương tiện



- ❑ 6.1. Ứng dụng mạng đa phương tiện
- ❑ 6.2. Truyền phát video đã lưu trữ
- ❑ 6.3. Truyền giọng nói qua IP
- ❑ 6.4. Giao thức cho các ứng dụng đàm thoại thời gian thực
- ❑ 6.5. Hỗ trợ của mạng cho dữ liệu đa phương tiện

Multimedia: dữ liệu âm thanh

- Tín hiệu âm thanh tương tự được lấy mẫu với tốc độ cố định
 - Điện thoại: 8,000 mẫu/giây
 - CD nhạc: 44,100 mẫu/giây
- Mỗi mẫu đều được lượng tử hóa, ví dụ
 - Làm tròn trong $2^8=256$ giá trị
 - Mỗi giá trị lượng tử được biểu diễn bởi các bits, ví dụ 8 bits cho 256 values

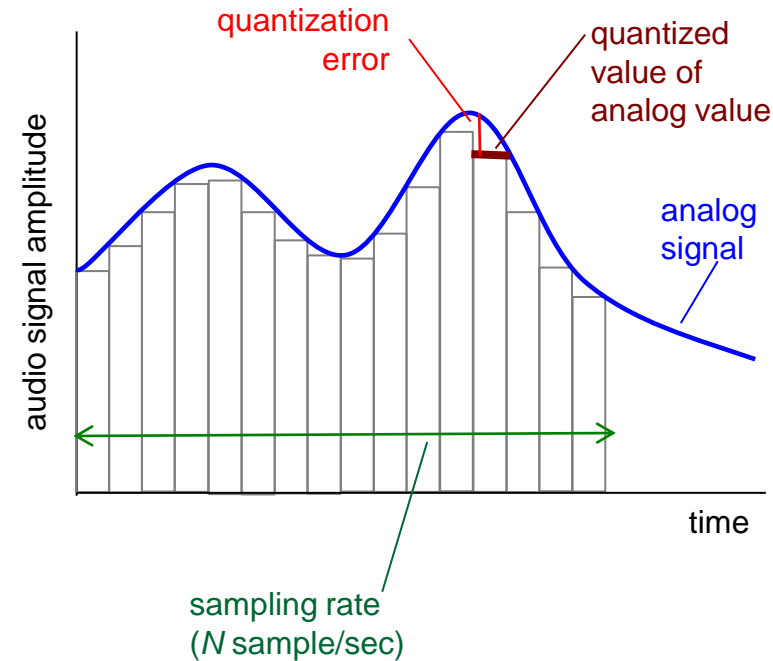


Multimedia: dữ liệu âm thanh

- Ví dụ: 8,000 mẫu/giây, 256 giá trị lượng tử: 64,000 bps
- Bên nhận chuyển các bits trở lại tín hiệu tương tự:
 - Giảm bớt một ít chất lượng

Ví dụ tốc độ

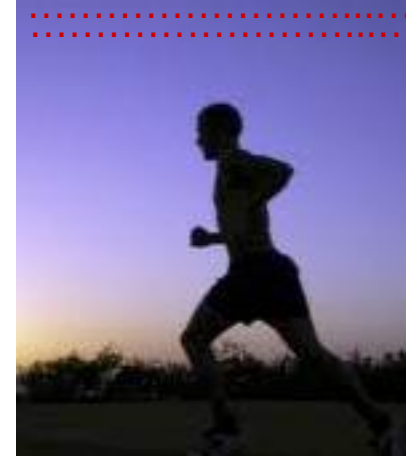
- CD: 1.411 Mbps
- MP3: 96, 128, 160 kbps
- Internet telephony: 5.3 kbps



Multimedia: video

- video: dãy các hình ảnh được hiển thị với tốc độ cố định
 - Ví dụ 24 hình/giây
- Ảnh số: ma trận các điểm
 - Mỗi điểm biểu diễn bởi các bits
- Mã hóa: sử dụng sự dư thừa *trong* và *giữa* các ảnh để giảm số bits dùng mã hóa ảnh
 - Không gian (trong ảnh)
 - Thời gian (từ một ảnh đến ảnh tiếp theo)

spatial coding example: instead of sending N values of same color (all purple), send only two values: color value (*purple*) and number of repeated values (N)



frame i

temporal coding example: instead of sending complete frame at $i+1$, send only differences from frame i

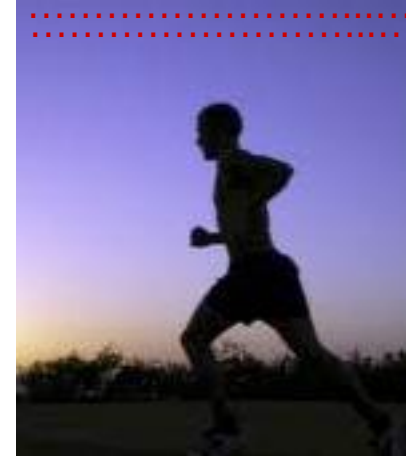


frame $i+1$

Multimedia: video

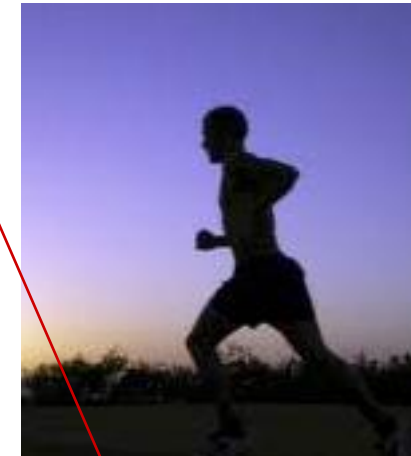
- **CBR: (constant bit rate):**
video mã hóa với tốc độ cố định
- **VBR: (variable bit rate):**
video mã hóa với tốc độ thay đổi
- **Ví dụ:**
 - MPEG I (CD-ROM) 1.5 Mbps
 - MPEG2 (DVD) 3-6 Mbps
 - MPEG4 (often used in Internet, < 1 Mbps)

spatial coding example: instead of sending N values of same color (all purple), send only two values: color value (*purple*) and number of repeated values (N)



frame i

temporal coding example: instead of sending complete frame at $i+1$, send only differences from frame i

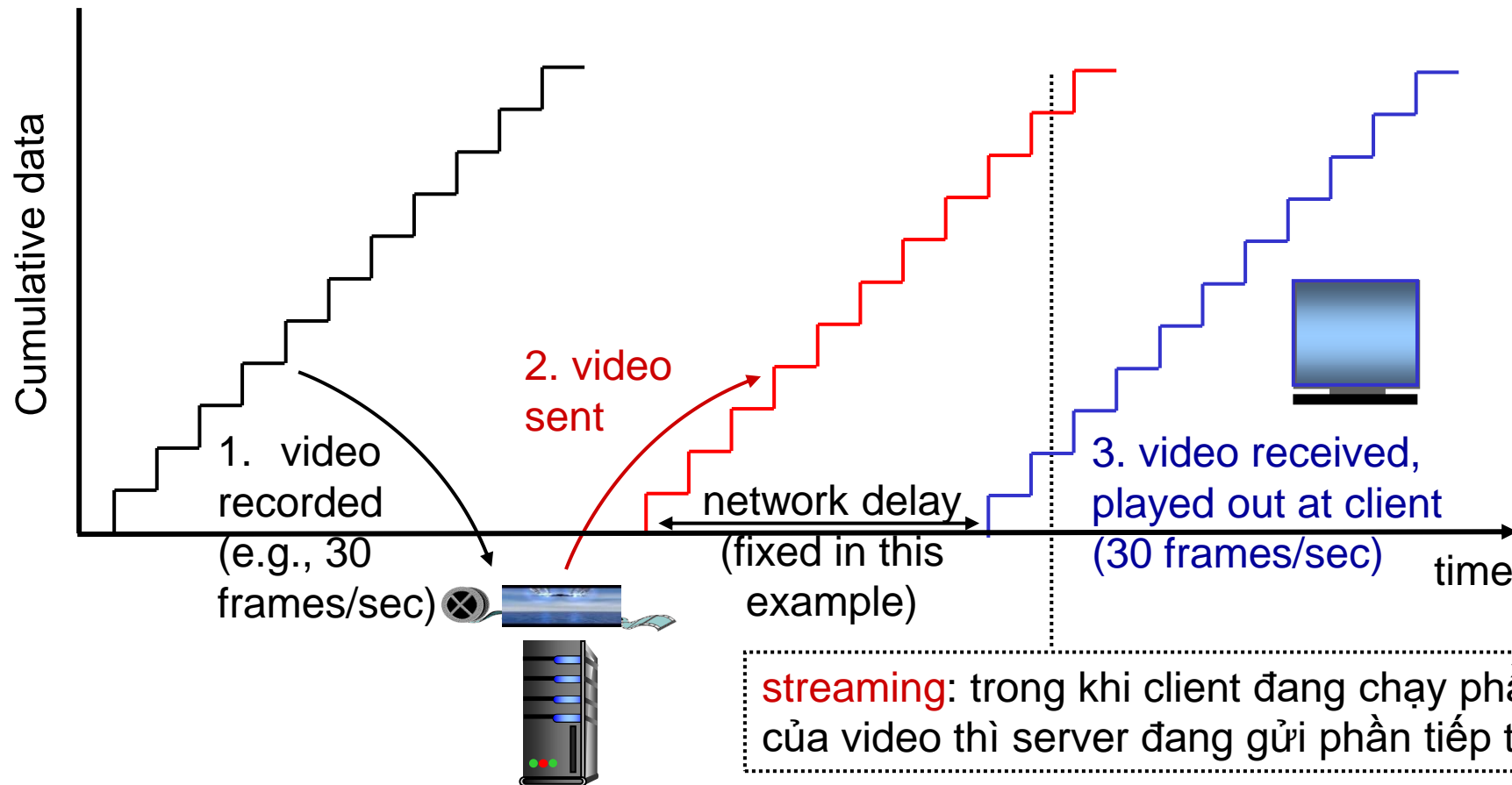


frame $i+1$

Mạng đa phương tiện: 3 dạng ứng dụng

- **phát** audio, video **đã lưu trữ**
 - **phát**: có thể bắt đầu phát trước khi tải xuống toàn bộ tệp
 - **Lưu trữ (tại server)**: có thể truyền audio/video nhanh hơn hiển thị ở bên nhận (sẽ storing/buffering tại client)
 - Ví dụ YouTube, Netflix, Hulu
- **Hội thoại** voice/video over IP
 - Cuộc hội thoại giữa người với người
 - Ví dụ Skype
- **Phát trực tiếp** audio, video
 - Ví dụ truyền hình trực tiếp các sự kiện thể thao (bóng đá)

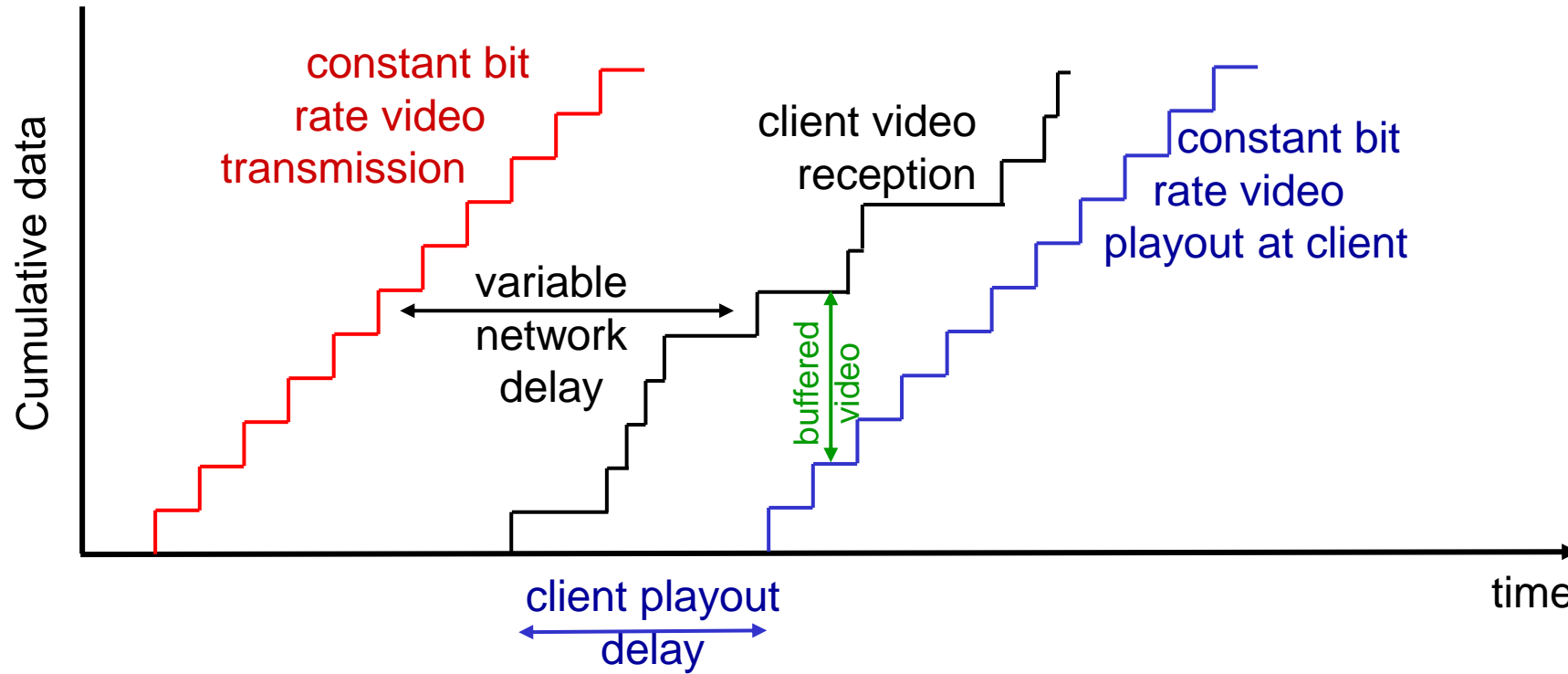
Truyền phát video:



Phát video đã lưu trữ: những thách thức

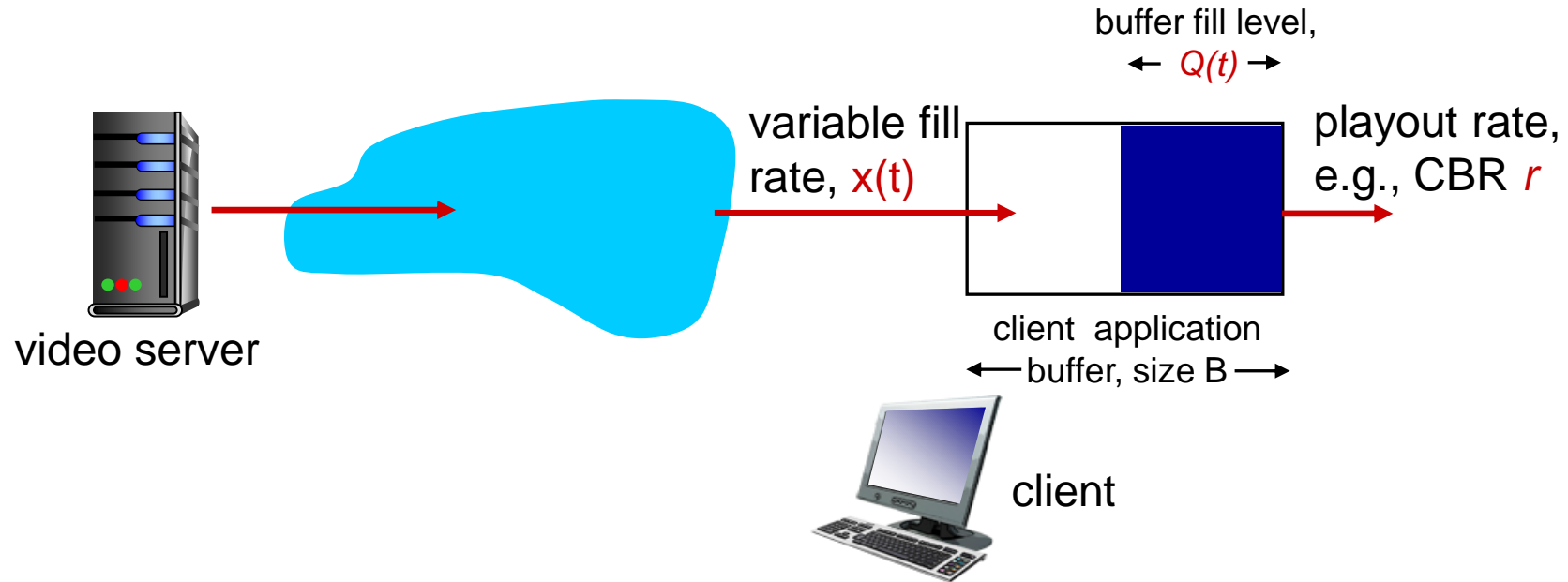
- **Ràng buộc về tính liên tục khi phát:** khi client bắt đầu thì phải phát lại đúng như thời gian gốc
 - ... nhưng **độ trễ mạng thay đổi**, do vậy sẽ cần **client-side buffer** để đáp ứng yêu cầu hiển thị
- **Các thách thức khác:**
 - Tương tác phía client: tạm dừng, chuyển tiếp nhanh, tua lại, nhảy tới video
 - Các gói video có thể bị mất, truyền lại

Phát video đã lưu trữ : revisited

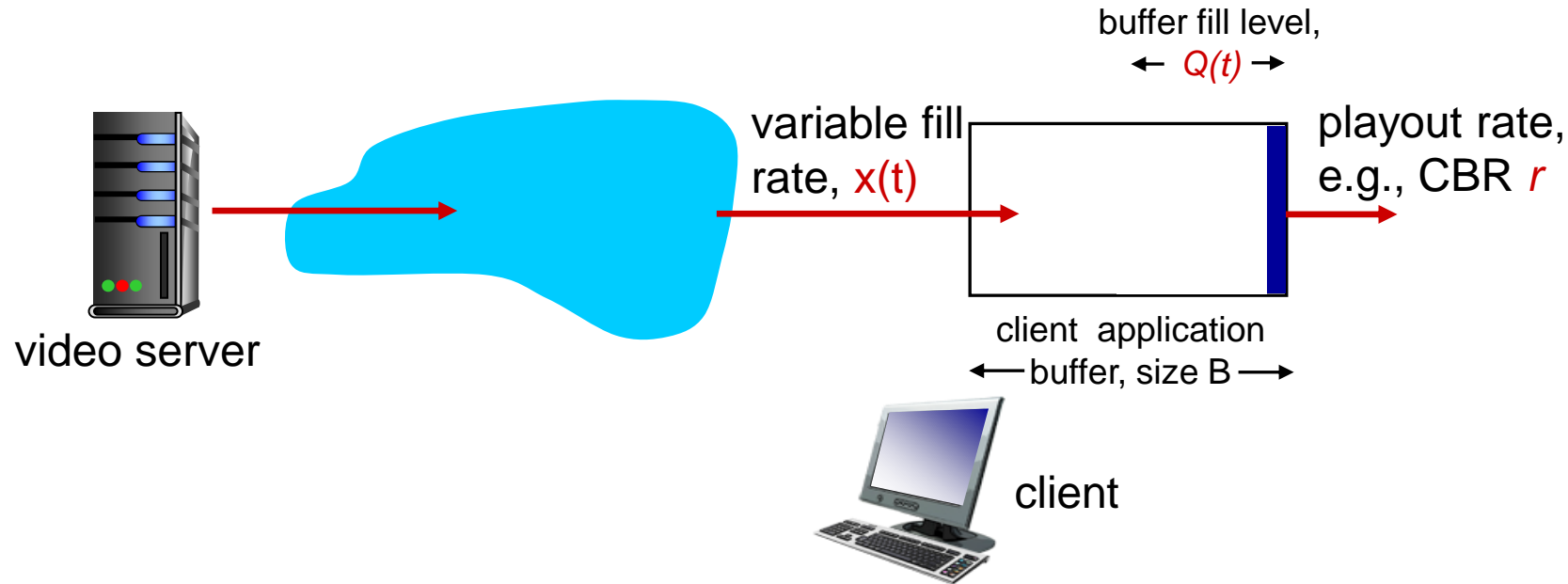


- *client-side buffering và trễ hiển thị*: bù cho độ trễ mạng và trễ xử lý

Bộ đệm tại Client, giải phóng bộ đệm

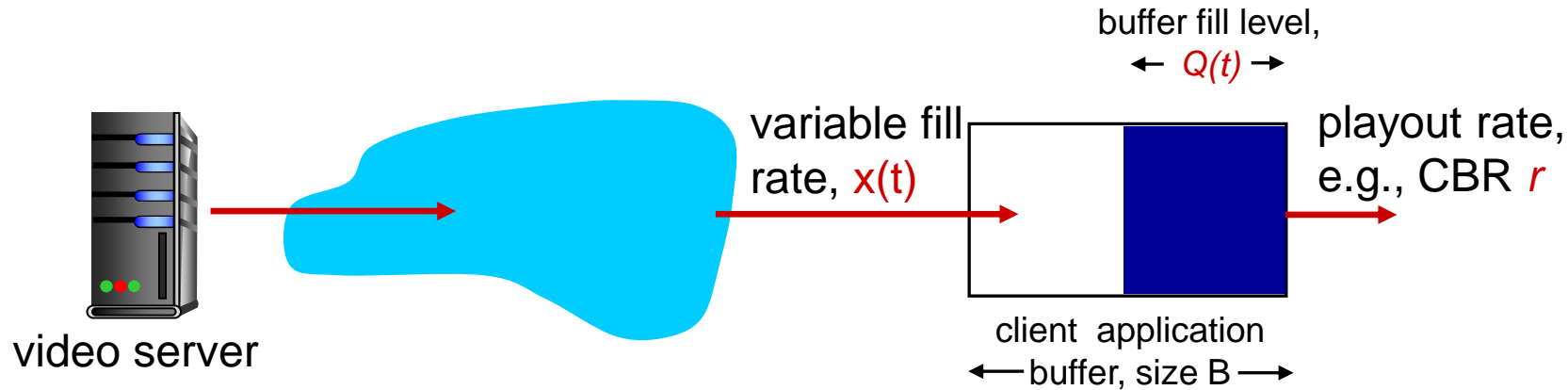


Bộ đệm tại Client, giải phóng bộ đệm



1. Khởi tạo dữ liệu buffer khi chương trình ứng dụng bắt đầu tại t_p
2. Chương trình ứng dụng bắt đầu tại thời điểm t_p ,
3. Tốc độ dữ liệu vào buffer thay đổi theo thời gian $x(t)$ và tốc độ giải phóng dữ liệu cho chương trình ứng dụng là hằng số r

Bộ đệm tại Client, giải phóng bộ đệm



Bộ đệm phát: tốc độ trung bình dữ liệu vào (\bar{x}), tốc độ dữ liệu ra (r):

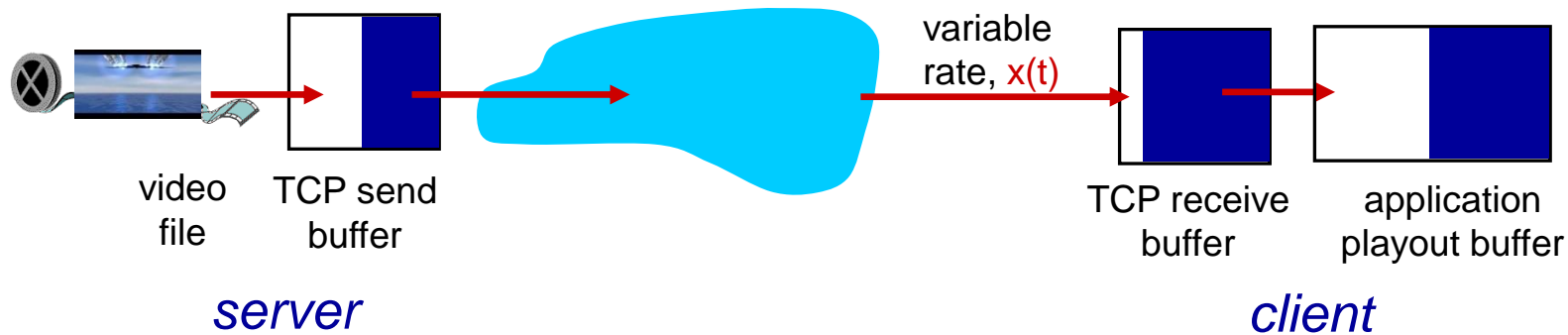
- $\bar{x} < r$: đến một lúc buffer sẽ trống rỗng (đây là nguyên nhân dừng hình của video)
- $\bar{x} > r$: buffer sẽ không trống rỗng với điều kiện độ trễ khởi tạo phát đủ lớn phù hợp với sự thay đổi của $x(t)$
 - *Cân bằng độ trễ khởi tạo phát*: buffer sẽ không cạn kiệt nếu độ trễ phát lớn, nhưng độ trễ lớn chỉ duy trì đến lúc người sử dụng bắt đầu xem

Phát đa phương tiện: UDP

- server gửi với tốc độ phù hợp cho client
 - Thông thường: tốc độ gửi = tốc độ mã hóa = tốc độ không đổi
 - Tốc độ truyền có thể bị quên đến mức tắc nghẽn
- Độ trễ phát ngắn (2-5 giây) loại bỏ trễ mạng
- Phục hồi lỗi: tầng ứng dụng, thời gian cho phép
- RTP [RFC 2326]: các dạng tải trọng đa phương tiện
- UDP có thể không đi qua firewalls

Phát đa phương tiện : HTTP

- Tập tin có thể được truy xuất qua HTTP GET
- Gửi với tốc độ tối đa có thể của TCP



- Tốc độ biến thiên vì điều khiển tắc nghẽn của TCP, truyền lại
- Độ trễ phát lớn: làm trơn tốc độ phát của TCP
- HTTP/TCP dễ dàng truyền qua firewalls

Multimedia networking: outline

9.1 multimedia networking applications

9.2 streaming *stored* video

9.3 voice-over-IP

9.4 protocols for *real-time* conversational applications

9.5 network support for multimedia

Voice-over-IP (VoIP)

- *VoIP yêu cầu độ trễ điểm-điểm*: cần thiết để duy trì hội thoại trực tuyến
 - Có thể nhận biết độ trễ cao, giảm khả năng tương tác
 - < 150 msec: tốt
 - > 400 msec: xấu
 - Bao gồm mức ứng dụng (đóng gói, phát dữ liệu), độ trễ mạng
- *Khởi tạo phiên*: người gọi quảng bá địa chỉ IP, số hiệu cổng và giải thuật mã hóa như thế nào?
- *Dịch vụ giá trị gia tăng*: chuyển tiếp cuộc gọi, lọc, ghi âm
- *Dịch vụ khẩn cấp*: 911

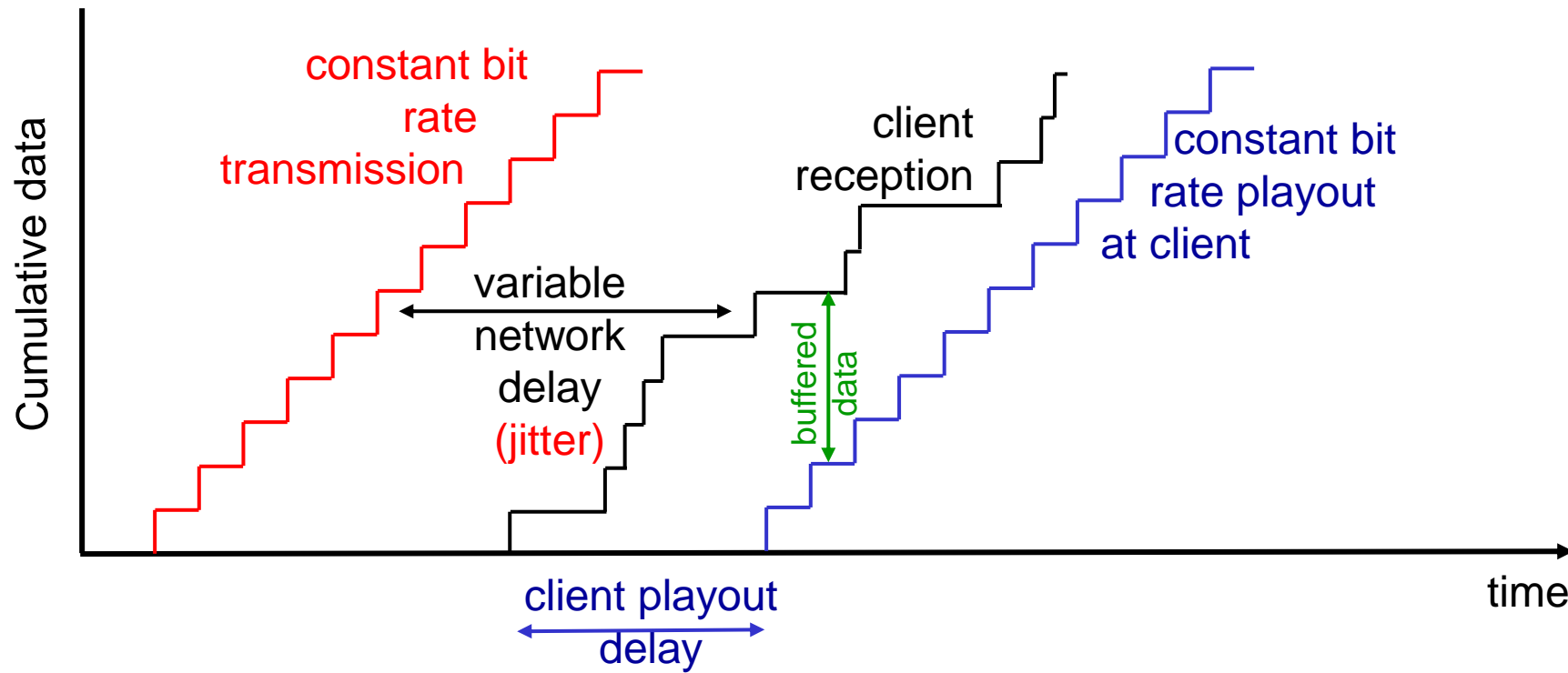
Các đặc trưng của VoIP

- Âm thanh người nói: đan xen giữa âm thanh và khoảng lặng.
 - 64 kbps trong khi âm thanh phát ra
 - Các gói chỉ được tạo ra khi âm thanh phát ra
 - Các khối 20 mili giây với tốc độ 8 Kbytes/giây: 160 bytes dữ liệu
- header tầng ứng dụng được thêm vào mỗi khối
- Khối+header được đóng gói vào phân đoạn UDP hoặc TCP
- Ứng dụng gửi phân đoạn vào socket sau 20 mili giây một trong quá trình âm thanh được nói ra

VoIP: mất gói, độ trễ

- **network loss:** IP datagram mất do nghẽn mạng (tràn bộ đệm router)
- **delay loss:** IP datagram đến quá muộn cho ứng dụng bên nhận
 - Độ trễ: xử lý, hàng đợi trong mạng; độ trễ trong hệ thống cuối (sender, receiver)
 - Độ trễ tối đa chấp nhận được: 400 mili giây
- **loss tolerance:** tùy thuộc vào mã hóa giọng nói mất mát có thể che dấu, tỷ lệ mất gói trong khoảng 1% đến 10% có thể chấp nhận được

Chênh lệch độ trễ



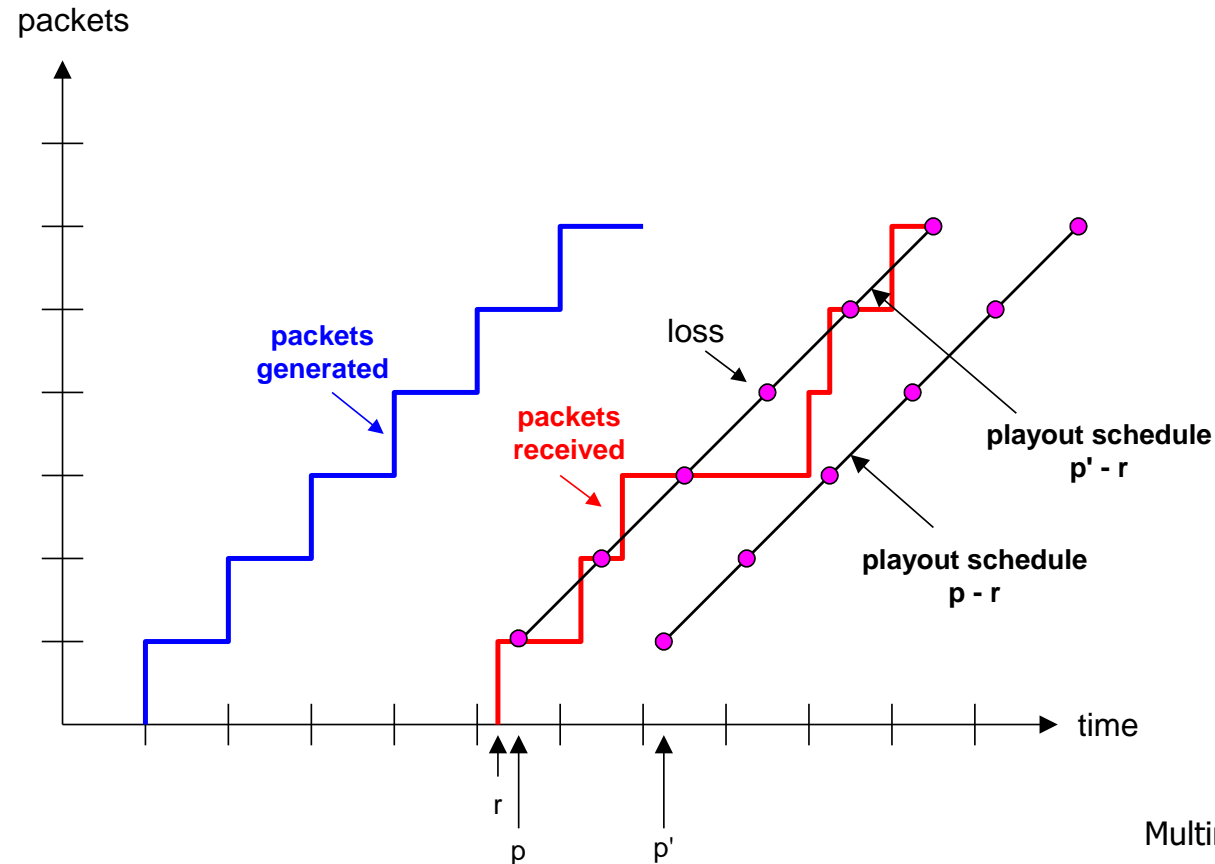
- Độ trễ điểm-điểm của hai gói tin liên tục: chênh lệch có thể trên/dưới 20 mili giây (sai lệch thời gian truyền)

VoIP: độ trễ phát cố định

- Bên nhận cố gắng phát mỗi khối chính xác q mili giây sau khi khối được tạo ra.
 - Khối được dán nhãn thời gian t : phát tại thời điểm $t+q$
 - Khối đến sau $t+q$: dữ liệu đến quá muộn cho ứng dụng: dữ liệu “mất”
- Cân đối khi chọn q :
 - q lớn: giảm số gói bị mất
 - q bé: trải nghiệm tương tác tốt hơn

VoIP: độ trễ phát cố định

- bên gửi tạo ra các gói sau mỗi 20 mili giây trong suốt thời gian nói.
- gói đầu tiên nhận được tại thời điểm r
- lịch phát đầu tiên: bắt đầu tại thời điểm p
- lịch phát thứ hai: bắt đầu tại thời điểm p'



Độ trễ phát thích nghi (I)

- **Mục đích:** độ trễ phát thấp, tỷ lệ mất gói tin đến muộn thấp
- **Phương pháp:** điều chỉnh thời gian phát thích nghi:
 - Ước lượng độ trễ mạng, điều chỉnh thời gian phát cho mỗi lần nói phát ra
 - Khoảng im lặng được nén và kéo dài
 - Các khối vẫn được phát sau mỗi 20 mili giây suốt thời gian nói
- Ước tính độ trễ gói tương thích: (EWMA – trung bình di chuyển theo hàm mũ, còn gọi là ước lượng TCP RTT):

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

Diagram illustrating the components of the EWMA formula:

- d_i : delay estimate after i th packet
- $(1-\alpha)$: small constant, e.g. 0.1
- $r_i - t_i$: time received - time sent (timestamp)
- $(r_i - t_i)$: measured delay of i th packet

Độ trễ phát thích nghi (2)

- Đồng thời thuận tiện cho tính độ lệch trung bình của độ trễ, v_i :

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

- Ước lượng d_i , v_i được tính cho mỗi gói nhận được, nhưng chỉ sử dụng tại thời điểm bắt đầu nói
- Đối với gói đầu tiên của âm thanh nói ra thời gian phát là:

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

- Các gói còn lại của âm thanh nói được phát theo chu kỳ

Độ trễ phát thích nghi (3)

Q: Làm thế nào để người nhận xác định được đâu là gói đầu tiên trong một lần nói?

- Nếu không xảy ra mất gói, bên nhận nhìn vào nhãn thời gian liên tiếp
 - Khác nhau của các nhãn liên tiếp > 20 msec --> bắt đầu nói chuyện.
- Trong trường hợp xảy ra mất gói, bên nhận phải xem cả nhãn thời gian và số hiệu chuỗi
 - Khác nhau của các nhãn liên tiếp > 20 msec **và** số hiệu chuỗi không có khoảng trống (gaps) --> bắt đầu nói chuyện.

VoiP: phục hồi từ mất gói (I)

Challenge: phục hồi từ mất gói do độ trễ nhỏ có thể chấp nhận được giữa truyền ban đầu và phát lại

- Mỗi ACK/NAK cần \sim một RTT
- Thay thế: *sửa lỗi chuyển tiếp (FEC)*
 - Gửi đủ số bits cho phép phục hồi mà không cần truyền lại (ví dụ mã chẵn lẻ hai chiều)

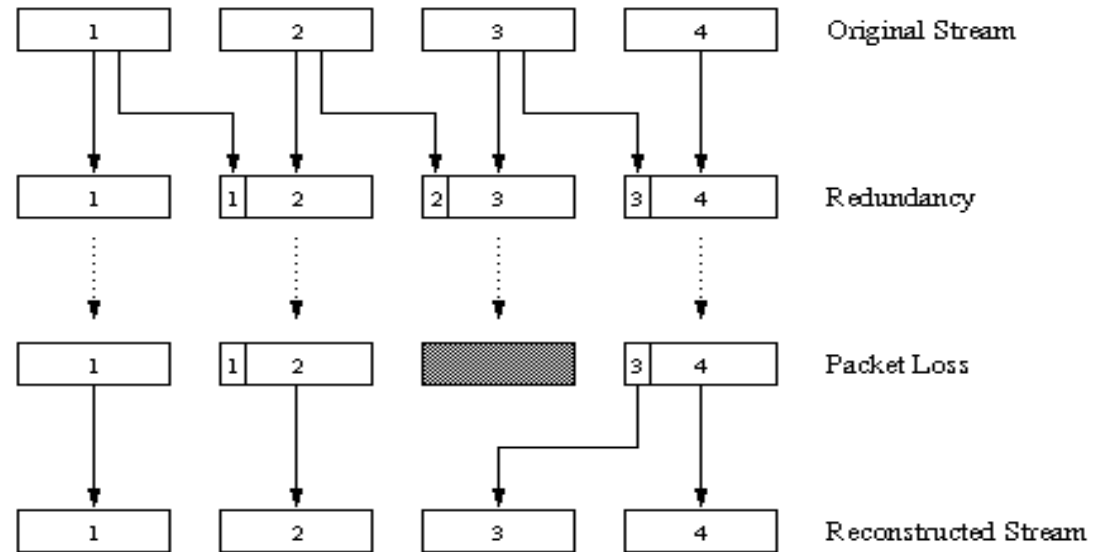
FEC Đơn giản

- Đối với mỗi nhóm gồm n khối, tạo khối thặng dư của n khối gốc
- Gửi $n+1$ khối, tăng bằng thông bởi hệ số $1/n$
- Có thể xây dựng lại n khối ban đầu nếu nhiều nhất một khối bị mất từ $n+1$ khối, với độ trễ phát.

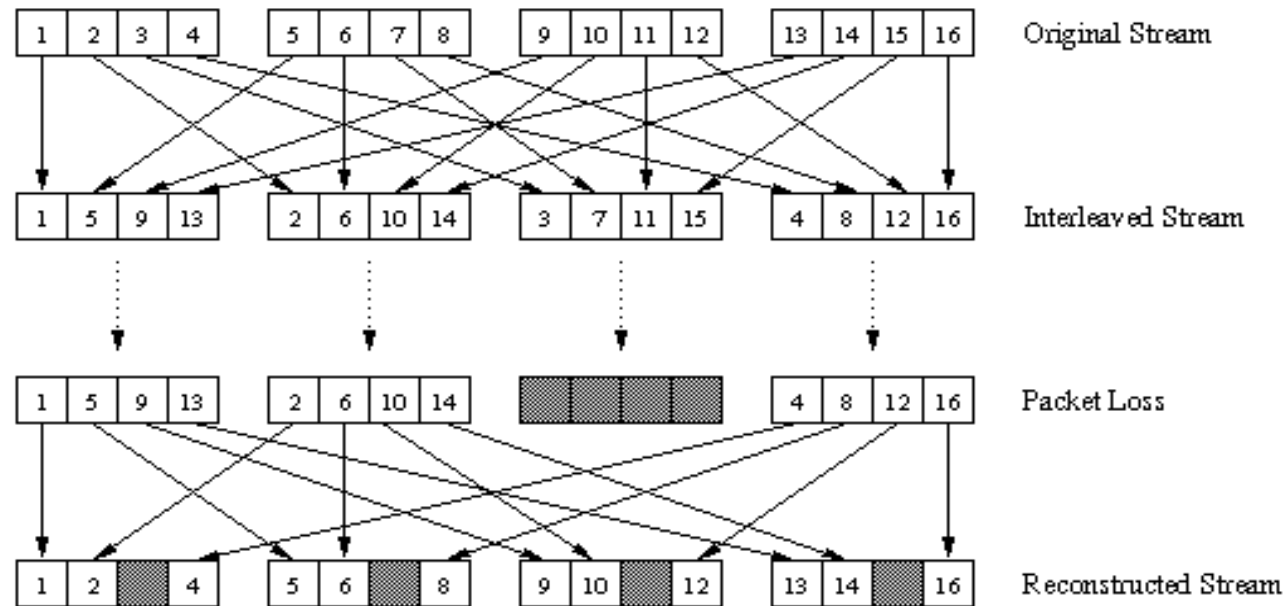
VoiP: phục hồi từ mất gói (2)

Lược đồ FEC khác:

- “cồng theo luồng chất lượng thấp hơn”
- Gửi luồng âm thanh độ phân giải thấp hơn dưới dạng thông tin dư thừa
- Ví dụ luồng đại diện PCM hoạt động ở 64 kbps và luồng thặng dư GSM hoạt động ở 13 kbps
- Mất mát không liên tiếp: bên nhận có thể che đậy được mất mát
- khái quát hóa: cũng có thể nối thêm khối thứ $(n-1)$ và $(n-2)$ tốc độ bit thấp



VoiP: phục hồi từ mất gói (3)

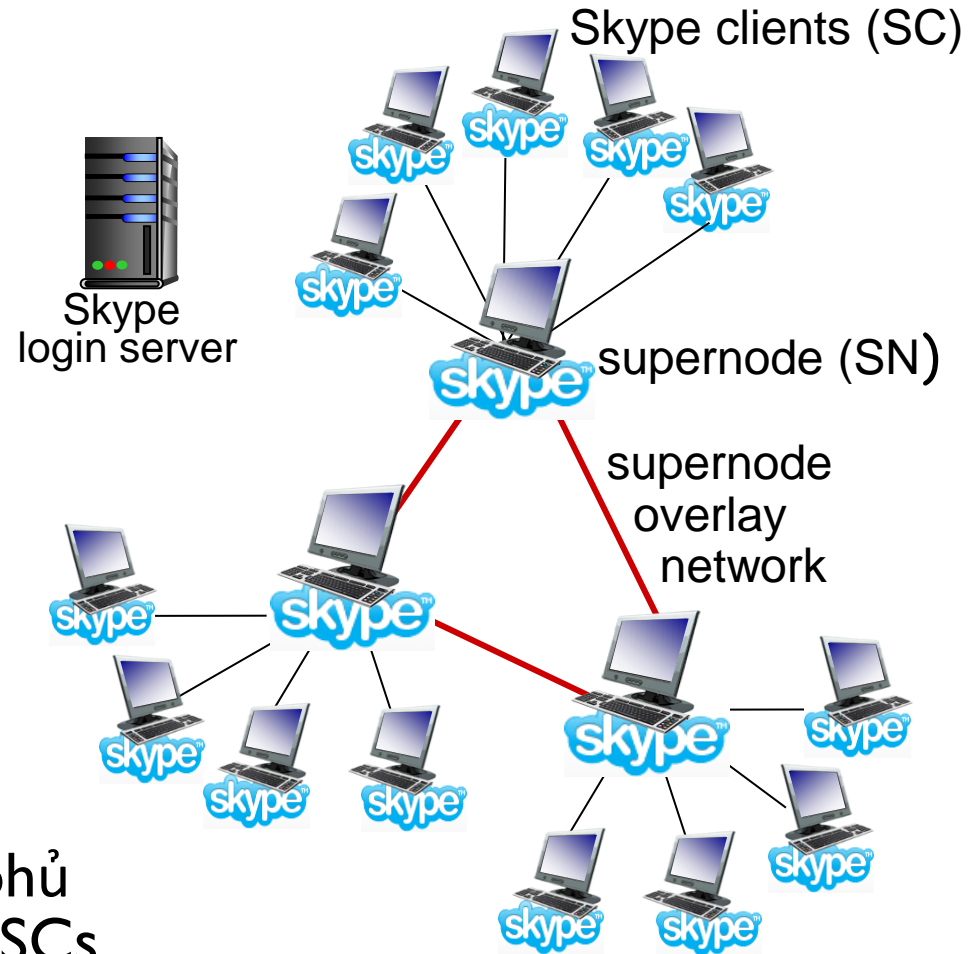


Xen kẽ để che dấu mất mát:

- Khối âm thanh được chia thành các đơn vị nhỏ hơn, ví dụ: bốn đơn vị 5 msec cho mỗi đoạn âm thanh 20 msec
- gói chứa các đơn vị nhỏ từ các khối khác nhau
- nếu gói bị mất, vẫn có hầu hết các đoạn gốc
- không có chi phí dự phòng, nhưng làm tăng độ trễ phát

Voice-over-IP: Skype

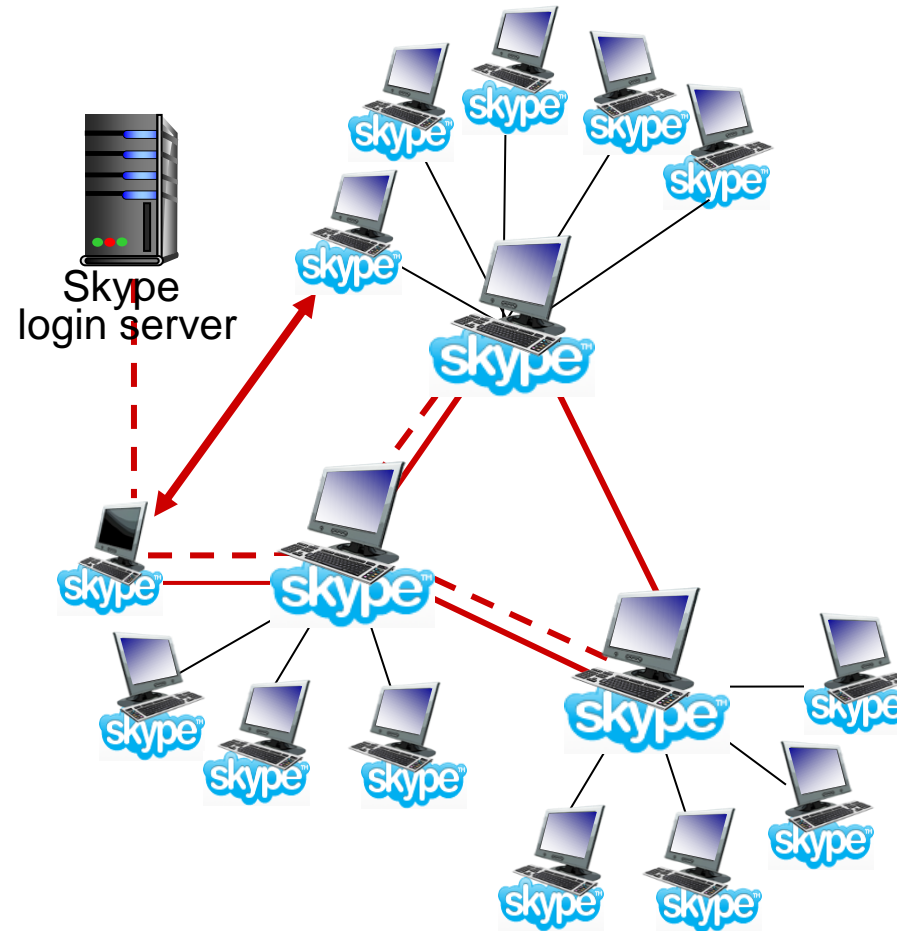
- Giao thức tầng ứng dụng độc quyền
 - Mã hóa các bản tin
- Các thành phần P2P:
 - **clients:** Skype ngang hàng kết nối trực tiếp với nhau cho VoIP call
 - **super nodes (SN):** Skype ngang hàng với chức năng đặc biệt
 - **overlay network:** mạng phủ thông qua SNs xác định SCs
 - **login server**



P2P voice-over-IP: Skype

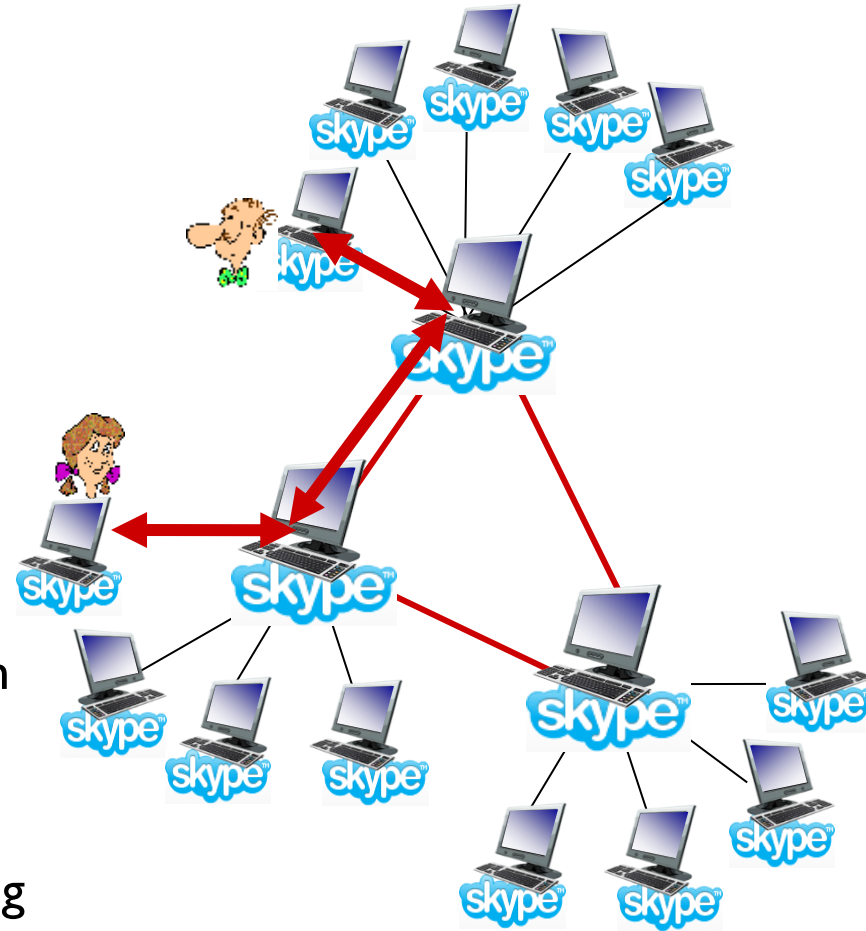
Hoạt động của Skype client :

1. Gia nhập mạng Skype bằng kết nối đến SN (IP trong cache) sử dụng TCP
2. Đăng nhập (username, password) tới Skype login server trung tâm
3. Nhận địa chỉ IP để gọi từ SN, SN overlay
 - hoặc từ danh sách client bạn bè
4. khởi tạo cuộc gọi trực tiếp đến người được gọi



Skype: peers as relays

- **problem:** cả Alice, Bob đều ở sau “NATs”
 - NAT ngăn chặn người ở ngoài khởi tạo kết nối vào trong
 - Người ở trong có thể khởi tạo kết nối ra ngoài
- **relay solution:** Alice, Bob duy trì kết nối mở tới SNs của mình
 - Alice báo hiệu với SN của mình kết nối tới Bob
 - Alice’s SN kết nối tới Bob’s SN
 - Bob’s SN kết nối tới Bob thông qua kết nối mở Bob đã khởi tạo tới SN của mình



Multimedia networking: outline

9.1 multimedia networking applications

9.2 streaming *stored* video

9.3 voice-over-IP

9.4 protocols for *real-time* conversational applications: RTP, SIP

9.5 network support for multimedia

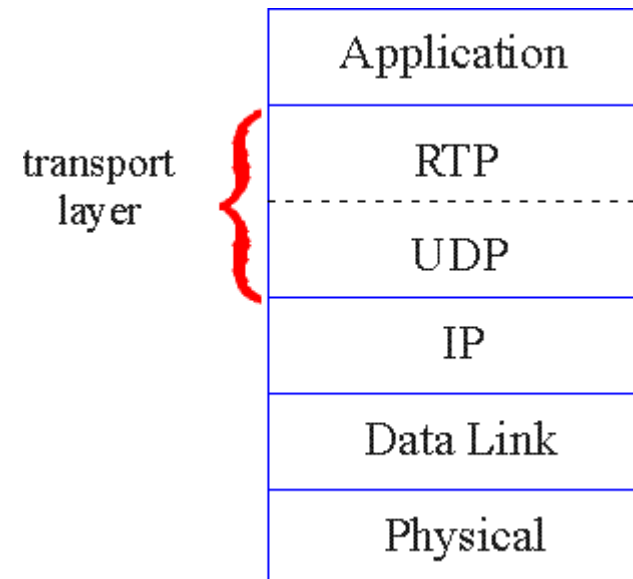
Giao thức thời gian thực (RTP)

- RTP chỉ định cấu trúc gói cho các gói mang dữ liệu âm thanh, video
- RFC 3550
- Gói RTP cung cấp
 - Nhận diện loại tải trọng
 - Số thứ tự gói
 - Nhãn thời gian
- RTP chạy trong hệ thống cuối
- Các gói RTP đóng trong các phân đoạn UDP
- Khả năng tương tác: nếu hai ứng dụng VoIP chạy RTP, chúng có thể hoạt động cùng nhau

RTP chạy trên UDP

Các thư viện RTP cung cấp giao diện lớp vận chuyển mở rộng UDP:

- số hiệu cổng, địa chỉ IP
- nhận diện kiểu tải trọng
- đánh số thứ tự gói
- dán nhãn thời gian



Ví dụ RTP

example: gửi giọng nói được mã hóa PCM 64 kbps qua RTP

- ứng dụng thu thập dữ liệu được mã hóa theo từng khối, ví dụ: cứ sau 20 msec = 160 byte trong một khối
- khối âm thanh + RTP header định dạng gói RTP đều được đóng trong phân đoạn UDP
- RTP header chỉ ra kiểu mã hóa âm thanh trong mỗi gói
 - Bên gửi có thể thay đổi mã hóa trong thời gian hội thoại
- RTP header cũng chứa số thứ tự khối, nhãn thời gian

RTP và QoS

- RTP *không* cung cấp bất cứ kỹ thuật nào đảm bảo dữ liệu được chuyển đúng thời gian hoặc QoS nào khác
- Đóng gói RTP chỉ thấy ở các hệ thống đầu cuối (không phải bởi các bộ định tuyến trung gian)
 - bộ định tuyến cung cấp dịch vụ nỗ lực tốt nhất, không có nỗ lực đặc biệt để đảm bảo các gói RTP đến đích kịp thời

RTP header

<i>payload type</i>	<i>sequence number</i>	<i>time stamp</i>	<i>Synchronization Source ID</i>	<i>Miscellaneous fields</i>
-------------------------	----------------------------	-------------------	--------------------------------------	---------------------------------

Kiểu tải trọng (7 bits): cho biết kiểu mã hóa dữ liệu được sử dụng hiện thời. Nếu bên gửi thay đổi mã hóa trong cuộc gọi, bên gửi thông báo cho bên nhận thông qua kiểu tải trọng

Payload type 0: PCM mu-law, 64 kbps

Payload type 3: GSM, 13 kbps

Payload type 7: LPC, 2.4 kbps

Payload type 26: Motion JPEG

Payload type 31: H.261

Payload type 33: MPEG2 video

Số hiệu chuỗi (16 bits): tăng lên một cho mỗi gói RTP gửi đi

❖ Phát hiện mất gói, phục hồi chuỗi các gói tin

RTP header

<i>payload type</i>	<i>sequence number</i>	<i>time stamp</i>	<i>Synchronization Source ID</i>	<i>Miscellaneous fields</i>
-------------------------	----------------------------	-------------------	--------------------------------------	---------------------------------

- **Nhãn thời gian (32 bits long):** lấy mẫu hiện thời của byte thứ nhất trong gói dữ liệu RTP
 - Đối với âm thanh, đồng hồ nhãn thời gian sẽ tăng lên 1 cho mỗi chu kỳ lấy mẫu (ví dụ: mỗi 125 μ sec cho đồng hồ lấy mẫu 8 KHz)
 - Khi nguồn được kích hoạt, nếu ứng dụng tạo ra các khối của 160 mẫu đã mã hóa, nhãn thời gian tăng lên 160 đối với mỗi gói RTP. Đồng hồ nhãn thời gian tiếp tục tăng với tốc độ không đổi khi nguồn không được kích hoạt nữa.
- **SSRC field (32 bits long):** định danh nguồn của luồng RTP. Mỗi luồng trong phiên RTP có SSRC riêng biệt

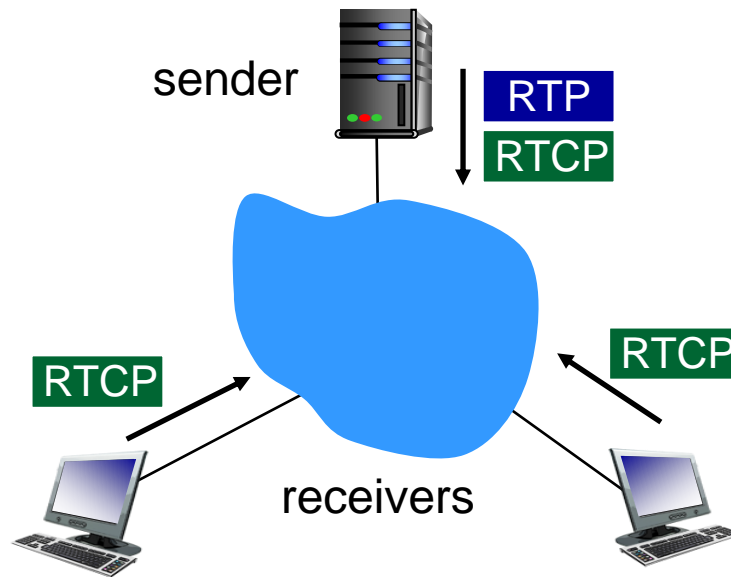
Lập trình RTSP/RTP

- Xây dựng một server đóng gói các khung video đã lưu trữ vào các gói RTP
 - Lấy khung video, thêm RTP headers, tạo UDP segments, gửi segments tới UDP socket
 - Bao gồm các số hiệu chuỗi và nhãn thời gian
 - client RTP cung cấp cho người sử dụng
- Đồng thời viết chương trình phía client của RTSP
 - Các câu lệnh thực hiện play/pause
 - server RTSP cung cấp cho người sử dụng

Giao thức điều khiển thời gian thực (RTCP)

- Hoạt động cùng với RTP
- Mỗi ứng dụng tham gia trong phiên RTP theo định kỳ gửi các gói điều khiển RTCP tới tất cả các ứng dụng còn lại
- Mỗi gói RTCP chứa báo cáo của sender và/hoặc receiver
 - Thống kê báo cáo hữu ích cho ứng dụng: số gói được gửi, số gói mất, chênh lệch thời gian đến
- Dùng thông tin phản hồi điều khiển hiệu năng
 - sender thay đổi số lượng truyền dựa vào thông tin phản hồi

RTCP: đa người gửi multicast



- Mỗi phiên RTP: thường là một địa chỉ multicast đơn; tất cả các gói RTP /RTCP thuộc về phiên sử dụng địa chỉ multicast
- Các gói RTP, RTCP được phân biệt với nhau thông qua các số cổng riêng biệt
- Để hạn chế lưu lượng truy cập, mỗi ứng dụng tham gia giảm lưu lượng RTCP khi số lượng ứng dụng tham gia tăng lên

RTCP: các kiểu gói

Gói báo cáo bên nhận:

- phần mất của các gói, số hiệu chuỗi cuối cùng, độ lệch trung bình của các gói đến

Gói báo cáo bên gửi:

- SSRC của RTP luồng, thời gian hiện tại, số lượng gói đã gửi, số lượng bytes đã gửi

Gói mô tả nguồn:

- Địa chỉ e-mail của người gửi, tên người gửi, SSRC của luồng RTP liên kết
- Cung cấp ánh xạ giữa SSRC và tên của user/host

RTCP: đồng bộ hóa luồng

- RTCP có thể đồng bộ hóa các luồng đa phương tiện khác nhau trong một phiên RTP
- Ví dụ: ứng dụng hội nghị truyền hình: mỗi người gửi tạo một luồng RTP cho video, một cho âm thanh.
- Nhãn thời gian trong các gói RTP gắn với đồng hồ lấy mẫu video, audio
 - *không* liên quan đến đồng hồ bên ngoài
- Mỗi gói báo cáo người gửi RTCP chứa (cho hầu hết các gói mới được tạo ra liên kết với luồng RTP):
 - Nhãn thời gian của gói RTP
 - Thời gian hệ thống xác định thời điểm gói tin được tạo
- Bên nhận sử dụng liên kết để đồng bộ hóa phát lại âm thanh, video

RTCP: mở rộng băng thông

RTCP cố gắng giới hạn lưu lượng của mình ở mức 5% băng thông của phiên

Ví dụ : một người gửi, gửi video với tốc độ 2 Mbps

- RTCP cố gắng giới hạn lưu lượng RTCP ở mức 100 Kbps
- RTCP cung cấp 75% tốc độ tới người nhận; còn lại 25% tới người gửi
- 75 kbps được chia sẻ đồng đều giữa các máy thu:
 - Với R máy thu, mỗi máy nhận được lưu lượng truyền RTCP là $75/R$ kbps.
- Người gửi nhận lưu lượng truyền RTCP là 25 kbps.
- Người tham gia xác định thời gian truyền gói RTCP bằng cách tính kích thước gói RTCP trung bình (trên toàn bộ phiên) và chia cho tốc độ được phân bổ

SLP: giao thức khởi tạo phiên [RFC 3261]

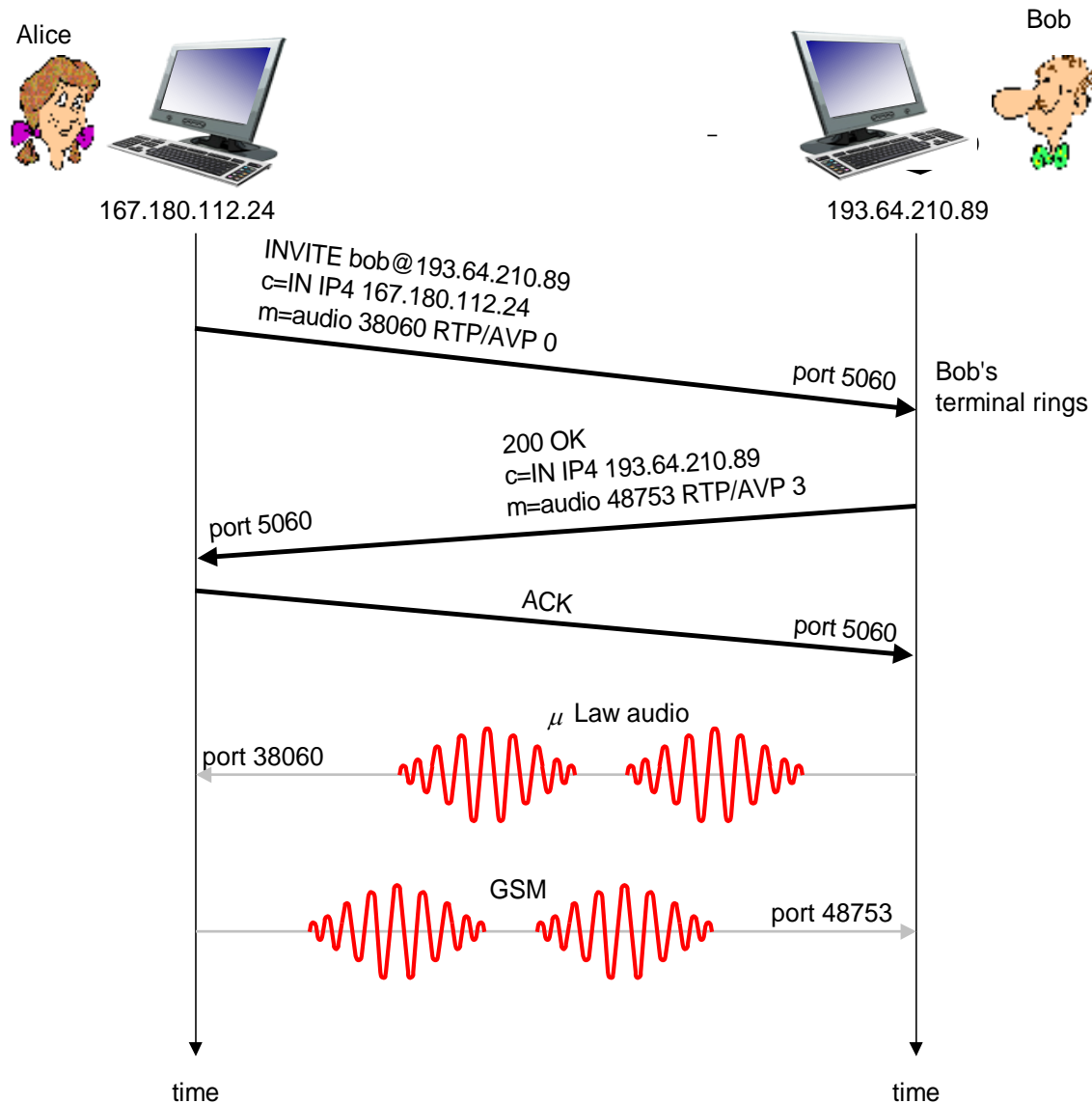
Tầm nhìn dài hạn:

- Tất cả các cuộc gọi điện thoại, cuộc gọi hội nghị video diễn ra qua Internet
- Mỗi người được xác định bằng tên hoặc địa chỉ email, thay vì theo số điện thoại
- Có thể đến được người được gọi, cho dù người được gọi đang ở đâu, hay dùng thiết bị IP gì

Dịch vụ SIP

- SIP cung cấp các kỹ thuật để thiết lập cuộc gọi:
 - Đối với người gọi cho người được gọi biết mình muốn thiết lập cuộc gọi tới
 - Vì vậy người gọi, người nhận thống nhất về loại phương tiện, mã hóa
 - Kết thúc cuộc gọi
- Xác định địa chỉ IP hiện thời của người nhận:
 - Ánh xạ định danh ghi nhớ đến địa chỉ IP hiện thời
- Quản trị cuộc gọi:
 - thêm luồng phương tiện mới trong khi gọi
 - thay đổi mã hóa trong khi gọi
 - mời người khác
 - chuyển, giữ cuộc gọi

Ví dụ: thiết lập cuộc gọi đến địa chỉ IP đã biết



- Bản tin SIP mời của Alice chỉ ra số hiệu cổng, địa chỉ IP, mã hóa mà Alice muốn nhận (PCM μ law)
- Bản tin 200 OK của Bob's chỉ ra số hiệu cổng, địa chỉ IP, mã hóa mà Bob muốn nhận (GSM)
- Bản tin SIP có thể được gửi qua TCP hoặc UDP; ở đây gửi qua RTP/UDP
- Cổng SIP ngầm định là 5060

Thiết lập cuộc gọi (more)

- Đàm phán mã:
 - Giả sử Bob không có bộ mã hóa PCM μ law
 - Bob sẽ trả lời bằng 606 Not Acceptable Reply, liệt kê các bộ mã hóa của mình. Sau đó Alice có thể gửi bản tin INVITE mới, với bộ mã hóa mới
- Từ chối cuộc gọi
 - Bob có thể từ chối cuộc gọi bằng trả lời “busy,” “gone,” “payment required,” “forbidden”
- Phương tiện có thể được truyền qua RTP hoặc một số giao thức khác

Ví dụ bản tin của SIP

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Lưu ý:

- Cú pháp bản tin HTTP
- sdp = session description protocol
- Call-ID là duy nhất cho tất cả các cuộc gọi

- Ở đây chúng ta không biết địa chỉ IP của Bob
 - Cần có server SIP trung gian
- Alice gửi, nhận các bản tin SIP sử dụng cổng SIP ngầm định 506
- Alice chỉ rõ trong header rằng SIP client gửi, nhận bản tin SIP qua UDP

Chuyển đổi tên, định vị người dùng

- Người gọi muốn gọi đến người được gọi, nhưng chỉ có tên hoặc địa chỉ email của người đó.
- Cần nhận được địa chỉ IP hiện thời của người được gọi:
 - Người dùng di chuyển
 - Giao thức DHCP
 - Người dùng có thiết bị địa chỉ IP khác nhau (PC, smartphone, car device)
- Kết quả có thể dựa trên:
 - thời gian của ngày (làm việc, ở nhà)
 - Người gọi (không muốn ông chủ gọi cho bạn khi đang ở nhà)
 - Trạng thái của người được gọi (cuộc gọi gửi tới hộp thư thoại khi người được gọi đang nói chuyện với ai đó)

Quản lý đăng ký SIP

- Một chức năng của SIP server: **quản lý đăng ký**
- khi Bob bắt đầu SIP client, client gửi bản tin SIP REGISTER tới server đăng ký của Bob

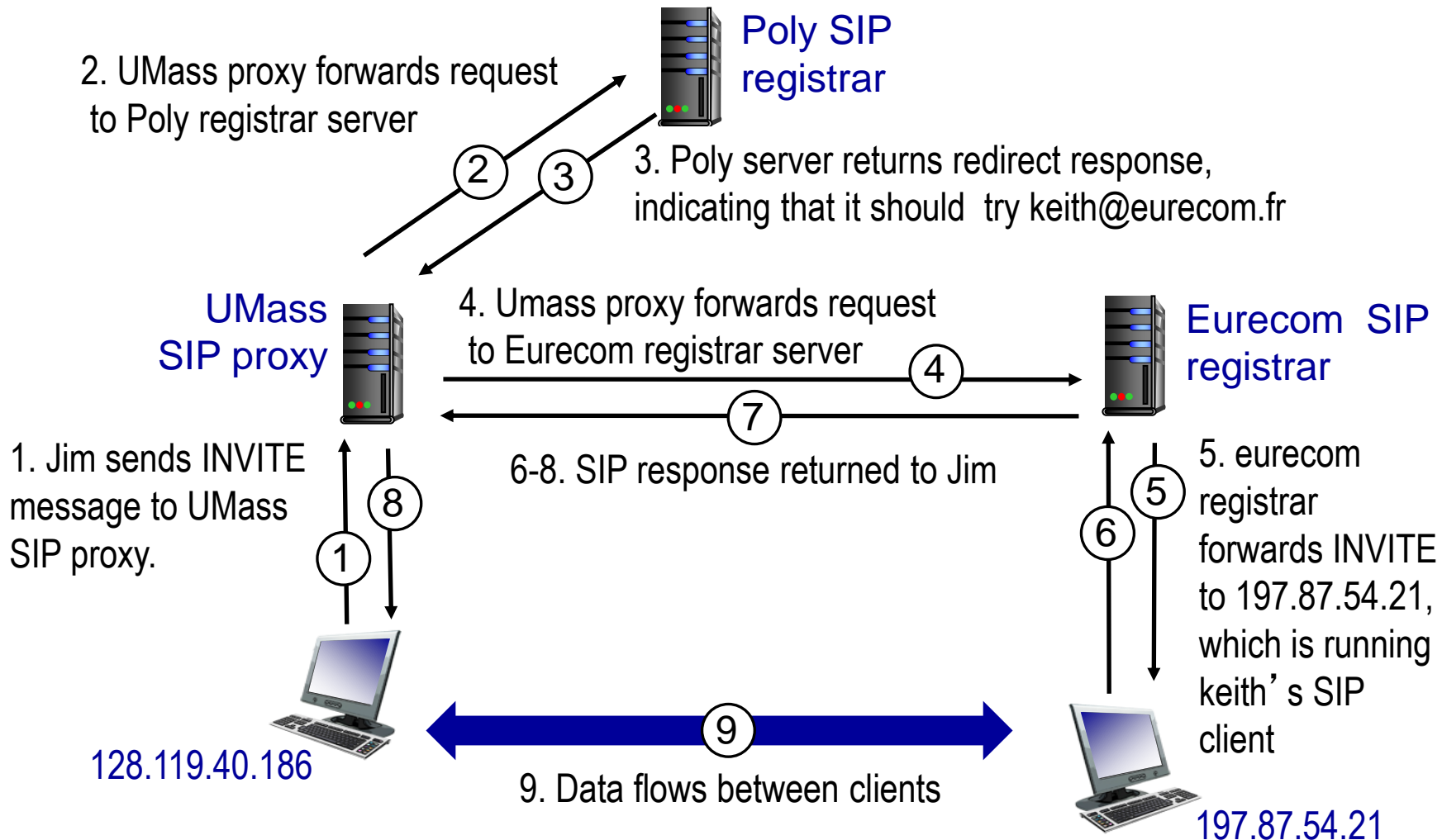
register message:

```
REGISTER sip:domain.com SIP/2.0  
Via: SIP/2.0/UDP 193.64.210.89  
From: sip:bob@domain.com  
To: sip:bob@domain.com  
Expires: 3600
```

SIP ủy quyền

- Một chức năng khác của SIP server: *ủy quyền*
- Alice bản tin mời đến proxy server của mình
 - Chứa địa chỉ sip:bob@domain.com
 - proxy chịu trách nhiệm định tuyến bản tin SIP tới người được gọi, có thể thông qua nhiều proxies
- Bob gửi trả lời thông qua cùng tập hợp SIP proxies
- proxy trả về bản tin trả lời của Bob cho Alice
 - Có chứa địa chỉ IP của Bob
- SIP proxy tương tự DNS server cục bộ thêm vào thiết lập TCP

Ví dụ SIP: jim@umass.edu calls keith@poly.edu



Multimedia networking: outline

9.1 multimedia networking applications

9.2 streaming *stored* video

9.3 voice-over-IP

9.4 protocols for *real-time* conversational applications

9.5 network support for multimedia

Mạng hỗ trợ cho đa phương tiện

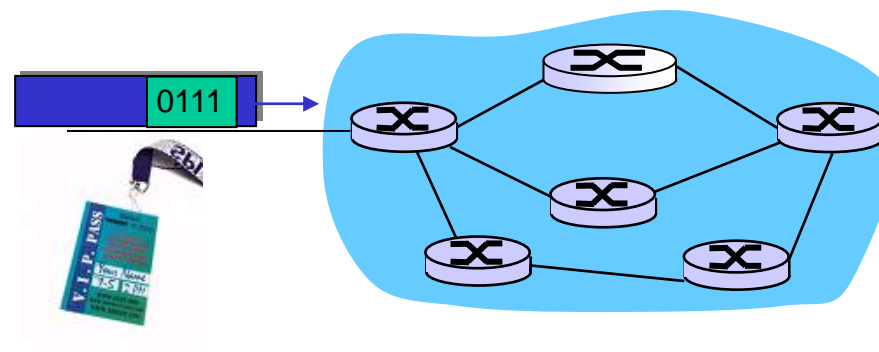
Approach	Granularity	Guarantee	Mechanisms	Complex	Deployed?
Making best of best effort service	All traffic treated equally	None or soft	No network support (all at application)	low	everywhere
Differentiated service	Traffic “class”	None of soft	Packet market, scheduling, policing.	med	some
Per-connection QoS	Per-connection flow	Soft or hard after flow admitted	Packet market, scheduling, policing, call admission	high	little to none

Mạng với khả năng tốt nhất có thể

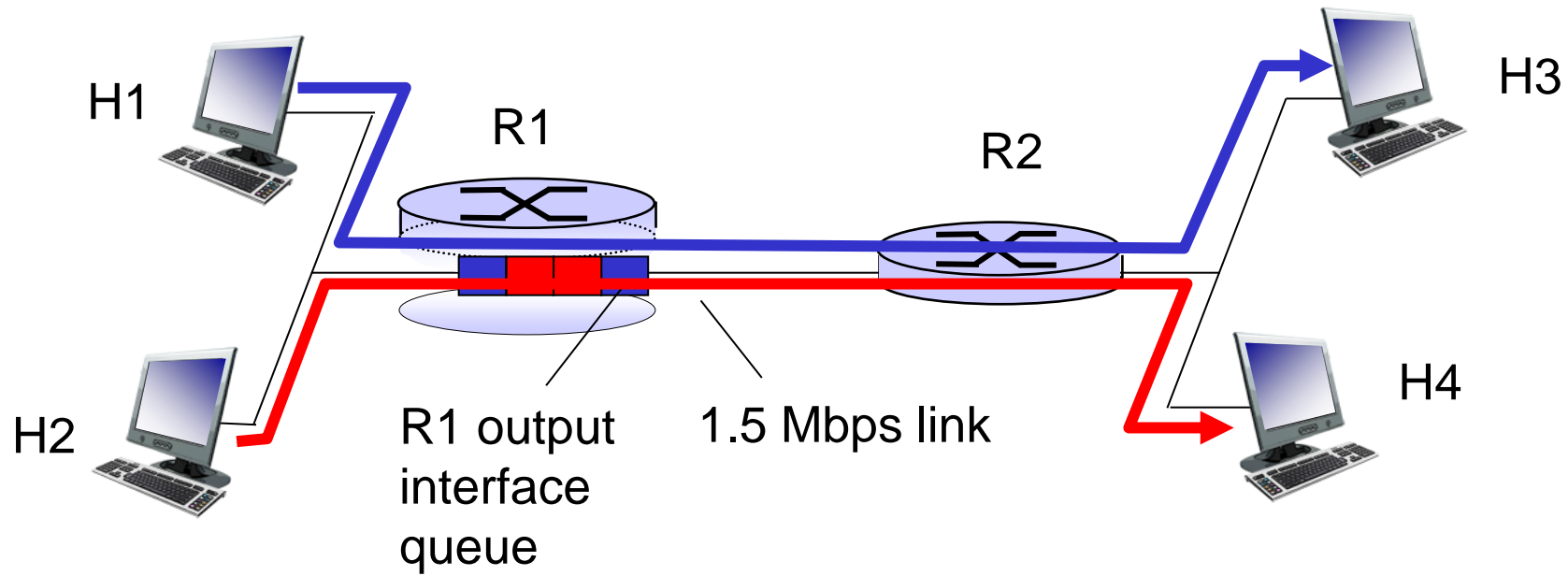
- **Phương pháp:** triển khai đủ thông lượng liên kết do vậy tắc nghẽn không xảy ra, luồng đa phương tiện không bị trễ hay mất
 - Cách thức tổ chức mạng đơn giản (sử dụng năng lực tốt nhất của mạng hiện thời)
 - Chi phí bang thông cao
- Những thách thức:
 - **Kích thước mạng:** bằng thông bao nhiêu là “đủ?”
 - **Xác định nhu cầu lưu lượng mạng:** cần xác định được lưu lượng mạng cho dành cho các luồng dữ liệu.

Cung cấp nhiều lớp dịch vụ

- Cho đến nay: làm với năng lực tốt nhất của dịch vụ
 - Một kích thước phù hợp tất cả mô hình dịch vụ
- Thay thế: nhiều lớp dịch vụ
 - Phân chia luồng vào các lớp
 - Mạng xử lý các loại lưu lượng khác nhau thông qua các lớp khác nhau (tương tự: dịch vụ VIP so với dịch vụ thông thường)
- Chi tiết: sự khác biệt dịch vụ giữa các lớp, không phải giữa các kết nối riêng lẻ

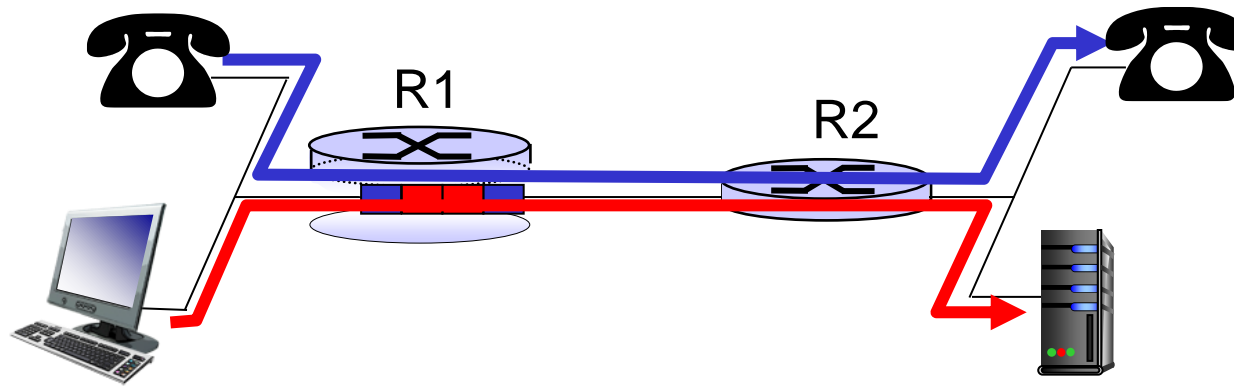


Nhiều lớp dịch vụ: kịch bản



Kịch bản I: trộn HTTP và VoIP

- Ví dụ: 1 Mbps VoIP, HTTP chia sẻ đường 1.5 Mbps.
 - HTTP có thể làm tắc nghẽn router, gây nên mất âm thanh
 - Cần ưu tiên cho âm thanh hơn HTTP

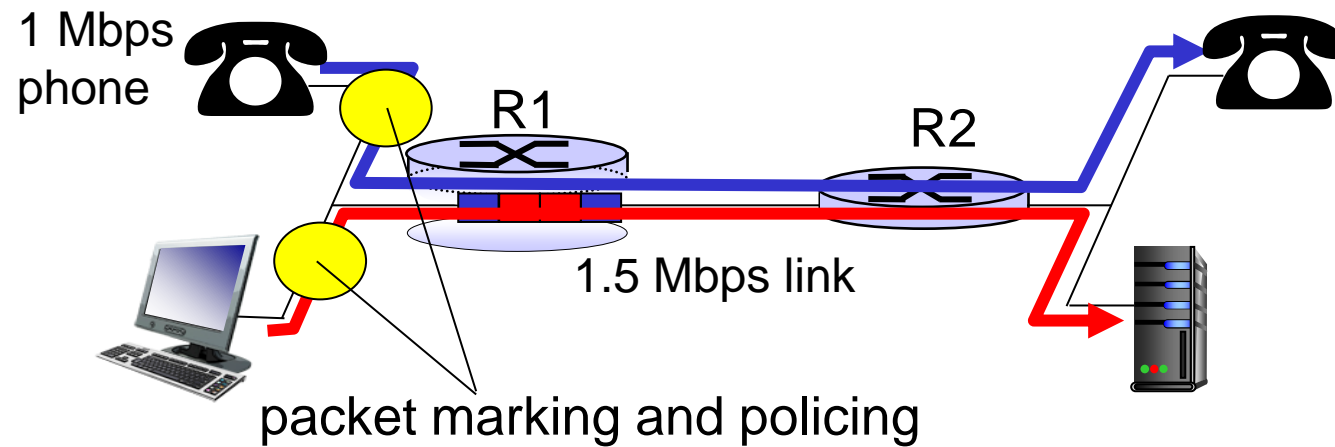


Nguyên tắc I

đánh dấu gói cần thiết cho bộ định tuyến để phân biệt giữa các lớp khác nhau; và chính sách bộ định tuyến mới để xử lý các gói phù hợp

Nguyên tắc đảm bảo QoS

- Điều gì xảy ra nếu ứng dụng hoạt động sai (VoIP gửi cao hơn tốc độ khai báo)
 - Kiểm soát: buộc tuân thủ nguồn phân bổ băng thông
- *Đánh dấu, kiểm soát* tại cạnh mạng

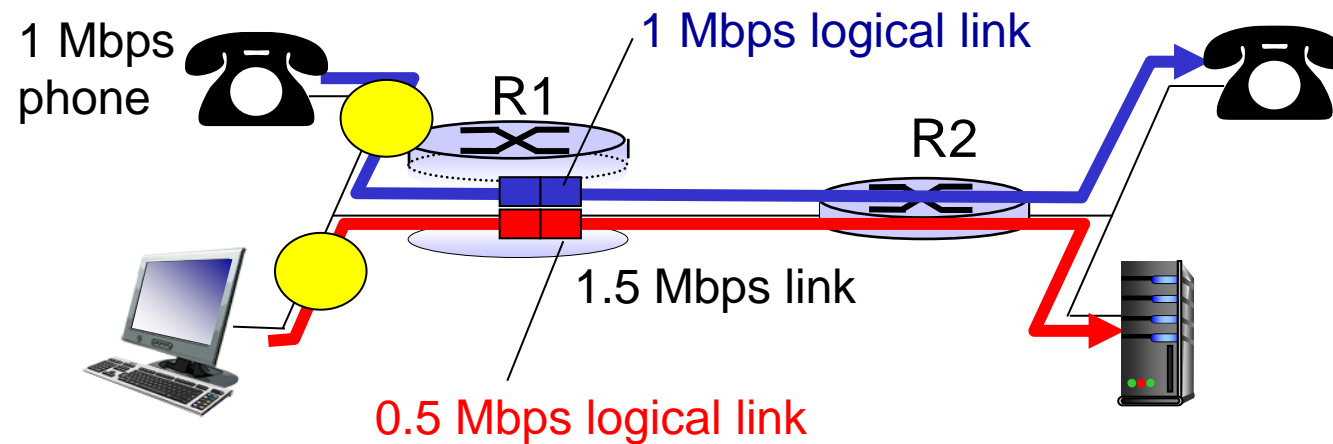


Nguyên tắc 2

Cung cấp sự bảo vệ (cách ly) cho một lớp đối với các lớp khác

Nguyên tắc đảm bảo QOS (more)

- Phân bổ băng thông cố định (không thể chia sẻ) cho luồng: sử dụng băng thông không hiệu quả nếu các luồng không sử dụng phân bổ của nó

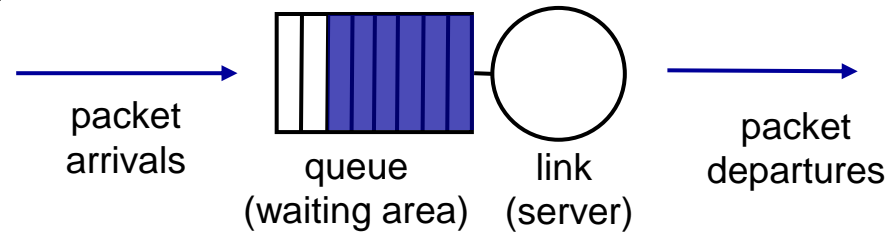


Nguyên tắc 3

Khi phân chia tài nguyên độc lập, cố gắng càng hiệu quả càng tốt

Cơ chế lập kế hoạch và kiểm soát

- **Lập lịch gói:** chọn gói xếp hàng tiếp theo để gửi tới liên kết đầu ra



- Một số phương pháp:
 - FCFS: đến trước được phục vụ trước
 - Ưu tiên nhiều lớp
 - Xoay vòng
 - Hàng đợi công bằng có trọng số (WFQ)

Cơ chế kiểm soát

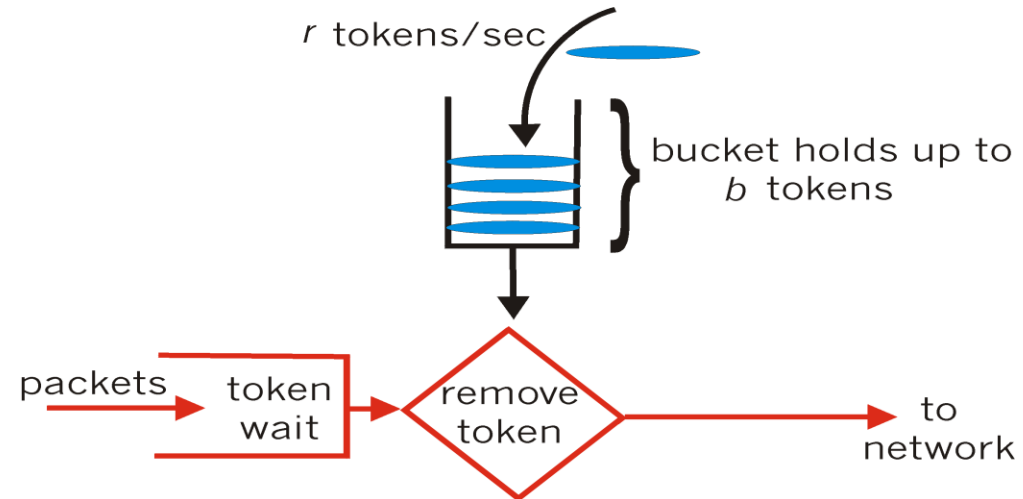
Mục đích: giới hạn luồng không vượt quá các tham số đã khai báo

Ba điều kiện được sử dụng phổ biến:

- **Tốc độ trung bình:** bao nhiêu gói tin được gửi trong một đơn vị thời gian
 - Câu hỏi quyết định: độ dài chu kỳ là bao nhiêu: 100 gói/giây hoặc 6000 gói/phút giống nhau về tốc độ trung bình!
- **Tốc độ tối đa:** ví dụ 6000 gói/phút (ppm) trung bình; 1500 ppm tốc độ tối đa
- **Kích thước cụm:** số lượng tối đa các gói được gửi liên tục nhau (không có sự can thiệp nào)

Cơ chế kiểm soát: thực hiện

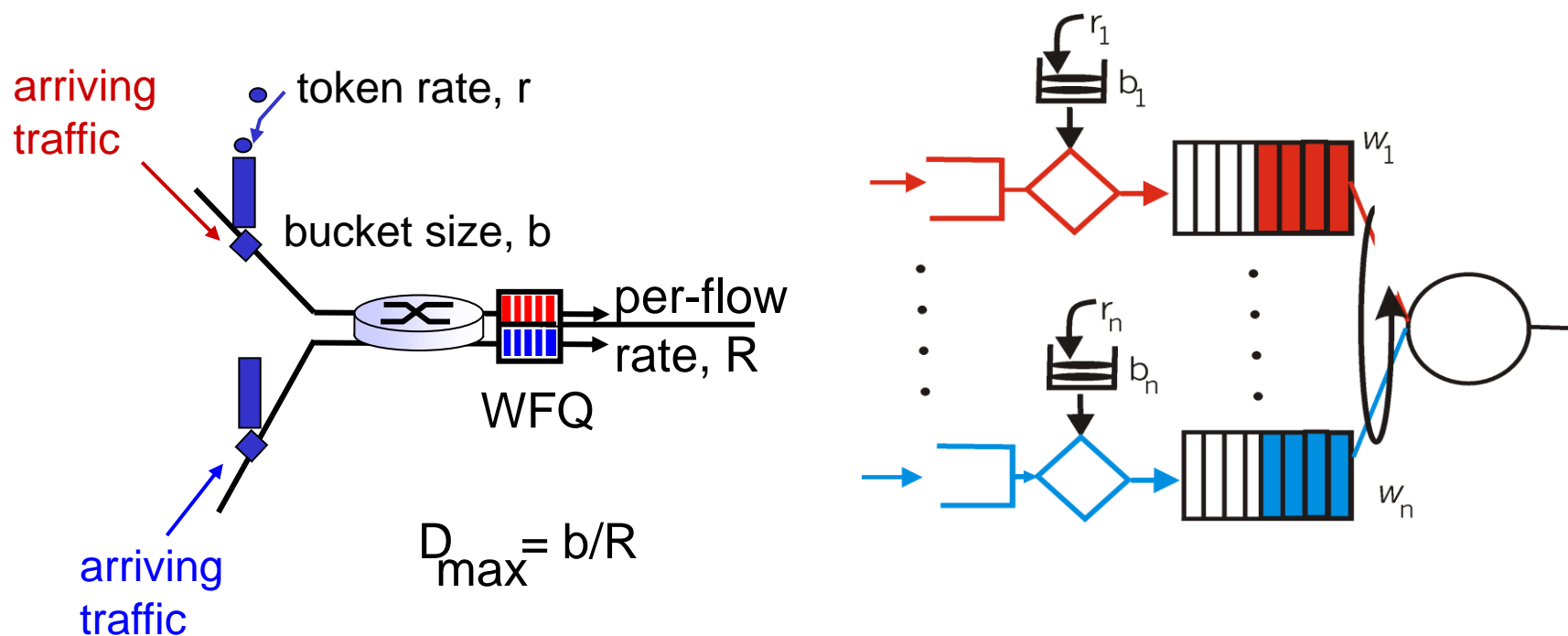
Thùng thẻ bài: giới hạn đầu vào đối với *kích thước cụm* và *tốc độ trung bình*



- Thùng có thể chứa b thẻ bài
- Tốc độ tạo thẻ bài r thẻ bài/giây trừ trường hợp thùng đầy
- Qua khoảng thời gian t : số gói tin được xác nhận nhỏ hơn hoặc bằng $(r t + b)$

Chính sách và đảm bảo chất lượng dịch vụ

- Kết hợp giữa thùng thẻ bài, WFQ để đảm bảo giới hạn trên của độ trễ, nghĩa là **đảm bảo QoS!**



Các dịch vụ khác nhau

- Muốn “định tính” các lớp dịch vụ
 - “tương tự như một giây dẫn”
 - Phân biệt dịch vụ: bạch kim, vàng, bạc
- **Khả năng mở rộng:** các chức năng đơn giản trong lõi mạng, tương đối phức tạp tại các routers biên (hoặc các trạm)
 - Báo hiệu, duy trì trạng thái định tuyến mỗi luồng khó khăn khi số lượng luồng lớn
- Không định nghĩa các lớp dịch vụ, cung cấp các thành phần chức năng để xây dựng các lớp dịch vụ

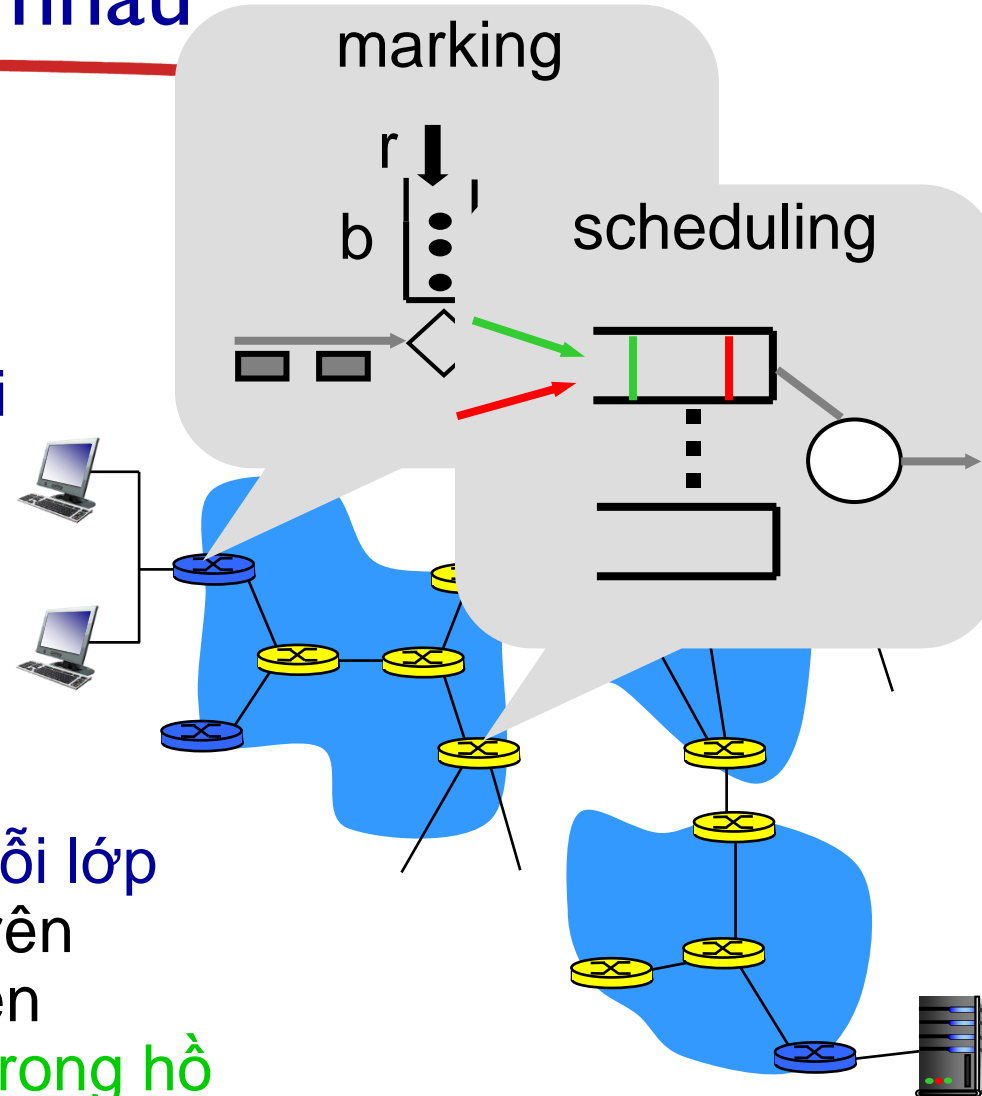
Kiến trúc dịch vụ khác nhau

Router biên: 

- quản lý lưu lượng trên mỗi luồng
- Đánh dấu các gói là trong hồ sơ và ngoài hồ sơ

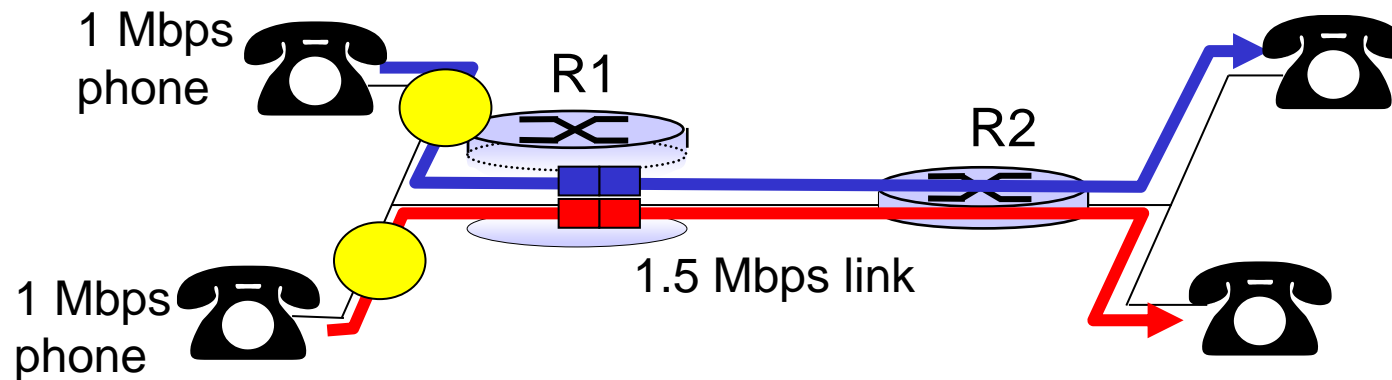
Router lõi :

- quản lý lưu lượng trên mỗi lớp
- Lập lịch và bộ đệm dựa trên đánh dấu tại các vùng biên
- Ưu tiên hơn cho các gói trong hồ sơ so với các gói ngoài hồ sơ



Đảm bảo QOS cho mỗi kết nối

- **Thực tế:** không thể hỗ trợ lưu lượng theo yêu cầu vượt quá khả năng của link



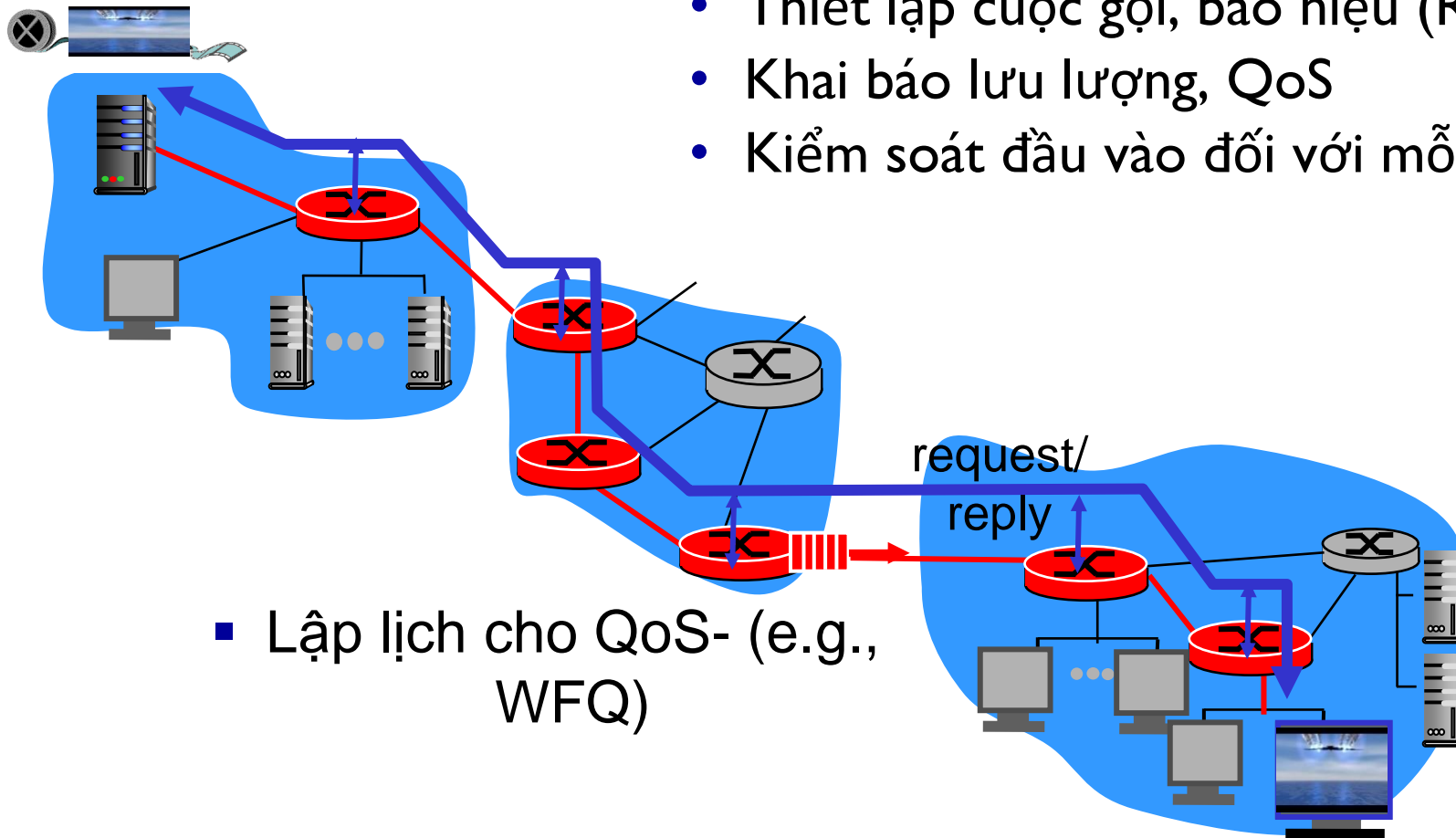
— Nguyên tắc 4 —

Cuộc gọi tới: khai báo luồng cần thiết, mạng có thể block cuộc gọi (vd busy signal) nếu không thể đáp ứng yêu cầu

Đảm bảo QoS

- Chuẩn bị tài nguyên

- Thiết lập cuộc gọi, báo hiệu (RSVP)
- Khai báo lưu lượng, QoS
- Kiểm soát đầu vào đối với mỗi phần tử



- Lập lịch cho QoS- (e.g., WFQ)

Multimedia networking: outline

- 9.1 multimedia networking applications
- 9.2 streaming *stored* video
- 9.3 voice-over-IP
- 9.4 protocols for *real-time* conversational applications
- 9.5 network support for multimedia