



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẢNG ĐỘC QUYỀN SÁNG CHẾ

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ

(11)



1-0024167

(51)⁷ H04N 21/231; H04N 21/234; H04N 21/2343; H04N 21/235; H04N 21/845; H04N 21/2662; H04N 21/435; H04N 21/438; H04N 21/44; H04N 21/84; H04L 29/06; H04N 21/258 (13) B

(21) 1-2014-03950

(22) 25/04/2013

(86) PCT/US2013/038247 25/04/2013

(87) WO2013/163448 31/10/2013

(30) 13/456,474 26/04/2012 US

(45) 25/06/2020 387

(43) 25/03/2015 324A

(73) QUALCOMM INCORPORATED (US)

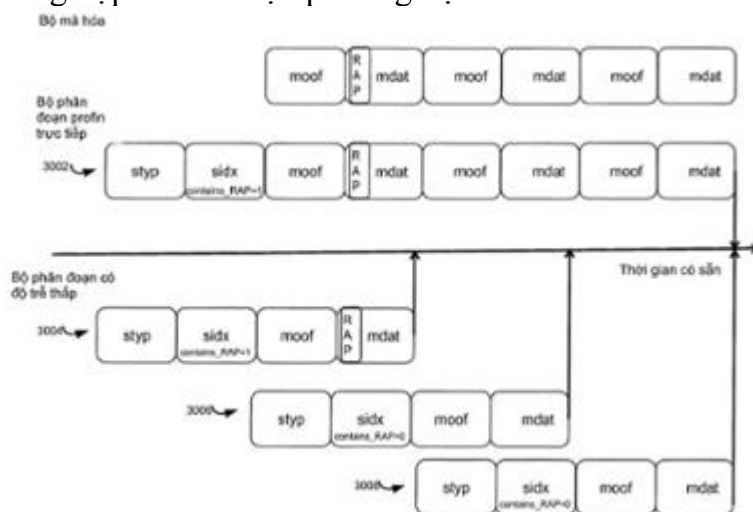
Attn: International IP Administration, 5775 Morehouse Drive, San Diego, California 92121, United States of America

(72) LUBY, Michael G. (US); WATSON, Mark (GB); VICISANO, Lorenzo (US); PAKZAD, Payam (US); WANG, Bin (US); CHEN, Ying (CN); STOCKHAMMER, Thomas (DE); BORRAN, Jaber Mohammad (US).

(74) Công ty TNHH Quốc tế D & N (D&N INTERNATIONAL CO.,LTD.)

(54) PHƯƠNG PHÁP TẠO CẤU TRÚC DỮ LIỆU PHỤC VỤ TRÊN MÁY CHỦ PHƯƠNG TIỆN

(57) Sáng chế đề cập đến hệ thống truyền liên tục yêu cầu theo khối cung cấp quy trình truyền liên tục có độ trễ thấp đối với phân trình diễn phương tiện. Các phân đoạn phương tiện được tạo ra theo giao thức mã hóa. Mỗi phân đoạn phương tiện bao gồm điểm truy cập ngẫu nhiên. Các đoạn phương tiện được mã hóa theo cùng giao thức. Các phân đoạn phương tiện được tổng hợp từ các đoạn phương tiện.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến hệ thống và phương pháp truyền liên tục phương tiện truyền thông cải tiến, cụ thể hơn đến hệ thống và phương pháp thích nghi với các điều kiện mạng và bộ đệm để tối ưu hóa việc trình diễn phương tiện truyền liên tục và cho phép phân phối đồng thời hiệu quả hoặc kịp thời dữ liệu phương tiện truyền liên tục.

Tình trạng kỹ thuật của sáng chế

Việc phân phối phương tiện truyền liên tục có thể ngày càng quan trọng khi âm thanh và video chất lượng cao ngày càng được phân phối phổ biến hơn qua mạng dựa trên gói, chẳng hạn như Internet, mạng di động và không dây, mạng qua đường dây điện, và các loại mạng khác. Chất lượng trình diễn bởi phương tiện truyền liên tục được phân phối có thể phụ thuộc vào một số yếu tố, bao gồm độ phân giải (hoặc các thuộc tính khác) của nội dung gốc, chất lượng mã hóa của nội dung gốc, khả năng mà các thiết bị thu có thể giải mã và trình diễn phương tiện, tính kịp thời và chất lượng của tín hiệu thu được ở bộ thu, v.v.. Để tạo ra trải nghiệm cảm nhận phương tiện truyền liên tục tốt, việc truyền tải và tính kịp thời của tín hiệu thu được tại bộ thu có thể đặc biệt quan trọng. Việc truyền tải tốt có thể mang lại độ trung thực của dòng thu được ở bộ thu so với những gì mà bên gửi gửi, trong khi tính kịp thời có thể biểu diễn mức nhanh chóng bộ thu có thể bắt đầu phát ra nội dung sau khi có yêu cầu ban đầu đối với nội dung đó.

Hệ thống phân phối phương tiện có thể được mô tả là hệ thống có các nguồn phương tiện, các đích phương tiện và các kênh (theo thời gian và/hoặc không gian) tách riêng nguồn và đích. Thông thường, nguồn bao gồm bộ phát có thể truy cập phương tiện dưới dạng quản lý được kiểu điện tử, và bộ thu có khả năng kiểm soát điện tử việc thu phương tiện (hoặc tương đương) và cung cấp nó cho người sử dụng phương tiện (ví dụ, người dùng có thiết bị hiển thị được ghép nối theo cách nào đó với bộ thu, thiết bị lưu trữ hoặc phần tử, kênh khác, v.v.).

Trong khi nhiều biến thể có thể được tạo ra, trong ví dụ phổ biến, hệ thống phân phối phương tiện có một hoặc nhiều máy chủ có quyền truy cập vào nội dung phương

tiện dưới dạng điện tử, và một hoặc nhiều hệ thống hoặc thiết bị máy khách có thể gửi yêu cầu phương tiện đến máy chủ, và máy chủ truyền phương tiện bằng cách sử dụng bộ phát như là một phần của máy chủ, truyền đến bộ thu ở máy khách để phương tiện thu được có thể được sử dụng bởi máy khách theo cách nào đó. Trong ví dụ đơn giản, chỉ có một máy chủ và một máy khách, với một yêu cầu và phản hồi cho trước, nhưng không phải lúc nào cũng như vậy.

Thông thường, hệ thống phân phối phương tiện có thể được mô tả là mô hình "tải về" hoặc mô hình "truyền liên tục". Mô hình "tải về" có thể được đặc trưng bởi tính độc lập định thời giữa việc phân phối dữ liệu phương tiện và việc phát lại phương tiện đến người dùng hoặc thiết bị thu.

Ví dụ, phương tiện được tải về đủ sớm trước khi nó được cần đến hoặc sẽ được sử dụng và khi nó được sử dụng, càng sớm càng tốt để sẵn sàng có sẵn ở phía người nhận. Việc phân phối trong ngữ cảnh tải về thường được thực hiện bằng cách sử dụng giao thức truyền tập tin, chẳng hạn như, HTTP, FTP hoặc phân phối tập tin truyền tải đơn hướng (FLUTE - File Delivery over Unidirectional Transport) và tốc độ phân phối có thể được xác định bằng giao thức kiểm soát tắc nghẽn và/hoặc dòng cơ bản, như TCP/IP. Hoạt động của giao thức kiểm soát tắc nghẽn và/hoặc có thể độc lập với việc phát ra phương tiện đến người dùng hoặc đến thiết bị đích, mà có thể xảy ra đồng thời với việc tải về hoặc vào thời điểm khác.

Chế độ "truyền liên tục" có thể được đặc trưng bởi sự kết hợp chặt chẽ giữa việc định thời phân phối dữ liệu phương tiện và việc phát ra phương tiện cho người dùng hoặc thiết bị thu. Việc phân phối trong ngữ cảnh này thường được thực hiện bằng cách sử dụng giao thức truyền liên tục, chẳng hạn như RTSP (giao thức truyền liên tục thời gian thực) để kiểm soát và RTP (giao thức truyền tải thời gian thực) cho dữ liệu phương tiện. Tốc độ phân phối có thể được xác định bởi máy chủ truyền liên tục, thường phù hợp với tốc độ phát ra dữ liệu.

Một số nhược điểm của mô hình "tải về" có thể là, do tính độc lập định thời của việc phân phối và phát ra, hoặc dữ liệu phương tiện có thể không có sẵn khi nó được cần đến để phát ra (ví dụ do băng thông có sẵn ít hơn so với tốc độ dữ liệu phương tiện), làm cho việc phát ra ngừng trong giây lát ("tri hoãn"), dẫn đến trải nghiệm người dùng kém, hoặc dữ liệu phương tiện có thể cần được tải về rất sớm trước khi được phát ra (ví dụ do băng thông có sẵn lớn hơn tốc độ dữ liệu phương tiện), sử dụng tài nguyên

lưu trữ trên thiết bị thu mà có thể khan hiếm, và sử dụng tài nguyên mạng có giá trị cho việc phân phối mà có thể bị lãng phí nếu nội dung cuối cùng không được phát ra hoặc sử dụng theo cách khác.

Ưu điểm của mô hình "tải về" có thể là công nghệ cần thiết để thực hiện việc tải về này, ví dụ như HTTP, rất mạnh, được triển khai rộng rãi và có thể áp dụng trên rất nhiều ứng dụng. Máy chủ tải về và các giải pháp cho khả năng mở rộng của việc tải tập tin (ví dụ, máy chủ web HTTP và mạng phân phối nội dung) có thể dễ dàng có sẵn, khiến cho việc triển khai dịch vụ dựa trên công nghệ này trở nên đơn giản và ít chi phí.

Một số nhược điểm của mô hình "truyền liên tục" có thể là tốc độ phân phối dữ liệu phương tiện nói chung không phù hợp với băng thông có sẵn trên kết nối từ máy chủ đến máy khách và cần đến máy chủ trực tuyến chuyên dụng hoặc kiến trúc mạng phức tạp hơn cung cấp băng thông và độ trễ đảm bảo. Mặc dù các hệ thống truyền liên tục tồn tại có hỗ trợ thay đổi tốc độ phân phối dữ liệu theo băng thông có sẵn (ví dụ công nghệ truyền liên tục thích ứng Adobe Flash), chúng thường không hiệu quả như các giao thức điều khiển dòng truyền tải về như TCP sử dụng tất cả băng thông có sẵn.

Gần đây, các hệ thống phân phối phương tiện mới dựa trên sự kết hợp của mô hình "truyền liên tục" và "tải về" đã được phát triển và triển khai. Ví dụ về mô hình như vậy ở đây được gọi là mô hình "truyền liên tục yêu cầu theo khối", trong đó máy khách phương tiện yêu cầu các khối dữ liệu phương tiện từ cơ sở hạ tầng phục vụ sử dụng giao thức tải về, chẳng hạn như HTTP. Mỗi quan tâm trong hệ thống này có thể là khả năng để bắt đầu phát ra dòng, ví dụ, giải mã và kết xuất các dòng âm thanh và video thu được bằng cách sử dụng máy tính cá nhân và hiển thị video trên màn hình máy tính và phát âm thanh qua loa gắn sẵn, hoặc theo ví dụ khác, giải mã và kết xuất các dòng âm thanh và video thu được bằng cách sử dụng hộp chuyển đổi và hiển thị video trên thiết bị hiển thị truyền hình và phát âm thanh qua hệ thống âm thanh lập thể.

Các mối quan tâm khác, chẳng hạn như, khả năng giải mã các khối nguồn đủ nhanh để theo kịp với tốc độ truyền liên tục nguồn, để giảm thiểu độ trễ giải mã và giảm sử dụng tài nguyên CPU có sẵn chính là các vấn đề. Một mối quan tâm nữa là cung cấp giải pháp phân phối truyền liên tục mạnh mẽ và có khả năng mở rộng cho phép các thành phần của hệ thống gặp sự cố mà không ảnh hưởng xấu đến chất lượng của dòng được phân phối đến bộ thu. Các vấn đề khác có thể xảy ra dựa trên thông tin thay đổi nhanh chóng về phần trình diễn, khi nó đang được phân phối. Vì vậy, mong

muốn là có thể cung cấp quy trình và thiết bị cải tiến để khắc phục được các vấn đề nêu trên.

Bản chất kỹ thuật của sáng chế

Sáng chế đề xuất hệ thống truyền liên tục yêu cầu theo khối để cải thiện trải nghiệm người dùng và hiệu quả băng thông của hệ thống này, mà thường sử dụng hệ thống thu thập để tạo ra dữ liệu dưới dạng được phục vụ bởi máy chủ tập tin truyền thống (HTTP, FTP, hoặc máy chủ tương tự), trong đó hệ thống thu thập nhận nội dung và chuẩn bị nội dung dưới dạng tập tin hoặc các phần tử dữ liệu được phục vụ bởi máy chủ tập tin, mà có thể hoặc không bao gồm bộ nhớ đệm.

Theo một phương án, máy chủ phương tiện của hệ thống truyền liên tục yêu cầu theo khối cho phép truyền liên tục nội dung trình diễn phương tiện với độ trễ thấp. Các tập tin phân đoạn phương tiện tương đối lớn để truyền liên tục profin trực tiếp có thể được tổng hợp từ các đoạn phương tiện tương đối nhỏ hơn để truyền liên tục với độ trễ thấp. Các phân đoạn phương tiện và đoạn phương tiện được mã hóa theo cùng giao thức mã hóa.

Phần mô tả chi tiết sau đây cùng với các hình vẽ kèm theo sẽ cung cấp hiểu biết tốt hơn về bản chất và các ưu điểm của sáng chế.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ thể hiện các phần tử của hệ thống truyền liên tục yêu cầu theo khối theo các phương án của sáng chế;

Fig.2 là sơ đồ thể hiện hệ thống truyền liên tục yêu cầu theo khối trên Fig.1, thể hiện chi tiết hơn về các phần tử của hệ thống máy khách được ghép nối với cơ sở hạ tầng phục vụ khối ("BSI") để thu dữ liệu được xử lý bởi hệ thống thu thập nội dung;

Fig.3 là sơ đồ thể hiện cài đặt phần cứng/phần mềm của hệ thống thu thập;

Fig.4 là sơ đồ thể hiện cài đặt phần cứng/phần mềm của hệ thống máy khách;

Fig.5 là sơ đồ thể hiện các cấu trúc có thể có của bộ phận lưu trữ nội dung được thể hiện trên Fig.1, bao gồm các phân đoạn và tập tin mô tả trình diễn phương tiện ("MPD"), và sự tách rời phân đoạn, thời gian, và cấu trúc khác trong tập tin MPD;

Fig.6 là sơ đồ thể hiện các chi tiết của phân đoạn nguồn điển hình, chẳng hạn như, có thể được lưu trữ trong bộ phận lưu trữ nội dung được thể hiện trên Fig.1 và Fig.5;

Fig.7a và Fig.7B là các sơ đồ thể hiện việc lập chỉ mục đơn giản và phân cấp trong các tập tin;

Fig.8(a) là sơ đồ thể hiện việc định cỡ khối thay đổi theo các điểm tìm kiếm được đồng chỉnh trên các phiên bản của dòng phương tiện;

Fig.8(b) là sơ đồ thể hiện việc định cỡ khối thay đổi theo điểm tìm kiếm không được đồng chỉnh trên các phiên bản của dòng phương tiện;

Fig.9(a) là sơ đồ thể hiện bảng siêu dữ liệu;

Fig.9(b) là sơ đồ thể hiện việc truyền các khối và bảng siêu dữ liệu từ máy chủ đến máy khách;

Fig.10 là sơ đồ thể hiện các khối độc lập với các biên RAP (Random Access Point – điểm truy cập ngẫu nhiên);

Fig.11 là sơ đồ thể hiện việc định thời liên tục và gián đoạn trên các phân đoạn;

Fig.12 là sơ đồ thể hiện một khía cạnh của khối có khả năng mở rộng;

Fig.13 là sơ đồ thể hiện dạng biểu diễn đồ họa về sự tiến triển của các biến nhất định trong hệ thống truyền liên tục yêu cầu theo khối theo thời gian;

Fig.14 là sơ đồ thể hiện dạng biểu diễn đồ họa khác về sự tiến triển của các biến nhất định trong hệ thống truyền liên tục yêu cầu theo thời gian;

Fig.15 là sơ đồ thể hiện lưới dạng ô của các trạng thái dưới dạng hàm của các giá trị ngưỡng;

Fig.16 là lưu đồ thể hiện quy trình có thể được thực hiện trong bộ thu mà có thể yêu cầu các khối đơn và nhiều khối cho mỗi yêu cầu;

Fig.17 là lưu đồ thể hiện quy trình tạo đường ống truyền linh hoạt;

Fig.18 là sơ đồ thể hiện ví dụ về tập yêu cầu ứng viên, mức ưu tiên của chúng, và các kết nối mà chúng có thể được phát ra trên đó, tại thời điểm nhất định;

Fig.19 là sơ đồ thể hiện ví dụ về tập yêu cầu ứng viên, mức ưu tiên của chúng, và các kết nối mà chúng có thể được phát ra, phát triển từ thời điểm này sang thời điểm khác;

Fig.20 là lưu đồ thể hiện việc chọn proxy máy chủ đệm nhất quán dựa trên định danh tập tin;

Fig.21 là sơ đồ thể hiện việc xác định cú pháp cho ngôn ngữ biểu hiện phù hợp;
 Fig.22 là sơ đồ thể hiện ví dụ về hàm băm phù hợp;
 Fig.23 là sơ đồ thể hiện một ví dụ về các quy tắc xây dựng định danh tập tin;
 Fig.24(a) – Fig.24(e) là sơ đồ thể hiện biến động băng thông của kết nối TCP;
 Fig.25 là sơ đồ thể hiện các yêu cầu HTTP cho dữ liệu nguồn và dữ liệu sửa chữa;

Fig.26 là sơ đồ thể hiện ví dụ về thời gian sửa đổi kênh có và không có FEC (Forward Error Correction - Sửa lỗi chuyển tiếp);

Fig.27 là sơ đồ thể hiện các chi tiết của bộ tạo phân đoạn sửa chữa, như là một phần của hệ thống thu thập được thể hiện trên Fig.1, tạo ra phân đoạn sửa chữa từ các phân đoạn nguồn và các tham số kiểm soát;

Fig.28 là sơ đồ thể hiện mối quan hệ giữa các khối nguồn và khối sửa chữa;

Fig.29 là sơ đồ thể hiện thủ tục cho dịch vụ trực tiếp tại các thời điểm khác nhau ở máy khách;

Fig.30 là sơ đồ thể hiện mối quan hệ giữa các đoạn phương tiện cho quy trình truyền liên tục có độ trễ thấp và các đoạn phương tiện.

Trên các hình vẽ, các mục giống nhau được tham chiếu đến với số chỉ dẫn giống nhau và các chỉ mục phụ được cung cấp trong dấu ngoặc đơn để chỉ ra nhiều phiên bản của các mục tương tự hoặc giống nhau. Trừ khi được chỉ rõ khác đi, các chỉ mục phụ cuối cùng (ví dụ, "N" hoặc "M") không dự định giới hạn ở giá trị cụ thể bất kỳ và số phiên bản của một mục có thể khác với số phiên bản của mục khác ngay cả khi cùng một số được minh họa và chỉ mục phụ được sử dụng lại.

Mô tả chi tiết sáng chế

Như được mô tả ở đây, mục tiêu của hệ thống truyền liên tục là chuyển phương tiện từ vị trí lưu trữ của nó (hoặc vị trí mà nó đang được tạo ra) đến vị trí mà nó đang được sử dụng, tức là, trình diễn cho người dùng hoặc "được sử dụng hết" bởi người sử dụng hoặc hệ thống sử dụng điện tử. Lý tưởng nhất, hệ thống truyền liên tục có thể cung cấp khả năng phát lại không gián đoạn (hay rộng hơn, "sử dụng" không gián đoạn) ở đầu nhận và có thể bắt đầu phát một dòng hoặc tập hợp dòng ngay sau khi người dùng yêu cầu dòng này hoặc các dòng này. Vì lý do hiệu quả, cũng mong muốn rằng mỗi dòng được dừng lại khi người dùng chỉ ra rằng dòng này không còn cần thiết

nữa, chẳng hạn như, khi người dùng đang chuyển từ một dòng sang dòng khác hoặc tuân theo việc trình diễn của một dòng, ví dụ, dòng "phụ đề". Nếu thành phần phương tiện, chẳng hạn như video, tiếp tục được trình diễn, nhưng dòng khác được chọn để trình diễn thành phần phương tiện này, thường ưu tiên chiếm băng thông giới hạn với dòng mới này và ngăn chặn các dòng cũ.

Hệ thống truyền liên tục yêu cầu theo khối theo các phương án được mô tả ở đây mang lại nhiều lợi ích. Nên hiểu rằng hệ thống khả thi không cần phải bao gồm tất cả các tính năng được mô tả ở đây, do một số ứng dụng có thể cung cấp trải nghiệm thỏa đáng phù hợp với số lượng tính năng ít hơn tất cả các tính năng được mô tả ở đây.

Truyền liên tục HTTP

Kỹ thuật truyền liên tục HTTP là loại kỹ thuật truyền liên tục cụ thể. Với kỹ thuật truyền liên tục HTTP, các nguồn có thể là máy chủ web tiêu chuẩn và mạng phân phối nội dung (CDN - content delivery network) và có thể sử dụng tiêu chuẩn HTTP. Kỹ thuật này có thể liên quan đến việc phân đoạn dòng và việc sử dụng nhiều dòng, tất cả đều trong ngữ cảnh của yêu cầu HTTP tiêu chuẩn. Phương tiện, chẳng hạn như video, có thể được mã hóa ở nhiều tốc độ bit để tạo thành các phiên bản khác nhau, hoặc các dạng biểu diễn khác nhau. Các thuật ngữ "phiên bản" và "dạng biểu diễn" được dùng đồng nghĩa ở đây. Mỗi phiên bản hoặc dạng biểu diễn có thể được chia thành từng mảnh nhỏ, có lẽ có độ lớn vài giây, để tạo thành phân đoạn. Sau đó, mỗi phân đoạn có thể được lưu trữ trên máy chủ web hoặc CDN như một tập tin riêng biệt.

Ở phía máy khách, các yêu cầu có thể được thực hiện, bằng cách sử dụng HTTP, cho các phân đoạn riêng biệt được ghép lại với nhau liên tục bởi máy khách. Máy khách có thể chuyển sang các tốc độ dữ liệu khác nhau dựa trên băng thông có sẵn. Máy khách cũng có thể yêu cầu nhiều dạng biểu diễn, mỗi dạng biểu diễn trình diễn một thành phần phương tiện khác nhau, và có thể trình diễn phương tiện ở các dạng biểu diễn này một cách phối hợp và đồng bộ. Tín hiệu kích hoạt cho việc chuyển đổi có thể bao gồm việc chiếm bộ đệm và các số đo mạng. Khi hoạt động trong trạng thái ổn định, máy khách có thể xác định nhịp độ đối với các yêu cầu đến máy chủ để duy trì tình trạng chiếm bộ đệm đích.

Ưu điểm của kỹ thuật truyền liên tục HTTP có thể bao gồm khả năng thích ứng tốc độ bit, khởi động nhanh và tìm kiếm nhanh, và giảm thiểu phân phối không cần

thiết. Các lợi thế này đến từ việc kiểm soát phân phối chỉ một thời gian ngắn trước khi phát ra, tận dụng tối đa băng thông có sẵn (qua phương tiện có tốc độ bit thay đổi), và tối ưu hóa việc phân đoạn dòng và các thủ tục máy khách thông minh.

Thông tin mô tả trình diễn phương tiện có thể được cung cấp cho máy khách truyền liên tục HTTP sao cho máy khách có thể sử dụng tập hợp tập tin (ví dụ ở các định dạng được quy định bởi 3GPP, ở đây được gọi là phân đoạn 3gp) để cung cấp dịch vụ truyền liên tục cho người dùng. Thông tin mô tả trình diễn phương tiện, và có thể là các cập nhật của thông tin mô tả trình diễn phương tiện này, mô tả trình diễn phương tiện là tập hợp có cấu trúc của các phân đoạn, mỗi phân đoạn chứa các thành phần phương tiện sao cho máy khách có thể trình diễn phương tiện được bao gồm một cách đồng bộ và có thể cung cấp các tính năng tiên tiến, chẳng hạn như, tìm kiếm, chuyển đổi tốc độ bit và phân trình diễn chung các thành phần phương tiện ở các dạng biểu diễn khác nhau. Máy khách có thể sử dụng thông tin mô tả trình diễn phương tiện theo nhiều cách khác nhau cho việc cung cấp dịch vụ. Cụ thể, từ thông tin mô tả trình diễn phương tiện này, máy khách truyền liên tục HTTP có thể xác định phân đoạn nào trong tập hợp có thể được truy cập để dữ liệu hữu ích với khả năng của máy khách và người dùng trong dịch vụ truyền liên tục.

Theo một số phương án, thông tin mô tả trình diễn phương tiện có thể tĩnh, mặc dù các phân đoạn có thể được tạo ra theo cách động. Thông tin mô tả trình diễn phương tiện có thể nhỏ gọn nhất có thể để giảm thiểu thời gian truy cập và tải về cho dịch vụ. Khả năng kết nối máy chủ chuyên dụng khác có thể được giảm thiểu, ví dụ như việc đồng bộ hóa định thời đều đặn hoặc thường xuyên giữa máy khách và máy chủ.

Phân trình diễn phương tiện có thể được xây dựng để cho phép truy cập bằng thiết bị đầu cuối với khả năng khác nhau, chẳng hạn như, truy cập vào các loại mạng truy cập khác nhau, các điều kiện mạng hiện thời khác nhau, kích thước màn hình, tốc độ bit truy cập và hỗ trợ mã hóa-giải mã. Sau đó, máy khách có thể trích xuất thông tin thích hợp để cung cấp dịch vụ truyền liên tục cho người dùng.

Thông tin mô tả trình diễn phương tiện cũng có thể cho phép khả năng triển khai linh hoạt và nhỏ gọn theo các yêu cầu.

Trong trường hợp đơn giản nhất, mỗi dạng biểu diễn thay thế có thể được lưu trữ trong một tập tin 3GP, nghĩa là tập tin phù hợp như được định nghĩa trong 3GPP

TS26.244, hoặc bất kỳ tập tin nào khác tuân theo định dạng tập tin phương tiện cơ sở ISO theo được chỉ rõ trong chuẩn ISO/IEC 14496 - 12 hoặc thông số kỹ thuật được suy ra (chẳng hạn như định dạng tập tin 3GP được mô tả trong đặc tả kỹ thuật 3GPP 26.244). Trong phần mô tả dưới đây, khi đề cập đến tập tin 3GP, cần phải hiểu rằng chuẩn ISO/IEC 14496-12 và các thông số kỹ thuật được suy ra có thể ánh xạ tất cả các tính năng được mô tả sang các định dạng tập tin phương tiện cơ sở theo tiêu chuẩn ISO chung như được chỉ rõ trong tiêu chuẩn ISO/IEC 14496 - 12 hoặc thông số kỹ thuật được suy ra bất kỳ. Sau đó, máy khách có thể yêu cầu phần ban đầu của tập tin để biết được siêu dữ liệu phương tiện (thường được lưu trữ trong hộp tiêu đề Phim, còn được gọi là hộp "moov") cùng với các thời điểm đoạn phim và độ dịch byte. Sau đó, máy khách có thể phát ra yêu cầu lấy một phần HTTP để thu được các đoạn phim theo yêu cầu.

Trong một số phương án, có thể mong muốn chia mỗi dạng biểu diễn thành một số phân đoạn. Trong trường hợp mà định dạng phân đoạn dựa vào định dạng tập tin 3GP, thì phân đoạn chứa các lát thời gian không chồng lấn của các đoạn phim, được gọi là "chia theo thời gian". Mỗi trong số phân đoạn này có thể chứa nhiều đoạn phim và mỗi trong số đó có thể là một tập tin 3GP hợp lệ theo đúng nghĩa của nó. Theo phương án khác, dạng biểu diễn được chia thành phân đoạn ban đầu có chứa siêu dữ liệu (thường là hộp tiêu đề phim "moov") và tập hợp phân đoạn phương tiện, mỗi phân đoạn chứa dữ liệu phương tiện và phần nối của phân đoạn ban đầu và phân đoạn phương tiện bất kỳ tạo thành tập tin phương tiện 3GP hợp lệ cũng như phần nối của phân đoạn ban đầu và tất cả phân đoạn phương tiện của dạng biểu diễn tạo thành tập tin 3GP hợp lệ. Toàn bộ phần trình diễn có thể được tạo ra bằng cách phát ra từng phân đoạn lần lượt, ánh xạ các dấu thời gian cục bộ trong tập tin sang thời gian trình diễn toàn cầu theo thời gian bắt đầu của mỗi dạng biểu diễn.

Cần lưu ý rằng trong bản mô tả này việc tham chiếu đến "phân đoạn" cần được hiểu là bao gồm đối tượng dữ liệu bất kỳ được xây dựng hoàn toàn hoặc một phần hoặc được đọc từ phương tiện lưu trữ hoặc thu được theo cách khác nhờ yêu cầu giao thức tải về tập tin, bao gồm, chẳng hạn như, yêu cầu HTTP. Ví dụ, trong trường hợp HTTP, các đối tượng dữ liệu có thể được lưu trữ trong tập tin thực tế nằm trên đĩa hoặc phương tiện lưu trữ khác kết nối với hoặc tạo thành một phần của máy chủ HTTP, hoặc các đối tượng dữ liệu có thể được xây dựng bởi kịch bản CGI, hoặc

chương trình được thực hiện động khác, mà được thực hiện để phản hồi yêu cầu HTTP. Các thuật ngữ "tập tin" và "phân đoạn" được dùng đồng nghĩa ở đây, trừ khi được quy định khác. Trong trường hợp HTTP, phân đoạn có thể được coi là phần thân thực thể của phản hồi yêu cầu HTTP.

Các thuật ngữ "dạng biểu diễn" và "mục nội dung" được dùng đồng nghĩa ở đây. Trong nhiều ví dụ, phần trình diễn là âm thanh, video hoặc phần trình diễn phương tiện khác mà có thời gian "phát ra" xác định, nhưng các biến thể khác cũng có thể được sử dụng.

Các thuật ngữ "khối" và "đoạn" được dùng đồng nghĩa ở đây, trừ khi có quy định khác và thường đề cập đến tập hợp nhỏ nhất của dữ liệu được lập chỉ mục. Dựa trên việc lập chỉ mục có sẵn, máy khách có thể yêu cầu các phần khác nhau của đoạn trong các yêu cầu HTTP khác nhau, hoặc có thể yêu cầu một hoặc nhiều đoạn liên tiếp hoặc các phần của đoạn trong một yêu cầu HTTP. Trong trường hợp phân đoạn dựa trên định dạng tập tin phương tiện cơ sở theo tiêu chuẩn ISO hoặc phân đoạn dựa trên định dạng tập tin 3GP được sử dụng, đoạn thường đề cập đến đoạn phim được định nghĩa là sự kết hợp của hộp tiêu đề đoạn phim ('hoof') và hộp dữ liệu phương tiện ('mdat').

Ở đây, mạng mang dữ liệu được giả định là dựa trên gói để đơn giản hóa việc mô tả ở đây, với sự thừa nhận rằng, sau khi đọc sáng chế, người có hiểu biết trung bình trong lĩnh vực kỹ thuật có thể áp dụng các phương án của sáng chế được mô tả ở đây cho các loại mạng truyền dẫn khác, chẳng hạn như mạng dòng bit liên tục.

Ở đây, mã FEC được giả định để cung cấp khả năng bảo vệ chống lại thời gian phân phối dữ liệu lâu và thay đổi, để đơn giản hóa việc mô tả ở đây, với sự thừa nhận rằng, sau khi đọc sáng chế, người có hiểu biết trung bình trong lĩnh vực kỹ thuật có thể áp dụng các phương án của sáng chế cho các loại vấn đề truyền dẫn dữ liệu khác, chẳng hạn như, hỏng dữ liệu do đảo bit. Ví dụ, khi không có FEC, nếu phần cuối cùng của đoạn được yêu cầu đến chậm hơn nhiều hoặc có biến đổi lớn về thời gian đến so với các phần trước của đoạn, thì thời gian sửa đổi nội dung có thể lớn và thay đổi, trong khi đó bằng cách sử dụng FEC và các yêu cầu song song, chỉ phần lớn dữ liệu được yêu cầu cho đoạn cần đến đến trước khi nó có thể được khôi phục, do đó làm giảm thời gian sửa đổi nội dung và khả năng thay đổi về thời gian sửa đổi nội dung. Trong bản mô tả này, có thể giả định rằng dữ liệu cần được mã hóa (tức là, dữ liệu

nguồn) đã được chia thành các "ký hiệu" có chiều dài bằng nhau, mà có thể có chiều dài bất kỳ (xuống đến một bit), nhưng các ký hiệu có thể có độ dài khác nhau đối với các phần khác nhau của dữ liệu, ví dụ, các kích thước ký hiệu khác nhau có thể được sử dụng cho các khối dữ liệu khác nhau.

Trong bản mô tả này, để đơn giản hóa việc mô tả ở đây, giả sử rằng FEC được áp dụng cho "khối" hoặc "đoạn" của dữ liệu tại một thời điểm, tức là, "khối" là "khối nguồn" để phục vụ các mục đích mã hóa và giải mã FEC. Thiết bị máy khách có thể sử dụng chỉ mục phân đoạn được mô tả ở đây để giúp xác định cấu trúc khối nguồn của phân đoạn. Người có hiểu biết trung bình trong lĩnh vực kỹ thuật có thể áp dụng các phương án của sáng chế cho các loại cấu trúc khối nguồn khác, ví dụ như, khối nguồn có thể là một phần của đoạn, hoặc bao gồm một hoặc nhiều đoạn hoặc phần của đoạn.

Các mã FEC được xem xét sử dụng cùng với quy trình truyền liên tục yêu cầu theo khối thường là mã FEC có hệ thống, tức là, các ký hiệu nguồn của khối nguồn có thể được bao gồm như là một phần của việc mã hóa khối nguồn và do đó các ký hiệu nguồn được truyền đi. Người có hiểu biết trung bình trong lĩnh vực kỹ thuật hiểu rằng các phương án được mô tả ở đây cũng có thể áp dụng cho mã FEC không có hệ thống. Bộ mã hóa FEC có hệ thống tạo ra, từ một khối nguồn gồm các ký hiệu nguồn, một số ký hiệu sửa chữa và sự kết hợp của ít nhất một số ký hiệu trong số các ký hiệu nguồn và ký hiệu sửa chữa là các ký hiệu mã hóa được gửi qua kênh biểu diễn khối nguồn. Một số mã FEC có thể hữu ích để tạo ra một cách có hiệu quả nhiều ký hiệu sửa chữa khi cần thiết, chẳng hạn như "mã thêm thông tin" hoặc "mã nguồn (fountain)" và các ví dụ về các mã này bao gồm "mã phản ứng chuỗi" và "mã phản ứng chuỗi nhiều giai đoạn". Các mã FEC khác, như mã Reed-Solomon thực tế có thể chỉ tạo ra một số lượng hạn chế các ký hiệu sửa chữa cho mỗi khối nguồn.

Giả định rằng trong nhiều ví dụ máy khách được ghép nối với máy chủ phương tiện hoặc nhiều máy chủ phương tiện và máy khách yêu cầu phương tiện truyền liên tục trên một kênh hoặc nhiều kênh truyền hình từ máy chủ phương tiện hoặc nhiều máy chủ phương tiện. Tuy nhiên, các cách sắp xếp liên quan khác cũng có thể được thực hiện.

Các ví dụ về các lợi ích

Với quy trình truyền liên tục yêu cầu theo khối, máy khách phương tiện duy trì sự ghép nối giữa định thời các yêu cầu theo khối và định thời phát ra phương tiện đến người dùng. Mô hình này có thể giữ lại các ưu điểm của mô hình “tải về” được mô tả ở trên, trong khi tránh được một số nhược điểm xuất phát từ việc ngắt ghép nối thông thường giữa phát ra phương tiện với phân phối dữ liệu. Mô hình truyền liên tục yêu cầu theo khối sử dụng các cơ chế kiểm soát tốc độ và tắc nghẽn có sẵn trong giao thức truyền tải, chẳng hạn như TCP, để đảm bảo rằng băng thông có sẵn tối đa được sử dụng cho dữ liệu phương tiện. Ngoài ra, việc chia thông tin trình diễn phương tiện thành các khối cho phép mỗi khối dữ liệu phương tiện mã hóa được chọn từ tập hợp các phần mã hóa có sẵn.

Thao tác chọn này có thể dựa trên số lượng tiêu chí bất kỳ, bao gồm sự so khớp của tốc độ dữ liệu phương tiện với băng thông có sẵn, ngay cả khi băng thông có sẵn thay đổi theo thời gian, so khớp độ phân giải phương tiện hay độ phức tạp giải mã với cấu hình hoặc khả năng của máy khách, hoặc so khớp với sở thích người dùng, chẳng hạn như ngôn ngữ. Thao tác chọn cũng có thể bao gồm tải về và trình diễn các thành phần phụ trợ, chẳng hạn như, các thành phần khả năng truy cập, chú thích rời, phụ đề, video ngôn ngữ ký hiệu, v.v.. Các ví dụ về hệ thống hiện có sử dụng mô hình truyền liên tục yêu cầu theo khối bao gồm Move Networks™, Microsoft Smooth Streaming và Apple iPhone™ Streaming Protocol.

Thông thường, mỗi khối dữ liệu phương tiện có thể được lưu trữ trên máy chủ như là một tập tin riêng biệt và sau đó giao thức, như giao thức HTTP, được sử dụng, kết hợp với phần mềm máy chủ HTTP được thực hiện trên máy chủ, để yêu cầu tập tin dưới dạng một đơn vị. Thông thường, máy khách được cung cấp tập tin siêu dữ liệu, có thể, ví dụ ở định dạng XML (Extensible Markup Language – ngôn ngữ đánh dấu mở rộng) hoặc định dạng văn bản danh sách phát hoặc ở định dạng nhị phân, mô tả các tính năng của thông tin trình diễn phương tiện, chẳng hạn như, các phần mã hóa có sẵn (ví dụ, băng thông yêu cầu, độ phân giải, tham số mã hóa, loại phương tiện, ngôn ngữ), mà ở đây thường được gọi là "dạng biểu diễn", và cách thức mà các phần mã hóa được chia thành các khối. Ví dụ, siêu dữ liệu có thể bao gồm một URL (Uniform Resource Locator - phần tử định vị tài nguyên đồng nhất) cho mỗi khối. Bản thân URL có thể cung cấp sơ đồ, chẳng hạn như, được thêm chuỗi "http://" để chỉ ra rằng giao thức mà

sẽ được sử dụng để truy cập tài nguyên được ghi lại là HTTP. Một ví dụ khác là "ftp://" để chỉ ra rằng giao thức được sử dụng là FTP.

Trong các hệ thống khác, ví dụ, các khối phương tiện có thể được xây dựng "khi đang hoạt động" bởi máy chủ để phản hồi yêu cầu của máy khách chỉ rõ phần trình diễn phương tiện, theo thời gian, được yêu cầu. Ví dụ, trong trường hợp của HTTP với sơ đồ "http://", việc thực hiện yêu cầu của URL này cung cấp phản hồi yêu cầu có chứa một số dữ liệu cụ thể trong phần thân thực thể của phản hồi yêu cầu này. Việc thực hiện trong mạng về cách tạo ra phản hồi yêu cầu này có thể rất khác nhau, tùy thuộc vào việc thực hiện của máy chủ phục vụ các yêu cầu này.

Thông thường, mỗi khối có thể giải mã được một cách độc lập. Ví dụ trong trường hợp phương tiện video, mỗi khối có thể bắt đầu với một "điểm tìm kiếm". Trong một số sơ đồ mã hóa, điểm tìm kiếm được gọi là "Điểm truy cập ngẫu nhiên - Random Access Points" hoặc "RAP", mặc dù không phải tất cả các RAP có thể được chỉ định là điểm tìm kiếm. Tương tự, trong các sơ đồ mã hóa khác, điểm tìm kiếm bắt đầu tại khung "Làm mới dữ liệu độc lập" hoặc "IDR", trong trường hợp kỹ thuật mã hóa video H.264, mặc dù không phải tất cả IDR có thể được chỉ định là điểm tìm kiếm. Điểm tìm kiếm là vị trí trong phương tiện video (hoặc phương tiện khác), tại đó bộ giải mã có thể bắt đầu giải mã mà không yêu cầu dữ liệu bất kỳ về các khung, dữ liệu, hoặc mẫu trước đó, điều này có thể đúng khi khung hoặc mẫu đang được giải mã đã được mã hóa không theo cách độc lập, mà, ví dụ, dưới dạng sự chênh lệch giữa khung hiện thời và khung trước đó.

Mối quan tâm trong các hệ thống như vậy có thể là khả năng bắt đầu phát ra dòng, ví dụ khả năng giải mã và kết xuất các dòng âm thanh và video thu được bằng cách sử dụng máy tính cá nhân và hiển thị video trên màn hình máy tính và phát âm thanh qua loa gắn sẵn, hoặc theo ví dụ khác, giải mã và kết xuất các dòng âm thanh và video thu được bằng cách sử dụng hộp chuyển đổi và hiển thị video trên thiết bị hiển thị truyền hình và phát âm thanh qua hệ thống âm thanh lập thể. Mối quan tâm chính có thể là vấn đề giảm thiểu độ trễ giữa thời điểm khi người dùng quyết định xem nội dung mới được phân phối dưới dạng dòng và thực hiện hành động thể hiện quyết định này, ví dụ như, người dùng nhấp vào liên kết trong cửa sổ trình duyệt hoặc vào nút phát (play) trên thiết bị điều khiển từ xa, và thời điểm khi nội dung bắt đầu được hiển thị trên màn hình của người dùng, sau đây gọi là "thời gian sửa đổi nội dung". Mỗi

mỗi quan tâm này có thể được giải quyết bằng các phần tử của hệ thống cải tiến được mô tả ở đây.

Một ví dụ về thời gian sửa đổi nội dung là khi người dùng đang xem nội dung thứ nhất được phân phối qua dòng thứ nhất và sau đó người dùng quyết định xem nội dung thứ hai được gửi qua dòng thứ hai và bắt đầu hành động để bắt đầu xem nội dung thứ hai. Dòng thứ hai có thể được gửi từ cùng một tập hợp hoặc tập hợp máy chủ khác dưới dạng dòng thứ nhất. Một ví dụ khác về thời gian sửa đổi nội dung là khi người dùng đang truy cập vào một trang web và quyết định bắt đầu xem nội dung thứ nhất được gửi qua dòng thứ nhất bằng cách nhấp vào liên kết trong cửa sổ trình duyệt. Tương tự, người dùng có thể quyết định bắt đầu phát nội dung không phải ngay từ đầu, mà thời điểm nào đó trong dòng. Người dùng chỉ rõ cho thiết bị máy khách tìm kiếm một vị trí thời gian và người dùng có thể mong đợi rằng thời gian được chọn được kết xuất ngay lập tức. Việc giảm thiểu thời gian sửa đổi nội dung là rất quan trọng trong khi xem video để cho phép người dùng trải nghiệm lướt nhanh nội dung với chất lượng cao khi tìm kiếm và lấy mẫu dải rộng gồm các nội dung có sẵn.

Gần đây, đã trở thành thực tế phổ biến để xem xét sử dụng mã FEC (Forward Error Correction - sửa lỗi chuyển tiếp) để bảo vệ phương tiện truyền liên tục trong quá trình truyền. Khi được gửi qua mạng gói, ví dụ bao gồm Internet và mạng không dây, chẳng hạn như, các mạng được chuẩn hóa bởi nhóm như 3GPP, 3GPP2 và DVB, dòng nguồn được đặt vào các gói dữ liệu khi nó được tạo ra hoặc được cung cấp sẵn, và do đó các gói này có thể được sử dụng để mang dòng nguồn hoặc dòng nội dung theo thứ tự được tạo ra hoặc cung cấp sẵn đến các bộ thu.

Trong ứng dụng điển hình của các mã FEC cho các loại kịch bản này, bộ mã hóa có thể sử dụng mã FEC để tạo ra các gói sửa chữa, mà sau đó được gửi đi, ngoài các gói nguồn ban đầu chứa các dòng nguồn. Các gói sửa chữa có đặc điểm là, khi việc mất gói nguồn xảy ra, các gói sửa chữa thu được có thể được sử dụng để khôi phục dữ liệu chứa trong các gói nguồn bị mất. Các gói sửa chữa có thể được sử dụng để khôi phục nội dung của gói nguồn bị mất mà bị mất hoàn toàn, nhưng cũng có thể được sử dụng để khôi phục khi xảy ra mất gói tin một phần, dù là các gói sửa chữa thu được toàn bộ hay thậm chí các gói sửa chữa thu được một phần. Vì vậy, các gói sửa chữa thu được toàn bộ hoặc một phần có thể được sử dụng để khôi phục các gói nguồn bị mất toàn bộ hoặc một phần.

Trong các ví dụ khác, các loại lỗi có thể xảy ra với dữ liệu gửi đi, ví dụ như giá trị của các bit có thể bị đảo lộn, và do đó các gói sửa chữa có thể được sử dụng để sửa chữa lỗi này và cung cấp khả năng khôi phục gói nguồn chính xác nhất có thể. Trong các ví dụ khác, dòng nguồn không nhất thiết phải được gửi trong các gói rời rạc, mà thay vào đó có thể được gửi, ví dụ, dưới dạng dòng bit liên tục.

Có rất nhiều ví dụ về mã FEC có thể được sử dụng để cung cấp khả năng bảo vệ cho dòng nguồn. Mã Reed - Solomon là loại mã được biết đến rộng rãi để sửa lỗi và sửa xóa trong các hệ thống truyền thông. Để sửa xóa trên, ví dụ, các mạng dữ liệu gói, ứng dụng hiệu quả được biết đến rộng rãi của mã Reed - Solomon sử dụng các ma trận Cauchy hoặc Vandermonde như được mô tả trong tài liệu của L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols", Computer Communication Review, 27(2):24-36 (tháng 4 năm 1997) (sau đây được gọi là "Rizzo") và Bloemer và các cộng sự, "An XOR-Based Erasure-Resilient Coding Scheme", Technical Report TR-95-48, International Computer Science Institute, Berkeley, California (1995) (sau đây được gọi là "XOR - Reed - Solomon") hoặc trong tài liệu khác.

Các ví dụ khác về mã FEC bao gồm mã LDPC, mã phản ứng chuỗi như được mô tả trong tài liệu của Luby I và các mã phản ứng chuỗi nhiều giai đoạn như trong tài liệu của Shokrollahi I.

Ví dụ về quy trình giải mã FEC cho các biến thể của mã Reed - Solomon được mô tả trong Rizzo và XOR - Reed - Solomon. Trong các ví dụ này, quy trình giải mã có thể được áp dụng sau khi đã thu được đủ gói nguồn và gói dữ liệu sửa chữa. Quy trình giải mã có thể đòi hỏi tính toán chuyên sâu và, tùy thuộc vào tài nguyên CPU có sẵn, hoạt động này có thể mất nhiều thời gian để hoàn thành, tương ứng với khoảng thời gian kéo dài của phương tiện trong khối. Bộ thu có thể tính đến khoảng thời gian cần thiết để giải mã khi tính toán độ trễ cần thiết giữa khi bắt đầu nhận các dòng phương tiện và khi phát ra phương tiện. Độ trễ do giải mã này được cảm nhận bởi người dùng là độ trễ giữa yêu cầu của họ đối với dòng phương tiện cụ thể và khi bắt đầu phát lại. Vì thế, mong muốn là giảm thiểu được độ trễ này.

Trong nhiều ứng dụng, các gói có thể được chia nhỏ hơn thành các ký hiệu mà quy trình FEC được áp dụng. Một gói có thể chứa một hoặc nhiều ký hiệu (hoặc ít hơn một ký hiệu, nhưng thường là các ký hiệu không được chia giữa các nhóm gói tin trừ

khi các điều kiện lỗi giữa các nhóm gói tin được biết là có mối tương quan cao). Ký hiệu có thể có kích thước bất kỳ, nhưng thường là kích thước của ký hiệu nhiều nhất là bằng với kích thước của gói tin. Ký hiệu nguồn là các ký hiệu mã hóa dữ liệu mà sẽ được truyền đi. Ký hiệu sửa chữa là các ký hiệu được tạo ra từ các ký hiệu nguồn, trực tiếp hoặc gián tiếp mà được thêm vào các ký hiệu nguồn (tức là, dữ liệu sẽ được truyền đi có thể được khôi phục hoàn toàn nếu tất cả các ký hiệu nguồn có sẵn và không có ký hiệu sửa chữa nào có sẵn).

Một số mã FEC có thể được dựa trên khối, ở chỗ hoạt động mã hóa phụ thuộc vào (các) ký hiệu đang ở trong một khối và có thể độc lập với ký hiệu không ở trong khối đó. Với việc mã hóa dựa trên khối, bộ mã hóa FEC có thể tạo ra các ký hiệu sửa chữa cho khối từ các ký hiệu nguồn trong khối đó, sau đó chuyển sang khối tiếp theo và không cần phải tham chiếu đến các ký hiệu nguồn ngoài các ký hiệu cho khối hiện thời đang được mã hóa. Trong khi truyền, khối nguồn bao gồm các ký hiệu nguồn có thể được biểu diễn bởi khối mã hóa bao gồm các ký hiệu mã hóa (mà có thể là một số ký hiệu nguồn, một số ký hiệu sửa chữa, hoặc cả hai). Với sự hiện diện của các ký hiệu sửa chữa, không phải tất cả các ký hiệu nguồn đều cần thiết trong mỗi khối mã hóa.

Đối với một số mã FEC, cụ thể mã Reed - mã Solomon, thời gian mã hóa và giải mã có thể tăng không thực tế khi số lượng ký hiệu mã hóa cho mỗi khối nguồn tăng. Vì vậy, trong thực tế, thường có cận trên (255 là giới hạn thực tế ước tính cho một số ứng dụng) trên tổng số ký hiệu mã hóa có thể được tạo ra cho mỗi khối nguồn, cụ thể là trong trường hợp điển hình khi quy trình mã hóa hoặc giải mã Reed - Solomon được thực hiện bởi phần cứng tùy chỉnh, ví dụ, quy trình MPE - FEC sử dụng mã Reed - Solomon được bao gồm như là một phần của tiêu chuẩn DVB- H để bảo vệ dòng không bị mất gói tin được thực hiện bằng phần cứng chuyên dụng trong điện thoại di động mà được giới hạn ở tổng số 255 ký hiệu mã hóa Reed - Solomon cho mỗi khối nguồn. Do các ký hiệu thường phải được đặt vào phần tải của gói tin riêng biệt, điều này đặt ra cận trên thực tế về độ dài tối đa của khối nguồn được mã hóa. Ví dụ, nếu phần tải của gói tin được giới hạn ở 1024 byte hoặc ít hơn và mỗi gói tin mang một ký hiệu mã hóa, thì khối nguồn mã hóa có thể có nhiều nhất là 255 kilobyte, và tất nhiên đây cũng là cận trên về kích thước của bản thân khối nguồn.

Các mối quan tâm khác, chẳng hạn như khả năng có thể giải mã các khối nguồn đủ nhanh để theo kịp với tốc độ truyền liên tục nguồn, để giảm thiểu độ trễ giải mã

đưa vào bởi quy trình giải mã FEC, và có thể chỉ sử dụng một phần nhỏ của CPU có sẵn trên thiết bị thu tại bất kỳ thời điểm nào trong quá trình giải mã FEC, được giải quyết nhờ các phần tử được mô tả ở đây.

Nhu cầu cung cấp giải pháp phân phối truyền trực tiếp mạnh mẽ và có khả năng mở rộng mà cho phép các thành phần của hệ thống gặp sự cố mà không ảnh hưởng xấu đến chất lượng của các dòng phân phối đến bộ thu.

Hệ thống truyền liên tục yêu cầu theo khối cần hỗ trợ các thay đổi về cấu trúc hoặc siêu dữ liệu của phần trình diễn, ví dụ như, thay đổi về số lượng phần mã hóa phương tiện có sẵn hoặc thay đổi về các tham số của phần mã hóa phương tiện như tốc độ bit, độ phân giải, tỷ lệ khung hình, các tham số thay đổi trong bộ mã hóa-giải mã âm thanh hoặc video trong siêu dữ liệu khác như các URL gắn với tập tin nội dung. Các thay đổi này có thể được yêu cầu vì một số lý do bao gồm chỉnh sửa cùng với nội dung từ các nguồn khác nhau, chẳng hạn như, quảng cáo hay các phân đoạn khác nhau của phần trình diễn lớn hơn, việc sửa đổi các URL hoặc các tham số khác trở nên cần thiết do các thay đổi trong cơ sở hạ tầng phục vụ, ví dụ, do thay đổi cấu hình, sự cố thiết bị hoặc khôi phục từ sự cố thiết bị hoặc các lý do khác.

Có nhiều phương pháp trong đó phần trình diễn có thể được điều khiển bởi tập tin danh sách được cập nhật liên tục. Do tập tin này được cập nhật liên tục, nên ít nhất một số trong các thay đổi mô tả ở trên có thể được thực hiện trong các bản cập nhật này. Nhược điểm của phương pháp thông thường là các thiết bị máy khách phải liên tục phải lấy ra, còn được gọi là "hỏi vòng", tập tin danh sách phát, đặt tải tin lên cơ sở hạ tầng phục vụ và tập tin này có thể không được lưu trữ tạm lâu hơn khoảng thời gian cập nhật, khiến cho nhiệm vụ của cơ sở hạ tầng phục vụ trở nên khó khăn hơn nhiều. Điều này được giải quyết nhờ các phần tử được mô tả ở đây sao cho các nội dung cập nhật mô tả ở trên được cung cấp mà không cần sự hỏi vòng liên tục của máy khách đối với tập tin siêu dữ liệu.

Một vấn đề khác, cụ thể là trong dịch vụ trực tiếp, thường được biết đến từ việc phân phối phát rộng, đó là thiếu khả năng để cho người dùng xem nội dung đã được phát rộng sớm hơn so với thời gian khi người dùng tham gia chương trình. Thông thường, việc ghi âm cá nhân cục bộ sử dụng lượng lưu trữ cục bộ không cần thiết hoặc không thể thực hiện do máy khách không được điều chỉnh sang chương trình này hoặc bị cấm bởi các quy tắc bảo vệ nội dung. Việc ghi mạng và xem có dịch thời gian được

ưu tiên, nhưng đòi hỏi kết nối riêng biệt của người dùng đến máy chủ và cơ sở hạ tầng riêng biệt và giao thức phân phối riêng biệt so với dịch vụ trực tiếp, dẫn đến trùng lặp cơ sở hạ tầng và chi phí máy chủ lớn. Điều này cũng được giải quyết nhờ các phần tử được mô tả ở đây.

Tổng quan hệ thống

Một phương án của sáng chế được mô tả dựa vào Fig.1, Fig.1 thể hiện sơ đồ giản lược của hệ thống truyền liên tục yêu cầu theo khối theo sáng chế.

Trên Fig.1, hệ thống truyền liên tục yêu cầu theo khối 100 được minh họa, bao gồm cơ sở hạ tầng phục vụ khối (BSI - block serving infrastructure) 101 có hệ thống thu thập 103 để thu thập nội dung 102, chuẩn bị nội dung và đóng gói nội dung để phục vụ bởi máy chủ truyền liên tục HTTP 104 bằng cách lưu trữ nội dung vào bộ phận lưu trữ nội dung 110 mà có thể truy cập được cho cả hai hệ thống thu thập 103 và máy chủ truyền liên tục HTTP 104. Như được thể hiện, hệ thống 100 cũng có thể bao gồm bộ nhớ đệm HTTP 106. Khi hoạt động, máy khách 108, chẳng hạn như, máy khách truyền liên tục HTTP, sẽ gửi yêu cầu 112 đến máy chủ truyền liên tục HTTP 104 và thu được các phản hồi 114 từ máy chủ truyền liên tục HTTP 104 hoặc bộ nhớ đệm HTTP 106. Trong mỗi trường hợp, các phần tử được thể hiện trên Fig.1 có thể được thực hiện, ít nhất là một phần bằng phần mềm, trong đó bao gồm mã chương trình được thực hiện trên bộ xử lý hoặc thiết bị điện tử khác.

Nội dung có thể bao gồm phim ảnh, âm thanh, video phẳng 2D, video 3D, các loại video khác, hình ảnh, văn bản được định thời, siêu dữ liệu được định thời hoặc tương tự. Một số nội dung có thể bao gồm dữ liệu được trình diễn hoặc được sử dụng theo cách được định thời, chẳng hạn như, dữ liệu để trình diễn thông tin phụ trợ (định danh trạm, quảng cáo, báo giá cổ phiếu, chuỗi Flash™, v.v.) cùng với phương tiện khác đang được phát ra. Các dạng trình diễn lai khác cũng có thể được sử dụng mà có thể kết hợp các phương tiện khác và/hoặc ngoài phương tiện chỉ đơn thuần là âm thanh và video.

Như được thể hiện trên Fig.2, các khối phương tiện có thể được lưu trữ trong cơ sở hạ tầng phục vụ khối 101(1), mà có thể là, ví dụ, máy chủ HTTP, thiết bị mạng phân phối nội dung, proxy HTTP, proxy hoặc máy chủ FTP, hoặc một số máy chủ phương tiện hoặc hệ thống khác. Cơ sở hạ tầng phục vụ khối 101(1) được kết nối với

mạng 122, mà có thể, ví dụ, là mạng giao thức Internet ("IP") như Internet. Máy khách hệ thống truyền liên tục yêu cầu theo khối được thể hiện có sáu thành phần chức năng, cụ thể là bộ chọn khối 123, được cung cấp cùng với siêu dữ liệu mô tả ở trên và thực hiện chức năng chọn các khối hoặc một phần khối được yêu cầu từ các khối có sẵn được chỉ báo bởi siêu dữ liệu, bộ yêu cầu khối 124, nhận lệnh yêu cầu từ bộ chọn khối 123 và thực hiện các hoạt động cần thiết để gửi yêu cầu cho khối được chỉ rõ, các phần của một khối hoặc nhiều khối, cơ sở hạ tầng phục vụ khối 101(1) qua mạng 122 và nhận dữ liệu bao gồm khối này, cũng như bộ đệm khối 125, bộ giám sát bộ đệm 126, bộ giải mã phương tiện 127 và một hoặc nhiều bộ chuyển đổi phương tiện 128 hỗ trợ sử dụng phương tiện.

Khối dữ liệu thu được bởi bộ yêu cầu khối 124 được chuyển qua để lưu trữ tạm thời đến bộ đệm khối 125 mà lưu trữ dữ liệu phương tiện. Theo cách khác, dữ liệu khối thu được có thể được lưu trữ trực tiếp vào bộ đệm khối 125 như được thể hiện trên Fig.1. Bộ giải mã phương tiện 127 được cung cấp với dữ liệu phương tiện bởi bộ đệm khối 125 và thực hiện chuyển đổi trên dữ liệu này nếu cần thiết để cung cấp dữ liệu nhập phù hợp cho bộ chuyển đổi phương tiện 128, bộ phận này kết xuất phương tiện dưới dạng thích hợp cho người dùng hoặc cho hoạt động sử dụng khác. Ví dụ về bộ chuyển đổi phương tiện bao gồm các thiết bị hiển thị hình ảnh, chẳng hạn như, các bộ phận được tìm thấy trong điện thoại di động, hệ thống máy tính hoặc TV, và cũng có thể bao gồm các thiết bị phát âm thanh, chẳng hạn như loa hoặc tai nghe.

Một ví dụ về bộ giải mã phương tiện là chức năng biến đổi dữ liệu ở các định dạng được mô tả trong chuẩn mã hóa video H.264 thành dạng biểu diễn tương tự hoặc kỹ thuật số của các khung video, chẳng hạn như ánh xạ điểm ảnh định dạng YUV với đầu thời gian trình diễn gắn kèm cho mỗi khung hoặc mẫu.

Bộ giám sát bộ đệm 126 thu thông tin liên quan đến nội dung của bộ đệm khối 125 và dựa trên thông tin này và có thể là thông tin khác nữa, cung cấp dữ liệu nhập đến bộ chọn khối 123, được sử dụng để xác định việc chọn khối để yêu cầu, như được mô tả ở đây.

Về thuật ngữ được sử dụng ở đây, mỗi khối có thời gian "phát ra" hoặc "thời lượng" biểu diễn thời lượng cần cho bộ thu để phát phương tiện được bao gồm trong khối ở tốc độ bình thường. Trong một số trường hợp, việc phát ra phương tiện trong khối có thể phụ thuộc vào việc đã thu được dữ liệu từ khối trước đó. Trong các trường

hợp hiếm gặp, việc phát ra một số phương tiện trong khối có thể phụ thuộc vào khối tiếp theo, trong trường hợp đó, thời gian phát ra cho khối được xác định đối với phương tiện mà có thể được phát ra trong khối mà không tham chiếu đến khối tiếp theo, và thời gian phát ra cho khối tiếp theo được tăng lên cộng thêm thời gian phát ra phương tiện trong khối mà chỉ có thể phát ra sau khi đã thu được khối tiếp theo. Do việc bao gồm phương tiện trong một khối mà phụ thuộc vào khối tiếp theo là trường hợp hiếm gặp, trong phần còn lại của sáng chế, giả sử rằng phương tiện trong một khối không phụ thuộc vào các khối tiếp theo, nhưng lưu ý rằng người có hiểu biết trung bình trong lĩnh vực kỹ thuật hiểu rằng các biến thể có thể được dễ dàng thêm vào các phương án được mô tả dưới đây.

Bộ thu có thể có các phần điều khiển như "tạm dừng", "tua nhanh", "tua ngược", v.v. mà có thể dẫn đến việc khối được sử dụng để phát ra với tốc độ khác nhau, nhưng nếu bộ thu có thể thu được và giải mã từng chuỗi khối liên tiếp trong thời gian tổng bằng hoặc ít hơn thời gian phát ra tổng của chúng không bao gồm khối cuối cùng trong chuỗi thì bộ thu có thể trình diễn phương tiện đến người dùng mà không phải trì hoãn. Trong một số phần mô tả ở đây, một vị trí cụ thể trong dòng phương tiện được gọi là "thời điểm" cụ thể trong phương tiện, tương ứng với thời gian mà có thể đã trôi qua giữa khi bắt đầu phát ra phương tiện và thời gian khi đạt được vị trí cụ thể trong dòng video. Thời điểm hoặc vị trí trong dòng phương tiện là một khái niệm thông thường. Ví dụ, khi dòng video bao gồm 24 khung mỗi giây, khung thứ nhất có thể được cho là có vị trí hoặc thời gian $t = 0,0$ giây và khung thứ 241 có thể được cho là có vị trí hoặc thời gian $t = 10,0$ giây. Đương nhiên, trong dòng video dựa trên khung, vị trí hoặc thời điểm không cần phải liên tục, do mỗi bit trong dòng từ bit thứ nhất của khung thứ 241 đến ngay trước bit thứ nhất của khung thứ 242 có thể đều có cùng giá trị thời gian.

Sử dụng các thuật ngữ trên, hệ thống truyền liên tục yêu cầu theo khối (BRSS - block-request streaming system) bao gồm một hoặc nhiều máy khách mà có thể đưa ra yêu cầu đến một hoặc nhiều máy chủ nội dung (ví dụ, máy chủ HTTP, máy chủ FTP, v.v.). Hệ thống thu thập bao gồm một hoặc nhiều bộ xử lý thu thập, trong đó bộ xử lý thu thập nhận nội dung (theo thời gian thực hoặc không), xử lý nội dung để được sử dụng bởi BRSS và lưu trữ nội dung vào bộ phận lưu trữ có thể truy cập được bởi máy chủ nội dung, có thể cùng với siêu dữ liệu được tạo ra bởi bộ xử lý thu thập.

BRSS cũng có thể chứa các bộ đệm nội dung phối hợp với máy chủ nội dung. Máy chủ nội dung và bộ đệm nội dung có thể là máy chủ HTTP và bộ đệm HTTP thông thường mà nhận yêu cầu cho tập tin hoặc phân đoạn dưới dạng yêu cầu HTTP bao gồm địa chỉ URL, và cũng có thể bao gồm một dãy byte, để yêu cầu ít hơn toàn bộ tập tin hoặc phân đoạn được chỉ báo bởi URL. Máy khách có thể bao gồm máy khách HTTP thông thường mà yêu cầu các máy chủ HTTP và xử lý các phản hồi cho các yêu cầu này, trong đó máy khách HTTP được điều khiển bởi hệ thống máy khách mới định ra các yêu cầu, chuyển chúng cho máy khách HTTP, nhận phản hồi từ máy khách HTTP và xử lý chúng (hoặc lưu trữ, chuyển đổi, v.v.) để cung cấp chúng đến máy phát thông tin trình diễn để phát ra bởi thiết bị máy khách. Thông thường, hệ thống máy khách không biết trước phương tiện nào sẽ được cần đến (do nhu cầu có thể phụ thuộc vào dữ liệu người dùng nhập, thay đổi về dữ liệu người dùng nhập, v.v.), vì vậy đây được gọi là hệ thống "truyền liên tục" mà trong đó phương tiện được "sử dụng" ngay khi nó được nhận, hoặc không lâu sau đó. Kết quả là, độ trễ phản hồi và hạn chế băng thông có thể gây ra độ trễ trong phần trình diễn, chẳng hạn như, gây ra tạm dừng trong phần trình diễn khi dòng kịp đến nơi mà người dùng đang sử dụng phần trình diễn.

Để cung cấp phần trình diễn mà được nhận thấy là có chất lượng tốt, một số chi tiết có thể được thực hiện trong BRSS, ở đầu máy khách, ở đầu thu thập, hoặc cả hai. Trong một số trường hợp, các chi tiết được thực hiện khi xem xét, và để xử lý, giao diện máy khách - máy chủ tại mạng. Theo một số phương án, cả hai hệ thống máy khách và hệ thống thu thập nhận biết được việc cải tiến, trong khi đó theo các phương án khác, chỉ có một bên nhận thức được việc cải tiến. Trong trường hợp này, toàn bộ hệ thống được hưởng lợi từ việc cải tiến mặc dù một bên không nhận biết được, trong khi trong các trường hợp khác, chỉ có được lợi ích nếu cả hai bên đều nhận biết được nó, nhưng khi một bên không nhận biết được, thì nó vẫn hoạt động tốt không có sự cố.

Như được thể hiện trên Fig.3, hệ thống thu thập có thể được cài đặt dưới dạng kết hợp của các thành phần phần cứng và phần mềm, theo các phương án khác nhau. Hệ thống thu thập có thể bao gồm tập hợp lệnh có thể được thực hiện để làm cho hệ thống thực hiện một hoặc nhiều phương pháp bất kỳ được mô tả ở đây. Hệ thống có thể được thực hiện như một máy cụ thể dưới dạng máy tính. Hệ thống này có thể là máy tính máy chủ, máy tính cá nhân (PC - personal computer), hoặc bất kỳ hệ thống nào có khả năng thực hiện tập hợp lệnh (tuần tự hay cách khác) mà xác định hành

động cần được thực hiện bởi hệ thống đó. Hơn nữa, trong khi chỉ có hệ thống duy nhất được minh họa, thuật ngữ "hệ thống" cũng được hiểu là bao gồm tập hợp hệ thống bất kỳ thực hiện một cách riêng lẻ hoặc phối hợp một tập hợp (hoặc nhiều tập hợp) lệnh để thực hiện một hoặc nhiều phương pháp bất kỳ được mô tả ở đây.

Hệ thống thu thập có thể bao gồm bộ xử lý thu thập 302 (ví dụ, đơn vị xử lý trung tâm (CPU)), bộ nhớ 304 có thể lưu trữ mã chương trình trong quá trình thực hiện, và ổ đĩa 306, tất cả đều giao tiếp với nhau qua bus 300. Hệ thống này còn có thể bao gồm màn hình hiển thị video 308 (ví dụ, màn hình tinh thể lỏng (LCD) hoặc ống tia catốt (CRT)). Hệ thống này cũng có thể bao gồm thiết bị đầu vào dùng chữ-số 310 (ví dụ, bàn phím), và thiết bị giao diện mạng 312 để nhận nguồn nội dung và phân phối nội dung lưu trữ.

Ổ đĩa 306 có thể bao gồm vật ghi đọc được bằng máy trên đó có thể lưu trữ một hoặc nhiều tập hợp lệnh (ví dụ, phần mềm) thực hiện một hoặc nhiều phương pháp hoặc chức năng bất kỳ được mô tả ở đây. Các lệnh này cũng có thể nằm trong, hoàn toàn hoặc ít nhất một phần, bộ nhớ 304 và/hoặc trong bộ xử lý thu thập 302 trong khi thực hiện bởi hệ thống, với bộ nhớ 304 và bộ xử lý thu thập 302 cũng cấu thành vật ghi đọc được bằng máy.

Như được thể hiện trên Fig.4, hệ thống máy khách có thể được cài đặt dưới dạng kết hợp của các thành phần phần cứng và phần mềm, theo các phương án khác nhau. Hệ thống máy khách có thể bao gồm tập hợp lệnh có thể được thực hiện để làm cho hệ thống này thực hiện một hoặc nhiều phương pháp bất kỳ được mô tả ở đây. Hệ thống này có thể được cài đặt như một máy cụ thể dưới dạng máy tính. Hệ thống này có thể là máy chủ, máy tính cá nhân (PC), hoặc bất kỳ hệ thống nào có khả năng thực hiện tập hợp lệnh (tuần tự hay cách khác) mà xác định hành động cần được thực hiện bởi hệ thống đó. Hơn nữa, trong khi chỉ có hệ thống duy nhất được minh họa, thuật ngữ "hệ thống" cũng được hiểu là bao gồm tập hợp hệ thống bất kỳ thực hiện một cách riêng lẻ hoặc phối hợp một tập hợp (hoặc nhiều tập hợp) lệnh để thực hiện một hoặc nhiều phương pháp bất kỳ được mô tả ở đây.

Hệ thống máy khách có thể bao gồm bộ xử lý máy khách 402 (ví dụ, đơn vị xử lý trung tâm (CPU)), bộ nhớ 404 có thể lưu trữ mã chương trình trong quá trình thực hiện, và ổ đĩa 406, tất cả đều giao tiếp với nhau qua bus 400. Hệ thống này còn có thể bao gồm màn hình hiển thị video 408 (ví dụ, màn hình tinh thể lỏng (LCD) hoặc ống

tia catốt (CRT)). Hệ thống này cũng có thể bao gồm thiết bị đầu vào dùng chữ-số 410 (ví dụ, bàn phím), và thiết bị giao diện mạng 412 để gửi yêu cầu và nhận phản hồi.

Ổ đĩa 406 có thể bao gồm vật ghi đọc được bằng máy trên đó có thể lưu trữ một hoặc nhiều tập hợp lệnh (ví dụ, phần mềm) thực hiện một hoặc nhiều phương pháp hoặc chức năng bất kỳ được mô tả ở đây. Các lệnh này cũng có thể nằm trong, hoàn toàn hoặc ít nhất một phần, bộ nhớ 404 và/hoặc trong bộ xử lý máy khách 402 trong khi thực hiện bởi hệ thống, với bộ nhớ 404 và bộ xử lý máy khách 402 cũng cấu thành vật ghi đọc được bằng máy.

Sử dụng định dạng tập tin 3GPP

Định dạng tập tin 3GPP hoặc định dạng tập tin bất kỳ khác dựa trên định dạng tập tin phương tiện cơ sở ISO, chẳng hạn như, định dạng tập tin MP4 hoặc các định dạng tập tin 3GPP2, có thể được sử dụng như là định dạng bộ chứa cho quy trình truyền liên tục HTTP với các tính năng sau đây. Chỉ mục phân đoạn có thể được bao gồm trong mỗi phân đoạn để báo hiệu độ dịch thời gian và dãy byte, để máy khách có thể tải về các phần thích hợp của tập tin hay phân đoạn phương tiện theo yêu cầu. Định thời trình diễn chung của toàn bộ phần trình diễn phương tiện và định thời cục bộ trong mỗi tập tin 3GP hoặc phân đoạn phương tiện có thể được đồng chỉnh một cách chính xác. Rãnh trong một tập tin 3GP hoặc phân đoạn phương tiện có thể được đồng chỉnh một cách chính xác. Rãnh trên các dạng biểu diễn cũng có thể được đồng chỉnh bằng cách gán mỗi trong số chúng vào dòng thời gian chung sao cho việc chuyển qua dạng biểu diễn có thể liền mạch và việc trình diễn phối hợp các thành phần phương tiện trong các dạng biểu diễn khác nhau có thể là đồng bộ.

Định dạng tập tin có thể chứa profin cho quy trình truyền liên tục thích ứng với các thuộc tính sau. Tất cả dữ liệu phim có thể được chứa trong các đoạn phim - hộp "moov" có thể không chứa thông tin mẫu bất kỳ. Dữ liệu mẫu âm thanh và video có thể được đặt xen kẽ, với các yêu cầu tương tự như đối với profin tải về tuần tự như được chỉ rõ trong TS26.244. Hộp "moov" có thể được đặt ở đầu tập tin, tiếp theo là dữ liệu độ dịch đoạn, cũng được gọi là chỉ mục phân đoạn, chứa thông tin độ dịch theo thời gian và dãy byte cho mỗi đoạn hoặc ít nhất là tập con các đoạn trong phân đoạn chứa.

Phần mô tả trình diễn phương tiện có thể tham chiếu tập tin sau profin tải về tuần tự hiện có. Trong trường hợp này, máy khách có thể sử dụng phần mô tả trình diễn phương tiện đơn giản là để chọn phiên bản thay thế thích hợp từ trong số các phiên bản có sẵn. Máy khách cũng có thể sử dụng yêu cầu nhận một phần HTTP với các tập tin phù hợp với profin tải về tuần tự để yêu cầu các tập con của mỗi phiên bản thay thế và do đó thực hiện một dạng ít hiệu quả của truyền liên tục thích ứng. Trong trường hợp này, các dạng biểu diễn khác nhau chứa phương tiện trong profin tải về tuần tự vẫn có thể tuân theo dòng thời gian chung để cho phép chuyển đổi liên mạch giữa các dạng biểu diễn.

Tổng quan về các phương pháp tiên tiến

Trong các phần sau đây, các phương pháp để cải thiện hệ thống truyền liên tục yêu cầu theo khối sẽ được mô tả. Nên hiểu rằng một số cải tiến trong số các cải tiến này có thể được sử dụng khi có hoặc không có các cải tiến khác, tùy thuộc vào nhu cầu của ứng dụng. Trong hoạt động chung, bộ thu yêu cầu máy chủ hoặc bộ phát khác cung cấp các khối dữ liệu hoặc các phần của khối dữ liệu cụ thể. Tập tin, còn được gọi là phân đoạn, có thể chứa nhiều khối và được gắn với một dạng biểu diễn của phần trình diễn phương tiện.

Tốt hơn là, thông tin lập chỉ mục, còn được gọi là "lập chỉ mục phân đoạn" hay "ánh xạ phân đoạn", được tạo ra để cung cấp ánh xạ từ số lần phát ra hoặc số lần giải mã sang các độ dịch byte của các khối hoặc đoạn tương ứng trong một phân đoạn. Việc lập chỉ mục phân đoạn này có thể được bao gồm trong phân đoạn này, thường ở đầu phân đoạn (ít nhất là một số ánh xạ phân đoạn là ở đầu) và thường là nhỏ. Chỉ mục phân đoạn cũng có thể được cung cấp trong phân đoạn hoặc tập tin chỉ mục riêng biệt. Cụ thể là trong trường hợp mà chỉ mục phân đoạn được chứa trong phân đoạn này, bộ thu có thể tải về một số hoặc tất cả các ánh xạ phân đoạn này một cách nhanh chóng và sau đó sử dụng nó để xác định ánh xạ giữa độ dịch thời gian và vị trí byte tương ứng của các đoạn gắn với các độ dịch thời gian này trong tập tin.

Bộ thu có thể sử dụng độ dịch byte để yêu cầu dữ liệu từ các đoạn gắn với độ dịch thời gian cụ thể, mà không cần phải tải về tất cả dữ liệu gắn với các đoạn khác không gắn với độ dịch thời gian đang quan tâm. Bằng cách này, ánh xạ phân đoạn hoặc lập chỉ mục phân đoạn có thể cải thiện đáng kể khả năng mà bộ thu có thể trực

tiếp truy cập vào các phần của phân đoạn có liên quan đến các độ dịch thời gian hiện thời đang quan tâm, với các lợi ích như: cải thiện thời gian sửa đổi nội dung, khả năng nhanh chóng thay đổi từ một dạng biểu diễn sang dạng biểu diễn khác khi điều kiện mạng thay đổi, và giảm lãng phí tài nguyên mạng tải về phương tiện mà không được phát ra tại bộ thu.

Trong trường hợp chuyển từ một dạng biểu diễn (dưới đây gọi là dạng biểu diễn "chuyển đi") sang một dạng biểu diễn khác (dưới đây gọi là dạng biểu diễn "chuyển đến") được xem xét, chỉ mục phân đoạn cũng có thể được sử dụng để xác định thời gian bắt đầu của điểm truy cập ngẫu nhiên trong dạng biểu diễn chuyển đến để xác định lượng dữ liệu được yêu cầu trong dạng biểu diễn chuyển đi để đảm bảo rằng chuyển đổi liên mạch được kích hoạt theo nghĩa là phương tiện ở dạng biểu diễn chuyển đi được tải về tới đa đến thời điểm trình diễn sao cho việc phát ra dạng biểu diễn chuyển đến có thể bắt đầu liên tục từ điểm truy cập ngẫu nhiên đó.

Các khối biểu diễn phân đoạn của phương tiện video hoặc phương tiện khác mà bộ thu yêu cầu cần để tạo ra đầu ra cho người dùng của bộ thu. Bộ thu của phương tiện có thể là thiết bị máy khách, chẳng hạn như, khi bộ thu nhận nội dung từ máy chủ truyền nội dung. Ví dụ bao gồm hộp chuyển đổi, máy tính, bàn điều khiển trò chơi, TV được trang bị chuyên dụng, thiết bị cầm tay, điện thoại di động được trang bị chuyên dụng hoặc bộ thu máy khách khác.

Nhiều phương pháp quản lý bộ đệm tiên tiến được mô tả ở đây. Ví dụ, phương pháp quản lý bộ đệm cho phép máy khách yêu cầu các khối có chất lượng phương tiện cao nhất có thể được nhận đúng thời điểm để được phát ra liên tục. Tính năng kích thước khối thay đổi cải thiện hiệu suất nén. Khả năng có nhiều kết nối để truyền các khối đến thiết bị yêu cầu trong khi giới hạn tần số của các yêu cầu cung cấp khả năng cải thiện hiệu suất truyền. Khối dữ liệu thu được một phần có thể được sử dụng để tiếp tục trình diễn phương tiện. Kết nối có thể được tái sử dụng cho nhiều khối mà không cần phải chuyển giao kết nối khi bắt đầu đối với tập hợp khối cụ thể. Tính nhất quán trong việc chọn máy chủ từ nhiều máy chủ có thể có bởi nhiều máy khách được cải thiện, điều này làm giảm tần suất của nội dung trùng lặp trong các máy chủ gần đó và cải thiện khả năng mà máy chủ chứa toàn bộ tập tin. Máy khách có thể yêu cầu các khối phương tiện dựa trên siêu dữ liệu (chẳng hạn như các phần mã hóa phương tiện có sẵn) được nhúng trong các URL cho tập tin có chứa các khối phương tiện. Hệ thống

có thể cung cấp việc tính toán và giảm thiểu thời lượng đệm cần thiết trước khi việc phát ra nội dung có thể bắt đầu mà không gây ra việc tạm dừng tiếp theo trong khi phát ra phương tiện. Bảng thông có sẵn có thể được chia sẻ giữa nhiều khối phương tiện, được điều chỉnh khi thời điểm phát ra của mỗi khối đến gần, do đó, nếu cần thiết, phần lớn hơn của bảng thông có sẵn có thể được phân bổ cho khối với thời điểm phát ra gần nhất.

Quy trình truyền liên tục HTTP có thể sử dụng siêu dữ liệu. Siêu dữ liệu mức trình diễn bao gồm, ví dụ, thời lượng dòng, các phần mã hóa có sẵn (tốc độ bit, các phần mã hóa-giải mã, độ phân giải không gian, tốc độ khung, ngôn ngữ, các loại phương tiện), con trỏ đến siêu dữ liệu dòng cho mỗi phần mã hóa và thông tin bảo vệ nội dung (thông tin quản lý quyền kỹ thuật số (DRM - digital rights management)). Siêu dữ liệu dòng có thể là URL cho tập tin phân đoạn.

Siêu dữ liệu phân đoạn có thể bao gồm dãy byte so với thông tin thời gian cho các yêu cầu trong một phân đoạn và xác định điểm truy cập ngẫu nhiên (RAP) hoặc điểm tìm kiếm khác, trong đó một số hoặc tất cả thông tin này có thể là một phần của việc lập chỉ mục phân đoạn hoặc ánh xạ phân đoạn.

Các dòng có thể bao gồm nhiều phần mã hóa có cùng nội dung. Khi đó, mỗi phần mã hóa có thể được chia thành các phân đoạn trong đó mỗi phân đoạn tương ứng với một đơn vị lưu trữ hoặc tập tin. Trong trường hợp HTTP, phân đoạn thường là tài nguyên có thể được tham chiếu đến bởi URL và yêu cầu URL như vậy dẫn đến việc trả về phân đoạn là phần thân thực thể của thông báo phản hồi yêu cầu. Phân đoạn có thể bao gồm nhiều nhóm hình ảnh (GOP - group of pictures). Mỗi GOP có thể còn bao gồm nhiều đoạn trong đó việc lập chỉ mục phân đoạn cung cấp thông tin thời gian/độ dịch byte cho từng đoạn, tức là, đơn vị lập chỉ mục là đoạn.

Các đoạn hoặc các phần của đoạn có thể được yêu cầu qua các kết nối TCP song song để tăng thông lượng. Điều này có thể giảm thiểu các vấn đề phát sinh khi chia sẻ kết nối trên liên kết nút cổ chai hoặc khi kết nối bị mất do tắc nghẽn, do đó làm tăng tốc độ tổng thể và độ tin cậy phân phối, mà có thể cải thiện đáng kể tốc độ và độ tin cậy của thời gian sửa đổi nội dung. Bảng thông có thể được trao đổi với độ trễ bằng cách yêu cầu cao hơn, nhưng cần thận trọng để tránh việc yêu cầu quá xa trong tương lai có thể làm tăng nguy cơ thiếu hụt trầm trọng.

Nhiều yêu cầu cho phân đoạn trên cùng máy chủ có thể được tạo đường ống (thực hiện yêu cầu tiếp theo trước khi yêu cầu hiện thời hoàn thành) để tránh trễ khởi động TCP lặp đi lặp lại. Yêu cầu các đoạn liên tiếp có thể được tổng hợp thành một yêu cầu.

Một số CDN ưu tiên tập tin lớn và có thể kích hoạt việc tìm nạp phụ đối toàn bộ tập tin từ máy chủ gốc khi lần thứ đầu thấy yêu cầu dài. Tuy nhiên, hầu hết CDN sẽ phục vụ yêu cầu dài từ bộ nhớ đệm nếu dữ liệu có sẵn. Do đó, có thể có lợi nếu có phần nào đó của máy khách yêu cầu toàn bộ tập tin của phân đoạn. Sau đó, các yêu cầu này có thể được hủy bỏ nếu cần thiết.

Điểm chuyển hợp lệ có thể là điểm tìm kiếm, cụ thể là RAP, trong dòng đích. Có thể có các cài đặt khác nhau, chẳng hạn như, cấu trúc GOP cố định hoặc sự đồng chỉnh các RAP trên các dòng (dựa trên chỗ bắt đầu của phương tiện hoặc dựa trên GOP).

Theo một phương án, các phân đoạn và các GOP có thể được đồng chỉnh trên các dòng tốc độ khác nhau. Theo phương án này, GOP có thể có kích cỡ thay đổi và có thể chứa nhiều đoạn, nhưng các đoạn không cần đồng chỉnh giữa các dòng tốc độ khác nhau.

Theo một số phương án, sự dư thừa tập tin có thể được sử dụng để mang lại lợi ích. Trong các phương án này, mã xóa được áp dụng cho từng phân đoạn để tạo ra các phiên bản dư của dữ liệu. Tốt hơn là, định dạng nguồn không được thay đổi do sử dụng FEC, và phân đoạn sửa chữa bổ sung, ví dụ như dạng biểu diễn phụ thuộc của dạng biểu diễn ban đầu, chứa dữ liệu sửa chữa FEC được tạo ra và được dành sẵn như là bước bổ sung trong hệ thống thu thập. Máy khách, mà có khả năng tái tạo đoạn chỉ sử dụng dữ liệu nguồn cho đoạn đó, có thể chỉ yêu cầu dữ liệu nguồn cho đoạn này trong phân đoạn từ máy chủ. Nếu máy chủ không có sẵn hoặc kết nối đến máy chủ chậm, mà có thể được xác định trước hoặc sau khi yêu cầu dữ liệu nguồn, dữ liệu sửa chữa bổ sung có thể được yêu cầu cho đoạn này từ đoạn sửa chữa, điều này làm giảm thời gian phân phối một cách đáng tin cậy đủ dữ liệu để khôi phục đoạn, có thể sử dụng kỹ thuật giải mã FEC để sử dụng sự kết hợp của nguồn thu được và dữ liệu sửa chữa để khôi phục dữ liệu nguồn của đoạn. Hơn nữa, dữ liệu sửa chữa bổ sung có thể được yêu cầu để cho phép khôi phục đoạn nếu cần đoạn này cấp bách, tức là, thời gian phát ra của đoạn sắp xảy ra, điều này làm tăng khả năng chia sẻ dữ liệu cho phân đoạn

đó trên liên kết nhưng hiệu quả hơn so với việc đóng các kết nối khác trên liên kết này để giải phóng băng thông. Điều này cũng có thể giảm thiểu nguy cơ thiếu hụt trầm trọng do sử dụng các kết nối song song.

Định dạng đoạn có thể là dòng được lưu trữ gồm các gói RTP (giao thức truyền tải thời gian thực) có đồng bộ hóa âm thanh/video thu được qua RTCP (giao thức điều khiển truyền tải thời gian thực).

Định dạng phân đoạn cũng có thể là dòng được lưu trữ gồm các gói MPEG - 2 TS có đồng bộ hóa âm thanh/video thu được qua định thời khoảng MPEG - 2 TS.

Sử dụng báo hiệu và/hoặc tạo khối để truyền liên tục hiệu quả hơn

Một số tính năng có thể được sử dụng hoặc không được sử dụng, trong hệ thống truyền liên tục yêu cầu theo khối, để mang lại hiệu năng được cải thiện. Hiệu năng có thể liên quan đến khả năng phát ra phần trình diễn không cần trì hoãn, thu được dữ liệu phương tiện trong các hạn chế băng thông, và/hoặc làm như vậy trong phạm vi tài nguyên bộ xử lý bị hạn chế tại máy khách, máy chủ và/hoặc hệ thống thu thập. Một số tính năng này sẽ được mô tả dưới đây.

Lập chỉ mục trong các phân đoạn

Để đưa ra yêu cầu GET một phần cho các đoạn phim, máy khách có thể được thông báo về độ dịch byte và thời gian bắt đầu giải mã hoặc thời gian trình diễn của tất cả các thành phần phương tiện được chứa trong các đoạn trong tập tin hoặc phân đoạn và cũng như các đoạn nào bắt đầu hoặc chứa điểm truy cập ngẫu nhiên (và do đó phù hợp để sử dụng làm điểm chuyển giữa các dạng biểu diễn thay thế), trong đó thông tin này thường được gọi là lập chỉ mục phân đoạn hoặc ánh xạ phân đoạn. Thời gian bắt đầu giải mã hoặc thời gian trình diễn có thể được thể hiện trực tiếp hoặc có thể được thể hiện là delta so với thời gian tham chiếu.

Thời gian này và thông tin chỉ mục độ dịch byte có thể yêu cầu ít nhất là 8 byte dữ liệu cho mỗi đoạn phim. Ví dụ, đối với bộ phim hai giờ được chứa trong một tập tin duy nhất, với các đoạn phim 500ms, đây sẽ là tổng của khoảng 112 kilobyte dữ liệu. Việc tải về tất cả dữ liệu này khi bắt đầu phần trình diễn có thể tạo ra độ trễ khởi động bổ sung đáng kể. Tuy nhiên, dữ liệu thời gian và độ dịch byte có thể được mã hóa phân cấp, do đó máy khách có thể nhanh chóng tìm thấy một khối nhỏ dữ liệu thời gian và

độ dịch liên quan đến điểm trong phần trình diễn mà tại đó nó mong muốn để bắt đầu. Thông tin này cũng có thể được phân phối trong phân đoạn sao cho phần tinh chỉnh nào đó của chỉ mục phân đoạn có thể nằm xen kẽ với dữ liệu phương tiện.

Lưu ý rằng nếu dạng biểu diễn được phân đoạn theo thời gian thành nhiều phân đoạn, việc sử dụng mã hóa thứ bậc này có thể không cần thiết, do dữ liệu thời gian hoàn thành và độ dịch cho từng phân đoạn có thể đã khá nhỏ. Ví dụ, nếu phân đoạn là một phút thay vì hai giờ trong ví dụ trên, thông tin thời gian - chỉ mục độ dịch byte khoảng 1 kilobyte dữ liệu, mà thường có thể vừa khít trong một gói tin TCP/IP.

Có thể có các tùy chọn khác nhau để thêm dữ liệu thời gian và độ dịch byte của đoạn vào tập tin 3GPP:

Thứ nhất, hộp truy cập ngẫu nhiên đoạn phim (MFRA - Movie Fragment Random Access Box) có thể được sử dụng cho mục đích này. MFRA cung cấp một bảng, mà có thể hỗ trợ người đọc tìm kiếm các điểm truy cập ngẫu nhiên trong tập tin bằng cách sử dụng các đoạn phim. Để hỗ trợ chức năng này, MFRA ngẫu nhiên chứa độ dịch byte của các hộp MFRA chứa các điểm truy cập ngẫu nhiên. MFRA có thể được đặt ở cuối hoặc gần cuối của tập tin, nhưng không nhất thiết phải như vậy. Bằng cách quét từ cuối tập tin cho hộp độ dịch truy cập ngẫu nhiên đoạn phim và sử dụng thông tin kích thước trong đó, có thể xác định chỗ bắt đầu của MFRA. Tuy nhiên, việc đặt MFRA vào cuối cho dòng HTTP thường đòi hỏi ít nhất 3 - 4 yêu cầu HTTP để truy cập dữ liệu mong muốn: ít nhất một yêu cầu để yêu cầu MFRA từ cuối tập tin, một yêu cầu để thu được MFRA và yêu cầu cuối cùng để thu được đoạn mong muốn trong tập tin. Vì vậy, việc đặt ở đầu có thể được mong muốn do sau đó MFRA có thể được tải về cùng với dữ liệu phương tiện đầu tiên trong một yêu cầu duy nhất. Ngoài ra, việc sử dụng MFRA để truyền liên tục HTTP có thể không hiệu quả, vì không có thông tin trong "MFRA" cần thiết ngoài thời gian và moof_offset và việc xác định độ dịch thay vì các độ dài có thể đòi hỏi nhiều bit hơn.

Thứ hai, hộp vị trí mục ("ILOC") có thể được sử dụng. "ILOC" cung cấp danh mục các tài nguyên siêu dữ liệu trong đó hoặc tập tin khác, bằng cách định vị tập tin chứa của chúng, độ dịch trong tập tin đó, và chiều dài của chúng. Ví dụ, hệ thống có thể tích hợp tất cả các tài nguyên siêu dữ liệu tham chiếu từ bên ngoài vào một tập tin, điều chỉnh lại độ dịch tập tin và thông tin tham chiếu tập tin phù hợp. Tuy nhiên,

"ILOC" dự định cung cấp vị trí của siêu dữ liệu do đó nó có thể khó mà cùng tồn tại với siêu dữ liệu thực tế.

Cuối cùng, và có lẽ thích hợp nhất, là đặc điểm kỹ thuật của hộp mới, được gọi là hộp chỉ mục thời gian ("TIDX"), đặc biệt dành riêng cho mục đích cung cấp thời gian hoặc thời lượng chính xác của đoạn và độ dịch byte một cách hiệu quả. Điều này được mô tả chi tiết hơn trong phần tiếp theo. Hộp thay thế có cùng chức năng có thể là hộp chỉ mục phân đoạn ("SIDX"). Ở đây, trừ khi được nêu khác, hai hộp này có thể hoán đổi cho nhau, vì cả hai hộp cung cấp khả năng cung cấp thời gian hoặc thời lượng chính xác của đoạn và độ dịch byte một cách hiệu quả. Sự khác biệt giữa TIDX và SIDX được cung cấp dưới đây. Cách để trao đổi các hộp TIDX và hộp SIDX là rõ ràng, vì cả hai hộp đều cài đặt chỉ mục phân đoạn.

Lập chỉ mục phân đoạn

Phân đoạn có thời gian bắt đầu xác định một số lượng byte xác định. Nhiều đoạn có thể được nối thành phân đoạn duy nhất và máy khách có thể đưa ra yêu cầu xác định dãy byte cụ thể trong phân đoạn tương ứng với đoạn hoặc tập con đoạn được yêu cầu. Ví dụ, khi HTTP được sử dụng như là giao thức yêu cầu, thì phần tiêu đề dài HTTP có thể được sử dụng cho mục đích này. Phương pháp này đòi hỏi máy khách có quyền truy cập vào "chỉ mục phân đoạn" của phân đoạn mà chỉ rõ vị trí trong phân đoạn của các đoạn khác nhau. "Chỉ mục phân đoạn" này có thể được cung cấp như một phần của siêu dữ liệu. Phương pháp này có kết quả là rất ít tập tin cần được tạo ra và được quản lý so với các phương pháp mà trong đó mỗi khối được giữ trong một tập tin riêng biệt. Việc quản lý sự tạo ra, phân phối và lưu trữ số lượng rất lớn tập tin (mà có thể mở rộng đến hàng ngàn cho phần trình diễn trong 1 giờ) có thể rất phức tạp và dễ bị lỗi và do đó việc giảm số lượng tập tin là một lợi thế.

Nếu máy khách chỉ biết thời gian bắt đầu mong muốn của một phần nhỏ của phân đoạn, nó có thể yêu cầu toàn bộ tập tin, sau đó đọc tập tin này để xác định vị trí bắt đầu phát ra thích hợp. Để cải thiện việc sử dụng băng thông, phân đoạn có thể bao gồm tập tin chỉ mục như là siêu dữ liệu, trong đó tập tin chỉ mục này ánh xạ các dãy byte của các khối riêng biệt riêng biệt với các dải thời gian mà các khối tương ứng với, được gọi là lập chỉ mục phân đoạn hoặc ánh xạ phân đoạn. Siêu dữ liệu này có thể được định dạng như là dữ liệu XML hoặc có thể là giá trị nhị phân, ví dụ sau đây là

cấu trúc nguyên tử và cấu trúc hộp của định dạng tập tin 3GPP. Việc lập chỉ mục có thể đơn giản, trong đó thời gian và dãy byte của mỗi khối tuyệt đối so với phần đầu của tập tin, hoặc chúng có thể được phân cấp, trong đó một số khối được nhóm lại thành các khối cha (và thành khối ông bà, v.v.) và thời gian và dãy byte cho một khối cho trước được thể hiện tương ứng với thời gian và/hoặc dãy byte của khối cha của khối này.

Ví dụ về cấu trúc ánh xạ chỉ mục

Theo một phương án, dữ liệu nguồn ban đầu cho một dạng biểu diễn của dòng phương tiện có thể được chứa trong một hoặc nhiều tập tin phương tiện ở đây được gọi là "phân đoạn phương tiện", trong đó mỗi phân đoạn phương tiện chứa dữ liệu phương tiện được sử dụng để phát lại phân đoạn thời gian liên tục của phương tiện, ví dụ như, 5 phút phát lại phương tiện.

Fig.6 thể hiện cấu trúc tổng thể làm ví dụ của phân đoạn phương tiện. Trong mỗi phân đoạn, ở đầu hoặc trải rộng khắp phân đoạn nguồn, cũng có thể có thông tin lập chỉ mục, mà bao gồm ánh xạ phân đoạn thời gian/độ dịch byte. Theo một phương án, ánh xạ phân đoạn thời gian/độ dịch byte có thể là danh sách các cặp thời gian/độ dịch byte $(T(0), B(0)), (T(1), B(1)), \dots, (T(i), B(i)), \dots, (T(n), B(n))$, trong đó $T(i-1)$ biểu diễn thời gian bắt đầu trong phân đoạn để phát lại của đoạn thứ i của phương tiện so với thời gian bắt đầu ban đầu của phương tiện trong số tất cả các phân đoạn phương tiện, $T(i)$ biểu diễn thời gian kết thúc cho đoạn thứ i (và do đó là thời gian bắt đầu cho đoạn tiếp theo), và độ dịch byte $B(i-1)$ là chỉ mục byte tương ứng của phần đầu dữ liệu trong phân đoạn nguồn mà tại đó đoạn thứ i của phương tiện bắt đầu so với phần đầu của phân đoạn nguồn, và $B(i)$ là chỉ mục byte cuối tương ứng đoạn thứ i (và do đó là chỉ mục của byte thứ nhất của đoạn tiếp theo). Nếu phân đoạn chứa nhiều thành phần phương tiện, thì $T(i)$ và $B(i)$ có thể được cung cấp cho mỗi thành phần trong phân đoạn theo cách tuyệt đối hoặc có thể được thể hiện tương ứng với thành phần phương tiện khác phục vụ thành phần phương tiện tham chiếu.

Theo phương án này, số lượng các đoạn trong phân đoạn nguồn là n , trong đó n có thể khác nhau giữa các phân đoạn.

Theo phương án khác, độ dịch thời gian trong chỉ mục phân đoạn cho từng phân đoạn có thể được xác định bằng thời gian bắt đầu tuyệt đối của đoạn thứ nhất và

thời lượng của từng phân đoạn. Trong trường hợp này, chỉ mục phân đoạn có thể ghi lại thời gian bắt đầu của đoạn thứ nhất và thời lượng của tất cả các đoạn được bao gồm trong phân đoạn này. Chỉ mục phân đoạn có thể cũng có thể chỉ ghi lại tập con các đoạn. Trong trường hợp đó, chỉ mục phân đoạn ghi lại thời lượng của phân đoạn nhỏ mà được xác định là một hoặc nhiều đoạn liên tiếp, kết thúc hoặc ở cuối của phân đoạn chứa, hoặc ở đầu của phân đoạn nhỏ tiếp theo.

Đối với mỗi đoạn, có thể cũng có giá trị chỉ báo việc đoạn có bắt đầu tại hoặc có chứa điểm tìm kiếm hay không, tức là, tại điểm trong đó không có phương tiện sau điểm đó phụ thuộc vào phương tiện bất kỳ trước điểm đó, và do đó phương tiện từ đoạn đó trở đi có thể được phát ra độc lập với các đoạn trước đó. Nói chung, điểm tìm kiếm là điểm trong phương tiện mà việc phát ra có thể bắt đầu một cách độc lập với tất cả phương tiện trước đó. Fig.6 cũng thể hiện một ví dụ đơn giản về việc lập chỉ mục phân đoạn có thể có cho phân đoạn nguồn. Trong ví dụ này, giá trị độ dịch thời gian tính theo đơn vị mili giây, và do đó đoạn thứ nhất của phân đoạn nguồn này bắt đầu 20 giây tính từ phần đầu của phương tiện, và đoạn thứ nhất có thời gian phát ra là 485 mili giây. Độ dịch byte của phần đầu của đoạn thứ nhất là 0, và độ dịch byte của phần cuối của đoạn thứ nhất/phần đầu của đoạn thứ hai là 50.245, và do đó đoạn thứ nhất có kích thước 50,245 byte. Nếu đoạn hoặc phân đoạn nhỏ không bắt đầu với điểm truy cập ngẫu nhiên, nhưng điểm truy cập ngẫu nhiên được chứa trong đoạn hoặc phân đoạn nhỏ, thì thời gian giải mã hoặc hiệu số thời gian trình diễn giữa thời gian bắt đầu và thời gian RAP thực tế có thể được cho trước. Điều này cho phép rằng trong trường hợp chuyển sang phân đoạn phương tiện này, máy khách có thể biết chính xác thời gian cho đến khi việc chuyển từ dạng biểu diễn cần được trình diễn.

Ngoài ra, hoặc theo cách khác, các kỹ thuật lập chỉ mục đơn giản hoặc phân cấp, lập chỉ mục tuần tự và/hoặc lập chỉ mục lai có thể được sử dụng.

Bởi vì thời lượng mẫu cho rãnh khác nhau có thể không giống nhau (ví dụ, mẫu video có thể được hiển thị trong 33ms, trong khi mẫu âm thanh có thể kéo dài 80ms), rãnh khác nhau trong đoạn phim có thể không bắt đầu và kết thúc chính xác cùng lúc, nghĩa là âm thanh có thể bắt đầu trước một chút hoặc sau một chút so với video, với điều ngược lại đúng với đoạn trước đó, để bù lại. Để tránh nhầm lẫn, các dấu thời gian được chỉ rõ trong dữ liệu thời gian và độ dịch byte có thể được xác định so với một rãnh cụ thể và có thể là cùng rãnh cho từng dạng biểu diễn. Thông thường đây là rãnh

video. Điều này cho phép máy khách xác định chính xác khung video tiếp theo khi nó chuyển đổi các dạng biểu diễn.

Cần phải chú ý trong quá trình trình diễn để duy trì mối liên hệ chặt chẽ giữa các thang thời gian của rãnh và thời gian trình diễn, để đảm bảo việc phát ra trơn tru và duy trì đồng bộ hóa âm thanh/video mặc dù có các vấn đề trên.

Fig.7 minh họa một số ví dụ, chẳng hạn như, chỉ mục đơn giản 700 và chỉ mục phân cấp 702.

Hai ví dụ cụ thể về hộp chứa ánh xạ phân đoạn được cung cấp dưới đây, một hộp được gọi là hộp chỉ mục thời gian ('TIDX') và một hộp được gọi là ('SIDX'). Định nghĩa này tuân theo cấu trúc hộp theo định dạng tập tin phương tiện cơ sở ISO. Các thiết kế khác cho các hộp này để xác định cú pháp tương tự và với cùng ngữ nghĩa và chức năng nên rõ ràng với người đọc.

Hộp chỉ mục thời gian

Định nghĩa

Loại hộp: 'tidx'

Bộ chứa: Tập tin

Bắt buộc: Không có

Số lượng: Số lượng bất kỳ số không hoặc một

Hộp chỉ mục thời gian có thể cung cấp tập hợp thời gian và các chỉ mục độ dịch byte gắn với các vùng nhất định của tập tin với khoảng thời gian nhất định của phần trình diễn. Hộp chỉ mục thời gian có thể bao gồm trường TargetType (loại đích), mà chỉ ra kiểu dữ liệu tham chiếu. Ví dụ, hộp chỉ mục thời gian với TargetType "moof" cung cấp chỉ mục đến các đoạn phương tiện chứa trong tập tin cả về thời gian lẫn độ dịch byte. Hộp chỉ mục thời gian với TargetType của hộp chỉ mục thời gian có thể được sử dụng để xây dựng chỉ mục thời gian phân cấp, cho phép người dùng tập tin nhanh chóng điều hướng đến phần chỉ mục yêu cầu.

Chỉ mục phân đoạn có thể, ví dụ, có cú pháp như sau:

```
aligned(8) class TimeIndexBox
extends FullBox('frai') {
    unsigned int(32) targettype;
```

```

unsigned int(32) time_reference_track_ID;
unsigned int(32) number_of_elements;
unsigned int(64) first_element_offset;
unsigned int(64) first_element_time;
for(i=1; i <= number_of_elements; i++)
{
    bit (1)      random_access_flag;
    unsigned int(31) length;
    unsigned int(32) deltaT;
}
}

```

Ngữ nghĩa

TargetType: là loại dữ liệu hộp được tham chiếu bởi hộp chỉ mục thời gian này. Có thể là (phần tiêu đề đoạn phim-Movie Fragment Header ("moof") hoặc hộp chỉ mục thời gian (Time Index Box ("tidx")).

Time-reference_track_id: Chỉ rõ rãnh mà độ dịch thời gian trong chỉ mục này được xác định.

Number_of_elements: Số lượng phần tử được lập chỉ mục bởi hộp chỉ mục thời gian.

First_element_offset: Độ dịch byte từ đầu của tập tin của phần tử được lập chỉ mục thứ nhất.

First_element_time: Thời gian bắt đầu của phần tử được lập chỉ mục thứ nhất, sử dụng thang thời gian được chỉ rõ trong hộp phần tiêu đề phương tiện của rãnh được xác định bởi time_reference_track_id.

Random_access_flag: Bằng một khi thời gian bắt đầu của phần tử này là điểm truy cập ngẫu nhiên, bằng không trong các trường hợp khác.

Length: Chiều dài của phần tử được lập chỉ mục tính theo byte.

DeltaT: Hiệu số xét về thang thời gian được chỉ rõ trong hộp phần tiêu đề phương tiện của rãnh được xác định bởi time_reference_track_id giữa thời gian bắt đầu của phần tử này và thời gian bắt đầu của phần tử tiếp theo.

Hộp chỉ mục phân đoạn

Hộp chỉ mục phân đoạn ('sidx') cung cấp chỉ mục nhỏ gọn của các đoạn phim và các hộp chỉ mục phân đoạn khác trong phân đoạn. Có hai cấu trúc vòng lặp trong

hộp chỉ mục phân đoạn. Vòng lặp thứ nhất ghi lại mẫu thứ nhất của phân đoạn nhỏ, có nghĩa là, mẫu trong đoạn phim thứ nhất được tham chiếu bởi vòng lặp thứ hai. Vòng lặp thứ hai cung cấp chỉ mục của phân đoạn nhỏ. Bộ chứa cho hộp 'sidx' trực tiếp là tập tin hay phân đoạn.

Cú pháp

```
aligned(8) class SegmentIndexBox extends FullBox('sidx', version, 0) {
    unsigned int(32) reference_track_ID;
    unsigned int(16) track_count;
    unsigned int(16) reference_count;
    for (i=1; i<= track_count; i++)
    {
        unsigned int(32)    track_ID;
        if (version==0)
        {
            unsigned int(32)    decoding_time;
        } else
        {
            unsigned int(64)    decoding_time;
        }
    }
    for(i=1; i <= reference_count; i++)
    {
        bit (1)                reference_type;
        unsigned int(31)        reference_offset;
        unsigned int(32)        subsegment_duration;
        bit(1)                  contains_RAP;
        unsigned int(31)        RAP_delta_time;
    }
}
```

Ngữ nghĩa:

Reference_track_ID cung cấp track_ID cho rãnh tham chiếu.

Track_count: số lượng rãnh được lập chỉ mục trong vòng lặp sau (1 hoặc lớn hơn);

Reference_count: số lượng phần tử được lập chỉ mục bởi vòng lặp thứ hai (1 hoặc lớn hơn);

track_ID: ID của rãnh mà đoạn rãnh được bao gồm trong đoạn phim thứ nhất được xác định bởi chỉ mục này; chính xác một track_ID trong vòng lặp này bằng với reference_track_ID;

decoding_time: thời gian giải mã cho mẫu thứ nhất trong rãnh được xác định bởi track_ID trong đoạn phim được tham chiếu bởi mục thứ nhất trong vòng lặp thứ hai, được thể hiện là thang thời gian của rãnh (được ghi lại trong trường thang thời gian của hộp phần tiêu đề phương tiện của rãnh);

Reference_type: khi được thiết lập là 0, cho thấy rằng tham chiếu đến hộp đoạn phim ('moof'); khi được thiết lập là 1, cho thấy rằng tham chiếu đến hộp chỉ mục phân đoạn ('sidx');

Reference_offset: khoảng cách tính theo byte từ byte thứ nhất sau hộp chỉ mục phân đoạn chứa, đến byte thứ nhất của hộp tham chiếu;

Subsegment_duration: khi tham chiếu đến hộp chỉ mục phân đoạn, trường này mang tổng của các trường subsegment_duration trong vòng lặp thứ hai của hộp; khi tham chiếu đến đoạn phim, trường này mang tổng thời lượng mẫu của các mẫu trong rãnh tham chiếu, trong đoạn phim được chỉ báo và các đoạn phim tiếp theo đến đoạn phim thứ nhất được ghi lại bởi mục nhập tiếp theo trong vòng lặp, hoặc phần kết thúc của phân đoạn, tùy theo phần tử nào sớm hơn; thời lượng được thể hiện trong thang thời gian của rãnh (như được ghi lại trong trường thang thời gian của hộp phần tiêu đề phương tiện của rãnh);

Contains_RAP: khi tham chiếu đến đoạn phim, thì bit này có thể là 1 nếu đoạn rãnh trong đoạn phim cho rãnh có track_ID bằng reference_track_ID chứa ít nhất một điểm truy cập ngẫu nhiên, ngược lại bit này được thiết lập là 0; khi tham chiếu đến chỉ mục phân đoạn, thì bit này được thiết lập là 1 chỉ khi một trong các thông tin tham chiếu trong chỉ mục phân đoạn có bit này được thiết lập là 1, và 0 nếu ngược lại;

RAP_delta_time: nếu contains_RAP là 1, cung cấp thời gian trình diễn (hộp thành) của điểm truy cập ngẫu nhiên (RAP); được dự trữ với giá trị 0 nếu contains_RAP là 0. Thời gian được thể hiện là hiệu số giữa thời gian giải mã của mẫu thứ nhất của phân đoạn được ghi lại bởi mục nhập này và thời gian trình diễn (hộp

thành) của điểm truy cập ngẫu nhiên, trong rãnh với track_ID bằng reference_track_ID.

Sự khác biệt giữa TIDX và SIDX

TIDX và SIDX cung cấp cùng chức năng liên quan đến việc lập chỉ mục. Vòng lặp thứ nhất của SIDX cung cấp định thời chung bổ sung cho đoạn phim thứ nhất, nhưng việc định thời gian chung có thể cũng được chứa trong bản thân đoạn phim, tuyệt đối hoặc tương đối so với rãnh tham chiếu.

Vòng lặp thứ hai của SIDX thực hiện chức năng của TIDX. Cụ thể, SIDX cho phép có hỗn hợp các đích để tham chiếu cho mỗi chỉ mục được tham chiếu đến bởi reference_type, trong khi đó TIDX chỉ tham chiếu đến duy nhất TIDX hoặc duy nhất MOOF. Number_of_elements trong TIDX tương ứng với reference_count trong SIDX, time - reference_track_id trong TIDX tương ứng với reference_track_ID trong SIDX, first_element_offset trong TIDX tương ứng với reference_offset trong mục nhập thứ nhất của vòng lặp thứ hai, first_element_time trong TIDX tương ứng với decoding_time của reference_track trong vòng lặp thứ nhất, random_access_flag trong TIDX tương ứng với contains_RAP trong SIDX với khả năng tự do bổ sung mà trong SIDX RAP có thể không nhất thiết phải được đặt ở đầu đoạn, và do đó yêu cầu RAP_delta_time, chiều dài trong TIDX tương ứng với reference_offset trong SIDX và cuối cùng deltaT trong TIDX tương ứng với subsegment_duration trong SIDX. Vì vậy các chức năng của hai hộp là tương đương nhau.

Kích thước khối thay đổi và các khối GOP phụ

Đối với phương tiện video, mối quan hệ giữa cấu trúc mã hóa video và cấu trúc khối cho các yêu cầu có thể rất quan trọng. Ví dụ, nếu mỗi khối bắt đầu với điểm tìm kiếm, chẳng hạn như điểm truy cập ngẫu nhiên ("RAP"), và mỗi khối biểu diễn một khoảng thời gian video bằng nhau, thì vị trí của ít nhất một số điểm tìm kiếm trong phương tiện video là cố định và các điểm tìm kiếm sẽ xuất hiện tại các khoảng đều đặn trong khi mã hóa video. Như đã biết bởi người có hiểu biết trung bình trong lĩnh vực mã hóa video, hiệu suất nén có thể được cải thiện nếu điểm tìm kiếm được đặt theo mối quan hệ giữa các khung video, và cụ thể, nếu chúng được đặt tại các khung có ít điểm chung với các khung trước. Yêu cầu rằng các khối biểu diễn lượng thời gian

bằng nhau như vậy sẽ đặt ra hạn chế cho mã hóa video, như vậy quy trình nén có thể ít tối ưu.

Do đó, mong muốn là cho phép vị trí của các điểm tìm kiếm trong phân trình diễn video được chọn bởi hệ thống mã hóa video, chứ không phải là yêu cầu điểm tìm kiếm tại các vị trí cố định. Việc cho phép hệ thống mã hóa video chọn điểm tìm kiếm dẫn đến khả năng nén video được cải thiện và do đó có thể cung cấp chất lượng phương tiện video tốt hơn bằng cách sử dụng băng thông có sẵn cho trước, kết quả là cải thiện trải nghiệm người dùng. Hệ thống truyền liên tục yêu cầu theo khối hiện thời có thể yêu cầu tất cả các khối có cùng thời lượng (theo thời gian video), và mỗi khối phải bắt đầu với điểm tìm kiếm và như vậy đây là nhược điểm của các hệ thống hiện có.

Hệ thống truyền liên tục yêu cầu theo khối mới mang lại ưu điểm so với các hệ thống ở trên sẽ được mô tả dưới đây. Theo một phương án, quy trình mã hóa video của phiên bản thứ nhất của thành phần video có thể được tạo cấu hình để chọn vị trí của các điểm tìm kiếm để tối ưu hóa hiệu suất nén, nhưng với yêu cầu rằng có giá trị tối đa về thời lượng giữa các điểm tìm kiếm. Yêu cầu thứ hai này hạn chế việc chọn điểm tìm kiếm bởi quy trình mã hóa và do đó làm giảm hiệu suất nén. Tuy nhiên, việc giảm hiệu suất nén là nhỏ so với kết quả phát sinh nếu cần đến vị trí cố định thông thường cho các điểm tìm kiếm, với điều kiện là giá trị tối đa về thời lượng giữa các điểm tìm kiếm không quá nhỏ (ví dụ, lớn hơn khoảng một giây). Hơn nữa, nếu giá trị tối đa về thời lượng giữa các điểm tìm kiếm là một vài giây, thì việc giảm hiệu suất nén so với việc định vị hoàn toàn tự do điểm tìm kiếm nói chung là rất nhỏ.

Theo nhiều phương án, bao gồm cả phương án này, có thể là một số RAP không phải là điểm tìm kiếm, ví dụ, có thể có một khung mà là RAP ở giữa hai điểm tìm kiếm liên tiếp mà không được chọn là điểm tìm kiếm, ví dụ vì RAP quá gần về thời gian với các điểm tìm kiếm xung quanh, hoặc vì lượng dữ liệu phương tiện giữa các điểm tìm kiếm trước hoặc sau RAP và RAP là quá nhỏ.

Vị trí của các điểm tìm kiếm trong tất cả các phiên bản khác của phân trình diễn phương tiện có thể bị hạn chế là giống như các điểm tìm kiếm trong phiên bản thứ nhất (ví dụ, tốc độ dữ liệu phương tiện cao nhất). Điều này làm giảm hiệu suất nén cho các phiên bản khác so với phiên bản cho phép bộ mã hóa tự do chọn điểm tìm kiếm.

Việc sử dụng các điểm tìm kiếm thường đòi hỏi một khung có thể giải mã được một cách độc lập, điều này thường tạo ra hiệu suất nén thấp cho khung đó. Các khung mà không cần phải giải mã được một cách độc lập có thể được mã hóa tham chiếu đến dữ liệu trong các khung khác, điều này thường làm tăng hiệu suất nén cho khung đó với một lượng phụ thuộc vào lượng tương đồng giữa khung được mã hóa và khung tham chiếu. Tốt hơn là, việc chọn hiệu quả vị trí tìm kiếm sẽ chọn, làm khung điểm tìm kiếm, khung có mức tương đồng thấp với các khung trước đó và do đó giảm thiểu mức chịu phạt hiệu suất nén xảy ra bằng cách mã hóa khung này theo cách có thể giải mã được độc lập.

Tuy nhiên, mức độ tương đồng giữa một khung và khung tham chiếu tiềm năng có liên quan chặt chẽ giữa các dạng biểu diễn khác nhau của nội dung, vì nội dung gốc là như nhau. Kết quả là, việc hạn chế các điểm tìm kiếm trong các biến thể khác có cùng vị trí với điểm tìm kiếm trong biến thể thứ nhất không tạo ra sự khác biệt lớn trong hiệu suất nén.

Tốt hơn là, cấu trúc điểm tìm kiếm được sử dụng để xác định cấu trúc khối. Tốt hơn là, mỗi điểm tìm kiếm xác định sự bắt đầu của một khối, và có thể có một hoặc nhiều khối bao gồm dữ liệu giữa hai điểm tìm kiếm liên tiếp. Do thời lượng giữa các điểm tìm kiếm không được cố định để mã hóa với khả năng nén tốt, không phải tất cả các khối cần phải có cùng thời lượng phát ra. Theo một số phương án, các khối được đồng chỉnh giữa các phiên bản nội dung - có nghĩa là, nếu có một khối trải trên một nhóm cụ thể các khung trong một phiên bản nội dung, thì có một khối trải trên cùng nhóm khung trong phiên bản nội dung khác. Các khối cho một phiên bản nhất định của nội dung không chồng lấn và mỗi khung của nội dung được chứa trong đúng một khối của mỗi phiên bản.

Tính năng cho phép mà cho phép sử dụng hiệu quả thời lượng thay đổi giữa các điểm tìm kiếm, và do đó cả thời lượng thay đổi của các GOP, là tạo chỉ mục phân đoạn hoặc ánh xạ phân đoạn mà có thể được bao gồm trong một phân đoạn hoặc được cung cấp bởi các phương tiện khác cho máy khách, tức là, đây là siêu dữ liệu gắn với phân đoạn này trong dạng biểu diễn này mà có thể được cung cấp bao gồm thời gian bắt đầu và thời lượng của mỗi khối của phần trình diễn. Máy khách có thể sử dụng dữ liệu lập chỉ mục phân đoạn khi xác định khối mà tại đó để bắt đầu phần trình diễn khi người dùng đã yêu cầu rằng phần trình diễn bắt đầu tại điểm cụ thể ở trong phân đoạn. Nếu

như siêu dữ liệu không được cung cấp, thì phần trình diễn chỉ có thể bắt đầu tại chỗ bắt đầu của nội dung, hoặc tại điểm ngẫu nhiên hoặc xấp xỉ gần với điểm mong muốn (ví dụ bằng cách chọn khối bắt đầu bằng cách chia điểm bắt đầu được yêu cầu (theo thời gian) cho thời lượng trung bình của khối để cung cấp chỉ mục của khối bắt đầu).

Theo một phương án, mỗi khối có thể được cung cấp như là một tập tin riêng biệt. Theo phương án khác, nhiều khối liên tiếp có thể được gộp lại thành một tập tin duy nhất để tạo thành phân đoạn. Theo phương án thứ hai này, siêu dữ liệu cho mỗi phiên bản có thể được cung cấp bao gồm thời gian bắt đầu và thời lượng của mỗi khối và độ dịch byte trong tập tin mà tại đó khối bắt đầu. Siêu dữ liệu này có thể được cung cấp để phản hồi yêu cầu giao thức ban đầu, tức là, có sẵn riêng biệt với phân đoạn hoặc tập tin, hoặc có thể được chứa trong cùng một tập tin hoặc phân đoạn như bản thân các khối, ví dụ như ở chỗ bắt đầu của tập tin. Người có hiểu biết trung bình trong lĩnh vực kỹ thuật sẽ thấy rằng, siêu dữ liệu này có thể được mã hóa dưới dạng nén như gzip hoặc mã hóa delta hoặc ở dạng nhị phân, để làm giảm tài nguyên mạng cần thiết để truyền tải siêu dữ liệu đến máy khách.

Fig.6 thể hiện ví dụ về việc lập chỉ mục phân đoạn, trong đó các khối có kích thước thay đổi, và trong đó phạm vi của các khối là GOP một phần, tức là, một phần của dữ liệu phương tiện giữa một RAP và RAP tiếp theo. Trong ví dụ này, các điểm tìm kiếm được chỉ báo bởi chỉ báo RAP, trong đó giá trị chỉ báo RAP bằng 1 chỉ ra khối bắt đầu bằng hoặc chứa một RAP, hoặc điểm tìm kiếm, và trong đó chỉ báo RAP bằng 0 chỉ ra rằng khối không chứa RAP hoặc điểm tìm kiếm. Trong ví dụ này, ba khối thứ nhất, tức là từ byte 0 đến 157.033, bao gồm GOP thứ nhất, có thời lượng trình diễn 1,623 giây, với thời gian trình diễn chạy từ 20 giây vào nội dung 21,623 giây. Trong ví dụ này, khối thứ nhất trong ba khối thứ nhất có 0,485 giây thời gian trình diễn, và bao gồm 50.245 byte thứ nhất của dữ liệu phương tiện trong phân đoạn này. Trong ví dụ này, khối 4, 5, và 6 bao gồm GOP thứ hai, khối 7 và 8 bao gồm GOP thứ ba, và khối 9, 10 và 11 bao gồm GOP thứ tư. Lưu ý rằng có thể có các RAP khác trong dữ liệu phương tiện mà không được gán là điểm tìm kiếm, và do đó không được báo hiệu là RAP trong ánh xạ phân đoạn.

Trở lại Fig.6, nếu máy khách hoặc bộ thu muốn truy cập nội dung bắt đầu tại độ dịch thời gian khoảng 22 giây vào phần trình diễn phương tiện, thì trước tiên máy khách có thể sử dụng thông tin khác, chẳng hạn như MPD được mô tả chi tiết hơn sau

đây, để trước tiên xác định rằng dữ liệu phương tiện liên quan nằm trong phân đoạn này. Máy khách có thể tải về phần thứ nhất của phân đoạn để thu được chỉ mục phân đoạn, mà trong trường hợp này chỉ là một vài byte, ví dụ bằng cách sử dụng yêu cầu dãy byte HTTP. Bằng cách sử dụng lập chỉ mục phân đoạn, máy khách có thể xác định rằng khối thứ nhất mà nó nên tải về là khối thứ nhất với độ dịch thời gian tối đa 22 giây và bắt đầu với một RAP, tức là, điểm tìm kiếm. Trong ví dụ này, mặc dù khối 5 có độ dịch thời gian nhỏ hơn 22 giây, nghĩa là độ dịch thời gian của nó là 21,965 giây, việc lập chỉ mục phân đoạn chỉ ra rằng khối 5 không bắt đầu với một RAP, mà thay vào đó, dựa trên việc lập chỉ mục phân đoạn, máy khách chọn tải về khối 4, do thời gian bắt đầu của nó tối đa là 22 giây, nghĩa là độ dịch thời gian của nó là 21,623 giây, và nó bắt đầu với một RAP. Như vậy, dựa trên việc lập chỉ mục phân đoạn, máy khách sẽ thực hiện yêu cầu dãy byte HTTP bắt đầu ở độ dịch byte 157.034.

Nếu việc lập chỉ mục phân đoạn không có sẵn, thì máy khách có thể phải tải về tất cả 157.034 byte dữ liệu trước đó trước khi tải về dữ liệu này, điều này dẫn đến thời gian khởi động lâu hơn rất nhiều, hoặc thời gian sửa đổi kênh tăng, và lãng phí việc tải dữ liệu mà không hữu ích. Theo cách khác, nếu việc lập chỉ mục phân đoạn không có sẵn, máy khách có thể tính gần đúng nơi dữ liệu mong muốn bắt đầu trong phân đoạn này, nhưng việc tính gần đúng này có thể kém và có thể bỏ lỡ thời điểm thích hợp và sau đó yêu cầu quay trở lại, điều này lại làm tăng độ trễ khởi động.

Nói chung, mỗi khối bao gồm một phần của dữ liệu phương tiện mà, cùng với các khối trước đó, có thể được phát bởi máy phát phương tiện. Như vậy, cấu trúc khối và việc báo hiệu cấu trúc khối lập chỉ mục phân đoạn cho máy khách, được chứa trong phân đoạn hoặc được cung cấp cho máy khách qua phương tiện khác, có thể cải thiện đáng kể khả năng của máy khách để cung cấp thời gian sửa đổi kênh nhanh chóng, và phát ra liên tục trong khi có sự thay đổi và xáo trộn của mạng. Sự hỗ trợ của các khối có thời lượng thay đổi, và các khối mà chỉ bao gồm các phần của GOP, khi được kích hoạt bởi việc lập chỉ mục phân đoạn, có thể cải thiện đáng kể trải nghiệm truyền liên tục. Ví dụ, trở lại Fig.6 và ví dụ mô tả ở trên, khi máy khách muốn bắt đầu phát ra ở xấp xỉ 22 giây trong phần trình diễn, máy khách có thể yêu cầu, qua một hoặc nhiều yêu cầu, dữ liệu trong khối 4, và sau đó nạp dữ liệu này vào máy phát phương tiện ngay khi nó có sẵn để bắt đầu phát lại. Như vậy, trong ví dụ này, việc phát ra bắt đầu ngay khi 42.011 byte của khối 4 được nhận tại máy khách, do đó cho phép có thời gian

sửa đổi kênh nhanh chóng. Nếu thay vào đó máy khách cần phải yêu cầu toàn bộ GOP trước khi việc phát ra bắt đầu, thời gian sửa đổi kênh sẽ được lâu hơn, vì đây là 144.211 byte dữ liệu.

Trong các phương án khác, RAP hoặc các điểm tìm kiếm cũng có thể xuất hiện ở giữa khối, và có thể có dữ liệu trong chỉ mục phân đoạn mà chỉ báo vị trí mà RAP hoặc điểm tìm kiếm đó ở trong khối hoặc đoạn. Trong các phương án khác, độ dịch thời gian có thể là thời gian giải mã khung thứ nhất trong khối, thay vì thời gian trình diễn của khung thứ nhất trong khối.

Fig.8(a) và Fig.8(b) minh họa ví dụ về định cỡ khối thay đổi theo cấu trúc điểm tìm kiếm được đồng chỉnh trên các phiên bản hoặc các dạng biểu diễn; Fig.8 (a) minh họa việc định cỡ thay đổi với các điểm tìm kiếm được đồng chỉnh trên các phiên bản của dòng phương tiện, trong khi Fig.8(b) minh họa việc định cỡ khối thay đổi với các điểm tìm kiếm không được đồng chỉnh trên các phiên bản của dòng phương tiện.

Thời gian được thể hiện ở trên cùng tính theo giây, các khối và điểm tìm kiếm của hai phân đoạn cho hai dạng biểu diễn được thể hiện từ trái sang phải theo thời gian của chúng liên quan đến dòng thời gian này, và do đó độ dài của mỗi khối được thể hiện tỷ lệ thuận với thời gian phát ra nó và không tỷ lệ thuận với số byte trong khối. Trong ví dụ này, việc lập chỉ mục phân đoạn cho cả hai phân đoạn của hai dạng biểu diễn sẽ có cùng độ dịch thời gian cho các điểm tìm kiếm, nhưng số lượng khối hoặc đoạn có thể khác nhau giữa các điểm tìm kiếm, và độ dịch byte khác nhau đối với các khối do lượng dữ liệu phương tiện khác nhau trong mỗi khối. Trong ví dụ này, nếu máy khách muốn chuyển đổi từ dạng biểu diễn từ 1 sang dạng biểu diễn 2 vào thời gian trình diễn khoảng 23 giây, thì máy khách có thể yêu cầu qua khối 1.2 trong phân đoạn để có dạng biểu diễn 1, và bắt đầu yêu cầu phân đoạn này để có dạng biểu diễn 2 bắt đầu ở khối 2.2, và do đó việc chuyển đổi sẽ xảy ra ở phần trình diễn trùng với điểm tìm kiếm 1.2 trong dạng biểu diễn 1, mà ở cùng thời điểm với điểm tìm kiếm 2.2 trong dạng biểu diễn 2.

Như được nêu rõ từ phần mô tả trên đây, hệ thống truyền liên tục yêu cầu theo khối được mô tả không hạn chế việc mã hóa video để đặt các điểm tìm kiếm tại các vị trí cụ thể trong nội dung và điều này làm giảm nhẹ một trong các vấn đề của các hệ thống hiện có.

Trong các phương án được mô tả ở trên, hệ thống này được tổ chức sao cho các điểm tìm kiếm cho các dạng biểu diễn khác nhau của cùng một nội dung trình diễn được đồng chỉnh. Tuy nhiên, trong nhiều trường hợp, tốt hơn là nói lòng yêu cầu đồng chỉnh này. Ví dụ, đôi khi các công cụ mã hóa đã được sử dụng để tạo ra các dạng biểu diễn mà không có khả năng tạo ra các dạng biểu diễn đã đồng chỉnh điểm tìm kiếm. Theo ví dụ khác, phần trình diễn nội dung có thể được mã hóa thành các dạng biểu diễn khác nhau một cách độc lập, không có sự đồng chỉnh điểm tìm kiếm giữa dạng biểu diễn khác nhau này. Theo ví dụ khác, dạng biểu diễn có thể chứa nhiều điểm tìm kiếm vì nó có tốc độ thấp hơn và thông thường nó cần phải được chuyển hoặc nó chứa điểm tìm kiếm để hỗ trợ các chế độ kỹ xảo, chẳng hạn như, tua nhanh hoặc tua ngược hoặc tìm kiếm nhanh. Vì vậy, mong muốn là có thể cung cấp các phương pháp mà làm cho hệ thống truyền liên tục yêu cầu theo khối có khả năng đối phó hiệu quả và liên tục với các điểm tìm kiếm không đồng chỉnh trên các dạng biểu diễn khác nhau cho phần trình diễn nội dung.

Theo phương án này, vị trí của các điểm tìm kiếm trên các dạng biểu diễn có thể không đồng chỉnh. Các khối được xây dựng sao cho một khối mới bắt đầu tại mỗi điểm tìm kiếm, và do đó có thể không có sự đồng chỉnh giữa các khối của các phiên bản khác nhau của phần trình diễn. Ví dụ về cấu trúc điểm tìm kiếm không được đồng chỉnh giữa các dạng biểu diễn khác nhau được thể hiện trên Fig.8(b). Thời gian được thể hiện ở trên cùng tính theo giây, và các khối và điểm tìm kiếm của hai phân đoạn cho hai dạng biểu diễn được thể hiện từ trái sang phải theo thời gian của chúng liên quan đến dòng thời gian này, và do đó độ dài của mỗi khối được thể hiện tỷ lệ thuận với thời gian phát ra nó và không tỷ lệ thuận với số byte trong khối. Trong ví dụ này, việc lập chỉ mục phân đoạn cho cả hai phân đoạn của hai dạng biểu diễn sẽ có độ dịch thời gian có thể khác nhau cho các điểm tìm kiếm, và số lượng các khối hoặc đoạn có thể khác nhau giữa các điểm tìm kiếm, và độ dịch byte khác nhau đối với các khối do lượng dữ liệu phương tiện khác nhau trong mỗi khối. Trong ví dụ này, nếu máy khách muốn chuyển từ dạng biểu diễn 1 sang dạng biểu diễn 2 vào thời gian trình diễn khoảng 25 giây, thì máy khách có thể yêu cầu qua khối 1.3 trong phân đoạn để có dạng biểu diễn 1, và bắt đầu yêu cầu phân đoạn này để có dạng biểu diễn 2 bắt đầu ở khối 2.3, và do đó việc chuyển đổi sẽ xảy ra ở phần trình diễn trùng với điểm tìm kiếm 2.3 trong dạng biểu diễn 2, mà ở giữa phần phát ra của khối 1.3 trong dạng biểu diễn 1, và

do đó một số phương tiện cho khối 1.2 sẽ không được phát ra (mặc dù dữ liệu phương tiện cho các khung của khối 1.3 không được phát ra có thể được nạp vào bộ đệm của bộ thu để giải mã các khung khác của khối 1.3 được phát ra).

Theo phương án này, hoạt động của bộ chọn khối 123 có thể được sửa đổi sao cho bất cứ khi nào cần chọn một khối từ dạng biểu diễn khác với các phiên bản đã chọn trước đó, khối mới nhất có khung thứ nhất không muộn hơn khung đứng sau khung cuối cùng của khối chọn cuối cùng sẽ được chọn.

Phương án được mô tả cuối cùng này có thể loại bỏ yêu cầu hạn chế vị trí của các điểm tìm kiếm trong các phiên bản khác so với phiên bản thứ nhất và do đó làm tăng hiệu suất nén cho các phiên bản này, tạo ra phần trình diễn chất lượng cao hơn cho băng thông có sẵn nhất định và cải thiện trải nghiệm của người dùng. Một mối quan tâm nữa là các công cụ mã hóa video thực hiện chức năng đồng chỉnh điểm tìm kiếm trên nhiều phần mã hóa (phiên bản) của nội dung có thể không được phổ biến rộng rãi và do lợi thế của phương án được mô tả cuối cùng này là công cụ mã hóa video hiện có có thể được sử dụng. Lợi thế khác đó là việc mã hóa các phiên bản khác nhau của nội dung có thể tiến hành song song mà không cần sự phối hợp giữa các quy trình mã hóa cho các phiên bản khác nhau. Lợi thế khác đó là các phiên bản bổ sung nội dung có thể được mã hóa và được bổ sung vào phần trình diễn sau đó, mà không cần phải cung cấp cho các công cụ mã hóa danh sách vị trí điểm tìm kiếm cụ thể.

Nói chung, khi mà hình ảnh được mã hóa như là nhóm các hình ảnh (GOP), hình ảnh thứ nhất trong chuỗi có thể là điểm tìm kiếm, nhưng không phải luôn luôn như vậy.

Phân chia khối tối ưu

Một vấn đề quan tâm trong hệ thống truyền liên tục yêu cầu theo khối là sự tương tác giữa cấu trúc của phương tiện được mã hóa, ví dụ phương tiện video, và cấu trúc khối được sử dụng cho các yêu cầu theo khối. Như đã biết bởi người có hiểu biết trung bình trong lĩnh vực mã hóa video, thường thì số lượng bit cần thiết cho dạng biểu diễn mã hóa của mỗi khung video sẽ khác nhau, đôi khi đáng kể, giữa các khung. Kết quả là mối quan hệ giữa lượng dữ liệu thu được và thời lượng phương tiện được mã hóa bởi dữ liệu đó có thể không đơn giản. Hơn nữa, việc chia dữ liệu phương tiện thành khối trong hệ thống truyền liên tục yêu cầu theo khối còn gia tăng thêm sự phức

tạp. Cụ thể, trong một số hệ thống, dữ liệu phương tiện của khối có thể không được phát ra cho đến khi toàn bộ khối đã được nhận, ví dụ như sự sắp xếp dữ liệu phương tiện trong khối hoặc sự phụ thuộc giữa các mẫu phương tiện trong khối sử dụng mã xoá có thể tạo ra đặc tính này. Kết quả của tương tác phức tạp giữa kích thước khối và thời lượng khối và nhu cầu có thể thu được toàn bộ khối trước khi bắt đầu phát ra đó là hệ thống máy khách thường áp dụng phương pháp bảo toàn trong đó dữ liệu phương tiện được đệm trước khi việc phát ra bắt đầu. Việc đệm như vậy dẫn đến thời gian sửa đổi kênh dài và do đó trải nghiệm người dùng kém.

Pakzad mô tả "phương pháp phân chia khối" là phương pháp mới và hiệu quả để xác định cách để phân chia dòng dữ liệu thành các khối liên tiếp dựa trên cấu trúc cơ bản của các dòng dữ liệu và còn mô tả một số ưu điểm của các phương pháp này trong ngữ cảnh của hệ thống truyền liên tục. Phương án khác của sáng chế để áp dụng các phương pháp phân chia khối của Pakzad cho hệ thống truyền liên tục yêu cầu theo khối được mô tả dưới đây. Phương pháp này có thể bao gồm việc sắp xếp dữ liệu phương tiện sẽ được trình diễn theo trình tự thời gian trình diễn gần đúng, sao cho thời gian phát ra của phần tử bất kỳ cho trước của dữ liệu phương tiện (ví dụ như khung video hoặc mẫu âm thanh) khác với thời gian phát ra của phần tử dữ liệu phương tiện dữ liệu liên tiếp bất kỳ bởi một lượng ít hơn một ngưỡng cụ thể. Dữ liệu phương tiện nên được sắp xếp như vậy có thể được coi là dòng dữ liệu theo ngôn ngữ của Pakzad và phương pháp bất kỳ trong số các phương pháp của Pakzad áp dụng cho dòng dữ liệu này xác định biên khối với dòng dữ liệu đó. Dữ liệu giữa cặp biên khối liên tiếp bất kỳ được coi là một "khối" theo ngôn ngữ của sáng chế và các phương pháp của sáng chế được áp dụng để cung cấp phần trình diễn dữ liệu phương tiện trong hệ thống truyền liên tục yêu cầu theo khối. Như đã biết bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật khi đọc sáng chế, một số lợi thế của các phương pháp được bộc lộ bởi Pakzad có thể được thực hiện trong ngữ cảnh của hệ thống truyền liên tục yêu cầu theo khối.

Như được mô tả trong Pakzad, việc xác định cấu trúc khối của phân đoạn, bao gồm các khối mà chứa các GOP một phần hoặc các phần khác ngoài GOP, có thể ảnh hưởng đến khả năng cho phép thời gian sửa đổi kênh nhanh của máy khách. Trong Pakzad, phương pháp được đưa ra mà, cho trước thời gian khởi động mục tiêu, sẽ cung cấp cấu trúc khối và tốc độ tải về mục tiêu mà đảm bảo rằng nếu máy khách bắt đầu tải

về dạng biểu diễn tại điểm tìm kiếm bất kỳ và bắt đầu phát ra sau khi thời gian khởi động mục tiêu đã trôi qua thì việc phát ra sẽ tiếp diễn liên tục miễn là tại mỗi điểm thời gian, lượng dữ liệu máy khách đã tải về bằng ít nhất là tốc độ tải về mục tiêu nhân với thời gian trôi qua từ khi bắt đầu tải về. Có lợi khi máy khách có thể truy cập thời gian khởi động mục tiêu và tốc độ tải về mục tiêu, vì điều này cung cấp cho máy khách phương tiện để xác định khi nào bắt đầu phát lại dạng biểu diễn tại thời điểm sớm nhất, và cho phép máy khách tiếp tục phát ra dạng biểu diễn miễn là việc tải về đáp ứng điều kiện được mô tả ở trên. Vì vậy, phương pháp mô tả sau đây cung cấp phương tiện gộp thời gian khởi động mục tiêu và tốc độ tải về mục tiêu trong phần mô tả trình diễn phương tiện, để nó có thể được sử dụng cho các mục đích mô tả ở trên.

Mô hình dữ liệu trình diễn phương tiện

Fig.5 minh họa các cấu trúc có thể có của bộ phận lưu trữ nội dung được thể hiện trên Fig.1, bao gồm các phân đoạn và tập tin mô tả trình diễn phương tiện ("MPD"), và sự tách rời phân đoạn, thời gian, và cấu trúc khác trong tập tin MPD. Chi tiết về các cài đặt có thể có của các cấu trúc hoặc tập tin MPD sẽ được mô tả dưới đây. Trong nhiều ví dụ, MPD được mô tả như là một tập tin, nhưng các cấu trúc không phải tập tin cũng có thể được sử dụng.

Như được minh họa, bộ phận lưu trữ nội dung 110 chứa các phân đoạn nguồn 510, MPD 500 và phân đoạn sửa chữa 512. MPD có thể bao gồm các bản ghi khoảng thời gian 501, mà có thể bao gồm các bản ghi dạng biểu diễn 502, có chứa thông tin phân đoạn 503 chẳng hạn như tham chiếu đến phân đoạn khởi tạo 504 và phân đoạn phương tiện 505.

Fig.9(a) minh họa bảng siêu dữ liệu 900 làm ví dụ, trong khi Fig.9(b) minh họa ví dụ về cách thức mà máy khách truyền liên tục HTTP 902 thu được bảng siêu dữ liệu 900 và khối phương tiện 904 qua kết nối với máy chủ truyền liên tục HTTP 906.

Trong các phương pháp được mô tả ở đây, "mô tả trình diễn phương tiện" được cung cấp bao gồm thông tin về các dạng biểu diễn của phần trình diễn phương tiện có sẵn cho máy khách. Dạng biểu diễn có thể thay thế cho nhau theo nghĩa là máy khách chọn một trong các phương án thay thế khác nhau, hoặc chúng có thể bổ sung cho nhau theo nghĩa là máy khách chọn một vài dạng biểu diễn, mỗi dạng biểu diễn có thể từ tập hợp các phương án thay thế, và trình diễn chúng một cách phối hợp. Có lợi nếu các dạng biểu diễn có thể được gán cho các nhóm, với máy khách được lập trình hoặc

tạo cấu hình để hiểu rằng, đối với các dạng biểu diễn trong một nhóm, mỗi trong số chúng có thể thay thế cho nhau, trong khi đó với các dạng biểu diễn từ các nhóm khác nhau thì nhiều hơn một dạng biểu diễn phải được trình diễn một cách phối hợp. Nói cách khác, nếu có nhiều hơn một dạng biểu diễn trong một nhóm, máy khách chọn một dạng biểu diễn từ nhóm đó, một dạng biểu diễn từ nhóm tiếp theo, v.v., để tạo thành phân trình diễn.

Có lợi nếu thông tin mô tả các dạng biểu diễn có thể bao gồm các chi tiết của các phần mã hóa-giải mã phương tiện được áp dụng bao gồm các profin và mức độ của các phần mã hóa-giải mã được yêu cầu để giải mã dạng biểu diễn, tốc độ khung video, độ phân giải video và tốc độ dữ liệu. Máy khách nhận thông tin mô tả trình diễn phương tiện có thể sử dụng thông tin này để xác định trước xem là dạng biểu diễn có phù hợp để giải mã hay trình diễn hay không. Điều này có ưu điểm bởi vì nếu thông tin khác biệt chỉ được chứa trong dữ liệu nhị phân của dạng biểu diễn, thì sẽ cần phải yêu cầu dữ liệu nhị phân từ tất cả các dạng biểu diễn và phân tích cú pháp và trích xuất thông tin liên quan nhằm phát hiện thông tin về tính phù hợp của nó. Các yêu cầu này và việc trích xuất phụ lục phân tích cú pháp đối với dữ liệu có thể mất thời gian, dẫn đến thời gian khởi động lâu và do đó trải nghiệm người dùng kém.

Ngoài ra, thông tin mô tả trình diễn phương tiện có thể bao gồm thông tin hạn chế các yêu cầu của máy khách dựa trên thời gian trong ngày. Ví dụ với dịch vụ trực tiếp, máy khách có thể được hạn chế ở việc yêu cầu các phần của phân trình diễn gần với "thời gian phát rộng hiện thời". Điều này có ưu điểm do để phát rộng trực tiếp, có thể mong muốn thanh lọc dữ liệu từ cơ sở hạ tầng phục vụ cho nội dung được phát rộng nhiều hơn ngưỡng cho trước trước thời điểm phát rộng hiện thời. Điều này có thể được mong muốn để tái sử dụng các tài nguyên lưu trữ trong cơ sở hạ tầng phục vụ. Điều này cũng có thể được mong muốn phụ thuộc vào loại dịch vụ được cung cấp, ví dụ như, trong một số trường hợp phân trình diễn có thể được tạo ra sẵn chỉ ở dạng trực tiếp vì mô hình thuê bao nhất định của thiết bị máy khách thu, trong khi đó các phân trình diễn phương tiện khác có thể được tạo ra sẵn trực tiếp và theo nhu cầu, và các trình diễn khác có thể được tạo ra sẵn chỉ ở dạng trực tiếp với lớp thứ nhất của thiết bị máy khách, chỉ theo nhu cầu với lớp thứ hai của thiết bị máy khách, và kết hợp của dạng trực tiếp hoặc theo nhu cầu với lớp thứ ba của thiết bị máy khách. Các phương pháp được mô tả trong mô hình dữ liệu trình diễn phương tiện (bên dưới) cho phép

máy khách được thông báo về các chính sách này để máy khách có thể tránh đưa ra các yêu cầu và điều chỉnh dịch vụ cho người dùng, đối với dữ liệu mà có thể không có sẵn trong cơ sở hạ tầng phục vụ. Theo phương án khác, ví dụ, máy khách có thể đưa ra thông báo cho người dùng rằng dữ liệu này không có sẵn.

Theo phương án khác của sáng chế, các phân đoạn phương tiện có thể phù hợp với định dạng tập tin phương tiện cơ sở ISO được mô tả trong ISO/IEC 14496-12 hoặc có thông số kỹ thuật phái sinh (chẳng hạn như định dạng tập tin 3GP được mô tả trong đặc tả thông số kỹ thuật 3GPP 26.244). Việc sử dụng phần định dạng tập tin 3GPP (ở trên) mô tả các cải tiến mới đối với định dạng tập tin phương tiện cơ sở ISO cho phép sử dụng hiệu quả các cấu trúc dữ liệu của định dạng tập tin này trong hệ thống truyền liên tục yêu cầu theo khối. Như được mô tả trong đây, thông tin có thể được cung cấp trong tập tin cho phép lập ánh xạ nhanh chóng và hiệu quả giữa các phân đoạn thời gian của phần trình diễn phương tiện và các dãy byte trong tập tin. Bản thân dữ liệu phương tiện có thể được cấu trúc theo quy trình xây dựng phân đoạn phim được chỉ rõ trong tiêu chuẩn ISO/IEC14496 – 12. Thông tin này cung cấp thời gian và độ dịch byte mà có thể được cấu trúc phân cấp hoặc là một khối thông tin duy nhất. Thông tin này có thể được cung cấp ở chỗ bắt đầu của tập tin. Việc cung cấp thông tin này bằng cách sử dụng quy trình mã hóa hiệu quả như được mô tả trong việc sử dụng phần định dạng tập tin 3GPP dẫn đến việc máy khách có thể lấy được thông tin này một cách nhanh chóng, ví dụ như sử dụng các yêu cầu yêu cầu GET HTTP một phần, trong trường hợp giao thức tải về tập tin được sử dụng bởi hệ thống truyền liên tục yêu cầu theo khối là HTTP, điều này dẫn đến thời gian khởi động ngắn, thời gian tìm kiếm hoặc chuyển đổi dòng ngắn và do đó trải nghiệm người dùng được cải thiện.

Các dạng biểu diễn trong phần trình diễn phương tiện được đồng bộ hóa theo dòng thời gian chung để đảm bảo việc chuyển đổi liền mạch giữa các dạng biểu diễn, thường là các phương án thay thế, và để đảm bảo trình diễn đồng bộ hai hoặc nhiều dạng biểu diễn. Do đó, việc định thời mẫu của phương tiện được chứa trong các dạng biểu diễn trong phần trình diễn phương tiện truyền liên tục HTTP thích ứng có thể liên quan đến dòng thời gian chung liên tục trên nhiều phân đoạn.

Khối phương tiện mã hóa chứa nhiều loại phương tiện, ví dụ như âm thanh và video, có thể có thời gian kết thúc trình diễn khác nhau đối với các loại phương tiện khác nhau. Trong hệ thống truyền liên tục yêu cầu theo khối, các khối phương tiện này

có thể được phát ra liên tục theo cách mà mỗi loại phương tiện được phát liên tiếp và do đó các mẫu phương tiện thuộc một loại từ một khối có thể được phát ra trước các mẫu phương tiện thuộc loại khác của khối đứng trước, mà dưới đây được gọi là "ghép nối khối liên tục". Theo phương án khác, các khối phương tiện này có thể được phát ra theo cách mà mẫu sớm nhất thuộc loại bất kỳ của một khối được phát sau mẫu muộn nhất thuộc loại bất kỳ của khối trước đó, mà dưới đây được gọi là "ghép nối khối gián đoạn". Việc ghép nối khối liên tục có thể thích hợp khi cả hai khối chứa phương tiện từ cùng một mục nội dung và cùng dạng biểu diễn, được mã hóa theo thứ tự, hoặc trong các trường hợp khác. Thông thường, trong một dạng biểu diễn, việc ghép nối khối liên tục có thể được áp dụng khi ghép nối hai khối. Đây là điều thuận lợi do quy trình mã hóa hiện thời có thể được áp dụng và việc phân đoạn có thể được thực hiện mà không cần phải đồng chỉnh rãnh phương tiện tại biên khối. Điều này được minh họa trên Fig.10, trong đó dòng video 1000 bao gồm khối 1202 và các khối khác, với các RAP, chẳng hạn như, RAP 1204.

Thông tin mô tả trình diễn phương tiện

Phần trình diễn phương tiện có thể được xem là tập hợp có cấu trúc của các tập tin trên máy chủ truyền liên tục HTTP. Máy khách truyền liên tục HTTP có thể tải về đầy đủ thông tin để trình diễn dịch vụ truyền liên tục cho người dùng. Các dạng biểu diễn khác có thể bao gồm một hoặc nhiều tập tin 3GP hoặc các phần của tập tin 3GP phù hợp với định dạng 3GPP hoặc ít nhất là phù hợp với tập cấu trúc dữ liệu đã được xác định rõ mà có thể dễ dàng chuyển đổi từ hoặc thành tập tin 3GP.

Phần trình diễn phương tiện có thể được mô tả bằng thông tin mô tả trình diễn phương tiện. Thông tin mô tả trình diễn phương tiện (MPD) có thể chứa siêu dữ liệu mà máy khách có thể sử dụng để xây dựng các yêu cầu tập tin thích hợp, ví dụ yêu cầu GET HTTP, để truy cập dữ liệu ở thời điểm thích hợp và cung cấp dịch vụ truyền liên tục cho người dùng. Thông tin mô tả trình diễn phương tiện có thể cung cấp đầy đủ thông tin cho máy khách truyền liên tục HTTP để chọn tập tin 3GPP thích hợp và đoạn của tập tin. Các đơn vị được báo hiệu cho máy khách để có thể truy cập được gọi là các phân đoạn.

Trong số các phần tử khác, thông tin mô tả trình diễn phương tiện có thể chứa các thành phần và các thuộc tính như sau.

Phần tử mô tả trình diễn phương tiện

Phần tử đóng gói siêu dữ liệu được sử dụng bởi máy khách truyền liên tục HTTP để cung cấp dịch vụ truyền liên tục cho người dùng cuối. Phần tử mô tả trình diễn phương tiện có thể chứa một hoặc nhiều thuộc tính và các phần tử sau đây.

Version: Số phiên bản của giao thức để đảm bảo khả năng mở rộng.

PresentationIdentifier: Thông tin sao cho phần trình diễn có thể được xác định duy nhất trong số các phần trình diễn khác. Cũng có thể chứa các trường hoặc tên riêng.

UpdateFrequency: Tần số cập nhật của thông tin mô tả trình diễn phương tiện, tức là mức độ thường xuyên mà máy khách có thể nạp lại thông tin mô tả trình diễn phương tiện thực tế. Nếu không có mặt, phần trình diễn phương tiện có thể không thay đổi. Việc cập nhật phần trình diễn phương tiện có thể có nghĩa là phần trình diễn phương tiện không thể được lưu trữ.

MediaPresentationDescriptionURI: URI để ghi ngày tháng của thông tin mô tả trình diễn phương tiện.

Stream: Mô tả các loại dòng hoặc phần trình diễn phương tiện: video, âm thanh hoặc văn bản. Loại dòng video có thể chứa âm thanh và có thể chứa văn bản.

Service: Mô tả loại dịch vụ với các thuộc tính bổ sung. Các loại dịch vụ có thể là trực tiếp và theo nhu cầu. Điều này có thể được sử dụng để thông báo cho máy khách rằng việc tìm kiếm và truy cập vượt quá thời gian hiện thời là không được phép.

MaximumClientPreBufferTime: Lượng thời gian tối đa máy khách có thể thực hiện tác vụ đệm trước dòng phương tiện. Thời gian này có thể phân biệt quy trình truyền liên tục với quy trình tải về tuần tự nếu máy khách bị hạn chế tải về vượt quá thời gian đệm trước tối đa. Giá trị này có thể không có mặt, chỉ ra rằng không có giới hạn về việc tác vụ đệm trước có thể áp dụng.

SafetyGuardIntervalLiveService: Thông tin về thời gian quay vòng tối đa của dịch vụ trực tiếp tại máy chủ. Cung cấp một chỉ báo cho máy khách về thông tin đã có thể truy cập được tại thời điểm hiện thời. Thông tin này có thể cần thiết nếu máy khách và máy chủ dự kiến sẽ hoạt động theo thời gian UTC và không có sự đồng bộ hóa thời gian chặt chẽ.

TimeShiftBufferDepth: Thông tin về mức lùi lại mà máy khách có thể di chuyển trong dịch vụ trực tiếp so với thời điểm hiện thời. Bằng cách mở rộng độ sâu này, các dịch vụ xem có dịch thời gian và xem lại có thể được cho phép mà không có các thay đổi cụ thể trong việc cung cấp dịch vụ.

LocalCachingPermitted: Cờ này cho thấy nếu máy khách HTTP có thể lưu đệm dữ liệu tải về một cách cục bộ sau khi dữ liệu đã được phát.

LivePresentationInterval: Chứa khoảng thời gian mà phần trình diễn có thể có sẵn bằng cách chỉ rõ **StartTime** (thời gian bắt đầu) và **EndTime** (thời gian kết thúc). **StartTime** chỉ báo thời gian bắt đầu của các dịch vụ và **EndTime** chỉ báo thời gian kết thúc của dịch vụ. Nếu **EndTime** không được xác định, thì thời gian kết thúc là không biết tại thời điểm hiện tại và **UpdateFrequency** có thể đảm bảo rằng máy khách có quyền truy cập vào thời gian kết thúc trước thời gian kết thúc thực tế của dịch vụ.

OnDemandAvailabilityInterval: Khoảng thời gian trình diễn chỉ rõ sự sẵn có của dịch vụ trên mạng. Nhiều khoảng thời gian trình diễn có thể được cung cấp. Máy khách HTTP có thể không có khả năng truy cập dịch vụ bên ngoài cửa sổ thời gian cụ thể bất kỳ. Bằng việc cung cấp **OnDemand Interval** (khoảng thời gian theo nhu cầu), việc xem có dịch thời gian bổ sung có thể được xác định. Thuộc tính này cũng có thể xuất hiện cho dịch vụ trực tiếp. Trong trường hợp thuộc tính này có mặt cho dịch vụ trực tiếp, máy chủ có thể đảm bảo rằng máy khách có thể truy cập vào dịch vụ là **OnDemand Service** (dịch vụ theo nhu cầu) trong tất cả các khoảng thời gian sẵn sàng được cung cấp. Do đó, **LivePresentationInterval** (khoảng thời gian trình diễn trực tiếp) có thể không trùng với bất kỳ **OnDemandAvailabilityInterval** (khoảng thời gian có sẵn theo nhu cầu) nào.

MPDFileInfoDynamic: Mô tả việc xây dựng động mặc định các tập tin trong phần trình diễn phương tiện. Chi tiết hơn sẽ được cung cấp dưới đây. Các đặc điểm kỹ thuật mặc định về mức MPD có thể tiết kiệm được khả năng lặp không cần thiết nếu cùng quy tắc cho một số hoặc tất cả các dạng biểu diễn thay thế được sử dụng.

MPDCodecDescription: Mô tả các phân mã hóa-giải mã mặc định chính trong phần trình diễn phương tiện. Chi tiết hơn sẽ được cung cấp dưới đây. Đặc điểm kỹ thuật mặc định trên mức MPD có thể tiết kiệm được việc lặp lại không cần thiết nếu cùng phân mã hóa-giải mã cho một số hoặc tất cả các dạng biểu diễn được sử dụng.

MPDMoveBoxHeaderSizeDoesNotChange: Cờ chỉ ra nếu phần tiêu đề MoveBox thay đổi về kích thước giữa các tập tin riêng biệt trong toàn bộ phần trình diễn phương tiện. Cờ này có thể được sử dụng để tối ưu hóa việc tải về và chỉ có thể có mặt trong trường hợp các định dạng phân đoạn cụ thể, đặc biệt là đối với các phân đoạn chứa tiêu đề MOOV.

FileURIPattern: Mô hình được sử dụng bởi máy khách để tạo ra các thông báo yêu cầu đối với các tập tin trong phần trình diễn phương tiện. Các thuộc tính khác nhau cho phép tạo ra các URI duy nhất cho mỗi tập tin trong phần trình diễn phương tiện. URI cơ sở có thể là URI HTTP.

Dạng biểu diễn thay thế: Mô tả danh sách các dạng biểu diễn.

Phần tử AlternativeRepresentation:

Là phần tử XML mà gói gọn tất cả siêu dữ liệu cho một dạng biểu diễn. Phần tử AlternativeRepresentation có thể chứa các thuộc tính và các phần tử sau đây.

RepresentationID: ID duy nhất cho dạng biểu diễn thay thế cụ thể này trong phần trình diễn phương tiện.

FileInfoStatic: Cung cấp danh sách rõ ràng về thời gian bắt đầu và URI của tất cả các tập tin của phần trình diễn thay thế. Việc cung cấp tĩnh danh sách tập tin có thể mang lại lợi thế là mô tả chính xác thời gian của phần trình diễn phương tiện, nhưng có thể không nhỏ gọn, đặc biệt là nếu các dạng biểu diễn thay thế có chứa nhiều tập tin. Ngoài ra, các tên tập tin có thể có tên tùy ý.

FileInfoDynamic: Cung cấp cách ngầm định để xây dựng danh sách thời gian bắt đầu và URI của phần trình diễn thay thế. Việc cung cấp động danh sách tập tin có thể mang lại lợi thế là dạng biểu diễn nhỏ gọn hơn. Nếu chỉ trình tự thời gian bắt đầu được cung cấp, thì lợi thế thời gian cũng có ở đây, nhưng các tên tập tin sẽ được xây dựng tự động dựa trên FilePatternURI. Nếu chỉ có thời lượng của mỗi phân đoạn được cung cấp thì dạng biểu diễn là nhỏ gọn và có thể phù hợp để sử dụng trong dịch vụ trực tiếp, nhưng việc tạo ra các tập tin có thể được điều chỉnh theo định thời chung.

APMoveBoxHeaderSizeDoesNotChange: Cờ chỉ báo nếu phần tiêu đề MoveBox thay đổi về kích thước giữa các tập tin riêng biệt trong phần mô tả thay thế. Cờ này có thể được sử dụng để tối ưu hóa việc tải về và chỉ có thể có mặt trong trường

hợp định dạng phân đoạn cụ thể, đặc biệt là đối với các phân đoạn chứa tiêu đề MOOV.

APCodecDescription: Mô tả các phần mã hóa-giải mã chính của tập tin trong phần trình diễn thay thế.

Phần tử mô tả phương tiện

MediaDescription: Phần tử có thể gói gọn tất cả siêu dữ liệu cho phương tiện được chứa trong dạng biểu diễn này. Cụ thể nó có thể chứa thông tin về các rãnh trong phần trình diễn thay thế này cũng như các nhóm rãnh được đề xuất, nếu có. Các thuộc tính MediaDescription chứa các thuộc tính sau:

TrackDescription: Thuộc tính XML gói gọn tất cả siêu dữ liệu cho phương tiện được chứa trong dạng biểu diễn này. Thuộc tính TrackDescription chứa các thuộc tính sau:

TrackID: ID duy nhất cho rãnh trong dạng biểu diễn thay thế. Phần tử này có thể được sử dụng trong trường hợp rãnh là một phần của mô tả nhóm.

Bitrate: Tốc độ bit của rãnh.

TrackCodecDescription: Thuộc tính XML có chứa tất cả các thuộc tính trên phần mã hóa-giải mã được sử dụng trong rãnh này. Thuộc tính TrackCodecDescription chứa các thuộc tính sau:

MediaName: Thuộc tính xác định loại phương tiện. Loại phương tiện bao gồm "âm thanh", "video", "văn bản", "ứng dụng", và "thông báo".

Codec: loại mã hóa-giải mã bao gồm profile và mức độ.

LanguageTag: Thẻ ngôn ngữ nếu có.

MaxWidth, MaxHeight: Đối với video, chiều cao và rộng của video được chứa trong điểm ảnh.

SamplingRate: Cho âm thanh, tốc độ lấy mẫu.

GroupDescription: Thuộc tính cung cấp đề xuất đến máy khách về việc nhóm thích hợp dựa trên các tham số khác nhau.

GroupType: Loại mà dựa vào đó máy khách có thể quyết định làm thế nào để nhóm các rãnh.

Thông tin trong mô tả trình diễn phương tiện có lợi khi được sử dụng bởi máy khách truyền liên tục HTTP để thực hiện các yêu cầu cho tập tin/phân đoạn hoặc các

phần của chúng vào các thời điểm thích hợp, chọn phân đoạn từ các dạng biểu diễn đầy đủ phù hợp với khả năng, ví dụ về băng thông truy cập, khả năng hiển thị, khả năng mã hóa-giải mã, v.v. cũng như sở thích của người dùng như ngôn ngữ, v.v.. Hơn nữa, do mô tả trình diễn phương tiện mô tả các dạng biểu diễn có thời gian đồng chỉnh và được ánh xạ sang dòng thời gian chung, máy khách cũng có thể sử dụng thông tin trong MPD trong phân trình diễn phương tiện đang diễn ra để bắt đầu các hoạt động thích hợp để chuyển đổi giữa các dạng biểu diễn, để trình diễn các dạng biểu diễn một cách phối hợp hoặc để tìm kiếm trong khi trình diễn phương tiện.

Báo hiệu thời gian bắt đầu phân đoạn

Dạng biểu diễn có thể được chia, theo thời gian, thành nhiều phân đoạn. Vấn đề định thời giữa các rãnh tồn tại giữa đoạn cuối cùng của một phân đoạn và đoạn tiếp theo của phân đoạn tiếp theo. Ngoài ra, vấn đề định thời tồn tại trong trường hợp phân đoạn có thời lượng không đổi được sử dụng.

Việc sử dụng cùng thời lượng cho mỗi phân đoạn có thể có lợi thế là MPD nhỏ gọn và tĩnh. Tuy nhiên, mỗi phân đoạn vẫn có thể bắt đầu ở điểm truy cập ngẫu nhiên. Như vậy, hoặc là việc mã hóa video có thể được hạn chế để cung cấp các điểm truy cập ngẫu nhiên tại các điểm cụ thể, hoặc các thời lượng phân đoạn thực tế có thể không chính xác như được chỉ rõ trong MPD. Có thể mong muốn rằng hệ thống truyền liên tục không đặt ra hạn chế không cần thiết đối với quy trình mã hóa video và như vậy tùy chọn thứ hai có thể được ưu tiên.

Cụ thể, nếu thời lượng tập tin được chỉ rõ trong MPD là d giây, thì tập tin thứ n có thể bắt đầu tập với điểm truy cập ngẫu nhiên tại hoặc ngay sau thời điểm $(n - 1)d$.

Trong phương pháp này, mỗi tập tin có thể bao gồm thông tin về thời gian bắt đầu chính xác của phân đoạn xét về thời gian trình diễn chung. Ba cách có thể có để báo hiệu điều này bao gồm:

(1) Thứ nhất, hạn chế thời gian bắt đầu của mỗi phân đoạn đến thời gian chính xác được chỉ rõ trong MPD. Nhưng sau đó bộ mã hóa phương tiện có thể không có bất kỳ tính linh hoạt nào trong việc bố trí các khung IDR và có thể yêu cầu mã hóa cụ thể cho quy trình truyền liên tục tập tin.

(2) Thứ hai, thêm thời gian bắt đầu chính xác vào MPD cho từng phân đoạn. Đối với trường hợp theo nhu cầu, tính nhỏ gọn của MPD có thể giảm. Đối với trường hợp trực tiếp, điều này có thể yêu cầu cập nhật thường xuyên MPD, mà có thể làm giảm khả năng mở rộng.

(3) Thứ ba, thêm thời gian chung hoặc thời gian bắt đầu chính xác so với thời gian bắt đầu được công bố của dạng biểu diễn hoặc thời gian bắt đầu được công bố của phân đoạn trong MPD vào phân đoạn theo nghĩa là phân đoạn này có chứa thông tin đó. Thời gian này có thể được thêm vào một hộp mới dành riêng cho truyền liên tục thích ứng. Hộp này cũng có thể bao gồm thông tin được cung cấp bởi hộp "TIDX" hoặc "SIDX". Hệ quả của phương pháp thứ ba này là khi tìm đến một vị trí cụ thể ở gần chỗ bắt đầu của một trong các phân đoạn, máy khách có thể, dựa trên MPD, chọn phân đoạn tiếp theo phân đoạn có chứa điểm tìm kiếm yêu cầu. Phản hồi đơn giản trong trường hợp này có thể là di chuyển điểm tìm kiếm về phía trước đến chỗ bắt đầu của phân đoạn được lấy ra (ví dụ, đến điểm truy cập ngẫu nhiên tiếp theo sau điểm tìm kiếm). Thông thường, các điểm truy cập ngẫu nhiên được cung cấp ít nhất vài giây một lần (và thường có độ lợi mã hóa rất nhỏ từ việc làm cho chúng ít thường xuyên hơn) và do đó, trong trường hợp xấu nhất điểm tìm kiếm có thể được di chuyển một vài giây muộn hơn so với được chỉ rõ. Ngoài ra, máy khách có thể xác định khi lấy ra thông tin tiêu đề cho phân đoạn rằng điểm tìm kiếm yêu cầu trong thực tế ở trong phân đoạn trước đó và yêu cầu phân đoạn đó để thay thế. Điều này có thể dẫn đến sự gia tăng thường xuyên về thời gian cần thiết để thực hiện hoạt động tìm kiếm.

Danh sách phân đoạn truy cập được

Phân trình diễn phương tiện bao gồm tập hợp dạng biểu diễn mà mỗi dạng biểu diễn cung cấp một số phiên bản mã hóa khác nhau cho nội dung phương tiện ban đầu. Bản thân các dạng biểu diễn có lợi nếu chứa thông tin về các tham số khác biệt của dạng biểu diễn so với các tham số khác. Chúng cũng chứa, một cách rõ ràng hoặc ngầm định, danh sách phân đoạn truy cập được.

Phân đoạn có thể được phân biệt trong các phân đoạn phi thời gian chỉ chứa siêu dữ liệu và các phân đoạn phương tiện mà chủ yếu chứa dữ liệu phương tiện. Mô tả trình diễn phương tiện ("MPD") có lợi nếu ngẫu nhiên xác định và gán các thuộc tính khác nhau cho mỗi trong số phân đoạn này, một cách ngầm định hoặc rõ ràng. Có

lợi nếu các thuộc tính gán cho mỗi phân đoạn bao gồm khoảng thời gian trong đó phân đoạn có thể truy cập được, các tài nguyên và giao thức qua đó các phân đoạn có thể truy cập được. Ngoài ra, có lợi nếu các phân đoạn phương tiện được gán các thuộc tính như thời gian bắt đầu của phân đoạn trong phần trình diễn phương tiện, và thời lượng của phân đoạn trong phần trình diễn phương tiện.

Trong trường hợp trình diễn phương tiện là loại "theo nhu cầu", có lợi nếu được chỉ báo bởi một thuộc tính trong mô tả trình diễn phương tiện như `OnDemandAvailabilityInterval`, thì mô tả trình diễn phương tiện thường mô tả toàn bộ phân đoạn và cũng cung cấp chỉ báo khi các phân đoạn truy cập được và khi các phân đoạn không thể truy cập. Thời gian bắt đầu của các phân đoạn có lợi khi được thể hiện so với chỗ bắt đầu trình diễn phương tiện sao cho hai máy khách bắt đầu phát lại cùng phần trình diễn phương tiện, nhưng ở các thời điểm khác nhau, có thể sử dụng cùng mô tả trình diễn phương tiện cũng như cùng các phân đoạn phương tiện. Điều này cải thiện một cách có lợi khả năng lưu trữ đệm các phân đoạn.

Trong trường hợp trình diễn phương tiện là loại "trực tiếp", có lợi nếu được chỉ báo bởi một thuộc tính trong mô tả trình diễn phương tiện như thuộc tính `Service` (dịch vụ), thì các phân đoạn bao gồm phần trình diễn phương tiện vượt ra ngoài thời gian thực tế trong ngày thường không được tạo ra hoặc ít nhất là không thể truy cập được mặc dù các phân đoạn được mô tả đầy đủ trong MPD. Tuy nhiên, với chỉ báo rằng dịch vụ trình diễn phương tiện là loại "trực tiếp", máy khách có thể tạo ra danh sách phân đoạn có thể truy cập được cùng với thuộc tính định thời cho thời gian nội bộ máy khách NOW theo thời gian thực dựa trên thông tin chứa trong MPD và thời gian tải về của MPD. Có lợi nếu máy chủ hoạt động theo cách làm cho nguồn tài nguyên có thể truy cập được sao cho máy khách tham chiếu hoạt động với phiên bản của MPD tại thời gian thực NOW có thể truy cập được các tài nguyên này.

Cụ thể, máy khách tham chiếu tạo ra danh sách các phân đoạn có thể truy cập cùng với thuộc tính định thời cho thời gian nội bộ máy khách NOW theo thời gian thực dựa trên thông tin chứa trong MPD và thời gian tải về của MPD. Khi thời gian trôi đi, máy khách sẽ sử dụng cùng MPD và sẽ tạo ra danh sách phân đoạn có thể truy cập mới mà có thể được sử dụng để liên tục phát ra phần trình diễn phương tiện. Vì vậy, máy chủ có thể công bố các phân đoạn trong MPD trước khi các phân đoạn

thực sự có thể truy cập được. Điều này là có lợi, vì nó làm giảm việc cập nhật và tải về MPD thường xuyên.

Giả sử rằng danh sách các phân đoạn, mỗi phân đoạn có thời gian bắt đầu, tS , được mô tả một cách rõ ràng bởi danh sách phát trong các phần tử như `FileInfoStatic` hoặc một cách ngầm định bằng cách sử dụng phần tử như `FileInfoDynamic`. Phương pháp có lợi để tạo ra danh sách phân đoạn sử dụng `FileInfoDynamic` được mô tả dưới đây. Căn cứ vào quy tắc xây dựng này, máy khách có quyền truy cập vào danh sách các URI cho mỗi dạng biểu diễn, r , trong đây gọi là `FileURI( $r, i$ )`, và thời gian bắt đầu $tS(r, i)$ cho mỗi phân đoạn với chỉ mục i .

Việc sử dụng thông tin trong MPD để tạo ra cửa sổ thời gian truy cập được của các phân đoạn có thể được thực hiện bằng cách sử dụng quy tắc sau đây.

Đối với dịch vụ kiểu "theo nhu cầu", có lợi nếu được chỉ báo bởi thuộc tính như `Service` (dịch vụ), nếu thời gian thực hiện thời ở máy khách NOW nằm trong phạm vi sẵn có bất kỳ, có lợi nếu được thể hiện bởi phần tử MPD như `OnDemandAvailabilityInterval`, thì tất cả phân đoạn được mô tả của phần trình diễn theo nhu cầu này là có thể truy cập được. Nếu thời gian thực hiện thời ở máy khách NOW nằm ngoài phạm vi sẵn có bất kỳ, thì không phân đoạn nào trong số các phân đoạn được mô tả của phần trình diễn theo nhu cầu này là có thể truy cập được.

Đối với dịch vụ kiểu "trực tiếp", có lợi nếu được chỉ báo bởi thuộc tính như `Service`, thời gian bắt đầu $tS(r, i)$ có lợi nếu thể hiện thời gian sẵn có theo thời gian thực. Thời gian bắt đầu sẵn có có thể được suy ra là kết hợp của thời gian dịch vụ trực tiếp của sự kiện và thời gian quay vòng tại máy chủ để chụp hình, mã hóa, và xuất bản. Thời gian cho quy trình này có thể, ví dụ, được được chỉ rõ trong MPD, ví dụ bằng cách sử dụng khoảng bảo vệ an toàn tG được chỉ rõ, ví dụ, là `SafetyGuardIntervalLiveService` trong MPD. Điều này sẽ cung cấp sự khác biệt tối thiểu giữa thời gian UTC và sự sẵn có của dữ liệu trên máy chủ truyền liên tục HTTP. Theo phương án khác, MPD chỉ rõ một cách rõ ràng thời gian sẵn có của phân đoạn trong MPD mà không cung cấp thời gian quay vòng là hiệu số giữa thời gian trực tiếp của sự kiện và thời gian quay vòng. Trong phần mô tả sau đây, giả định rằng thời gian chung bất kỳ được được chỉ rõ là thời gian sẵn có. Người có hiểu biết trung bình trong lĩnh vực phát rộng phương tiện trực tiếp có thể suy ra thông tin này từ thông tin thích hợp trong phần mô tả trình diễn phương tiện sau khi đọc phần mô tả này.

Nếu thời gian thực hiện thời ở máy khách NOW nằm ngoài phạm vi bất kỳ của khoảng trình diễn trực tiếp, có lợi nếu được thể hiện bởi phần tử MPD như LivePresentationInterval, thì không phân đoạn nào trong số các phân đoạn được mô tả của phần trình diễn trực tiếp này có thể truy cập được. Nếu thời gian thực hiện thời ở máy khách NOW nằm trong khoảng trình diễn trực tiếp, thì ít nhất là các phân đoạn nhất định trong số các phân đoạn được mô tả của phần trình diễn trực tiếp này có thể truy cập được.

Giới hạn của các phân đoạn có thể truy cập được điều chỉnh bởi các giá trị sau:

Thời gian thực NOW (có sẵn cho máy khách).

Độ sâu bộ đệm dịch thời được cho phép tTSB, ví dụ được chỉ rõ là TimeShiftBufferDepth trong mô tả trình diễn phương tiện.

Máy khách tại thời gian sự kiện tương đối t_i chỉ có thể được phép yêu cầu các phân đoạn với thời gian bắt đầu $tS(r, i)$ trong khoảng giữa (NOW - tTSB) và NOW hoặc trong khoảng sao cho thời gian kết thúc của phân đoạn với thời lượng d cũng được bao gồm tạo ra khoảng giữa (NOW - tTSB - d) và NOW.

Cập nhật MPD

Theo một số phương án, máy chủ không biết trước tập tin hoặc thông tin định vị phân đoạn và thời gian bắt đầu của phân đoạn, bởi vì ví dụ như vị trí máy chủ sẽ thay đổi, hoặc phần trình diễn phương tiện bao gồm một số quảng cáo từ máy chủ khác, hoặc thời lượng trình diễn phương tiện không rõ, hoặc máy chủ muốn làm mờ thông tin định vị cho các phân đoạn theo sau.

Theo phương án này, máy chủ có thể chỉ mô tả các phân đoạn đã có thể truy cập được hoặc truy cập được ngay sau phiên bản này của MPD đã được xuất bản. Hơn nữa, theo một số phương án, có lợi nếu máy khách sử dụng phương tiện gần với phương tiện được mô tả trong MPD sao cho người dùng trải nghiệm chương trình phương tiện được chứa càng gần càng tốt với việc tạo ra nội dung phương tiện. Ngay khi máy khách dự đoán rằng nó đến chỗ kết thúc của phân đoạn phương tiện được mô tả trong MPD, có lợi nếu nó yêu cầu phiên bản mới của MPD để tiếp tục phát ra liên tục với kỳ vọng rằng máy chủ đã xuất bản MPD mới mô tả các phân đoạn phương tiện mới. Có lợi nếu máy chủ tạo ra các phiên bản mới của MPD và cập nhật MPD sao cho máy khách có thể dựa vào các thủ tục để cập nhật liên tục. Máy chủ có thể làm thích

ứng các thủ tục cập nhật MPD với việc tạo ra phân đoạn và có thể thực hiện xuất bản đến các thủ tục của máy khách tham chiếu mà hoạt động như máy khách thông thường.

Nếu phiên bản mới của MPD chỉ mô tả độ sớm thời gian ngắn, thì máy khách cần phải thường xuyên yêu cầu các phiên bản mới của MPD. Điều này có thể gây ra vấn đề về khả năng mở rộng và lưu lượng đường liên kết lên và liên kết xuống không cần thiết do yêu cầu thường xuyên không cần thiết.

Vì vậy, một mặt sẽ thích hợp nếu mô tả các phân đoạn càng nhiều càng tốt trong tương lai mà không cần phải làm cho chúng có thể truy cập được, và mặt khác cho phép các cập nhật không lường trước trong MPD thể hiện các vị trí máy chủ mới, cho phép chèn nội dung mới như quảng cáo hoặc để cung cấp thay đổi trong các tham số mã hóa-giải mã.

Hơn nữa, theo một số phương án, thời lượng của các phân đoạn phương tiện có thể nhỏ, chẳng hạn như, trong khoảng vài giây. Thời lượng của phân đoạn phương tiện có lợi nếu linh hoạt để điều chỉnh theo kích thước phân đoạn phù hợp mà có thể được tối ưu hóa theo các đặc tính phân phối hoặc đệm, để bù cho trễ đầu cuối trong các dịch vụ trực tiếp hoặc các khía cạnh khác xử lý việc lưu trữ hay phân phối phân đoạn, hoặc vì các lý do khác. Đặc biệt là trong trường hợp các phân đoạn là nhỏ so với thời gian trình diễn phương tiện, thì lượng đáng kể tài nguyên phân đoạn phương tiện và thời gian bắt đầu cần phải được mô tả trong mô tả trình diễn phương tiện. Kết quả là, kích thước của mô tả trình diễn phương tiện có thể lớn và có thể ảnh hưởng xấu đến thời gian tải về mô tả trình diễn phương tiện và do đó ảnh hưởng đến độ trễ khởi động của phân trình diễn phương tiện và đến việc sử dụng băng thông trên liên kết truy cập. Vì vậy, có lợi nếu không chỉ cho phép mô tả về danh sách phân đoạn phương tiện sử dụng danh sách phát mà còn cho phép mô tả bằng cách sử dụng các mẫu hoặc các quy tắc xây dựng URL. Mẫu và quy tắc xây dựng URL được sử dụng đồng nghĩa với nhau trong bản mô tả này.

Ngoài ra, mẫu có thể có lợi nếu được sử dụng để mô tả định vị phân đoạn trong các trường hợp trực tiếp vượt thời gian hiện thời. Trong trường hợp này, cập nhật của MPD bản thân nó không cần thiết do các phần tử định vị cũng như danh sách phân đoạn được mô tả bởi mẫu. Tuy nhiên, sự kiện bất ngờ vẫn có thể xảy ra đòi hỏi phải có thay đổi trong mô tả của các dạng biểu diễn hoặc phân đoạn. Các thay đổi về mô tả

trình diễn phương tiện truyền liên tục HTTP thích ứng có thể cần thiết khi nội dung từ nhiều nguồn khác nhau được ghép lại với nhau, ví dụ, khi quảng cáo đã được chèn vào. Nội dung từ các nguồn khác nhau có thể khác nhau theo nhiều cách khác nhau. Một lý do khác, trong phần trình diễn trực tiếp, là có thể là cần thay đổi các URL được sử dụng cho các tập tin nội dung để dự phòng cho khả năng gặp sự cố từ một máy chủ gốc trực tiếp đến máy chủ khác.

Theo một số phương án, có lợi nếu MPD được cập nhật, sau đó các bản cập nhật của MPD được thực hiện sao cho MPD cập nhật tương thích với MPD trước đó theo nghĩa là máy khách tham chiếu và bất kỳ máy khách cài đặt nào có thể tạo ra danh sách chức năng đồng nhất của các phân đoạn truy cập được từ MPD cập nhật bất cứ lúc nào cho đến thời gian hợp lệ của MPD trước đó do nó đã thực hiện từ phiên bản trước đó của MPD. Yêu cầu này đảm bảo rằng (a) máy khách ngay lập tức có thể bắt đầu sử dụng MPD mới mà không cần đồng bộ hóa với MPD cũ, vì nó tương thích với MPD cũ trước thời gian cập nhật; và (b) thời gian cập nhật không cần phải được đồng bộ với thời gian mà tại đó thay đổi thực tế đối với MPD xảy ra. Nói cách khác, bản cập nhật của MPD có thể được quảng cáo trước và máy chủ có thể thay thế phiên bản cũ của MPD khi thông tin mới có sẵn mà không cần phải duy trì các phiên bản khác của MPD.

Hai khả năng có thể tồn tại để định thời phương tiện trên bản cập nhật MPD cho tập hợp dạng biểu diễn hoặc tất cả các dạng biểu diễn. Hoặc là (a) dòng thời gian chung hiện có vẫn tiếp tục trên các bản cập nhật MPD (trong đây gọi là "cập nhật MPD liên tục"), hoặc (b) dòng thời gian hiện thời kết thúc và dòng thời gian mới bắt đầu với phân đoạn sau đoạn thay đổi (trong đây gọi là "cập nhật MPD gián đoạn").

Sự khác biệt giữa các tùy chọn này có thể được thấy rõ khi xem xét rằng các rãnh của đoạn phương tiện, và do đó là của phân đoạn, thường không bắt đầu và kết thúc tại cùng thời điểm vì mức chia nhỏ mẫu khác nhau trên các rãnh. Trong khi trình diễn bình thường, các mẫu của một rãnh của đoạn có thể được trình diễn *trước* một số mẫu của rãnh khác của đoạn trước đó, tức là có loại chồng lấn nào đó giữa các đoạn mặc dù có thể không có sự chồng lấn *trong một rãnh duy nhất*.

Sự khác biệt giữa (a) và (b) là liệu việc chồng lấn như vậy có thể được cho phép trên bản cập nhật MPD hay không. Khi cập nhật MPD là do việc ghép nối nội dung hoàn toàn riêng biệt, việc chồng lấn như vậy thường rất khó đạt được do nội dung mới

cần phần mã hóa mới để được ghép với nội dung trước đó. Do đó, có lợi nếu cung cấp khả năng cập nhật gián đoạn phần trình diễn phương tiện bằng cách khởi động lại dòng thời gian cho các phân đoạn nhất định và cũng có thể xác định một tập hợp dạng biểu diễn mới sau khi cập nhật. Ngoài ra, nếu nội dung đã được mã hóa một cách độc lập và được phân đoạn, thì cũng tránh được việc điều chỉnh dấu thời gian để phù hợp với dòng thời gian chung của mẫu nội dung trước đó.

Khi việc cập nhật là vì các lý do nhỏ hơn chẳng hạn như chỉ thêm các phân đoạn phương tiện mới vào danh sách phân đoạn phương tiện đã mô tả, hoặc nếu vị trí của các URL thay đổi, thì việc chồng lấn và cập nhật liên tục có thể được cho phép.

Trong trường hợp cập nhật MPD gián đoạn, dòng thời gian của phân đoạn cuối cùng của dạng biểu diễn trước đó kết thúc vào thời gian kết thúc trình diễn mới nhất của mẫu bất kỳ trong phân đoạn này. Dòng thời gian của dạng biểu diễn tiếp theo (hay chính xác hơn, thời gian trình diễn thứ nhất của phân đoạn phương tiện thứ nhất của phần mới của phần trình diễn phương tiện, còn được gọi là chu kỳ mới) thường và có lợi nếu bắt đầu ngay lúc kết thúc của phần trình diễn của chu kỳ cuối cùng sao cho đảm bảo được việc phát ra liên tục và liền mạch.

Hai trường hợp được minh họa trên Fig.11.

Tốt hơn và có lợi nếu hạn chế các cập nhật MPD ở các biên phân đoạn. Lý do cho việc hạn chế thay đổi hoặc cập nhật như vậy ở các biên phân đoạn là như sau. Thứ nhất, thay đổi đối với siêu dữ liệu nhị phân cho mỗi dạng biểu diễn, điển hình là phần đầu phim, có thể xảy ra ít nhất là ở các biên phân đoạn. Thứ hai, mô tả trình diễn phương tiện có thể chứa các con trỏ (URL) đến các phân đoạn. Theo nghĩa là MPD là cấu trúc dữ liệu "chiếc ô" nhóm lại với nhau tất cả các tập tin phân đoạn gắn với phần trình diễn phương tiện. Để duy trì mối quan hệ chứa này, mỗi phân đoạn có thể được tham chiếu bởi một MPD duy nhất và khi MPD được cập nhật, có lợi nếu cập nhật tại biên phân đoạn.

Biên phân đoạn thường không yêu cầu phải được đồng chỉnh, tuy nhiên đối với trường hợp nội dung được ghép nối từ các nguồn khác nhau, và với các cập nhật MPD gián đoạn nói chung, có hiệu quả khi đồng chỉnh các biên phân đoạn (cụ thể, phân đoạn cuối cùng của mỗi dạng biểu diễn có thể kết thúc tại cùng khung video và có thể không chứa các mẫu âm thanh với thời gian bắt đầu trình diễn muộn hơn thời gian trình diễn của khung đó). Sau đó cập nhật gián đoạn có thể bắt đầu một tập hợp dạng

biểu diễn mới tại thời điểm chung, được gọi là chu kỳ. Thời gian bắt đầu hợp lệ của tập hợp dạng biểu diễn mới này được cung cấp, ví dụ như thời gian bắt đầu chu kỳ. Thời gian bắt đầu tương đối của mỗi dạng biểu diễn được thiết lập lại về không và thời gian bắt đầu của chu kỳ đặt tập hợp dạng biểu diễn vào chu kỳ mới này trong dòng thời gian trình diễn phương tiện chung.

Đối với cập nhật MPD liên tục, biên phân đoạn không cần phải được liên kết. Mỗi phân đoạn của từng dạng biểu diễn thay thế có thể được điều chỉnh bởi mô tả trình diễn phương tiện và do đó yêu cầu cập nhật cho phiên bản mới của mô tả trình diễn phương tiện, thường được gây ra bởi dự đoán rằng không có phân đoạn phương tiện bổ sung nào được mô tả trong MPD hoạt động, có thể xảy ra tại các thời điểm khác nhau tùy thuộc vào tập hợp dạng biểu diễn được sử dụng bao gồm tập hợp dạng biểu diễn được dự kiến sẽ được sử dụng.

Để hỗ trợ cập nhật trong các phần tử MPD và các thuộc tính trong trường hợp tổng quát hơn, bất kỳ phần tử nào không chỉ dạng biểu diễn hoặc tập hợp dạng biểu diễn có thể được gắn với thời gian hợp lệ. Vì vậy, nếu các phần tử nhất định của MDP cần phải được cập nhật, ví dụ khi mà số lượng dạng biểu diễn được thay đổi hoặc các quy tắc xây dựng URL được thay đổi, thì các phần tử này có thể được cập nhật riêng biệt vào các thời điểm cụ thể, bằng cách cung cấp nhiều bản sao của phần tử này với thời gian hợp lệ riêng biệt.

Có lợi nếu tính hợp lệ được gắn với thời gian phương tiện chung, sao cho phần tử đã mô tả gắn với thời gian hợp lệ là hợp lệ trong chu kỳ của dòng thời gian chung của phần trình diễn phương tiện.

Như đã mô tả ở trên, theo một phương án, thời gian hợp lệ chỉ được thêm vào tập hợp dạng biểu diễn đầy đủ. Sau đó, mỗi tập hợp đầy đủ tạo thành một chu kỳ. Sau đó, thời gian hợp lệ tạo thành thời gian bắt đầu của chu kỳ này. Nói cách khác, trong trường hợp cụ thể sử dụng phần tử hợp lệ, tập hợp dạng biểu diễn đầy đủ có thể hợp lệ trong một chu kỳ thời gian, được chỉ báo bởi thời gian hợp lệ chung cho tập hợp dạng biểu diễn. Thời gian hợp lệ của tập hợp dạng biểu diễn được gọi là chu kỳ. Khi bắt đầu một chu kỳ mới, tính hợp lệ của tập hợp dạng biểu diễn trước đó hết hạn và các tập hợp dạng biểu diễn mới là hợp lệ. Lưu ý rằng, tốt hơn là, thời gian hợp lệ của các chu kỳ là riêng biệt.

Như đã mô tả ở trên, thay đổi đối với mô tả trình diễn phương tiện xảy ra tại biên phân đoạn, và như vậy đối với mỗi dạng biểu diễn, việc thay đổi một phần tử thực sự xảy ra tại biên phân đoạn tiếp theo. Sau đó, máy khách có thể tạo thành MPD hợp lệ bao gồm danh sách các phân đoạn cho từng thời điểm trong thời gian trình diễn của phương tiện.

Việc ghép nối khối gián đoạn có thể thích hợp trong trường hợp các khối chứa dữ liệu phương tiện từ các dạng biểu diễn khác nhau, hoặc từ các nội dung khác nhau, ví dụ từ phân đoạn nội dung và quảng cáo, hoặc trong các trường hợp khác. Có thể có yêu cầu trong hệ thống truyền liên tục yêu cầu theo khối rằng thay đổi đối với siêu dữ liệu trình diễn chỉ xảy ra tại biên khối. Điều này có thể có lợi vì các lý do cài đặt vì việc cập nhật các tham số bộ giải mã phương tiện trong khối có thể phức tạp hơn việc chỉ cập nhật chúng giữa các khối. Trong trường hợp này, có thể có lợi nếu chỉ rõ là các khoảng hợp lệ như đã mô tả ở trên có thể được hiểu là gần đúng, như vậy là một phần tử được coi là hợp lệ từ biên khối thứ nhất không sớm hơn so với chỗ bắt đầu của khoảng hợp lệ đã chỉ rõ đến biên khối thứ nhất không sớm hơn so với chỗ kết thúc khoảng hợp lệ đã chỉ rõ.

Phương án làm ví dụ trên đây mô tả cải tiến mới đối với hệ thống truyền liên tục yêu cầu theo khối được mô tả trong phần được trình diễn sau đây có tiêu đề là thay đổi đối với phần trình diễn phương tiện.

Báo hiệu thời lượng phân đoạn

Các cập nhật gián đoạn chia một cách có hiệu quả phần trình diễn thành một loạt các khoảng rời nhau, được gọi là chu kỳ. Mỗi chu kỳ có dòng thời gian riêng của nó để định thời mẫu phương tiện. Việc định thời phương tiện của dạng biểu diễn trong chu kỳ có lợi nếu có thể được chỉ ra bằng cách chỉ rõ danh sách nhỏ gọn riêng biệt gồm các thời lượng phân đoạn cho mỗi chu kỳ hoặc cho mỗi dạng biểu diễn trong chu kỳ.

Thuộc tính, ví dụ được gọi là thời gian bắt đầu chu kỳ, gắn với các phần tử trong MPD có thể chỉ rõ thời gian hợp lệ của một số phần tử nhất định trong thời gian trình diễn phương tiện. Thuộc tính này có thể được thêm vào các phần tử bất kỳ (các thuộc tính mà có thể được gán tính hợp lệ có thể được thay đổi cho các phần tử) của MPD.

Với cập nhật MPD gián đoạn, các phân đoạn của tất cả các dạng biểu diễn có thể kết thúc tại chỗ gián đoạn. Điều này thường có nghĩa ít nhất là phân đoạn cuối cùng trước chỗ gián đoạn có thời lượng khác với các phân đoạn trước đó. Việc báo hiệu thời lượng phân đoạn có thể bao gồm chỉ ra rằng tất cả phân đoạn có cùng thời lượng hoặc chỉ ra thời lượng riêng biệt cho mỗi phân đoạn. Có thể mong muốn có một dạng biểu diễn nhỏ gọn cho danh sách các thời lượng đoạn mà hiệu quả trong trường hợp nhiều phân đoạn có cùng thời lượng.

Có lợi nếu thời lượng của mỗi phân đoạn trong một dạng biểu diễn hay tập hợp dạng biểu diễn có thể được thực hiện với một chuỗi duy nhất mà chỉ rõ tất cả thời lượng phân đoạn trong một khoảng duy nhất từ khi bắt đầu cập nhật gián đoạn, tức là chỗ bắt đầu của chu kỳ cho đến khi phân đoạn phương tiện cuối cùng được mô tả trong MPD. Theo một phương án, định dạng của phần tử này là chuỗi văn bản phù hợp với luật sinh mà chứa danh sách các mục nhập thời lượng phân đoạn mà mỗi mục nhập chứa thuộc tính thời lượng *dur* và một số nhân tùy chọn *mult* của thuộc tính chỉ ra rằng dạng biểu diễn này chứa *< mult >* phân đoạn mục nhập thứ nhất có thời lượng *< dur >* của mục nhập thứ nhất, rồi đến *< mult >* phân đoạn mục nhập thứ hai có thời lượng *< dur >* của mục nhập thứ hai, v.v..

Mỗi mục nhập thời lượng chỉ rõ thời lượng của một hoặc nhiều phân đoạn. Nếu giá trị *< dur >* theo sau là ký tự "*" và một con số, thì con số này chỉ rõ số lượng phân đoạn liên tiếp có thời lượng này tính theo giây. Nếu không có dấu nhân "*", số lượng phân đoạn là một. Nếu có dấu "*" và không có con số nào theo sau, thì tất cả các phân đoạn tiếp theo có thời lượng được chỉ rõ và có thể không có thêm mục nhập nào trong danh sách. Ví dụ, chuỗi "30 *" có nghĩa là tất cả các phân đoạn có thời lượng là 30 giây. Chuỗi "30 * 12 10,5" chỉ ra rằng 12 phân đoạn có thời lượng 30 giây, tiếp theo là một phân đoạn có thời lượng 10,5 giây.

Nếu thời lượng phân đoạn được chỉ rõ riêng cho từng dạng biểu diễn thay thế, thì tổng thời lượng phân đoạn trong mỗi khoảng có thể giống nhau cho mỗi dạng biểu diễn. Trong trường hợp của rãnh video, khoảng này có thể kết thúc trên cùng khung trong mỗi dạng biểu diễn thay thế.

Người có hiểu biết trung bình trong lĩnh vực kỹ thuật, khi đọc sáng chế, có thể tìm thấy các cách tương tự và tương đương với để thể hiện thời lượng phân đoạn một cách nhỏ gọn.

Theo phương án khác, thời lượng của phân đoạn được báo hiệu là không đổi cho tất cả các phân đoạn trong dạng biểu diễn ngoại trừ phân đoạn cuối cùng bởi thuộc tính thời lượng tín hiệu <duration>. Thời lượng của phân đoạn cuối cùng trước bản cập nhật gián đoạn có thể ngắn hơn miễn là điểm bắt đầu của bản cập nhật gián đoạn tiếp theo hoặc chỗ bắt đầu của chu kỳ mới được cung cấp, thì điều này có nghĩa là thời lượng của phân đoạn cuối cùng đạt đến chỗ bắt đầu của chu kỳ tiếp theo.

Thay đổi và cập nhật đối với siêu dữ liệu biểu diễn

Việc chỉ ra các thay đổi của siêu dữ liệu biểu diễn được mã hóa nhị phân, chẳng hạn như, thay đổi tiêu đề phim "MOOV" có thể được thực hiện theo các cách khác nhau: (a) có thể có một hộp MOOV cho tất cả các dạng biểu diễn trong tập tin riêng biệt được tham chiếu trong MPD, (b) có thể có một hộp MOOV cho mỗi dạng biểu diễn thay thế trong tập tin riêng biệt được tham chiếu trong mỗi dạng biểu diễn thay thế, (c) mỗi đoạn có thể chứa hộp MOOV và do đó là độc lập, (d) có thể có hộp MOOV cho tất cả các dạng biểu diễn trong một tập tin 3GP cùng với MPD.

Lưu ý rằng trong trường hợp (a) và (b), có thể có lợi nếu một hộp 'MOOV' được kết hợp với khái niệm tính hợp lệ ở bên trên với nghĩa rằng nhiều hộp 'MOOV' hơn có thể được tham chiếu trong MPD miễn là tính hợp lệ của chúng riêng biệt nhau. Ví dụ, với việc xác định biên chu kỳ, tính hợp lệ của 'MOOV' trong chu kỳ cũ có thể hết hạn với sự bắt đầu của chu kỳ mới.

Trong trường hợp của tùy chọn (a), tham chiếu đến hộp MOOV duy nhất có thể được gán phần tử hợp lệ. Nhiều tiêu đề trình diễn có thể được cho phép, nhưng chỉ có một tiêu đề có thể hợp lệ tại một thời điểm. Theo phương án khác, thời gian hợp lệ của toàn bộ tập hợp dạng biểu diễn trong chu kỳ hoặc toàn bộ chu kỳ như được xác định trên đây có thể được sử dụng như là thời gian hợp lệ cho siêu dữ liệu biểu diễn này, mà thường được cung cấp như là tiêu đề MOOV.

Trong trường hợp của tùy chọn (b), tham chiếu đến hộp MOOV của từng dạng biểu diễn có thể được gán phần tử hợp lệ. Nhiều tiêu đề dạng biểu diễn có thể được cho phép, nhưng chỉ có một tiêu đề có thể hợp lệ tại một thời điểm. Theo phương án khác, thời gian hợp lệ của toàn bộ dạng biểu diễn hoặc toàn bộ chu kỳ như được xác định trên đây có thể được sử dụng như là thời gian hợp lệ cho siêu dữ liệu biểu diễn này, mà thường được cung cấp như là tiêu đề MOOV.

Trong trường hợp của tùy chọn (c), không thêm được báo hiệu trong MPD, nhưng việc báo hiệu bổ sung trong dòng phương tiện có thể được thêm vào để chỉ ra nếu hộp MOOV sẽ thay đổi cho phân đoạn bất kỳ sắp tới. Điều này được giải thích thêm ở dưới đây trong ngữ cảnh của "báo hiệu cập nhật trong siêu dữ liệu phân đoạn".

Báo hiệu cập nhật trong siêu dữ liệu phân đoạn

Để tránh cập nhật thường xuyên phần mô tả trình diễn phương tiện để biết được các cập nhật tiềm năng, có lợi nếu báo hiệu bất kỳ bản cập nhật nào cùng với các phân đoạn phương tiện. Có thể cung cấp phần tử hoặc các phần tử bổ sung trong bản thân phân đoạn phương tiện mà có thể chỉ ra rằng siêu dữ liệu cập nhật, chẳng hạn như, mô tả trình diễn phương tiện là có sẵn và có thể truy cập được trong một lượng thời gian nhất định để tiếp tục tạo ra thành công danh sách phân đoạn có thể truy cập. Ngoài ra, các phần tử này có thể cung cấp định danh tập tin, chẳng hạn như một URL, hoặc thông tin mà có thể được sử dụng để xây dựng định danh tập tin, cho tập tin siêu dữ liệu được cập nhật. Tập tin siêu dữ liệu được cập nhật có thể bao gồm siêu dữ liệu tương đương bằng với siêu dữ liệu được cung cấp trong tập tin siêu dữ liệu ban đầu cho phần trình diễn sửa đổi để chỉ ra khoảng hợp lệ cùng với siêu dữ liệu bổ sung mà cũng được kèm theo các khoảng hợp lệ. Chỉ báo này có thể được cung cấp trong phân đoạn phương tiện của tất cả các dạng biểu diễn có sẵn cho phần trình diễn phương tiện. Máy khách truy cập vào hệ thống truyền liên tục yêu cầu theo khối, khi phát hiện ra chỉ báo trong khối phương tiện, có thể sử dụng giao thức tải về tập tin hoặc phương tiện khác để lấy tập tin siêu dữ liệu được cập nhật. Nhờ đó, máy khách có thể được cung cấp thông tin về các thay đổi trong mô tả trình diễn phương tiện và thời gian mà tại đó chúng sẽ xảy ra hoặc đã xảy ra. Có lợi nếu mỗi máy khách yêu cầu mô tả trình diễn phương tiện cập nhật một lần duy nhất khi thay đổi đó xảy ra chứ không "hỏi vòng" và thu được tập tin nhiều lần cho các cập nhật hoặc thay đổi có thể có.

Các ví dụ về thay đổi bao gồm bổ sung hoặc loại bỏ các dạng biểu diễn, thay đổi đối với một hoặc nhiều dạng biểu diễn như thay đổi tốc độ bit, độ phân giải, tỷ lệ khung hình, các rãnh được bao gồm hoặc các tham số mã hóa-giải mã, và thay đổi đối với các quy tắc xây dựng URL, ví dụ như máy chủ gốc khác cho quảng cáo. Một số thay đổi có thể chỉ ảnh hưởng đến phân đoạn khởi tạo như đoạn tiêu đề phim

("MOOV") gắn với một dạng biểu diễn, trong khi các thay đổi khác có thể ảnh hưởng đến mô tả trình diễn phương tiện (MPD).

Trong trường hợp nội dung theo nhu cầu, các thay đổi này và thời gian của chúng có thể được biết trước và có thể được báo hiệu trong mô tả trình diễn phương tiện.

Đối với nội dung trực tiếp, thay đổi có thể không được biết đến cho đến thời điểm mà chúng xảy ra. Một giải pháp đó là cho phép mô tả trình diễn phương tiện có sẵn tại một URL cụ thể sẽ được cập nhật động và yêu cầu máy khách thường xuyên yêu cầu MPD này để phát hiện các thay đổi. Giải pháp này có nhược điểm về khả năng mở rộng (hiệu quả đệm và tải của máy chủ gốc). Trong một kịch bản với số lượng lớn khán giả, bộ nhớ đệm có thể thu được nhiều yêu cầu đối với MPD sau khi phiên bản trước đó đã hết hạn từ bộ nhớ đệm và trước khi phiên bản mới được nhận và tất cả các phiên bản này có thể được chuyển tiếp đến máy chủ gốc. Máy chủ gốc có thể cần phải liên tục xử lý các yêu cầu từ các bộ đệm cho mỗi phiên bản cập nhật của MPD. Ngoài ra, các bản cập nhật có thể không dễ dàng được đồng chỉnh thời gian phù hợp với các thay đổi trong phần trình diễn phương tiện.

Do một trong các lợi thế của kỹ thuật truyền liên tục HTTP là khả năng sử dụng cơ sở hạ tầng và dịch vụ web tiêu chuẩn để có khả năng mở rộng, nên giải pháp ưu tiên có thể chỉ liên quan đến tập tin "tĩnh" (tức là đưa vào bộ nhớ đệm được) và không dựa trên việc máy khách "hỏi vòng" tập tin để xem chúng đã thay đổi hay chưa.

Các giải pháp được mô tả và đề xuất để giải quyết việc cập nhật siêu dữ liệu bao gồm mô tả trình diễn phương tiện và siêu dữ liệu biểu diễn nhị phân chẳng hạn như các đoạn "MOOV" trong phần trình diễn phương tiện truyền liên tục HTTP thích ứng.

Đối với trường hợp nội dung trực tiếp, điểm mà tại đó các MPD hoặc "MOOV" có thể thay đổi có thể không được biết đến khi MPD được xây dựng. Do việc "hỏi vòng" thường xuyên MPD để kiểm tra các bản cập nhật thường nên được tránh, vì lý do băng thông và khả năng mở rộng, cập nhật đối với MPD có thể được chỉ rõ "trong dài băng" trong bản thân các tập tin phân đoạn, tức là, mỗi phân đoạn phương tiện có thể có tùy chọn để chỉ rõ các cập nhật. Tùy thuộc vào các định dạng phân đoạn từ (a) đến (c) trên đây, việc cập nhật khác nhau có thể được báo hiệu.

Nói chung, có lợi nếu chỉ báo sau đây có thể được cung cấp trong tín hiệu trong phân đoạn: chỉ báo rằng MPD có thể được cập nhật trước khi yêu cầu phân đoạn tiếp theo trong dạng biểu diễn này hoặc phân đoạn tiếp theo bất kỳ có thời gian bắt đầu lớn hơn thời gian bắt đầu của phân đoạn hiện thời. Việc cập nhật có thể được công bố trước chỉ ra rằng bản cập nhật chỉ cần xảy ra tại phân đoạn bất kỳ muộn hơn phân đoạn kế tiếp. Việc cập nhật MPD này cũng có thể được sử dụng để cập nhật siêu dữ liệu biểu diễn nhị phân như tiêu đề phim trong trường hợp phân tử định vị của phân đoạn phương tiện được thay đổi. Tín hiệu khác có thể chỉ ra rằng với việc hoàn thành phân đoạn này, không có thêm phân đoạn đứng trước về thời gian nên được yêu cầu.

Trong trường hợp các phân đoạn được định dạng theo định dạng phân đoạn (c), tức là, mỗi phân đoạn phương tiện có thể chứa siêu dữ liệu tự khởi tạo như tiêu đề phim, thì tín hiệu khác có thể được thêm vào chỉ ra rằng phân đoạn tiếp theo có chứa tiêu đề phim MOOV cập nhật. Có lợi nếu điều này cho phép tiêu đề phim được đưa vào phân đoạn, nhưng tiêu đề phim chỉ cần được yêu cầu bởi máy khách nếu phân đoạn trước đó chỉ báo cập nhật tiêu đề phim hoặc trong trường hợp tìm kiếm hoặc truy cập ngẫu nhiên khi chuyển đổi dạng biểu diễn. Trong các trường hợp khác, máy khách có thể đưa ra yêu cầu dãy byte cho phân đoạn mà không bao gồm tiêu đề phim từ việc tải về, do đó tiết kiệm băng thông một cách có lợi.

Theo phương án khác nữa, nếu chỉ báo cập nhật MPD được báo hiệu, thì tín hiệu cũng có thể chứa phân tử định vị như URL cho mô tả trình diễn phương tiện cập nhật. MPD cập nhật có thể mô tả phần trình diễn cả *trước và sau khi* cập nhật, sử dụng các thuộc tính hợp lệ như chu kỳ mới và cũ trong trường hợp cập nhật gián đoạn. Điều này có lợi nếu có thể được sử dụng để cho phép xem có dịch thời gian như sẽ được mô tả thêm dưới đây, và có lợi khi cho phép cập nhật MPD được báo hiệu tại thời điểm bất kỳ trước khi thay đổi chứa trong đó có hiệu lực. Máy khách có thể ngay lập tức tải về MPD mới và áp dụng nó cho phần trình diễn đang xảy ra.

Trong một cài đặt cụ thể, việc báo hiệu thay đổi bất kỳ cho mô tả trình diễn phương tiện, các tiêu đề MOOV hoặc chỗ kết thúc của phần trình diễn có thể được chứa trong hộp thông tin truyền liên tục được định dạng theo các quy tắc của định dạng phân đoạn sử dụng cấu trúc hộp của định dạng tập tin phương tiện cơ sở ISO. Hộp này có thể cung cấp tín hiệu cụ thể cho bản cập nhật bất kỳ trong số các bản cập nhật khác nhau nào.

Hộp thông tin truyền liên tục

Định nghĩa

Loại hộp: 'sinf'

Bộ chứa: Không

Bắt buộc: Không

Số lượng: Không hoặc một.

Hộp thông tin truyền liên tục chứa thông tin về phần trình diễn trực tuyến trong đó tập tin là một phần.

Cú pháp

```
aligned(8) class StreamingInformationBox
extends FullBox('sinf') {
    unsigned int(32) streaming_information_flags;
    /// Sau đây là các trường tùy chọn
    string mpd_location
}
```

Ngữ nghĩa

Streaming_information_flags chứa logic OR của không hoặc nhiều phần tử sau đây

0x00000001 Cập nhật tiêu đề phim như sau

0x00000002 Cập nhật phần mô tả trình diễn

0x00000004 Kết thúc trình diễn

mpd_location có mặt khi và chỉ khi cờ cập nhật mô tả trình diễn được thiết lập và cung cấp định vị tài nguyên thống nhất (URL) cho mô tả trình diễn phương tiện mới.

Trường hợp sử dụng làm ví dụ của các cập nhật MPD cho dịch vụ trực tiếp

Giả sử nhà cung cấp dịch vụ muốn cung cấp sự kiện bóng đá trực tiếp bằng cách sử dụng quy trình truyền liên tục yêu cầu theo khối cải tiến được mô tả ở đây. Có lẽ hàng triệu người dùng có thể muốn truy cập vào phần trình diễn của sự kiện này. Sự kiện trực tiếp thỉnh thoảng bị gián đoạn bởi các đoạn ngắt quãng hoặc khi thời gian hết được triệu gọi, hoặc quãng tạm lắng khác trong hành động, trong đó quảng cáo có thể

được thêm vào. Thông thường, không có hoặc có ít thông báo trước về thời gian chính xác của đoạn ngắt quãng này.

Nhà cung cấp dịch vụ có thể cần phải cung cấp cơ sở hạ tầng dư thừa (ví dụ, bộ mã hóa và máy chủ) để cho phép chuyển đổi liên mạch trong trường hợp bất kỳ thành phần nào gặp sự cố trong các sự kiện trực tiếp.

Giả sử người dùng, Anna, truy cập dịch vụ trên xe buýt bằng thiết bị di động của mình, và dịch vụ có sẵn ngay lập tức. Ngồi cạnh cô ấy là người dùng khác, Paul, xem sự kiện trên máy tính xách tay của anh ấy. Một bàn thắng được ghi và cả hai ăn mừng sự kiện này cùng một lúc. Paul nói với Anna rằng bàn thắng thứ nhất trong trận bóng thậm chí còn thú vị hơn và Anna sử dụng dịch vụ để cô ấy có thể xem sự kiện này 30 phút trước đó. Sau khi thấy bàn thắng, cô trở lại sự kiện trực tiếp.

Để giải quyết trường hợp sử dụng đó, nhà cung cấp dịch vụ có thể cập nhật MPD, báo hiệu cho máy khách rằng MPD cập nhật là có sẵn và cho phép máy khách truy cập dịch vụ truyền liên tục sao cho nó có thể trình diễn dữ liệu gần với thời gian thực.

Việc cập nhật MPD là khả thi theo cách không đồng bộ đối với phân phối các phân đoạn, như đã giải thích ở phần khác trong bản mô tả. Máy chủ có thể cung cấp đảm bảo cho bộ thu rằng MPD không được cập nhật lúc nào đó. Máy chủ có thể dựa vào MPD hiện thời. Tuy nhiên, không cần báo hiệu rõ ràng khi MPD được cập nhật trước chu kỳ cập nhật tối thiểu nào đó.

Việc phát ra đồng bộ hoàn toàn là khó có thể đạt được do máy khách có thể hoạt động trên các phiên bản cập nhật MPD khác nhau và do đó, máy khách có thể bị dịch đi. Bằng cách sử dụng cập nhật MPD, máy chủ có thể truyền tải các thay đổi và máy khách có thể được cảnh báo về các thay đổi đó, ngay cả trong khi đang trình diễn. Việc báo hiệu trong băng trên cơ sở từng phân đoạn có thể được sử dụng để chỉ rõ cập nhật của MPD, vì vậy các bản cập nhật có thể được giới hạn trong các biên phân đoạn, nhưng phải có thể được chấp nhận trong hầu hết các ứng dụng.

Phần tử MPD có thể được thêm vào để cung cấp thời gian xuất bản theo thời gian thực của MPD cũng như hộp cập nhật MPD tùy chọn được thêm vào ở phần đầu của các phân đoạn để báo hiệu rằng bản cập nhật MPD là bắt buộc. Các bản cập nhật này có thể được thực hiện theo thứ bậc, như với các MPD.

"Thời gian xuất bản" MPD cung cấp một định danh duy nhất cho MPD và thời điểm MPD được phát hành. Nó cũng cung cấp mốc neo cho các thủ tục cập nhật.

Hộp cập nhật MPD có thể được tìm thấy trong MPD sau hộp "styp", và được xác định bởi hộp Type = "mupe", không cần bộ chứa, không bắt buộc và có một số lượng số không hoặc một. Hộp cập nhật MPD chứa thông tin về trình diễn phương tiện trong đó phân đoạn này là một phần.

Cú pháp làm ví dụ như sau:

```
aligned(8) class MPDUpdateBox
extends FullBox('mupe') {
    unsigned int(3) mpd information flags;
    unsigned int(1) new-location flag;
    unsigned int(28) latest_mpd_update time;
    /// Sau đây là các trường tùy chọn
    string mpd_location
}
```

Ngữ nghĩa của các đối tượng khác nhau của lớp MPDUpdateBox có thể như sau:

x00 Cập nhật mô tả trình diễn phương tiện bây giờ

0x01 Cập nhật mô tả trình diễn phương tiện trước

0x02 Kết thúc trình diễn

0x03 - 0x07 Dành riêng

new_location_flag: nếu được thiết lập là 1, thì mô tả trình diễn phương tiện mới có sẵn tại vị trí mới được chỉ rõ tại mpd_location.

latest_mpd_update time: chỉ rõ thời gian (tính bằng ms) trước thời điểm cập nhật MPD cần thiết liên quan đến thời gian phát hành MPD của MPD mới nhất. Máy khách có thể chọn để cập nhật MPD bất cứ lúc nào tính từ bây giờ.

mpd_location: có mặt khi và chỉ khi new_location_flag được thiết lập và nếu như vậy, mpd_location cung cấp URL cho mô tả trình diễn phương tiện mới.

Nếu băng thông được sử dụng bởi các bản cập nhật là một vấn đề, máy chủ có thể cung cấp các MPD cho các khả năng nhất định của thiết bị sao cho chỉ có các phần này được cập nhật.

Xem có dịch thời gian và PVR mạng

Khi việc xem có dịch thời gian được hỗ trợ, có thể xảy ra khả năng là suốt thời gian phục vụ của phiên, hai hoặc nhiều MPD hoặc tiêu đề phim là hợp lệ. Trong trường hợp này bằng cách cập nhật MPD khi cần thiết, nhưng bổ sung thêm cơ chế hợp lệ hoặc khái niệm chu kỳ, MPD hợp lệ có thể tồn tại trong toàn bộ cửa sổ thời gian. Điều này có nghĩa là máy chủ có thể đảm bảo rằng MPD và tiêu đề phim bất kỳ được công bố cho khoảng thời gian bất kỳ trong cửa sổ thời gian hợp lệ để phục vụ xem có dịch thời gian. Nhiệm vụ của máy khách là đảm bảo rằng MPD và siêu dữ liệu có sẵn cho thời gian trình diễn hiện thời là hợp lệ. Việc chuyển một phiên trực tiếp sang một phiên PVR mạng chỉ sử dụng cập nhật MPD nhỏ cũng có thể được hỗ trợ.

Các phân đoạn phương tiện đặc biệt

Một vấn đề khi định dạng tập tin theo tiêu chuẩn ISO/IEC 14.496-12 được sử dụng trong hệ thống truyền liên tục yêu cầu theo khối đó là, như đã được mô tả trong phần trước, có thể có lợi nếu lưu trữ dữ liệu phương tiện cho một phiên bản duy nhất của phần trình diễn trong nhiều tập tin, được sắp xếp trong các phân đoạn thời gian liên tiếp. Hơn nữa, có thể có lợi nếu sắp xếp mỗi tập tin bắt đầu với điểm truy cập ngẫu nhiên. Hơn nữa, có thể có lợi nếu chọn vị trí của điểm tìm kiếm trong quá trình mã hóa video và phân đoạn phần trình diễn thành nhiều tập tin, mỗi tập tin bắt đầu với điểm tìm kiếm dựa trên việc chọn điểm tìm kiếm đã thực hiện trong quá trình mã hóa, trong đó mỗi điểm truy cập ngẫu nhiên có thể có hoặc có thể không được đặt ở đầu của tập tin mà trong đó mỗi tập tin bắt đầu với điểm truy cập ngẫu nhiên. Theo một phương án có các đặc điểm được mô tả ở trên, siêu dữ liệu trình diễn, hoặc mô tả trình diễn phương tiện, có thể chứa thời lượng chính xác của mỗi tập tin, trong đó thời lượng được lấy ra, ví dụ, chỉ ra sự khác nhau giữa thời gian bắt đầu của phương tiện video của tập tin và thời gian bắt đầu của phương tiện video của tập tin tiếp theo. Dựa trên thông tin này trong siêu dữ liệu trình diễn, máy khách có thể xây dựng ánh xạ giữa dòng thời gian chung cho phần trình diễn phương tiện và dòng thời gian cục bộ cho phương tiện trong mỗi tập tin.

Theo phương án khác, có lợi nếu kích thước của siêu dữ liệu trình diễn có thể được giảm bằng cách thay vào đó chỉ rõ rằng mỗi tập tin hoặc phân đoạn có cùng thời

lượng. Tuy nhiên trong trường hợp này và khi các tập tin phương tiện được xây dựng theo phương pháp trên, thời lượng của mỗi tập tin có thể không chính xác bằng với thời lượng được chỉ rõ trong mô tả trình diễn phương tiện bởi vì điểm truy cập ngẫu nhiên có thể không tồn tại vào thời điểm mà chính xác là thời lượng được chỉ rõ từ chỗ bắt đầu của tập tin.

Theo phương án khác, sáng chế đề xuất phương pháp hoạt động chính xác của hệ thống truyền liên tục yêu cầu theo khối mặc dù có sự khác biệt nêu trên. Trong phương pháp này, có thể đề xuất phần tử trong mỗi tập tin chỉ rõ việc ánh xạ của dòng thời gian cục bộ của phương tiện trong tập tin (mà có nghĩa là dòng thời gian bắt đầu từ dấu thời gian không (zero) dựa vào đó việc giải mã và dấu thời gian thành phần của mẫu phương tiện trong tập tin được chỉ rõ theo tiêu chuẩn ISO/IEC 14.496-12) đến dòng thời gian trình diễn chung. Thông tin ánh xạ này có thể bao gồm một dấu thời gian duy nhất trong thời gian trình diễn chung tương ứng với dấu thời gian không (zero) trong dòng thời gian tập tin cục bộ. Theo cách khác, thông tin ánh xạ có thể bao gồm giá trị độ dịch mà chỉ rõ hiệu số giữa thời gian trình diễn chung tương ứng với dấu thời gian không (zero) trong dòng thời gian tập tin cục bộ và thời gian trình diễn chung tương ứng với chỗ bắt đầu của tập tin theo thông tin được cung cấp trong siêu dữ liệu trình diễn.

Ví dụ về các hộp này có thể ví dụ là hộp thời gian giải mã đoạn rãnh ('tfdt') hoặc hộp điều chỉnh đoạn rãnh ('tfad') cùng với hộp điều chỉnh phương tiện đoạn rãnh ('tfma').

Máy khách làm ví dụ bao gồm khả năng tạo ra danh sách phân đoạn

Phân dưới đây mô tả máy khách làm ví dụ. Máy khách có thể được sử dụng như máy khách tham chiếu cho máy chủ để đảm bảo việc tạo ra và cập nhật MPD thích hợp.

Máy khách truyền liên tục HTTP được dẫn hướng bởi thông tin được cung cấp trong MPD. Giả sử rằng máy khách có quyền truy cập vào MPD mà nó nhận được vào thời gian T, tức là, thời gian nó đã có thể nhận được MPD thành công. Việc xác định nhận thành công có thể bao gồm việc máy khách thu nhận MPD cập nhật hoặc máy khách xác minh rằng MPD chưa được cập nhật do nhận thành công trước đó.

Hoạt động của máy khách làm ví dụ được giới thiệu ở đây. Để cung cấp dịch vụ truyền liên tục cho người dùng, đầu tiên máy khách phân tích cú pháp MPD và tạo ra danh sách các phân đoạn có thể truy cập cho từng dạng biểu diễn cho thời gian cục bộ của máy khách tại thời gian hệ thống hiện tại, có tính đến các thủ tục tạo ra danh sách phân đoạn như chi tiết dưới đây, có thể sử dụng danh sách phát hoặc sử dụng các quy tắc xây dựng URL. Sau đó, máy khách chọn một hoặc nhiều dạng biểu diễn dựa trên thông tin trong các thuộc tính biểu diễn và thông tin khác, ví dụ, băng thông có sẵn và khả năng của máy khách. Tùy thuộc vào việc nhóm, dạng biểu diễn có thể được trình diễn độc lập hoặc phối hợp với các dạng biểu diễn khác.

Đối với mỗi dạng biểu diễn, máy khách thu được siêu dữ liệu nhị phân như tiêu đề "MOOV" cho dạng biểu diễn, nếu có mặt, và các phân đoạn phương tiện của các dạng biểu diễn được chọn. Máy khách truy cập nội dung phương tiện bằng cách yêu cầu phân đoạn hoặc dãy byte của phân đoạn, có thể bằng cách sử dụng danh sách phân đoạn. Ban đầu, máy khách có thể đệm phương tiện trước khi bắt đầu phân trình diễn và, một khi phân trình diễn bắt đầu, máy khách vẫn tiếp tục sử dụng nội dung phương tiện bằng cách liên tục yêu cầu phân đoạn hoặc các phần phân đoạn có tính đến các thủ tục cập nhật MPD.

Máy khách có thể chuyển đổi các dạng biểu diễn có tính đến thông tin MPD cập nhật và/hoặc thông tin cập nhật từ môi trường của máy khách, ví dụ như, thay đổi băng thông có sẵn. Với bất kỳ yêu cầu nào đối với phân đoạn phương tiện có chứa điểm truy cập ngẫu nhiên, máy khách có thể chuyển sang dạng biểu diễn khác. Khi di chuyển về phía trước, tức là, tiếp diễn thời gian hệ thống hiện tại (được gọi là "thời gian NOW" để biểu diễn thời gian liên quan đến phân trình diễn), máy khách sử dụng các phân đoạn truy cập được. Với mỗi bước tiếp diễn thời gian NOW, máy khách có thể mở rộng danh sách phân đoạn có thể truy cập được cho từng dạng biểu diễn theo các thủ tục được chỉ rõ ở đây.

Nếu vẫn chưa đạt tới chỗ kết thúc của phân trình diễn phương tiện và nếu thời gian phát lại hiện thời nằm trong ngưỡng mà máy khách dự kiến sẽ sử dụng phương tiện của phương tiện được mô tả trong MPD cho dạng biểu diễn sử dụng hoặc sẽ được sử dụng bất kỳ, thì máy khách có thể yêu cầu bản cập nhật của MPD, với thời gian nhận của thời gian lấy phương tiện mới T. Sau khi thu được, máy khách sẽ tính đến MPD có thể được cập nhật và thời gian mới T trong khi tạo ra các danh sách phân

đoạn có thể truy cập được. Fig.29 minh họa một thủ tục cho các dịch vụ trực tiếp tại các thời điểm khác nhau ở máy khách.

Tạo ra danh sách phân đoạn truy cập được

Giả sử rằng máy khách truyền liên tục HTTP có quyền truy cập vào MPD và có thể muốn tạo ra danh sách phân đoạn truy cập được cho thời gian thực NOW. Máy khách được đồng bộ hóa với tham chiếu thời gian chung với độ chính xác nhất định, nhưng có lợi nếu không yêu cầu có sự đồng bộ trực tiếp với máy chủ truyền liên tục HTTP.

Tốt hơn là, danh sách phân đoạn truy cập được cho từng dạng biểu diễn được xác định là cặp danh sách của thời gian bắt đầu phân đoạn và phần tử định vị phân đoạn trong đó thời gian bắt đầu phân đoạn có thể được xác định là liên quan đến chỗ bắt đầu của dạng biểu diễn mà không bị mất tính tổng quát. Chỗ bắt đầu của dạng biểu diễn có thể được đồng chỉnh với chỗ bắt đầu của một chu kỳ nếu khái niệm này được áp dụng. Nếu không, chỗ bắt đầu của dạng biểu diễn có thể là chỗ bắt đầu của phần trình diễn phương tiện.

Máy khách sử dụng quy tắc xây dựng URL và định thời, ví dụ, như được xác định rõ hơn dưới đây. Khi danh sách phân đoạn đã mô tả được thu nhận, danh sách này còn được giới hạn ở danh sách truy cập được, mà có thể là tập con các phân đoạn của phần trình diễn phương tiện hoàn chỉnh. Việc xây dựng được điều chỉnh bởi giá trị hiện thời của đồng hồ theo thời gian NOW ở máy khách. Nói chung, phân đoạn chỉ có sẵn cho thời gian NOW bất kỳ trong tập hợp thời gian sẵn có. Đối với thời gian NOW bên ngoài cửa sổ này, không có phân đoạn nào có sẵn. Ngoài ra, đối với dịch vụ trực tiếp, giả sử thời gian kiểm tra nhất định cung cấp thông tin về tới chừng nào trong tương lai mà phương tiện được mô tả. Thời gian kiểm tra được xác định trên trục thời gian của phương tiện được ghi MPD; khi thời gian phát lại của máy khách đạt đến thời gian kiểm tra, có lợi nếu nó yêu cầu MPD mới. Khi thời gian phát lại của máy khách đạt đến thời gian kiểm tra, có lợi nếu nó yêu cầu MPD mới.

Sau đó, danh sách phân đoạn tiếp tục bị hạn chế bởi thời gian kiểm tra cùng với thuộc tính MPD `TimeShiftBufferDepth` sao cho chỉ các phân đoạn phương tiện có sẵn là phân đoạn mà tổng thời gian bắt đầu của phân đoạn phương tiện và thời gian bắt đầu biểu diễn nằm trong khoảng giữa NOW trừ đi `timeShiftBufferDepth` trừ đi thời lượng

của phân đoạn được mô tả cuối cùng và giá trị nhỏ hơn của một trong hai giá trị thời gian kiểm tra hoặc NOW.

Khởi có khả năng mở rộng

Đôi khi băng thông có sẵn giảm xuống thấp đến nỗi khối hay các khối hiện đang được thu tại bộ thu trở thành khó có thể được thu hoàn toàn kịp thời để được phát ra mà không cần tạm dừng phần trình diễn. Bộ thu có thể phát hiện trước các tình huống như vậy. Ví dụ, bộ thu có thể xác định rằng nó đang thu các khối mã hóa 5 đơn vị phương tiện cứ sau mỗi 6 đơn vị thời gian, và có bộ đệm cho 4 đơn vị phương tiện, để bộ thu có thể phải ngừng, hoặc tạm dừng, phần trình diễn, khoảng 24 đơn vị thời gian sau đó. Với thông báo đầy đủ, bộ thu có thể phản ứng với tình hình như vậy, ví dụ, loại bỏ dòng các khối hiện thời và bắt đầu yêu cầu khối hay các nhiều khối từ dạng biểu diễn khác của nội dung, chẳng hạn như, dạng biểu diễn sử dụng ít băng thông hơn trên mỗi đơn vị thời gian phát ra. Ví dụ, nếu bộ thu đã chuyển sang dạng biểu diễn trong đó các khối được mã hóa để có thêm ít nhất 20% thời gian video cho các khối có cùng kích thước, bộ thu có thể có thể loại bỏ sự cần thiết phải ngừng lại cho đến khi tình hình băng thông được cải thiện.

Tuy nhiên, có thể là lãng phí khi bộ thu hoàn toàn loại bỏ dữ liệu đã thu được từ dạng biểu diễn đã loại bỏ. Theo một phương án của hệ thống truyền liên tục yêu cầu theo khối được mô tả ở đây, dữ liệu trong mỗi khối có thể được mã hóa và sắp xếp theo cách mà các tiền tố nhất định của dữ liệu trong khối có thể được sử dụng để tiếp tục trình diễn mà không cần phần còn lại của khối đã được nhận. Ví dụ, kỹ thuật đã biết về mã hóa video có khả năng mở rộng có thể được sử dụng. Ví dụ về các phương pháp mã hóa video như vậy bao gồm phương pháp mã hóa có khả năng mở rộng H.264 (SVC) hoặc khả năng mở rộng thời gian của mã hóa video tiên tiến H.264 (AVC). Có lợi nếu phương pháp này cho phép trình diễn tiếp tục dựa trên phần của một khối đã được nhận ngay cả khi việc nhận khối hay các khối có thể bị loại bỏ, ví dụ như do các thay đổi trong băng thông có sẵn. Ưu điểm khác đó là một tập tin dữ liệu duy nhất có thể được sử dụng như là nguồn cho nhiều dạng biểu diễn khác nhau của nội dung. Điều này là có thể, ví dụ, bằng cách sử dụng các yêu cầu GET HTTP một phần mà chọn tập con của khối tương ứng với dạng biểu diễn được yêu cầu.

Một sự cải thiện được mô tả chi tiết ở đây là phân đoạn được cải tiến, ánh xạ phân đoạn có khả năng mở rộng. Ánh xạ phân đoạn có khả năng mở rộng bao gồm các vị trí của các lớp khác nhau trong phân đoạn sao cho máy khách có thể truy cập vào các phần của phân đoạn và trích xuất các lớp này. Theo phương án khác, dữ liệu phương tiện trong phân đoạn được xếp thứ tự sao cho chất lượng của phân đoạn này ngày càng tăng trong khi tải về dữ liệu dần dần từ đầu của phân đoạn này. Theo phương án khác, việc tăng dần chất lượng được áp dụng cho mỗi khối hoặc đoạn chứa trong phân đoạn, sao cho các yêu cầu đoạn có thể được hoàn thành để thực hiện phương pháp có khả năng mở rộng.

Fig.12 là hình vẽ thể hiện một khía cạnh của khối có khả năng mở rộng. Trên hình vẽ, bộ phát 1200 xuất ra siêu dữ liệu 1202, lớp có khả năng mở rộng 1 (1204), lớp có khả năng mở rộng 2 (1206), và lớp có khả năng mở rộng 3 (1208), với lớp sau bị trễ. Sau đó, bộ thu 1210 có thể sử dụng siêu dữ liệu 1202, lớp có khả năng mở rộng 1 (1204), và lớp có khả năng mở rộng 2 (1206) để trình diễn phần trình diễn phương tiện 1212.

Lớp có khả năng mở rộng độc lập

Như đã giải thích ở trên, không mong muốn hệ thống truyền liên tục yêu cầu theo khối phải ngừng lại khi bộ thu không thể thu được các khối đã yêu cầu của một dạng biểu diễn cụ thể của dữ liệu phương tiện kịp thời cho việc phát ra nó, vì điều đó thường tạo ra trải nghiệm người dùng kém. Việc ngừng lại này có thể tránh được, giảm bớt hoặc giảm nhẹ bằng cách hạn chế tốc độ dữ liệu của các dạng biểu diễn được chọn ít hơn nhiều so với băng thông có sẵn, do đó, rất ít khả năng là phần cho trước bất kỳ của phần trình diễn sẽ không được thu kịp thời, nhưng chiến lược này có nhược điểm là chất lượng phương tiện nhất thiết phải thấp hơn nhiều so với mức có thể được hỗ trợ theo nguyên tắc bởi băng thông có sẵn. Phần trình diễn chất lượng thấp hơn mức có thể cũng có thể được hiểu như là trải nghiệm người dùng kém. Vì vậy, các nhà thiết kế hệ thống truyền liên tục yêu cầu theo khối phải đối mặt với sự lựa chọn trong việc thiết kế các thủ tục của máy khách, lập trình máy khách hoặc cấu hình phần cứng, để yêu cầu một phiên bản nội dung mà có tốc độ dữ liệu thấp hơn nhiều so với băng thông có sẵn, trong trường hợp này người dùng có thể trải nghiệm phương tiện chất lượng kém, hoặc để yêu cầu một phiên bản nội dung mà có tốc độ dữ liệu gần với băng thông có

sẵn, trong trường hợp này người dùng có thể phải gặp khả năng cao xảy ra sự tạm dừng trình diễn khi có thay đổi băng thông có sẵn.

Để xử lý tình huống như vậy, hệ thống truyền liên tục yêu cầu theo khối được mô tả dưới đây có thể được tạo cấu hình để xử lý nhiều lớp có khả năng mở rộng một cách độc lập, sao cho bộ thu có thể đưa ra các yêu cầu theo lớp và bộ phát có thể phản hồi các yêu cầu theo lớp này.

Theo các phương án này, dữ liệu phương tiện được mã hóa cho mỗi khối có thể được chia thành nhiều phần rời nhau, trong đây gọi là các "lớp", sao cho sự kết hợp của các lớp bao gồm toàn bộ dữ liệu phương tiện cho khối và sao cho máy khách mà thu được các tập con nhất định của các lớp có thể thực hiện giải mã và trình diễn dạng biểu diễn của nội dung. Theo phương pháp này, việc sắp xếp thứ tự dữ liệu trong dòng được thực hiện sao cho các dải tiếp giáp tăng dần về chất lượng và siêu dữ liệu phản ánh điều này.

Một ví dụ về kỹ thuật có thể được sử dụng để tạo ra các lớp với các đặc tính nêu trên là kỹ thuật SVC, ví dụ như được mô tả trong các tiêu chuẩn ITU - T H.264/SVC. Ví dụ khác về kỹ thuật có thể được sử dụng để tạo ra các lớp với các đặc tính nêu trên là kỹ thuật của các lớp có khả năng mở rộng thời gian như được mô tả trong chuẩn ITU - T H.264/AVC.

Trong các phương án này, siêu dữ liệu có thể được cung cấp trong MPD hoặc trong bản thân phân đoạn, cho phép xây dựng các yêu cầu cho các lớp riêng biệt của khối cho trước bất kỳ và/hoặc kết hợp của các lớp và/hoặc lớp cho trước của nhiều khối và/hoặc kết hợp của các lớp của nhiều khối. Ví dụ, các lớp bao gồm một khối có thể được lưu trữ trong một tập tin và siêu dữ liệu có thể được cung cấp chỉ rõ cụ thể dãy byte trong tập tin tương ứng với các lớp riêng biệt.

Giao thức tải về tập tin có khả năng xác định dãy byte, ví dụ HTTP 1.1, có thể được sử dụng để yêu cầu các lớp riêng biệt hoặc nhiều lớp. Hơn nữa, như được hiểu rõ bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật khi xem sáng chế, các kỹ thuật được mô tả ở trên liên quan đến việc xây dựng, yêu cầu và tải về các khối có kích thước khác nhau và các dạng kết hợp khác nhau của các khối này cũng có thể được áp dụng trong ngữ cảnh này.

Các dạng kết hợp

Một số phương án sẽ được mô tả dưới đây, mà có thể có lợi khi được sử dụng bởi máy khách truyền liên tục yêu cầu theo khối để đạt được sự cải thiện trong trải nghiệm người dùng và/hoặc sự giảm yêu cầu về năng lực cơ sở hạ tầng phục vụ so với các kỹ thuật hiện có bằng cách sử dụng dữ liệu phương tiện được chia thành các lớp như đã mô tả ở trên.

Theo phương án thứ nhất, các kỹ thuật đã biết của hệ thống truyền liên tục yêu cầu theo khối có thể được áp dụng với việc sửa đổi mà trong một số trường hợp các phiên bản khác nhau của nội dung được thay thế bằng các kết hợp khác nhau của các lớp. Tức là khi hệ thống hiện có có thể cung cấp hai dạng biểu diễn khác nhau của nội dung, hệ thống cải tiến được mô tả ở đây có thể cung cấp hai lớp, trong đó một dạng biểu diễn của nội dung trong hệ thống hiện có là tương tự về tốc độ bit, chất lượng và có thể là các giá trị đo khác so với lớp thứ nhất trong hệ thống cải tiến và dạng biểu diễn thứ hai của nội dung trong hệ thống hiện có là tương tự về tốc độ bit, chất lượng và có thể là các giá trị đo khác so với kết hợp của hai lớp trong hệ thống cải tiến. Kết quả là khả năng lưu trữ cần thiết trong hệ thống cải tiến được giảm so đi với khả năng lưu trữ cần thiết trong hệ thống hiện có. Hơn nữa, trong khi máy khách của hệ thống hiện có có thể đưa ra yêu cầu đối với các khối của một dạng biểu diễn hoặc dạng biểu diễn khác, các máy khách của hệ thống cải tiến có thể đưa ra các yêu cầu đối với lớp thứ nhất hoặc cả hai lớp của khối. Kết quả là, trải nghiệm người dùng trong hai hệ thống là tương tự nhau. Hơn nữa, việc cải thiện tác vụ bộ nhớ đệm được cung cấp ngay cả đối với các phân đoạn chung có chất lượng khác nhau được sử dụng mà sau đó được lưu trữ đệm với khả năng cao hơn.

Theo phương án thứ hai, máy khách trong hệ thống truyền liên tục yêu cầu theo khối cải tiến sử dụng phương pháp theo lớp đang được mô tả có thể duy trì bộ đệm dữ liệu riêng biệt cho mỗi trong số một vài lớp mã hóa phương tiện. Như được hiểu rõ bởi người có hiểu biết trung bình trong lĩnh vực quản lý dữ liệu trong các thiết bị máy khách, các bộ đệm "riêng biệt" này có thể được cài đặt bằng cách phân bổ các vùng bộ nhớ riêng biệt về mặt vật lý hoặc logic cho các bộ đệm riêng biệt hoặc bằng các kỹ thuật khác, trong đó dữ liệu đệm được lưu trữ trong một hoặc nhiều vùng bộ nhớ và việc tách riêng dữ liệu từ các lớp khác nhau được thực hiện một cách logic thông qua việc sử dụng các cấu trúc dữ liệu có chứa tham chiếu đến các vị trí lưu trữ dữ liệu từ các lớp riêng biệt và như vậy trong phần dưới đây thuật ngữ "bộ đệm riêng biệt" nên

được hiểu là bao gồm phương pháp bất kỳ trong đó dữ liệu của các lớp khác nhau có thể được xác định một cách riêng biệt. Máy khách phát hành yêu cầu cho các lớp riêng của mỗi khối dựa trên việc chiếm mỗi bộ đệm, ví dụ, các lớp có thể được sắp xếp theo thứ tự ưu tiên sao cho yêu cầu dữ liệu từ một lớp có thể không được đưa ra nếu việc chiếm bộ đệm bất kỳ cho lớp thấp hơn theo thứ tự ưu tiên nằm ở dưới ngưỡng cho lớp thấp hơn đó. Trong phương pháp này, ưu tiên nhận dữ liệu từ các lớp thấp hơn theo thứ tự ưu tiên sao cho nếu băng thông có sẵn giảm xuống dưới mức yêu cầu để cũng nhận được các lớp cao hơn theo thứ tự ưu tiên thì chỉ có các lớp thấp hơn được yêu cầu. Hơn nữa, các ngưỡng gắn với các lớp khác nhau có thể khác nhau, sao cho lớp thấp hơn có ngưỡng cao hơn. Trong trường hợp băng thông có sẵn thay đổi sao cho dữ liệu cho lớp cao hơn không thể được nhận trước thời điểm phát ra của khối, thì dữ liệu cho các lớp thấp hơn cần phải đã nhận được và do đó phân trình diễn có thể tiếp tục với chỉ riêng các lớp thấp hơn. Ngưỡng chiếm bộ đệm có thể được xác định theo byte dữ liệu, thời lượng phát ra của dữ liệu chứa trong bộ đệm, số lượng khối hoặc bất kỳ biện pháp thích hợp nào khác.

Theo phương án thứ ba, các phương pháp trong phương án thứ nhất và thứ hai có thể được kết hợp sao cho có thể cung cấp nhiều dạng biểu diễn phương tiện mà mỗi dạng biểu diễn bao gồm tập con các lớp (như trong phương án thứ nhất) và sao cho phương án thứ hai được áp dụng cho tập con các lớp trong dạng biểu diễn.

Theo phương án thứ tư, các phương pháp trong phương án thứ nhất, thứ hai và/hoặc thứ ba có thể được kết hợp với phương án trong đó nhiều dạng biểu diễn độc lập của nội dung được cung cấp sao cho, ví dụ, ít nhất một trong các dạng biểu diễn độc lập bao gồm nhiều lớp mà các kỹ thuật trong các phương án thứ nhất, thứ hai và/hoặc thứ ba được áp dụng.

Bộ quản lý bộ đệm tiên tiến

Cùng với bộ giám sát bộ đệm 126 (xem Fig.2), bộ quản lý bộ đệm tiên tiến có thể được sử dụng để tối ưu hóa bộ đệm ở phía máy khách. Các hệ thống truyền liên tục yêu cầu theo khối muốn đảm bảo rằng việc phát ra phương tiện có thể bắt đầu một cách nhanh chóng và tiếp tục trôi chảy, đồng thời cung cấp chất lượng phương tiện tối đa cho người dùng hoặc cho thiết bị đích. Điều này có thể yêu cầu rằng máy khách yêu cầu các khối có chất lượng phương tiện cao nhất, nhưng điều đó cũng có thể được bắt

đầu một cách nhanh chóng và nhận được kịp thời sau đó để được phát ra mà không buộc phải tạm dừng trong phần trình diễn.

Trong các phương án sử dụng bộ quản lý bộ đệm tiến tiến, bộ quản lý xác định cần yêu cầu các khối dữ liệu phương tiện nào và khi nào thực hiện các yêu cầu đó. Bộ quản lý bộ đệm cao cấp có thể, ví dụ, được cung cấp một tập hợp siêu dữ liệu cho nội dung sẽ được trình diễn, siêu dữ liệu này bao gồm danh sách các dạng biểu diễn có sẵn cho nội dung và siêu dữ liệu cho mỗi dạng biểu diễn. Siêu dữ liệu cho dạng biểu diễn có thể bao gồm thông tin về tốc độ dữ liệu của dạng biểu diễn và các tham số khác, chẳng hạn như video, âm thanh hoặc các phần mã hóa-giải mã và các tham số mã hóa-giải mã, độ phân giải video, độ phức tạp giải mã, ngôn ngữ âm thanh và bất kỳ tham số nào khác có thể ảnh hưởng đến sự lựa chọn dạng biểu diễn tại máy khách.

Siêu dữ liệu cho dạng biểu diễn cũng có thể bao gồm định danh cho các khối mà các dạng biểu diễn đã được phân đoạn thành, các định danh này cung cấp thông tin cần thiết cho máy khách để yêu cầu một khối. Ví dụ, nếu giao thức yêu cầu là HTTP, định danh có thể là URL HTTP có thể cùng với thông tin bổ sung xác định dãy byte hoặc khoảng thời gian trong tập tin được xác định bởi URL này, dãy byte hoặc khoảng thời gian xác định khối cụ thể trong tập tin được xác định bởi URL đó.

Theo phương án cụ thể, bộ quản lý bộ đệm tiến tiến xác định khi nào bộ thu thực hiện yêu cầu đối với các khối mới và bản thân nó có thể xử lý việc gửi các yêu cầu này. Theo khía cạnh mới, bộ quản lý bộ đệm tiến tiến đưa ra yêu cầu đối với các khối mới theo giá trị của tỷ lệ cân bằng để cân bằng giữa việc sử dụng quá nhiều băng thông và việc hết phương tiện trong quy trình phát ra trực tuyến.

Thông tin nhận được bởi bộ giám sát bộ đệm 126 từ bộ đệm khối 125 có thể bao gồm các chỉ báo của từng sự kiện khi dữ liệu phương tiện được nhận, bao nhiêu dữ liệu đã được nhận, khi nào việc phát ra dữ liệu phương tiện bắt đầu hoặc dừng lại, và tốc độ phát ra phương tiện. Dựa trên thông tin này, bộ giám sát bộ đệm 126 có thể tính toán một biến số biểu diễn kích thước bộ đệm hiện thời, $B_{hiện\ thời}$. Trong ví dụ này, $B_{hiện\ thời}$ biểu diễn lượng phương tiện có trong bộ đệm hoặc các bộ đệm của máy khách hoặc thiết bị khác và có thể được đo bằng đơn vị thời gian để $B_{hiện\ thời}$ biểu diễn lượng thời gian mà nó sẽ cần để phát ra tất cả phương tiện được biểu diễn bởi các khối hoặc các phần khối được lưu trữ trong bộ đệm hoặc các bộ đệm nếu không có khối bổ sung hoặc một phần khối đã được nhận. Như vậy, $B_{hiện\ thời}$ biểu diễn "thời lượng phát ra", ở

tốc độ phát ra bình thường, của dữ liệu phương tiện có sẵn tại máy khách, nhưng chưa phát.

Khi thời gian trôi qua, giá trị của $B_{hiện\ thời}$ sẽ giảm khi phương tiện được phát ra và có thể tăng mỗi khi dữ liệu mới cho khối được nhận. Lưu ý rằng, nhằm mục đích giải thích, giả sử rằng một khối được nhận khi toàn bộ dữ liệu của khối có sẵn tại bộ yêu cầu khối 124, nhưng các phương pháp khác có thể được sử dụng thay thế, ví dụ, để tính đến việc nhận các phần khối. Trong thực tế, việc nhận một khối có thể diễn ra trong một khoảng thời gian.

Fig.13 minh họa biến thể của giá trị của $B_{hiện\ thời}$ theo thời gian, khi phương tiện được phát ra và các khối được nhận. Như được thể hiện trên Fig.13, giá trị của $B_{hiện\ thời}$ bằng không cho thời gian nhỏ hơn t_0 , cho thấy rằng không có dữ liệu đã được nhận. Tại t_0 , khối thứ nhất được nhận và giá trị của $B_{hiện\ thời}$ tăng lên để bằng với thời gian phát ra của khối nhận được này. Tại thời điểm này, việc phát ra chưa bắt đầu và như vậy giá trị của $B_{hiện\ thời}$ vẫn không đổi, cho đến khi thời gian t_1 , tại đó khối thứ hai đến và $B_{hiện\ thời}$ tăng thêm bởi kích thước của khối thứ hai này. Tại thời điểm này, việc phát ra bắt đầu và giá trị của $B_{hiện\ thời}$ bắt đầu giảm tuyến tính, cho đến thời gian t_2 , tại đó khối thứ ba đến.

Sự tiến triển của $B_{hiện\ thời}$ vẫn tiếp tục theo cách "răng cưa" này, tăng từng bước mỗi lần một khối được nhận (ở thời gian t_2, t_3, t_4, t_5 và t_6) và giảm trơn tru khi dữ liệu được phát ra ở giữa. Lưu ý rằng trong ví dụ này, việc phát ra tiến hành với tốc độ phát ra bình thường đối với nội dung và do đó độ dốc của đường cong giữa khi nhận khối chính xác là bằng -1, có nghĩa là một giây của dữ liệu phương tiện được phát cho mỗi giây của thời gian thực trôi qua. Với phương tiện dựa trên khung được phát ra tại số lượng khung nhất định trên mỗi giây, ví dụ, 24 khung mỗi giây, độ dốc -1 sẽ được tính xấp xỉ bằng các hàm bước nhỏ chỉ báo việc phát ra mỗi khung dữ liệu riêng biệt, ví dụ như, các bước bằng -1/24 của một giây khi một khung được phát ra.

Fig.14 thể hiện ví dụ khác về sự tiến triển của $B_{hiện\ thời}$ theo thời gian. Trong ví dụ này, khối thứ nhất đến t_0 và việc phát ra bắt đầu ngay lập tức. Việc khối đến và việc phát ra tiếp tục cho đến thời gian t_3 , tại đó giá trị của $B_{hiện\ thời}$ đạt đến không. Khi điều đó xảy ra, không có dữ liệu phương tiện nào còn có sẵn để phát ra, buộc phải tạm dừng trình diễn phương tiện. Tại thời gian t_4 , khối thứ tư được nhận và việc phát ra có thể bắt đầu lại. Do đó, ví dụ này thể hiện trường hợp trong đó việc nhận khối thứ tư muộn

hơn mong muốn, dẫn đến tạm dừng trong khi phát ra và do đó dẫn đến trải nghiệm người dùng kém. Như vậy, mục tiêu của bộ quản lý bộ đệm tiên tiến và các tính năng khác là để giảm xác suất xảy ra sự kiện này, đồng thời duy trì chất lượng phương tiện cao.

Bộ giám sát bộ đệm 126 cũng có thể tính toán số đo khác, $B_{tỷ lệ}(t)$, đó là tỷ lệ của phương tiện đã nhận được trong một khoảng thời gian nhất định so với chiều dài của khoảng thời gian. Cụ thể hơn, $B_{tỷ lệ}(t)$ bằng $T_{nhận được}/(T_{hiện tại} - t)$, trong đó $T_{nhận được}$ là lượng phương tiện (được đo bằng thời gian phát ra) nhận được trong khoảng thời gian từ t , thời gian sớm hơn so với thời điểm hiện thời cho đến thời điểm hiện thời, $T_{hiện tại}$.

$B_{tỷ lệ}(t)$ có thể được sử dụng để đo tốc độ thay đổi của $B_{hiện thời}$. $B_{tỷ lệ}(t) = 0$ là trường hợp không có dữ liệu được nhận kể từ thời gian t ; $B_{hiện thời}$ sẽ được giảm ($T_{hiện tại} - t$) kể từ thời điểm đó, giả sử phương tiện đang được phát ra. $B_{tỷ lệ}(t) = 1$ là trường hợp phương tiện được nhận với cùng lượng như nó đang được phát ra, cho thời gian ($T_{hiện tại} - t$); $B_{hiện thời}$ sẽ có giá trị tại thời gian $T_{hiện tại}$ giống như tại thời gian t . $B_{tỷ lệ}(t) > 1$ là trường hợp nhiều dữ liệu đã được nhận hơn mức cần thiết để phát ra cho thời gian ($T_{hiện tại} - t$); $B_{hiện thời}$ sẽ tăng lên từ thời gian t đến thời gian $T_{hiện tại}$.

Bộ giám sát bộ đệm 126 còn tính toán giá trị *Trạng thái*, mà có thể có các giá trị rời rạc. Bộ giám sát bộ đệm 126 còn được trang bị chức năng, **NewState**($B_{hiện thời}$, $B_{tỷ lệ}$), trong đó, cho trước giá trị hiện thời của $B_{hiện thời}$ và giá trị của $B_{tỷ lệ}$ cho $t < T_{hiện tại}$, cung cấp giá trị *Trạng thái* mới làm đầu ra. Bất cứ khi nào $B_{hiện thời}$ và $B_{tỷ lệ}$ làm cho hàm này trả về giá trị khác với giá trị hiện thời của *Trạng thái*, giá trị mới này được gán cho *Trạng thái* và giá trị *Trạng thái* mới được chỉ rõ cho bộ chọn khối 123.

Hàm **NewState** có thể được đánh giá có sự tham chiếu đến không gian của tất cả các giá trị có thể có của các cặp ($B_{hiện thời}$, $B_{tỷ lệ}(T_{hiện tại} - T_x)$) trong đó T_x có thể là giá trị (được tạo cấu hình) cố định, hoặc có thể được suy ra từ $B_{hiện thời}$, ví dụ như bởi bảng cấu hình mà ánh xạ từ các giá trị của $B_{hiện thời}$ sang các giá trị của T_x , hoặc có thể phụ thuộc vào giá trị trước đó của *Trạng thái*. Bộ giám sát bộ đệm 126 được cung cấp một hoặc nhiều phần chia của không gian này, trong đó mỗi phần chia bao gồm tập hợp các vùng rời nhau, mỗi vùng được chú thích với một giá trị *Trạng thái*. Sau đó, việc đánh giá hàm **NewState** bao gồm hoạt động xác định phần chia và xác định vùng mà cặp ($B_{hiện thời}$, $B_{tỷ lệ}(T_{hiện tại} - T_x)$) nằm trong đó. Sau đó, giá trị trả về là các chú

thích gắn với vùng đó. Trong trường hợp đơn giản, chỉ có một phần chia được cung cấp. Trong trường hợp phức tạp hơn, việc phân chia có thể phụ thuộc vào cặp $(B_{hiện\ thời}, B_{tỷ\ lệ}(T_{hiện\ tại} - T_x))$ tại thời điểm trước khi đánh giá hàm **NewState** hoặc phụ thuộc vào các yếu tố khác.

Theo phương án cụ thể, việc phân chia mô tả ở trên có thể dựa trên bảng cấu hình có chứa một số giá trị ngưỡng cho $B_{hiện\ thời}$ và một số giá trị ngưỡng cho $B_{tỷ\ lệ}$. Cụ thể, cho các giá trị ngưỡng cho $B_{hiện\ thời}$ là $B_{ngưỡng}(0) = 0, B_{ngưỡng}(1), \dots, B_{ngưỡng}(n_1), B_{ngưỡng}(n_1 + 1) = \infty$, trong đó n_1 là số giá trị ngưỡng khác 0 cho $B_{hiện\ thời}$. Cho giá trị ngưỡng cho $B_{tỷ\ lệ}$ là $B_{r-ngưỡng}(0) = 0, B_{r-ngưỡng}(1), \dots, B_{r-ngưỡng}(n_2), B_{r-ngưỡng}(n_2 + 1) = \infty$, trong đó n_2 là số giá trị ngưỡng cho $B_{tỷ\ lệ}$. Các giá trị ngưỡng này xác định phần chia bao gồm lưới $(n_1 + 1) \times (n_2 + 1)$ gồm các ô, trong đó ô thứ i của hàng thứ j tương ứng với vùng mà trong đó $B_{ngưỡng}(i - 1) \leq B_{hiện\ thời} < B_{ngưỡng}(i)$ và $B_{r-ngưỡng}(j - 1) \leq B_{tỷ\ lệ} < B_{r-ngưỡng}(j)$. Mỗi ô của lưới mô tả ở trên được chú thích một giá trị trạng thái, chẳng hạn như bằng cách gắn với các giá trị cụ thể được lưu trữ trong bộ nhớ, và sau đó hàm **NewState** trả về giá trị trạng thái gắn với ô được chỉ báo bởi các giá trị $B_{hiện\ thời}$ và $B_{tỷ\ lệ}(T_{hiện\ tại} - T_x)$.

Theo phương án khác, giá trị trẻ có thể được gắn với mỗi giá trị ngưỡng. Trong phương pháp cải tiến này, đánh giá của hàm **NewState** có thể dựa trên phần chia tạm thời được xây dựng bằng cách sử dụng tập hợp giá trị ngưỡng sửa đổi tạm thời như sau. Đối với mỗi giá trị ngưỡng $B_{hiện\ thời}$ nhỏ hơn dải $B_{hiện\ thời}$ tương ứng với ô được chọn trên đánh giá cuối cùng của **NewState**, giá trị ngưỡng được giảm bằng cách trừ đi giá trị trẻ gắn với ngưỡng đó. Đối với mỗi giá trị ngưỡng $B_{hiện\ thời}$ lớn hơn dải $B_{hiện\ thời}$ tương ứng với ô được chọn trên đánh giá cuối cùng của **NewState**, giá trị ngưỡng được tăng thêm bằng cách cộng thêm giá trị trẻ gắn với ngưỡng đó. Đối với mỗi giá trị ngưỡng $B_{tỷ\ lệ}$ nhỏ hơn dải $B_{tỷ\ lệ}$ tương ứng với ô được chọn dựa trên đánh giá cuối cùng của **NewState**, giá trị ngưỡng được giảm bằng cách trừ đi giá trị trẻ gắn với ngưỡng đó. Đối với mỗi giá trị ngưỡng $B_{tỷ\ lệ}$ lớn hơn dải $B_{tỷ\ lệ}$ tương ứng với ô đã chọn dựa trên đánh giá cuối cùng của **NewState**, giá trị ngưỡng được tăng thêm bằng cách cộng thêm giá trị trẻ gắn với ngưỡng đó. Các giá trị ngưỡng sửa đổi được sử dụng để đánh giá giá trị của **NewState** và sau đó các giá trị ngưỡng được trả về giá trị ban đầu của chúng.

Các cách khác để xác định các phần chia của không gian sẽ trở nên rõ ràng đối với người có hiểu biết trung bình trong lĩnh vực kỹ thuật khi đọc sáng chế. Ví dụ, phần

chia có thể được xác định bằng cách sử dụng các bất đẳng thức dựa trên các tổ hợp tuyến tính của $B_{tỷ lệ}$ và $B_{hiện thời}$, ví dụ như các ngưỡng bất đẳng thức tuyến tính có dạng $\alpha_1.B_{tỷ lệ} + \alpha_2.B_{hiện thời} \leq \alpha_0$ cho các giá trị thực $\alpha_0, \alpha_1, \alpha_2$, để xác định các nửa không gian trong không gian tổng thể và xác định mỗi tập rời nhau là giao của một số nửa không gian đó.

Mô tả ở trên minh họa cho quy trình cơ bản. Như được hiểu rõ bởi người có hiểu biết trung bình trong lĩnh vực lập trình thời gian thực khi xem sáng chế, việc triển khai hiệu quả có thể được thực hiện. Ví dụ, tại mỗi thời điểm mà thông tin mới được cung cấp đến bộ giám sát bộ đệm 126, có thể tính toán thời gian trong tương lai mà ở đó *NewState* sẽ chuyển sang giá trị mới nếu, ví dụ, không có thêm dữ liệu cho các khối được nhận. Sau đó bộ định thời được thiết lập cho thời điểm này và khi không có thêm dữ liệu nhập, sự kết thúc của bộ định thời này sẽ làm cho giá trị *Trạng thái* mới được gửi đến bộ chọn khối 123. Kết quả là, việc tính toán chỉ cần được thực hiện khi thông tin mới được cung cấp cho bộ giám sát bộ đệm 126 hoặc khi bộ định thời kết thúc, chứ không được thực hiện liên tục.

Các giá trị thích hợp của *Trạng thái* có thể là "thấp", "ổn định" và "đầy đủ". Ví dụ về tập hợp giá trị ngưỡng phù hợp và lưới ô kết quả được thể hiện trên Fig.15.

Trên Fig.15, các ngưỡng $B_{hiện thời}$ được thể hiện trên trục hoành theo mili giây, với các giá trị trễ được thể hiện phía dưới là "+/- giá trị". Ngưỡng $B_{tỷ lệ}$ được thể hiện trên trục tung tính theo từng nghìn (ví dụ, nhân 1000) với giá trị trễ được thể hiện phía dưới là "+/- giá trị". Các giá trị trạng thái được chú thích trong các ô lưới là "L", "S" và "F" tương ứng với "thấp", "ổn định" và "đầy đủ".

Bộ chọn khối 123 nhận thông báo từ bộ yêu cầu khối 124 bất cứ khi nào có cơ hội yêu cầu khối mới. Như đã mô tả ở trên, bộ chọn khối 123 được cung cấp thông tin về các khối có sẵn và siêu dữ liệu cho các khối này, ví dụ bao gồm thông tin về tốc độ dữ liệu phương tiện của mỗi khối.

Thông tin về tốc độ dữ liệu phương tiện của khối có thể bao gồm tốc độ dữ liệu phương tiện thực tế của khối cụ thể (tức là, kích thước khối theo byte chia cho thời gian phát ra tính theo giây), tốc độ dữ liệu phương tiện trung bình của dạng biểu diễn mà khối thuộc về hoặc số đo bằng thông cần thiết, trên cơ sở duy trì, để phát ra dạng biểu diễn mà khối thuộc về mà không cần tạm dừng, hoặc kết hợp của các thành phần trên đây.

Bộ chọn khối 123 chọn khối dựa trên giá trị *Trạng thái* cuối cùng được chỉ rõ bởi bộ giám sát bộ đệm 126. Khi giá trị trạng thái này là "ổn định", bộ chọn khối 123 chọn khối từ cùng dạng biểu diễn với khối đã được chọn trước đó. Khối được chọn là khối thứ nhất (theo thứ tự phát ra) có chứa dữ liệu phương tiện cho khoảng thời gian trong phân trình diễn mà không có dữ liệu phương tiện trước đó đã được yêu cầu.

Khi giá trị trạng thái là "thấp", bộ chọn khối 123 chọn khối từ dạng biểu diễn với tốc độ dữ liệu phương tiện thấp hơn so với khối đã được chọn trước đó. Một số yếu tố có thể ảnh hưởng đến việc chọn chính xác dạng biểu diễn trong trường hợp này. Ví dụ, bộ chọn khối 123 có thể được cung cấp chỉ báo về tốc độ tổng của dữ liệu đang đến và có thể chọn dạng biểu diễn với tốc độ dữ liệu phương tiện nhỏ hơn giá trị đó.

Khi giá trị trạng thái là "đầy đủ", bộ chọn khối 123 chọn khối từ dạng biểu diễn có tốc độ dữ liệu phương tiện cao hơn so với khối đã được chọn trước đó. Một số yếu tố có thể ảnh hưởng đến việc chọn chính xác dạng biểu diễn trong trường hợp này. Ví dụ, bộ chọn khối 123 có thể được cung cấp chỉ báo về tốc độ tổng của dữ liệu vào và có thể chọn dạng biểu diễn với tốc độ dữ liệu phương tiện không lớn hơn giá trị đó.

Một số yếu tố khác có thể ảnh hưởng đến hoạt động của bộ chọn khối 123. Cụ thể, tần suất mà tốc độ dữ liệu phương tiện của khối được chọn được tăng lên có thể bị hạn chế, ngay cả khi bộ giám sát bộ đệm 126 tiếp tục chỉ báo trạng thái "đầy đủ". Hơn nữa, có thể là bộ chọn khối 123 nhận được chỉ báo trạng thái "đầy đủ" nhưng không có khối phương tiện nào có tốc độ dữ liệu cao hơn có sẵn (ví dụ vì khối được chọn cuối cùng đã sẵn sàng cho tốc độ dữ liệu phương tiện có sẵn cao nhất). Trong trường hợp này, bộ chọn khối 123 có thể trì hoãn việc chọn khối tiếp theo với thời gian được chọn sao cho tổng lượng dữ liệu phương tiện được đệm trong bộ đệm khối 125 bị chặn trên.

Các yếu tố khác có thể ảnh hưởng đến tập hợp khối xem xét trong quy trình chọn. Ví dụ, các khối có sẵn có thể được giới hạn ở các khối từ các dạng biểu diễn có độ phân giải mã hóa nằm trong một phạm vi cụ thể được cung cấp cho bộ chọn khối 123.

Bộ chọn khối 123 cũng có thể nhận các dữ liệu nhập từ các thành phần khác mà có thể giám sát các khía cạnh khác của hệ thống, chẳng hạn như tính sẵn có của tài nguyên tính toán để giải mã phương tiện. Nếu nguồn tài nguyên này trở nên khan hiếm, bộ chọn khối 123 có thể chọn khối mà việc giải mã của khối được chỉ rõ là có độ

phức tạp tính toán thấp hơn trong siêu dữ liệu (ví dụ, các dạng biểu diễn với độ phân giải hoặc tốc độ khung thấp hơn nói chung là có độ phức tạp giải mã thấp hơn).

Phương án trên đây có ưu điểm đáng kể ở chỗ việc sử dụng giá trị $B_{tỷ lệ}$ trong việc đánh giá hàm *NewState* trong bộ giám sát bộ đệm 126 cho phép tăng nhanh hơn về chất lượng khi bắt đầu trình diễn so với phương pháp mà chỉ xem xét $B_{hiện thời}$. Khi không xem xét $B_{tỷ lệ}$, một lượng lớn dữ liệu đệm có thể được tích lũy trước khi hệ thống có thể chọn các khối với tốc độ dữ liệu phương tiện cao hơn và do đó có chất lượng cao hơn. Tuy nhiên, khi giá trị $B_{tỷ lệ}$ lớn, điều này chỉ ra rằng băng thông có sẵn cao hơn nhiều so với tốc độ dữ liệu phương tiện của các khối nhận được trước đó và ngay cả với dữ liệu được đệm tương đối ít (tức là giá trị thấp đối với $B_{hiện thời}$), vẫn an toàn khi yêu cầu theo khối có tốc độ dữ liệu phương tiện cao hơn và vì vậy có chất lượng cao hơn. Tương tự, nếu giá trị $B_{tỷ lệ}$ thấp (chẳng hạn như <1), điều này chỉ ra rằng băng thông có sẵn đã giảm xuống dưới tốc độ dữ liệu phương tiện của các khối được yêu cầu trước đó và do đó, ngay cả khi $B_{hiện thời}$ cao, hệ thống sẽ chuyển sang tốc độ dữ liệu phương tiện thấp hơn và do đó có chất lượng thấp hơn, ví dụ để tránh đạt đến điểm mà $B_{hiện thời} = 0$ và việc phát ra phương tiện bị dừng lại. Hoạt động cải thiện này có thể rất quan trọng trong môi trường mà điều kiện mạng và tốc độ phân phối có thể thay đổi một cách nhanh chóng và động, ví dụ, người dùng truyền liên tục đến các thiết bị di động.

Ưu điểm khác có được cung cấp nhờ sử dụng dữ liệu cấu hình để xác định sự phân chia của không gian của các giá trị ($B_{hiện thời}$, $B_{tỷ lệ}$). Dữ liệu cấu hình này có thể được cung cấp cho bộ giám sát bộ đệm 126 như là một phần của siêu dữ liệu trình diễn hoặc bằng phương tiện động khác. Do, trong triển khai thực tế, hành vi kết nối mạng của người dùng có thể rất khác nhau giữa các người dùng và theo thời gian đối với một người dùng, có thể rất khó để dự đoán được các cách phân chia mà sẽ hoạt động tốt cho tất cả người dùng. Khả năng cung cấp thông tin cấu hình như vậy cho người dùng tự động cho phép thiết lập cấu hình tốt được phát triển theo thời gian theo kinh nghiệm tích lũy được.

Định cỡ yêu cầu thay đổi

Có thể cần đến tần suất yêu cầu cao nếu mỗi yêu cầu là cho một khối duy nhất và nếu mỗi khối mã hóa cho một phân đoạn phương tiện ngắn. Nếu các khối phương

tiện ngắn, việc phát ra video di chuyển từ khối này sang khối khác một cách nhanh chóng, điều này cung cấp các cơ hội thường xuyên hơn cho bộ thu để điều chỉnh hoặc thay đổi tốc độ dữ liệu được chọn của mình bằng cách thay đổi dạng biểu diễn, cải thiện xác suất mà việc phát ra có thể tiếp tục mà không bị ngừng lại. Tuy nhiên, nhược điểm của tần suất yêu cầu cao là chúng có thể không được bền vững trên các mạng nhất định, mà trong đó băng thông có sẵn bị hạn chế trong mạng từ máy khách đến máy chủ, ví dụ, trong các mạng WAN không dây như mạng WAN 3G và 4G không dây, ở đó dung lượng của các liên kết dữ liệu từ máy khách đến mạng bị hạn chế hoặc có thể bị hạn chế trong khoảng thời gian ngắn hoặc dài do sự thay đổi về các điều kiện vô tuyến.

Tần suất yêu cầu cao cũng có nghĩa là tải cao trên cơ sở hạ tầng phục vụ, điều này dẫn đến các chi phí liên quan về yêu cầu dung lượng. Vì vậy, mong muốn là thu được một số lợi ích của tần suất yêu cầu cao mà không có tất cả các nhược điểm này.

Theo một số phương án của hệ thống truyền liên tục yêu cầu theo khối, sự linh hoạt của tần suất yêu cầu cao được kết hợp với các yêu cầu tần suất thấp hơn. Theo phương án này, các khối có thể được xây dựng như đã mô tả ở trên và được tổng hợp thành các phân đoạn có chứa nhiều khối, như đã được mô tả ở trên. Vào lúc bắt đầu của phần trình diễn, các quy trình mô tả ở trên, trong đó mỗi yêu cầu tham chiếu đến một khối duy nhất hoặc nhiều yêu cầu đồng thời được thực hiện để yêu cầu các phần của một khối được áp dụng để đảm bảo thời gian sửa đổi kênh nhanh chóng và do đó mang lại trải nghiệm người dùng tốt hơn khi bắt đầu phần trình diễn. Sau đó, khi một điều kiện nhất định, như được mô tả dưới đây, được đáp ứng, máy khách có thể đưa ra các yêu cầu bao gồm nhiều khối trong một yêu cầu duy nhất. Điều này là có thể bởi vì các khối đã được tổng hợp thành các tập tin hoặc các phân đoạn lớn hơn và có thể được yêu cầu bằng cách sử dụng dãy byte hoặc dải thời gian. Dãy byte hoặc dải thời gian liên tiếp này có thể được tổng hợp thành một dãy byte hoặc dải thời gian lớn hơn tạo ra một yêu cầu duy nhất cho nhiều khối, và có thể thậm chí là các khối không liên tục trong một yêu cầu.

Một cấu hình cơ bản có thể được tạo ra bằng cách quyết định xem yêu cầu một khối duy nhất (hoặc một phần khối) hay yêu cầu nhiều khối liên tiếp đó là có tạo cấu hình dựa trên quyết định về việc các khối được yêu cầu có thể được phát ra hay không. Ví dụ, nếu có khả năng là sẽ cần phải thay đổi sang dạng biểu diễn khác sớm, thì tốt

hơn là máy khách thực hiện các yêu cầu cho các khối duy nhất, tức là, một lượng nhỏ dữ liệu phương tiện. Một lý do để thực hiện điều này đó là nếu yêu cầu nhiều khối được thực hiện khi việc chuyển đổi sang dạng biểu diễn khác có thể sắp xảy ra thì việc chuyển đổi có thể được thực hiện trước khi một số khối cuối cùng của yêu cầu được phát ra. Do đó, việc tải về các khối cuối cùng này có thể làm trì hoãn việc phân phối dữ liệu phương tiện của dạng biểu diễn mà việc chuyển đổi được thực hiện, điều này có thể gây ra sự dừng phát ra phương tiện.

Tuy nhiên, yêu cầu các khối duy nhất không dẫn đến tần suất yêu cầu cao hơn. Mặt khác, nếu không chắc rằng sẽ cần phải thay đổi sang dạng biểu diễn khác sớm, thì có thể ưu tiên yêu cầu nhiều khối, do tất cả các khối đó có thể sẽ được phát ra, và kết quả là tần suất yêu cầu thấp hơn, điều này có thể làm giảm đáng kể chi phí yêu cầu, đặc biệt là khi thông thường không có thay đổi nào sắp xảy ra trong dạng biểu diễn.

Trong các hệ thống kết hợp khối thông thường, lượng được yêu cầu trong từng yêu cầu không được điều chỉnh động, tức là, thông thường mỗi yêu cầu là cho toàn bộ tập tin, hoặc mỗi yêu cầu là cho xấp xỉ cùng một lượng tập tin của dạng biểu diễn (đôi khi được đo theo thời gian, đôi khi tính theo byte). Vì vậy, nếu tất cả các yêu cầu nhỏ hơn, thì chi phí yêu cầu cao, trong khi đó nếu tất cả các yêu cầu lớn hơn, thì điều này làm tăng khả năng xảy ra sự kiện dừng phương tiện, và/hoặc cung cấp chất lượng phát ra phương tiện thấp hơn nếu dạng biểu diễn có chất lượng thấp hơn được chọn để tránh phải nhanh chóng thay đổi dạng biểu diễn khi điều kiện mạng thay đổi.

Ví dụ về điều kiện mà khi đáp ứng, có thể làm cho các yêu cầu tiếp theo tham chiếu đến nhiều khối, đó là ngưỡng của kích thước bộ đệm, $B_{hiện\ thời}$. Nếu $B_{hiện\ thời}$ dưới ngưỡng này, thì mỗi yêu cầu được phát hành tham chiếu đến một khối duy nhất. Nếu $B_{hiện\ thời}$ lớn hơn hoặc bằng ngưỡng này thì mỗi yêu cầu được phát hành tham chiếu đến nhiều khối. Nếu yêu cầu được phát hành có tham chiếu đến nhiều khối, thì số lượng khối được yêu cầu trong mỗi yêu cầu duy nhất có thể được xác định theo một trong một số cách có thể có. Ví dụ, số lượng này có thể là hằng số, ví dụ, hai. Theo cách khác, số lượng khối được yêu cầu trong một yêu cầu duy nhất có thể phụ thuộc vào trạng thái bộ đệm và cụ thể là phụ thuộc vào $B_{hiện\ thời}$. Ví dụ, một số ngưỡng có thể được thiết lập, với số lượng khối được yêu cầu trong một yêu cầu duy nhất được suy ra từ giá trị cao nhất trong số các ngưỡng nhỏ hơn $B_{hiện\ thời}$.

Ví dụ khác về điều kiện mà khi đáp ứng, có thể làm cho các yêu cầu tham chiếu đến nhiều khối, đó là giá trị biến *Trạng thái* được mô tả ở trên. Ví dụ, khi giá trị *Trạng thái* là "ổn định" hoặc "đầy đủ", thì các yêu cầu có thể được phát hành cho nhiều khối, nhưng khi *Trạng thái* là "thấp", thì tất cả các yêu cầu có thể là cho một khối.

Phương án khác được thể hiện trên Fig.16. Trong phương án này, khi yêu cầu tiếp theo được phát hành (được xác định trong bước 1300), giá trị *Trạng thái* hiện thời và $B_{hiện\ thời}$ được sử dụng để xác định kích thước của yêu cầu tiếp theo. Nếu giá trị *Trạng thái* hiện thời là "thấp" hoặc giá trị *Trạng thái* hiện thời là "đầy đủ" và dạng biểu diễn hiện thời không phải là cao nhất có sẵn (được xác định trong bước 1310, nhánh "Có"), thì yêu cầu tiếp theo được chọn là ngắn, ví dụ như chỉ cho khối tiếp theo (khối được xác định và yêu cầu được thực hiện trong bước 1320). Lý do là đây chính là các điều kiện mà trong đó có khả năng là sẽ sớm có một sự thay đổi dạng biểu diễn. Nếu giá trị *Trạng thái* hiện thời là "ổn định" hoặc giá trị *Trạng thái* hiện thời là "đầy đủ" và dạng biểu diễn hiện thời là cao nhất có sẵn (được xác định trong bước 1310, nhánh "Không"), thì thời lượng của các khối liên tiếp được yêu cầu trong yêu cầu tiếp theo được chọn tỷ lệ thuận với một phần phân số α của $B_{hiện\ thời}$ đối với một số cố định $\alpha < 1$ (các khối được xác định trong bước 1330, yêu cầu được thực hiện trong bước 1340), ví dụ như, với $\alpha = 0,4$, nếu $B_{hiện\ thời} = 5$ giây, thì yêu cầu tiếp theo có thể là xấp xỉ 2 giây của khối, trong khi đó nếu $B_{hiện\ thời} = 10$ giây, thì yêu cầu tiếp theo có thể là xấp xỉ 4 giây của khối. Lý do đó là trong các điều kiện này có thể là không có sự chuyển sang dạng biểu diễn mới sẽ được thực hiện trong khoảng thời gian tỷ lệ với $B_{hiện\ thời}$.

Tạo đường ống truyền linh hoạt

Hệ thống truyền liên tục khối có thể sử dụng giao thức yêu cầu tập tin có giao thức truyền tải cơ bản cụ thể, ví dụ như giao thức TCP/IP. Vào lúc bắt đầu của kết nối giao thức TCP/IP hoặc kết nối giao thức truyền tải khác, có thể mất thời gian đáng kể để đạt được khả năng sử dụng đầy đủ băng thông có sẵn. Điều này có thể dẫn đến "bồi thường khởi động kết nối" mỗi khi một kết nối mới được bắt đầu. Ví dụ, trong trường hợp giao thức TCP/IP, việc bồi thường khởi động kết nối xảy ra do cả thời gian cần cho giao thức bắt tay TCP ban đầu để thiết lập kết nối và thời gian cần cho giao thức kiểm soát tắc nghẽn để đạt được khả năng sử dụng đầy đủ băng thông có sẵn.

Trong trường hợp này, có thể được mong muốn phát hành nhiều yêu cầu bằng cách sử dụng một kết nối duy nhất, để giảm tần suất mà bồi thường khởi động kết nối phát sinh. Tuy nhiên, một số giao thức truyền tập tin, ví dụ như HTTP, không cung cấp cơ chế để hủy bỏ một yêu cầu, ngoài việc đóng hoàn toàn kết nối lớp truyền tải và do đó phải chịu bồi thường khởi động kết nối khi kết nối mới được thiết lập thế chỗ cho kết nối cũ. Yêu cầu đã phát hành có thể cần phải được hủy bỏ nếu xác định được là băng thông có sẵn đã thay đổi và tốc độ dữ liệu phương tiện khác được yêu cầu thay thế, ví dụ, có quyết định chuyển sang dạng biểu diễn khác. Một lý do khác để hủy bỏ yêu cầu đã phát hành có thể là nếu người dùng đã yêu cầu rằng phần trình diễn phương tiện kết thúc và phần trình diễn mới bắt đầu (có lẽ của cùng mục nội dung ở điểm khác trong phần trình diễn hoặc có lẽ của mục nội dung mới).

Như đã biết, bồi thường khởi động kết nối có thể tránh được bằng cách giữ kết nối mở và sử dụng lại cùng một kết nối cho các yêu cầu tiếp theo và cũng đã biết rằng kết nối có thể được giữ để sử dụng đầy đủ nếu nhiều yêu cầu được phát hành cùng một lúc trên cùng một kết nối (kỹ thuật được gọi là "tạo đường ống" trong ngữ cảnh HTTP). Tuy nhiên, nhược điểm của việc phát hành nhiều yêu cầu cùng một lúc, hay rộng hơn là theo cách mà nhiều yêu cầu được phát hành trước khi các yêu cầu trước đó đã hoàn thành trên kết nối, có thể là sau đó kết nối này được cam kết mang các phản hồi cho các yêu cầu đó và vì vậy nếu mong muốn thực hiện thay đổi đối với các yêu cầu được phát hành trở nên hấp dẫn thì kết nối có thể bị đóng nếu cần thiết phải hủy bỏ yêu cầu đã phát hành mà không còn mong muốn nữa.

Xác suất một yêu cầu đã phát hành cần phải được hủy bỏ có thể một phần phụ thuộc vào thời lượng của khoảng thời gian giữa khi phát hành các yêu cầu và thời gian phát ra của khối được yêu cầu theo nghĩa là khi khoảng thời gian này cao thì xác suất mà yêu cầu phát hành cần phải được hủy bỏ cũng cao (vì có khả năng là băng thông có sẵn thay đổi trong khoảng thời gian đó).

Như đã biết, một số giao thức tải về tập tin có đặc tính là có lợi nếu một kết nối lớp truyền tải cơ bản duy nhất có thể được sử dụng cho nhiều yêu cầu tải về. Ví dụ, HTTP có đặc tính này, do việc tái sử dụng một kết nối duy nhất cho nhiều yêu cầu tránh được "bồi thường khởi động kết nối" mô tả ở trên cho các yêu cầu không phải yêu cầu đầu tiên. Tuy nhiên, nhược điểm của phương pháp này là kết nối được cam kết sẽ truyền tải dữ liệu được yêu cầu trong mỗi yêu cầu đã phát hành và do đó nếu yêu

cầu hoặc các yêu cầu cần phải được hủy bỏ sau đó, thì kết nối có thể bị đóng, chịu bởi thường khởi động kết nối khi kết nối thay thế được thiết lập, hoặc máy khách có thể chờ đợi để nhận dữ liệu mà không còn cần thiết nữa, phải chịu độ trễ khi nhận dữ liệu tiếp theo.

Phần dưới đây mô tả phương án mà vẫn giữ được các ưu điểm của việc tái sử dụng kết nối mà không chịu các nhược điểm này và đó cũng cải thiện thêm tần số mà các kết nối có thể được tái sử dụng.

Các phương án của hệ thống truyền liên tục khối mô tả trong đây được tạo cấu hình để tái sử dụng một kết nối cho nhiều yêu cầu mà không cần phải cam kết kết nối khi bắt đầu đối với các tập hợp yêu cầu. Về cơ bản, một yêu cầu mới được phát hành trên một kết nối hiện thời khi các yêu cầu đã phát hành trên kết nối này chưa hoàn thành, nhưng gần hoàn thành. Một lý do không phải chờ đợi cho đến khi các yêu cầu hiện thời hoàn thành đó là nếu yêu cầu trước đó đã hoàn thành, thì tốc độ kết nối có thể làm suy giảm, tức là, phiên TCP cơ bản sau đó có thể đi vào trạng thái rồi, hoặc biến TCP *CWnd* có thể được giảm đáng kể, do đó làm giảm đáng kể tốc độ tải về ban đầu của yêu cầu mới được phát hành trên kết nối đó. Một lý do để chờ đợi cho đến khi gần hoàn thành trước khi phát hành yêu cầu bổ sung là bởi vì nếu yêu cầu mới được phát hành lâu trước khi các yêu cầu trước đó hoàn thành, thì yêu cầu mới phát hành có thể không bắt đầu trong một khoảng thời gian đáng kể, và đây có thể là trường hợp mà trong khoảng thời gian này trước khi yêu cầu mới phát hành bắt đầu quyết định làm cho yêu cầu mới không còn hợp lệ nữa, ví dụ, do quyết định chuyển đổi dạng biểu diễn. Như vậy, phương án của máy khách thực hiện kỹ thuật này sẽ phát hành yêu cầu mới trên kết nối càng muộn càng tốt mà không làm chậm khả năng tải về của kết nối.

Phương pháp này bao gồm bước giám sát số lượng byte nhận được trên kết nối để phản hồi yêu cầu mới nhất được phát hành trên kết nối này và áp dụng một thử nghiệm cho số lượng này. Điều này có thể được thực hiện bằng cách để bộ thu (hoặc bộ phát, nếu phù hợp) được tạo cấu hình để giám sát và thử nghiệm.

Nếu thử nghiệm đạt, thì yêu cầu khác nữa có thể được phát hành trên kết nối. Một ví dụ về thử nghiệm phù hợp là xem liệu số byte nhận được có lớn hơn phần cố định của kích thước dữ liệu được yêu cầu. Ví dụ, phần này có thể là 80%. Một ví dụ khác về thử nghiệm phù hợp là dựa trên các tính toán sau đây, như được minh họa trên Fig.17. Trong tính toán này, cho R là đánh giá tốc độ dữ liệu của kết nối, T là đánh giá

thời gian trọn vòng (RTT – round trip time) và X là hệ số mà, ví dụ, có thể là hằng số được thiết lập là giá trị từ 0,5 đến 2, trong đó các đánh giá của R và T được cập nhật một cách thường xuyên (được cập nhật trong bước 1410). Cho S là kích thước của dữ liệu được yêu cầu trong yêu cầu cuối cùng, B là số byte của dữ liệu được yêu cầu nhận được (được tính theo bước 1420).

Thử nghiệm phù hợp sẽ là để bộ thu (hoặc bộ phát, nếu phù hợp) thực hiện thường trình để đánh giá bất đẳng thức $(S - B) < X.R.T$ (được thử nghiệm trong bước 1430), và nếu "Có" thì thực hiện hành động. Ví dụ, thử nghiệm có thể được thực hiện để xem liệu có yêu cầu khác sẵn sàng được phát hành trên kết nối này hay không (được thử nghiệm trong bước 1440), và nếu "Có" thì phát hành yêu cầu đến kết nối (bước 1450) và nếu "Không" thì quy trình trở lại bước 1410 để tiếp tục cập nhật và thử nghiệm. Nếu kết quả thử nghiệm trong bước 1430 là "Không" thì quy trình quay trở lại bước 1410 để tiếp tục cập nhật và thử nghiệm.

Thử nghiệm bất đẳng thức trong bước 1430 (được thực hiện bởi phần tử được lập trình thích hợp chẳng hạn) làm cho mỗi yêu cầu tiếp theo được phát hành khi lượng dữ liệu còn lại sẽ được nhận bằng X lần lượng dữ liệu có thể được nhận tại tốc độ nhận được đánh giá hiện thời trong một RTT. Một số phương pháp để đánh giá tốc độ dữ liệu, R , trong bước 1410 đã được biết đến trong lĩnh vực. Ví dụ, tốc độ dữ liệu có thể được đánh giá là Dt/t , trong đó Dt là số bit nhận được trong t giây trước và trong đó t có thể, ví dụ, là 1 hoặc 0,5s hoặc một khoảng thời gian khác. Phương pháp khác là tính trung bình lũy thừa có trọng số, hoặc bộ lọc IIR (đáp ứng xung vô hạn) của tốc độ dữ liệu đến. Một số phương pháp để đánh giá RTT, T , trong bước 1410 đã được biết đến trong lĩnh vực.

Thử nghiệm ở bước 1430 có thể được áp dụng cho tổng hợp của tất cả các kết nối hoạt động trên giao diện, như sẽ được giải thích chi tiết dưới đây.

Phương pháp này còn bao gồm bước xây dựng danh sách yêu cầu ứng viên, gắn từng yêu cầu ứng viên với tập hợp máy chủ phù hợp mà với nó yêu cầu có thể được thực hiện và sắp xếp thứ tự danh sách yêu cầu ứng viên theo thứ tự ưu tiên. Một số mục nhập trong danh sách yêu cầu ứng viên có thể có mức ưu tiên giống nhau. Máy chủ trong danh sách máy chủ phù hợp gắn với mỗi yêu cầu ứng viên được xác định bởi tên máy chủ. Mỗi tên máy chủ tương ứng với tập hợp địa chỉ giao thức Internet có thể được thu nhận từ hệ thống tên miền như đã biết. Vì vậy mỗi yêu cầu có thể có trong

danh sách yêu cầu ứng viên được gắn với tập hợp địa chỉ giao thức Internet, cụ thể là sự kết hợp của các tập hợp địa chỉ giao thức Internet gắn với các tên máy chủ gắn với máy chủ gắn với yêu cầu ứng viên đó. Bất cứ khi nào thử nghiệm được mô tả trong bước 1430 được đáp ứng cho kết nối, và không có yêu cầu mới được phát hành trên kết nối đó, yêu cầu có mức ưu tiên cao nhất trong danh sách yêu cầu ứng viên mà gắn với địa chỉ giao thức Internet của đích kết nối được chọn và yêu cầu này được phát hành trên kết nối. Yêu cầu này cũng được loại bỏ khỏi danh sách yêu cầu ứng viên.

Các yêu cầu ứng viên có thể được loại bỏ (hủy bỏ) khỏi danh sách yêu cầu ứng viên, các yêu cầu mới có thể được thêm vào danh sách ứng viên với mức ưu tiên cao hơn so với yêu cầu hiện có trong danh sách ứng viên, và các yêu cầu hiện có trong danh sách ứng viên có thể có mức ưu tiên được thay đổi. Bản chất động của các yêu cầu trong danh sách yêu cầu ứng viên, và bản chất động của mức ưu tiên của chúng trong danh sách ứng viên, có thể thay đổi các yêu cầu mà có thể được phát hành tiếp theo tùy thuộc vào khi nào thử nghiệm thuộc loại được mô tả trong bước 1430 được đáp ứng.

Ví dụ, có khả năng là nếu câu trả lời cho thử nghiệm được mô tả trong bước 1430 là "Có" vào thời gian t thì yêu cầu tiếp theo được phát hành sẽ là yêu cầu A, trong khi đó nếu câu trả lời cho thử nghiệm được mô tả trong bước 1430 không phải là "Có" cho đến thời gian $t' > t$ thì yêu cầu tiếp theo được phát hành thay vào đó sẽ là yêu cầu B, bởi vì yêu cầu A đã được loại bỏ khỏi danh sách yêu cầu ứng viên giữa thời gian t và t' , hoặc bởi vì yêu cầu B đã được thêm vào danh sách yêu cầu ứng viên có mức ưu tiên cao hơn so với yêu cầu A giữa thời gian t và t' , hoặc bởi vì yêu cầu B ở trong danh sách ứng viên vào thời gian t nhưng với mức ưu tiên thấp hơn so với yêu cầu A, và giữa thời gian t và t' mức ưu tiên của yêu cầu B được áp dụng cao hơn so với yêu cầu A.

Fig.18 minh họa ví dụ về danh sách yêu cầu trong danh sách yêu cầu ứng viên. Trong ví dụ này, có ba kết nối, và có sáu yêu cầu trong danh sách ứng viên, được gắn nhãn là A, B, C, D, E và F. Mỗi trong số các yêu cầu trong danh sách ứng viên có thể được phát hành trên tập con các kết nối như được chỉ báo, ví dụ, yêu cầu A có thể được phát hành trên kết nối 1, trong khi yêu cầu F có thể được phát hành trên kết nối 2 hoặc kết nối 3. Mức ưu tiên của mỗi yêu cầu cũng được gắn nhãn trên Fig.18, và giá trị ưu tiên thấp hơn cho thấy rằng yêu cầu có mức ưu tiên cao hơn. Vì vậy, yêu cầu A và

yêu cầu B với mức ưu tiên 0 là các yêu cầu có mức ưu tiên cao nhất, trong khi yêu cầu F với giá trị ưu tiên 3 là có mức ưu tiên thấp nhất trong số các yêu cầu trong danh sách ứng viên.

Nếu, vào điểm này trong thời gian t , kết nối 1 vượt qua các thử nghiệm được mô tả trong bước 1430, thì yêu cầu A hoặc yêu cầu B được phát hành trên kết nối 1. Nếu thay vào đó kết nối 3 vượt qua các thử nghiệm mô tả trong bước 1430 tại thời gian t , thì yêu cầu D được phát hành trên kết nối 3, do yêu cầu D là yêu cầu có mức ưu tiên cao nhất có thể được phát hành trên kết nối 3.

Giả sử rằng với tất cả các kết nối câu trả lời cho thử nghiệm mô tả trong bước 1430 từ thời gian t đến thời gian sau đó t' là "Không", và giữa thời gian t và t' yêu cầu A thay đổi mức ưu tiên từ 0 thành 5, yêu cầu B được loại bỏ khỏi danh sách ứng viên, và yêu cầu G mới với mức ưu tiên 0 được thêm vào danh sách ứng viên. Sau đó, vào thời gian t' , danh sách ứng viên mới có thể được thể hiện như trên Fig.19.

Nếu vào thời gian t' kết nối 1 vượt qua thử nghiệm được mô tả trong bước 1430, thì yêu cầu C với mức ưu tiên 4 được phát hành trên kết nối 1, vì nó là yêu cầu có mức ưu tiên cao nhất trong danh sách ứng viên có thể được phát hành trên kết nối 1 vào thời điểm này.

Giả sử trong cùng tình huống này thay vì yêu cầu A sẽ được phát hành trên kết nối 1 vào thời gian t (đây là một trong hai lựa chọn có mức ưu tiên cao nhất cho kết nối 1 vào thời gian t như được thể hiện trên Fig.18). Do câu trả lời cho thử nghiệm được mô tả trong bước 1430 từ thời gian t đến thời gian sau đó t' là "Không" cho tất cả các kết nối, kết nối 1 vẫn đang phân phối dữ liệu cho đến ít nhất là thời gian t' đối với các yêu cầu được phát hành trước thời gian t , và do đó yêu cầu A sẽ không bắt đầu cho đến ít nhất là thời gian t' . Việc phát hành yêu cầu C vào thời gian t' là một quyết định tốt hơn so với việc phát hành yêu cầu A vào thời gian t , do yêu cầu C bắt đầu tại cùng thời điểm sau thời gian t' khi yêu cầu A bắt đầu, và do vào thời gian đó yêu cầu C có mức ưu tiên cao hơn so với yêu cầu A.

Theo phương án khác, nếu thử nghiệm thuộc loại được mô tả trong bước 1430 được áp dụng cho tổng hợp các kết nối hoạt động, thì một kết nối có thể được chọn mà có đích có địa chỉ IP được gán với yêu cầu thứ nhất trong danh sách yêu cầu ứng viên hoặc một yêu cầu khác với cùng mức ưu tiên như của yêu cầu thứ nhất.

Một số phương pháp có thể phục vụ cho xây dựng danh sách yêu cầu ứng viên. Ví dụ, danh sách ứng viên có thể chứa n yêu cầu biểu diễn các yêu cầu cho n phần dữ liệu tiếp theo của dạng biểu diễn hiện thời của phần trình diễn theo trình tự thời gian, trong đó yêu cầu cho phần dữ liệu sớm nhất có mức ưu tiên cao nhất và yêu cầu cho phần dữ liệu muộn nhất có mức ưu tiên thấp nhất. Trong một số trường hợp n có thể là một. Giá trị của n có thể phụ thuộc vào kích thước bộ đệm $B_{hiện\ thời}$, hoặc biến *Trạng thái* hoặc một số đo khác của trạng thái về mức chiếm bộ đệm máy khách. Ví dụ, một số giá trị ngưỡng có thể được thiết lập cho $B_{hiện\ thời}$ và giá trị gắn với mỗi ngưỡng và sau đó là giá trị của n được lấy là giá trị gắn với ngưỡng cao nhất mà nhỏ hơn $B_{hiện\ thời}$.

Phương án mô tả ở trên đảm bảo phân bổ linh hoạt các yêu cầu đến các kết nối, đảm bảo ưu tiên tái sử dụng kết nối hiện có ngay cả khi yêu cầu có mức ưu tiên cao nhất không thích hợp cho kết nối đó (vì địa chỉ IP đích của kết nối không phải là địa chỉ được phân bổ cho tên máy chủ bất kỳ gắn với yêu cầu đó). Sự phụ thuộc của n vào $B_{hiện\ thời}$ hoặc biến *Trạng thái* hoặc một số đo khác về mức chiếm bộ đệm của máy khách đảm bảo rằng các yêu cầu "không theo thứ tự ưu tiên" không được phát hành khi máy khách có nhu cầu cấp bách về việc phát hành và hoàn thành yêu cầu gắn với phần dữ liệu tiếp theo được phát ra theo trình tự thời gian.

Có lợi nếu các phương pháp này có thể được kết hợp với HTTP và FEC.

Lựa chọn máy chủ phù hợp

Như đã biết, tập tin được tải về bằng cách sử dụng giao thức tải về tập tin thường được xác định bởi một định danh bao gồm tên máy chủ và tên tập tin. Ví dụ đây là trường hợp đối với giao thức HTTP trong đó định danh là định danh tài nguyên thống nhất (URI – Uniform resource protocol). Tên máy chủ có thể tương ứng với nhiều máy chủ, được xác định bởi các địa chỉ giao thức Internet. Ví dụ, đây là phương pháp phổ biến để trải tải tin của yêu cầu từ nhiều máy khách trên nhiều máy vật lý. Cụ thể, phương pháp này thường được thực hiện bởi mạng phân phối nội dung (CDN – content delivery networks). Trong trường hợp này yêu cầu được phát hành trên kết nối đến máy chủ vật lý bất kỳ dự kiến sẽ thành công. Một số phương pháp được biết đến, trong đó máy khách có thể chọn từ trong số các địa chỉ giao thức Internet gắn với một tên máy chủ. Ví dụ, các địa chỉ này thường được cung cấp cho máy khách thông qua hệ thống tên miền và được cung cấp theo thứ tự ưu tiên. Sau đó, máy khách có thể

chọn địa chỉ IP (thứ nhất) có mức ưu tiên cao nhất. Tuy nhiên, nhìn chung không có sự phối hợp giữa máy khách về cách mà lựa chọn này được thực hiện, với kết quả là các máy khách khác nhau có thể yêu cầu cùng một tập tin từ các máy chủ khác nhau. Điều này có thể dẫn đến cùng một tập tin được lưu trữ trong bộ nhớ đệm của nhiều máy chủ gần đó, và làm giảm hiệu quả của cơ sở hạ tầng bộ nhớ đệm.

Điều này có thể được xử lý bởi hệ thống làm tăng một cách có lợi xác suất mà hai máy khách yêu cầu cùng một khối sẽ yêu cầu khối này từ cùng máy chủ. Các phương pháp mới được mô tả ở đây bao gồm bước chọn từ trong số các địa chỉ giao thức Internet có sẵn theo cách được xác định bởi định danh của tập tin được yêu cầu và theo cách mà các máy khách khác nhau được trình diễn với các lựa chọn giống nhau hoặc tương tự đối với các địa chỉ giao thức Internet và định danh tập tin sẽ thực hiện lựa chọn giống nhau.

Phương án thứ nhất của phương pháp này được mô tả dựa trên Fig.20. Máy khách trước tiên thu nhận tập hợp địa chỉ giao thức Internet $IP_1, IP_2, \dots, IP_n$, như được thể hiện trong bước 1710. Nếu có một tập tin mà yêu cầu sẽ được phát hành cho tập tin đó, như được quyết định trong bước 1720, thì máy khách xác định địa chỉ IP nào để phát hành yêu cầu cho tập tin, như được xác định trong các bước 1730 – 1770. Cho trước tập hợp địa chỉ giao thức Internet và định danh cho tập tin được yêu cầu, phương pháp này bao gồm bước sắp xếp thứ tự các địa chỉ giao thức Internet theo cách được xác định bởi định danh tập tin. Ví dụ, đối với mỗi địa chỉ IP, chuỗi byte được xây dựng bao gồm việc nối địa chỉ giao thức Internet và định danh tập tin, như được thể hiện trong bước 1730. Hàm băm được áp dụng cho chuỗi byte này, như trong bước 1740, và giá trị băm kết quả được sắp xếp theo thứ tự cố định, như trong bước 1750, ví dụ theo số thứ tự tăng dần, suy ra thứ tự trên các địa chỉ IP. Cùng hàm băm có thể được sử dụng bởi tất cả máy khách, do đó đảm bảo rằng cùng kết quả được tạo ra bởi hàm băm trên đầu vào cho trước cho tất cả các máy khách. Hàm băm có thể được tạo cấu hình tĩnh trên tất cả các máy khách trong tập hợp máy khách, hoặc tất cả các máy khách trong tập hợp máy khách có thể thu được mô tả một phần hoặc mô tả đầy đủ của hàm băm khi các máy khách thu được danh sách các địa chỉ giao thức Internet, hoặc tất cả máy khách trong tập hợp máy khách có thể thu được mô tả một phần hoặc toàn bộ của hàm băm khi máy khách thu được định danh tập tin, hoặc hàm băm có thể được xác định bằng phương tiện khác. Các địa chỉ giao thức Internet đầu tiên theo thứ tự

này được chọn và sau đó địa chỉ này được sử dụng để thiết lập kết nối và phát hành yêu cầu cho toàn bộ hoặc các phần của tập tin, như được thể hiện trong các bước 1760 và 1770.

Phương pháp trên có thể được áp dụng khi kết nối mới được thành lập để yêu cầu tập tin. Phương pháp này cũng có thể được áp dụng khi một số kết nối được thiết lập là có sẵn và một trong chúng có thể được chọn để phát hành yêu cầu mới.

Hơn nữa, khi kết nối đã thiết lập là có sẵn và yêu cầu có thể được chọn từ tập hợp yêu cầu ứng viên với mức ưu tiên bằng nhau, thứ tự trên các yêu cầu ứng viên được suy ra, ví dụ, bằng cùng phương pháp giá trị băm được mô tả ở trên và yêu cầu ứng viên xuất hiện đầu tiên theo thứ tự này được chọn. Các phương pháp có thể được kết hợp để chọn cả kết nối lẫn yêu cầu ứng viên từ tập hợp kết nối và yêu cầu có mức ưu tiên bằng nhau, một lần nữa bằng cách tính toán giá trị băm cho mỗi kết nối của kết nối và yêu cầu, sắp xếp thứ tự các giá trị băm theo thứ tự cố định và chọn kết nối mà xuất hiện đầu tiên theo thứ tự được suy ra trên tập hợp các kết nối của yêu cầu và kết nối.

Phương pháp này có lợi thế vì lý do sau: phương pháp điển hình được thực hiện bởi cơ sở hạ tầng phục vụ khối, chẳng hạn như cơ sở hạ tầng được thể hiện trên Fig.1 (BSI 101) hoặc Fig.2 (BSIS 101), và cụ thể là phương pháp thường được thực hiện bởi CDN, là để cung cấp nhiều máy chủ proxy đệm nhận các yêu cầu máy khách. Máy chủ proxy đệm có thể không được cung cấp tập tin được yêu cầu trong một yêu cầu nhất định và trong trường hợp này các máy chủ đó thường gửi yêu cầu đến máy chủ khác, nhận phản hồi từ máy chủ này, thường bao gồm cả tập tin được yêu cầu, và chuyển tiếp phản hồi đến máy khách. Máy chủ proxy đệm cũng có thể lưu trữ (đệm) tập tin được yêu cầu để có thể phản hồi ngay lập tức với các yêu cầu tiếp theo cho tập tin. Phương pháp phổ biến được mô tả ở trên có đặc điểm là tập hợp tập tin được lưu trữ trên máy chủ proxy đệm phần lớn là được xác định bởi tập hợp yêu cầu mà máy chủ proxy đệm đã nhận được.

Phương pháp mô tả ở trên có ưu điểm như sau. Nếu tất cả các máy khách trong tập hợp máy khách được cung cấp cùng danh sách địa chỉ IP thì các máy khách này sẽ sử dụng cùng địa chỉ IP cho tất cả các yêu cầu được phát hành cho cùng một tập tin. Nếu có hai danh sách địa chỉ giao thức Internet khác nhau và mỗi máy khách được cung cấp một trong hai danh sách này thì máy khách sẽ sử dụng nhiều nhất là hai địa

chỉ giao thức Internet khác nhau cho tất cả các yêu cầu được phát hành cho cùng một tập tin. Nói chung, nếu các danh sách các địa chỉ giao thức Internet được cung cấp cho các máy khách là tương tự nhau thì máy khách sẽ sử dụng tập hợp nhỏ của các địa chỉ giao thức Internet được cung cấp cho tất cả các yêu cầu phát hành cho cùng một tập tin. Do các máy khách lân cận có xu hướng được cung cấp danh sách địa chỉ giao thức Internet tương tự, có khả năng là các máy khách lân cận phát hành các yêu cầu cho tập tin từ chỉ một phần nhỏ của các máy chủ proxy đệm có sẵn cho các máy khách này. Như vậy, sẽ chỉ có một phần nhỏ của các máy chủ proxy đệm thực hiện đệm tập tin, điều này có lợi là giảm thiểu lượng tài nguyên bộ nhớ đệm được sử dụng để đệm tập tin.

Tốt hơn là, hàm băm có đặc tính là một phần rất nhỏ của các đầu vào khác nhau được ánh xạ đến cùng đầu ra, và rằng các đầu vào khác nhau được ánh xạ đến các đầu ra về cơ bản là ngẫu nhiên, để đảm bảo rằng với tập hợp các địa chỉ IP cho trước, phần tập tin mà một địa chỉ cho trước trong số các địa chỉ IP là địa chỉ đầu tiên trong danh sách sắp xếp được tạo ra ở bước 1750 là xấp xỉ như nhau cho tất cả các địa chỉ IP trong danh sách. Mặt khác, điều quan trọng là hàm băm có tính xác định, theo nghĩa là với một đầu vào cho trước, đầu ra của hàm băm là giống nhau cho tất cả các máy khách.

Ưu điểm khác của phương pháp mô tả ở trên là như sau. Giả sử rằng tất cả các máy khách trong tập hợp máy khách được cung cấp cùng danh sách địa chỉ IP. Do các tính chất của hàm băm mô tả trên đây, có khả năng là các yêu cầu cho các tập tin khác nhau từ các máy khách này sẽ được trải đều trên tập hợp địa chỉ giao thức Internet, điều này có nghĩa là các yêu cầu sẽ được trải đều trên máy chủ proxy đệm. Do đó, tài nguyên bộ nhớ đệm dùng để lưu trữ các tập tin này được trải đều trên các máy chủ proxy đệm, và yêu cầu cho tập tin được trải đều trên các máy chủ proxy đệm. Vì vậy, phương pháp này cung cấp cả khả năng cân bằng lưu trữ và cân bằng tải trên cơ sở hạ tầng bộ nhớ đệm.

Một số biến thể của phương pháp mô tả ở trên đã được biết đến bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật và trong nhiều trường hợp, các biến thể này giữ lại đặc tính đó là tập hợp tập tin được lưu trữ trên proxy cho trước được xác định ít nhất một phần bởi tập hợp các yêu cầu mà máy chủ proxy đệm đã nhận được. Trong trường hợp thông thường trong đó tên máy chủ cho trước dùng cho nhiều máy

chủ proxy đệm vật lý, thông thường là tất cả các máy chủ này cuối cùng sẽ lưu trữ một bản sao của tập tin bất kỳ cho trước mà được yêu cầu thường xuyên. Việc trùng lặp như vậy có thể không được mong muốn, vì tài nguyên lưu trữ trên máy chủ proxy đệm bị giới hạn và do đó các tập tin có thể được, đôi khi, loại bỏ (thanh lọc) khỏi bộ nhớ đệm. Phương pháp mới được mô tả ở đây đảm bảo rằng các yêu cầu cho tập tin cho trước được chuyển đến máy chủ proxy đệm theo cách sao cho sự trùng lặp này được giảm bớt, do đó làm giảm nhu cầu loại bỏ tập tin khỏi bộ nhớ đệm và do đó làm tăng khả năng mà tập tin bất kỳ cho trước có mặt trong (tức là, chưa được thanh lọc ra khỏi) bộ nhớ proxy đệm.

Khi một tập tin có mặt trong bộ nhớ proxy đệm, phản hồi gửi cho máy khách sẽ nhanh hơn, điều này có lợi thế trong việc làm giảm xác suất mà tập tin đã yêu cầu đến muộn, điều này có thể dẫn đến sự tạm dừng phát ra phương tiện và do đó có trải nghiệm người dùng kém. Ngoài ra, khi tập tin không có mặt trong bộ nhớ proxy đệm, yêu cầu này có thể được gửi đến máy chủ khác, gây ra phụ tải trên cả cơ sở hạ tầng phục vụ lẫn kết nối mạng giữa các máy chủ. Trong nhiều trường hợp máy chủ mà yêu cầu được gửi đến có thể ở vị trí xa và việc truyền tập tin từ máy chủ này lại cho máy chủ proxy đệm có thể phải trả chi phí truyền. Do đó, phương pháp mới được mô tả ở đây cho phép giảm chi phí truyền.

Yêu cầu tập tin toàn bộ theo xác suất

Mỗi quan tâm cụ thể trong trường hợp giao thức HTTP được sử dụng với yêu cầu dải đó là hoạt động của máy chủ đệm thường được sử dụng để cung cấp khả năng mở rộng trong cơ sở hạ tầng phục vụ. Trong khi có thể phổ biến khi máy chủ đệm HTTP hỗ trợ phân tiêu đề dải HTTP, hoạt động chính xác của các máy chủ đệm HTTP khác nhau sẽ thay đổi theo cài đặt. Hầu hết các triển khai máy chủ đệm phục vụ các yêu cầu dải từ bộ nhớ đệm trong trường hợp tập tin có sẵn trong bộ nhớ đệm. Một cài đặt phổ biến của máy chủ đệm HTTP luôn chuyển tiếp các yêu cầu HTTP phía dưới chứa tiêu đề dải đến nút phía trên trừ khi máy chủ đệm có bản sao của tập tin (máy chủ đệm hoặc máy chủ gốc). Trong một số cài đặt, phản hồi phía trên cho yêu cầu dải là toàn bộ tập tin, và toàn bộ tập tin này được lưu trữ đệm và phản hồi cho yêu cầu dải phía dưới được trích xuất từ tập tin này và gửi đi. Tuy nhiên, trong ít nhất một cài đặt, phản hồi phía trên cho yêu cầu dải chỉ là các byte dữ liệu trong bản thân yêu cầu dải,

và các byte dữ liệu này không được lưu trữ đệm mà thay vào đó chỉ cần được gửi như là phản hồi cho yêu cầu dài phía dưới. Kết quả là, việc sử dụng các tiêu đề dài bởi máy khách có thể có hậu quả là bản thân tập tin không bao giờ được đưa vào bộ nhớ đệm và các đặc tính khả năng mở rộng mong muốn của mạng sẽ bị mất.

Trong phần mô tả trên, sáng chế mô tả hoạt động của máy chủ proxy đệm và đồng thời mô tả phương pháp yêu cầu các khối từ tập tin mà là sự tổng hợp của nhiều khối. Ví dụ, điều này có thể đạt được bằng cách sử dụng tiêu đề yêu cầu dài HTTP. Yêu cầu này được gọi là "yêu cầu một phần" trong phần mô tả dưới đây. Phương án khác nữa sẽ được mô tả dưới đây, phương án này có lợi trong trường hợp cơ sở hạ tầng phục vụ khối 101 không cung cấp hỗ trợ đầy đủ cho tiêu đề dài HTTP. Thông thường, máy chủ trong cơ sở hạ tầng phục vụ khối, ví dụ như mạng phân phối nội dung, hỗ trợ các yêu cầu một phần nhưng có thể không lưu trữ phản hồi cho các yêu cầu một phần trong bộ phận lưu trữ cục bộ (bộ nhớ đệm). Máy chủ này có thể thực hiện yêu cầu một phần bằng cách chuyển tiếp yêu cầu đó đến máy chủ khác, trừ khi toàn bộ tập tin được lưu trữ trong bộ phận lưu trữ cục bộ, trong trường hợp đó phản hồi có thể được gửi mà không cần chuyển tiếp yêu cầu đến máy chủ khác.

Hệ thống truyền liên tục yêu cầu theo khối mà tận dụng sự cải tiến kết hợp khối mới được mô tả ở trên có thể hoạt động kém nếu cơ sở hạ tầng phục vụ khối bộc lộ hoạt động này, vì tất cả các yêu cầu, là yêu cầu một phần, sẽ được chuyển tiếp đến máy chủ khác và không có yêu cầu nào sẽ được phục vụ bởi máy chủ proxy đệm, làm thất bại mục tiêu cung cấp máy chủ proxy đệm đầu tiên. Trong quá trình truyền liên tục yêu cầu theo khối như được mô tả ở trên, có thể lúc nào đó máy khách yêu cầu một khối mà ở đầu của tập tin.

Theo phương pháp mới được mô tả ở đây, bất cứ khi nào một điều kiện nhất định được đáp ứng, các yêu cầu này có thể được chuyển đổi từ các yêu cầu cho khối thứ nhất trong tập tin thành các yêu cầu cho toàn bộ tập tin. Khi yêu cầu cho toàn bộ tập tin được nhận bởi máy chủ proxy đệm, máy chủ proxy thường lưu trữ phản hồi. Vì vậy việc sử dụng các yêu cầu này làm cho tập tin được đưa vào bộ nhớ đệm của các máy chủ proxy đệm cục bộ sao cho các yêu cầu tiếp theo, cho dù là cho toàn bộ tập tin hay các yêu cầu một phần có thể được phục vụ trực tiếp bởi máy chủ proxy đệm này. Các điều kiện có thể là sao cho trong tập hợp các yêu cầu gắn với tập tin nhất định, ví dụ như tập hợp các yêu cầu được tạo ra bởi tập hợp máy khách đang xem mục nội

dung này, tình trạng này sẽ được đáp ứng đối với ít nhất một phần được cung cấp của các yêu cầu này.

Ví dụ về điều kiện thích hợp đó là số được chọn ngẫu nhiên lớn hơn ngưỡng được cung cấp. Ngưỡng này có thể được thiết lập sao cho sự chuyển đổi một yêu cầu khối duy nhất thành yêu cầu toàn bộ tập tin xảy ra trung bình đối với một phần được cung cấp của các yêu cầu, ví dụ, một lần trong số mười lần (trong trường hợp này số ngẫu nhiên có thể được chọn trong khoảng $[0,1]$ và ngưỡng có thể là 0,9). Ví dụ khác về điều kiện thích hợp đó là hàm băm được tính toán trên thông tin nào đó gắn với khối và thông tin nào đó gắn với máy khách nhận một giá trị trong tập hợp giá trị được cung cấp. Phương pháp này có ưu điểm là với tập tin được yêu cầu thường xuyên, tập tin này sẽ được đưa vào bộ nhớ đệm của máy chủ proxy đệm cục bộ tuy nhiên hoạt động của hệ thống truyền liên tục yêu cầu theo khối không được thay đổi nhiều so với hoạt động tiêu chuẩn trong đó mỗi yêu cầu là cho một khối duy nhất. Trong nhiều trường hợp, khi sự chuyển đổi yêu cầu từ một yêu cầu khối duy nhất thành yêu cầu toàn bộ tập tin xảy ra, các thủ tục của máy khách theo cách khác sẽ tiếp tục yêu cầu các khối khác trong tập tin này. Nếu đúng như vậy, thì các yêu cầu này có thể bị loại bỏ vì các khối đó sẽ được nhận trong mọi trường hợp do việc yêu cầu toàn bộ tập tin.

Xây dựng URL, tạo ra và tìm kiếm danh sách phân đoạn

Việc tạo ra danh sách phân đoạn giải quyết vấn đề làm thế nào máy khách có thể tạo ra danh sách phân đoạn từ MPD tại thời gian cục bộ ở máy khách NOW cụ thể cho một dạng biểu diễn cụ thể mà bắt đầu tại thời gian bắt đầu nào đó StartTime có liên quan đến sự bắt đầu của phần trình diễn phương tiện cho trường hợp theo yêu cầu hoặc được thể hiện theo thời gian thực. Danh sách phân đoạn có thể bao gồm phần tử định vị, ví dụ như một URL chỉ đến siêu dữ liệu biểu diễn ban đầu tùy chọn, cũng như danh sách phân đoạn phương tiện. Mỗi phân đoạn phương tiện có thể được gán một thời gian bắt đầu starttime, thời lượng và phần tử định vị. Thông thường, StartTime thể hiện giá trị xấp xỉ của thời gian phương tiện của phương tiện có trong một phân đoạn, nhưng không nhất thiết là thời gian chính xác của mẫu. StartTime được sử dụng bởi máy khách truyền liên tục HTTP để phát hành yêu cầu tải về tại thời điểm thích hợp. Việc tạo ra danh sách phân đoạn, bao gồm cả thời gian bắt đầu của mỗi phân đoạn, có thể được thực hiện theo các cách khác nhau. Các URL có thể được cung cấp như là

danh sách phát hay có lợi nếu quy tắc xây dựng URL có thể được sử dụng cho dạng biểu diễn nhỏ gọn của danh sách phân đoạn này.

Danh sách phân đoạn dựa trên việc xây dựng URL có thể, ví dụ, được thực hiện nếu MPD báo hiệu bằng thuộc tính hoặc phần tử cụ thể như `fileDynamicInfo` hoặc tín hiệu tương đương. Cách chung để tạo danh sách phân đoạn từ việc xây dựng URL được cung cấp dưới đây trong đoạn "Tổng quan về bộ xây dựng URL". Việc xây dựng dựa trên danh sách phát có thể, ví dụ, được báo hiệu bởi tín hiệu khác. Việc tìm kiếm trong danh sách phân đoạn và nhận được thời gian phương tiện chính xác có lợi khi được thực hiện trong ngữ cảnh này.

Tổng quan về bộ xây dựng URL

Như đã mô tả trước đây, theo một phương án của sáng chế, có thể cung cấp tập tin siêu dữ liệu có chứa quy tắc xây dựng URL cho phép các thiết bị của máy khách xây dựng các định danh tập tin cho các khối của phần trình diễn. Phần dưới đây mô tả biện pháp cải tiến mới cho hệ thống truyền liên tục yêu cầu theo khối mà cung cấp các thay đổi trong tập tin siêu dữ liệu, bao gồm các thay đổi đối với quy tắc xây dựng URL, thay đổi số lượng phân mã hóa có sẵn, thay đổi siêu dữ liệu gắn với các phân mã hóa có sẵn như tốc độ bit, tỷ lệ khung hình, độ phân giải, phân mã hóa-giải mã âm thanh hoặc video hoặc các tham số mã hóa-giải mã, hoặc các tham số khác.

Trong phần cải tiến mới này, có thể cung cấp dữ liệu bổ sung gắn với mỗi phần tử của tập tin siêu dữ liệu chỉ rõ khoảng thời gian trong toàn bộ phần trình diễn. Trong khoảng thời gian này, phần tử có thể được coi là hợp lệ và nếu không thì trong khoảng thời gian đó phần tử này có thể được bỏ qua. Hơn nữa, cú pháp của siêu dữ liệu có thể được cải tiến sao cho phần tử trước đây được phép xuất hiện chỉ một lần hoặc nhiều nhất một lần có thể xuất hiện nhiều lần. Hạn chế bổ sung có thể được áp dụng trong trường hợp này, cung cấp hạn chế đó cho các phần tử mà các khoảng thời gian cụ thể được chỉ rõ phải tách rời nhau. Tại bất kỳ thời điểm nào, việc chỉ xem xét các phần tử có khoảng thời gian có chứa kết quả thời gian nhất định tạo ra tập tin siêu dữ liệu phù hợp với cú pháp của siêu dữ liệu ban đầu. Gọi khoảng thời gian đó là khoảng thời gian hợp lệ. Do đó, phương pháp này cung cấp các thay đổi thuộc loại mô tả trên đây cho việc báo hiệu trong một tập tin siêu dữ liệu. Có lợi nếu phương pháp này có thể được

sử dụng để cung cấp phần trình diễn phương tiện mà hỗ trợ các thay đổi thuộc loại được mô tả tại các điểm đã chỉ rõ trong phần trình diễn.

Bộ xây dựng URL

Như đã mô tả ở đây, một đặc điểm chung của hệ thống truyền liên tục yêu cầu theo khối là sự cần thiết phải cung cấp cho máy khách "siêu dữ liệu" xác định các phần mã hóa có sẵn của phương tiện và cung cấp thông tin cần cho máy khách để yêu cầu các khối từ các phần mã hóa này. Ví dụ trong trường hợp của HTTP, thông tin này có thể bao gồm các URL cho các tập tin có chứa các khối phương tiện. Tập tin danh sách phát có thể được cung cấp trong đó liệt kê các URL cho các khối cho phần mã hóa cho trước. Nhiều tập tin danh sách phát được cung cấp, một tập tin cho mỗi phần mã hóa, cùng với danh sách tổng thể của các danh sách phát liệt kê các danh sách phát tương ứng với phần mã hóa khác nhau. Nhược điểm của hệ thống này là siêu dữ liệu có thể trở nên khá lớn và do đó phải mất thời gian để được yêu cầu khi máy khách bắt đầu quy trình truyền liên tục. Nhược điểm khác nữa của hệ thống này là hiển nhiên trong trường hợp nội dung trực tiếp, khi các tập tin tương ứng với các khối dữ liệu phương tiện được tạo ra "ngay khi cần" từ dòng phương tiện đang được ghi theo thời gian thực (trực tiếp), ví dụ sự kiện thể thao trực tiếp hay chương trình tin tức. Trong trường hợp này các tập tin danh sách phát có thể được cập nhật mỗi khi khối mới có sẵn (ví dụ cứ mỗi vài giây). Thiết bị máy khách có thể liên tục lấy tập tin danh sách phát để xác định xem các khối mới có sẵn hay không và thu được URL của chúng. Điều này có thể đặt ra tải đáng kể lên cơ sở hạ tầng phục vụ và cụ thể có nghĩa là tập tin siêu dữ liệu không thể được lưu trữ lâu hơn khoảng thời gian cập nhật, mà bằng với kích thước khối thường có độ lớn một vài giây.

Một khía cạnh quan trọng của hệ thống truyền liên tục yêu cầu theo khối đó là phương pháp được sử dụng để thông báo cho máy khách về các định danh tập tin, ví dụ URL, nên được sử dụng, cùng với giao thức tải về tập tin, để yêu cầu các khối. Ví dụ, phương pháp trong đó với mỗi dạng biểu diễn của phần trình diễn cung cấp tập tin danh sách phát mà liệt kê các URL của tập tin chứa các khối của dữ liệu phương tiện. Nhược điểm của phương pháp này là ít nhất một số tập tin danh sách phát cần phải được tải về trước khi có thể bắt đầu phát ra, làm tăng thời gian sửa đổi kênh do đó gây ra trải nghiệm người dùng kém. Đối với phần trình diễn phương tiện dài với một số

hoặc nhiều dạng biểu diễn, danh sách các URL tập tin có thể lớn và do đó tập tin danh sách phát có thể lớn, làm tăng thêm thời gian sửa đổi kênh.

Nhược điểm khác của phương pháp này xảy ra trong trường hợp nội dung trực tiếp. Trong trường hợp này, danh sách đầy đủ các URL không được cung cấp sẵn trước và tập tin danh sách phát được cập nhật định kỳ khi các khối mới trở nên có sẵn và máy khách định kỳ yêu cầu tập tin danh sách phát, để nhận được phiên bản cập nhật. Bởi vì tập tin này được cập nhật thường xuyên, nó không thể được lưu trữ lâu trong các máy chủ proxy đệm. Điều này có nghĩa là rất nhiều yêu cầu trong số các yêu cầu cho tập tin này sẽ được chuyển tiếp đến máy chủ khác và cuối cùng đến máy chủ tạo ra tập tin đó. Trong trường hợp trình diễn phương tiện phổ biến, điều này có thể tạo ra tải cao trên máy chủ này và mạng, mà có thể dẫn đến thời gian phản hồi chậm và do đó thời gian sửa đổi kênh cao và trải nghiệm người dùng kém. Trong trường hợp xấu nhất, máy chủ trở nên quá tải và điều này dẫn đến việc một số người dùng không thể xem phần trình diễn.

Mong muốn thiết kế hệ thống truyền liên tục yêu cầu theo khối để tránh việc đặt các hạn chế về dạng của định danh tập tin có thể được sử dụng. Điều này là do một số cân nhắc có thể thúc đẩy việc sử dụng các định danh ở dạng cụ thể. Ví dụ, trong trường hợp cơ sở hạ tầng phục vụ khối là mạng phân phối nội dung, có thể có các quy ước về đặt tên tập tin hoặc lưu trữ tập tin liên quan đến mong muốn phân phối lưu trữ hoặc tải phục vụ trên mạng hoặc yêu cầu khác dẫn đến các dạng cụ thể của định danh tập tin mà không thể được dự đoán vào thời điểm thiết kế hệ thống.

Phần dưới đây mô tả phương án khác có khả năng giảm bớt các nhược điểm nêu trên, trong khi duy trì được khả năng linh hoạt để chọn quy ước định danh tập tin thích hợp. Trong phương pháp này, siêu dữ liệu có thể được cung cấp cho mỗi dạng biểu diễn của phần trình diễn phương tiện bao gồm một quy tắc xây dựng định danh tập tin. Quy tắc xây dựng định danh tập tin có thể, ví dụ, bao gồm một chuỗi văn bản. Để xác định định danh tập tin cho khối nhất định của phần trình diễn, phương pháp giải thích quy tắc xây dựng định danh tập tin có thể được cung cấp, phương pháp này bao gồm bước xác định các tham số đầu vào và đánh giá quy tắc xây dựng định danh tập tin cùng với các tham số đầu vào. Các tham số đầu vào có thể, ví dụ, bao gồm chỉ mục của tập tin cần được xác định, trong đó tập tin thứ nhất có chỉ mục không, tập tin thứ hai có chỉ mục một, tập tin thứ ba có chỉ mục hai, v.v.. Ví dụ, trong trường hợp mọi

tập tin trải trên cùng thời lượng (hoặc thời lượng xấp xỉ bằng nhau), thì chỉ mục của tập tin gắn với thời gian cho trước bất kỳ trong phần trình diễn có thể dễ dàng được xác định. Ngoài ra, thời gian trong phần trình diễn kéo dài của mỗi tập tin có thể được cung cấp trong phần trình diễn hay siêu dữ liệu của phiên bản.

Theo một phương án, quy tắc xây dựng định danh tập tin có thể bao gồm một chuỗi văn bản có thể chứa định danh đặc biệt nhất định tương ứng với các tham số đầu vào. Phương pháp đánh giá quy tắc xây dựng định danh tập tin bao gồm bước xác định vị trí của các định danh đặc biệt trong chuỗi văn bản và thay thế mỗi định danh đặc biệt này bằng một dạng biểu diễn chuỗi có giá trị của tham số đầu vào tương ứng.

Theo phương án khác, quy tắc xây dựng định danh tập tin có thể bao gồm một chuỗi văn bản phù hợp với ngôn ngữ biểu thức. Ngôn ngữ biểu thức bao gồm định nghĩa về cú pháp mà biểu thức bằng ngôn ngữ đó có thể tuân theo và bộ quy tắc để đánh giá chuỗi phù hợp với cú pháp này.

Phần mô tả dưới đây sẽ mô tả một ví dụ cụ thể, dựa trên Fig.21. Ví dụ về định nghĩa cú pháp cho ngôn ngữ biểu thức phù hợp được xác định theo dạng Augmented Backus – Naur, như được thể hiện trên Fig.21. Ví dụ về quy tắc để đánh giá chuỗi theo luật sinh <biểu thức> trên Fig.21 bao gồm biến đổi đệ quy chuỗi tuân theo luật sinh <biểu thức> (<biểu thức>) thành một chuỗi tuân theo luật sinh <ký tự> như sau:

<Biểu thức> tuân theo luật sinh <ký tự> là không đổi.

<Biểu thức> tuân theo luật sinh <biến> được thay thế bằng giá trị của biến được xác định bởi chuỗi <thể xác thực> của luật sinh <biến>.

<Biểu thức> tuân theo luật sinh <hàm> được đánh giá bằng cách đánh giá mỗi đối số của nó theo các quy tắc này và áp dụng phép biến đổi cho các đối số này phụ thuộc vào phần tử <thể xác thực> của luật sinh <hàm> như được mô tả dưới đây.

<Biểu thức> tuân theo phương án cuối cùng của luật sinh <biểu thức> được đánh giá bằng cách đánh giá hai phần tử <biểu thức> và áp dụng phép toán cho các đối số này phụ thuộc vào phần tử <toán tử> của phương án cuối cùng của luật sinh <biểu thức> như được mô tả dưới đây.

Trong phương pháp được mô tả ở trên giả sử rằng việc đánh giá diễn ra trong ngữ cảnh trong đó các biến có thể được xác định. Biến là một cặp (tên, giá trị), trong đó "tên" là chuỗi tuân theo cú pháp luật sinh <thể xác thực> và "giá trị" là chuỗi tuân theo luật sinh <ký tự>. Một số biến có thể được xác định ngoài quy trình đánh giá

trước khi quy trình đánh giá bắt đầu. Các biến khác có thể được xác định trong chính quy trình đánh giá. Tất cả các biến là "chung" theo nghĩa là chỉ có một biến tồn tại với mỗi "tên" có thể có.

Ví dụ về hàm là hàm "printf". Hàm này nhận một hoặc nhiều đối số. Đối số thứ nhất có thể tuân theo luật sinh <chuỗi> (sau đây gọi là "chuỗi"). Hàm printf đánh giá phiên bản chuyển đổi của đối số thứ nhất. Việc chuyển đổi được áp dụng là giống như hàm "printf" của thư viện tiêu chuẩn C, với các đối số bổ sung có trong luật sinh <hàm> cung cấp các đối số bổ sung dự kiến bởi hàm printf của thư viện tiêu chuẩn C.

Một ví dụ khác về hàm là hàm "băm". Hàm này nhận hai đối số, đối số thứ nhất có thể là chuỗi và đối số thứ hai có thể phù hợp với luật sinh <số> (sau đây gọi là "số"). Hàm "băm" áp dụng thuật toán băm cho đối số thứ nhất và trả về kết quả là một số nguyên không âm nhỏ hơn đối số thứ hai. Ví dụ về hàm băm phù hợp được đưa ra trong hàm C được thể hiện trên Fig.22, các đối số của hàm này là chuỗi đầu vào (không bao gồm dấu ngoặc kép kèm theo) và giá trị số đầu vào. Các ví dụ khác về hàm băm cũng đã được biết đến bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật.

Một ví dụ khác về hàm là hàm "subst" mà nhận một, hai hoặc ba đối số chuỗi. Trong trường hợp một đối số được cung cấp, kết quả của hàm "subst" là đối số thứ nhất. Trong trường hợp hai đối số được cung cấp thì kết quả của hàm "subst" được tính bằng cách xóa bất kỳ lần xuất hiện nào của đối số thứ hai (không bao gồm dấu ngoặc kép kèm theo) trong đối số thứ nhất và trả về đối số thứ nhất được sửa đổi. Trong trường hợp ba đối số được cung cấp thì kết quả của hàm "subst" được tính bằng cách thay thế bất kỳ lần xuất hiện nào của đối số thứ hai (không bao gồm dấu ngoặc kép kèm theo) trong đối số thứ nhất bằng đối số thứ ba (không bao gồm dấu ngoặc kép kèm theo) và trả về đối số thứ nhất được sửa đổi như vậy.

Một số ví dụ về các toán tử là toán tử cộng, trừ, chia, nhân, và toán tử môđun, được nhận dạng bởi luật sinh <toán tử> '+', '-', '/', '*', '%' tương ứng. Các toán tử này yêu cầu các luật sinh <biểu thức> ở một trong hai phía của luật sinh <toán tử> để đánh giá thành con số. Việc đánh giá toán tử bao gồm việc áp dụng các phép toán số học thích hợp (cộng, trừ, chia, nhân và môđun tương ứng) cho hai số này theo cách thông thường và trả về kết quả dưới dạng phù hợp với luật sinh <số>.

Một ví dụ khác về toán tử là toán tử gán, xác định bởi luật sinh <toán tử> '='. Toán tử này yêu cầu đối số bên trái đánh giá chuỗi nội dung phù hợp với luật sinh <thể

xác thực>. Nội dung của chuỗi được xác định là chuỗi ký tự trong dấu ngoặc kép kèm theo. Toán tử “bằng” làm cho các biến có tên là <thẻ xác thực> tương đương với nội dung của đối số bên trái được gán một giá trị tương đương với kết quả của việc đánh giá đối số bên phải. Giá trị này cũng là kết quả của việc đánh giá biểu thức toán tử.

Một ví dụ khác về toán tử là toán tử chuỗi, xác định bởi luật sinh <toán tử> ';'. Kết quả của việc đánh giá toán tử này là đối số bên phải. Lưu ý rằng, cũng như với tất cả các toán tử, cả hai đối số được đánh giá và đối số bên trái được đánh giá trước.

Theo một phương án của sáng chế, định danh của tập tin có thể thu được bằng cách đánh giá quy tắc xây dựng định danh tập tin theo quy tắc trên đây với một tập hợp cụ thể của các biến đầu vào mà xác định tập tin yêu cầu. Ví dụ về biến đầu vào là biến với tên gọi "chỉ mục" và giá trị bằng với số chỉ mục của tập tin trong phần trình diễn. Một ví dụ khác của biến đầu vào là biến với tên "tốc độ bit" và giá trị bằng với tốc độ bit trung bình của phiên bản yêu cầu của phần trình diễn.

Fig.23 minh họa một số ví dụ về các quy tắc xây dựng định danh tập tin, trong đó các biến đầu vào là "id", cung cấp định danh cho dạng biểu diễn của phần trình diễn mong muốn và "seq", cung cấp số thứ tự cho tập tin.

Người có hiểu biết trung bình trong lĩnh vực kỹ thuật khi đọc sáng chế sẽ hiểu rằng, rất nhiều biến thể của phương pháp trên đây có thể được tạo ra. Ví dụ, không phải tất cả các hàm và toán tử được mô tả ở trên có thể được cung cấp hoặc các hàm bổ sung hoặc các toán tử bổ sung có thể được cung cấp.

Các quy tắc xây dựng URL và việc định thời

Phần dưới đây mô tả các quy tắc xây dựng URI cơ bản để gán URI cho tập tin hoặc phân đoạn cũng như thời gian bắt đầu cho từng phân đoạn trong dạng biểu diễn và phần trình diễn phương tiện.

Đối với việc này, giả sử mô tả trình diễn phương tiện tại máy khách là có sẵn.

Giả sử rằng máy khách truyền liên tục HTTP đang phát ra phương tiện sẽ được tải về trong phần trình diễn phương tiện. Thời gian trình diễn thực tế của máy khách HTTP có thể được xác định đối với nơi thời gian trình diễn có liên quan đến sự bắt đầu của phần trình diễn. Lúc khởi tạo, thời gian trình diễn $t = 0$ có thể được giả định.

Tại thời điểm bất kỳ t , máy khách HTTP có thể tải về dữ liệu bất kỳ với thời gian phát - tP (cũng liên quan đến sự bắt đầu của phần trình diễn) tại

MaximumClientPreBufferTime trước thời gian trình diễn thực tế t và dữ liệu bất kỳ cần thiết do tương tác của người dùng, ví dụ như tìm kiếm, tua nhanh, v.v.. Trong một số phương án, MaximumClientPreBufferTime thậm chí có thể không được xác định theo nghĩa là máy khách có thể tải về dữ liệu trước thời gian phát hiện thời - thời gian tP mà không có hạn chế.

Máy khách HTTP có thể tránh việc tải về dữ liệu không cần thiết, ví dụ như các phân đoạn bất kỳ từ dạng biểu diễn mà không dự kiến sẽ được phát ra thường có thể không được tải về.

Quy trình cơ bản trong việc cung cấp dịch vụ truyền liên tục có thể là tải về dữ liệu bằng cách tạo ra các yêu cầu phù hợp để tải về toàn bộ tập tin/phân đoạn hoặc tập con của tập tin/phân đoạn, ví dụ bằng cách sử dụng các yêu cầu GET HTTP hoặc yêu cầu GET HTTP một phần. Phần mô tả này đề cập đến cách truy cập dữ liệu cho thời gian phát ra cụ thể tP nhưng nói chung máy khách có thể tải về dữ liệu cho khoảng thời gian phát lớn hơn để tránh các yêu cầu không hiệu quả. Máy khách HTTP có thể giảm thiểu số lượng/tần số của các yêu cầu HTTP trong việc cung cấp dịch vụ truyền liên tục.

Để truy cập dữ liệu phương tiện ở thời gian phát tP hoặc ít nhất là gần với thời gian phát tP trong một dạng biểu diễn cụ thể, máy khách xác định URL đến tập tin có chứa thời gian phát và thêm vào đó xác định dãy byte trong tập tin để truy cập vào thời gian phát này.

Mô tả trình diễn phương tiện có thể gán id biểu diễn, r , cho mỗi dạng biểu diễn, ví dụ như bằng cách sử dụng thuộc tính RepresentationID. Nói cách khác, nội dung của MPD, khi được viết bởi hệ thống thu thập hoặc khi được đọc bởi máy khách, sẽ được giải thích sao cho có phần gán này. Để tải về dữ liệu cho một thời gian phát cụ thể tP cho một dạng biểu diễn cụ thể với id r , máy khách có thể xây dựng một URI thích hợp cho tập tin.

Mô tả trình diễn phương tiện có thể gán cho mỗi tập tin hoặc phân đoạn của từng dạng biểu diễn r các thuộc tính sau:

(a) số thứ tự i của tập tin trong dạng biểu diễn r , với $i = 1, 2, \dots, N_r$, (b) thời gian bắt đầu tương đối của tập tin với id dạng biểu diễn r và chỉ mục tập tin i liên quan đến thời gian trình diễn, được xác định là $ts(r, i)$, (c) URI tập tin cho tập tin/phân đoạn với id dạng biểu diễn r và chỉ mục tập tin i , ký hiệu là FileURI (r, i).

Theo một phương án, thời gian bắt đầu của tập tin và các URI tập tin có thể được cung cấp một cách rõ ràng cho dạng biểu diễn. Theo phương án khác, danh sách các URI tập tin có thể được cung cấp một cách rõ ràng trong đó mỗi URI tập tin nhận chỉ mục i được gán theo vị trí trong danh sách và thời gian bắt đầu của phân đoạn này được suy ra là tổng của tất cả các thời lượng phân đoạn cho các phân đoạn từ 1 đến $i - 1$. Thời lượng của mỗi phân đoạn có thể được cung cấp theo quy tắc bất kỳ được mô tả ở trên. Ví dụ, người có hiểu biết trung bình trong toán học cơ bản có thể sử dụng các phương pháp khác để suy ra phương pháp để dễ dàng suy ra thời gian bắt đầu từ một phân tử duy nhất hoặc thuộc tính duy nhất và vị trí/chỉ mục của URI tập tin trong dạng biểu diễn.

Nếu quy tắc xây dựng URI động được cung cấp trong MPD, thì thời gian bắt đầu của mỗi tập tin và mỗi URI tập tin có thể được xây dựng động bằng cách sử dụng quy tắc xây dựng, chỉ mục của tập tin được yêu cầu và có khả năng một số tham số bổ sung được cung cấp trong mô tả trình diễn phương tiện. Thông tin này có thể, ví dụ, được cung cấp trong các thuộc tính MPD và các phân tử, chẳng hạn như, FileURIPattern và FileInfoDynamic. FileURIPattern cung cấp thông tin về cách xây dựng URI dựa trên số thứ tự chỉ mục tập tin i và id dạng biểu diễn r . FileURIFormat được xây dựng như sau:

```
FileURIFormat = sprintf ("% s% s% s% s% s.% s", BaseURI, BaseFileName,
RepresentationIDFormat, SeparatorFormat,
FileSequenceIDFormat, FileExtension);
```

và FileURI (r, i) được xây dựng như sau:

```
FileURI ( $r, i$ ) = sprintf (FileURIFormat,  $r, i$ );
```

Thời gian bắt đầu tương đối $ts(r, i)$ cho mỗi tập tin/phân đoạn có thể được suy ra từ thuộc tính nào đó chứa trong MPD mô tả thời lượng của phân đoạn trong dạng biểu diễn này, ví dụ như thuộc tính FileInfoDynamic. MPD cũng có thể chứa chuỗi các thuộc tính FileInfoDynamic mà chung cho tất cả các dạng biểu diễn trong phần trình diễn phương tiện hoặc ít nhất là cho tất cả các dạng biểu diễn trong một khoảng thời gian theo cùng cách như đã nêu ở trên. Nếu dữ liệu phương tiện cho thời gian phát cụ thể tP của dạng biểu diễn r được yêu cầu, chỉ mục tương ứng $i(r, tP)$ có thể được suy ra là $i(r, tP)$ sao cho thời gian phát của chỉ mục này nằm trong khoảng ở giữa thời gian

bắt đầu của $ts(r, i(r, tP))$ và $ts(r, i(r, tP) + 1)$. Việc truy cập phân đoạn có thể bị hạn chế thêm bởi các trường hợp trên, ví dụ như phân đoạn này không thể truy cập được.

Để truy cập thời gian phát tP chính xác một khi chỉ mục và URI của phân đoạn tương ứng được thu nhận phụ thuộc vào định dạng phân đoạn thực tế. Trong ví dụ này, không mất tính tổng quát giả định rằng các phân đoạn phương tiện có dòng thời gian cục bộ bắt đầu từ 0. Để truy cập và trình diễn dữ liệu ở thời gian phát tP , máy khách có thể tải về dữ liệu tương ứng với thời gian cục bộ từ tập tin/phân đoạn mà có thể được truy cập thông qua URI FileURI (r, i) với $i = i(r, t_p)$.

Nói chung, máy khách có thể tải về toàn bộ tập tin và sau đó có thể truy cập vào thời gian phát tP . Tuy nhiên, không nhất thiết tất cả thông tin của tập tin 3GP cần phải được tải về, do tập tin 3GP cung cấp các cấu trúc để ánh xạ thời gian cục bộ đến dãy byte. Vì vậy, chỉ có dãy byte cụ thể để truy cập thời gian phát tP có thể đủ để phát phương tiện miễn là thông tin truy cập ngẫu nhiên đầy đủ có sẵn. Ngoài ra thông tin đầy đủ về cấu trúc và ánh xạ của dãy byte và thời gian cục bộ của phân đoạn phương tiện có thể được cung cấp trong phần đầu của phân đoạn, ví dụ bằng cách sử dụng chỉ mục phân đoạn. Bằng cách truy cập vào phần đầu, ví dụ 1200 byte của phân đoạn, máy khách có thể có đủ thông tin để truy cập trực tiếp vào dãy byte cần thiết cho thời gian phát tP .

Trong một ví dụ khác, giả sử rằng chỉ mục phân đoạn, có thể được chỉ rõ là hộp "tidx" như dưới đây, có thể được sử dụng để xác định độ dịch byte của đoạn hay các đoạn yêu cầu. Các yêu cầu GET một phần có thể được tạo thành cho đoạn hay các đoạn yêu cầu. Có các phương án khác, ví dụ, máy khách có thể phát hành yêu cầu tiêu chuẩn cho tập tin và hủy bỏ nó khi hộp "tidx" thứ nhất đã được nhận.

Tìm kiếm

Máy khách có thể cố gắng tìm thời gian trình diễn cụ thể tp trong dạng biểu diễn. Dựa trên MPD, máy khách có quyền truy cập vào thời gian bắt đầu phân đoạn phương tiện và URL phân đoạn phương tiện của từng phân đoạn trong dạng biểu diễn. Máy khách có thể nhận được `segment_index` chỉ mục phân đoạn của phân đoạn mà có nhiều khả năng nhất chứa các mẫu phương tiện cho thời gian trình diễn tp như là chỉ mục phân đoạn tối đa i , mà thời gian bắt đầu $ts(r, i)$ nhỏ hơn hoặc bằng với thời gian

trình diễn tp tức là $\text{segment_index} = \max \{i | tS(r, i) \leq tp\}$. URL phân đoạn được thu nhận là $\text{FielURI}(r, i)$.

Lưu ý rằng thông tin thời gian trong MPD có thể là gần đúng, do các vấn đề liên quan đến việc bố trí các điểm truy cập ngẫu nhiên, đồng chỉnh các rãnh phương tiện và độ dịch thời gian phương tiện. Kết quả là, phân đoạn được xác định bởi các thủ tục trên có thể bắt đầu tại thời điểm sau một chút so với tp và dữ liệu phương tiện cho thời gian trình diễn tp có thể ở trong phân đoạn phương tiện trước đó. Trong trường hợp tìm kiếm, thời gian tìm kiếm có thể được cập nhật để tương đương với thời gian mẫu thứ nhất của tập tin được lấy ra, hoặc tập tin trước đó có thể được lấy ra để thay thế. Tuy nhiên, lưu ý rằng trong thời gian phát ra liên tục, kể cả các trường hợp có sự chuyển đổi giữa các dạng biểu diễn/phiên bản thay thế, dữ liệu phương tiện cho thời gian giữa tp và điểm bắt đầu phân đoạn lấy được vẫn có sẵn.

Để tìm kiếm chính xác thời gian trình diễn tp, máy khách truyền liên tục HTTP cần truy cập vào điểm truy cập ngẫu nhiên (RAP). Để xác định điểm truy cập ngẫu nhiên trong phân đoạn phương tiện trong trường hợp truyền liên tục HTTP thích ứng 3GPP, máy khách có thể, ví dụ, sử dụng thông tin trong hộp 'tidx' hoặc 'sidx', nếu có, để xác định vị trí điểm truy cập ngẫu nhiên và thời gian trình diễn tương ứng trong phần trình diễn phương tiện. Trong trường hợp phân đoạn là đoạn phim 3GPP, máy khách có thể sử dụng thông tin trong hộp 'moof' và 'mdat', ví dụ, để xác định vị trí các RAP và thu được thời gian trình diễn cần thiết từ thông tin trong đoạn phim và thời gian bắt đầu phân đoạn được suy ra từ MPD. Nếu không có sẵn RAP với thời gian trình diễn trước thời gian trình diễn được yêu cầu tp, máy khách có thể truy cập vào phân đoạn trước đó hoặc chỉ sử dụng điểm truy cập ngẫu nhiên thứ nhất làm kết quả tìm kiếm. Khi phân đoạn phương tiện bắt đầu với RAP, các thủ tục này rất đơn giản.

Cũng lưu ý rằng không nhất thiết là tất cả thông tin của phân đoạn phương tiện cần phải được tải về để truy cập thời gian trình diễn tp. Máy khách có thể, ví dụ, ban đầu yêu cầu hộp 'tidx' hoặc 'sidx' từ chỗ bắt đầu của phân đoạn phương tiện bằng cách sử dụng các yêu cầu dãy byte. Bằng cách sử dụng các hộp 'tidx' hoặc 'sidx', việc định thời phân đoạn có thể được ánh xạ đến các dãy byte của phân đoạn. Bằng cách liên tục sử dụng yêu cầu HTTP một phần, chỉ các phần có liên quan của phân đoạn phương tiện cần được truy cập, để cải thiện trải nghiệm người dùng và có độ trễ khởi động thấp.

Tạo danh sách phân đoạn

Phần trên đã mô tả rõ ràng cách cài đặt máy khách truyền liên tục HTTP đơn giản mà sử dụng thông tin được cung cấp bởi MPD để tạo ra danh sách phân đoạn cho dạng biểu diễn có thời lượng phân đoạn gần đúng được báo hiệu là *dur*. Theo một số phương án, máy khách có thể gán cho các phân đoạn trong phương tiện trong dạng biểu diễn các chỉ mục liên tiếp $i = 1, 2, 3, \dots$, tức là phân đoạn phương tiện thứ nhất được gán chỉ mục $i = 1$, phân đoạn phương tiện thứ hai được gán chỉ mục $i = 2$, v.v.. Sau đó, danh sách phân đoạn phương tiện với các chỉ mục phân đoạn i được gán $startTime[i]$ và $URL[i]$ được tạo ra, ví dụ như sau. Đầu tiên, chỉ mục i được thiết lập là 1. Thời gian bắt đầu của phân đoạn phương tiện thứ nhất được thu nhận là 0, $startTime[1] = 0$. URL của phân đoạn phương tiện i , $URL[i]$, được thu nhận là $FileURI(r, i)$. Quy trình này được tiếp tục cho tất cả các phân đoạn phương tiện được mô tả với chỉ mục i và $startTime[i]$ của phân đoạn phương tiện i được thu nhận là $(i - 1) * dur$ và $URL[i]$, được thu nhận là $FileURI(r, i)$.

Các yêu cầu HTTP/TCP đồng thời

Mối quan tâm trong hệ thống truyền liên tục yêu cầu theo khối là mong muốn luôn luôn yêu cầu các khối có chất lượng cao nhất mà hoàn toàn có thể được nhận kịp thời để phát ra. Tuy nhiên, tốc độ dữ liệu đến có thể không được biết trước và vì vậy có thể xảy ra khả năng khối được yêu cầu không đến kịp thời để được phát ra. Điều này dẫn đến cần phải tạm dừng phát ra phương tiện, và dẫn đến trải nghiệm người dùng kém. Vấn đề này có thể được giảm nhẹ bằng các thuật toán của máy khách mà sử dụng phương pháp bảo toàn đối với việc chọn các khối để yêu cầu bằng cách yêu cầu các khối chất lượng thấp hơn (và do đó có kích thước nhỏ hơn) mà có nhiều khả năng được nhận kịp thời, ngay cả khi tốc độ dữ liệu đến giảm đi trong quá trình nhận khối. Tuy nhiên phương pháp bảo toàn này có nhược điểm là có thể cung cấp việc phát ra chất lượng thấp hơn cho người dùng hoặc thiết bị đích, cũng chính là trải nghiệm người dùng kém. Vấn đề này có thể được khuếch đại khi nhiều kết nối HTTP được sử dụng cùng lúc để tải về các khối khác nhau, như được mô tả dưới đây, do tài nguyên mạng có sẵn được chia sẻ trên các kết nối và do đó đang được sử dụng đồng thời cho các khối với thời gian phát ra khác nhau.

Có thể có lợi nếu máy khách để phát hành các yêu cầu cho nhiều khối đồng thời, trong ngữ cảnh này "đồng thời" có nghĩa là phản hồi đối với các yêu cầu đang xảy ra trong khoảng thời gian chùng lún, và đây không nhất thiết là trường hợp mà các yêu cầu được đưa ra vào chính xác cùng lúc hoặc thậm chí gần như cùng lúc. Trong trường hợp giao thức HTTP, phương pháp này có thể cải thiện việc sử dụng băng thông có sẵn do hoạt động của giao thức TCP (như đã biết). Điều này có thể đặc biệt quan trọng để cải thiện thời gian sửa đổi nội dung, như khi nội dung mới được yêu cầu lần thứ nhất các kết nối HTTP/TCP tương ứng mà trên đó dữ liệu cho các khối được yêu cầu có thể chậm bắt đầu, và do đó việc sử dụng một vài kết nối HTTP/TCP vào thời điểm này có thể tăng mạnh thời gian phân phối dữ liệu của các khối thứ nhất. Tuy nhiên, việc yêu cầu các khối hoặc đoạn khác nhau trên các kết nối HTTP/TCP khác nhau cũng có thể dẫn đến suy giảm hiệu năng, do các yêu cầu đối với các khối sẽ được phát ra lần thứ nhất cạnh tranh với các yêu cầu cho các khối tiếp theo, việc tải về HTTP/TCP cạnh tranh khác nhau rất nhiều tốc độ về phân phối và do đó thời gian hoàn thành yêu cầu có thể rất khác nhau, và thường là không thể kiểm soát được việc tải về HTTP/TCP nào sẽ hoàn toàn nhanh chóng và việc tải về HTTP/TCP nào sẽ chậm hơn, và do đó có khả năng là ít nhất một số lần việc tải về HTTP/TCP của một số ít khối thứ nhất sẽ hoàn thành cuối cùng, do đó dẫn đến thời gian sửa đổi kênh lớn và thay đổi.

Giả sử mỗi khối hoặc đoạn của phân đoạn được tải về qua kết nối HTTP/TCP riêng biệt, và rằng số lượng kết nối song song là n và thời lượng phát ra của mỗi khối là t giây, và tốc độ truyền liên tục nội dung gắn với phân đoạn này là S . Khi máy khách lần thứ nhất bắt đầu truyền liên tục nội dung, yêu cầu có thể được phát hành cho n khối thứ nhất, biểu diễn $n * t$ giây dữ liệu phương tiện.

Như đã biết bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật, có sự khác nhau lớn trong tốc độ dữ liệu của các kết nối TCP. Tuy nhiên, để đơn giản hóa cho việc mô tả, giả sử một cách lý tưởng là tất cả các kết nối tiến hành song song sao cho khối thứ nhất sẽ được nhận hoàn toàn vào cùng lúc với $n - 1$ khối khác được yêu cầu. Để đơn giản hóa việc mô tả, giả sử rằng băng thông tổng hợp được sử dụng bởi n kết nối tải về được cố định ở giá trị B cho toàn bộ thời gian tải về, và tốc độ truyền liên tục S không thay đổi trên toàn bộ dạng biểu diễn. Giả sử thêm rằng cấu trúc dữ liệu phương tiện là cấu trúc sao cho việc phát ra khối có thể được thực hiện khi toàn bộ

khối có sẵn tại máy khách, tức là, việc phát ra khối chỉ có thể bắt đầu sau khi toàn bộ khối được nhận, ví dụ, do cấu trúc mã hóa video cơ bản, hoặc bởi vì quy trình mã hóa đang được sử dụng để mã hóa mỗi đoạn hoặc khối một cách riêng biệt, và do đó toàn bộ đoạn hoặc khối cần phải được nhận trước khi nó có thể được giải mã. Vì vậy, để đơn giản hóa phần mô tả dưới đây, giả sử rằng toàn bộ khối cần phải được nhận trước khi bất kỳ phần nào của khối có thể được phát ra. Sau đó, thời gian cần thiết trước khi khối thứ nhất đến và có thể được phát ra là xấp xỉ $n * t * S/B$.

Vì mong muốn giảm thiểu thời gian sửa đổi nội dung, do đó mong muốn giảm thiểu $n * t * S/B$. Giá trị của t có thể được xác định bởi các yếu tố như cấu trúc mã hóa video cơ bản và cách thức phương pháp thu thập được sử dụng, và do đó t có thể nhỏ hợp lý, nhưng giá trị rất nhỏ của t dẫn đến ánh xạ phân đoạn quá phức tạp và có thể có thể không tương thích với quy trình mã hóa và giải mã video hiệu quả, nếu được sử dụng. Giá trị của n cũng có thể ảnh hưởng đến giá trị của B , tức là, B có thể lớn hơn khi số lượng kết nối n lớn hơn, và do đó việc giảm số lượng kết nối, n , có tác dụng tiêu cực đến khả năng làm giảm lượng băng thông có sẵn được sử dụng, B , và như vậy có thể không có hiệu quả trong việc đạt được mục tiêu làm giảm thời gian sửa đổi nội dung. Giá trị của S phụ thuộc vào việc dạng biểu diễn nào được chọn để tải về và phát ra, và một cách lý tưởng S nên càng gần B càng tốt để tối đa hóa chất lượng phát ra của phương tiện với các điều kiện mạng cho trước. Vì vậy, để đơn giản hóa việc mô tả, giả sử rằng S xấp xỉ bằng B . Sau đó, thời gian sửa đổi kênh tỷ lệ thuận với $n * t$. Vì vậy, việc sử dụng nhiều kết nối hơn để tải về các đoạn khác nhau có thể làm giảm thời gian sửa đổi kênh nếu băng thông tổng hợp được sử dụng bởi các kết nối tỷ lệ thuận tuyến tính dưới với số lượng kết nối, điều này thường xảy ra.

Ví dụ, giả sử $t = 1$ giây, và với $n = 1$ giá trị của $B = 500$ Kbps, và với $n = 2$ giá trị của $B = 700$ Kbps, và với $n = 3$ giá trị của $B = 800$ Kbps. Giả sử rằng dạng biểu diễn với $S = 700$ Kbps được chọn. Thì, với $n = 1$ thời gian tải về cho khối thứ nhất là $1 * 700/500 = 1,4$ giây, với $n = 2$ thời gian tải về cho khối thứ nhất là $2 * 700/700 = 2$ giây, và với $n = 3$ thời gian tải về cho khối thứ nhất là $3 * 700/800 = 2,625$ giây. Hơn nữa, khi số lượng kết nối tăng, sự biến thiên trong tốc độ tải riêng biệt của các kết nối có khả năng tăng (mặc dù thậm chí với một kết nối có thể sẽ có sự biến thiên đáng kể nào đó). Như vậy, trong ví dụ này, thời gian sửa đổi kênh và sự biến thiên trong thời gian sửa đổi kênh tăng lên khi số lượng kết nối tăng. Bằng trực giác, các khối đang

được phân phối có các mức ưu tiên khác nhau, tức là, khối thứ nhất có thời hạn phân phối sớm nhất, khối thứ hai có thời hạn phân phối sớm thứ hai, v.v., trong khi đó các kết nối tải về mà trên đó các khối đang được phân phối đang cạnh tranh tài nguyên mạng trong khi phân phối, và do đó các khối với thời hạn sớm nhất trở thành chậm hơn khi nhiều khối cạnh tranh hơn được yêu cầu. Mặt khác, ngay cả trong trường hợp này, cuối cùng việc sử dụng nhiều hơn một kết nối tải về cho phép hỗ trợ tốc độ truyền cao hơn một cách bền vững, ví dụ, với ba kết nối với tốc độ truyền liên tục lên đến 800Kbps có thể được hỗ trợ trong ví dụ này, trong khi chỉ có một dòng 500Kbps có thể được hỗ trợ với một kết nối.

Trong thực tế, như đã mô tả ở trên, tốc độ dữ liệu của kết nối có thể rất khác nhau cả trong cùng một kết nối theo thời gian và giữa các kết nối và kết quả là, n khối yêu cầu thường không hoàn thành cùng một lúc và trong thực tế thường có thể là trường hợp một khối có thể hoàn thành trong thời gian bằng một nửa so với khối khác. Điều này dẫn đến hoạt động không thể đoán trước vì trong một số trường hợp khối thứ nhất có thể hoàn thành sớm hơn nhiều so với các khối khác và trong các trường hợp khác, khối thứ nhất có thể hoàn thành muộn hơn nhiều so với các khối khác, và kết quả là việc bắt đầu phát ra có thể, trong một số trường hợp, xảy ra tương đối nhanh và trong trường hợp khác có thể xảy ra chậm. Hoạt động không thể đoán trước này có thể gây khó chịu cho người dùng và do đó có thể được coi là một trải nghiệm người dùng kém.

Do đó, điều cần thiết là có các phương pháp mà trong đó nhiều kết nối TCP có thể được sử dụng để cải thiện thời gian sửa đổi kênh và sự biến thiên trong thời gian sửa đổi kênh, trong khi đồng thời hỗ trợ tốc độ truyền liên tục chất lượng tốt có thể. Điều cần thiết là có các phương pháp mà cho phép chia sẻ băng thông có sẵn được phân bổ cho mỗi khối để được điều chỉnh khi thời gian phát ra của khối đang đến, do đó, nếu cần thiết, một phần lớn của băng thông có sẵn có thể được phân bổ cho khối với thời gian phát ra gần nhất.

Yêu cầu HTTP/TCP phối hợp

Dưới đây mô tả phương pháp sử dụng các yêu cầu HTTP/TCP đồng thời theo kiểu phối hợp. Bộ thu có thể sử dụng nhiều yêu cầu HTTP/TCP phối hợp đồng thời, ví dụ như sử dụng nhiều yêu cầu dây byte HTTP, trong đó mỗi yêu cầu là cho một phần

của đoạn trong phân đoạn nguồn, hoặc toàn bộ đoạn của phân đoạn nguồn, hoặc một phân hoặc đoạn sửa chữa của phân đoạn sửa chữa, hoặc cho toàn bộ đoạn sửa chữa của phân đoạn sửa chữa.

Các ưu điểm của yêu cầu HTTP/TCP phối hợp cùng với việc sử dụng dữ liệu sửa chữa FEC có thể đặc biệt quan trọng để cung cấp thời gian sửa đổi kênh nhanh thống nhất. Ví dụ, tại thời gian sửa đổi kênh có khả năng là các kết nối TCP vừa được bắt đầu hoặc đã rồi trong một khoảng thời gian, trong trường hợp đó cửa sổ tắc nghẽn, cwnd, có giá trị tối thiểu cho các kết nối, và do đó tốc độ phân phối của các kết nối TCP này sẽ mất một số thời gian trọn vòng (RTT) để tăng tốc, và sẽ có sự biến thiên lớn về tốc độ phân phối trên các kết nối TCP khác nhau trong thời gian tăng tốc này.

Tổng quan về phương pháp không FEC sẽ được mô tả dưới đây, phương pháp này là phương pháp yêu cầu HTTP/TCP phối hợp trong đó chỉ có dữ liệu phương tiện của khối nguồn được yêu cầu bằng cách sử dụng nhiều kết nối HTTP/TCP đồng thời, tức là, không có dữ liệu sửa chữa FEC được yêu cầu. Với phương pháp không FEC, các phần của cùng đoạn được yêu cầu trên các kết nối khác nhau, ví dụ, bằng cách sử dụng yêu cầu dãy byte HTTP cho các phần của đoạn, và do đó ví dụ mỗi yêu cầu dãy byte HTTP là dành cho một phần của dãy byte được chỉ ra trong ánh xạ phân đoạn cho đoạn đó. Đây có thể là trường hợp mà yêu cầu HTTP/TCP riêng biệt tăng tốc độ phân phối để tận dụng toàn bộ băng thông có sẵn trên một số RTT, và do đó có khoảng thời gian tương đối dài trong đó tốc độ phân phối nhỏ hơn băng thông có sẵn, và do đó nếu một kết nối HTTP/TCP duy nhất được sử dụng để tải về, ví dụ, đoạn thứ nhất của nội dung được phát ra, thời gian sửa đổi kênh có thể rất lớn. Sử dụng phương pháp không FEC, việc tải về các phần khác nhau của cùng đoạn trên các kết nối HTTP/TCP khác nhau có thể làm giảm đáng kể thời gian sửa đổi kênh.

Tổng quan về phương pháp FEC sẽ được mô tả dưới đây, đó là phương pháp yêu cầu HTTP/TCP phối hợp trong đó dữ liệu phương tiện của phân đoạn nguồn và dữ liệu sửa chữa FEC được tạo ra từ dữ liệu phương tiện được yêu cầu bằng cách sử dụng nhiều kết nối HTTP/TCP đồng thời. Với phương pháp FEC, các phần của cùng đoạn và dữ liệu sửa chữa FEC được tạo ra từ đoạn đó được yêu cầu trên các kết nối khác nhau, sử dụng yêu cầu dãy byte HTTP cho các phần của đoạn này, và do đó ví dụ mỗi yêu cầu dãy byte HTTP là cho một phần của dãy byte được chỉ ra trong ánh xạ phân đoạn cho đoạn đó. Đây có thể là trường hợp mà yêu cầu HTTP/TCP riêng biệt tăng tốc

độ phân phối để tận dụng toàn bộ băng thông có sẵn trên một số RTT, và do đó có khoảng thời gian tương đối dài trong đó tốc độ phân phối nhỏ hơn băng thông có sẵn, và do đó nếu một kết nối HTTP/TCP duy nhất được sử dụng để tải về ví dụ đoạn thứ nhất của một nội dung được phát ra, thời gian sửa đổi kênh có thể rất lớn. Việc sử dụng phương pháp FEC có các ưu điểm giống như phương pháp không FEC, và có ưu điểm bổ sung là không phải tất cả dữ liệu được yêu cầu cần phải đến trước khi đoạn có thể được khôi phục, do đó tiếp tục làm giảm thời gian sửa đổi kênh và sự biến thiên trong thời gian sửa đổi kênh. Bằng cách đưa ra các yêu cầu trên các kết nối TCP khác nhau, và yêu cầu cao hơn cũng bằng cách yêu cầu dữ liệu sửa chữa FEC trên ít nhất một trong số các kết nối, lượng thời gian cần thiết để cung cấp đủ lượng dữ liệu để, ví dụ, khôi phục phân đoạn được yêu cầu thứ nhất mà cho phép bắt đầu phát lại phương tiện, có thể được giảm đi rất nhiều và được làm cho phù hợp hơn nếu kết nối TCP phối hợp và dữ liệu sửa chữa FEC không được sử dụng.

Các hình vẽ từ Fig.24(a) – Fig.24(e) thể hiện ví dụ về biến động tốc độ phân phối của 5 kết nối TCP chạy trên cùng một liên kết đến cùng máy khách từ cùng máy chủ web HTTP của mạng EVDO (tối ưu hóa dữ liệu phát triển mô phỏng). Trên Fig.24(a) – Fig.24(e), trục X thể hiện thời gian theo giây, và trục Y thể hiện tốc độ mà các bit được nhận tại máy khách trên mỗi trong số 5 kết nối TCP được đo trên các khoảng thời gian 1 giây cho mỗi kết nối. Trong mô phỏng cụ thể này, có tổng số 12 kết nối TCP chạy trên liên kết này, và do đó mạng được tải tương đối trong thời gian được thể hiện, điều này có thể là điển hình khi nhiều hơn một máy khách đang truyền liên tục trong cùng một ô của mạng di động. Lưu ý rằng mặc dù tốc độ phân phối có phần nào tương quan theo thời gian, có các khác biệt lớn về tốc độ phân phối của 5 kết nối ở nhiều thời điểm.

Fig.25 cho thấy cấu trúc yêu cầu có thể có cho đoạn có kích thước 250.000 bit (khoảng 31,25 KB) trong đó có 4 yêu cầu dãy byte HTTP được thực hiện song song cho các phần khác nhau của đoạn này, tức là, kết nối HTTP thứ nhất yêu cầu 50.000 bit thứ nhất, kết nối HTTP thứ hai yêu cầu 50.000 bit tiếp theo, kết nối HTTP thứ ba yêu cầu 50.000 bit tiếp theo, và kết nối HTTP thứ tư yêu cầu 50.000 bit tiếp theo. Nếu FEC không được sử dụng, tức là phương pháp không FEC, thì chỉ có 4 yêu cầu cho đoạn trong ví dụ này. Nếu FEC được sử dụng, tức là phương pháp FEC, thì trong ví dụ

này có kết nối HTTP bổ sung mà yêu cầu thêm 50.000 bit dữ liệu sửa chữa FEC của phân đoạn sửa chữa được tạo ra từ đoạn này.

Fig.26 là sơ đồ phóng đại về vài giây thứ nhất của 5 kết nối TCP được thể hiện trên Fig.24(a) – Fig.24(e), trong đó trên Fig.26, trục X thể hiện thời gian theo các khoảng 100 mili giây, và trục Y thể hiện tốc độ mà bit được nhận tại máy khách trên mỗi trong số 5 kết nối TCP được đo theo các khoảng 100 mili giây. Một đường thể hiện tổng lượng bit đã được nhận ở máy khách cho đoạn từ 4 kết nối HTTP thứ nhất (không bao gồm kết nối HTTP mà dữ liệu FEC được yêu cầu trên đó), tức là, lượng bit đến bằng cách sử dụng phương pháp không FEC. Đường khác thể hiện tổng lượng bit đã được nhận tại máy khách cho đoạn từ tất cả 5 kết nối HTTP (bao gồm kết nối HTTP mà dữ liệu FEC được yêu cầu trên đó), tức là, lượng bit đến bằng cách sử dụng phương pháp FEC. Đối với phương pháp FEC, giả sử rằng đoạn có thể được giải mã FEC từ khi nhận bit 200.000 bit bất kỳ trong số 250.000 bit được yêu cầu, mà có thể được thực hiện nếu, ví dụ như mã FEC Reed - Solomon được sử dụng, và có thể về cơ bản được thực hiện nếu, ví dụ, mã RaptorQ được mô tả trong Luby IV được sử dụng. Đối với phương pháp FEC trong ví dụ này, đủ dữ liệu được nhận để khôi phục đoạn bằng cách sử dụng kỹ thuật giải mã FEC sau 1 giây, cho phép thời gian sửa đổi kênh là 1 giây (giả định rằng dữ liệu cho các đoạn sau đó có thể được yêu cầu và được nhận trước khi đoạn thứ nhất được phát ra toàn bộ). Đối với phương pháp không FEC trong ví dụ này, tất cả dữ liệu cho 4 yêu cầu phải được nhận trước khi đoạn có thể được khôi phục, điều này xảy ra sau 1,7 giây, dẫn đến thời gian sửa đổi kênh là 1,7 giây. Vì vậy, trong ví dụ được thể hiện trên Fig.26, phương pháp không FEC tệ hơn 70% về thời gian sửa đổi kênh so với phương pháp FEC. Một trong số các lý do cho ưu điểm được thể hiện theo phương pháp FEC trong ví dụ này là, đối với phương pháp FEC, việc nhận 80% dữ liệu được yêu cầu bất kỳ cho phép khôi phục đoạn, trong khi đối với phương pháp không FEC, việc nhận 100% dữ liệu được yêu cầu là cần thiết. Như vậy, phương pháp không FEC phải chờ đợi kết nối TCP chậm nhất hoàn thành việc phân phối, và bởi vì các thay đổi tự nhiên trong tốc độ phân phối TCP có khuynh hướng là thay đổi lớn trong tốc độ phân phối của kết nối TCP chậm nhất so với kết nối TCP trung bình. Với phương pháp FEC trong ví dụ này, một kết nối TCP chậm không xác định thời điểm khi mà đoạn có thể được khôi phục. Thay vào đó, đối với phương pháp

FEC, việc cung cấp đủ dữ liệu phần nhiều là chức năng tốc độ phân phối TCP trung bình hơn là tốc độ phân phối TCP trong trường hợp tệ hơn.

Có nhiều biến thể của phương pháp không FEC và phương pháp FEC được mô tả ở trên. Ví dụ, các yêu cầu HTTP/TCP phối hợp có thể được sử dụng chỉ cho một số đoạn đầu tiên sau khi thời gian sửa đổi kênh đã diễn ra, và sau đó chỉ một yêu cầu HTTP/TCP duy nhất được sử dụng để tải về các đoạn khác, nhiều đoạn, hoặc toàn bộ phân đoạn. Theo ví dụ khác, số lượng kết nối HTTP/TCP phối hợp được sử dụng có thể là hàm của cả tính cấp bách của các đoạn đang được yêu cầu, tức là, khả năng sắp xảy ra của thời gian phát ra các đoạn này, và của các điều kiện mạng hiện thời.

Trong một số biến thể, các kết nối HTTP có thể được sử dụng để yêu cầu dữ liệu từ sửa chữa từ các phân đoạn sửa chữa. Trong các biến thể khác, lượng dữ liệu khác nhau có thể được yêu cầu trên các kết nối HTTP khác nhau, ví dụ, phụ thuộc vào kích thước hiện thời của bộ đệm phương tiện và tốc độ nhận dữ liệu ở máy khách. Trong biến thể khác, dạng biểu diễn nguồn không độc lập với nhau, mà thay vào đó biểu diễn phân mã hóa phương tiện theo lớp, trong đó ví dụ như dạng biểu diễn nguồn cải tiến có thể phụ thuộc vào dạng biểu diễn nguồn cơ sở. Trong trường hợp này, có thể có dạng biểu diễn sửa chữa tương ứng với dạng biểu diễn nguồn, và dạng biểu diễn sửa chữa khác tương ứng với sự kết hợp của dạng biểu diễn nguồn cơ sở và cải tiến.

Toàn bộ các phần tử khác bổ sung thêm các ưu điểm có thể được thực hiện bằng các phương pháp được bộc lộ ở trên. Ví dụ, số lượng kết nối HTTP được sử dụng có thể thay đổi tùy thuộc vào lượng phương tiện hiện thời trong bộ đệm phương tiện, và/hoặc tốc độ nhận vào bộ đệm phương tiện. Các yêu cầu HTTP phối hợp sử dụng FEC, tức là, phương pháp FEC mô tả ở trên và các biến thể của phương pháp này, có thể được sử dụng linh hoạt khi bộ đệm phương tiện tương đối trống, ví dụ, nhiều yêu cầu HTTP phối hợp hơn được thực hiện song song cho các phần khác nhau của đoạn thứ nhất, yêu cầu tất cả đoạn nguồn và một phần tương đối lớn dữ liệu sửa chữa từ đoạn sửa chữa tương ứng, và sau đó chuyển sang số lượng yêu cầu HTTP đồng thời ít hơn, yêu cầu các phần lớn hơn của dữ liệu phương tiện trên mỗi yêu cầu, và yêu cầu một phần nhỏ hơn của dữ liệu sửa chữa, ví dụ như chuyển sang 1, 2 hoặc 3 yêu cầu HTTP đồng thời, chuyển sang thực hiện các yêu cầu cho các đoạn đầy đủ hoặc nhiều đoạn liên tiếp trên mỗi yêu cầu, và chuyển sang không yêu cầu dữ liệu sửa chữa, khi bộ đệm phương tiện phát triển.

Theo ví dụ khác, lượng dữ liệu sửa chữa FEC có thể thay đổi như là hàm của kích thước bộ đệm phương tiện, tức là, khi bộ đệm phương tiện nhỏ, thì nhiều dữ liệu sửa chữa FEC hơn có thể được yêu cầu, và khi bộ đệm phương tiện phát triển thì lượng dữ liệu sửa chữa FEC được yêu cầu có thể giảm bớt, và tại một số điểm khi bộ đệm phương tiện đủ lớn, thì có thể không yêu cầu dữ liệu sửa chữa FEC, mà chỉ có dữ liệu từ phân đoạn nguồn của dạng biểu diễn nguồn. Các lợi ích của kỹ thuật cải tiến là chúng có thể cho phép thời gian sửa đổi kênh nhanh hơn và nhất quán hơn, và khả năng bền bỉ hơn khi có hiện tượng phương tiện bị giật hoặc đứng im, trong khi đồng thời giảm thiểu lượng băng thông bổ sung được sử dụng vượt quá lượng băng thông mà sẽ chỉ được sử dụng bởi quy trình phân phối phương tiện trong các phân đoạn nguồn bằng cách giảm cả lưu lượng thông báo yêu cầu và dữ liệu sửa chữa FEC, trong khi đồng thời cho phép hỗ trợ phương tiện tốc độ cao nhất có thể với các điều kiện mạng cho trước.

Các cải tiến khác khi sử dụng kết nối HTTP đồng thời

Yêu cầu HTTP/TCP có thể được hủy bỏ nếu điều kiện thích hợp được đáp ứng và yêu cầu HTTP/TCP khác có thể được thực hiện để tải về dữ liệu mà có thể thay thế dữ liệu được yêu cầu trong yêu cầu bị hủy bỏ, trong đó yêu cầu HTTP/TCP thứ hai có thể yêu cầu chính xác cùng dữ liệu như trong yêu cầu ban đầu, ví dụ, dữ liệu nguồn; hoặc dữ liệu chồng lấn, ví dụ như, một số dữ liệu nguồn và dữ liệu sửa chữa tương tự mà không được yêu cầu trong yêu cầu thứ nhất; hoặc dữ liệu hoàn toàn tách rời nhau, ví dụ, dữ liệu sửa chữa mà không được yêu cầu trong yêu cầu thứ nhất. Ví dụ về điều kiện thích hợp đó là yêu cầu không thành công do sự vắng mặt của phản hồi từ BSI (cơ sở hạ tầng máy chủ khối) trong thời gian được cung cấp hoặc sự cố trong việc thiết lập kết nối truyền tải đến BSI hoặc nhận được thông báo lỗi rõ ràng từ máy chủ hoặc điều kiện sự cố khác.

Ví dụ khác về điều kiện thích hợp đó là việc nhận dữ liệu đang được tiến hành chậm một cách bất thường, theo sự so sánh số đo của tốc độ kết nối (tốc độ dữ liệu đến để phản hồi yêu cầu đang quan tâm) với tốc độ kết nối dự kiến hoặc với ước lượng tốc độ kết nối cần thiết để nhận được phản hồi trước thời gian phát ra của dữ liệu phương tiện chứa trong đó hoặc thời điểm khác phụ thuộc vào thời điểm đó.

Phương pháp này có ưu điểm trong trường hợp mà BSI đôi khi bộc lộ sự cố hoặc hiệu suất kém. Trong trường hợp này phương pháp trên làm tăng xác suất mà máy khách có thể tiếp tục hoạt động phát ra đáng tin cậy đối với dữ liệu phương tiện mặc dù gặp sự cố hoặc hiệu suất kém trong BSI. Lưu ý rằng trong một số trường hợp có thể có lợi nếu thiết kế BSI theo cách sao cho BSI đôi khi bộc lộ sự cố hoặc hiệu suất kém này, ví dụ thiết kế này có thể có chi phí thấp hơn so với thiết kế thay thế mà không bộc lộ sự cố hoặc hiệu suất kém như vậy hoặc bộc lộ các đặc điểm này ít thường xuyên hơn. Trong trường hợp này, phương pháp mô tả ở đây còn có thêm ưu điểm là nó cho phép sử dụng thiết kế có chi phí thấp hơn cho BSI mà sau đó không làm giảm chất lượng của trải nghiệm người dùng.

Theo phương án khác, số lượng yêu cầu được phát hành cho dữ liệu tương ứng với một khối cho trước có thể phụ thuộc vào việc một điều kiện thích hợp đối với khối đó có được đáp ứng hay không. Nếu điều kiện này không được đáp ứng thì máy khách có thể bị hạn chế không thực hiện yêu cầu thêm đối với khối nếu việc hoàn tất thành công tất cả các yêu cầu dữ liệu hiện không đầy đủ đối với khối này sẽ cho phép khôi phục khối với xác suất cao. Nếu điều kiện này được đáp ứng thì số lượng yêu cầu lớn hơn đối với khối có thể được phát hành, tức là, hạn chế nêu trên không áp dụng. Ví dụ về điều kiện thích hợp đó là thời gian cho đến thời gian phát ra được lập lịch của khối hoặc thời gian khác phụ thuộc vào việc thời gian đó nằm ở dưới ngưỡng cho trước. Phương pháp này có ưu điểm bởi vì các yêu cầu bổ sung đối với dữ liệu cho khối được phát hành khi việc nhận khối trở nên cấp bách hơn, bởi vì thời gian phát ra dữ liệu phương tiện bao gồm khối đang đến gần. Trong trường hợp của giao thức truyền tải phổ biến như HTTP/TCP, các yêu cầu bổ sung này có tác dụng làm tăng phần của băng thông có sẵn dành riêng cho dữ liệu đóng góp vào việc nhận khối đang quan tâm. Điều này làm giảm thời gian cần thiết cho việc nhận đủ dữ liệu để khôi phục khối để hoàn thành và do đó làm giảm xác suất mà khối không thể được khôi phục trước thời gian phát ra được lập lịch của dữ liệu phương tiện bao gồm khối này. Như đã mô tả ở trên, nếu khối không thể được khôi phục trước thời gian phát ra được lập lịch của dữ liệu phương tiện bao gồm khối đó thì việc phát ra có thể tạm dừng dẫn đến trải nghiệm người dùng kém và do đó có lợi nếu các phương pháp mô tả ở đây làm giảm xác suất của trải nghiệm người dùng kém này.

Nên hiểu rằng trong bản mô tả này việc tham chiếu đến thời gian phát ra được lập lịch của khối dùng để chỉ thời gian mà tại đó dữ liệu phương tiện mã hóa bao gồm khối có thể có sẵn trước tiên tại máy khách để đạt được khả năng phát ra phần trình diễn mà không cần tạm dừng. Người có hiểu biết trung bình trong lĩnh vực hệ thống trình diễn phương tiện hiểu rằng thời gian này trong thực tế xuất hiện trước một chút so với thời gian thực xuất hiện phương tiện bao gồm khối tại các bộ chuyển đổi vật lý được sử dụng để phát ra (màn hình, loa, v.v.) do một số hàm chuyển đổi có thể cần phải được áp dụng cho dữ liệu phương tiện bao gồm khối đó để thực hiện phát ra thực tế khối này và các hàm này có thể đòi hỏi lượng thời gian nhất định để hoàn thành. Ví dụ dữ liệu phương tiện thường được truyền tải ở dạng nén và quy trình biến đổi giải nén có thể được áp dụng.

Phương pháp tạo các cấu trúc tập tin hỗ trợ phương pháp HTTP/FEC phối hợp

Phương án tạo ra cấu trúc tập tin mà có thể có lợi nếu được sử dụng bởi máy khách sử dụng các phương pháp HTTP/FEC phối hợp sẽ được mô tả dưới đây. Theo phương án này, với mỗi phân đoạn nguồn có một phân đoạn sửa chữa tương ứng được tạo ra như sau. Tham số R chỉ ra trung bình bao nhiêu dữ liệu sửa chữa FEC được tạo ra cho dữ liệu nguồn trong các phân đoạn nguồn. Ví dụ, $R = 0,33$ chỉ ra rằng nếu phân đoạn nguồn chứa 1,000Kb dữ liệu, thì phân đoạn sửa chữa tương ứng chứa khoảng 330Kb dữ liệu sửa chữa. Tham số S chỉ ra kích thước ký hiệu tính theo byte được sử dụng để mã hóa và giải mã FEC. Ví dụ, $S = 64$ chỉ ra rằng dữ liệu nguồn và dữ liệu sửa chữa bao gồm các ký hiệu, mỗi ký hiệu có kích thước 64 byte cho các mục đích mã hóa và giải mã FEC.

Phân đoạn sửa chữa có thể được tạo ra cho phân đoạn nguồn như sau. Mỗi đoạn của phân đoạn nguồn được coi là khối nguồn cho mục đích mã hóa FEC, và do đó mỗi đoạn được xử lý như là chuỗi các ký hiệu nguồn của khối nguồn mà từ đó các ký hiệu sửa chữa được tạo ra. Tổng số ký hiệu sửa chữa được tạo ra cho i đoạn thứ nhất được tính là $TNRS(i) = \text{ceiling}(R * B(i)/S)$, trong đó $\text{ceiling}(x)$ là hàm cho kết quả là số nguyên nhỏ nhất với giá trị ít nhất là bằng x . Như vậy, số lượng ký hiệu sửa chữa được tạo ra cho đoạn i là $NRS(i) = TNRS(i) - TNRS(i - 1)$.

Đoạn sửa chữa bao gồm phần nối các ký hiệu sửa chữa cho các đoạn, trong đó thứ tự của các ký hiệu sửa chữa trong phân đoạn sửa chữa là theo thứ tự của các đoạn

mà chúng được tạo ra từ đó, và trong đoạn các ký hiệu sửa chữa là theo thứ tự của định danh ký mã hóa (ESI). Cấu trúc đoạn sửa chữa tương ứng với cấu trúc phân đoạn nguồn được thể hiện trên Fig.27, bao gồm bộ tạo phân đoạn sửa chữa 2700.

Lưu ý rằng bằng cách xác định số lượng ký hiệu sửa chữa cho đoạn như được mô tả ở trên, tổng số ký hiệu sửa chữa cho tất cả các đoạn trước đó, và do đó chỉ mục byte trong phân đoạn sửa chữa, chỉ phụ thuộc vào R, S, B(i - 1) và B(i), và không phụ thuộc vào cấu trúc bất kỳ trong số các cấu trúc trước đó hoặc sau đó của các đoạn trong phân đoạn nguồn. Điều này là có lợi bởi vì nó cho phép máy khách nhanh chóng tính toán vị trí bắt đầu của khối sửa chữa trong phân đoạn sửa chữa, và cũng nhanh chóng tính toán số lượng ký hiệu sửa chữa trong khối sửa chữa đó, chỉ sử dụng thông tin cục bộ về cấu trúc của đoạn tương ứng của phân đoạn nguồn mà khối sửa chữa được tạo ra từ đó. Vì vậy, nếu máy khách quyết định bắt đầu tải về và phát ra đoạn từ giữa phân đoạn nguồn, nó cũng có thể nhanh chóng tạo ra và truy cập khối sửa chữa tương ứng từ trong phân đoạn sửa chữa tương ứng.

Số lượng ký hiệu nguồn trong khối nguồn tương ứng với đoạn i được tính là $NSS(i) = \text{ceiling}((B(i) - B(i - 1))/S)$. Ký hiệu nguồn cuối cùng được đệm với các byte không (zero) cho mục đích mã hóa và giải mã FEC nếu B(i) - B(i - 1) không phải là bội số của S, tức là, ký hiệu nguồn cuối cùng được đệm với các byte không (zero) để nó có kích thước S byte để phục vụ mục đích mã hóa và giải mã FEC, nhưng các byte không (zero) đệm này không được lưu trữ như là một phần của phân đoạn nguồn. Theo phương án này, các ESI cho ký hiệu nguồn là 0, 1,..., NSS(i) - 1 và các ESI cho các ký hiệu sửa chữa là NSS(i),..., NSS(i) + NRS(i) - 1.

URL cho phân đoạn sửa chữa theo phương án này có thể được tạo ra từ URL cho phân đoạn nguồn tương ứng đơn giản bằng cách thêm ví dụ hậu tố ".repair" vào URL của phân đoạn nguồn.

Thông tin chỉ mục sửa chữa và thông tin FEC cho phân đoạn sửa chữa được ngầm xác định bởi thông tin lập chỉ mục đối với phân đoạn nguồn tương ứng, và từ các giá trị của R và S, như được mô tả ở đây. Các độ dịch thời gian và cấu trúc đoạn bao gồm phân đoạn sửa chữa được xác định bằng độ dịch thời gian và cấu trúc của phân đoạn nguồn tương ứng. Độ dịch byte đến cuối của các ký hiệu sửa chữa trong phân đoạn sửa chữa tương ứng với đoạn i có thể được tính là $RB(i) = S * \text{ceiling}(R * B(i)/S)$. Số lượng byte trong phân đoạn sửa chữa tương ứng với đoạn i là $RB(i) - RB(i$

- 1), và do đó số lượng ký hiệu sửa chữa tương ứng với đoạn i được tính là $NRS(i) = (RB(i) - RB(i - 1))/S$. Số lượng ký hiệu nguồn tương ứng với đoạn i có thể được tính là $NSS(i) = \text{ceiling}((B(i) - B(i - 1))/S)$. Như vậy, theo phương án này, thông tin lập chỉ mục sửa chữa cho khối sửa chữa trong phân đoạn sửa chữa và thông tin FEC tương ứng có thể được ngầm suy ra từ R , S và thông tin lập chỉ mục cho đoạn tương ứng của phân đoạn nguồn tương ứng.

Ví dụ, xét ví dụ được thể hiện trên Fig.28, thể hiện đoạn 2 bắt đầu từ độ dịch byte $B(1) = 6410$ và kết thúc tại độ dịch byte $B(2) = 6,770$. Trong ví dụ này, kích thước ký hiệu là $S = 64$ byte, và đường dọc chấm thể hiện các độ dịch byte trong phân đoạn nguồn tương ứng với bội số của S . Kích thước phân đoạn sửa chữa tổng thể là một phần của kích thước phân đoạn nguồn được thiết lập là $R = 0,5$ trong ví dụ này. Số lượng ký hiệu nguồn trong khối nguồn cho đoạn 2 được tính là $NSS(2) = \text{ceiling}((6770 - 6410)/64) = \text{ceil}(5,625) = 6$, và 6 ký hiệu nguồn này có giá trị ESI 0,..., 5, tương ứng, trong đó ký hiệu nguồn thứ nhất là 64 byte thứ nhất của đoạn 2 mà bắt đầu tại chỉ mục byte 6410 trong phân đoạn nguồn, ký hiệu nguồn thứ hai là 64 byte tiếp theo của đoạn 2 mà bắt đầu tại chỉ mục byte 6474 trong phân đoạn nguồn, v.v.. Độ dịch byte cuối của khối sửa chữa tương ứng với đoạn 2 được tính là $RB(2) = 64 * \text{ceiling}(0,5 * 6770/64) = 64 * \text{ceiling}(52,89...) = 64 * 53 = 3392$, và độ dịch byte bắt đầu của khối sửa chữa tương ứng với đoạn 2 được tính là $RB(1) = 64 * \text{ceiling}(0,5 * 6410/64) = 64 * \text{ceiling}(50,07...) = 64 * 51 = 3264$, và do đó trong ví dụ này có hai ký hiệu sửa chữa trong khối sửa chữa tương ứng với đoạn 2 với ESI 6 và 7, tương ứng, bắt đầu tại độ dịch byte 3264 trong phân đoạn sửa chữa và kết thúc tại độ dịch byte 3392.

Lưu ý rằng, trong ví dụ được thể hiện trên Fig.28, mặc dù $R = 0,5$ và có 6 ký hiệu nguồn tương ứng với đoạn 2, số lượng ký hiệu sửa chữa không phải là 3, do có thể mong đợi nếu chỉ đơn giản sử dụng số lượng ký hiệu nguồn này để tính toán số lượng ký hiệu sửa chữa, nhưng thay vào đó ra kết quả là 2 theo các phương pháp được mô tả ở đây. Trái ngược với việc chỉ đơn giản sử dụng số lượng ký hiệu nguồn của đoạn để xác định số lượng ký hiệu sửa chữa, các phương án được mô tả ở trên cho phép có thể tính toán vị trí của khối sửa chữa trong phân đoạn sửa chữa duy nhất từ thông tin chỉ mục liên quan đến khối nguồn tương ứng của phân đoạn nguồn tương ứng. Hơn nữa, do số lượng K các ký hiệu nguồn trong khối nguồn tăng lên, số lượng

ký hiệu sửa chữa, KR , của khối sửa chữa tương ứng gần xấp xỉ với $K * R$, nói chung, KR là lớn nhất là bằng $\text{ceil}(K * R)$ và KR nhỏ nhất là bằng $\text{floor}((K - 1) * R)$, trong đó $\text{floor}(x)$ là số nguyên lớn nhất lớn nhất là bằng x .

Người có hiểu biết trung bình trong lĩnh vực kỹ thuật hiểu rằng có nhiều biến thể của các phương án trên để tạo ra cấu trúc tập tin có thể được sử dụng một cách có lợi bởi máy khách sử dụng các phương pháp HTTP/FEC phối hợp. Theo một ví dụ về phương án thay thế, phân đoạn ban đầu cho dạng biểu diễn có thể được chia thành $N > 1$ phân đoạn song song, trong đó $i = 1, \dots, N$, phần cụ thể F_i của phân đoạn ban đầu được chứa trong phân đoạn song song thứ i của nó, và trong đó tổng cho $i = 1, \dots, N$ của F_i bằng 1. Trong phương án này, có thể có một ánh xạ phân đoạn chính được sử dụng để suy ra các ánh xạ phân đoạn cho tất cả các phân đoạn song song, tương tự như cách ánh xạ phân đoạn sửa chữa được suy ra từ ánh xạ phân đoạn nguồn trong phương án được mô tả ở trên. Ví dụ, ánh xạ phân đoạn chính có thể chỉ ra cấu trúc đoạn nếu tất cả dữ liệu phương tiện nguồn không được chia trong các phân đoạn song song mà thay vào đó được chứa trong một phân đoạn ban đầu, và sau đó ánh xạ phân đoạn cho phân đoạn song song thứ i có thể được suy ra từ ánh xạ phân đoạn chính bằng cách tính toán như sau, nếu lượng dữ liệu phương tiện trong tiền tố thứ nhất của các đoạn của phân đoạn ban đầu là L byte, thì tổng số byte của tiền tố này trong phân đoạn song song i thứ nhất là $\text{ceil}(L * G_i)$, trong đó G_i là tổng trên $j = 1, \dots, i$ của F_j . Theo ví dụ khác về phương án thay thế, các phân đoạn có thể bao gồm sự kết hợp của dữ liệu phương tiện nguồn ban đầu cho mỗi đoạn mà ngay sau đó là dữ liệu sửa chữa cho đoạn đó, tạo ra phân đoạn có chứa hỗn hợp của dữ liệu phương tiện nguồn và dữ liệu sửa chữa được tạo ra bằng cách sử dụng mã FEC từ dữ liệu phương tiện nguồn đó. Theo ví dụ khác về phương án thay thế, phân đoạn có chứa hỗn hợp dữ liệu phương tiện nguồn và dữ liệu sửa chữa có thể được chia thành nhiều phân đoạn song song có chứa hỗn hợp của dữ liệu phương tiện nguồn và dữ liệu sửa chữa.

Các phương pháp xử lý truyền liên tục có độ trễ thấp

Trong một số kịch bản triển khai, quy trình truyền liên tục có độ trễ thấp cho dịch vụ trực tiếp có thể được mong muốn. Ví dụ, trong trường hợp phân phối tại chỗ cục bộ một sự kiện, chẳng hạn như sự kiện thể thao hoặc buổi hòa nhạc, mong muốn

có độ trễ giữa hành động trực tiếp và phần trình diễn của dịch vụ trực tiếp trên máy khách càng ngắn càng tốt. Ví dụ, độ trễ tối đa 1 giây có thể được mong muốn.

Như đã mô tả ở trên, có lợi nếu bố trí mỗi tập tin lưu trữ phân đoạn của phần trình diễn phương tiện bắt đầu với điểm truy cập ngẫu nhiên (RAP). Một số profin, cụ thể là profin trực tiếp định dạng tập tin phương tiện cơ sở ISO, yêu cầu là mỗi phân đoạn phương tiện bắt đầu với một RAP.

Tuy nhiên, trong môi trường mà trong đó việc phân phối đầu-cuối có độ trễ thấp là cần thiết, thời lượng của mỗi phân đoạn phải ngắn để giảm thiểu độ trễ giữa hành động trực tiếp và phần trình diễn của sự kiện trực tiếp trên máy khách. Mong muốn là tránh chèn RAP vào trong từng phân đoạn được sử dụng cho quy trình truyền liên tục có độ trễ thấp. Ví dụ, các RAP trong video thường được thực hiện bởi các khung IDR. Hiệu quả mã hóa có thể được cải thiện bằng cách tránh sử dụng các khung IDR khung trong các phân đoạn ngắn mong muốn để truyền liên tục có độ trễ thấp.

Theo một phương án, dạng biểu diễn phù hợp với profin trực tiếp và dạng biểu diễn có độ trễ thấp của phần trình diễn phương tiện được tạo ra. Dạng biểu diễn phù hợp với profin trực tiếp có thời lượng phân đoạn phương tiện tương đối lớn hơn. Mỗi phân đoạn phương tiện của dạng biểu diễn phù hợp với profin trực tiếp có RAP ở đầu phân đoạn phương tiện. Dạng biểu diễn có độ trễ thấp có phân đoạn tương đối ngắn hơn (có thể được gọi là "đoạn phương tiện") mà có thể không chứa RAP. Máy khách hỗ trợ truyền liên tục có độ trễ thấp có thể nhận được các đoạn phương tiện được tạo ra cho dạng biểu diễn có độ trễ thấp của phần trình diễn phương tiện, trong khi đó máy khách không hỗ trợ truyền liên tục có độ trễ thấp có thể nhận được phân đoạn phương tiện được tạo ra cho dạng biểu diễn phù hợp với profin trực tiếp của phần trình diễn phương tiện.

Fig.30 minh họa mối quan hệ giữa các đoạn phương tiện để truyền liên tục có độ trễ thấp và các đoạn phương tiện. Phân đoạn phương tiện 3002 được tạo ra cho quy trình truyền liên tục profin trực tiếp có chứa RAP 3004 ở đầu của dữ liệu phương tiện ("mdat"). Ngược lại, trong số các đoạn phương tiện 3004, 3006 và 3008 được tạo ra cho quy trình truyền liên tục có độ trễ thấp, chỉ có đoạn phương tiện 3004 chứa RAP.

Các đoạn phương tiện được tạo ra khi cần và có sẵn để tải về bởi máy khách qua HTTP. Các đoạn phương tiện có thể được tích lũy thành các phân đoạn phương tiện phù hợp với các profin trực tiếp định dạng tập tin phương tiện cơ sở ISO, mà

không cần có bất kỳ sửa đổi nào đối với các đoạn phương tiện. Ví dụ, các đoạn phương tiện có thể được nối thành các phân đoạn phương tiện.

Cả phân đoạn phương tiện lẫn các đoạn phương tiện có thể được tạo ra bằng cách sử dụng cùng quy trình mã hóa. Theo cách này, phương tiện có thể được mã hóa hiệu quả để sử dụng bởi các máy khách hoạt động trong môi trường yêu cầu độ trễ đầu-cuối thấp và bởi các máy khách sử dụng giao thức đòi hỏi có RAP trong từng phân đoạn.

Theo một số phương án, chỉ mục phân đoạn (SIDX) được tạo ra cho mỗi đoạn phương tiện. SIDX có thể bao gồm dải thời gian trình diễn trong phân đoạn phương tiện và dãy byte tương ứng của phân đoạn phương tiện bị chiếm bởi đoạn phương tiện. Theo một số phương án, SIDX chỉ rõ việc RAP có mặt trong đoạn hay không. Trên Fig.30, trường `contains_RAP` của hộp SIDX của đoạn phương tiện 3004 được thiết lập là 1, chỉ ra rằng đoạn phương tiện 3004 có chứa RAP. Trường `contains_RAP` của hộp SIDX của các đoạn phương tiện 3006 và 3008 được thiết lập là 0, chỉ ra rằng các đoạn phương tiện 3006 và 3008 không chứa RAP. SIDX có thể còn chỉ ra thời gian trình diễn của RAP thứ nhất trong đoạn.

Theo một phương án, máy chủ phương tiện có thể tạo ra các đoạn cho quy trình truyền liên tục có độ trễ thấp và đẩy các đoạn này đến bộ nhớ đệm. Các bộ nhớ đệm có thể nối các đoạn để tạo ra các phân đoạn phương tiện tương thích với profin trực tiếp. Sau khi phân đoạn phương tiện được tạo ra, bộ nhớ đệm có thể thanh lọc các đoạn phương tiện đã được nối để tạo ra phân đoạn phương tiện.

Mô tả trình diễn phương tiện (MPD) duy nhất có thể lưu trữ thông tin về dạng biểu diễn thứ nhất có các phân đoạn phương tiện phù hợp với profin trực tiếp của phần trình diễn phương tiện và dạng biểu diễn thứ hai có các đoạn phương tiện của dòng có độ trễ thấp. Việc xem có dịch thời gian có thể được cung cấp bằng cách sử dụng các phân đoạn phương tiện để đệm dịch thời và các đoạn phương tiện để xử lý việc xem ở thời điểm gần trực tiếp của dòng. Máy khách có thể chuyển đổi giữa các dạng biểu diễn này, ví dụ, bắt đầu trong bộ đệm dịch thời và tiến gần hơn đến thời điểm trực tiếp bằng cách bỏ qua các phần của phần trình diễn phương tiện. Mỗi dạng biểu diễn của một MPD có thể được gán một thuộc tính thể hiện mảng các dạng biểu diễn có sẵn cho một lần trình diễn phương tiện duy nhất.

Trong MPD lưu trữ thông tin về dạng biểu diễn thứ nhất có các phân đoạn phương tiện và dạng biểu diễn thứ hai có các đoạn phương tiện, có thể có lợi nếu cung cấp thông tin chỉ ra các đoạn phương tiện nào của dạng biểu diễn thứ hai bắt đầu với một RAP. Ví dụ, MPD có thể bao gồm thuộc tính để chỉ ra tần số xuất hiện của các RAP trong các đoạn phương tiện. Theo một phương án, MPD bao gồm thuộc tính chỉ ra tần số về số lượng đoạn (tức là, mỗi đoạn phương tiện thứ x có chứa một RAP). Theo phương án khác, thuộc tính này chỉ ra tần số về khoảng cách thời gian giữa các RAP liên kế.

Theo cách khác, thông tin về các đoạn phương tiện có thể được lưu trữ trong MPD thứ nhất và thông tin về các phân đoạn phương tiện có thể được lưu trữ trong MPD thứ hai.

Theo một số phương án, MPD có thể báo hiệu các tham số cụ thể đối với dạng biểu diễn cụ thể, chẳng hạn như thời lượng tối đa của phân đoạn phương tiện hoặc đoạn phương tiện của dạng biểu diễn.

Các phương án khác cũng có thể được hình dung ra bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật sau khi đọc sáng chế. Theo các phương án khác, các tổ hợp hoặc tổ hợp con của các giải pháp được bộc lộ trên đây có thể được thực hiện theo cách có lợi. Các sắp xếp làm ví dụ về các thành phần được đưa ra ở đây để minh họa và cần phải hiểu rằng các cách kết hợp, bổ sung, sắp xếp lại, và tương tự có thể được dự tính trong các phương án thay thế của sáng chế. Vì vậy, cần phải hiểu rằng, mặc dù sáng chế đã được mô tả dựa trên các phương án làm ví dụ, người có hiểu biết trung bình trong lĩnh vực kỹ thuật hiểu rằng có thể thực hiện rất nhiều cải biến đối với các phương án này.

Ví dụ, các quy trình được mô tả ở đây có thể được thực hiện bằng cách sử dụng các bộ phận phần cứng, các bộ phận phần mềm, và/hoặc dạng kết hợp bất kỳ của chúng. Trong một số trường hợp, các bộ phận phần mềm có thể được cung cấp trong vật ghi bất biến, hữu hình để thực thi trên phần cứng mà được cung cấp phương tiện hoặc tách riêng với phương tiện. Bản mô tả và hình vẽ, theo đó, được xem là mang tính minh họa hơn là mang tính giới hạn. Tuy nhiên, rõ ràng là các cải biến và thay đổi khác nhau có thể được thực hiện mà không tách rời khỏi phạm vi rộng hơn của sáng chế như được nêu trong các yêu cầu bảo hộ, và rõ ràng là sáng chế dự định bao hàm tất cả các cải biến và dạng tương đương trong phạm vi của các yêu cầu bảo hộ sau đây.

Yêu cầu bảo hộ

1. Phương pháp tạo cấu trúc dữ liệu của nội dung cần được phục vụ, bằng cách sử dụng máy chủ phương tiện, phương pháp này bao gồm các bước:

thu nhận nội dung cần được phục vụ;

tạo ra các phân đoạn phương tiện biểu diễn nội dung này và được mã hóa theo giao thức mã hóa bao gồm một hoặc nhiều khung của phần trình diễn phương tiện được mã hóa trong từng phân đoạn phương tiện, trong đó điểm truy cập ngẫu nhiên có sẵn trong mỗi phân đoạn phương tiện;

tạo ra các đoạn phương tiện được mã hóa theo giao thức mã hóa, trong đó phân đoạn phương tiện bao gồm các đoạn phương tiện, và trong đó ít nhất một số trong các đoạn phương tiện bao gồm các điểm truy cập ngẫu nhiên và ít nhất một số đoạn phương tiện không bao gồm các điểm truy cập ngẫu nhiên, điểm truy cập ngẫu nhiên bao gồm vị trí trong phân đoạn mà tại đó bộ giải mã có thể giải mã các đoạn sau điểm truy cập ngẫu nhiên này một cách độc lập với các đoạn trước điểm truy cập ngẫu nhiên; và

tạo ra chỉ số phân đoạn cho phân đoạn phương tiện, chỉ số phân đoạn này bao gồm dải thời gian trình diễn cho mỗi đoạn phương tiện trong phân đoạn phương tiện, dãy byte tương ứng trong phân đoạn phương tiện bị chiếm bởi mỗi đoạn phương tiện, và chỉ báo sự có mặt của điểm truy cập ngẫu nhiên để chỉ báo xem điểm truy cập ngẫu nhiên có có mặt trong mỗi đoạn phương tiện hay không.

2. Phương pháp theo điểm 1, trong đó phân đoạn phương tiện được tạo ra bằng cách nối các đoạn phương tiện.

3. Phương pháp theo điểm 2, trong đó phương pháp này còn bao gồm bước tạo ra phân đoạn phương tiện trong bộ nhớ đệm và, trong đó sau khi phân đoạn phương tiện được tạo ra trong bộ nhớ đệm, các đoạn phương tiện dùng để tạo ra phân đoạn phương tiện được thanh lọc khỏi bộ nhớ đệm.

4. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm bước tạo ra một tập tin mô tả trình diễn phương tiện (media presentation description - MPD) duy nhất

lưu trữ thông tin về dạng biểu diễn thứ nhất của phần trình diễn phương tiện bao gồm các phân đoạn phương tiện và dạng biểu diễn thứ hai của phần trình diễn phương tiện bao gồm các đoạn phương tiện.

5. Phương pháp theo điểm 4, trong đó tập tin MPD bao gồm thuộc tính chỉ báo tần suất xuất hiện của các điểm truy cập ngẫu nhiên trong dạng biểu diễn thứ hai.

6. Phương pháp theo điểm 5, trong đó tần suất là khoảng thời gian.

7. Phương pháp theo điểm 5, trong đó tần suất là số đoạn phương tiện.

8. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm bước xác định các vị trí cho các điểm truy cập ngẫu nhiên trong số các đoạn phương tiện, trong đó các điểm truy cập ngẫu nhiên được định vị tại các điểm không cố định biến thiên trong số các đoạn phương tiện.

9. Máy chủ phương tiện bao gồm:

bộ xử lý được tạo cấu hình để:

thu nhận nội dung cần được phục vụ;

tạo ra các phân đoạn phương tiện biểu diễn nội dung này và được mã hóa theo giao thức mã hóa bao gồm một hoặc nhiều khung của phần trình diễn phương tiện được mã hóa trong từng phân đoạn phương tiện, trong đó điểm truy cập ngẫu nhiên có sẵn trong phân mỗi đoạn phương tiện;

tạo ra các đoạn phương tiện được mã hóa theo giao thức mã hóa, trong đó phân đoạn phương tiện bao gồm các đoạn phương tiện, và trong đó ít nhất một số trong các đoạn phương tiện bao gồm các điểm truy cập ngẫu nhiên và ít nhất một số đoạn phương tiện không bao gồm các điểm truy cập ngẫu nhiên, điểm truy cập ngẫu nhiên bao gồm vị trí trong phân đoạn mà tại đó bộ giải mã có thể giải mã các đoạn sau điểm truy cập ngẫu nhiên này một cách độc lập với các đoạn trước điểm truy cập ngẫu nhiên; và

tạo ra chỉ số phân đoạn cho phân đoạn phương tiện, chỉ số phân đoạn này bao gồm dải thời gian trình diễn cho mỗi đoạn phương tiện trong phân đoạn

phương tiện, dãy byte tương ứng trong phân đoạn phương tiện bị chiếm bởi mỗi đoạn phương tiện, và chỉ báo sự có mặt của điểm truy cập ngẫu nhiên để chỉ báo xem liệu điểm truy cập ngẫu nhiên có có mặt trong mỗi đoạn phương tiện hay không.

10. Máy chủ phương tiện theo điểm 9, trong đó phân đoạn phương tiện được tạo ra bằng cách nối các đoạn phương tiện.

11. Máy chủ phương tiện theo điểm 10, trong đó bộ xử lý còn được tạo cấu hình để tạo ra phân đoạn phương tiện trong bộ nhớ đệm và, trong đó sau khi phân đoạn phương tiện được tạo ra trong bộ nhớ đệm, các đoạn phương tiện dùng để tạo ra phân đoạn phương tiện được thanh lọc khỏi bộ nhớ đệm.

12. Máy chủ phương tiện theo điểm 9, trong đó bộ xử lý còn được tạo cấu hình để tạo ra một tập tin mô tả trình diễn phương tiện (media presentation description - MPD) duy nhất lưu trữ thông tin về dạng biểu diễn thứ nhất của phần trình diễn phương tiện bao gồm các phân đoạn phương tiện và dạng biểu diễn thứ hai của phần trình diễn phương tiện bao gồm các đoạn phương tiện.

13. Máy chủ phương tiện theo điểm 12, trong đó tập tin MPD bao gồm thuộc tính chỉ báo tần suất xuất hiện của các điểm truy cập ngẫu nhiên trong dạng biểu diễn thứ hai.

14. Máy chủ phương tiện theo điểm 13, trong đó tần suất là khoảng thời gian.

15. Máy chủ phương tiện theo điểm 14, trong đó tần suất là số đoạn phương tiện.

16. Máy chủ phương tiện theo điểm 9, trong đó bộ xử lý còn được tạo cấu hình để xác định các vị trí cho các điểm truy cập ngẫu nhiên trong số các đoạn phương tiện, trong đó các điểm truy cập ngẫu nhiên được định vị tại các điểm không cố định biến thiên trong số các đoạn phương tiện.

17. Vật ghi bất biến đọc được bằng máy tính lưu trữ các lệnh có thể thực thi bằng máy tính, mà khi được thực thi, khiến cho một hoặc nhiều thiết bị máy tính:

thu nhận nội dung cần được phục vụ;

tạo ra các phân đoạn phương tiện biểu diễn nội dung này và được mã hóa theo giao thức mã hóa bao gồm một hoặc nhiều khung của phần trình diễn phương tiện được mã hóa trong từng phân đoạn phương tiện, trong đó điểm truy cập ngẫu nhiên có sẵn trong mỗi phân đoạn phương tiện;

tạo ra các đoạn phương tiện được mã hóa theo giao thức mã hóa, trong đó phân đoạn phương tiện bao gồm các đoạn phương tiện, và trong đó ít nhất một số trong các đoạn phương tiện bao gồm các điểm truy cập ngẫu nhiên và ít nhất một số đoạn phương tiện không bao gồm các điểm truy cập ngẫu nhiên, điểm truy cập ngẫu nhiên bao gồm vị trí trong phân đoạn mà tại đó bộ giải mã có thể giải mã các đoạn sau điểm truy cập ngẫu nhiên này một cách độc lập với các đoạn trước điểm truy cập ngẫu nhiên; và

tạo ra chỉ số phân đoạn cho phân đoạn phương tiện, chỉ số phân đoạn này bao gồm dải thời gian trình diễn cho mỗi đoạn phương tiện trong phân đoạn phương tiện, dãy byte tương ứng trong phân đoạn phương tiện bị chiếm bởi mỗi đoạn phương tiện, và chỉ báo sự có mặt của điểm truy cập ngẫu nhiên để chỉ báo xem điểm truy cập ngẫu nhiên có có mặt trong mỗi đoạn phương tiện hay không.

18. Vật ghi đọc được bằng máy tính theo điểm 17, trong đó các lệnh này, khi được thực thi, khiến cho một hoặc nhiều thiết bị máy tính xác định các vị trí cho các điểm truy cập ngẫu nhiên trong số các đoạn phương tiện, trong đó các điểm truy cập ngẫu nhiên được định vị tại các điểm không cố định biến thiên trong số các đoạn phương tiện.

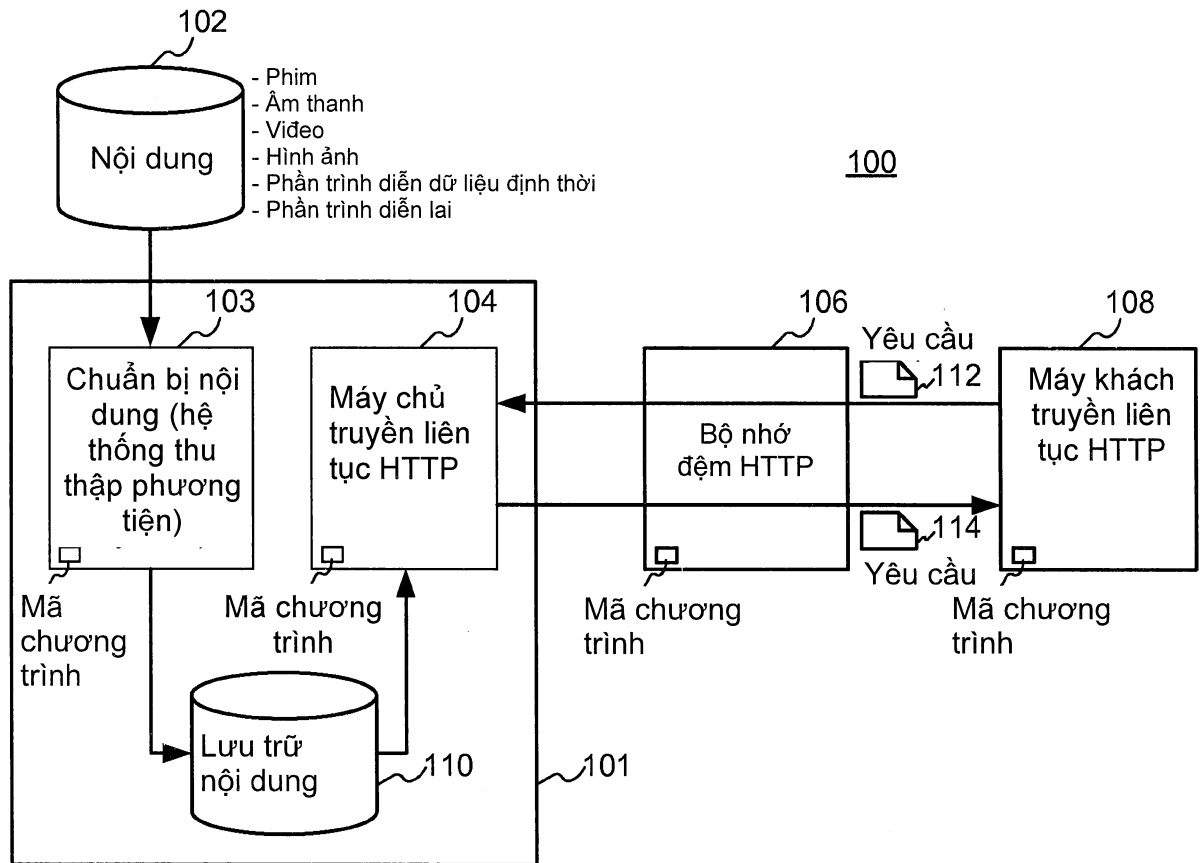


FIG. 1

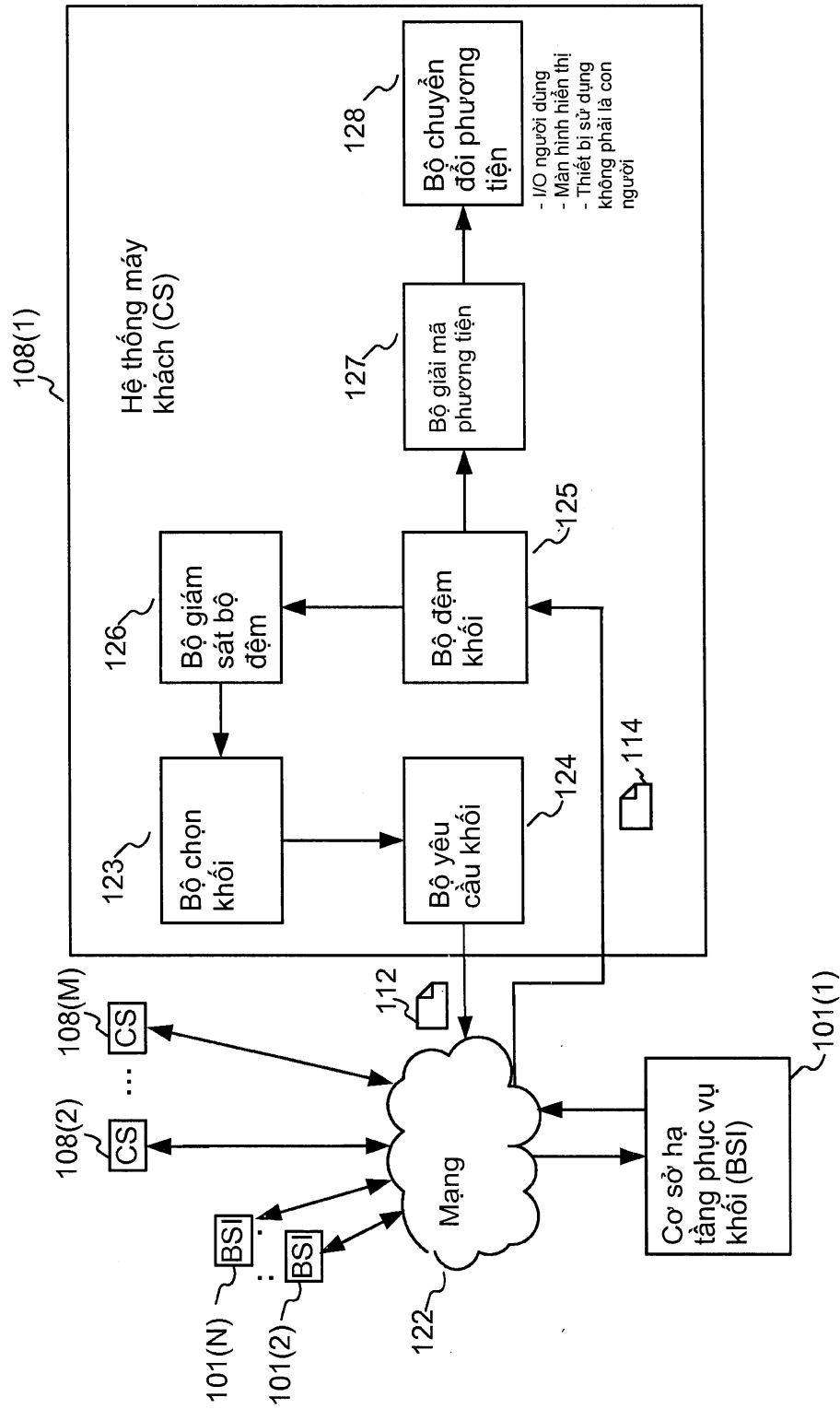


FIG. 2

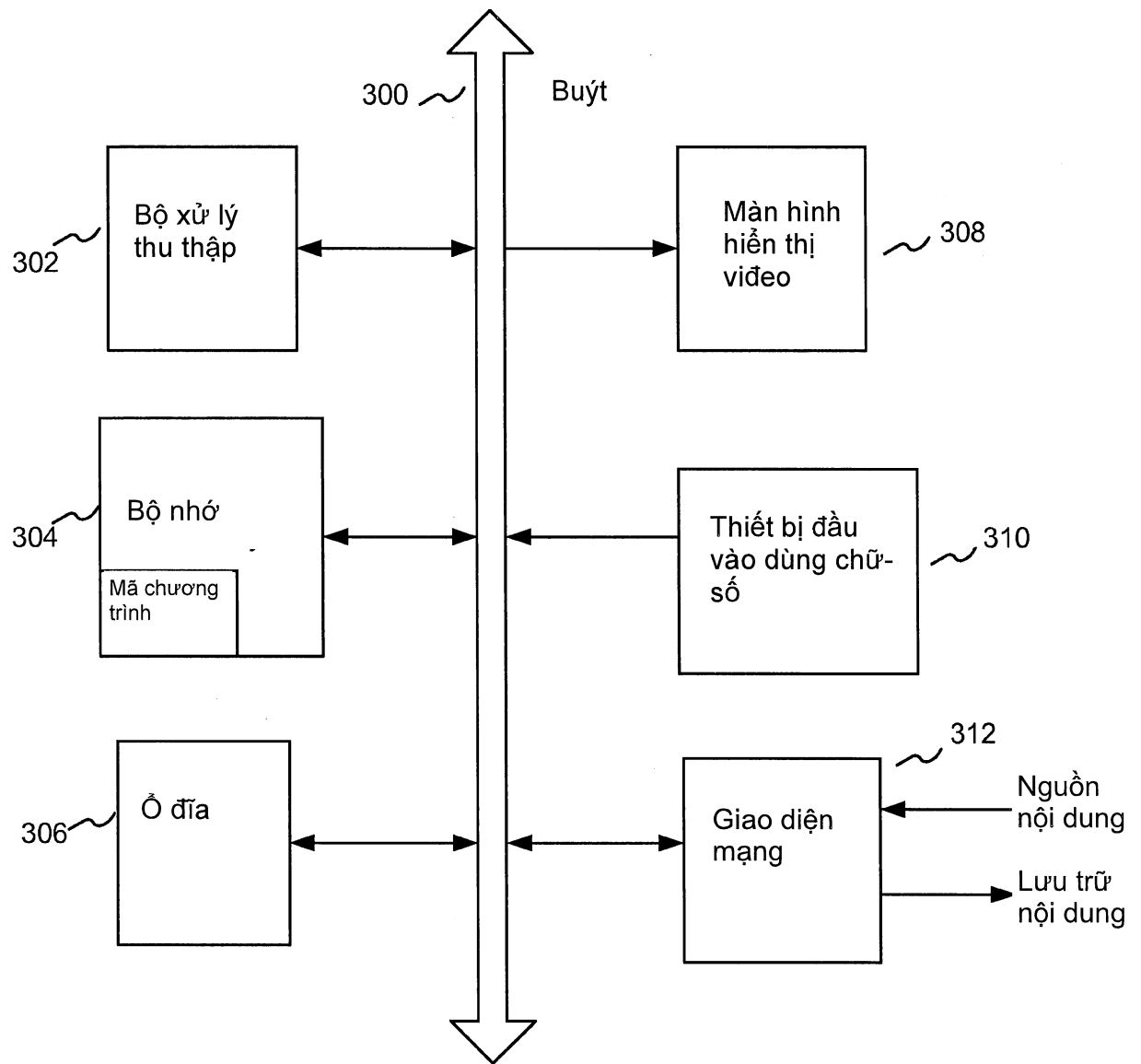


FIG. 3

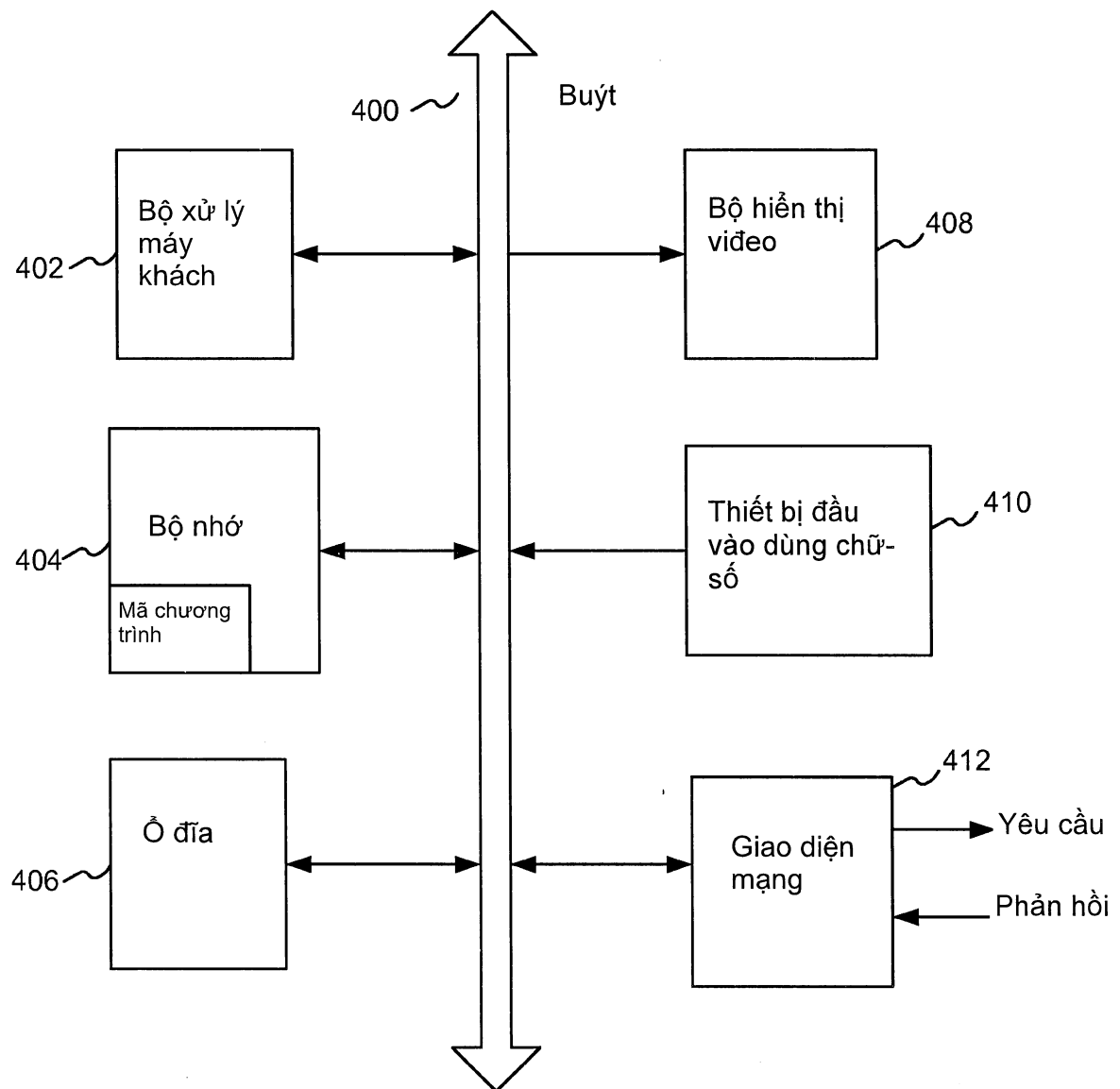


FIG. 4

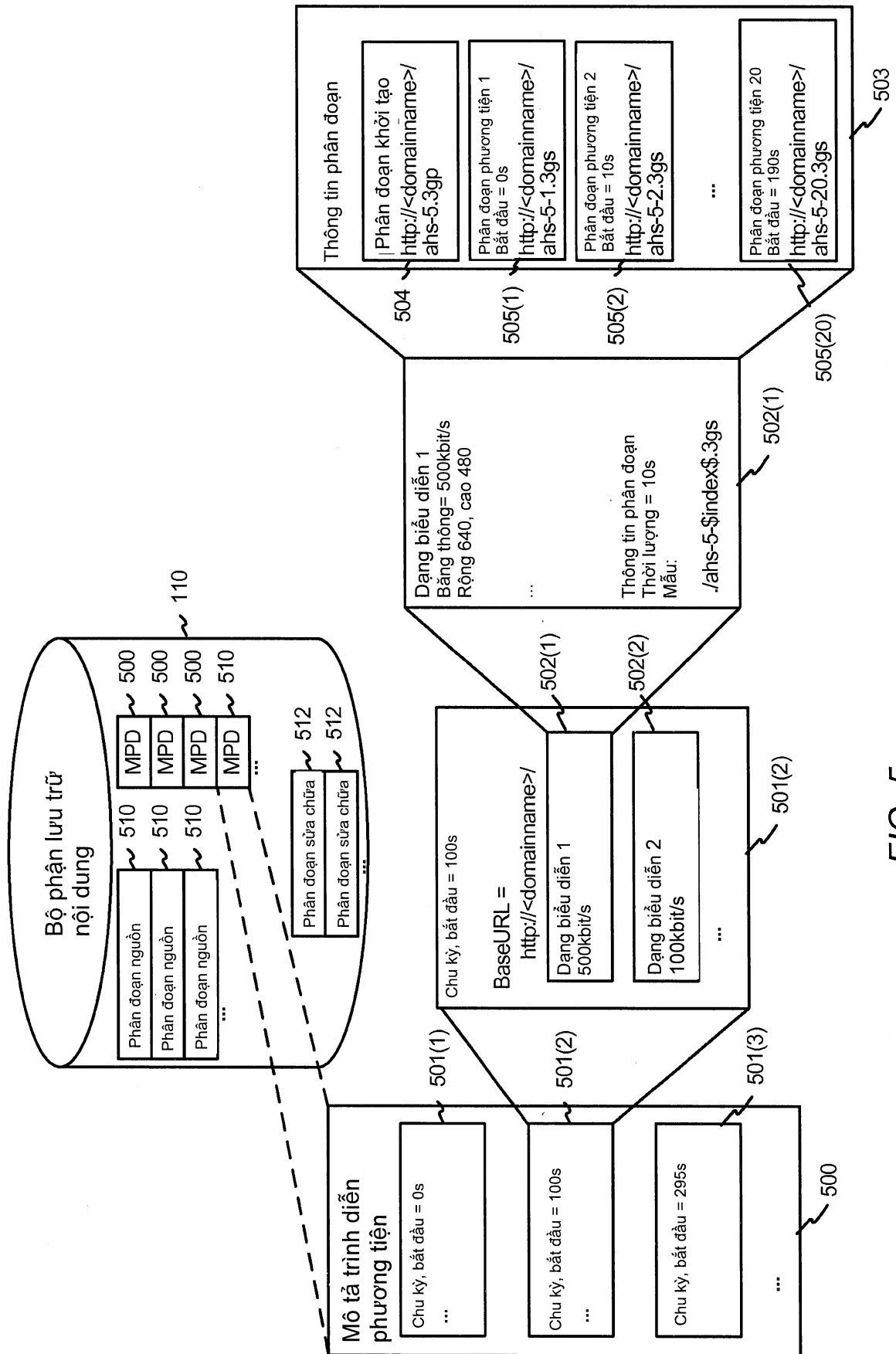


FIG. 5

Thông tin chỉ mục	Đoạn 1	Đoạn 2	Đoạn 3	...	Đoạn n

510

Độ dịch thời (ms)	Độ dịch byte	Chỉ báo RAP
20,000	0	1
20,485	50,245	0
21,100	101,354	0
21,623	157,034	1
21,965	198,045	0
22,540	256,654	0
23,045	301,245	1
24,000	358,436	0
24,605	400,983	1
25,132	465,745	0
25,845	504,103	0

FIG. 6

700
Chỉ mục đơn giản

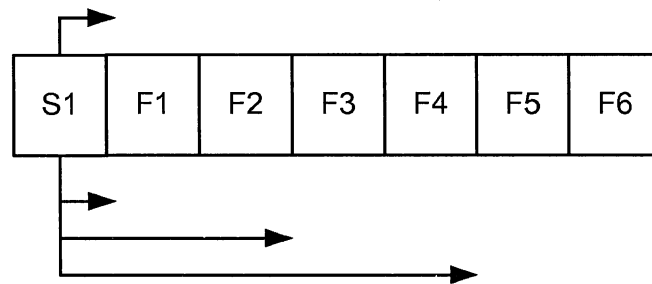


FIG. 7(a)

702
Chỉ mục phân cấp

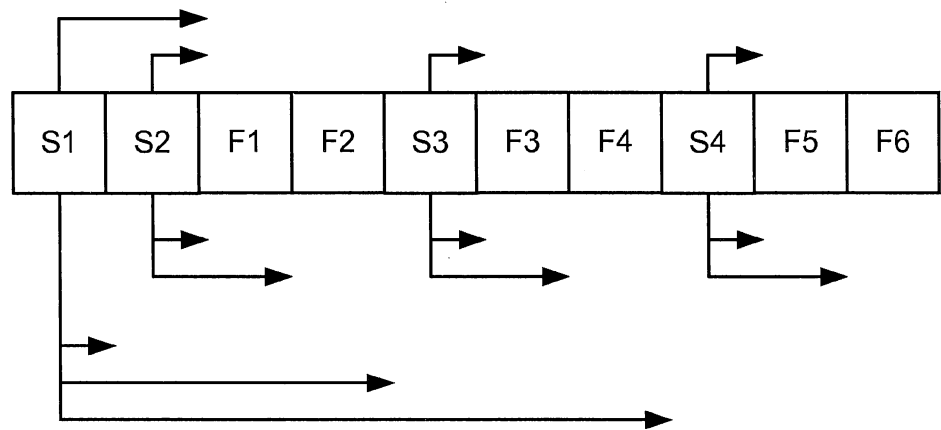


FIG. 7(b)

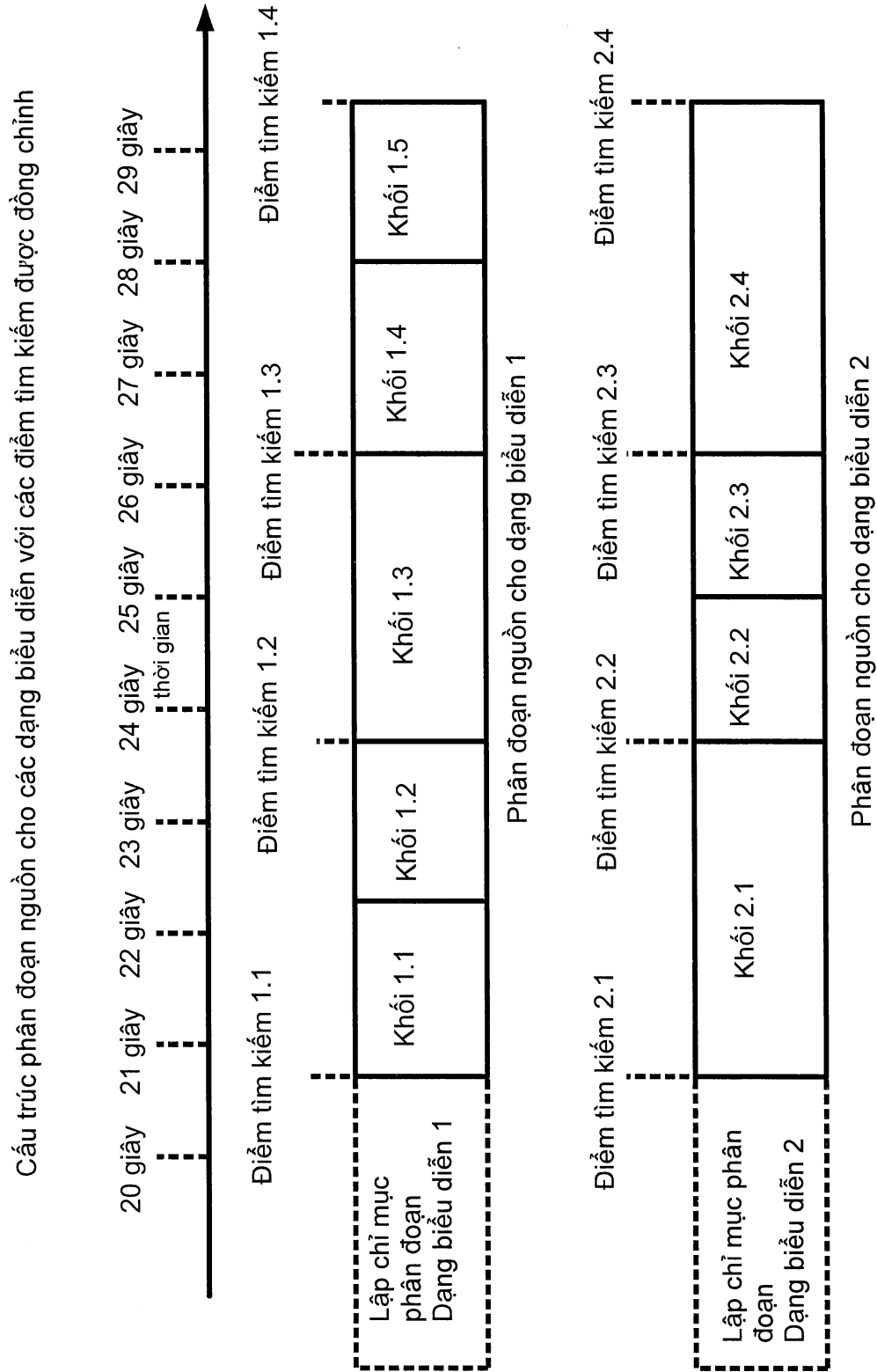


FIG. 8(a)

Cấu trúc phân đoạn nguồn cho các dạng biểu diễn với các điểm tìm kiếm không được đồng chỉnh

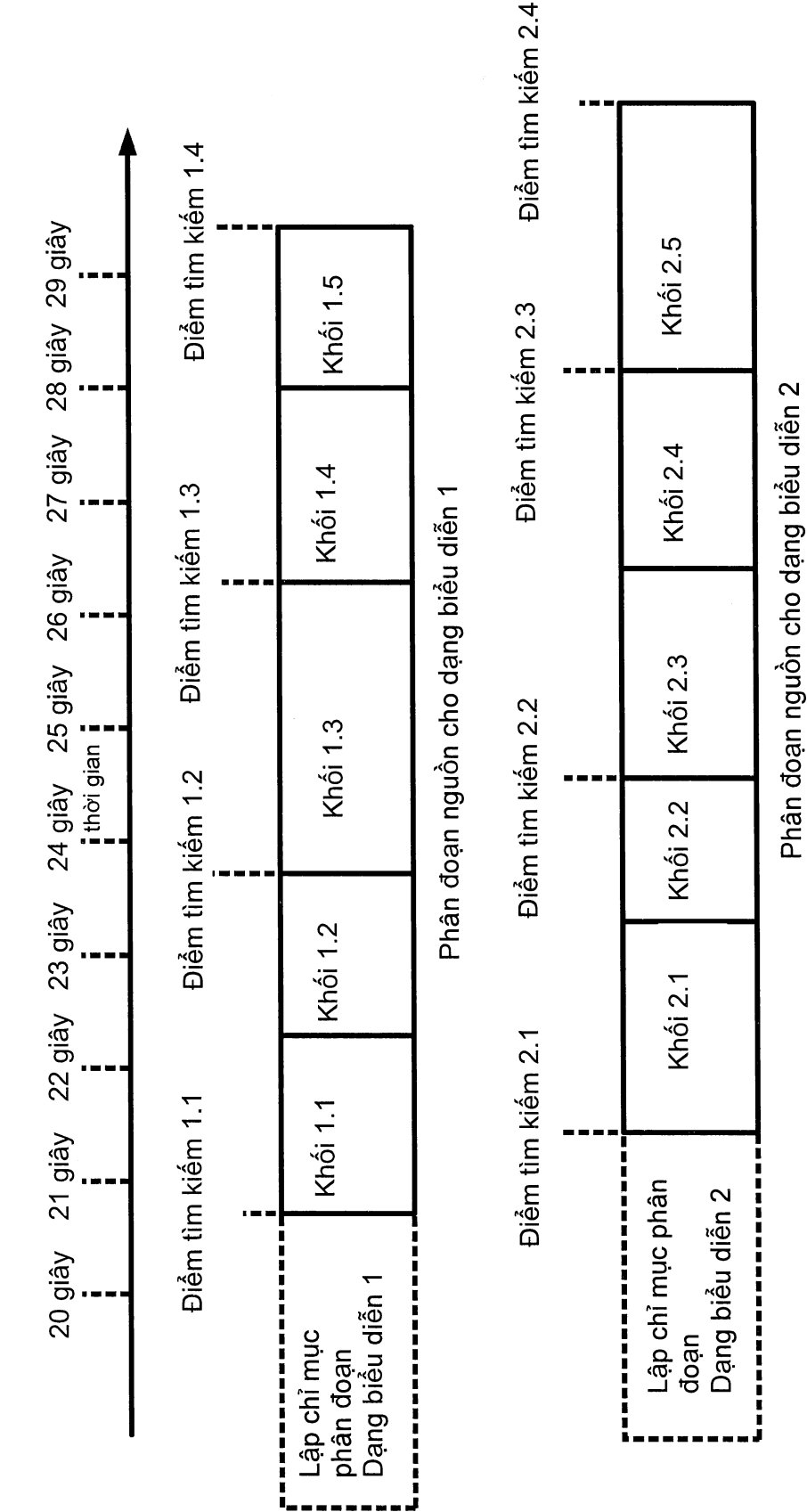


FIG. 8(b)

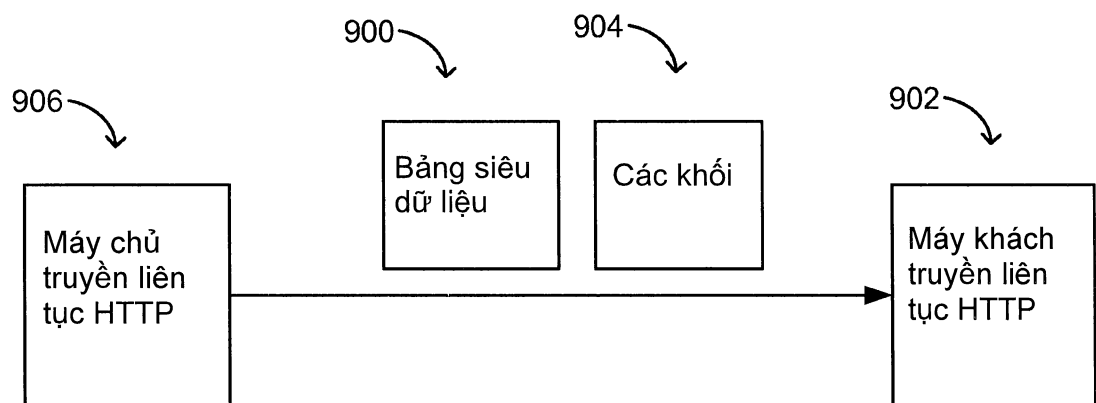
900 

Mức trình diễn
Thời lượng
Mã hóa
tốc độ bit
phần mã hóa - giải mã
độ phân giải
tốc độ khung
con trỏ đến siêu dữ liệu dòng
Thông tin bảo vệ nội dung/DRM

Siêu dữ liệu dòng
URL cho các tập tin phân đoạn

Siêu dữ liệu dòng
Dãy byte theo thời gian đối với các yêu cầu trong định
danh phân đoạn của các RAP

FIG. 9(a)

*FIG. 9(b)*

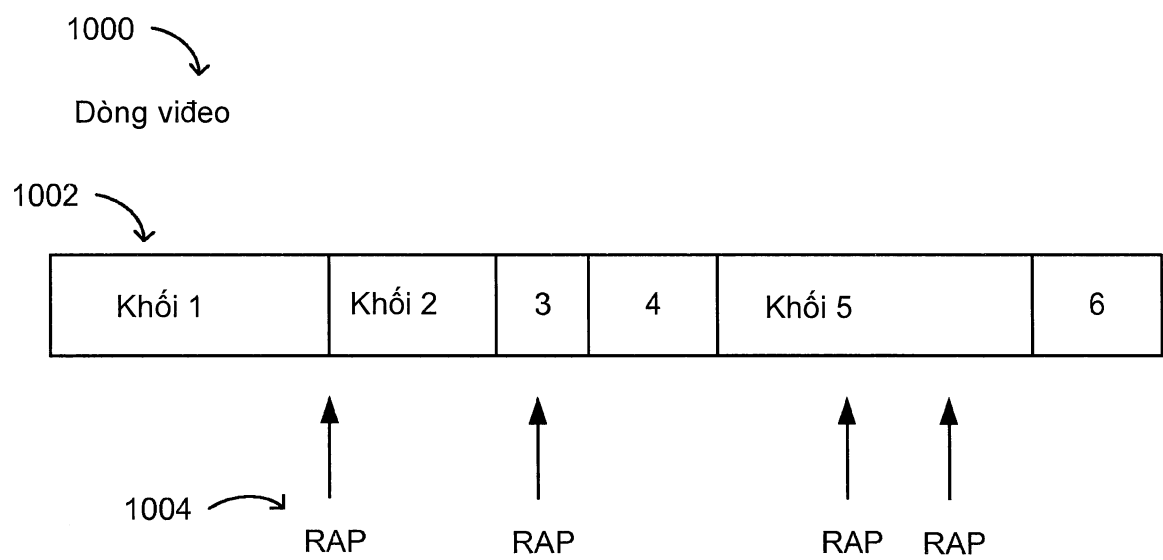
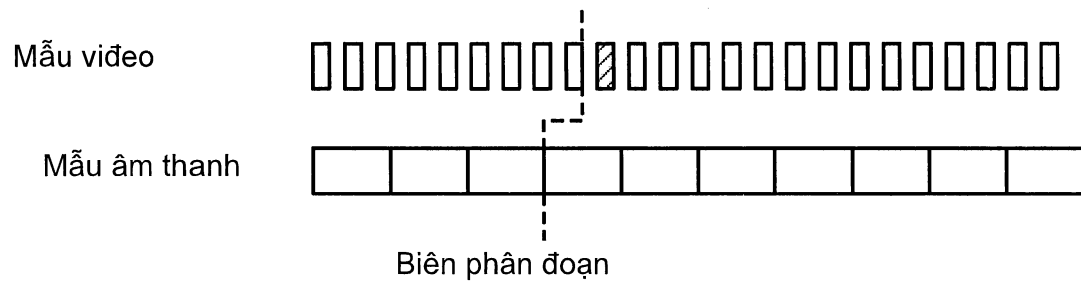
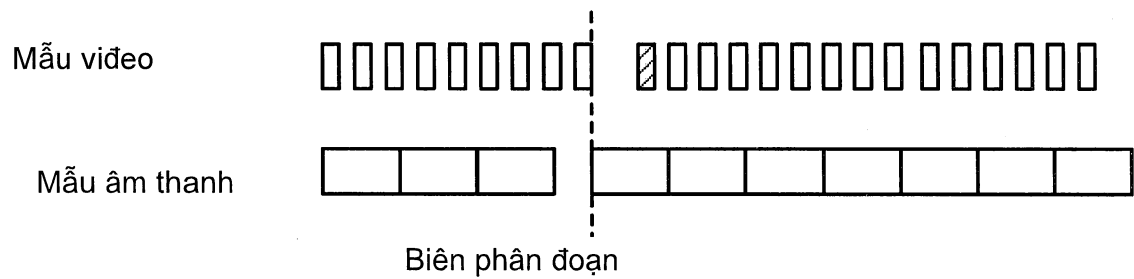


FIG. 10

Định thời liên tục trên các phân đoạn



Định thời gián đoạn trên các phân đoạn



Giả sử

* Khung có nét gạch được chọn là điểm biên vì lý do độc lập

* Âm thanh không bao giờ được bắt đầu muộn hơn video

Chú ý:

* Trong trường hợp liên tục, phân đoạn thứ hai là cùng nội dung với phân đoạn thứ nhất

* Trong trường hợp gián đoạn, phân đoạn thứ hai là nội dung khác với phân đoạn thứ nhất

FIG. 11

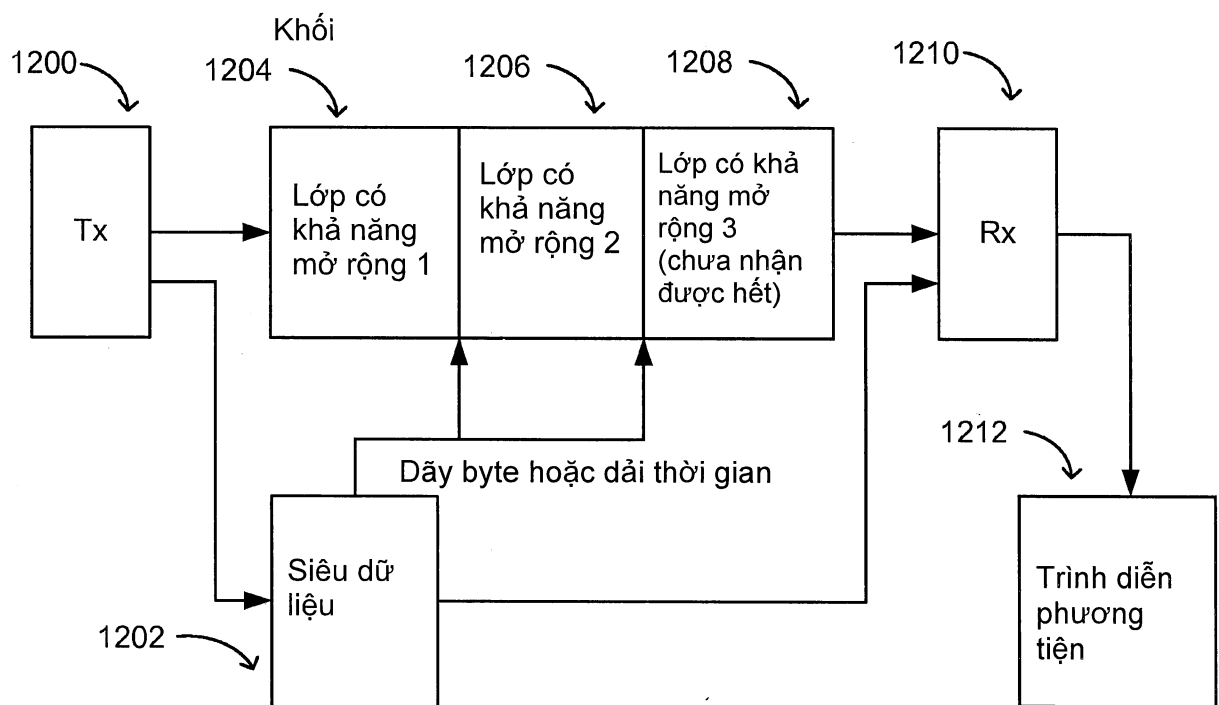


FIG. 12

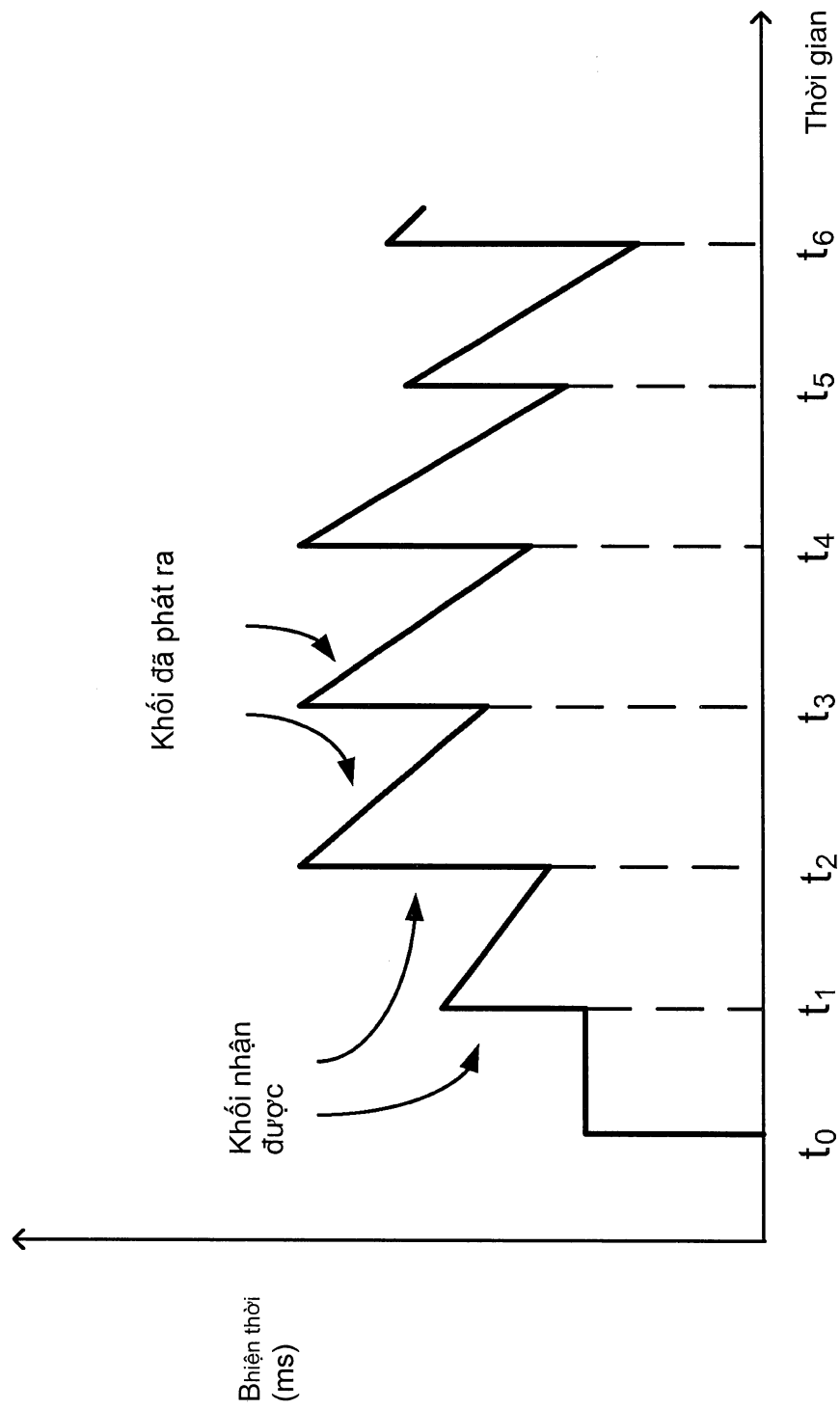


FIG. 13

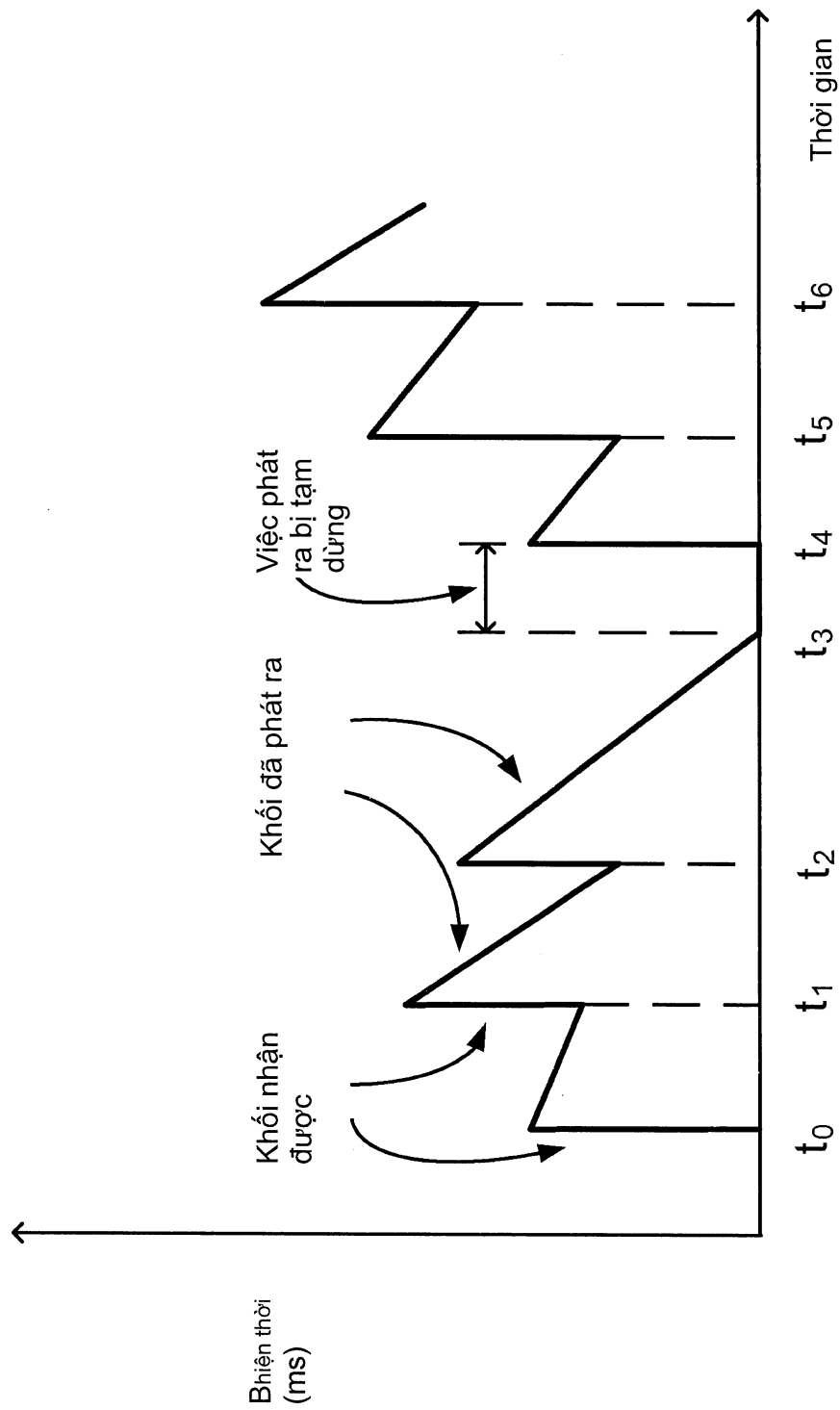


FIG. 14

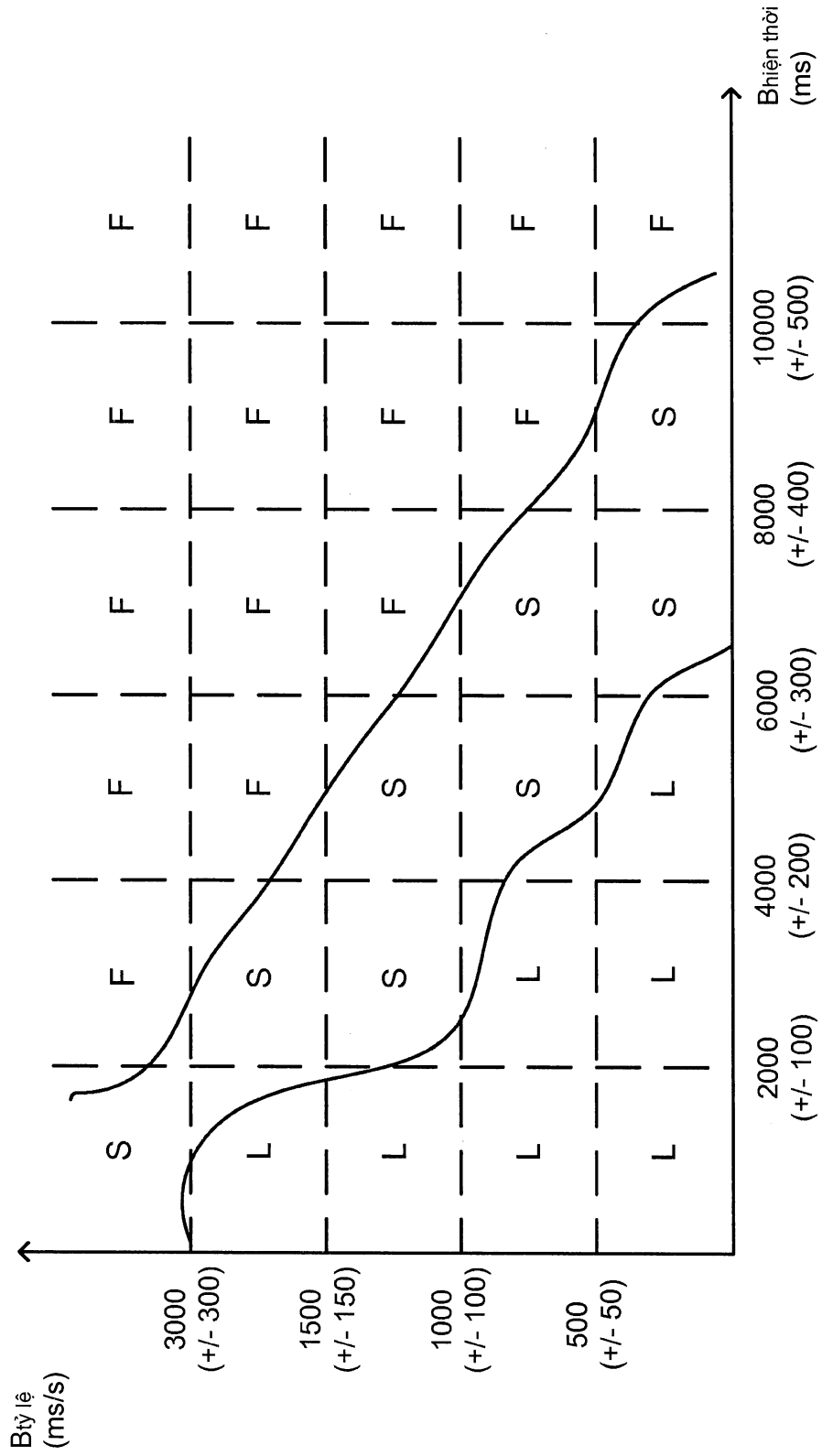


Fig. 15 L = thấp, S = ổn định, F = đầy đủ

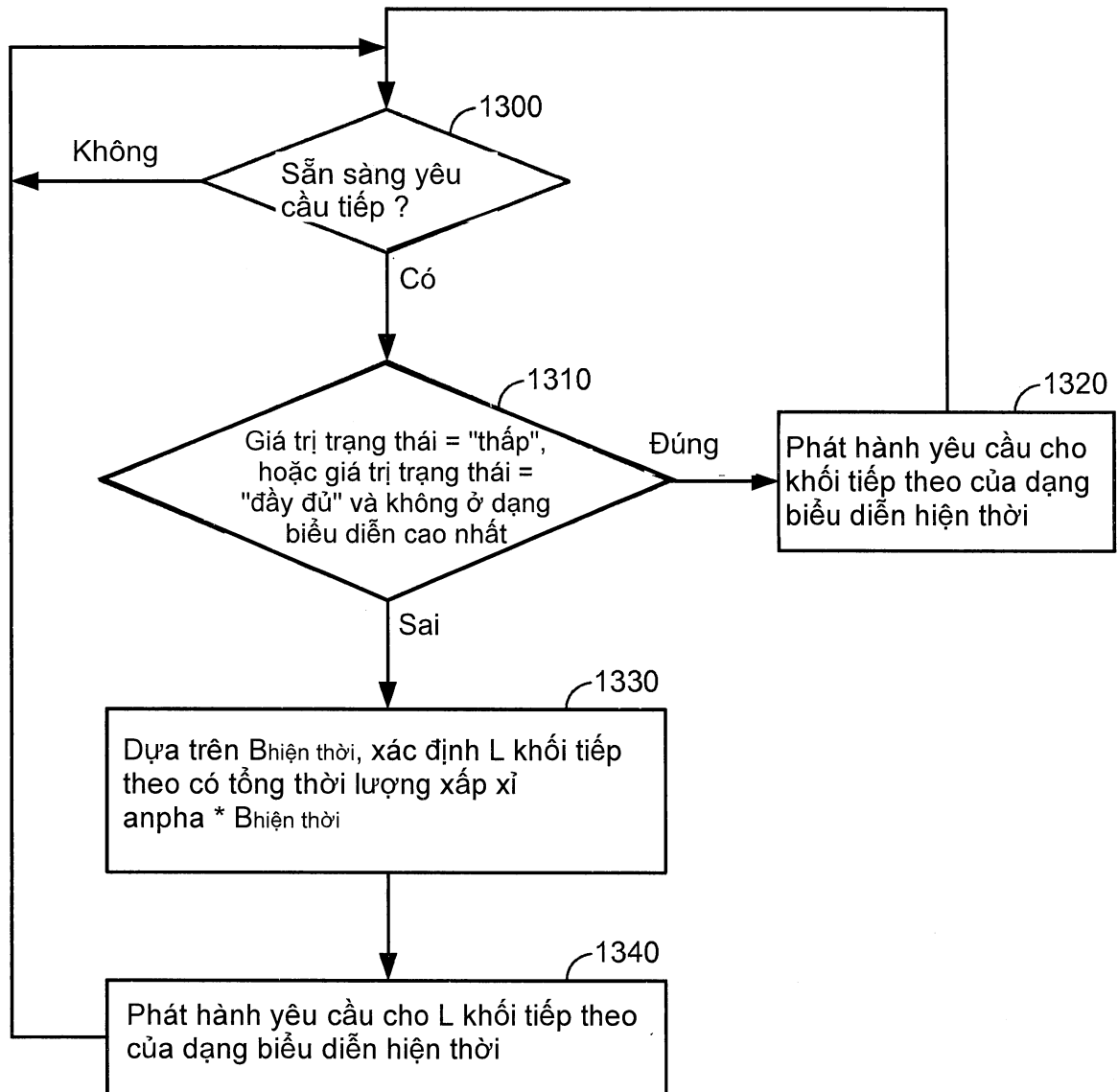


FIG. 16

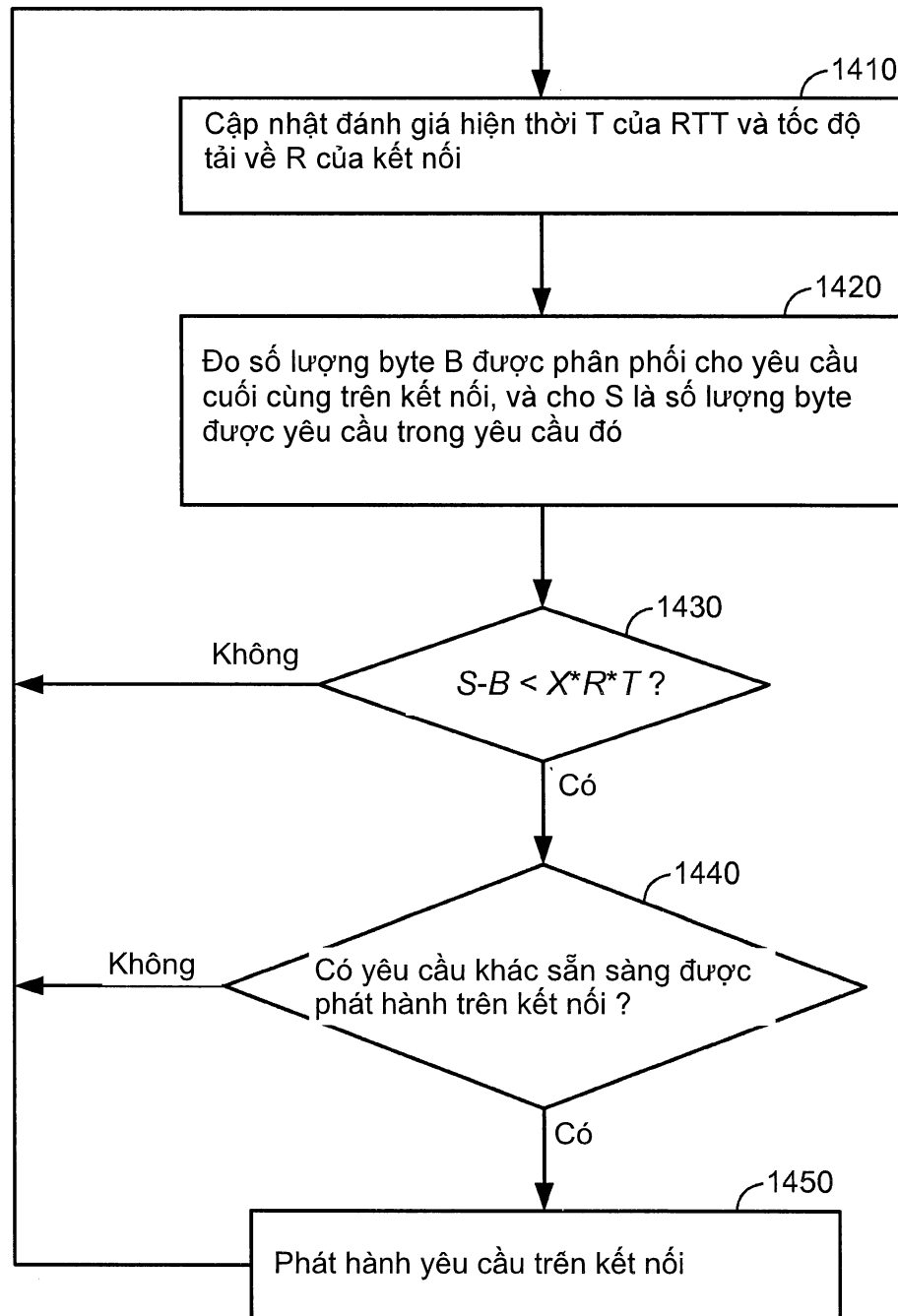


FIG. 17

Danh sách yêu cầu ứng viên	Mức ưu tiên	Được cho kết nối 1 ?	Được cho kết nối 2 ?	Được cho kết nối 3 ?
Yêu cầu A	0	X		
Yêu cầu B	0	X	X	
Yêu cầu C	1	X		
Yêu cầu D	1			X
Yêu cầu E	2			X
Yêu cầu F	3		X	X

FIG. 18

Danh sách yêu cầu ứng viên	Mức ưu tiên	Được cho kết nối 1 ?	Được cho kết nối 2 ?	Được cho kết nối 3 ?
Yêu cầu A	5	X		
Yêu cầu G	0			X
Yêu cầu C	4	X		
Yêu cầu D	1			X
Yêu cầu E	2			X
Yêu cầu F	3		X	X

FIG. 19

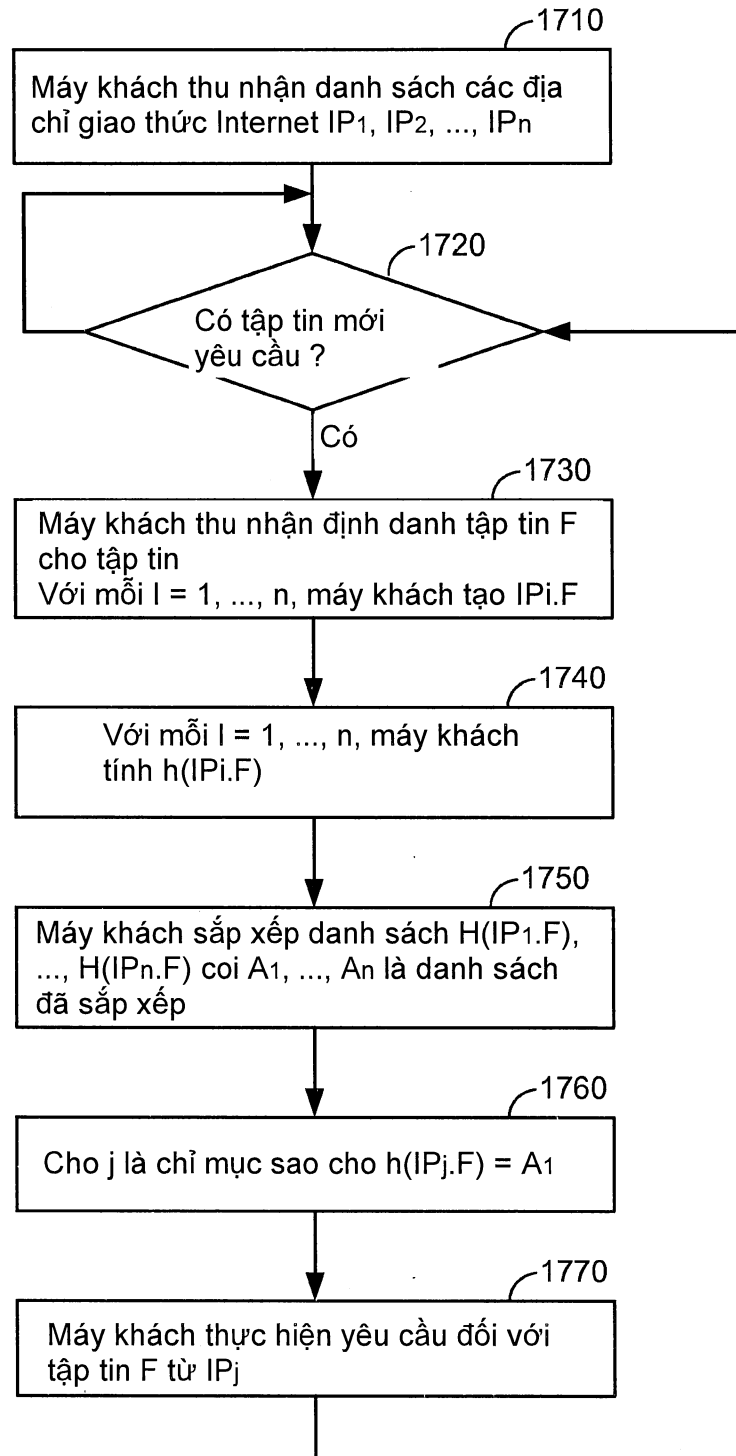


FIG. 20

```

<expression> ::= <literal> |
               <variable> |
               <function> |
               '(' <expression> ')' |
               <expression> <operator> <expression>

<literal> ::= <string> | <number>

<variable> ::= <token>

<function> ::= <token> '(' [ <expression> * (',' <expression> ) ] ')'

<operator> ::= 1* <opchar>

<token> ::= <tokenchar> * ( <tokenchar> | <digit> )

<string> ::= '"' * <char> '"'

<number> ::= [ '-' ] 1* <digit> [ '.' 1* <digit> ]

<digit> ::= '0-9'

<char> ::= <any ASCII char except '"'> | '/'

<tokenchar> ::= 'A-Z' | 'a-z' | '_'

<opchar> ::= '-' | '+' | '=' | '/' | ';' | '%' | '*' |

```

FIG. 21

```

unsigned long Hash( const char *p, unsigned long max )
{
    unsigned long hash = 0;
    while( *p != 0 )
        hash = ( hash << 5 ) ^ ( ( hash & 0xf8000000 ) >> 27 ) ^ *p++;
    return hash % max;
}

```

FIG. 22

Ví dụ 1

Tiêu đề	Ánh xạ miền cố định đơn giản
Quy tắc xây dựng định danh tập tin	<code>printf("http://<domainname>/%s_%3d.dfx", id, seq)</code>
ID tập tin	SPEED_PAO_S2_E1_5gsyd75
Số thứ tự	22
Kết quả	<code>http://<domainname>/SPEED_PAO_S2_E1_5gsyd75_022.dfx</code>

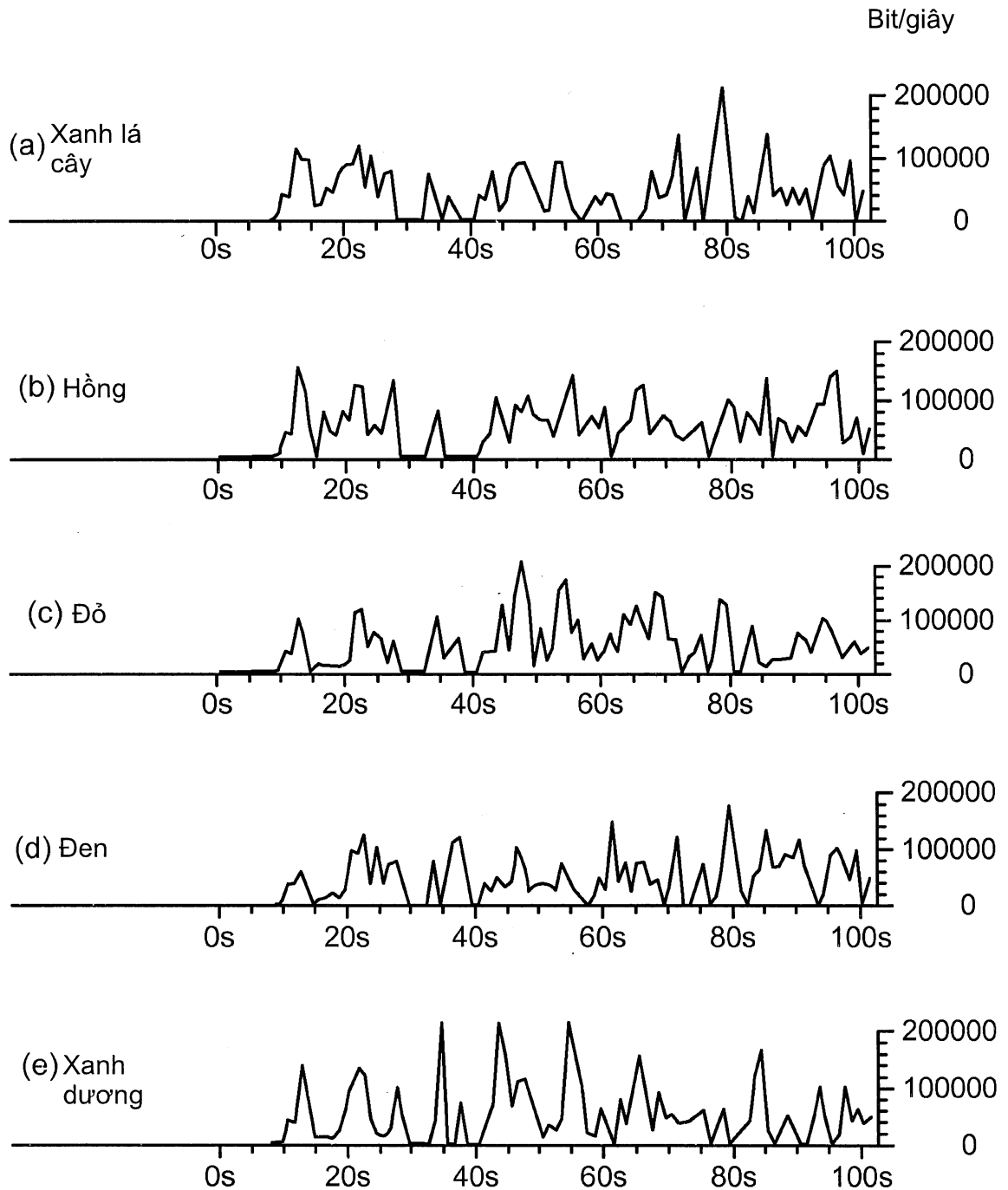
Ví dụ 2

Tiêu đề	Giá trị băm trên 1000 miền dựa trên ID tập tin
Quy tắc xây dựng định danh tập tin	<code>printf("http://%3d.<domainname>/%s_%3d.dfx", hash(id, 1000), id, seq)</code>
ID tập tin	SPEED_PAO_S2_E1_5gsyd75
Số thứ tự	22
Kết quả	<code>http://564.<domainname>/SPEED_PAO_S2_E1_5gsyd75_022.dfx</code>

Ví dụ 3

Tiêu đề	Giá trị băm trên 10 miền và 1000 danh mục
Quy tắc xây dựng định danh tập tin	<code>printf("http://%2d.<domainname>/%3d/%s_%3d.dfx", hash(id, 10), hash(id, 1000), id, seq)</code>
ID tập tin	SPEED_PAO_S2_E1_5gsyd75
Số thứ tự	22
Kết quả	<code>http://4.<domainname>/564/SPEED_PAO_S2_E1_5gsyd75_022.dfx</code>

FIG. 23

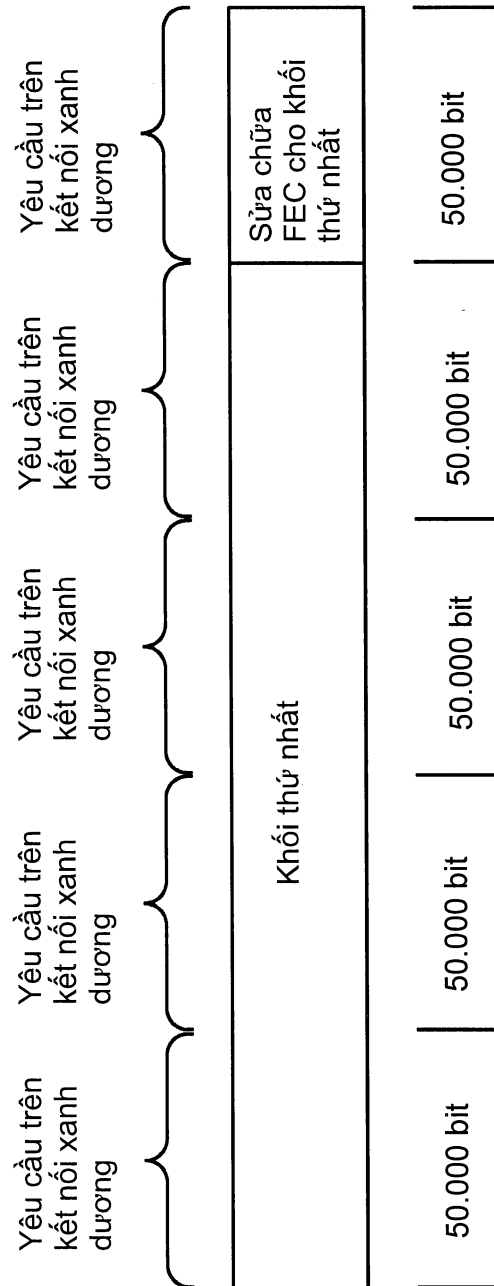


Dòng thời gian của các kết nối TCP

Sự dao động bằng thông điệp hình của 5 kết nối TCP

FIG. 24

Nhiều yêu cầu HTTP cho dữ liệu nguồn + dữ liệu sửa chữa



Ví dụ về thời gian sửa đổi kênh có và không có FEC

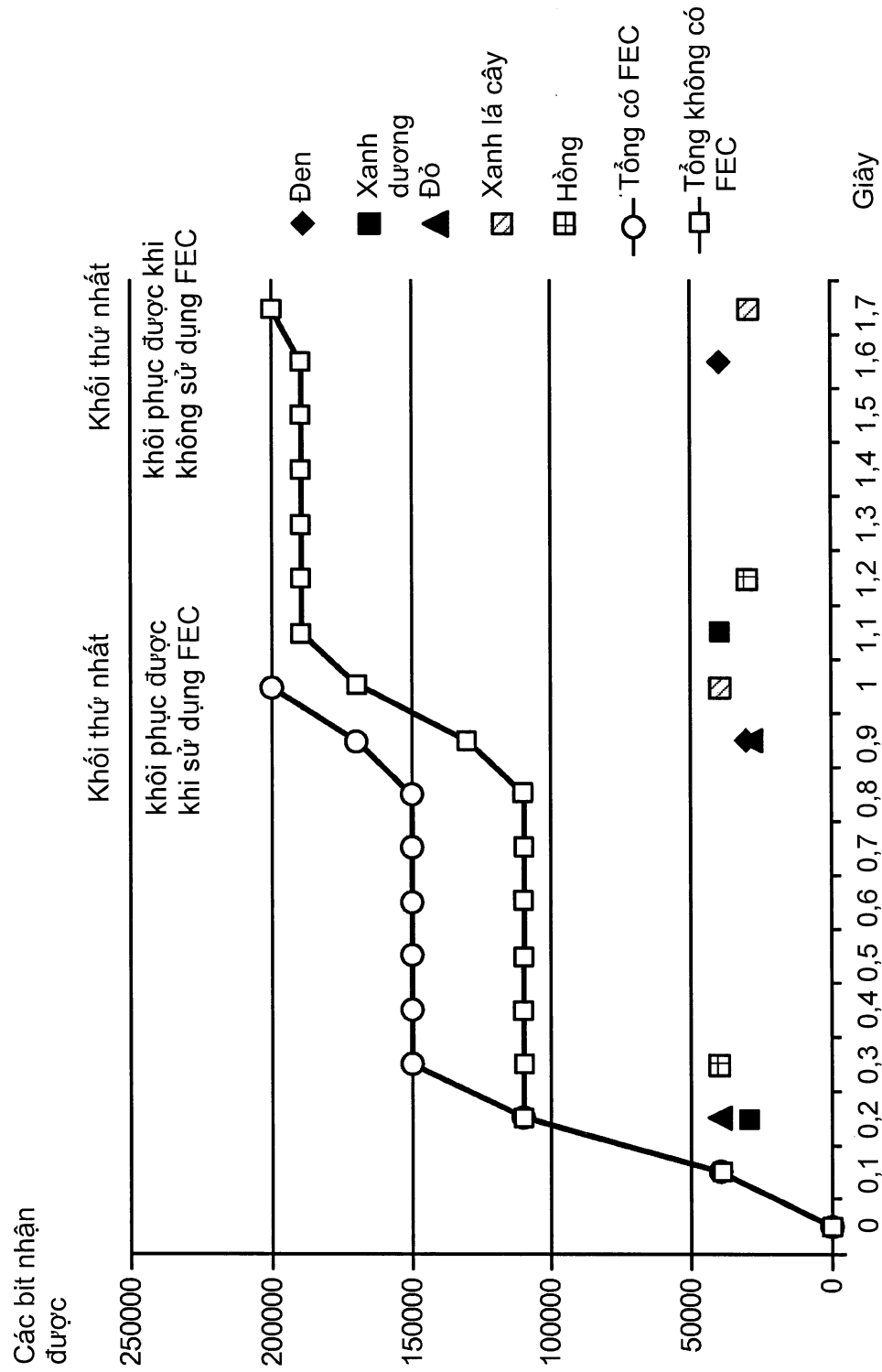


FIG. 26

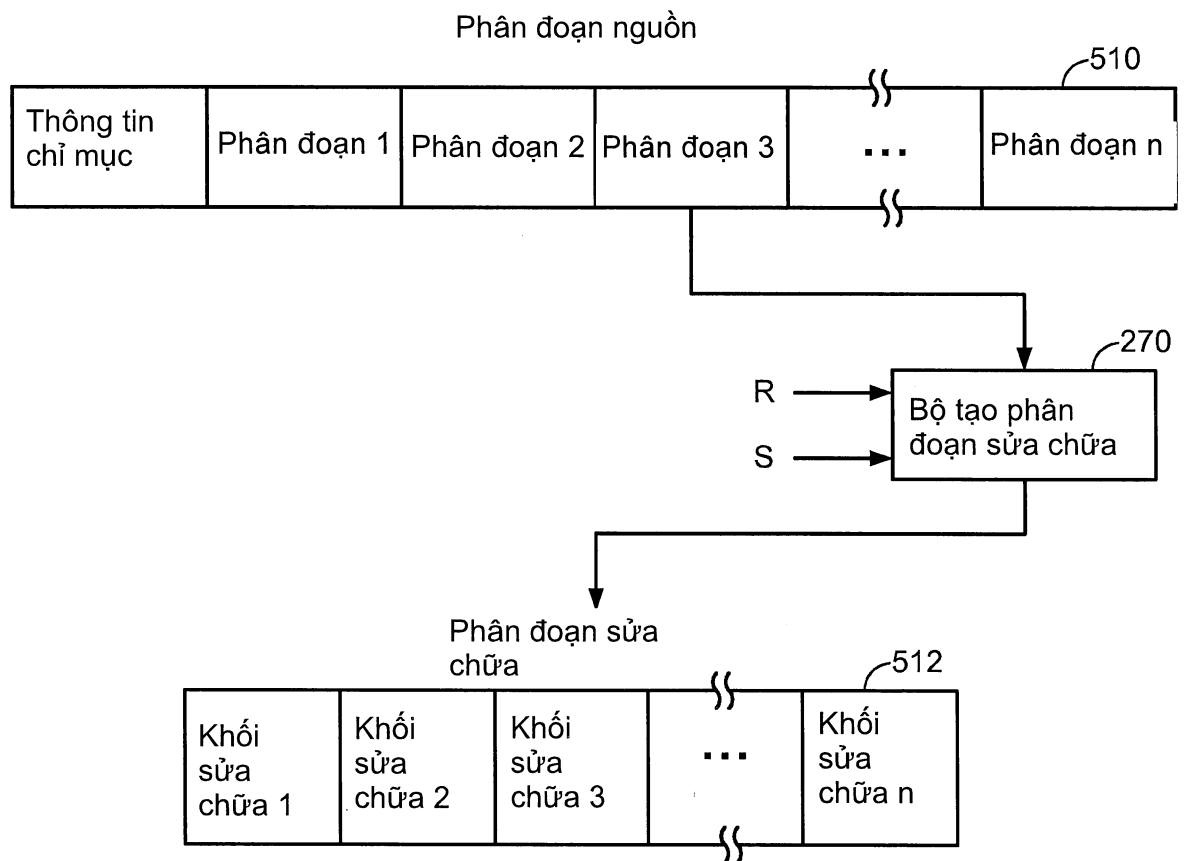


FIG. 27

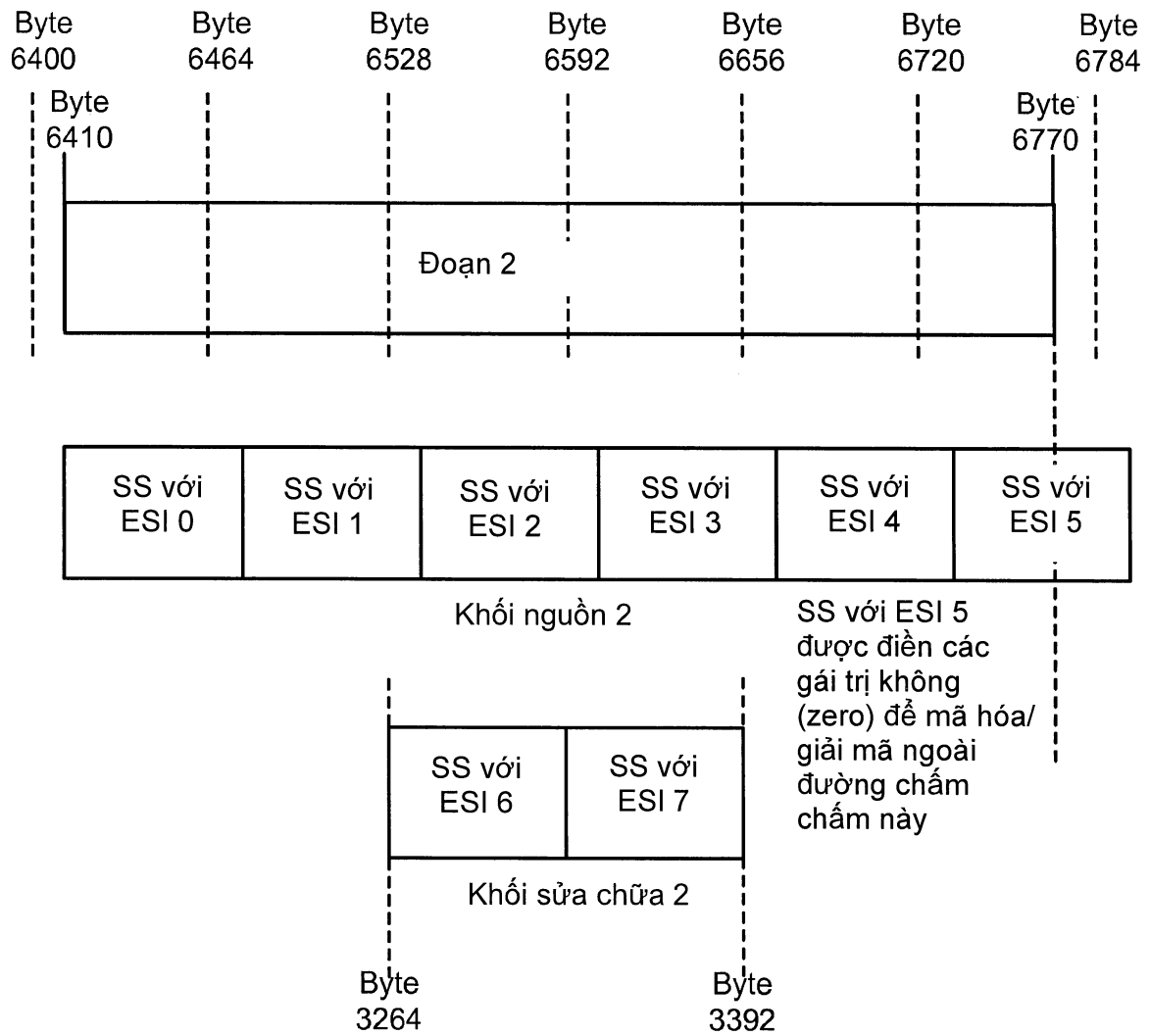


FIG. 28

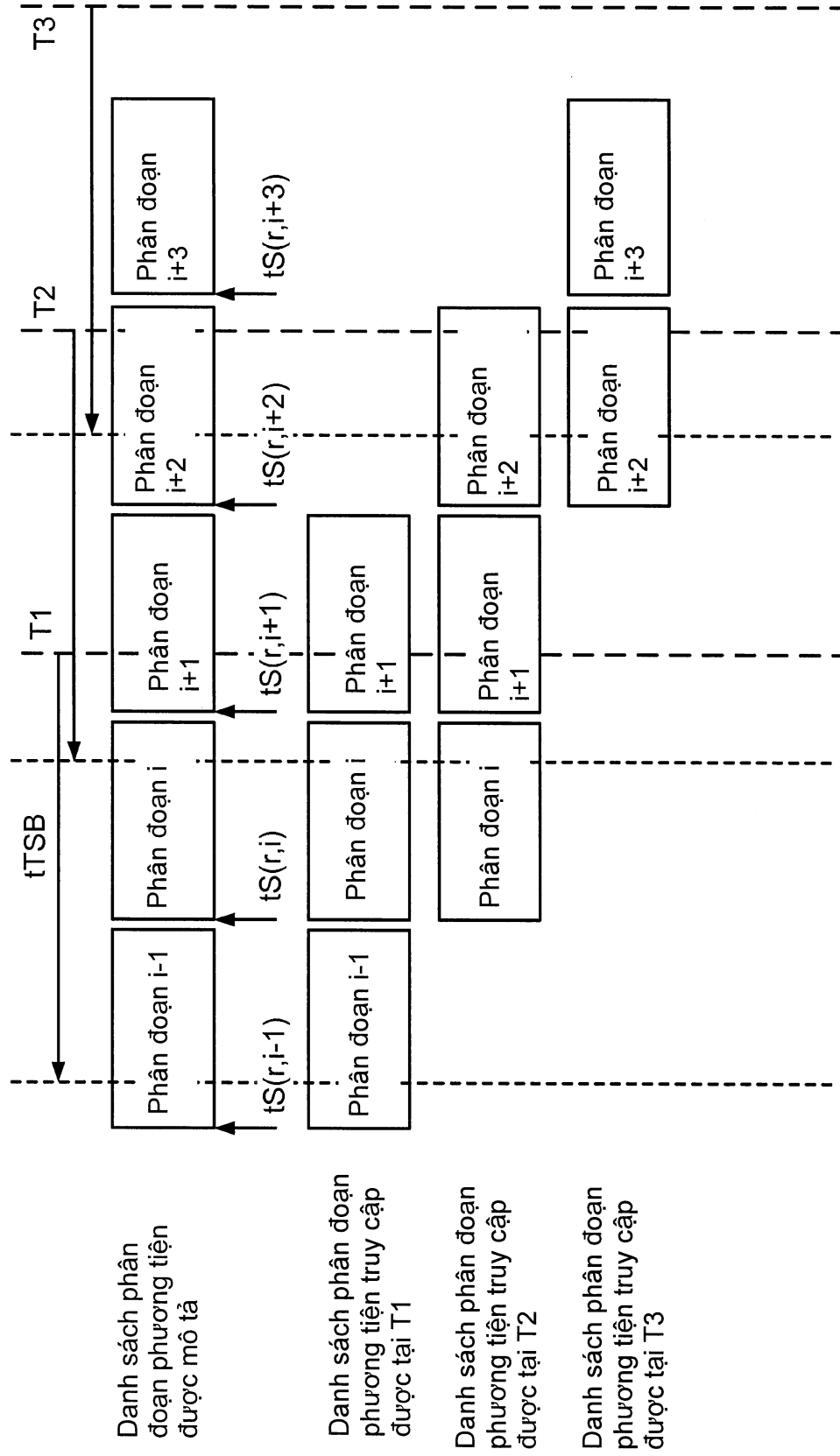


FIG. 29

