PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Rodrigo Laiola Guimarães

Caracterização dos mecanismos de encapsulamento para integração de serviços nas redes IP e MPEG

MONOGRAFIA DA DISCIPLINA DE REDES DE ALTA VELOCIDADE

DEPARTAMENTO DE INFORMÁTICA

Programa de Pós-Graduação em Informática

Rio de Janeiro
Junho de 2005

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Rodrigo Laiola Guimarães

Caracterização dos mecanismos de encapsulamento para integração de serviços nas redes IP e MPEG

Monografia da Disciplina de Redes de Alta Velocidade

Monografia apresentada como requisito parcial para aprovação na disciplina de Redes de Alta Velocidade do Programa de Pós-Graduação em Informática da PUC-Rio.

Orientador: Luiz Fernando Gomes Soares
Co-orientador: Marcio Ferreira Moreno

Rio de Janeiro, junho de 2005

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Rodrigo Laiola Guimarães

Caracterização dos mecanismos de encapsulamento para integração de serviços nas redes IP e MPEG

Monografia apresentada como requisito parcial para aprovação na disciplina de Redes de Alta Velocidade do Programa de Pós-Graduação em Informática da PUC-Rio.

Luiz Fernando Gomes Soares
Orientador
Departamento de Informática - PUC-Rio

Marcio Ferreira Moreno
Co-orientador
Departamento de Informática - PUC-Rio

Rio de Janeiro, 28 de junho de 2005

Rodrigo Laiola Guimarães

Graduado em Engenharia de Computação pela Universidade Federal do Espírito Santo (UFES) em 2004. Atualmente, integra o grupo de pesquisadores do Laboratório TeleMídia da PUC-Rio, desenvolvendo pesquisa na área de Redes de Computadores e Sistemas Hipermídia.

Ficha Catalográfica

Guimarães, Rodrigo Laiola

Descrição da Ficha Catalográfica

Informação Técnica da Ficha Catalográfica

Natureza da Ficha Catalográfica

Inclui referências bibliográficas.

Convergência de redes; MPEG sobre IP; IP sobre MPEG

Dedico este trabalho a todos àqueles que acreditam que a ousadia e o erro são caminhos para as grandes realizações.

Agradecimentos

Agradeço a minha família pelo incentivo e pela compreensão durante toda minha vida e trajetória acadêmica.

Aos Professores da Universidade Federal do Espírito Santo por minha formação na Graduação, sem a qual não teria chegado até aqui.

Minha sincera gratidão e admiração pelo meu orientador Luiz Fernando Gomes Soares por sua tamanha dedicação e incessante esforço em me ajudar durante o desenvolvimento deste trabalho de pesquisa científica.

Ao meu co-orientador, Marcio Moreno, por sua atenção, paciência e boa vontade em me ensinar, tendo sido um verdadeiro guia para o desenvolvimento deste trabalho, sua ajuda foi fundamental.

Aos meus colegas do TeleMídia, pelo companheirismo e ajuda prestados. Em especial a Rodrigo Borges pela amizade e discussões que ajudaram no desenvolvimento deste trabalho.

Agradeço a todos os funcionários e alunos da PUC-Rio pela boa convivência que tivemos durante estes poucos meses.

À CAPES e ao CNPq pelo apoio financeiro.

Acima de tudo e todos, agradeço a Deus por sempre ter iluminado meu caminho.

Resumo

Guimarães, Rodrigo Laiola. **Caracterização dos mecanismos de encapsulamento para integração de serviços nas redes IP e MPEG.** Rio de Janeiro, 2005. 48p. Monografia da Disciplina de Redes de Alta Velocidade - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O desempenho das redes de comunicação tem evoluído, e isso tem permitido a implementação de novos serviços que foram originalmente desenvolvidos para outros tipos de redes. Esse advento é usualmente chamado de convergência de redes, ou, convergência tecnológica e o resultado disso pode ser uma significante economia em infraestrutura, desenvolvimento, suporte e custos de gerenciamento. Nesse contexto, este trabalho descreve um conjunto de mecanismos de encapsulamento para integração de serviços existentes em redes IP e MPEG. São detalhados os mecanismos de transporte de fluxos MPEG-2 e MPEG-4 através de datagramas IP, assim como detalhados maneiras de datagramas IP serem transportados através de redes MPEG.

Palavras-chave

Convergência de redes; MPEG sobre IP; IP sobre MPEG

Sumário

Lista de tabelas

Lista de figuras

1 Introdução

O desempenho das redes de comunicação tem evoluído muito, e enquanto esse aumento de desempenho promoveu a eficiência dos serviços sobre as redes, também permitiu que fossem implementados novos serviços que foram desenvolvidos para outros tipos de redes (por exemplo, telefonia sobre a Internet ou comunicação de dados sobre a rede de TV a cabo).

No passado, esse desenvolvimento conduziu à substituição de redes de serviço específico, como por exemplo, a substituição do telex pelo serviço de fax na rede de telefone. Entretanto, atualmente, é observada uma integração de redes diferentes, usualmente chamada de convergência de redes, ou, convergência tecnológica. O resultado disso pode ser uma significante economia em infraestrutura, desenvolvimento, suporte e custos de gerenciamento.

Este capítulo descreve as motivações e os objetivos desta monografia, assim como apresenta sua estrutura.

1.1. Motivação

Inicialmente, a Internet foi concebida para o transporte de conteúdo estático (mídias discretas), basicamente dados textuais. Com o aumento do poder de processamento dos computadores e com enorme abrangência da Internet, tem crescido a demanda por serviços que possibilitem o transporte de dados de outras naturezas (mídias discretas) como, por exemplo, áudio e vídeo. Neste contexto, surge a necessidade de mecanismos que sejam capazes de transmitir vídeo e áudio em tempo real, recuperar o sincronismo, fazer a detecção de erros e, finalmente, possibilite o desenvolvimento de sistemas robustos.

Em outro cenário, nota-se que a evolução das técnicas de codificação digital de áudio e vídeo, aliada aos novos esquemas eficientes de modulação para transmissões digitais, tornaram possível o advento da TV digital e com ela uma vasta gama de novos serviços. A possibilidade de encapsulamento de dados para a difusão em conjunto com o áudio/vídeo abre diversas alternativas para a provisão

de serviços avançados. As emissoras poderão não somente disponibilizar uma programação de alta qualidade de imagem e som, mas também torná-la mais atraente, permitindo aos usuários interagirem com os programas sendo assistidos. Cabe ao sistema de TV digital, portanto, prover meios para que esses tipos de funcionalidades sejam oferecidos, definindo padrões de codificação, recuperação, sincronização e tratamento dos dados difundidos.

1.2. Objetivos

Esta monografia tem como principal objetivo detalhar os mecanismos de transporte de fluxos MPEG-2 e MPEG-4 através de datagramas IP, assim como detalhar as diversas maneiras que datagramas IP podem ser transportados através de redes MPEG.

O estudo sobre transporte de fluxos MPEG-2 e MPEG-4 através de datagramas IP será direcionado a aplicações de tempo real, nas quais deseja-se que o receptor execute os fluxos de diversos tipos de mídias à medida que os mesmos são recebidos. Para isso, serão apresentados mecanismos (protocolos) que podem ser utilizados em redes IP para que seja possível o transporte de fluxos com essas características.

Por sua vez, o transporte de datagramas IP, através de redes MPEG, será direcionado ao detalhamento dos mecanismos oferecidos por essas redes para o transporte de dados não-audiovisuais, em especial, datagramas IP.

1.3. Estrutura da Monografia

Esta monografia encontra-se organizada como a seguir. O Capítulo 2 discute os conceitos básicos dos padrões MPEG-2 e MPEG-4.

O Capítulo 3 apresenta um protocolo que é capaz transmitir áudio e vídeo MPEG em tempo real em redes IP, e dentre outras coisas, é capaz de recuperar o sincronismo. Nesse capítulo são detalhadas as particularidades para o transporte de áudio e vídeo utilizando esse protocolo.

O Capítulo 4 destaca as técnicas de transporte de datagramas IP através de redes MPEG.

Por fim, o Capítulo 5 tece as conclusões e descreve trabalhos futuros.

2 Conceitos básicos de codificação de áudio e vídeo

Com o intuito de estabelecer padrões internacionais para a representação e codificação de informações audiovisuais em formato digital, a ISO (*International Organization for Standardization*) e a IEC (*International Electrotecnical Commission*) formaram o grupo MPEG (*Motion Picture Coding Experts Group*), que iniciou seus trabalhos em maio de 1988. A família de padrões produzidos por esse grupo ficou popularmente conhecida como padrões MPEG e inclui, entre outros, os conjuntos de padrões MPEG-2 e MPEG-4. Esses padrões são apresentados a seguir.

2.1. MPEG-2

O padrão MPEG-2 foi iniciado em 1990 e publicado em 1995. O objetivo desse padrão é obter, para sinais de vídeo, taxas entre 1,5 Mbps e 15 Mbps, adequadas para sinais de televisão padrão (*Standard Definition Television – SDTV*), e entre 15 Mbps e 30 Mbps, para sinais de televisão de alta definição (*High Definition Television – HDTV*). O MPEG-2 é descrito no conjunto de padrões ISO/IEC 13818.

O padrão ISO/IEC 13818-1 (ISO/IEC 13818-1, 2000) (MPEG-2 *Systems*) estabelece como um ou mais sinais de áudio e vídeo, assim como outros dados (imagens estáticas, texto etc.), devem ser combinados de forma a serem transmitidos ou armazenados apropriadamente. Assim, questões como a multiplexação de sinais de áudio, vídeo e dados, gerenciamento de *buffers* nos decodificadores, sincronização entre mídias distintas durante a decodificação e a apresentação e respectiva identificação das mesmas são abordadas nesse padrão.

O MPEG-2 *Systems* define um *programa* como um conjunto de *fluxos elementares*, correspondendo a sinais de vídeo, áudio e dados, que podem ter algum relacionamento temporal entre si. Para os elementos que possuem relacionamento temporal, a base de tempo utilizada é comum a todos.

A partir da definição de programa, define-se dois formatos de multiplexação de fluxos no MPEG-2 *Systems*: o Fluxo de Transporte (*MPEG Transport Stream – MTS*) e o Fluxo de Programa (*MPEG Program Stream – MPS*). O *Fluxo de Transporte* pode conter múltiplos programas simultaneamente e é apropriado para transmissão e armazenamento em ambientes ruidosos, onde a ocorrência de erros é freqüente. Cada programa pode ter uma base de tempo diferente. O *Fluxo de Programa* é apropriado para uso em ambientes com baixa taxa de erros. Como um Fluxo de Programa só pode conter um programa, todos os fluxos elementares utilizam a mesma base de tempo. A estrutura de multiplexação do MPEG-2 *Systems* é apresentada na FIGURA 1.

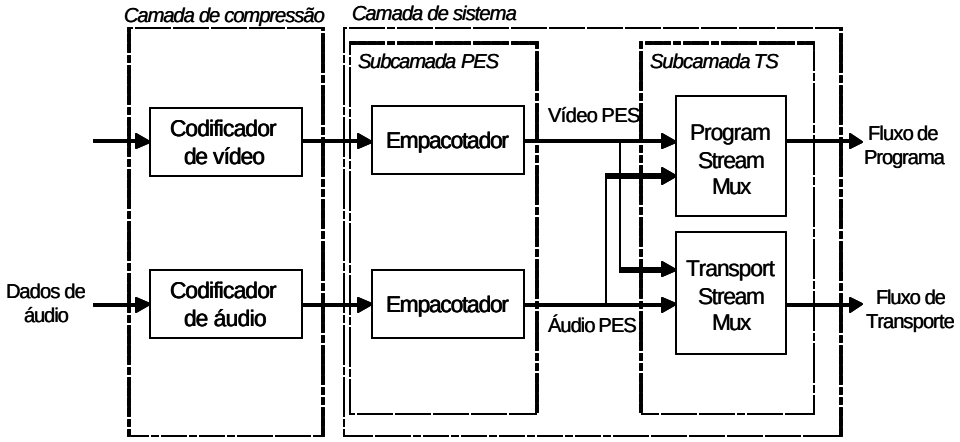

FIGURA 1. Estrutura de multiplexação do MPEG-2 (Soares et al., 2004)

Fluxos de Transporte podem apresentar taxa fixa ou variável, tal como cada programa ou fluxo elementar que o compõe. Conforme comentado anteriormente, todos os fluxos elementares que compõem um programa contido em um Fluxo de Transporte compartilham a mesma base de tempo, definida por um relógio de sistema comum (*System Time Clock*). Assim, em cada programa, uma base de tempo é definida por meio de uma referência de relógio (*Program Clock Reference – PCR*), sendo utilizada para a sincronização de todos os seus fluxos. Cabe ressaltar que diferentes programas podem ser formados a partir de diferentes bases de tempo.

Em um Fluxo de Transporte são permitidos até 32 fluxos de áudio e 16 fluxos de vídeo, além de fluxos de dados e de *metadados*, sendo que cada fluxo elementar possui uma identificação própria (*Packet Identification – PID*).

Para um Fluxo de Transporte, a estrutura de multiplexação pode ser dividida em duas camadas: *Camada de Compressão* e *Camada de Sistema*. A *camada de compressão* refere-se à codificação de cada um dos fluxos elementares. A *camada de sistema*, por sua vez, é dividida em duas subcamadas: a subcamada *Packetized Elementary Stream* (PES) e a subcamada *Transport Stream* (TS). A subcamada PES é responsável pela inserção de informações de sincronização entre partes de um fluxo e entre diferentes fluxos e pelo controle de *buffers*. Os fluxos elementares de áudio e vídeo, previamente codificados na camada de compressão, são transportados como carga útil dos pacotes definidos pela subcamada PES. Cada pacote PES contém uma ou mais unidades de apresentação de cada fluxo elementar (quadro ou figura, no caso de vídeo, uma ou mais amostras, no caso de áudio, por exemplo). Os pacotes PES, por sua vez, são multiplexados nos pacotes definidos pela subcamada TS.

2.2. MPEG-4

Dentre os padrões definidos pelo MPEG, o MPEG-4 (Koenen, 2002) destaca-se por oferecer não somente a codificação e transporte de objetos audiovisuais, mas também a capacidade de descrição de cenas baseadas nesses objetos (vídeo, áudio, texto, imagem etc). No MPEG-4, os objetos multimídia são codificados de modo independente, mas entre eles podem ser estabelecidos relacionamentos de forma a permitir a exibição de uma apresentação multimídia.

Os estudos para o desenvolvimento do MPEG-4 foram iniciados em 1993, porém, sua primeira versão somente foi apresentada em outubro de 1998 e a segunda versão em dezembro de 1999. Assim como nos outros padrões MPEG, o MPEG-4 é especificado como um conjunto de padrões (ISO/IEC 14496).

A maior parte da especificação baseada em objetos do MPEG-4 é definida na sua Parte 1. O MPEG-4 *Systems* (ISO/IEC 14496-1, 2001) destaca-se por sua relação direta com a produção de apresentações hipermídia/multimídia. Suas definições baseiam-se no conceito de que uma cena audiovisual é composta de objetos, cujas propriedades são definidas utilizando-se uma linguagem para descrição de cenas.

No MPEG-4, os objetos de mídia são codificados de modo independente gerando vários fluxos elementares. A partir dessa característica, técnicas de compactação e compressão podem ser aplicadas individualmente para cada objeto, aproveitando as características de cada mídia. Após a codificação e a compressão/compactação e antes da transmissão, os fluxos MPEG-4 são sincronizados de acordo com as relações previamente definidas na fase de autoria do documento.

A *camada de entrega*, também denominada *camada de transporte*, é a última camada antes da passagem dos fluxos MPEG-4 para os meios de transmissão.

Todos os serviços oferecidos pela camada de transporte são definidos e coordenados pela interface definida na Parte 6, DMIF (*Delivery Multimedia Integration Framework*) (ISO/IEC 14496-6, 2000), do MPEG-4. Através do DMIF são definidos os canais de transporte, independente do protocolo adotado, utilizando a interface DAI (*DMIF Application Interface*). Essa interface permite também o gerenciamento de sessões, oferecendo uma abstração para as aplicações no mapeamento dos fluxos MPEG-4 para os canais de transporte.

Para exemplificar as funções do DMIF, pode ser interessante compará-lo a um protocolo com funções próximas e bastante conhecido: o FTP (*File Transfer Protocol*). A princípio, o DMIF é bastante parecido com o FTP e a diferença essencial está no fato de que o FTP retorna um conjunto de dados, enquanto que o DMIF retorna ponteiros indicando o caminho para recuperação de um determinado fluxo de dados. Quando o DMIF é executado, do ponto de vista do cliente, a primeira ação realizada é o estabelecimento de uma sessão com o servidor remoto. Após o estabelecimento da sessão, os fluxos são selecionados e ocorre uma requisição para que eles comecem a ser enviados (*streaming*). Na camada de entrega onde a conexão foi solicitada, ponteiros serão retornados para as conexões de transporte onde os fluxos serão recuperados, estabelecendo assim a comunicação entre os dois lados.

Além de oferecer funcionalidades de um protocolo, o DMIF, através da DAI, permite utilizar serviços de protocolos diversos, funcionando como uma estrutura genérica (*framework*) para acesso a serviços de rede. Dessa forma, as mensagens definidas pela DAI podem variar dependendo da rede de comunicações em

operação. Um outro ponto importante considerado no projeto do DMIF é a Qualidade de Serviço; a DAI permite que o usuário DMIF especifique os requisitos necessários para o fluxo desejado. A especificação DMIF oferece sugestões de como executar tarefas que negociem a QoS (Qualidade de Serviço) em um certo número de tipos de rede, como por exemplo para a Internet. A interface DAI é também utilizada para oferecer acesso a materiais difundidos em *broadcast* como também arquivos locais, consolidando o seu princípio de uma interface única e uniforme para acessar conteúdo multimídia em uma variedade de tecnologias de entrega.

3 Transporte de fluxo MPEG através de datagramas IP

Inicialmente, a Internet foi concebida para o transporte de mídias discretas, basicamente dados textuais. Com o aumento do poder de processamento dos computadores e com imensa abrangência da Internet, tem crescido a demanda por serviços que possibilitem o transporte de mídias contínuas como, por exemplo, áudio e vídeo.

Uma idéia inicial seria utilizar os protocolos já existentes para o transporte desses dados não-textuais. Contudo, esses protocolos geralmente não provêem todas as funcionalidades necessárias para tal transporte.

Em aplicações de tempo real deseja-se que o receptor execute os fluxos de diversos tipos de mídias à medida que os mesmos são recebidos, ao invés de armazená-los todo em um arquivo para só então executá-los. As mídias de áudio e vídeo normalmente toleram perdas, porém são extremamente rígidas quanto ao retardo e a variações estatísticas do retardo.

Na arquitetura IP, o protocolo TCP (*Transmission Control Protocol*) da camada de transporte faz controle de erro e fluxo, e isso pode comprometer aplicações de tempo real. Como se pode notar, nessas condições o TCP não é adequado para o transporte de áudio e vídeo. Já o protocolo de transporte UDP (*User Datagram Protocol*), que provê um serviço não confiável, deveria ser satisfatório, contanto que a variação do tempo de trânsito na rede possa ser caracterizada e as taxas de perda sejam aceitáveis. Não obstante, o UDP não oferece técnicas de recuperação de sincronismo, retransmissão e correção de erro.

Neste contexto, surge a necessidade de um protocolo que seja capaz de transmitir vídeo e áudio em tempo real, recuperar o sincronismo, fazer a detecção de erros e, finalmente, possibilite o desenvolvimento de sistemas robustos. Para que sejam possíveis serviços de vídeo sob demanda, além de um protocolo de transporte de áudio e vídeo em tempo real, necessita-se de um mecanismo que possibilite a interatividade do usuário na distribuição de vídeo. Protocolos que transportam fluxos MPEG e que possuem tais características serão explorados nas próximas seções deste capítulo.

3.1. RTP

O protocolo para transporte em tempo real (*RTP – Real-time Transport Protocol)* (RFC 1889, 1996), foi desenvolvido pelo grupo de trabalho *Audio-Video transport* do *Internet Engineering Task Force* (IETF) e posteriormente adotado por outras organizações de padronização. Este é o principal padrão para transporte de áudio/vídeo em redes de IP, junto com seus perfis associados e formatos de carga útil.

O RTP provê funções de transporte fim-a-fim para que aplicações possam transmitir dados em tempo real, como áudio e vídeo, ou dados de simulação, sobre serviços *multicast* ou *unicast*. Como o próprio nome diz, o RTP provê garantias de entrega em tempo real.

Tipicamente, o RTP é construído sobre UDP/IP, podendo aceitar perdas, mas também tendo a flexibilidade para usar técnicas de recuperação, como retransmissão e correção de erro quando necessário. Essas técnicas dão a uma aplicação grande flexibilidade para tratar problemas de rede de uma maneira satisfatória, ao invés de se limitarem às carências da camada de transporte da arquitetura IP.

Apesar da maioria das implementações do RTP serem feitas sobre UDP/IP, esta não é a única forma possível, ou seja, nada no protocolo do RTP requer UDP ou IP. Como exemplo, em algumas implementações utiliza-se RTP sobre TCP/IP, e em outros casos RTP em redes não-IP, como nas redes ATM (*Asynchronous Transfer Mode*).

A FIGURA 2 mostra como o protocolo RTP é geralmente inserido na pilha de protocolos TCP/IP.

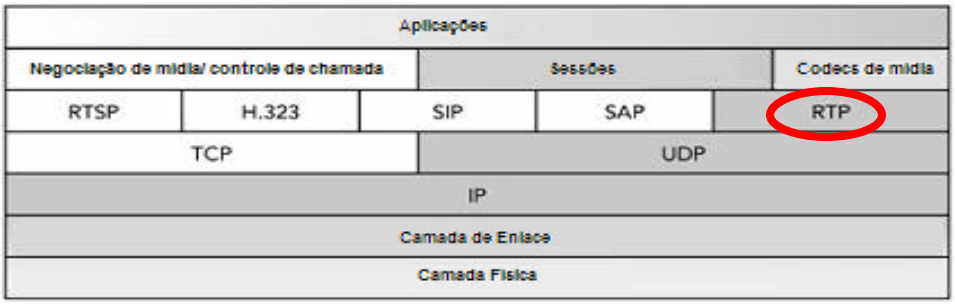

FIGURA 2. Pilha de protocolos multimídia (Perkins, 2003)

Antes de analisar o protocolo RTP mais a fundo, é interessante ter uma visão geral das responsabilidades dos transmissores e receptores em um sistema. O transmissor é responsável por capturar os dados audiovisuais para transmissão, como também, gerar os pacotes RTP. Ele também pode participar na correção de erro e controle de congestionamento, adaptando o fluxo de mídia transmitido após receber um retorno do receptor. A FIGURA 3 apresenta o diagrama do processo de envio.

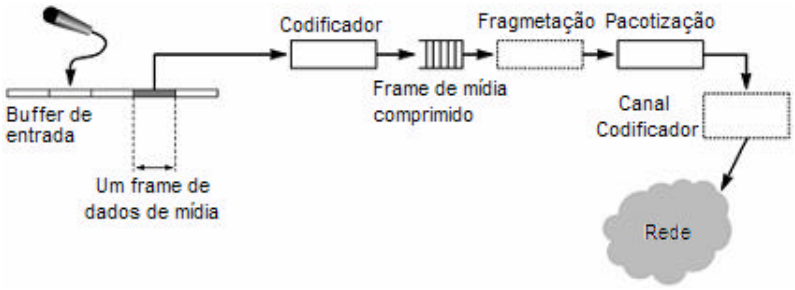

FIGURA 3. Diagrama de blocos de transmissor RTP (Perkins, 2003)

A mídia não comprimida é colocada em um buffer, do qual se produz *frames* comprimidos. Estes *frames* podem ser codificados de diversas formas dependendo do algoritmo de compressão utilizado.

Os *frames* comprimidos são então carregados em pacotes RTP para envio. Se os frames são muito grandes, eles podem ser fragmentados em vários pacotes RTP. Caso eles sejam pequenos, outros *frames* podem ser colocados dentro do mesmo pacote RTP. Dependendo do esquema de correção de erro utilizado, um canal codificador pode ser usado para fazer a correção de erro dos pacotes ou para reordenar os pacotes antes de transmiti-los.

Depois que os pacotes RTP são enviados, o *buffer* de mídia que contém os dados enviados é liberado. Entretanto, o transmissor não pode descartar estes dados, pois os mesmos podem ser necessários para correção de erro. Isso pode significar que o transmissor deve armazenar os dados por um tempo após os pacotes terem sido enviados, dependendo do *codec* ou esquema de correção de erro utilizado.

O transmissor é responsável por gerar um relatório periódico de status para os fluxos de mídia que estão sendo gerados. Ele também recebe uma avaliação de

qualidade de recepção de outros participantes e pode usá-la para adaptar sua transmissão.

O receptor por sua vez, é responsável por retirar os pacotes RTP da rede, corrigir qualquer perda, recuperar a linha do tempo, descomprimir a mídia, e apresentar o resultado para o usuário. Além disso, o receptor também envia informações sobre a qualidade da recepção, permitindo que ao transmissor adaptar a transmissão. A FIGURA 4 esquematiza o diagrama de blocos do processo de recepção. Dependendo da implementação algumas operações podem ser realizadas em uma ordem diferente, de acordo com a necessidade.

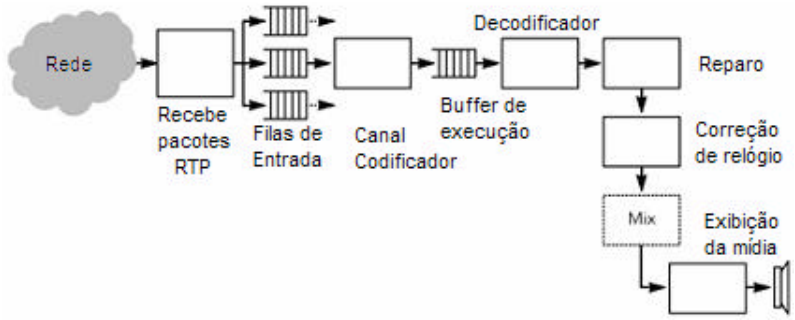

FIGURA 4. Diagrama de blocos de receptor RTP (Perkins, 2003)

O primeiro passo do processo é receber os pacotes da rede, fazer sua validação e correção, e inseri-lo em uma fila específica de entrada. Os pacotes são retirados destas filas e passados para um codificador opcional para correção de perda. Seguindo o codificador, os pacotes são inseridos em um buffer de execução específico. Este *buffer* é ordenado por *timestamp* (marca do tempo), e o processo de inserção de pacotes no *buffer* corrige qualquer desordenação possivelmente introduzida durante o transporte. Os pacotes permanecem no *buffer* de execução até que um *frame* completo seja recebido. O cálculo da quantidade de atraso adicionado é um dos aspectos mais críticos no projeto de uma implementação RTP. Cada pacote é rotulado com o tempo de execução desejado para o *frame* correspondente.

Os pacotes são agrupados para montar os *frames* e qualquer erro, ou falta de *frames*, é reparada. Depois dessa fase, os *frames* são decodificados (dependendo do *codec* usado pode ser necessário decodificar a mídia antes de reparar *frames* incompletos). Neste ponto, deve haver diferenças entre os relógios do transmissor e receptor. O receptor deve compensar essa diferença de relógio

para evitar espaçamentos na execução. Então, a mídia é finalmente exibida para o usuário.

Nas próximas subseções o protocolo RTP será detalhado, bem como suas particularidades para transporte de fluxos MPEG-2 e MPEG-4.

3.1.1.Sessões RTP

Uma sessão consiste de um grupo de participantes que se comunicam utilizando o mesmo serviço. Um participante pode estar ativo em múltiplas sessões RTP – por exemplo, uma sessão para troca de dados de áudio e outra sessão para troca de dados de vídeo. Para cada participante, a sessão é identificada pelo endereço de rede e um par de portas: uma de numeração par para pacotes de dados, e outra porta de numeração imediatamente superior (ímpar) para pacotes de controle do RTP. O par de portas padrão é 5004 e 5005 para UDP/IP, mas muitas aplicações alocam portas dinamicamente durante a configuração da sessão. As sessões RTP são projetadas para transportar um único tipo de mídia por vez, ou seja, em uma comunicação multimídia, cada tipo de mídia deve estar associado a uma sessão RTP individual.

A última revisão da especificação RTP relaxou o requerimento de que a porta de dados tenha número par, e além disso, permite que portas RTP e RTCP sejam não-adjacentes. Esta modificação tornou possível o uso do RTP em ambientes onde certos tipos de dispositivos de tradução de endereços de rede (*Network Address Translation – NAT*) estão presentes. Se possível, para que seja mantida a compatibilidade com implementações mais antigas, é recomendado utilizar portas adjacentes, mesmo isso não sendo estritamente necessário.

3.1.2. Detalhando o pacote RTP

O RTP é composto por duas partes: um protocolo de transferência de dados (*RTP Data Transfer Protocol*) e um protocolo de controle. O protocolo de transferência de dados gerencia a entrega de dados fim-a-fim em tempo real entre sistemas. Além de incorporar um número de seqüência para detecção de perda, *timestamp* para recuperação do relógio, tipo de carga útil, identificadores de fonte e um marcador para indicar eventos significantes dentro do fluxo de mídia, é definido

um nível adicional para molde à carga útil da mídia. Também são especificadas regras para uso do *timestamp* e número de seqüência, embora essas regras sejam dependentes do perfil e formato da carga útil utilizada.

Na FIGURA 5 é ilustrado o formato do pacote de transferência de dados do RTP. Pode-se dividir este pacote em quatro partes principais: o cabeçalho RTP, a extensão de cabeçalho (opcional), o cabeçalho da carga útil (opcional) e a carga útil propriamente dita. As duas primeiras partes serão exploradas nesta seção, enquanto que as duas últimas serão detalhadas em seções posteriores para carga útil de fluxos MPEG-2 e MPEG-4.

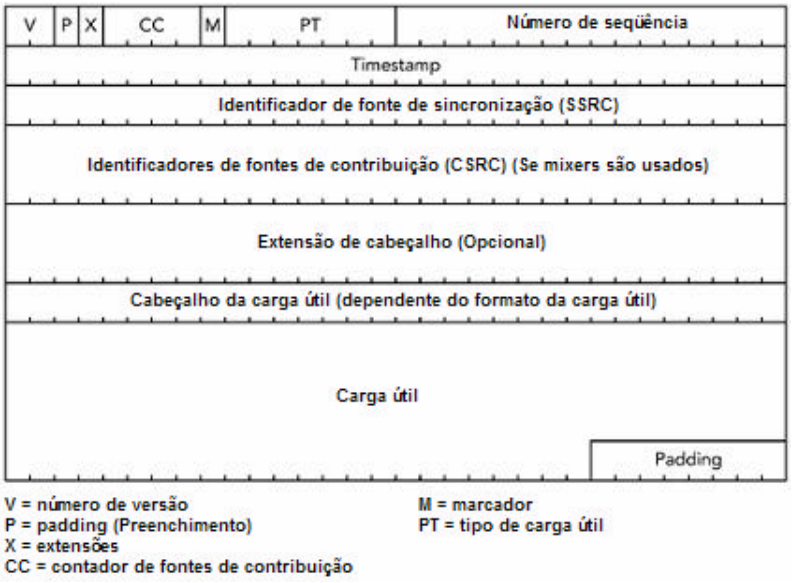

FIGURA 5. Pacote de transferência de dados do RTP (Perkins, 2003)

O cabeçalho do pacote de dados RTP tipicamente tem 12 octetos, embora possa conter uma lista de fontes de contribuição, na qual pode-se expandir o comprimento de quatro em quatro até 60 octetos. Os campos do cabeçalho são: tipo de carga útil, o número de seqüência, a marca temporal (*timestamp*) e um identificador de fonte de sincronização. Além disso, existe um contador de fontes de contribuição, um marcador para eventos, um indicador de preenchimento (*padding*), uma extensão de cabeçalho, e o número da versão. A seguir, os principais campos do cabeçalho RTP são descriminados:

Tipo de carga útil: o campo tipo de carga útil (PT) identifica a mídia transportada no pacote RTP. A aplicação receptora examina o tipo de carga útil para determinar como tratar os dados – por exemplo, passando-o para um determinado

descompressor. A exata interpretação deste campo é definida pelo perfil RTP, o qual associa o tipo de carga útil com as especificações de seu formato. O tamanho deste campo é 7 bits.

Número de seqüência: o número de seqüência é utilizado para identificar os pacotes e para prover ao receptor uma indicação se os pacotes estão sendo perdidos ou entregues fora de ordem. Ele não é usado para programar a exibição dos pacotes – o que é feito pelo *timestamp* – embora permita que o receptor reorganize a ordem em que os pacotes foram enviados. Este campo é um inteiro sem sinal de 16 bits, e aumenta de um em um a cada pacote de dados enviados, sendo que volta a zero quando o número máximo é extrapolado.

Timestamp: a marcação temporal (*timestamp*) denota o instante de amostragem do primeiro octeto dos dados de mídia em um pacote, e é usado para programar a exibição destes dados. Este é um campo de 32 bits sem sinal que aumenta a uma taxa dependente da mídia e volta para zero quando o valor máximo é excedido. O valor inicial do *timestamp* é escolhido aleatoriamente, ao invés de começar de zero. Como o número de seqüência, esta precaução visa aumentar a segurança de fluxos RTP. No contexto do MPEG, este campo é freqüentemente configurado com uma freqüência de 90 KHz.

Fonte de Sincronização: a fonte de sincronização (*Synchronization Source - SSRC*) identifica os participantes de uma sessão RTP. O SSRC é um inteiro de 32 bits, escolhido aleatoriamente pelos participantes no momento em que eles estabelecem uma sessão. Escolhido o identificador SSRC, o participante o utiliza no envio de seus pacotes. Pelo fato do SSRC ser escolhido localmente, dois participantes podem ter o mesmo valor. Esse conflito pode ser detectado quando uma aplicação recebe um pacote de outra que contem o mesmo identificador SSRC. Nesse caso, existe todo um processo de substituição de SSRC. Com isso é assegurado que cada participante terá um SSRC único.

Todos os pacotes com mesmo SSRC possuem o mesmo relógio e fazem parte do mesmo conjunto de número de seqüência. Desta forma, o receptor pode agrupar os pacotes por SSRC antes da exibição. Se um participante gera múltiplos fluxos em uma sessão RTP – por exemplo, de duas câmeras separadas – cada um

deve ser identificado por um SSRC diferente, e o receptor pode distinguir cada pacote pertencente a cada fluxo.

Fontes de Contribuição: em circunstâncias normais, os dados RTP são gerados por uma única fonte, mas pode acontecer de vários fluxos RTP passarem por um dispositivo de modo que várias fontes de dados podem contribuir em um pacote de dados. A lista de fontes de contribuição (*Contributing Sources – CSRCs*) identifica os participantes que contribuíram em um pacote RTP, não sendo contudo responsáveis por seu relógio e sincronização. Cada identificador de fonte de contribuição é formado por um inteiro de 32 bits, correspondendo ao SSRC do participante que contribuiu no pacote. O tamanho da lista CSRC é indicado pelo campo CC do cabeçalho RTP.

Marcador: o bit de marcação (M) é usado para indicar eventos de interesse dentro do fluxo de mídia. Seu significado preciso é definido pelo perfil RTP e tipo de mídia em uso.

Para fluxos de áudio operando sob o perfil RTP para conferências de áudio e vídeo com o mínimo de controle, ao marcador é atribuído um para indicar o primeiro pacote enviado após um período de silêncio, caso contrário é marcado como zero. Um bit de marcação igual a um, indica a aplicação que pode ser um bom momento para ajustar o ponto de exibição, pois uma pequena variação no tamanho de um período de silêncio não é usualmente perceptível para ouvintes (uma mudança no ponto de exibição enquanto o áudio está sendo exibido é audível).

Em todos os casos, o bit de marcação funciona como uma dica para a aplicação, a qual deve ser projetada para funcionar mesmo se pacotes com o bit de marcação são perdidos.

Padding: o bit de *padding* (P) é usado para indicar que à carga útil foi adicionado um preenchimento para completar o tamanho do pacote. Se o preenchimento é utilizado, o bit P é marcado e o último octeto da carga útil é preenchido com um contador do número de octetos de preenchimento utilizados.

Número de versão: nada pacote RTP possui um número de versão, indicado pelo campo V.

Extensão de cabeçalho: o RTP permite a extensão de cabeçalho, indicada pelo bit X igual a 1. A extensão é de tamanho variável, e começa com um campo de tipo de 16 bits seguido de outro campo de 16 bits, o qual contem o tamanho da extensão em octetos (excluindo os 32 bits iniciais).

3.1.3. RTCP

O protocolo de controle do RTP (*RTP Control Protocol - RTCP*) provê informações da qualidade de recepção e sincronização entre fluxos de mídia, dentre outras. A informação enviada pelo RTCP é necessária para sincronização, por exemplo, entre fluxos e pode ser útil para adaptar a transmissão de acordo com avaliação de qualidade de recepção. O RTCP funciona em conjunto com o RTP e periodicamente reporta essas informações. Apesar de os pacotes de dados serem enviados tipicamente em poucos milisegundos, o protocolo de controle opera na ordem de segundos.

Esses dois protocolos trabalham juntos, mas cada qual pode ser usado independentemente. Como o RTP não tem um mecanismo de controle sobre a conexão (transmissão e recepção), usa-se o RTCP para esta finalidade. Com esta especilialização, diminui-se o *overhead* que o uso de somente um único protocolo teria.

Cinco tipos de pacotes RTCP são definidos na especificação do RTP: relatório do receptor (RR), relatório do transmissor (SR), descrição da fonte (SDES), gerenciamento de membros (BYE), e definido pela aplicação (APP). Todos eles seguem uma estrutura comum – ilustrada na FIGURA 6 – embora o formato específico mude dependendo do tipo de pacote.

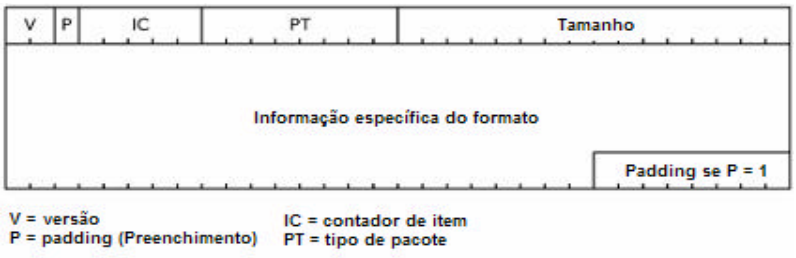

FIGURA 6. Formato básico do pacote RTCP (Perkins, 2003)

A seguir, os principais campos do cabeçalho RTP são descriminados:

Número de versão (V): campo de 2 bits. É sempre igual a 2 para a versão corrente do RTP.

Padding (P): campo de 1 bit que indica que o pacote foi preenchido com octetos adicionais para completar seu tamanho. Se este bit for igual a 1, um ou mais octetos de preenchimento foram adicionados ao fim do pacote, e o último octeto contém o número de octetos de preenchimento adicionados.

Contador de Item (IC): alguns tipos de pacotes contem uma lista de itens. O campo do contador de itens é usado por esses tipos de pacotes para indicar o número de itens incluídos no pacote (este campo tem diferentes nomes nos diferentes tipos de pacotes dependendo do seu uso).

Tipo do pacote (PT): campo que identifica o tipo do pacote, ou seja, o tipo de informação levada no pacote.

Tamanho: o campo Tamanho denota o comprimento do conteúdo do pacote a partir do fim do cabeçalho. Este campo é medido em unidades de palavras de 32 bits. O tamanho zero é um comprimento válido, indicando que o pacote consiste apenas dos 4 octetos do cabeçalho.

Na seqüência, são apresentados os 5 tipos de pacotes RTCP:

RTCP RR (Relatório do Receptor): um dos usos do RTCP é para relatar a qualidade de recepção, o que é feito através dos pacotes RTCP de relatório do receptor (RR) enviados por todos os participantes que recebem os dados.

O retorno sobre a qualidade de recepção através de pacotes RR é útil não só para o transmissor, como também para os outros participantes e ferramentas de monitoramento. Esse retorno pode permitir que o transmissor adapte suas transmissões de acordo com as informações reportadas pelo receptor. Além disso, com as informações provenientes de outros participantes pode-se determinar se o problema é local ou comum a todos os receptores.

RTCP SR (Relatório do Transmissor): além do relatório de qualidade de recepção, o RTCP fornece pacotes de relatório do transmissor (SR) que são enviados por participantes que estão enviando dados. Isso provê informação da mídia sendo enviada, sem que seja necessário receber os dados para obtê-la.

RTCP SDES (Descrição da Fonte): o RTCP pode ser utilizado para enviar pacotes de descrição da fonte (SDES) que provêem a identificação do participante e outros detalhes, como localização, e-mail e número de telefone.

RTCP BYE (Gerenciamento de membros): um pacote BYE é gerado quando um participante deixa a sessão, ou quando muda o SSRC – por exemplo, devido a um conflito.

RTCP APP (Pacotes definidos pela aplicação): Permite extensões definidas pela aplicação.

3.1.4. Carga útil dos pacotes RTP

No RTP são definidos formatos de carga útil para os diferentes tipos de mídia que são transportados. Os formatos da carga útil são referenciados pelo perfil RTP, e definem propriedades dos dados transportados.

O cabeçalho principal dos pacotes RTP provê informações que são comuns a todos os formatos de carga útil. Contudo, para cada formato de carga útil será necessária informação extra. Essa informação faz parte de um cabeçalho adicional que é definido como parte da especificação do formato de carga útil. O cabeçalho da carga útil é definido na especificação do formato da carga útil e está logo após a extensão do cabeçalho principal, como visto anteriormente.

Nas próximas subseções serão detalhadas as cargas úteis dos fluxos MPEG-2 e MPEG-4.

3.1.4.1. Carga útil para fluxo MPEG-2

Esta seção descreve o esquema de encapsulamento dos fluxos MPEG-2 de áudio, vídeo e *System* para transporte na carga útil de pacotes RTP.

3.1.4.1.1. Encapsulamento de MPEG *System* e MPEG TS

Cada pacote RTP contém um *timestamp* derivado do relógio de referência de 90KHz do transmissor. Esse relógio é sincronizado com uma referência de relógio (*Program Clock Reference – PCR* ou *System Clock Reference – SCR*) e representa o tempo de transmissão do primeiro byte da carga útil do pacote. O *timestamp* não é passado ao decodificador MPEG no receptor. Seu uso é um pouco diferente do que normalmente é no RTP, pois ele não é considerado o *timestamp* da apresentação. O propósito primário do *timestamp* RTP é estimar e reduzir qualquer variação estatística do retardo (*jitter*) introduzido pela rede e sincronizar a flutuação relativa do tempo entre o transmissor e o receptor.

Para fluxos MPEG-2 TS a carga útil do RTP conterá um número integral de pacotes de transporte MPEG-2. Para evitar ineficiência nos sistemas finais, dados provenientes de múltiplos pacotes MTS pequenos (normalmente fixado em 188 bytes) são agregados dentro de um único pacote RTP. O número de pacotes de transporte contidos é computado dividindo-se o tamanho da carga útil RTP pelo comprimento do pacote MTS (188).

Para fluxos de programas MPEG-2 não existem restrições de encapsulamento, ou seja, esses fluxos são tratados como um fluxo empacotado de bytes.

A utilização dos campos do cabeçalho RTP é realizada da seguinte maneira:

Tipo de carga útil (PT): um identificador diferente é utilizado para fluxo de transporte e fluxo de programa MPEG-2;

Marcador (M): igual a 1 sempre que o *timestamp* é descontínuo (o que deve acontecer quando um transmissor chaveia de uma fonte de dados para outra). Isso permite que o receptor que esteja sincronizando o fluxo, ignorar a diferença entre o *timestamp* atual e o *timestamp* antigo nos seus detectores da fase do relógio.

***Timestamp*:** campo de 32 bits com um *timestamp* de 90K Hz representando o momento de transmissão do primeiro byte do pacote.

3.1.4.1.2. Encapsulamento do vídeo MPEG-2

Considerando que quadros de MPEG podem ser grandes, eles regularmente serão fragmentados em pacotes. As regras de fragmentação se aplicam na seguinte ordem:

- o Video_Sequence_Header MPEG, quando presente, sempre estará no começo da carga útil do RTP;

- um GOP_header MPEG, quando presente, sempre estará no começo da carga útil do RTP, ou seguirá o Video_Sequence_Header;

- um Picture_Header MPEG, quando presente, estará sempre no começo da carga útil RTP, ou seguirá o GOP_Header.

Os campos do cabeçalho RTP são usados da seguinte forma:

Tipo de carga útil (PT): identificador do fluxo de vídeo;

Marcador (M): quando igual a 1, indica que o pacote contém o código de fim do *frame* MPEG.

***Timestamp*:** para pacotes que contêm somente a seqüência de vídeo e/ou o cabeçalho GOP, o *timestamp* é aquele do quadro subseqüente.

A FIGURA 7 apresenta o cabeçalho da carga útil para fluxo de vídeo MPEG. A seguir, são descriminados seus campos.

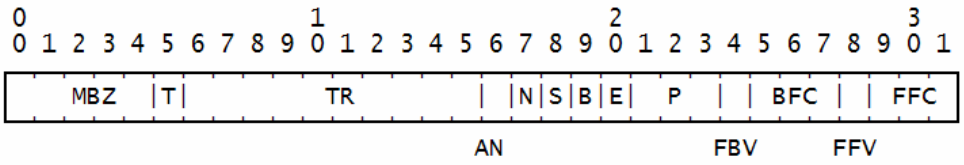

FIGURA 7. Cabeçalho da carga útil de vídeo MPEG-2

MBZ: campo de 5 bits reservado para uso futuro. Na atual especificação deve ser configurado com o valor zero.

T: campo de 1 bit que quando configurado com o valor 1, indica que o cabeçalho da carga útil (vídeo MPEG) é estendido.

TR: referência temporal (10 bits). Indica a referência temporal do quadro atual.

AN: usado para sinalizar mudanças na informação de cabeçalho de quadro para cargas úteis MPEG-2. Deve ser fixado a 0 quando não é usado.

N: cabeçalho de novo quadro (1 bit). Usado para cargas MPEG-2 quando o bit anterior (AN) é igual a 1. Caso contrário, deve ser igual a 0. Este bit é igual a 1 quando a informação contida nos cabeçalhos de quadros anteriores não podem ser usados para reconstruir um cabeçalho do quadro atual. Isso acontece quando o quadro atual é codificado usando um conjunto diferente de parâmetros.

S: presença de cabeçalho de seqüência (1 bit). Utilizado para detectar a presença do cabeçalho de seqüência no pacote RTP.

B: início de trecho (*Beginning-of-Slice – BS*). Campo de 1 bit que é igual a 1 quando o pacote de carga útil é a parte inicial do trecho MPEG.

E: fim de trecho (*End-of-slice – ES*). Campo de 1 bit que é igual a 1 quando o último byte da carga útil é o fim do trecho MPEG.

P: tipo do quadro (3 bits). Este valor é constante para cada pacote RTP de um dado quadro. Quadros do tipo I, P, B e D são representados pelos valores 1, 2, 3, 4 respectivamente. O valor 000B é proibido e 101B – 111B são reservados para dar suporte a futuras extensões da especificação MPEG.

FBV: full_pel_backward_vector

BFC: backward_f_code

FFV: full_pel_forward_vector

FFC: forward_f_code

3.1.4.1.3. Encapsulemento do áudio MPEG-2

Vários *frames* de áudio podem ser encapsulados dentro de um pacote RTP. Nesse caso, o número de *frames* deve estar contido dentro do pacote e o cabeçalho de fragmentação deve ser definido como 0.

Da mesma forma que pacotes relativamente pequenos são usados, um *frame* pode ser grande suficiente de modo a precisar ser fragmentado em vários

pacotes. Nesse caso, o cabeçalho indicador de fragmentação presente para o tipo de carga útil de áudio MPEG-2 deve ser igual a 1.

Os campos do cabeçalho RTP são usados da seguinte forma:

Tipo de carga útil (PT): identificador do fluxo de áudio;

Marcador (M): quando igual a 1 indica "surto de voz", caso contrário, é igual a 0;

***Timestamp*:** representa o tempo de apresentação de um *frame* de áudio.

Na FIGURA 8, é apresentado o cabeçalho da carga útil específico para áudio MPEG-2. Este cabeçalho deve ser anexado a cada pacote RTP entre o cabeçalho do pacote e os dados da carga útil.

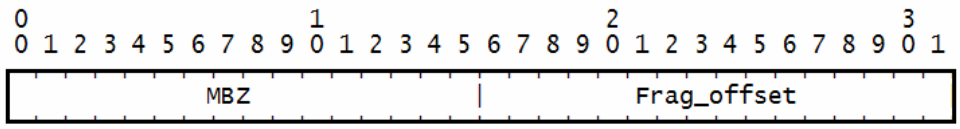

FIGURA 8. Cabeçalho da carga útil de áudio MPEG-2

Frag_offset: *offset* dentro do *frame* de áudio para os dados neste pacote.

3.1.4.2. Carga útil para fluxo MPEG-4

Esta seção descreve o formato da carga útil do protocolo RTP para transporte de fluxos de áudio e vídeo MPEG-4 sem usar o MPEG-4 *Systems*. Esses formatos de carga útil serão utilizados em sistemas que possuem intrinsecamente a funcionalidade de gerenciamento de fluxos, não requerendo tal funcionalidade do MPEG-4 *Systems*.

Uma vantagem óbvia de se utilizar esses formatos de carga útil de áudio e vídeo é que eles podem ser manipulados de forma unificada juntamente com formatos definidos para codificadores que não são MPEG-4. Uma desvantagem é que a interoperabilidade com ambientes que usam MPEG-4 *Systems* pode ser difícil.

3.1.4.2.1. Encapsulamento do MPEG-4 visual

Um fluxo de bits visuais MPEG-4 é mapeado diretamente em pacotes RTP sem a adição de campos de cabeçalho extra ou remoção de qualquer elemento de sintaxe

visual. O modo de fluxo de configuração/elementar combinado deve ser usado de forma que a informação de configuração será levada para a mesma porta RTP como o fluxo elementar.

A utilização dos campos do cabeçalho RTP se dá da seguinte forma:

Tipo de carga útil (PT): cada perfil RTP para uma classe específica de aplicações terá um identificador distinto.

Extensão (X): definido no perfil RTP usado.

Número de seqüência: incrementado de um em um para cada pacote RTP enviado. Como mencionado anteriormente, por questões de segurança, o valor inicial é escolhido randomicamente.

Marcador (M): quando igual a 1, indica o último pacote de um VOP (*Vídeo Object Plane*). Quando vários VOPs são carregados no mesmo pacote, ao marcador deve ser atribuído o valor 1.

***Timestamp*:** indica o rótulo de amostragem do VOP contido no pacote RTP. Sua freqüência padrão é 90 kHz. Além disso, pode-se destacar as seguintes características:

- A regra de fragmentação recomenda não mapear mais de um VOP por pacote RTP, pois dessa forma o *timestamp* indicará exclusivamente o tempo do VOP contido no pacote. Por outro lado, um vídeo MPEG-4 pode gerar VOPs muito pequenos. Nesse caso, para reduzir o overhead, a fragmentação permite a concatenação de múltiplos VOPs em um pacote RTP. Nesse caso, o *timestamp* indica o menor dos tempos entre os VOPs. A informação do *timestamp* dos outros VOPs é derivada dos campos de *timestamp* no cabeçalho do VOP (modulo_time_base e vop_time_increment);

- Se o pacote RTP contém somente informação de configuração e/ou campos de Group_of_VideoObjectPlane(), o *timestamp* do próximo VOP na ordem de codificação é usado;

- Se o pacote RTP contém somente informação visual_object_sequence_end_code, o timestamp do VOP imediatamente precedente na ordem de codificação é usado.

Um fluxo de bits visuais MPEG-4 fragmentado é mapeado diretamente em pacotes RTP sem a adição de campos de cabeçalho extra ou remoção de qualquer elemento de sintaxe visual. O modo de fluxos de Configuração/Elementar combinados é usado. O cabeçalho é um dos seguintes:

- Informação de configuração (Cabeçalho de seqüência de objetos visuais, cabeçalho de objeto visual e cabeçalho de camada de objeto de vídeo);

- visual_object_sequence_end_code;

- o cabeçalho da função de ponto de entrada para um fluxo elementar (Group_of_VideoObjectPlane() ou cabeçalho de VideoObjectPlane(), video_plane_with_short_header(), MeshObject() ou FaceObject());

- cabeçalho do pacote de vídeo (video_packet_header() excluindo next_resync_marker());

- cabeçalho de gob_layer().

3.1.4.2.2. Encapsulamento do áudio MPEG-4

O áudio MPEG-4 é um novo tipo de padrão que integra diferentes tipos de ferramentas de codificação de áudio. Fluxos de áudio MPEG-4 devem ser formatados pela ferramenta LATM (*Low-overhead MPEG-4 Audio Transport Multiplex*). Fluxos baseados nessa ferramenta consistem de uma seqüência de *audioMuxElements* que incluem um ou mais *frames* de áudio. Um *audioMuxElement* completo ou parte de um deve ser mapeado diretamente dentro da carga útil do RTP sem remover nenhum elemento de sintaxe do *audioMuxElement*. O primeiro byte de cada *audioMuxElement* deve estar localizado na primeira alocação da carga útil do pacote RTP.

A utilização dos campos do cabeçalho RTP se dá da seguinte forma:

Tipo de carga útil (PT): cada perfil RTP para uma classe específica de aplicações terá um identificador distinto.

Marcador (M): o bit marcador indica os limites do *audioMuxElement*. Quando seu valor é igual a 1, está sendo indicado que o pacote RTP contém um *audioMuxElement* completo ou último fragmento de um *audioMuxElement.*

***Timestamp*:** indica o rótulo de amostragem do primeiro *frame* de áudio contido no pacote RTP. Sua freqüência padrão é 90 kHz.

Número de seqüência: incrementado de um em um para cada pacote RTP enviado. Como mencionado anteriormente, por questões de segurança, o valor inicial é escolhido randomicamente.

É recomendado colocar um *audioMuxElement* dentro de cada pacote RTP. Se o tamanho do *audioMuxElement* exceder o tamanho da carga útil do pacote RTO, ele pode ser fragmentado e espalhado em vários pacotes.

Para transmissão de fluxos escaláveis, dados de áudio de cada camada devem ser transportados em diferentes pacotes, permitindo assim que camadas diferentes possam ser tratadas distintamente no nível IP. Toda configuração de dados de fluxos escaláveis está contida em um *StreamMuxConfig* do LATM e todas camadas escaláveis dividem o mesmo *StreamMuxConfig*. O mapeamento entre cada camada e sua configuração é obtida pela informação contida no cabeçalho do LATM nos dados de áudio.

3.2. RTSP

O *Real Time Streaming Protocol*, ou RTSP, foi desenvolvido através de uma parceria da RealNetworks, Netscape Communications e Universidade de Columbia no grupo de trabalho *Multiparty Music* (MMUSIC) do IETF (RFC 2326, 1998). O RTSP é um protocolo cliente-servidor de controle de apresentação multimídia, projetado para tratar fluxos multimídia sobre redes IP.

Os fluxos controlados pelo RTSP geralmente usam o RTP, mas o seu funcionamento não depende deste mecanismo de transporte utilizado para a entrega de mídia contínua. A intenção do protocolo RTSP é controlar múltiplas sessões de entrega de dados, e prover um meio para escolha dos canais de entrega tais como UDP, UDP *multicast* e TCP, além do RTP. Embora apareça em seu nome, o RTSP só provê garantia de entrega em tempo real quando faz uso do protocolo RTP.

O RTSP foi projetado para trabalhar com mídias baseadas no tempo, como fluxos de áudio e vídeo MPEG. Na verdade, a função deste protocolo não é a entrega de fluxos, como é feito no RTP, mas sim dar suporte ao controle da

apresentação, tendo como principal contexto os sistemas de vídeo sob demanda (VOD). Em outras palavras, o RTSP atua como um "controle remoto" sobre os servidores de mídia, possibilitando aos clientes executarem operações como play, pause, etc. (controle VCR).

Ao contrário do HTTP (*Hyper Text Transport Protocol*), o RTSP permite que o servidor de mídia envie requisições para o cliente (conexões persistentes). Na FIGURA 9 é onde o RTSP está na pilha de protocolos da arquitetura IP.

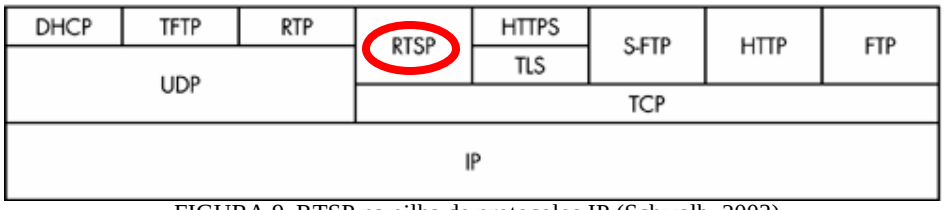

FIGURA 9. RTSP na pilha de protocolos IP (Schwalb, 2003)

A visão geral da apresentação e as propriedades das mídias que a compõem – tais como suas codificações, idiomas, e outros parâmetros que possibilitam ao cliente a escolha da combinação que lhe é mais apropriada – são definidas em um arquivo de descrição de apresentação. Esse arquivo não necessariamente está armazenado no servidor de mídia, e pode ser obtido utilizando-se HTTP ou outro meio como, por exemplo, e-mail.

Nesse arquivo, cada fluxo de mídia que é controlado individualmente pelo RTSP possui uma URL RTSP, a qual faz referência a um fluxo armazenado em um servidor de mídia. Vários fluxos podem estar localizados em diferentes servidores como, por exemplo, fluxos de áudio e vídeo com o intuito de balanceamento de carga.

Cada apresentação e fluxo de mídia podem ser identificados por uma URL RTSP que possui o seguinte formato:

rtsp_URL = ("rtsp:" | "rtspu:") "//" host [":" port] [abs_path]

onde,

host = <endereço IP válido>

*port = <*DÍGITO> (Se a porta não for especificada, é considerada a porta 554)*

abs_path = <caminho no host onde o recurso se encontra>

Um exemplo de URL RTSP é:

rtsp://media.example.com:554/twister/audiotrack

que identifica um fluxo de áudio dentro da apresentação “twister”, o qual pode ser controlado via requisições RTSP sobre uma conexão TCP na porta 554 do host media.example.com.

Não existe a noção de conexão RTSP. Ao invés disso, um servidor mantém uma sessão que é rotulada por um identificador. Uma sessão RTSP não é amarrada a uma conexão do nível de transporte, como por exemplo uma conexão TCP. Durante uma sessão RTSP, um cliente pode abrir e fechar várias conexões de transporte com o servidor para realizar requisições RTSP. Alternativamente, ele também pode usar um protocolo de transporte não-orientado a conexão assim como o UDP.

Uma sessão tipicamente consiste em um cliente utilizar um mecanismo de transporte para recebimento de fluxo de mídia contínuo, iniciar e finalmente encerrar o fluxo.

Os dois principais tipos de mensagem do protocolo RTSP são requisições (*Request*) e respostas (*Response*). Requisições contêm métodos, e é através destes que é possível modificar o estado em um servidor de mídia.

Muitos métodos no RTSP não modificam o estado. Entretanto, os métodos apresentados a seguir desempenham um papel fundamental para a alocação e uso de recursos no servidor. São eles:

- *Setup*: faz o servidor alocar recursos para um fluxo e inicia uma sessão RTSP;
- *Play* e *Record*: inicia a transmissão de dados de um fluxo alocado via *SETUP*;
- *Pause*: suspende um fluxo temporariamente sem liberar os recursos previamente alocados no servidor;
- *Teardown*: libera recursos associados a um fluxo. A sessão RTSP é encerrada.

As transições de estado são descritas na TABELA 1.

TABELA 1. Transição de estado das sessões RTSP

State	Message Sent	State after Response
Init	Setup	Ready
	Teardown	Init
	Play	Playing
Ready	Record	Recording
	Teardown	Init
	Setup	Ready
Playing	Pause	Ready
	Teardown	Init
	Play	Playing
	Setup	Playing (changed transport)
Recording	Pause	Ready
	Teardown	Init
	Record	Recording
	Setup	Recording (changed transport)

Esses comandos necessitam ser disparados utilizando-se um protocolo confiável (que garante a entrega) como o TCP.

Tanto as mensagens de requisição quanto às de resposta carregam o identificador da sessão RTSP que é recebido pelo cliente como resposta da requisição e iniciação da sessão. O identificador de sessão é gerado randomicamente pelo servidor de mídia e deve ter pelo menos 8 octetos por questões de segurança. Uma vez que o cliente possui este identificador, ele deve usá-lo em cada requisição relacionada à sessão.

Como se pode notar, um identificador identifica uma sessão através das sessões de transporte ou conexões. Desta forma, mensagens de controle para mais de uma URL RTSP podem ser enviadas dentro de uma única sessão RTSP. Da mesma forma, é possível que clientes usem a mesma sessão para controlar vários fluxos que constituem uma apresentação. Além disso, o identificador de sessão é necessário para distinguir entre requisições de entrega de uma mesma URL provenientes do mesmo cliente.

4 Transporte de datagramas IP através de redes MPEG

A possibilidade de encapsulamento de dados para a difusão em conjunto com o áudio/vídeo abre diversas alternativas para a provisão de serviços avançados. No contexto de TV digital, as emissoras poderão não somente disponibilizar uma programação de alta qualidade de imagem e som, mas também torná-la mais atraente, permitindo aos usuários interagirem com os programas sendo assistidos. Cabe ao sistema de TV digital, portanto, prover meios para que esses tipos de funcionalidades sejam oferecidos, definindo padrões de codificação, recuperação, sincronização e tratamento dos dados difundidos. Nas próximas seções serão abordados os mecanismos de encapsulamento de datagramas IP em redes MPEG.

4.1. Transporte de datagramas IP sobre MPEG-2

O MPEG-2 TS tem sido amplamente aceito não somente por prover serviços de TV digital, mas também como uma tecnologia para construção de redes IP. Exemplos incluem os padrões *Digital Video Broadcast* (DVB) e o *Advanced Television Systems Committe* (ATSC) para TV digital. Um dos principais benefícios da utilização do Fluxo de Transporte do MPEG-2 é que o mesmo pode transportar, além de fluxos de vídeo e áudio, fluxos de dados.

Os dados podem ser transportados através de serviços assíncronos, síncronos ou sincronizados. Dados assíncronos implicam que nenhuma marca de tempo (*timestamp*) está associada aos dados. Informações síncronas assumem que marcas de tempo estão associadas com os dados, mas eles não estão relacionados à temporização de outros fluxos de áudio ou vídeo. Finalmente, os dados são sincronizados quando estão associados a marcas de tempo que fazem referência a instantes particulares de fluxos de áudio e/ou vídeo.

Os mecanismos de encapsulamento e transporte de fluxos de dados variam de acordo com a aplicação a que se destinam, ou seja, conforme a natureza da informação e seus respectivos requisitos, principalmente relativos a atraso e sincronização.

Assim, faz-se necessário um conjunto de padrões definindo uma interface entre o MPEG-2 TS e uma rede IP. Isso sugere um novo método de encapsulamento para datagramas IP, e pressupõe a utilização de protocolos para associar pacotes IP com propriedades de canais lógicos providos por um MPEG-2 TS.

A especificação do DVB para difusão de dados define três diferentes maneiras de transportar dados em MPEG-2 TS. São elas:

(i) *Data Streaming*: pacotes de dados podem ser encapsulados e transportados dentro de pacotes PES.

(ii) *Multiprotocol Encapsulation* (MPE): pacotes de dados podem ser transportados dentro dos pacotes de seção definidos pelas tabelas internas de sistema no DSM-CC (*Digital Storage Media Command and Control*);

(iii) *Data Piping*: um protocolo da camada de adaptação segmenta pacotes de dados diretamente em uma seqüência de células.

A FIGURA 10 exibe os possíveis pontos de entrada para transporte de dados sobre MPEG-2.

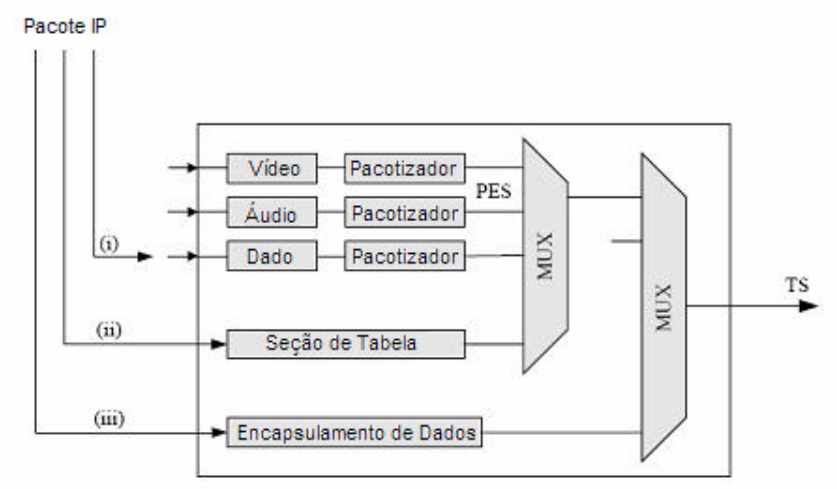

FIGURA 10. Transporte de dados sobre MPEG (Collini-Nocker, 2004)

As subseções a seguir detalham cada uma das formas de transporte de dados no MPEG-2.

4.1.1. Data Streaming

A subcamada PES é responsável pela inserção de informações de sincronização entre partes de um fluxo e entre diferentes fluxos e pelo controle de *buffers*. Os fluxos elementares de áudio e vídeo, previamente codificados na camada de compressão, são transportados como carga útil dos pacotes definidos pela subcamada PES. Cada pacote PES contém uma ou mais unidades de apresentação de cada fluxo elementar (quadro ou figura, no caso de vídeo, uma ou mais amostras, no caso de áudio, por exemplo). Os pacotes PES, por sua vez, são multiplexados nos pacotes definidos pela subcamada TS.

O transporte de dados através de pacotes PES pode ser realizado de duas formas: os dados podem ser inseridos no campo PES_private_data, que possui, no máximo, 16 bytes, ou podem ser inseridos como a carga útil do pacote PES. No primeiro caso, os dados privados acompanham a carga útil do pacote PES. No segundo caso, a carga útil é constituída pelos dados privados que podem ser síncronos ou assíncronos. Os descritores e as seções privadas são fluxos criados pelos usuários, que determinam seu significado.

4.1.2. MPE

Outra forma de transporte de dados é através de seções do DSM-CC (*Digital Storage Media Command and Control*), padrão ISO/IEC 13818-6 (ISO/IEC TR 13818-6, 1998). Nessa alternativa, os dados são encapsulados através do protocolo *Data Download Protocol* ou de seções endereçáveis do DSM-CC, cujas mensagens DSM-CC são transportadas através de pacotes PES.

As seções endereçáveis do DSM-CC podem ser usadas para encapsular dados assíncronos, tais como datagramas IP. O destino pode ser um equipamento específico ou um grupo de equipamentos. Uma seção endereçável pode conter um campo de *checksum* para proteção contra erros.

O dados encapsulados através de seções endereçáveis do DSM-CC são transmitidos através de fluxos elementares de dados do Fluxo de Transporte do MPEG-2. Os dados assíncronos (datagramas) podem ser encapsulados diretamente em uma seção endereçável ou podem ser encapsulados no protocolo LLC (*Logical Link Control*) e, adicionalmente, utilizar o protocolo SNAP (*Sub*

Network Access Protocol) para indicar o protocolo sendo transportado. O resultado, então, é encapsulado em uma seção endereçável. Para o protocolo IP, o encapsulamento LLC pode ser dispensado, sendo os datagramas encapsulados diretamente nas seções endereçáveis.

O MPE (*Multiprotocol Encapsulation*) foi inicialmente proposto em 1996 e desde então se tornou amplamente aceito e utilizado como padrão para transporte de datagramas IP sobre DVB. A especificação do MPE do DVB (também suportado pelo ATSC) usa seções privadas para o transporte de datagramas IP. Os pacotes de dados são encapsulados em datagram_sections, as quais são complacentes com o formato da seção DSM-CC para dados privados e esse encapsulamento faz uso de um endereço de nível MAC (*Medium Access Control*).

Na FIGURA 11 é exibido o cabeçalho do pacote MPE para o IPv4.

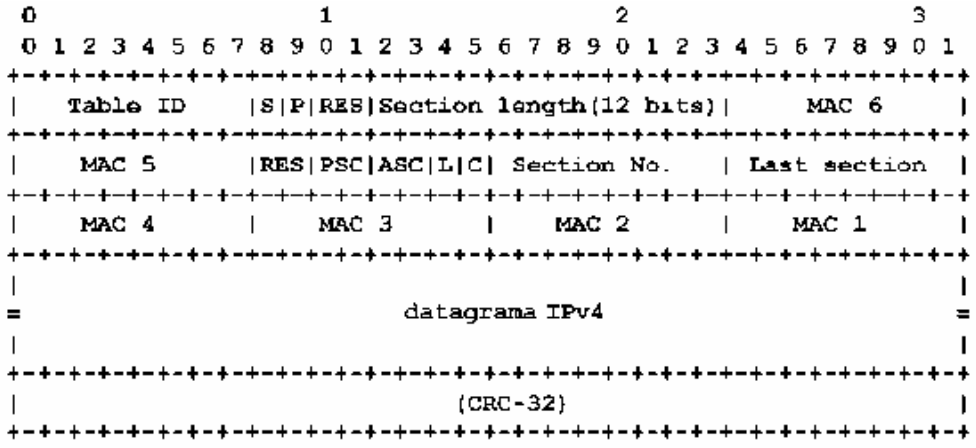

FIGURA 11. Cabeçalho MPE (Collini-Nocker, 2004)

O MPE usa semânticas de seção de tabela (*table section*). Para indicar a posição do início de uma seção, o campo Pointer (1 byte), é inserido em todos os pacotes TS onde o bit de PUSI (payload_unit_start_indicator) do pacote TS é igual a 1. Isso especifica o *offset* para o início do pacote de seção (na FIGURA 11 o campo Pointer não é exibido). Além do campo de comprimento (*length field*), do endereço MAC do receptor para endereçamento e filtragem, o MPE oferece vários outros campos (o campo Table ID, por exemplo, é configurado com o valor 0x3E para indicar o uso da seção DSM-CC com dado privado).

O cabeçalho do encapsulamento MPE não possui explicitamente um campo de tipo (por exemplo, para diferenciar entre IPv4 e IPv6). Se tal distinção é requerida, uma opção deve ser especificada no cabeçalho de encapsulamento MPE

para indicar a adição de um cabeçalho LLC, e esse deve ser seguido por um cabeçalho SNAP. Como se pode notar, isso tudo introduz um *overhead* adicional.

Atualmente, o MPE é o esquema de encapsulamento mais utilizado. Embora cuidadosamente projetado, o MPE não é ideal para o transporte de dados IP, primeiramente em termos de *overhead*, e depois pela dificuldade de uso para a maioria dos serviços IP atrativos comercialmente via *broadcast.* Além disso, a utilização do IPv6 com MPE demanda overhead adicional e processamento no receptor.

O desenvolvimento de um novo tipo de encapsulamento, o ULE (*Ultra Light Encapsulation*), tem se mostrado mais leve e flexível, possibilitando o aumento da eficiência e vazão. Embora seja esperado que o MPE continue sendo usado por muitos anos, acredita-se que o ULE possa e deva ser largamente empregado, mais especificamente para *IP-multicast* e transporte com IPv6 (Collini-Nocker, 2004). O ULE será o foco da próxima seção.

4.1.3. Data Piping

O mecanismo de encapsulamento ULE (*Ultra Light Encapsulation*) permite a inserção de dados diretamente no cabeçalho do pacote TS. Em outras palavras, o campo de adaptação do pacote TS provê um meio para transportar dados diretamente sobre o Fluxo de Transporte, sem a utilização de seções particulares ou de encapsulamento em pacotes PES. As estruturas de *metadados* também permitem que sejam inseridos dados dos usuários.

As metas de qualquer encapsulameto para transmissão de pacotes IP sobre um enlace DVB são eficiência, em termos de *overhead* e processamento, e flexibilidade de forma que protocolos de camadas superiores sejam suportados. Por conseguinte, o cabeçalho de ULE mínimo é mais curto, e tem menos campos que o cabeçalho do MPE, como mostrado na FIGURA 12.

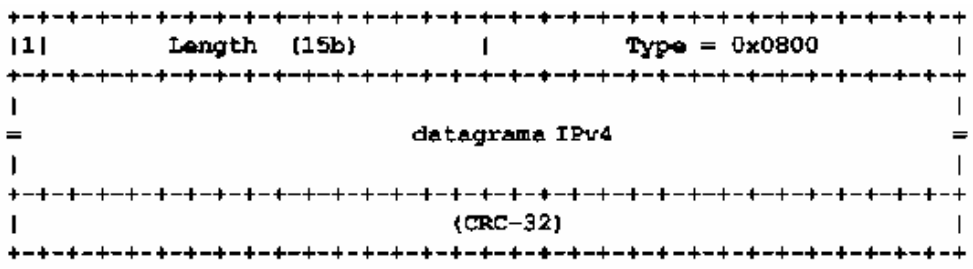

FIGURA 12. Cabeçalho ULE mínimo (Collini-Nocker, 2004)

No ULE, a SNDU (*Sub-Network Data Unit*) é usada para descrever o pacote de IP encapsulado. A diferença de 8 bytes por SNDU entre o cabeçalho de MPE mínimo (12 bytes) e o cabeçalho de ULE mínimo (4 bytes) é significante para SNDUs pequenas, como por exemplo, as de reconhecimento do TCP.

O número de campos de cabeçalho é um fator que está ligado ao número de instruções para processar um pacote. O MPE tem 18 campos (mais alguns quando LLC/SNAP é usado). A presença obrigatória de endereço MAC no MPE foi evitada no ULE para melhorar eficiência de transmissão de *IP-multicast* e *unicast* em alguns cenários (por exemplo, usuários finais diretamente conectados). Quando o endereçamento do receptor é requerido, o ULE adiciona 6 bytes de endereço NPA (*Network Point of Attachment*), análogo ao endereço MAC no MPE. A presença desse campo é indicada pelo bit D (o primeiro bit no cabeçalho do ULE). No ULE o número de campos é reduzido então a 3 (ou 4 quando o endereço de NPA está presente). Conseqüentemente, o menor número de campos melhora a eficiência de processamento.

Diferentemente do MPE, no que necessitava utilizar cabeçalhos adicionais (LLC/SNAP) para especificar o tipo da SNDU, o ULE fornece explicitamente um campo de tipo (*Type field*). O campo de tipo segue a semântica do cabeçalho Next do IPv6 para permitir encadeamento de cabeçalho.

O ULE oferece um campo de 15 bits de comprimento que limita o tamanho de uma SNDU em 32KB. SNDUs grandes precisam ser fragmentadas em pacotes TS, e geralmente não preenchem completamente a carga útil dos pacotes. A parte não utilizada da carga útil pode ser ignorada (*padding*), mas o MPE (e tabelas de seção em geral) pode suportar a pacotização pelo uso do campo Pointer (*Start Pointer*). O ULE preserva essas semânticas na pacotização. A FIGURA 13 ilustra as formas de encapsulamento dos pacotes no ULE. Em contraste com o MPE, o ULE descreve o procedimento de pacotização explicitamente e discute o uso de um Packing Threshold *timeout* que entra em ação quando o encapsulador não tem nenhuma SNDU adicional para processar, ou seja, ao fim de um período, se o encapsulador ULE não receber nenhum dado novo, o resto do pacote TS é completado com bytes de enchimento (*padding*), assegurando transmissão oportuna.

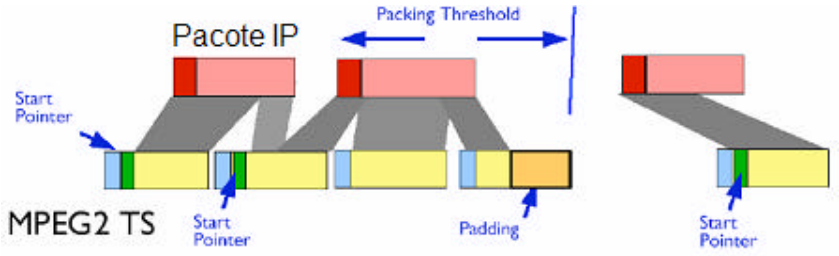

FIGURA 13. ULE com e sem empacotamento (Collini-Nocker, 2004)

Um resumo das principais diferenças do ULE comparado ao MPE são: suporte a um campo de tipo (*Type field*) que provê códigos para IPv4, IPv6, MPLS, etc; suporte a endereço fonte e de destino (ambos opcionais); projeto simples e não ambíguo para uma implementação enxuta; nenhuma "funcionalidade escondida", o que proporciona uma melhor interoperabilidade; melhoramento na eficiência (menor *overhead* e menos instruções).

5 Conclusões

A evolução das redes de comunicação tem possibilitado a implementação de novos serviços que foram originalmente desenvolvidos para outros tipos de redes. Atualmente, nota-se que ao invés substituição de redes de serviço específico como ocorreu no passado, tem-se optado pela integração de redes diferentes, advento esse chamado de convergência de redes, ou, convergência tecnológica. Como resultado dessa integração pode-se destacar a economia em infraestrutura, desenvolvimento, suporte e custos de gerenciamento, além é claro de uma maior flexibilidade na provisão de serviços.

Na Internet, com o aumento do poder de processamento dos computadores e crescente demanda por serviços que possibilitem o transporte de dados de mídias contínuas, foram desenvolvidos mecanismos capazes de transmitir vídeo e áudio em tempo real, recuperar o sincronismo e fazer a detecção de erros. Um exemplo disso é o RTP, o qual provê funções de transporte fim-a-fim para que aplicações possam transmitir dados em tempo real. Como o próprio nome diz, o RTP provê garantias de entrega em tempo real.

Nas redes MPEG, nota-se que a evolução das técnicas de codificação digital de áudio e vídeo, aliada aos novos esquemas eficientes de modulação para transmissões digitais, tornaram possível o advento da TV digital e com ela uma vasta gama de novos serviços. A possibilidade de encapsulamento de dados para a difusão em conjunto com o áudio/vídeo tem aberto diversas alternativas para a provisão de serviços avançados. Desta forma, as emissoras poderão não somente disponibilizar uma programação de alta qualidade de imagem e som, mas também torná-la mais atraente, permitindo aos usuários interagirem com os programas sendo assistidos. Nesse cenário, foram apresentados três esquemas de encapsulamento de datagramas IP em redes MPEG-2. Atualmente, o MPE é o esquema de encapsulamento mais utilizado. Entretanto, o MPE não é ideal para o transporte de dados IP (grande *overhead*). O desenvolvimento do encapsulamento o ULE tem se mostrado mais leve e flexível, possibilitando o aumento da eficiência. Embora seja esperado que o MPE continue sendo usado por muitos

anos devido a grande quantidade de terminais de acesso que o tem implementado, acredita-se que o ULE possa e deva ser largamente empregado.

As principais dificuldades encontradas ficaram por conta de entender as RFCs que descrevem os mecanismos de encapsulamento apresentados. Além disso, não foi encontrada uma referência que apresentasse de forma satisfatória como é realizado o encapsulamento de datagramas IP sobre MPEG-4.

6 Referências Bibliográficas

(Collini-Nocker, 2004) Collini-Nocker, B., Fairhurst, G., "**ULE versus MPE as an IP over DVB encapsulation"**, Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs '04**)**, Ilkley, U.K., 2004. http://www.comp.brad.ac.uk/het-net/HET-NETs04/CameraPapers/P70.pdf

(ISO/IEC 13818-1, 2000) ISO/IEC 13818-1. "**Information technology -- Generic coding of moving pictures and associated audio information: Systems",** 2000.

(ISO/IEC TR 13818-6, 1998) ISO/IEC TR 13818-6. "**Information technology -- Generic coding of moving pictures and associated audio information -- Part 6: Extensions for DSM-CC",** 1998.

(ISO/IEC 14496-1, 2001) ISO/IEC 14496-1. "**Information technology -- Coding of audio-visual objects – Part 1: Systems"**, 2001.

(ISO/IEC 14496-6, 2000) ISO/IEC 14496-6. "**Information technology -- Coding of audio-visual objects – Part 6: Delivery Multimedia Integration Framework (DMIF)"**, 2000.

(Koenen, 2002) KOENEN R., "**Overwiew of the MPEG-4 Standard - ISO/IEC JTC1/SC29/WG11 N4688"**, Março de 2002. Disponível em *http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm.* Acesso em 30 out. 04.

(Perkins, 2003) Perkings, C., **"RTP: Audio and Video for the Internet"**, Addison Wesley, 2003, ISBN: 0-672-32249-8

(RFC 1889, 1996) Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.. **"RTP: A Transport Protocol for Real-Time Applications,"**, RFC 1889, 1996, IETF http://www.rfc-editor.org/rfc/rfc1889.txt

(RFC 2326, 1998) Schulzrinne, H., Rao, A., Lanphier, R., "**Real-Time Streaming Protocol (RTSP)"**, RFC 2326, 1998, IETF http://www.rfc-editor.org/rfc/rfc2326.txt

(Schwalb, 2003) Schwalb, E., **"iTV Handbook: Technologies and Standards"**, Prentice Hall PTR, 2003, ISBN: 0-13-100312-7

(Soares et al., 2004) Soares, L. F. G., Colcher, S., Rodrigues, R., **"Relatório TV Digital: Identificação dos Cenários Tecnológicos de Interesse"**, Laboratório Telemídia, PUC-Rio, 2004.