

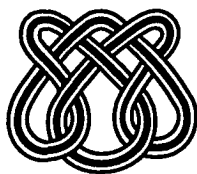
UNIVERSIDADE DE SÃO PAULO

**Multimídia sobre IP
Protocolos e Padrões**

**Renato de Freitas Bulcão Neto
Maria da Graça Campos Pimentel**

Nº 51

NOTAS DIDÁTICAS



Instituto de Ciências Matemáticas de São Carlos

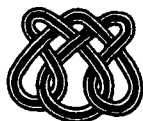
UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2585

Multimídia sobre IP
Protocolos e Padrões

Renato de Freitas Bulcão Neto
Maria da Graça Campos Pimentel

Nº 51

NOTAS
DIDÁTICAS



São Carlos
Set./2001

Índice

1. Introdução.....	1
2. Rede Multimídia: Metas e Desafios.....	1
3. Qualidade de Serviço - QoS.....	3
4. IP Multicast.....	4
4.1 Taxonomia de Aplicações IP Multicast	6
4.2 Requisitos para Multicasting.....	6
5. RSVP – Protocolo de Reserva de Recursos	7
5.1 Características do RSVP	8
5.2 Fluxos de Dados do RSVP	9
5.3 Mecanismos do Protocolo RSVP	10
6. RTP – Protocolo de Transporte de Tempo-real	12
6.1 Arquitetura do Protocolo RTP.....	12
6.2 Protocolo de Transferência de Dados RTP	13
6.3 Cabeçalho RTP.....	14
7. RTCP – Protocolo de Controle de Tempo-real	15
7.1 Funções do RTCP.....	15
7.2 Pacotes RTCP.....	16
8. RTSP – Protocolo de Streaming de Tempo-real	17
8.1 Operações do RTSP	17
8.2 Métodos do RTSP	17
8.3 Relacionamento com outros Protocolos.....	19
8.4 Situações do Uso do RTSP.....	19
9. Conclusões	22
Bibliografia.....	24

Índice de Figuras

Figura 1 – Fluxos de dados <i>unicast</i>	4
Figura 2 – Fluxo de dados <i>multicast</i>	4
Figura 3 – <i>Spanning-tree</i> e pacotes gerados da origem ao grupo <i>multicast</i>	5
Figura 4 – Sessão RSVP.....	10
Figura 5 – Modelo <i>host</i> RSVP	11
Figura 6 – Arquitetura do Protocolo RTP	13
Figura 7 – Cabeçalho RTP	14

1. Introdução

Redes de computadores foram projetadas para conectar computadores em diferentes localidades para que eles pudessem compartilhar dados e se comunicar. Antigamente, a maioria dos dados transmitida nas redes era dados textuais. Hoje, com a ascensão das tecnologias de rede e de multimídia, animação, voz e vídeo tornam-se cada vez mais populares na Internet. Telefonia via IP, Internet TV, aprendizado à distância, simulação distribuída, grupos de trabalho distribuídos e vídeo-conferência são somente algumas das aplicações que apresentam caráter multimídia [LIU98].

O casamento entre a multimídia e as redes de computadores promovem em uma escala maior o uso do computador como uma ferramenta de comunicação. Muitos acreditam que algum dia as redes multimídia substituirão o telefone, a televisão e outras invenções que dramaticamente modificaram nossas vidas.

Este trabalho tem por objetivo apresentar os desafios encontrados em se transmitir dados multimídia sobre redes IP, os protocolos que têm por responsabilidade transmitir, sincronizar e apresentar esta categoria de dados, bem como o relacionamento destes protocolos com os demais protocolos da pilha TCP/IP.

2. Rede Multimídia: Metas e Desafios

Projetar uma rede que transmita dados multimídia não é trivial. Pode-se esperar, no mínimo, três dificuldades. Primeiro, comparado com as aplicações tradicionais textuais, dados multimídia geralmente requerem muito mais largura de banda. Um trecho típico de 25s de um filme *QuickTime* de 320x240px precisaria de 2.3 MB, equivalente a 1.000 telas de dados textuais [LIU98].

Segundo, a maioria dos dados multimídia requer tráfego de tempo-real. Dados de áudio e vídeo precisam ser exibidos continuamente na mesma taxa em que foram gerados. Se os dados não chegam a tempo, o processo de exibição para e ouvidos e olhos humanos podem facilmente perceber tal falha. Em uma telefonia via IP, seres humanos podem tolerar uma latência de 250ms; se a latência exceder este limite, a voz parecerá com uma chamada roteada através de um canal de satélite e os usuários reclamarão da qualidade da chamada. Além da latência, congestionamentos na rede também têm efeitos sérios sobre o tráfego de tempo-real. Se a rede estiver congestionada, o único efeito sobre um tráfego que não seja tempo-real é que a transferência levará mais tempo para completar, mas dados de tempo-real se tornam obsoletos e serão descartados se não chegarem a tempo. Se nenhuma ação apropriada for tomada, a retransmissão dos pacotes perdidos agravaria a situação e comprimiria a rede.

Terceiro, fluxos de dados multimídia são geralmente em rajadas. Apenas aumentar a largura de banda não resolverá o problema de transmissão em rajadas. Para a maioria das aplicações multimídia, o receptor tem um *buffer* de tamanho limitado. Se nenhuma medida for tomada para regular o fluxo de dados, ele pode gerar uma sobrecarga (ou um fluxo leve demais) no *buffer* da aplicação. Quando os dados chegarem muito rápido, o *buffer* irá sobrecarregar-se e alguns pacotes serão perdidos, resultando em uma qualidade pobre. Quando os dados chegarem muito lentos, o *buffer* ficará ocioso e a aplicação em espera.

Contrariamente à alta largura de banda e aos dados multimídia de tráfego de tempo-real e em rajadas, na vida real, as redes são compartilhadas por milhões de usuários, e têm largura de banda limitada, atrasos e disponibilidade imprevisíveis. Como resolver esses problemas é um desafio que as redes multimídia precisam encarar.

A possibilidade de responder a esse desafio vem da arquitetura de software de rede existente e do rápido desenvolvimento do hardware. A base da Internet, o TCP/UDP/IP, fornece uma variedade de serviços que as aplicações multimídia podem usar. Embora redes ATM forneçam alta largura de banda, sejam orientadas à conexão e possam atender a diferentes níveis de qualidade de serviço de diferentes tipos de aplicação, muitos poucos usuários têm redes ATM em suas organizações e muito menos em seus próprios *desktops*.

Por outro lado, a Internet está crescendo exponencialmente. Tecnologias LAN e WAN baseadas no protocolo IP conectam redes cada vez maiores em todo o mundo à Internet. Na verdade, a Internet tem se tornado a plataforma da maioria das atividades de rede. Esta é a principal razão para desenvolver protocolos multimídia para a Internet. Um outro benefício de rodar multimídia sobre IP é que os usuários podem ter serviços de dados e multimídia integrados em uma única rede, sem investir em um outro hardware de rede e construir a interface entre as duas redes.

Todavia, uma rede de datagramas compartilhada como a Internet não é naturalmente adequada para tráfego de tempo-real. Para rodar multimídia sobre a Internet, várias questões precisam ser resolvidas. Primeiro, multimídia significa dados extremamente densos e tráfego pesado tendo o hardware de fornecer largura de banda suficiente.

Segundo, aplicações multimídia são geralmente relacionadas a *multicast*, isto é, o mesmo fluxo de dados, e não múltiplas cópias, é enviado a um grupo de receptores. Em uma vídeo-conferência, dados de vídeo precisam ser enviados a todos os participantes ao mesmo tempo. Protocolos para aplicações multimídia precisam usar *multicast* para reduzir o tráfego gerado por estas. Terceiro, o preço de usar uma rede de recursos compartilhados é sua disponibilidade imprevisível. Aplicações de tempo-real requerem largura de banda garantida quando a transmissão começa. Assim, deve haver mecanismos de reserva de recursos ao longo do caminho de transmissão.

Quarto, a Internet é uma rede de datagramas comutada por pacotes na qual pacotes são roteados independentemente ao longo das redes compartilhadas. Tecnologias atuais não garantem que dados de tempo-real alcancem seu destino sem serem desordenados ou perdidos. Novos protocolos de transporte precisam cuidar de questões de temporização para que o áudio e o vídeo sejam exibidos continuamente com a sincronização e a temporização corretas. Por último, devem existir algumas operações-padrão para aplicações que gerenciam a entrega e apresentação de dados multimídia.

A solução para todas essas questões levantadas é classificar todo o tráfego, alocar prioridades para diferentes aplicações e fazer reservas de recursos. O grupo de trabalho de Serviços Integrados do IETF (*Internet Engineering Task Force*) desenvolveu um modelo de serviços avançado chamado Serviços Integrados que incluem serviços de melhor esforço e de tempo-real. Serviços de tempo-real capacitarão redes IP a fornecer qualidade de serviço às aplicações multimídia. O próximo capítulo apresenta um dos requisitos básicos de infraestrutura IP para dados multimídia: o fornecimento de qualidade de serviço.

3. Qualidade de Serviço - QoS

Em uma definição mais genérica, Qualidade de Serviço (QoS) significa fornecimento consistente e previsível no serviços de distribuição de dados, satisfazendo as exigências das aplicações. QoS é a capacidade que um elemento da rede, seja ele uma aplicação, um *host* ou um roteador, tem de oferecer um certo nível de garantia de que o tráfego e o serviço requerido possam ser satisfeitos. Para que se consiga QoS, é necessária a cooperação de todas as camadas da rede, desde o topo até o fim, assim como cada um dos elementos da rede de uma conexão fim-a-fim [SOU99].

QoS não cria *bandwidth*. Não é possível para uma rede oferecer o que ela não tem, portanto a disponibilidade de *bandwidth* é o ponto de partida. QoS somente gerencia a *bandwidth* de acordo com a demanda da aplicação e de acordo com a configuração da rede. Portanto, a QoS, com um nível de garantia de serviço assegurado, requer uma alocação de recursos para cada fluxo de dados. Uma prioridade para a QoS tem sido assegurar que o esforço de conseguir um tráfego melhor não seja finalizado quando as reservas são feitas. Aplicações que exigem uma QoS (aplicações de mais alta prioridade) não devem desativar aplicações Internet (aplicações de mais baixa prioridade).

Diferentes aplicações em uma mesma rede podem ter diferentes subconjuntos de parâmetros de qualidade de serviço, com diferentes valores requeridos, e alguns parâmetros não devem ser mutuamente independentes. Em geral é difícil separar parâmetros de qualidade de serviço de outros parâmetros do sistema. Entretanto, uma característica que diferencia, é que parâmetros de qualidade de serviço estão sujeitos a negociação entre componentes do sistema.

O tipo de aplicação tem influência decisiva nos parâmetros requeridos do sistema. Os requisitos de qualidade de serviço podem mudar durante uma sessão. Uma teleconsulta médica usando vídeo de baixa qualidade, em um momento precisa mostrar imagens de raios-X de alta qualidade; isto requer renegociação dos parâmetros de qualidade de serviço para aumentar a *bandwidth* para imagens de raios-X.

Latência

Tempo gasto pela mensagem entre um nó transmissor e o receptor. Este *delay* pode acontecer no caminho da transmissão ou em um dispositivo neste caminho. Em um roteador, a latência é a quantidade de tempo gasto entre um pacote quando é recebido e quando o mesmo é enviado. Isto também causa a propagação de *delay* [SOU99].

Jitter

Problema que ocorre quando um vídeo ou som é transmitido através da rede e pacotes não chegam a seu destino em ordem ou no tempo certo; eles são causados pela latência. A distorção deste sinal é a sua variação em relação ao sinal original. *Jitter* é uma distorção de um pacote que chega comparado com o pacote original. Esta distorção, particularmente, danifica transmissões de dados multimídia.

Bandwidth

Medida da capacidade de transmissão de dados, geralmente expressados em *kilobits* por segundo (Kbps) ou *megabits* por segundo (Mbps). *Bandwidth* indica, teoricamente, a

capacidade máxima de uma conexão, mas como a teoria da *bandwidth* é aproximada, fatores negativos como *delays* na transmissão causam deterioração na qualidade. Quando a *bandwidth* é aumentada, pode-se transferir mais dados.

Pacotes Perdidos

Taxa de pacotes perdidos durante uma transmissão, geralmente expressa em porcentagem.

4. IP Multicast

Extensão do protocolo IP no qual aplicações enviam uma cópia da informação a um endereço de grupo alcançando todos os destinatários que querem recebê-la. Sem o *multicasting*, a mesma informação precisa ser enviada ao longo da rede várias vezes, uma para cada destino, ou enviando um *broadcasting* a todos da rede, consumindo processamento e largura de banda desnecessárias, e/ou limitando o número de participantes [IPM95]. Envolve grupos de receptores e transmissores que participam de sessões *multicast*; somente aqueles receptores em um grupo realmente recebem o tráfego para aquela sessão do grupo.

Na **Figura 1**, três cópias dos mesmos dados (D) são enviadas ponto-a-ponto como D1, D2 e D3 aos receptores 1, 2 e 3 em uma aplicação vídeo-conferência. Estas são transmissões *unicast*, enviadas ponto-a-ponto de um transmissor a um receptor [IPM95].

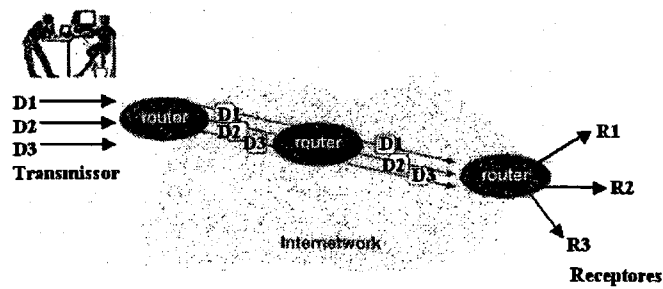


Figura 1 – Fluxos de dados unicast

Na **Figura 2**, uma cópia do mesmo dado (D) é enviada em *multicast* aos receptores 1, 2 e 3 em uma aplicação vídeo-conferência. Note as economias de largura de banda localmente e ao longo da rede – imagine as economias de largura de banda para 100 receptores.

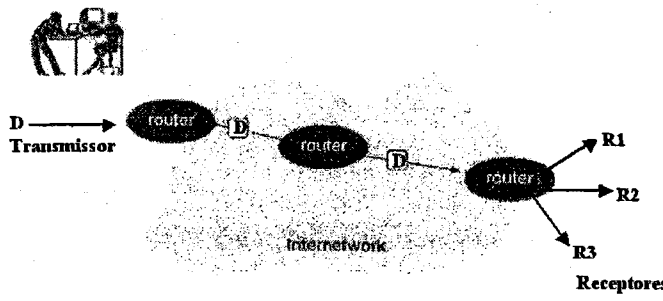


Figura 2 – Fluxo de dados multicast

Multicasting feito dentro de um único segmento de LAN é ela própria. O IEEE 802 e outros protocolos de LANS fornecem endereços *multicast* ao nível de enlace. Um pacote com um endereço *multicast* é transmitido em um segmento de LAN. Aquelas estações que forem membros do grupo *multicast* correspondente reconhecem o endereço *multicast* e aceitam o pacote. Neste caso, somente uma única cópia do pacote é transmitida. Esta técnica funciona por causa da natureza de *broadcast* (difusão) de uma LAN: uma transmissão de qualquer estação é recebida por todas as outras estações da mesma LAN [STA98].

Em um ambiente Internet, o *multicasting* funciona um pouco diferente. Considere a **Figura 3**, onde LANs são interconectadas por roteadores. Os roteadores conectam-se uns aos outros através de linhas de alta velocidade ou por meio de uma WAN (rede N4) [STA98]. O caminho de menor custo da origem a cada rede que inclui os membros do grupo *multicast* é determinado. Isto resulta em uma *spanning-tree* (árvore de distribuição) da configuração. Perceba que esta não é uma árvore de distribuição completa da configuração, pois ela inclui apenas as redes que contêm os membros do grupo *multicast*.

A origem transmite um único pacote sobre N1 para o roteador D. O roteador D faz duas cópias do pacote, para transmitir sobre as linhas L3 e L4. O roteador B recebe o pacote de L3 e o transmite em N3, onde ele é lido pelos membros do grupo *multicast* na rede. Enquanto isso, o roteador C recebe o pacote enviado em L4. Ele precisa agora entregar esse pacote para E e F. Se a rede N4 for uma rede *broadcast* (*Ethernet*), C precisa somente transmitir uma cópia do pacote para ambos os roteadores lerem. Se N4 for uma WAN comutada por pacotes, C precisa fazer duas cópias do pacote e endereçar uma ao roteador E e outra a F. Cada um desses roteadores, por sua vez, retransmite o pacote recebido em N5 e N6, respectivamente. Com a técnica *multicast*, necessita-se de somente 8 cópias do pacote. Se fossem utilizadas as técnicas de *broadcasting* e *unicasting*, seriam necessários 13 e 11 cópias do pacote original até atingir todos os destinos desejados, respectivamente.

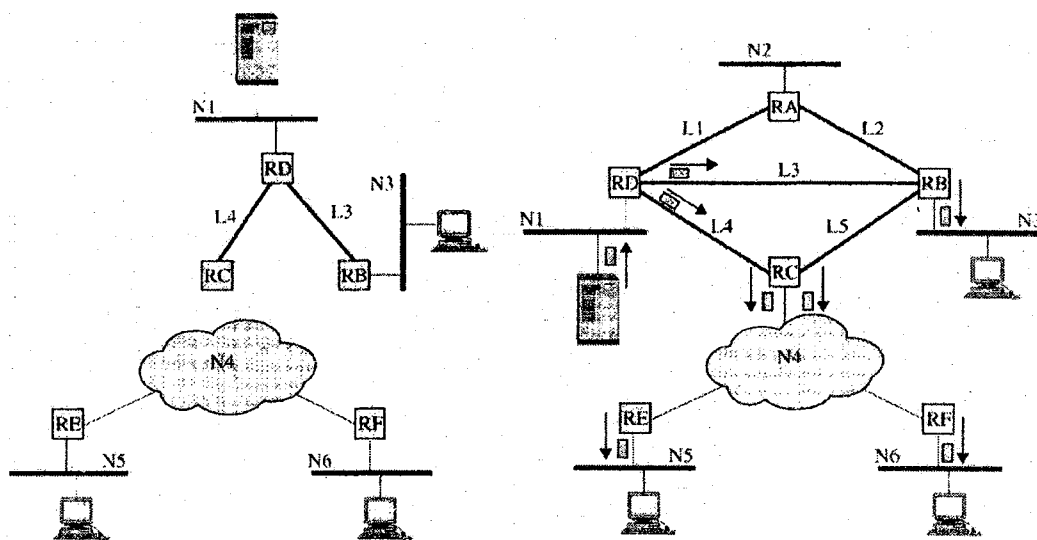


Figura 3 – *Spanning-tree* e pacotes gerados da origem ao grupo *multicast*

4.1 Taxonomia de Aplicações IP Multicast

As relações entre o transmissor e o receptor são o que principalmente diferenciam as aplicações *multicast* das aplicações *unicast*. Pode-se, assim, caracterizar três categorias gerais de aplicações *multicast* [QUI98]:

Um-para-muitos: um único *host* envia para dois ou mais receptores. Aplicações que fazem distribuição de áudio e vídeo agendada (palestras, aulas, reuniões, etc.), assim como aquelas que realizam monitoramento (preços de estoque, equipamento sísmico, sistemas de segurança, etc.) estão incluídas nesta categoria.

Muitos-para-um: qualquer número de receptores enviando dados de volta a uma origem transmissora via *unicast* ou *multicast*. Aplicações dessa categoria são tipicamente aplicações requisição-resposta (RR) bidirecional, onde “ou muitos ou um” podem gerar a requisição, como aplicações que realizam descoberta de recursos em sistemas distribuídos (localização de serviços), e *juke boxes*, nas quais usuários enviam suas solicitações de áudio dentro de uma fila agendada.

Muitos-para-muitos: qualquer número de *hosts* enviando para o mesmo endereço de grupo *multicast*, bem como recebendo dele. Aplicações que realizam conferências multimídia estão incluídas nesta categoria.

4.2 Requisitos para Multicasting

Em uma transmissão comum *unicast* na Internet onde cada datagrama tem um endereço único de destino, a tarefa de cada roteador é de retransmitir o datagrama através do caminho mais curto desse roteador para a rede de destino. Com a transmissão *multicast*, ao roteador pode ser requerido retransmitir duas ou mais cópias de um datagrama de entrada. Portanto, a funcionalidade geral de um roteamento *multicast* é mais complexa do que de um roteamento *unicast*. Segue abaixo uma lista de funções necessárias:

1. Uma convenção é necessária para identificar um endereço *multicast*. O IPv4 apresenta uma classe de endereçamento específica para o *multicasting*, a classe de endereços D. O IPv6 não possui uma noção de classes, mas tem um modelo de endereço no qual os 8 bits mais significativos do pacote IP são todos 1.

2. Cada nó da rede precisa traduzir entre um endereço IP *multicast* e uma lista de redes que contém membros deste grupo. Esta informação permite que o nó construa uma árvore de distribuição de caminho mais curto para todas as redes que contêm os membros do grupo.

3. Um roteador precisa traduzir entre um endereço IP *multicast* e um endereço *multicast* de uma sub-rede a fim de entregar um datagrama IP *multicast* na rede de destino. Por exemplo, nas redes IEEE 802, incluindo a Ethernet, um endereço MAC tem 16 ou 48 bits. Em ambos os casos, se o bit de mais alta ordem for 1, então ele é um endereço *multicast*. Portanto, para a entrega *multicast*, um roteador conectado a uma rede IEEE 802 precisa traduzir um endereço IPV4 de 32 bits ou IPV6 de 128 bits em um endereço IEEE *multicast* MAC de 16 ou 48 bits.

4. Embora alguns endereços *multicast* possam ser relacionados permanentemente, o caso mais comum é que os endereços *multicast* sejam gerados dinamicamente e que hosts individuais possam entrar e sair dos grupos *multicast* dinamicamente. Portanto, um mecanismo é necessário para que um host individual informe aos roteadores conectados à

mesma rede que ele de sua inclusão e exclusão de um grupo *multicast*. O IPv4 utiliza o protocolo IGMP (*Internet Group Management Protocol*) para gerenciar o dinamismo existente entre os membros de grupos *multicast*. O IPv6 utiliza o IGMPv6.

5. Os roteadores precisam trocar informações. Primeiro, eles precisam saber quais sub-redes incluem membros de um dado grupo *multicast*. Segundo, eles precisam de informação suficiente para calcular o menor caminho a cada rede que contém os membros do grupo. Estas exigências implicam na necessidade de um protocolo de roteamento. O IPv4 utiliza os protocolos de roteamento *multicast* DVMRP (*Distance Vector Multicast Routing Protocol*), MOSPF (*Multicast Open Shortest Path First*) ou PIM (*Protocol Independent Multicast*). Não foram encontrados na literatura protocolos para *multicasting* que trabalhem com o esquema de endereçamento do IPv6, no entanto, como já foram feitas as versões para IPv6 dos protocolos RIP (*Routing Internet Protocol*) e OSPF (*Open Shortest Path First*) para *unicast*, espera-se que os protocolos DVMRP e MOSPF recebam tal *upgrade*.

6. Cada roteador precisa determinar os caminhos de roteamento *multicast* com base nos endereços de origem e destino.

7. Receptores heterogêneos: enviar pacotes a receptores com uma vasta variedade de capacidades de largura de banda, características de latência e congestionamento de rede. Isso acaba acarretando a dúvida sobre que protocolo de transporte usar: TCP ou UDP [QUI98].

8. Entrega confiável de dados: garantir que todos os dados enviados são recebidos por todos os receptores. Certas aplicações *multicast* aceitam a perda de dados, por exemplo, uma videoconferência. No entanto, há aplicações que são intolerantes a perdas, por exemplo, *Web-caching*.

5. RSVP – Protocolo de Reserva de Recursos

Um esquema de roteamento dinâmico pode rapidamente responder a falhas na rede ao rotear pelos pontos de falha. Além disto, roteamento dinâmico pode tratar com congestionamentos, primeiro com o balanceamento de carga para nivelar a carga ao longo da rede, e segundo por rotear em regiões de desenvolvimento de congestionamento usando roteamento de menor custo. No caso do *multicasting*, esquemas de roteamento dinâmicos têm sido suplementados com capacidades de roteamento *multicast* que tiram vantagem de caminhos compartilhados de uma fonte a destinos *multicast* para minimizar o número de pacotes duplicados [STA98].

Uma outra ferramenta disponível aos roteadores é a habilidade de processar pacotes com base no rótulo de QoS. Os roteadores podem (1) usar uma fila que preferêncie pacotes com base em QoS; (2) selecionar rotas alternativas com base em características de QoS de cada caminho; e (3) quando possível, invocar o tratamento de QoS na sub-rede do próximo *hop*.

Todas estas técnicas são para tratar o tráfego presente na rede, mas não são preventivos de nenhuma forma. Baseado somente no uso de roteamento dinâmico e QoS, um roteador é incapaz de antecipar congestionamentos e prever que as aplicações causem uma sobrecarga. O roteador pode simplesmente fornecer um serviço de entrega por melhor esforço, no qual os pacotes podem ser perdidos e outros podem ser entregues com menos QoS que a exigida.

Enquanto a demanda na rede cresce, parece que essa prevenção assim como a reação a congestionamentos é necessária. Este capítulo mostra uma estratégia de prevenção chamada reserva de recursos e o protocolo que a implementa é o RSVP (*Resource Reservation Protocol*), ou Protocolo de Reserva de Recursos.

Medidas preventivas podem ser úteis, tanto em transmissões *unicast* quanto *multicast*. Para *unicast*, duas aplicações concordam em uma QoS para uma sessão e esperam que a rede suporte essa QoS. Se a rede estiver carregada, ela pode não fornecer a QoS desejada e assim entregará os pacotes em um nível de QoS reduzido. Nesse caso, as aplicações poderiam ter preferido esperar antes de iniciar a sessão ou, no mínimo, terem sido alertadas imediatamente para reduzir a QoS. Uma forma de tratar com esta situação é reservar recursos para as aplicações a fim de atingir a QoS exigida. Os roteadores ao longo do caminho poderiam então pré-alocar recursos (espaço na fila de pacotes de entrada, capacidade de saída) para assegurar a QoS necessária. Se um roteador não pode atender à reserva de recursos pelo fato de reservas anteriores, então a aplicação poderia ser informada. As aplicações podem, então decidir tentar posteriormente, ou reduzir a QoS.

Portanto, o uso de reserva de recursos pode capacitar os roteadores a decidir antecipadamente se eles podem atender aos requisitos de entrega da transmissão *multicast* a todos os receptores *multicast* e os capacita a reservar os recursos apropriados, se possível.

5.1 Características do RSVP

Baseado em suas metas de projeto, as características do RSVP [BRA97] são:

- *Unicast e multicast*: RSVP faz reservas tanto para transmissões *unicast* quanto para *multicast*, adaptando-se dinamicamente às mudanças de membros nos grupos assim como às mudanças de rotas, e reservas de recursos baseadas nas exigências individuais dos membros *multicast*;
- Unidirecional: RSVP faz reservas para fluxo de dados unidirecional. Trocas de dados entre dois sistemas comunicantes finais exigem reservas de recursos nas duas direções;
- Reserva iniciada pelo receptor: o receptor de um fluxo de dados inicia e mantém a reserva de recursos para esse fluxo;
- Manutenção de *soft states* na Internet: RSVP mantém *soft states* nos roteadores intermediários e deixa a responsabilidade para mantê-los para os usuários finais;
- Fornecimento de estilos de reserva diferentes: RSVP permite que os usuários especifiquem como as reservas para o mesmo grupo *multicast* devem ser agregadas nos roteadores intermediários. Esta característica permite uso mais eficiente dos recursos da Internet;
- Operação transparente através de roteadores não-RSVP: como reservas de recursos RSVP são independentes do protocolo de roteamento, não há conflito fundamental em um ambiente misto, onde alguns roteadores não empregam o RSVP. Estes roteadores simplesmente usarão a técnica de entrega de melhor-esforço;
- Suporte ao IPv4 e ao IPv6: o RSVP pode explorar o campo *Tipo de Serviço* do cabeçalho IPv4 e o campo *Rótulo de Fluxo* do cabeçalho IPv6.

Vale a pena destacar duas dessas características relacionadas acima: a reserva iniciada pelo receptor e os *soft states*.

Reserva Iniciada pelo Receptor

Como já dito anteriormente, diferentes membros de um mesmo grupo *multicast* podem ter diferentes exigências de requisitos. Se o fluxo de transmissão origem pode ser dividido em sub-fluxos, então alguns membros *multicast* podem somente requisitar um único fluxo. Se existem múltiplas origens de transmissão para um grupo *multicast*, então um particular receptor *multicast* pode querer selecionar somente uma origem ou um subconjunto de todas as origens de transmissão para receber. Finalmente, exigências de QoS de diferentes receptores podem diferir dependendo do equipamento de saída, poder computacional, e velocidade do *link* do receptor.

Portanto, faz sentido os receptores serem responsáveis pelas reservas de recursos. Um transmissor precisa fornecer aos roteadores as características de tráfego da transmissão (taxa de dados, variabilidade, etc), mas são os receptores que devem especificar a QoS desejada. Os roteadores podem então agregar reservas de recursos *multicast* para tirar vantagem dos segmentos de caminho compartilhados ao longo da árvore de distribuição.

Soft State

O RSVP emprega a abordagem *soft state* ou não-orientada à conexão na qual o estado da reserva é mantido em cache nos roteadores que é instalado e periodicamente atualizado pelos sistemas finais. Se um estado não for atualizado dentro de um limite de tempo obrigatório, o roteador descarta o estado. Se uma nova rota torna-se preferível para um dado fluxo, os sistemas finais fornecem a reserva aos novos roteadores da rota.

5.2 Fluxos de Dados do RSVP

Três conceitos relacionados aos fluxos de dados formam a base da operação do RSVP: sessão, especificação de fluxo e especificação de filtro [STA98]. Sessão é um fluxo de dados identificado pelo seu destino. A razão para usar o termo sessão ao invés de simplesmente destino é que ele reflete a natureza *soft state* da operação RSVP. Uma vez que a reserva é feita em um roteador por um destino particular, o roteador considera isto como uma sessão e aloca os recursos para a vida desta sessão. Em particular, uma sessão é definida como a seguir:

Sessão = endereço IP de destino + identificador de protocolo IP + porta de destino

O endereço IP de destino pode ser *unicast* ou *multicast*. O identificador de protocolo indica o usuário do IP (TCP ou UDP, por exemplo), e a porta de destino é a porta TCP ou UDP para o usuário desse protocolo da camada de transporte. Se o endereço for *multicast*, a porta de destino pode não ser necessária, porque há tipicamente um endereço *multicast* diferente para diferentes aplicações.

Um pedido de reserva emitido por um sistema final de destino é chamado de descritor de fluxo, e consiste de um *flowspec* (especificação de fluxo) e um *filter spec* (especificação de filtro). A especificação de fluxo aponta uma QoS desejada e é usado para configurar parâmetros em um escalonador de pacotes de um nó. Isto é, o roteador transmitirá os pacotes com um dado conjunto de preferências baseadas nas especificações de fluxo correntes. A especificação de filtro define o conjunto de pacotes para o qual uma reserva é solicitada.

Portanto, a especificação de filtro juntamente com a sessão definem o conjunto de pacotes, ou fluxo, que estão para receber a QoS desejada. Quaisquer outros pacotes endereçados ao mesmo destino são tratados como tráfego de melhor esforço. Em geral, o conteúdo da especificação de fluxo é o seguinte:

$$\text{Especificação de fluxo} = \text{classe de serviço} + \text{Rspec} + \text{Tspec}$$

A classe de serviço é um identificador de um tipo de serviço sendo solicitado; ela inclui informações usadas pelo roteador para unir solicitações. Os outros dois parâmetros são conjuntos de valores numéricos. O parâmetro *Rspec* (R de reserva) descreve a QoS desejada, e o *Tspec* (T de tráfego) descreve o fluxo de dados. Os conteúdos de *Rspec* e *Tspec* são opacos para o RSVP.

Em princípio, a especificação de filtro pode designar um subconjunto arbitrário dos pacotes de uma sessão. Por exemplo, uma especificação de filtro poderia especificar somente fontes ou origens específicas, ou protocolos de origem específicos, ou em geral pacotes que tem compatibilidade com certos campos de cabeçalhos de quaisquer protocolos no pacote. A versão do RSVP atual usa uma especificação de filtro restrita consistindo dos seguintes elementos:

$$\text{Especificação de filtro} = \text{endereço de origem} + \text{porta de origem TCP/UDP}$$

A Figura 4 indica o relacionamento entre sessão, especificação de fluxo e especificação de filtro. Cada pacote de entrada é parte de no máximo uma sessão e é tratado de acordo com o fluxo lógico indicado na figura para essa sessão. Se um pacote não pertence a nenhuma sessão, a ele é dado um serviço de entrega por melhor esforço.

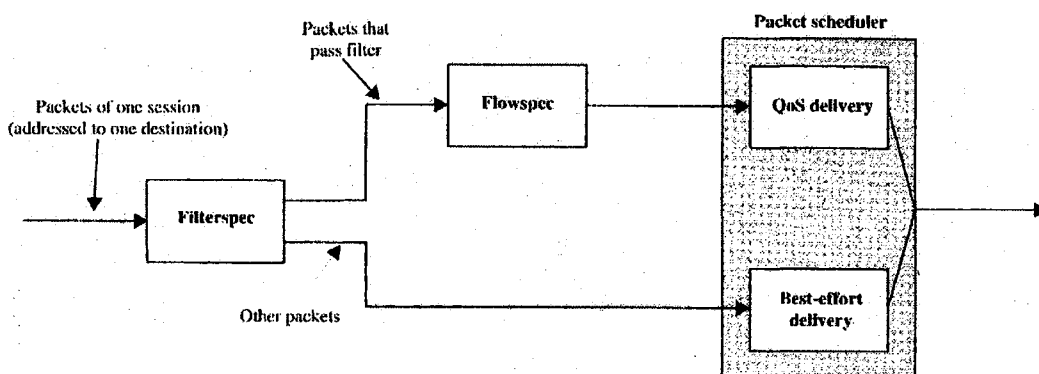


Figura 4 – Sessão RSVP

5.3 Mecanismos do Protocolo RSVP

O RSVP usa dois tipos básicos de mensagens: *Resv* e *Path*. Mensagens *Resv* originam-se em um grupo de receptores *multicast* e propagam-se em direção aos transmissores através da árvore de distribuição, sendo unidas e empacotadas quando apropriado em cada nó ao longo do caminho. Estas mensagens criam *soft states* nos roteadores da árvore de distribuição que define os recursos reservados para esta sessão. Por fim, as mensagens *Resv* unidas alcançam os *hosts* transmissores, permitindo que os *hosts* configurem parâmetros de controle de tráfego apropriados para o primeiro *hop* [STA98].

A mensagem *Path* é usada para fornecer informação de roteamento em sentido contrário ao da transmissão, assim como as mensagens *Resv*. Em todos os protocolos de roteamento *multicast* atualmente em uso, somente uma rota no sentido transmissor-receptor, na forma de uma árvore de distribuição, é mantida. Entretanto, as mensagens *Resv* devem propagar no sentido contrário ao da transmissão através de todos os roteadores intermediários e para todos os *hosts* transmissores. Na ausência de informação de roteamento reversa do protocolo de roteamento, o RSVP fornece isto com a mensagem *Path*. Cada *host* que deseja participar como um transmissor em um grupo *multicast* emite uma mensagem *Path* que é transmitida ao longo de toda a árvore de distribuição a todos os destinos *multicast*. Ao longo do caminho, cada roteador e cada *host* de destino cria um estado do caminho que indica o salto inverso para ser usado pela origem. Os caminhos tomados por estas mensagens são os mesmos caminhos tomados pelos pacotes de dados.

A **Figura 5** ilustra a operação do protocolo da perspectiva do *host*. Os seguintes eventos ocorrem:

- Um receptor se une a um grupo multicast ao enviar uma mensagem *join* IGMP a um roteador vizinho;
- Um transmissor em potencial emite uma mensagem *Path* ao endereço de grupo *multicast*;
- Um receptor recebe uma mensagem *Path* identificando o transmissor;
- Agora que o receptor tem a informação do caminho inverso, ele pode começar a enviar mensagens *Resv*, especificando os descritores de fluxo desejados;
- A mensagem *Resv* propaga-se através da rede e é entregue ao transmissor;
- O transmissor começa a enviar pacotes de dados;
- O receptor começa a receber os pacotes de dados.

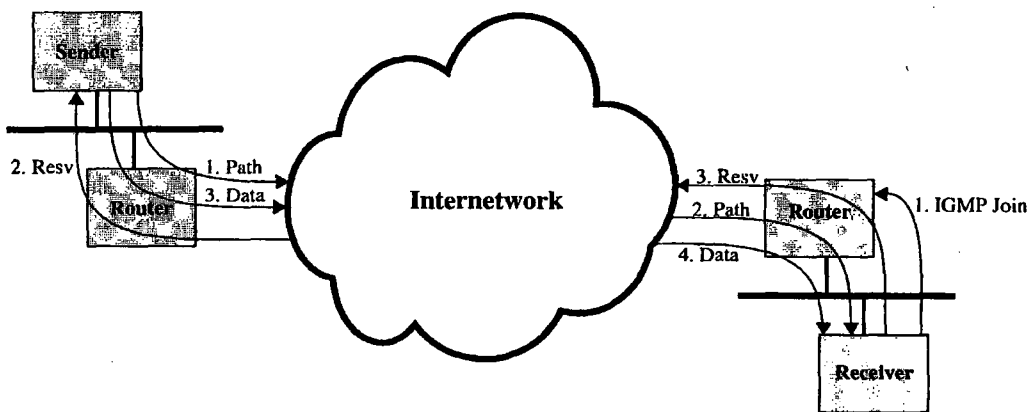


Figura 5 – Modelo *host* RSVP

6. RTP – Protocolo de Transporte de Tempo-real

O protocolo de transporte da arquitetura Internet mais usado é o TCP. Embora tenha dado sua contribuição a uma larga variedade de aplicações distribuídas, esse protocolo não é adequado para uso com aplicações de tempo-real. Um número de características do TCP o desqualificam para uso como protocolo de transporte para tais aplicações [STA98]:

- TCP configura uma conexão entre dois pontos, não sendo adequado para distribuição *multicast*.
- TCP inclui mecanismos para retransmissão de segmentos perdidos que, então, chegam fora de ordem. Tais mecanismos não são úteis para a maioria das aplicações de tempo-real.
- TCP não possui qualquer mecanismo para associação de informação de tempo aos segmentos, outro requisito de dados de tempo-real.

Outro protocolo largamente usado, o UDP, não apresenta as duas primeiras características, mas também não fornece informação de temporização.

Embora cada aplicação de tempo-real possa ter seus próprios mecanismos para suportar transporte em tempo-real, há um número de características comuns que justificam a definição de um protocolo comum. O protocolo projetado para o propósito de transporte de dados com características de tempo-real é chamado de RTP, definido na RFC 1889.

Uma distinção precisa ser feita entre aplicações de comunicação de tempo-real leves e rígidas. Aplicações de tempo-real leves podem tolerar a perda de alguma porção de dados, enquanto as rígidas, têm tolerância zero a perdas. Em geral, aplicações de tempo-real leves impõem menos exigências na rede, permitindo que se maximize a utilização da rede, mesmo à custa de alguns pacotes perdidos ou fora de ordem. Em aplicações de tempo-real rígidas um limite superior determinístico no *jitter* e alta confiabilidade precedem as considerações de utilização da rede. O RTP é mais bem adequado à comunicação de tempo-real leve. Ele não apresenta os mecanismos necessários para suportar tráfego de tempo-real rígido.

6.1 Arquitetura do Protocolo RTP

No RTP, há um forte acoplamento entre a funcionalidade do RTP e a funcionalidade da camada de aplicação. Além disso, o RTP é mais bem visto como um modelo que as aplicações podem usar diretamente para implementar um único protocolo. Sem a informação específica da aplicação, o RTP não é um protocolo completo. Por outro lado, o RTP impõe uma estrutura e define funções comuns para que aplicações de tempo-real individuais sejam auxiliadas em parte de suas responsabilidades. Dois conceitos-chave do projeto de arquitetura que o RTP segue são: o enquadramento ao nível de aplicação e o processamento de camadas integradas [STA98].

Enquadramento ao Nível de Aplicação

A aplicação transmissora pode recalcular os valores dos dados perdidos ao invés de armazená-los. Ela pode também fornecer valores aproximados, ao invés de simplesmente retransmitir valores perdidos, ou enviar novos dados que corrijam as consequências da perda original.

Para permitir que a aplicação tenha controle sobre a função de retransmissão, as camadas inferiores, tais como apresentação e transporte, devem tratar os dados nas unidades que a aplicação especificar. A aplicação deve quebrar o fluxo de dados em unidades de dados ao nível de aplicação (ADUs), e as camadas inferiores devem preservar estes limites dos ADUs enquanto processam os dados. O quadro de nível da aplicação é a unidade de recuperação de erros. Portanto, se uma porção de uma ADU é perdida na transmissão, a aplicação será incapaz de fazer uso das porções restantes. Nesse caso, a camada de aplicação descartará todas as porções que estão chegando e preparar a retransmissão de todo o ADU, se necessário.

Processamento de Camadas Integradas

A idéia é que camadas adjacentes possam ser fortemente acopladas e que o implementador possa estar livre para implementar as funções naquelas camadas de forma fortemente acopladas. A Figura 6 ilustra a maneira na qual o RTP concretiza o processamento de camadas integradas. O RTP é projetado para rodar acima de um protocolo de transporte não-orientado à conexão, tal como o UDP. O UDP fornece a funcionalidade básica de endereçamento de portas da camada de transporte. O RTP tem funções de nível de transporte adicionais, como seqüenciamento. Entretanto, o RTP só se torna completo por modificações e/ou adicionamentos para os cabeçalhos RTP incluírem a funcionalidade na camada de aplicação. A Figura 6 indica que vários padrões diferentes para codificação de dados de vídeo podem ser usados em conjunto com o RTP para a transmissão de vídeo.

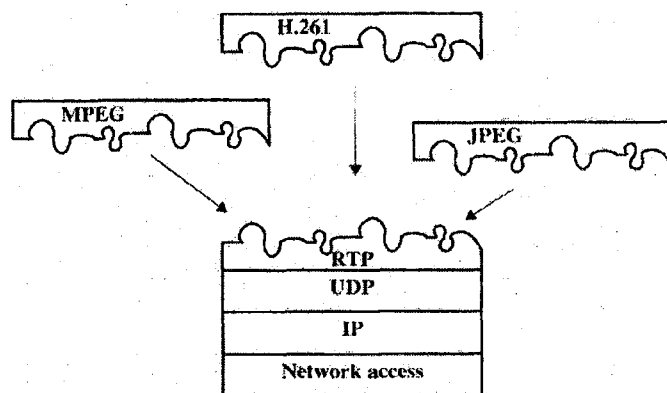


Figura 6 – Arquitetura do Protocolo RTP

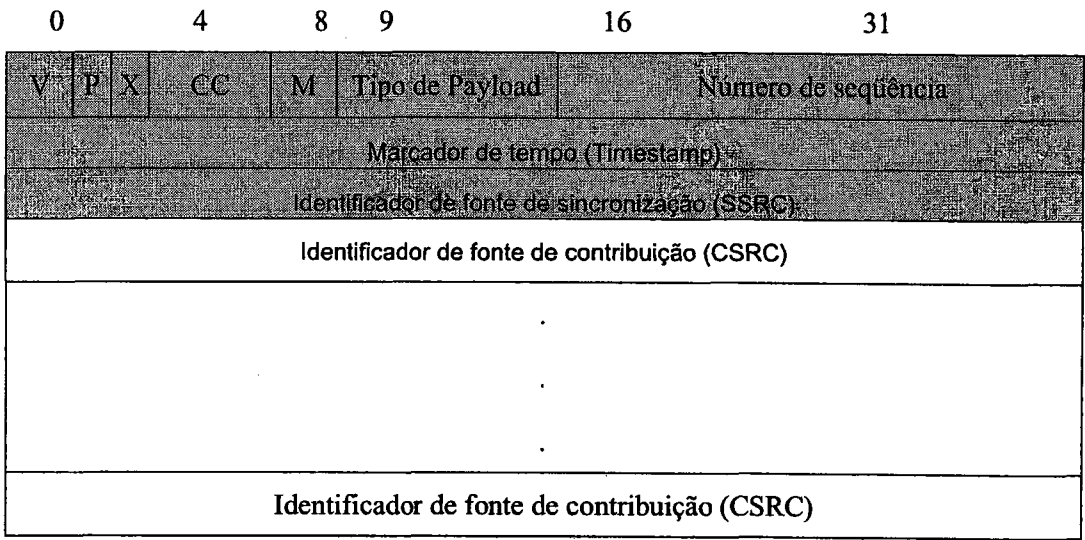
6.2 Protocolo de Transferência de Dados RTP

O RTP suporta a transferência de dados de tempo-real entre um número de participantes em uma sessão. Uma sessão é simplesmente uma associação lógica entre duas ou mais entidades RTP que é mantida pela duração da transferência de dados. Uma sessão é definida pelo seguinte:

- **Nº da porta RTP:** endereço da porta destino usado por todos participantes para transferências RTP. Se o UDP está na camada inferior, esta porta aparece no campo *Destination Port* do cabeçalho UDP.
- **Nº da porta RTCP:** endereço da porta destino usado por todos os participantes para transferências RTCP (vide próximo capítulo).

- **Endereços IP dos participantes:** pode ser endereço IP *multicast*, para que o grupo *multicast* defina os participantes, ou um conjunto de endereços IP *unicast*.

6.3 Cabeçalho RTP



V = versão - P = complemento - X = extensão - CC = contador de CSRC - M = marcador

Figura 7 – Cabeçalho RTP

Cada pacote RTP inclui um cabeçalho fixo e pode incluir campos adicionais específicos da aplicação. A **Figura 7** mostra o cabeçalho RTP de tamanho fixo. Os 12 primeiros bytes (parte sombreada) estão sempre presentes e consistem dos seguintes campos [SCH96a]:

- **Versão** (2 bits): a versão atual é a 2.
- **Complemento** (1 bit): indica se bytes complementares aparecerão no final dos dados transmitidos.
- **Extensão** (1 bit): quando setado, o cabeçalho é seguido por exatamente um cabeçalho de extensão, usado para extensões experimentais do RTP.
- **Contador de CSRC** (4 bits): o número de identificadores de CSRC que acompanham o cabeçalho.
- **Marcador** (1 bit): sua interpretação depende ao tipo de dado que está sendo carregado; para vídeo, ele é setado para marcar o fim de um quadro; para áudio, ele é setado para marcar o começo de uma conversa.
- **Tipo de payload** (7 bits): identifica o formato do tipo de dado RTP sendo transmitido.
- **Número de sequência** (16 bits): cada fonte inicia com um número randômico de sequência, que é decrementado de um para cada pacote RTP enviado, permitindo a detecção de perda e seqüenciamento de pacotes dentro de uma série de pacotes com o mesmo marcador de tempo.

- **Marcador de tempo (32 bits):** corresponde ao instante da geração do primeiro byte de dados do *payload*. Os valores precisam ser gerados a partir de um *clock* local na fonte.
- **Identificador de fonte de sincronização (32 bits):** um valor randomicamente gerado que identifica de forma única uma fonte em uma sessão.

Seguindo o cabeçalho fixo, pode haver um ou mais (no máximo 15) do seguinte campo:

- **Identificador de fonte de contribuição (32 bits):** identifica uma fonte de contribuição para o *payload*. Estes identificadores são fornecidos por um *mixer*.

O campo Tipo de payload identifica o tipo da mídia e o formato dos dados, incluindo o uso de compressão, encriptação, tais como: JPEG vídeo, MPA áudio, CellB vídeo, PCMA áudio, H261 vídeo, etc.

Pacotes de dados RTP não contêm nenhum campo de comprimento, portanto o tamanho máximo dos pacotes RTP é limitado somente pelos protocolos inferiores.

7. RTCP – Protocolo de Controle de Tempo-real

O RTP é usado somente para o transporte de dados de tempo-real, tipicamente na forma multicast entre todos os participantes em uma sessão. Um protocolo de controle separado chamado RTCP também opera na forma multicast para fornecer um feedback às fontes de dados RTP assim como a todos os participantes da sessão. O RTCP usa o mesmo serviço de transporte que o RTP (em geral, UDP) e um número de porta separado. Cada participante da sessão periodicamente emite um pacote RTCP a todos os outros membros da sessão.

7.1 Funções do RTCP

A RFC 1889 destaca quatro importantes funções desempenhadas pelo RTCP [LIU98]:

- **Monitoramento de QoS e controle de congestionamento:** esta é a principal função do RTCP. O RTCP fornece o *feedback* para uma aplicação sobre a qualidade da distribuição dos dados. A informação de controle é útil para transmissores, receptores e administradores de rede. O transmissor pode ajustar sua taxa de transmissão baseada no *feedback* relatado pelo receptor. Os receptores podem indicar problemas ocorridos como a perda de pacotes e o *jitter* em excesso. O *feedback* dos receptores permite que os administradores de rede possam avaliar o desempenho da rede para distribuição *multicast*, determinar se um congestionamento é local, regional ou global.
- **Identificação da fonte:** nos pacotes RTCP, as fontes são identificadas por números de 32 bits gerados randomicamente. Estes identificadores não são convenientes para os usuários. Os pacotes RTCP SDES contêm informação textual chamada de nomes canônicos como identificadores globalmente únicos dos participantes da sessão. Eles podem conter o nome do usuário, o número do seu telefone, seu endereço de e-mail e outras informações.

- Sincronização entre as mídias: os relatórios RTCP do transmissor contêm uma indicação de tempo real e o marcador de tempo correspondente RTP. Isto pode ser usado na sincronização entre as mídias, como sincronização em vídeo.
- Controle da quantidade de informações RTCP: os pacotes RTCP são enviados periodicamente entre os participantes. Quando o número de participantes aumenta, é necessário balancear entre obter informação de controle atualizada e limitar o tráfego de controle. A fim de atender grandes grupos *multicast*, o RTCP tem de evitar que o tráfego de controle esgote os recursos da rede. O RTP limita o tráfego de controle a, no máximo, 5% do tráfego total da sessão. Isto é reforçado para ajustar a taxa de geração do RTCP de acordo com o número de participantes.

7.2 Pacotes RTCP

O RTCP é o protocolo de controle projetado para trabalhar em conjunto com o RTP. Ele está padronizado nas RFC 1889 e 1890. Em uma sessão RTP, os participantes periodicamente enviam pacotes RTCP para obter o feedback sobre a qualidade de entrega dos dados e informação de grupos. A RFC 1889 define cinco tipos de pacotes para transportar informações de controle. Estes cinco tipos são [SCH96a]:

- **RR**: relatório do receptor. Os relatórios do receptor são gerados pelos participantes que não são transmissores ativos. Eles contêm o *feedback* sobre a qualidade de recepção sobre a entrega dos dados, incluindo o maior número de pacotes recebidos, o número de pacotes perdidos, o *jitter* entre a chegada dos pacotes e marcadores de tempo para calcular a latência de *round-trip* entre o transmissor e o receptor.
- **SR**: relatório do transmissor. Os relatórios do transmissor são gerados por transmissores ativos. Somado ao *feedback* sobre a qualidade de recepção como no **RR**, eles contêm uma seção de informação do transmissor, fornecendo informações na sincronização entre mídias, contadores de pacotes acumulativos, e número de pacotes enviados.
- **SDES**: itens de descrição da fonte. Eles contêm informações para descrever as fontes de transmissão.
- **BYE**: indica o fim da participação.
- **APP**: funções específicas da aplicação. Projetado para uso experimental enquanto novas aplicações e características são desenvolvidas.

Por questões de complexidade, os formatos dos tipos de pacotes RTCP não serão exibidos neste trabalho.

8. RTSP – Protocolo de Streaming de Tempo-real

Ao invés de armazenar grandes arquivos multimídia e exibí-los, dados multimídia são geralmente enviados através da rede em fluxos (*streams*). O *streaming* quebra os dados em pacotes com tamanho adequado para a transmissão entre os servidores e os clientes. Os dados de tempo-real fluem através da transmissão, são descomprimidos sob a forma de um pipeline. Um cliente pode exibir o primeiro pacote, descomprimir o segundo, enquanto está recebendo o terceiro. Portanto, o usuário pode começar a curtir a multimídia sem esperar pelo fim da transmissão.

O RTSP, protocolo de streaming de tempo-real, é um protocolo de apresentação multimídia cliente-servidor para permitir a entrega controlada de dados multimídia em fluxo sobre redes IP. Ele fornece a funcionalidade de um controle remoto, semelhante ao de um vídeo-cassete, para fluxos de áudio e vídeo, como PAUSE, FF, REW e posicionamento absoluto. Fontes de dados incluem dados ao vivo e clips armazenados.

O RTSP – RFC 2326 – é um protocolo ao nível de aplicação projetado para trabalhar com protocolos das camadas inferiores como RTP e RSVP para fornecer um completo serviço de streaming sobre a Internet. Ele fornece meios para escolha de canais de entrega, tais como UDP, multicast UDP e TCP, e mecanismos de entrega baseados no RTP. Ele funciona para grandes audiências multicast assim como para um único cliente unicast.

Não há nenhuma noção a uma conexão RTSP; ao invés disso, um servidor mantém uma sessão rotulada por um identificador. Uma sessão RTSP não está de nenhuma forma atrelada a uma conexão ao nível de transporte, como uma conexão TCP. Durante uma sessão RTSP, um cliente RTSP pode abrir e fechar muitas conexões de transporte confiáveis ao servidor para emitir suas solicitações RTSP. Alternativamente, pode-se usar um protocolo de transporte não-orientado à conexão, como o UDP. Os fluxos controlados pelo RTSP podem também usar o RTP.

8.1 Operações do RTSP

O RTSP estabelece e controla fluxos de áudio e vídeo contínuos entre os servidores de mídia e os clientes. Um servidor de mídia fornece serviços de gravação e exibição dos fluxos de mídia enquanto um cliente solicita dados de mídia contínuos do servidor de mídia. O RTSP é o controle remoto de rede entre o servidor e o cliente. Ele fornece as seguintes operações:

- Recuperação de mídia a partir de um servidor de mídia. O cliente pode solicitar uma descrição da apresentação, e pedir ao servidor para configurar uma sessão para enviar os dados solicitados.
- Convite de um servidor de mídia para uma conferência: o servidor de mídia pode ser convidado para a conferência para exibir mídias ou gravar uma apresentação.
- Adição de mídias a uma apresentação existente. O servidor ou o cliente pode notificar um ao outro sobre quaisquer mídias adicionais tornando-se disponíveis.

8.2 Métodos do RTSP

O RTSP visa fornecer os mesmos serviços ao streaming de áudio e vídeo como o HTTP faz com texto e gráficos. Ele é projetado intencionalmente para ter sintaxe e operações

similares de tal forma que a maioria dos mecanismos de extensão para o HTTP pode ser adicionada ao RTSP.

No RTSP, cada apresentação e fluxo de mídia são identificados por uma URL RTSP. A apresentação geral e as propriedades das mídias são definidas em um arquivo de descrição da apresentação, que pode incluir a codificação, idioma, URLs RTSP, endereço de destino, porta, e outros parâmetros. O arquivo de descrição da apresentação pode ser obtido pelo cliente usando HTTP, e-mail ou outros meios.

Na versão atual, os serviços e operações são suportados através dos seguintes métodos:

- **OPTIONS:** o cliente ou o servidor fala à outra parte as opções que pode aceitar.
- **DESCRIBE:** o cliente recupera a descrição de uma apresentação ou objeto de mídia identificado por uma URL de solicitação do servidor.
- **ANNOUNCE:** quando enviado do cliente para o servidor, ANNOUNCE torna conhecida a descrição de uma apresentação ou objeto de mídia identificado por uma URL de solicitação para um servidor. Quando enviado do servidor para o cliente, ANNOUNCE atualiza a descrição da sessão em tempo-real.
- **SETUP:** o cliente pede ao servidor para alocar recursos para um fluxo e inicia uma sessão RTSP.
- **PLAY:** o cliente pede ao servidor para iniciar o envio dos dados em um fluxo alocado via SETUP.
- **PAUSE:** o cliente temporariamente pára a entrega do fluxo sem liberar os recursos do servidor.
- **TEARDOWN:** o cliente pede ao servidor para parar a entrega de um fluxo específico e liberar os recursos associados com ele.
- **GET_PARAMETER:** recupera o valor de um parâmetro para uma apresentação ou fluxo especificados na URL.
- **SET_PARAMETER:** configura o valor de um parâmetro para uma apresentação ou fluxo especificados por uma URL.
- **REDIRECT:** o servidor informa aos clientes que ele deve conectar-se a um outro servidor. O cabeçalho de localização obrigatório indica a URL na qual o cliente deve se conectar.
- **RECORD:** o cliente inicia a gravação de uma variedade de dados de mídia de acordo com a descrição da apresentação.

Alguns destes métodos podem ser enviados tanto do servidor para o cliente quanto do cliente para o servidor, mas podem somente ser enviados em uma direção. Nem todos os métodos são necessários em um servidor totalmente funcional. Por exemplo, um servidor de mídia com dados ao vivo pode não suportar o método *PAUSE*.

As solicitações RTSP são geralmente enviadas em um canal independente do canal de dados. Elas podem ser transmitidas em conexões de transporte persistentes, ou como uma conexão por transação de solicitação/resposta, ou no modo não-orientado à conexão.

8.3 Relacionamento com outros Protocolos

O RTSP tem algumas características comuns em termos de funcionalidade com o HTTP. Ele pode também interagir com o HTTP naquele contato inicial com o conteúdo de streaming que, na maioria das vezes, é feito através de uma página Web [SCH96b].

Entretanto, o RTSP difere do HTTP quanto à entrega dos dados. O HTTP é um protocolo assimétrico, onde o cliente emite suas solicitações e o servidor as responde. No RTSP, ambos os clientes e servidores podem emitir solicitações. As solicitações RTSP não são também stateless; elas podem configurar parâmetros e continuar a controlar um fluxo de mídia por muito tempo depois da solicitação ter sido confirmada.

Reutilizar a funcionalidade do HTTP tem vantagens em, no mínimo, duas áreas: segurança e proxies. O RTSP reutiliza os mecanismos de segurança na Web. Todos os mecanismos de autenticação HTTP são diretamente aplicáveis. O RTSP pode também reutilizar os mecanismos de segurança da camada de rede ou transporte.

Enquanto que a maioria das mídias de tempo-real usará o RTP como protocolo de transporte, o RTSP não está atrelado ao RTP. O RTSP assume a existência de um formato de descrição da apresentação que pode expressar tanto propriedades estáticas quanto temporais de uma apresentação contendo vários fluxos de mídia.

8.4 Situações do Uso do RTSP

Mídia sob Demanda unicast

O cliente C solicita um filme dos servidores de mídia A (audio.example.com) e V (video.example.com). A descrição da mídia está armazenada em um servidor Web W. A descrição da mídia contém descrições da apresentação e todos seus fluxos, incluindo os codificadores/decodificadores que estão disponíveis, tipos de dados RTP dinâmicos, a pilha de protocolos, e informação de conteúdo, tais como idioma ou restrições de copyright. Ela pode também dar uma indicação sobre a linha de tempo do filme. Neste exemplo, o cliente está somente interessado na última parte do filme [SCH96b].

CLIENT → WEB SERVER: GET /twister.sdp HTTP/1.1

Host: www.example.com

Accept: application/sdp

WEB SERVER → CLIENT: HTTP/1.0 200 OK

Content-Type: application/sdp

v=0 o=- 2890844526 2890842807 IN IP4 192.16.24.202

s=RTSP Session m=audio 0 RTP/AVP 0

a=control:rtsp://audio.example.com/twister/audio.en

m=video 0 RTP/AVP 31

a=control:rtsp://video.example.com/twister/video

CLIENT → AUDIO SERVER: SETUP rtsp://audio.example.com/twister/audio.en RTSP/1.0

CSeq: 1

Transport: RTP/AVP/UDP;unicast;client_port=3056-3057

AUDIO SERVER → CLIENT: RTSP/1.0 200 OK

CSeq: 1

Session: 12345678

Transport: RTP/AVP/UDP;unicast;client_port=3056-3057;
server_port=5000-5001

CLIENT → VIDEO SERVER: SETUP rtsp://video.example.com/twister/video RTSP/1.0

CSeq: 1

Transport: RTP/AVP/UDP;unicast;client_port=3058-3059

VIDEO SERVER → CLIENT: RTSP/1.0 200 OK

CSeq: 1

Session: 23456789

Transport: RTP/AVP/UDP;unicast;client_port=3058-3059;
server_port=5002-5003

CLIENT → VIDEO SERVER: PLAY rtsp://video.example.com/twister/video RTSP/1.0

CSeq: 2

Session: 23456789

Range: smpte=0:10:00-

VIDEO SERVER → CLIENT: RTSP/1.0 200 OK

CSeq: 2

Session: 23456789

Range: smpte=0:10:00-0:20:00

RTP-Info: url=rtsp://video.example.com/twister/video;
seq=12312232;rtptime=78712811

CLIENT → AUDIO SERVER: PLAY rtsp://audio.example.com/twister/audio.en RTSP/1.0

CSeq: 2

Session: 12345678

Range: smpte=0:10:00-

AUDIO SERVER → CLIENT: RTSP/1.0 200 OK

CSeq: 2

Session: 12345678

Range: smpte=0:10:00-0:20:00

RTP-Info: url=rtsp://audio.example.com/twister/audio.en;

seq=876655;rtptime=1032181

CLIENT → AUDIO SERVER: TEARDOWN rtsp://audio.example.com/twister/audio.en RTSP/1.0

CSeq: 3

Session: 12345678

AUDIO SERVER → CLIENT: RTSP/1.0 200 OK

CSeq: 3

CLIENT → VIDEO SERVER: TEARDOWN rtsp://video.example.com/twister/video RTSP/1.0

CSeq: 3

Session: 23456789

VIDEO SERVER → CLIENT: RTSP/1.0 200 OK

CSeq: 3

Embora o áudio e o vídeo estejam em servidores diferentes, e possam começar em momentos bem próximos e possam mudar vagarosamente com respeito ao outro, o cliente pode sincronizar os dois usando métodos RTP, em particular a escala de tempo contida nos relatórios do transmissor RTCP.

Apresentação de mídia ao vivo usando multicast

O servidor de mídia M escolhe o endereço e a porta multicast. Aqui, assume-se que o servidor Web W contém somente um ponteiro para a descrição completa, enquanto o servidor de mídia M mantém a descrição completa [SCH96b].

CLIENT → WEB SERVER: GET /concert.sdp HTTP/1.1

Host: www.example.com

WEB SERVER → CLIENT: HTTP/1.1 200 OK

Content-Type: application/x-rtsl

<session><track src="rtsp://live.example.com/concert/audio"></session>

CLIENT → MEDIA SERVER: DESCRIBE rtsp://live.example.com/concert/audio RTSP/1.0

CSeq: 1

MEDIA SERVER → CLIENT: RTSP/1.0 200 OK

CSeq: 1

Content-Type: application/sdp

Content-Length: 44

v=0 o=- 2890844526 2890842807 IN IP4 192.16.24.202

s=RTSP Session m=audio 3456 RTP/AVP 0

a=control:rtsp://live.example.com/concert/audio

c=IN IP4 224.2.0.1/16

CLIENT → MEDIA SERVER: SETUP rtsp://live.example.com/concert/audio RTSP/1.0

CSeq: 2

Transport: RTP/AVP;multicast

MEDIA SERVER → CLIENT: RTSP/1.0 200 OK

CSeq: 2

Transport: RTP/AVP;multicast;destination=224.2.0.1;

port=3456-3457;tll=16

Session: 0456804596

CLIENT → MEDIA SERVER: PLAY rtsp://live.example.com/concert/audio RTSP/1.0

CSeq: 3

Session: 0456804596

MEDIA SERVER → CLIENT: RTSP/1.0 200 OK

CSeq: 3

Session: 0456804596

9. Conclusões

O projeto de redes multimídia deve atender às necessidades em termos de QoS dos dados de características de tempo-real. A Internet é tida, atualmente, como a principal plataforma para desenvolvimento de aplicações. Transferir dados multimídia sobre IP, embora seja um desafio, vem gerando muitos estudos para este intuito.

A técnica de *multicasting* traz importantes benefícios para as redes de computadores, como: economia no processamento de pacotes e no consumo de largura de banda, permitindo maior número de *hosts* participantes.

O protocolo RSVP tem papel importante como agente preventivo quanto a existência de recursos para as aplicações, em face do dinamismo que, principalmente, a Internet apresenta com relação a mudanças de rotas, congestionamentos, disponibilidade imprevisível de recursos, etc.

O RTP fornece serviços de entrega fim-a-fim para dados de características de tempo-real, como áudio e vídeo interativos, porém não fornecendo qualquer mecanismo de garantir a entrega na hora certa, necessitando do suporte das camadas inferiores que têm controle sobre os recursos em comutadores e roteadores. O RTP depende do RSVP para reservar os recursos e fornecer a QoS solicitados.

O RTCP é o protocolo de controle projetado para trabalhar em conjunto com o RTP. Em uma sessão RTP, os participantes periodicamente enviam pacotes RTCP para obter o feedback sobre a qualidade de entrega dos dados e informação de grupos.

O RTSP é um protocolo ao nível de aplicação projetado para trabalhar com protocolos das camadas inferiores como RTP e RSVP para fornecer um completo serviço de streaming sobre IP. Ele fornece meios para escolha de canais de entrega, como UDP, multicast UDP e TCP, e mecanismos de entrega baseados no RTP, tanto para grandes audiências multicast, como para um único cliente unicast.

Bibliografia

- [BRA97] BRADEN, R. et al. "*Resource ReSerVation Protocol (RSVP)*".
<http://www.ietf.org/rfc/rfc2205.txt>. 1997
- [IPM95] IP MULTICAST INITIATIVE (IPMI). "*IP Multicast Backgrounder*".
<http://www.ipmulticast.com/community/whitepapers/backgrounder.html>. 1995.
- [LIU98] LIU, C. *Multimedia Over IP: RSVP, RTP, RTCP, RTSP*. http://www.cis.ohio-state.edu/~jain/cis788-97/ip_multimedia/index.htm. 1998.
- [QUI98] QUINN, B. *IP Multicast Applications: Challenges and Solutions*.
<http://www.ipmulticast.com/techcentral/draft-quinn-multicast-apps-00.txt>. 1998.
- [SCH96a] SCHULZRINNE, H. et al. *RTP: A Transport Protocol for Real-Time Applications*.
<http://www.cis.ohio-state.edu/htbin/rfc/rfc1889.html>. 1996.
- [SCH96b] SCHULZRINNE, H. et al. *Real Time Streaming Protocol (RTSP)*.
<http://www.ietf.org/rfc/rfc2326.txt>. 1996.
- [SOU99] SOUSA, M. et al. *Protocolos de Reserva de Largura de Banda*. Trabalho da disciplina Redes de Alto Desempenho. ICMC-USP. 1999.
- [STA98] STALLINGS, W. *High Speed Networks: TCP/IP and ATM Networks*. Prentice Hall. 1998.