CAPÍTULO 1 - INTRODUÇÃO

1.1 Apresentação:

As Redes de Computadores foram desenvolvidas, inicialmente, para suportarem apenas tráfegos textuais. Com o avanço das tecnologias e a crescente demanda do mercado, a multimídia tem se tornado parte essencial dessa rede. Áudio, vídeo, animações, conferências tornam-se cada vez mais comuns e necessárias.

Para que esse novo enfoque seja suportado pela Rede, algumas adaptações devem ser feitas e consequentemente alguns problemas devem ser superados. Um dos principais problemas relacionados a essa mudança é o tipo de tráfego, pois tráfegos multimídia exigem muito mais da rede que tráfegos textuais. Os dados são maiores, exigindo maiores bandas de transmissão, buffers, controles de fluxos, entre outros.

Além desse problema, aplicações multimídia apresentam tráfegos com características de tempo real, onde os dados de áudio e vídeo devem ser entregues continuamente na taxa em que forem produzidos. Como a rede não estava acostumada a tratar tráfegos diferenciadamente, e que não podem sofrer *jitter* ou perda alguma, inteligência deve ser acrescentada a ela.

Contribuindo para o aumento das dificuldades relacionadas aos novos tráfegos uma nova rede está surgindo, a rede sem fio (*wireless*), e com ela, novos desafios. Além de solucionar os problemas existentes na rede fixa é necessário adaptá-los para redes móveis.

1.2 Estrutura do documento:

Esse documento tem por finalidade mostrar os problemas e as possíveis soluções ligadas a esse novo tráfego que surgiu na rede. Primeiramente descreve a definição de Qualidade de Serviço, seguido dos protocolos que dão suporte a ela. Para cada protocolo é apresentado suas características, suas funções e como solucionam as necessidades de QoS, além disso caso o protocolo possua alguma alteração que ainda esteja sendo discutida pelos órgão responsáveis (documentos denominados *draft's*), elas serão apresentadas. Nos capítulos seguintes são definidas as QoS para Redes IEEE e para as novas redes sem fio. Por fim um breve comentário sobre o estado da arte do QoS.

CAPÍTULO 2 - QUALIDADE DE SERVIÇO

2.1 O que é Qualidade de Serviço:

Qualidade de Serviço ou QoS é a habilidade que a rede possui de fornecer serviços qualitativamente e quantitativamente melhores a um determinado tráfego na rede.

Para melhor definir o conceito de Qualidade de Serviço é preciso olhar um pouco para a história da Internet e suas características:

No início, a Internet estava voltada apenas para tráfegos textuais, onde o nível de serviço utilizado, o "*best effort*", era suficiente para atender as necessidades da rede. O "*best effort*" é um modelo de serviço onde os dados são tratados com equidade e enviados em qualquer quantidade, sem pedir permissão ou sequer informar à rede. Esta por sua vez, entrega esses dados com o "maior esforço" possível, porém, sem a menor garantia de atraso, erro ou até mesmo de entrega. Conforme descrito no Capítulo 1, houve uma mudança nas características dos dados que trafegam na Internet e novas necessidades surgiram.

As arquiteturas da Rede também sofreram grande influência com o crescimento da Internet e uma diversificação nas tecnologias que a constitui pode ser observada. As tecnologias LAN e WAN, baseadas no protocolo IP, conectam redes cada vez maiores à Internet, sendo elas as responsáveis pelo desenvolvimento de grande parte das atividades na área de redes. Novos equipamentos de hardware também estão surgindo, como redes ATM, *Gigabit Ethernet*, FDDI, que oferecem altas taxas de transmissão, e com eles a necessidade de integrar essas tecnologias.

Justamente para que fosse possível controlar estes diferentes tipos de tráfegos, como áudio, vídeo e animações, integrando diversas tecnologias como frame relay, ATM, Ethernet e redes 802.1, SONET e redes com roteamento IP surgiu a noção de Qualidade de Serviço.

2.2 Serviços de QoS:

A Qualidade de Serviço atua na rede, permitindo que se estabeleça um controle sobre o tráfego fim-a-fim na mesma, fornecendo serviços melhores e mais previsíveis por:

- Suportar largura de banda (*bandwidth*) dedicada.
- Melhorar perdas características
- Evitar e gerenciar congestionamentos na rede.
- Dar forma ao tráfego da rede.
- Taxar prioridades aos tráfegos pela rede.
- Taxar e mapear os requisitos de acordo com as preferências do usuário.

QoS pode atuar de diversas formas dependendo da rede/aplicação, pois nem todas as técnicas de QoS são apropriadas para todos os tipos de rede/aplicação, podendo inclusive ser alterada durante uma sessão. Por exemplo, o ISP (Internet Service Provider) requer escalonamento e performance segura, ele oferecia serviço *best-effort* para seu tráfego, com a transferência de voz, vídeo, e outros aplicativos *real-time*, QoS responde as necessidades do ISP distinguindo esses diversos tipos de tráfegos, permitindo assim, que estes ofereçam serviços diferenciado para seus usuários. Por isso, existem alguns parâmetros pré-definidos de QoS divididos em cinco categorias:

- Orientado a performance: atraso fim-a-fim e taxa binária;
- Orientado a formato: resolução de vídeo, taxa de quadros e tipo de compressão;
- Orientado a sincronização: desvio entre o começo de seqüências de áudio e vídeo;
- Orientado a custo: mudanças na conexão e transmissão de dados;
- Orientado a usuário: imagem subjetiva e qualidade de som.

2.3 QoS na Rede:

A Internet trata-se de uma rede orientada a datagrama, ou seja, sem conexão, com meio compartilhado. Dessa forma a Internet **não** é naturalmente adequada para transferência de tráfegos de tempo-real.

Para que seja possível adaptar a Internet tornando-a apta à atender aos requisitos dos tráfegos multimídia, alguma inteligência deve ser adicionada à rede. Os protocolos de suporte a QoS permite que isso ocorra, através da resolução de determinados pontos na transferência de dados, como:

- Banda de transmissão, para suportar o tráfego multimídia que é muito pesado.
- Tráfegos *multicast*, onde há a necessidade de enviar um mesmo *stream* de dados para um grupo na internet. Os protocolos devem reduzir o tráfego.
- Recursos, as aplicações devem ser capazes de reservar recursos através dos protocolos
- Congestionamento, os protocolos de transporte devem ser utilizados de forma a garantir a entrega de dados de áudio e vídeo para que os mesmos possam ser apresentados de forma contínua e sincronizada.
- Para que as aplicações gerenciem as entregas dos dados multimídia operações sobre as mesmas devem ser fornecidas.

CAPÍTULO 3 - PROTOCOLOS DE SUPORTE A QOS

Para responder aos requisitos estipulados pela QoS, alguns protocolos foram criados. O desafio desses protocolos é empregar uma variedade de serviços para o controle de acesso a links compartilhados, buffers e recursos de processamento. Esses mecanismos incluem *traffic shapping*, controle de fluxo, *link schedule*, gerência de buffer, enfim meios para coordenar recursos compartilhados no nível de rede. Complementando esse mecanismo de nível baixo, protocolos de roteamento e sinalização controlam o dinamismo da rede redirecionando o trafego a nível de fluxo ou conexão. Rotinas de QoS selecionam o caminho para cada fluxo ou conexão para satisfazer diversos requisitos de performance e otimização de recursos. Entretanto para suportar altos throughput e baixos retardos quando se estabelece a conexão em grandes redes, o esquema de caminho selecionado não deve consumir excessiva largura de banda, memória e recursos de processamento renovado baseado em mudança quando disponível.

Neste capítulo são descritos alguns desses protocolos e futuras alterações nos mesmos que se encontram em forma de draft's na Internet.

3.1 RSVP

O Protocolo de Reserva de Recursos ou RSVP (*Resource ReserVation Protocol*), definido na [RFC2205], é um serviço IP que permite tráfegos de tempo real como voz ou vídeo, "economizar" recursos (como largura de banda, buffers, etc.) impedindo assim, que estes ultrapassem os seus limites. RSVP é usado pelo *host* para requisitar determinadas QoS da rede para uma aplicação de fluxo de dados. Os roteadores também utilizam RSVP, para entregar requisições de QoS para todos os nós ao longo do caminho do fluxo e para que se estabeleça e mantenha o serviço requisitado, em geral, a reserva de recursos.

Para ser possível obter a reserva de recurso é preciso que as estações finais suportem o RSVP e que os roteadores intermediários que não o implementam, e logo não são capazes de realizar a reserva, sejam capazes de fornecer um serviço proveitoso para as aplicações de tempo real.

O RSVP foi criado para ser um protocolo de sinalização para que, através dele, as aplicações pudessem realizar a reserva de recurso. Ele tem os seguintes atributos:

- RSVP é designado para operar com aplicações *multicast*, mas também suporta reserva de recursos para aplicações *unicast* assim como transmissões ponto-a-ponto. Ele se adapta dinamicamente as mudanças de membros de um mesmo grupo assim como a mudança de rota.

- RSVP é um protocolo *simplex*, ou seja, ele reserva recursos em apenas uma direção. Portanto, RSVP distingue transmissor do receptor, apesar de um mesmo aplicativo agir como os dois ao mesmo tempo.

- RSVP é orientado ao receptor, ou seja, o receptor inicia e mantém a reserva de recursos usada no fluxo de dados. Isso porque para acomodar eficientemente grandes grupos, membros de grupos dinâmicos e requisições de receptores heterogêneos, o RSVP fez dos receptores os responsáveis pela requisição das reservas de recurso. Esse processo funciona da seguinte maneira: o receptor faz a requisição de uma reserva e passa para o processo RSVP local. O protocolo RSVP leva a requisição para todos os nós (roteadores e hosts) ao longo do caminho até a fonte de dados. Como resultado o *overhead* da reserva do RSVP é logaritmicamente melhor que a linear em relação ao número de receptores.

- A utilização de *soft-state* nos roteadores e hosts, fornece um suporte maior para as mudanças ocorridas entre membros de um grupo e para adaptações automáticas às alterações de roteamento que os pacotes de dados possam sofrer.

- O RSVP não implementa o algoritmo de roteamento, mas depende dos atuais e futuros protocolos de roteamento.

- RSVP transporta e mantém parâmetros de controle de tráfego e de policiamento que são opacos ao RSVP.

- Fornece vários modelos de reserva ou "*styles*", para englobar uma variedade de aplicativos.

- Fornece operações transparentes pelos roteadores que não suportem RSVP.

- RSVP opera sobre o Ipv4 ou o Ipv6, ocupando um espaço de um protocolo de transporte na pilha de protocolos.

3.1.1 Serviços do RSVP:

- Fluxo de dados:

Existe o *stream* de dados que opera em modo *simplex*, indo de uma origem para múltiplos destinos, e o fluxo de dados que vai de um conjunto de transmissores para um conjunto de receptores, em transmissões *multicast*. Para transmissões *unicast* irá existir um único destinatário, porém, múltiplos transmissores, o RSVP é capaz de reservar recursos para transmissões multiplos-pontos-para-único-ponto (*multipoint-to-single-point*).

O RSVP define uma sessão para ser um fluxo de dados, contendo um destino particular e um protocolo de nível de transporte. Essa sessão é definida por três parâmetros (*DestAdress*, *ProtocolID*, [,*DstPort*]). Sendo *DestAdress* o endereço IP do pacote, podendo ser unicast ou multicast, *ProtocolID* o identificador do protocolo IP e *DstPort* um parâmetro que generaliza o destino, ou seja, *DstPort* pode ser definido como campo de destino do UDP/TCP, como um campo equivalente em qualquer outro protocolo ou por alguma informação especifica do aplicativo.

Na figura 3.1.1.a está um exemplo de um fluxo de dados um uma única sessão RSVP, assumindo o modo mulicast de distribuição.

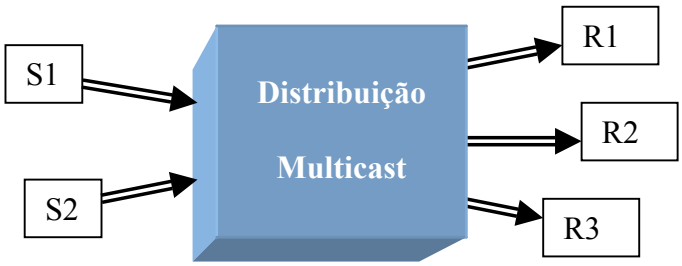

Figura 3.1.1.a - Sessão *multicast*, Si's são os transmissores e Ri's os receptores

- Modelo de Reserva:

Uma simples requisição de reserva RSVP consiste em "*flowspec*" junto com "*filter spec*", esse par é chamado de descritor de fluxo. O "*flowspec*" especifica a QoS desejada. O "*filter spec*", junto com a especificação da sessão, definem o conjunto de pacote de dados que irá receber a QoS requisitada pelo "*flowspec*". O "*flospec*" é usado para setar parâmetros nos nós do caminho ou em outro mecanismo a nível de link, enquanto o "*filter spec*" é usado para setar parâmetros no packet classifier. Os demais pacotes que não aparecem no *filter specs* são tratados como tráfegos best-effort.

- Estilos de Reserva:

Uma requisição de reserva inclui um conjunto de opções chamados de "*Reservation Style*".

Uma opção de reserva consiste no tratamento de reservas para diferentes transmissores com uma mesma sessão: estabelecendo uma reserva "distinct" para cada transmissor ou um reserva "shared" que é compartilhada pelos pacotes de um grupo de transmissores.

Outra opção de reserva controla a seleção de transmissores, podendo ser "explicit", uma lista contendo todos os transmissores selecionados, ou um "wildcard" que implicitamente seleciona todos os transmissores para uma sessão. Na opção "explicit" cada *filter spec* precisa coincidir com um transmissor, enquanto na "wildcard" não existe a necessidade do *filter spec*.

Combinando as opções acima descritas temos três tipos de estilos de reserva, são eles: **Fixed Filter (distinct) Style**, onde o fluxo é originado de apenas um transmissor (como por exemplo nas aplicações de vídeo), ele requer uma reserva separada por transmissor em cada tipo de transmissão, **Shared Explicit**, onde um fluxo de reserva compartilhado é originado de um limitado número de transmissores (uma aplicação de áudio por exemplo) e onde uma simples reserva pode ser aplicada para todos os transmissores do conjunto, e por último o **WildCard Filter** um fluxo de reserva compartilhado originado de todos os transmissores.

Seleção do transmissor	Reservas	
	Distinct	Shared
Explicit	Fixed Filtered (FF) *Style*	Shared-Explicit (SE) *Style*
WildCard	Não definida	WildCard-Filtered (WF) *Style*

Figura 3.1.1b : Atributos e estilos de reserva.

3.1.2 Mecanismo de funcionamento

Mensagens :

Existem dois tipos de mensagens fundamentais no RSVP: RESV e PATH.

A mensagem PATH contém o *flow spec* e as características do tráfego, enquanto na mensagem RESV estão os parâmetros de reserva, o *flow spec* e o *filter spec*

Essas mensagens funcionam da seguinte forma: Os transmissores enviam a mensagem PATH para o endereço destino. Os roteadores intermediários, que implementam o RSVP, localizados ao longo do caminho até o destino, retransmitem a mensagem PATH ao próximo nó até chegar ao receptor, que responde com uma mensagem RESV. Essa mensagem percorre o caminho reverso da mensagem PATH reservando recursos nos nós existentes, se esses nós aceitarem a reserva, banda de transmissão e espaço em *buffers* são alocados e o *flow state* é armazenado em cada um desses nós. Então os transmissores enviam os dados. Esse procedimento pode ser verificado na figura 3.1.2a.

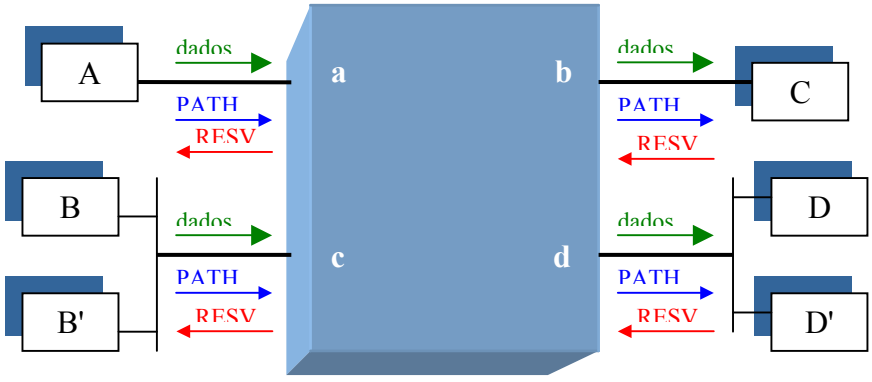

Figura 3.1.2.a - Sinalização do RSVP.

Soft State

O RSVP utiliza o *Soft State* para gerenciar a reserva de recursos nos roteadores e nos hosts. O Soft State é criado e atualizado periodicamente pelas mensagens PATH (gera o Path State) e RESV (reserva recurso), caso a mensagem de atualização não chegue em um determinado espaço de tempo ("cleanup timeout") o estado é deletado. Toda vez que uma nova rota é criada a próxima mensagem PATH inicializa o Path State e a mensagem RESV estabelece as novas reservas de recurso.

O problema do *soft-state* está no *overhead* gerado por essas mensagens que devem ser enviadas periodicamente. Alguns estudos tem sido feito sobre esse assunto e algumas draft's estão sendo analisadas na tentativa de solucionar esse problema.

Teardown

A mensagem Teardown remove o Path state e a reserva de recurso imediatamente. Embora não seja necessário, a mensagem Teardown deve ser enviada assim que a aplicação finalizar a transmissão.

Existem dois tipos de mensagens teardown: PathTear e a ResvTear, que viajam em direções contrária limpando respectivamente o path state e as reservas de recurso nos nós da rede, ao longo do caminho utilizado durante o processo de transmissão.

Mensagens de Erro

Existem basicamente duas mensagens de erro: ResvErr e a PathErr. A PathErr é relativamente simples, sendo enviada apenas para o transmissor que a gerou, não alterando o *path state* nos nós por onde passou. Já a ResvErr é mais complexa pois a requisição da reserva que foi negada pode ter sido resultado da combinação de várias requisições, com isso a ResvErr deve ser enviada a todos os receptores responsáveis. Para complicar ainda mais ela pode causar o que chamamos de "*Killer reservation*", gerado quando da combinação de diversas requisições se uma é negada as outras também são.

Mensagem de Confirmação

O receptor determina se deseja receber uma mensagem de confirmação inserindo na mensagem RESV a requisição de confirmação ResvConf. Com isso se o nó possui um espaço menor ou igual ao da requisição desejada a ResvConf é transmitida de volta imediatamente, enquanto se não existisse a necessidade de confirmação a mensagem RESV apenas não seria transmitida adiante.

Policy Control

O termo "Policy control" é usado para suportar acessos policiados durante as reservas RSVP. Na Internet cada domínio administrativo tem um policiamento diferente em relação as requisições, para isso o RSVP define um parâmetro chamado *Policy_data* que é capaz de carregar informações que permitirão aos módulos de policiamento locais decidir aceitar ou rejeitar a reserva.

O Formato do *Policy_data* não está definido na [RFC2205], na realidade nessa versão é deixado um "espaço" para um futuro suporte ao policiamento, seu formato é descrito na [RFC2750] (janeiro de 2000). O objeto *Policy_data* é carregado pelas mensagens RSVP e todos os nós que fazem o policiamento devem ser capazes de gerar, modificar ou remover esse objeto, mesmo que os receptores e os transmissores não o forneça. Os mecanismos de controle do policiamento são descritos nos protocolos RAP, COPS e COPS-RSVP, que estão definidos mais adiante.

Basicamente, o funcionamento de uma reserva em um nó ocorre da seguinte forma: quando a requisição de uma reserva chega ao nó, o RSVP *deamon* comunica-se com dois módulos locais de decisão, *admission control* e o *policy control*. O primeiro determina se o nó tem recursos disponíveis para que se possa fazer a reserva e o segundo determina se o pacote tem permissão administrativa para tal. Se ambas as verificações forem negadas, uma mensagem de erro é retornada para o processo que a gerou, caso contrário o RSVP *deamon* fixa parâmetros no *packet classifier* (classificador de pacotes) e no *packed scheduler* (escalonador de pacotes), esse último determina se o pacote deve ou não ser transmitido. Esse mecanismo utilizado para obter a QoS desejada é chamado de controle de tráfego ("traffic control").

Cabeçalho das Mensagens

As mensagens RSVP consistem em um cabeçalho fixo, seguido por um objeto. Na figura 3.1.2.b, temos o exemplo do cabeçalho fixo das mensagem RSVP, onde temos os campos: Vers, 4 bits contendo a versão do protocolo, Flags, 4 bits que ainda não foram definidos, Msg Type, 8 bits contendo o tipo da mensagem (Path, Resv, PathErr, etc), RSVP Checksum, 16 bits de controle, Send_TTL, 8 bits para o IP TTL do transmissor e o RSVP Length, 16 bits com o tamanho total da mensagem incluindo o objeto.

0		1	2 3
Vers	Flags	Msg Type	RSVP Checksum
Send_TTL		(Reservado)	RSVP Length

Figura 3.1.2.b - Cabeçalho fixo do RSVP

3.1.3 *Drafts* para o RSVP

Como um protocolo *soft-state*, o RSVP especifica que cada nó RSVP deve enviar periodicamente mensagens de controle para cada sessão RSVP, adicionalmente, cada nó também envia mensagens de atualização para seus vizinhos em todas as sessões existentes. O overhead causado por essas mensagens tem gerado propostas que tentam minimizá-lo.

Na *draft* [DRAFT1] a proposta para minimizar o overhead é através da utilização do mecanismo de "state compression". Esse mecanismo consiste em utilizar uma simples mensagem comprimida, que represente o RSVP state. Para tal, é necessário ter uma tabela que correlacione as mensagens aos RSVP states em todos os nós vizinhos e um algoritmo que mapeie essa relação com baixa probabilidade de colisão. Dois algoritmos são sugeridos o CRC32 e o MD5.

A *draft* [DRAFT2] diz que sendo o RSVP um protocolo fim-a-fim, o transmissor deve receber as mensagens enviadas de volta, o problema é que a mensagem enviada percorre todo o caminho do transmissor ao receptor e deste para o transmissor novamente, como podemos observar na figura 3.1.3.a. Para evitar isso a draft propõe que exista um roteador que seja capaz de gerar mensagens proxy, como resposta às mensagens do transmissor, isso seria feito sobre um controle de policiamento e o protocolo RSVP não seria modificado. Por exemplo, o transmissor envia uma mensagem Path, o roteador proxy que esta no caminho recebe a mensagem e de acordo com o policiamento, ao invés de enviar para o próximo nó do caminho, cria uma mensagem Resv e a envia para o transmissor novamente (figura 3.1.3.a), dessa forma ele impede a perda de tempo com o ciclo da mensagem.

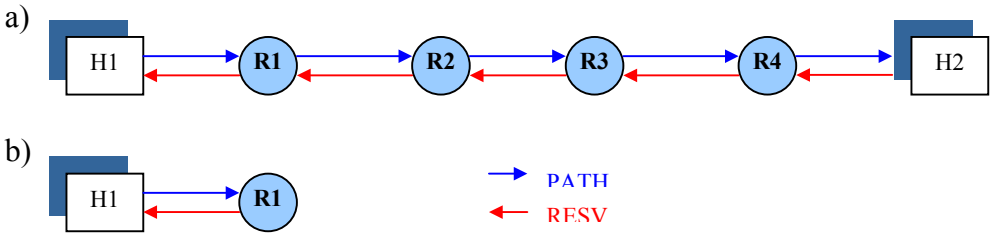

Figura 3.1.3.a - Roteador Proxy.

3.2 RTP e RTCP

O Protocolo de Transporte de Tempo Real, ou RTP (*Real-Time Transport Protocol*) foi desenvolvido junto com a Internet Engineering Task Force (IETF), e está definido na [RFC1889]. Trata-se de um protocolo que provê transporte fim-a-fim na rede para aplicações de tempo real como áudio, vídeo ou dados simulados sobre serviços *multicast* ou *unicast*. Normalmente ele trabalha junto com o UDP ou o TCP. O RTP não endereça reservas de recurso e nem garante a QoS para serviços de tempo real.

Para monitorar a entrega dos dados escalonados em uma grande rede multicast, utilizando pouco controle e identificação de funcionalidade, o transporte de dados possui o RTCP, um protocolo de controle oferecido pelo RTP. Tanto o RTP quanto o RTCP são executados em portas distintas em uma mesma máquina. Isso permite que os dois mantenham mensagens separadas, evitando qualquer confusão em relação ao tratamento de pacotes de dados de controle.

O RTP pode ser utilizado juntamente com o RSVP, nesse caso, cada sessão RTP irá corresponder a duas sessões RSVP, as quais são identificadas pelos endereços dos dados RTP e pelas portas de controle destino.

O RTP não apresenta noção de conexão, ele pode operar tanto sobre um meio orientado a conexão quanto em um não orientado.

3.2.1 Mecanismo de funcionamento do RTP

Idéia Básica

O RTP representa um novo estilo de protocolo que segue os princípios do *Application Level Framing (AFL)* e do *Integrated Layer Processing* propostos por Clark e Tennenhouse.

O AFL define que a aplicação deve quebrar os dados em unidades, denominadas ADUs (*Application data units*), as quais sejam significativas para as mesmas de forma a não dependerem da tecnologia da rede. As aplicações devem ser capazes de processarem cada ADU separadamente e potencialmente fora de ordem, fazendo com que a perda de algumas ADUs não implique no não processamento das demais. Na realidade a estação receptora irá decidir, em caso de perda de unidades, se deve ignorar (aplicações de tempo-real) ou pedir a retransmissão da ADU (transferência de arquivo). Com isso, como veremos adiante, as unidades de dados RTP implementam esse princípio carregando no cabeçalho números de seqüência e *timestamp*.

O *Integrated Layer Processing* é um princípio que defende que cada ADU deve ser processada em um único passo. Seus autores argumentam que muitas das funções da camada de transporte podem ser estruturadas de modo a permitir o uso mais eficiente desse princípio.

Transferência de dados

Um pacote RTP consiste em um cabeçalho fixo (Figura 3.2.1.a), uma lista de fontes com tamanho variável, uma extensão de cabeçalho, além da própria carga, a qual é codificada de acordo com o tipo de payload.

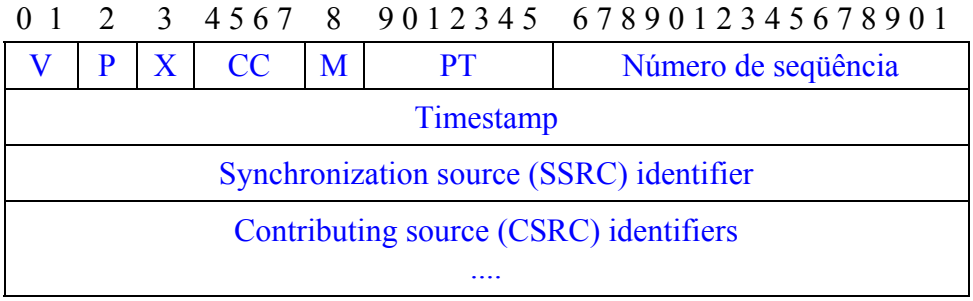

Figura 3.2.1.a: cabeçalho do pacote RTP

O cabeçalho possui os seguintes campos: V (*Versão*), são dois bits que identificam a versão do protocolo RTP que esta sendo usada, P (*padding*), se este bit estiver setado indica que existe um, ou mais de um, byte de padding no final que não faz parte do payload, X (*extensão*), se este bit estiver setado indica que o cabeçalho fixo é seguido exatamente por uma extensão de cabeçalho, CC (*contador de CSRC*), são 4 bits que indicam quantos CSRC's existem no cabeçalho, M (*marcador*), esse bit pode não existir dependendo do modelo utilizado. Sua interpretação está relacionada ao tipo de payload, para vídeos ele indica o final do quadro, já para áudio indica o início de um *talkspurt*, PT (*tipo de Payload*), identifica o tipo do formato de payload existente no pacote, como por exemplo JPEG vídeo ou GSM áudio. Os códigos dos

tipos são fornecidos através de uma tabela que os correlacionam aos seus respectivos *payloads*, Número de seqüência, são 16 bits que permitem ao receptor detectar a perda de pacotes bem como restaurar a seqüência, Timestamp, são 32 bits que descrevem o instante em que os dados foram gerados no pacote. Ele permite a sincronização e o cálculo do atraso entre os pacotes (*Jitter*). A freqüência do timestamp depende do tipo do payload, e é definida previamente nas especificações dos formatos de payload, SSRC, são 32 bits, gerados randomicamente, que identificam o transmissor. Dois transmissores localizados em uma mesma sessão RTP não podem possuir o mesmo SSRC, por último temos o CSRC, que é uma lista numerada de 0 à 15, com itens de 32 bits cada que identificam a contribuição do transmissor para o payload contido no pacote.

Multiplexação de Sessões RTP

O RTP, seguindo o princípio do *Integrated Layer Processing*, minimiza os pontos de multiplexação fazendo do endereço de destino, que define uma sessão RTP, o responsável pela mesma. Com isso, tipos diferentes de payload devem ser encapsulados separadamente, cada um sendo carregado em uma sessão diferente com o seu próprio endereço de destino. Isso pode provocar alguns problemas, como:

- Um RTP *mixer* não conseguirá combinar streams de medias incompatíveis dentro de uma mesmo stream.

- Diferentes reservas de QoS podem ser estabelecidas para cada sessão fazendo com que uma das sessões tenha seu pedido negado e o receptor tenha que implementar diferentes processos, um para cada mídia.

3.2.2 O RTCP

O RTCP é baseado na transmissão periódica de pacotes de controle para todos os participantes em uma sessão, usando o mesmo mecanismo de distribuição dos pacotes de dados. Os protocolos de níveis mais baixos devem fornecer multiplexação para os pacotes de dados e de controle, como por exemplo usando diferentes números de portas com o UDP.

O RTCP realiza as seguintes funções:

- Função de *feedback* na qualidade de distribuição de dados. Essa função permite que tanto as entidades envolvidas na sessão quanto àquelas que só promovem serviços, sejam capazes de monitorar e controlar o congestionamento na rede. O *feedback* é realizado pelos relatórios dos RTCP *sender* e *receiver*.

- RTCP transporta um identificador, chamado de *canonical name* ou CNAME, que permite manter o conhecimento sobre cada um dos participantes. Esse identificador também permite a associação de múltiplos *streams* de dados em um conjunto de sessões, para por exemplo sincronizar áudio e vídeo.

- De acordo com as funções anteriores, os receptores necessitam enviar pacotes RTCP, permitindo que cada um deles identifique e observe os demais.

- Estimativa e escalonamento do tamanho da sessão. A necessidade do controle da informação é balanceada para limitar o trafego de dados principalmente quando uma sessão possui vário membros.

Formato do pacote RTCP

Abaixo seguem vários pacotes RTCP e a informação de controle que eles levam.

- SR (*Sender report*) - gerado por participantes que estão transmitindo, carregam informações sobre sincronização *inter-media*, contadores de pacotes e números de dados enviados.

- RR (*Receiver report*) - retorno gerado por participantes que não estão transmitindo sobre a qualidade da recepção.

- SDES (*Source description items*) - inclui o CNAME, descreve as fontes.

- BYE - indica o fim da participação.

- APP (*Application specific functions*) - uso experimental.

3.2.3 *Draft* para RTP:

Uma proposta ([Draft3]) foi feita em cima do RTP para que este trabalhasse com áudio e vídeo conferência utilizando o mínimo de controle entre as sessões. Em particular, nenhum suporte para negociação de parâmetros ou controle de membros é fornecida. Essa draft espera ser útil em sessões onde a negociação ou controle de membros não são usados e em conjunto com um protocolo de nível mais alto.

Uma outra draft [DRAFT4] faz uma revisão no protocolo RTP esclarecendo alguns pontos funcionais do protocolo.

3.3 COPS

O Protocolo de Policiamento ou *Comom Open Policy Service* (COPS) está definido na [RFC2748] como sendo um simples protocolo que pode ser usado para trocar informações de policiamento, sobre QoS, entre um servidor PDP (*Policy Decision Point*) e seus clientes (Policy Enforcement Points ou PEPs).

As características do COPS incluem:

- O protocolo inclui um modelo cliente servidor onde o PEP envia requisições, updates e deletes para um PDP remoto e este retorna a decisão para o PEP.

- O protocolo usa o TCP como o protocolo de transporte, para troca de mensagens entre os clientes e o servidor.

- O protocolo pode ser estendido, ele foi desenvolvido para suportar diversas informações especificas de clientes sem requerer mudanças no próprio protocolo.

- COPS permite mensagens de nível de segurança para autenticação, replay, proteção e integridade de mensagem. Ele também pode reusar protocolos de segurança existentes.

- Uma vez que o PEP fez uma requisição o PDP pode instalá-la e relembrá-la até ela ser explicitamente deletada pelo PEP. Ao mesmo tempo decisões quanto ao estado da requisição instalada pode ser gerada assincronamente. O servidor também pode responder de forma diferente para cada requisição por causa do estado da requisição/decisão instalada previamente.

- Adicionalmente o protocolo permite que o servidor envie informações de configuração para os clientes e então permite que o servidor remova tal estado do cliente quando este não for mais aplicável.

3.3.1 Mecanismo de funcionamento

Modelo Básico

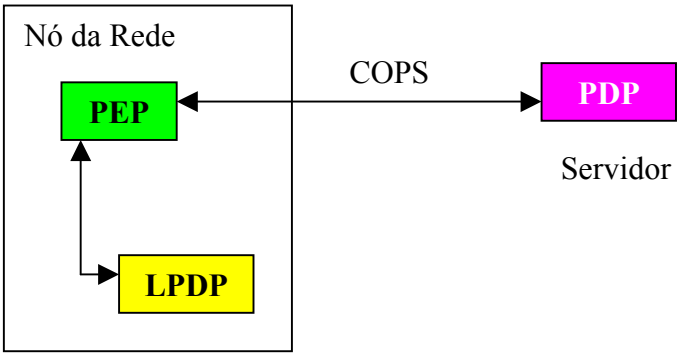

Figura 3.3.1.a - Ilustração do COPS

O modelo ilustrado na figura 3.3.1.a mostra vários componentes, nele COPS é usado para comunicar informações de policiamento entre o PEP e o PDP. O LPDP (*Local Policy Decision Point*) é opcional e pode ser usado para tomar decisões de policiamento locais na falta de um PDP, como por exemplo quando uma requisição é submetida ao PDP um timer é disparado para esperar a resposta, se este estoura o PEP assume que o PDP está fora e então envia para o LPDP.

O PEP é o responsável por iniciar a conexão TCP com o PDP, é através dela que irá ocorrer a troca de mensagens entre os dois. O PEP também é responsável por notificar quando uma mudança de estado é requisitada, e por deletar qualquer estado que não é mais utilizado.

Quando o PEP envia um pedido de configuração para o PDP este lhe envia em forma de mensagens de decisão os dados aplicados às configurações, ao terminar de instalar a configuração o PEP retorna para o PDP uma mensagem confirmando a mesma.

Mensagens

Diferentes tipos de clientes devem ter diferentes dados específicos, que requerem diferentes tipos de decisões de policiamento. Para distinguir cada clientes, o tipo do cliente é identificado em cada mensagem.

O formato do cabeçalho é comum a todas as mensagens, como podemos observar na figura 3.3.1.b, e possui os seguintes campos: Version, 4 bits com a versão do COPS, Flags, 4 bits que deve ser sempre 0 a menos que seja uma mensagem decorrente de uma outra mensagem COPS, OP Code, 8 bits que representam as operações do COPS (onde temos: 1 Requisiçã, 2 decisão, etc), Client-type, 16 bits que identificam o cliente do policiamento e Message Length, 32 bits tamanho da mensagem em bytes incluindo todos os objetos encapsulados.

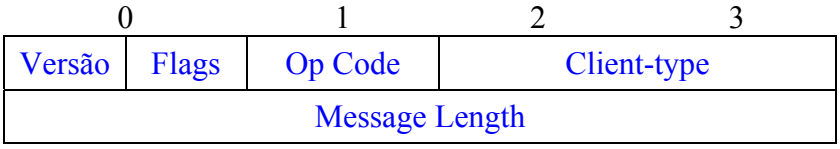

Figura 3.3.1.b - Cabeçalho das mensagens COPS

3.3.2 COPS para RSVP

A combinação do COPS com o RSVP permite, à gerência da rede, centralizar o controle e o monitoramento do RSVP, o que acarreta as seguintes habilidades:

- Assegurar largura de banda, jitter e limites de retardo para tráfegos time-sensitive, como transmissão de voz.
- Assegurar largura de banda para aplicações multimídia como videoconferência e aprendizado a distância.
- Previnir aplicativos que "roubam" largura de banda de atrasar fluxos prioritários ou prejudicar a performance de outros aplicativos que estão rodando na mesma rede.

COPS para RSVP suporta as seguintes características do RSVP:

- *Admission control* - A reserva do RSVP é aceita ou rejeitada baseada na disponibilidade de recursos na ligação fim-a-fim da rede.
- Garantia de banda - Assegura que uma vez reservada a largura de banda ela assim continuará até que a reserva seja cancelada.

- Classificação de dados - Enquanto a reserva não é cancelada, pacotes de dados pertencentes ao fluxo RSVP são separados dos demais e transmitidos como parte do fluxo reservado.

- Policiamento de dados - Pacotes de dados pertencentes ao fluxo RSVP, que excedem o tamanho da largura de banda, são marcados como um pacote de precedência baixa.

Funcionamento do COPS para o RSVP

O RSVP com policiamento COPS funciona da forma como demonstrado na figura 3.3.2.a . Primeiramente, o RSVP transmissor envia uma mensagem PATH ao PEP, que por sua vez envia ao PDP uma requisição COPS, se a decisão deste for positiva então o PEP transmite a mensagem PATH para o Receptor. O receptor então envia o pedido de reserva para o PEP que realiza o mesmo procedimento enviando a requisição e recebendo a decisão do PDP, se a decisão for favorável então a reserva é feita e a mensagem RESV enviada ao transmissor. O processamento de cada mensagem RSVP depende do policiamento armazenado no servidor em questão.

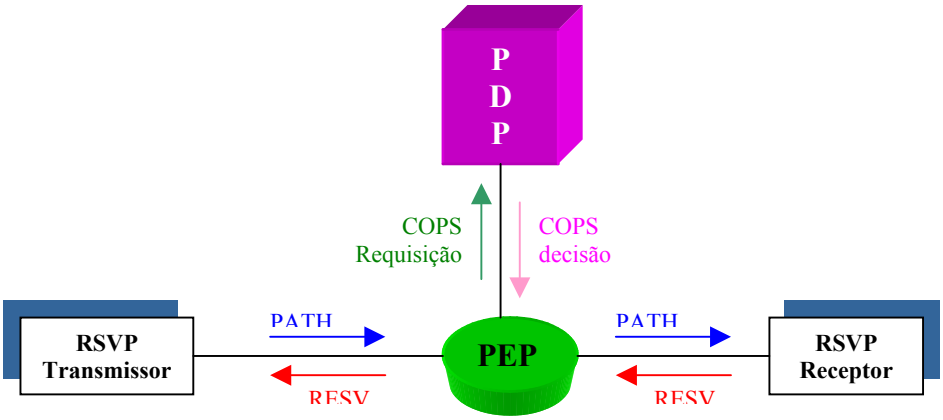

Figura 3.3.2.a - Exemplo de funcionamento do COPS com RSVP.

3.4 SIP

O Protocolo de Inicialização de Sessão ou *Session Initiation Protocol* (SIP) é um protocolo de nível de aplicação definido na [RFC2543], amigável e simples foi desenvolvido para estabelecer, modificar ou terminar sessões multimídia. Essas sessões incluem voz através da WAN, conferências, aprendizado a distância, telefone via Internet e aplicações similares.

SIP pode ser usado para iniciar sessões assim como convidar membros para sessões que foram estabelecidas por outros meios. Essas sessões podem ser tanto *multicast* quanto *unicast* e o inicializador não precisa ser necessariamente da mesma sessão de onde vem o convite. Ele pode iniciar uma chamada *multi-party* usando a unidade de controle de multiponto (multipoint control unit - MCU) ou uma interconexão *fully-meshed*. Gateways de Internet que conectam com o PSTN (switch público de telefone), podem usar o SIP para realizar chamadas entre eles.

3.4.1 Serviços SIP

O SIP oferece mecanismos necessários para que sistemas finais e servidores *proxy* possam oferecer os seguintes serviços:

- Call forwarding (transmissão de chamadas).
- Numeração para chamador e para quem está sendo chamado, onde cada número deve ser distinto, bem como participar de um esquema de nomes.
- *Personal Mobility* (mobilidade pessoal) - Capacidade de mapear nomes e realizar serviços de redirecionamento, onde usuário finais podem se locomover de uma subrede para outra sem se preocupar, pois a rede o identificará.
- Negociação e seleção de tipos de terminais .
- Autenticação tanto do chamador quanto do chamado.
- Chamada de transferência supervisionada e cega.
- SIP pode convidar usuários com ou sem reserva de recursos. Ele não faz a reserva, mas pode fornecer ao sistema final informações necessárias para que isso ocorra.

O SIP foi desenvolvido como parte dos dados IETF multimídia e a arquitetura de controle incorpora protocolos como RSVP, RTP, SAP (*Session Announcent Protocol*), SDP (*Session Description Protocol*), entre outros protocolos de sinalização e estabelecimento de chamadas. Entretanto, a funcionalidade e a operação do SIP não depende desses protocolos.

O SIP é independente do nível de pacote e simplesmente requer um serviço datagrama, pois ele possui seu próprio mecanismo de confiança. Tipicamente SIP é usado sobre o UDP ou o TCP, porém tecnicamente ele também pode rodar sobre IPX, frame relay, ATM AAL5 ou X.25.

3.4.2 Mecanismo de funcionamento

Existem dois modos de operação do SIP o modo Proxy e o modo de redirecionamento. No modo Proxy o funcionamento básico consistem em fazer uma chamada ao usuário através de uma mensagem, a SIP request, que é enviada ao servidor proxy, este por sua vez tenta localizar o usuário final e enviar a mensagem. A resposta do usuário final retorna para o servidor proxy que a transmite de volta para o servidor que gerou o chamado.

Já no modo de redirecionamento o usuário envia a mensagem SIP request para o servidor, que por sua vez busca o endereço do usuário final e quando o localiza transmite os dados do usuário final de volta para o servidor que a gerou e dessa forma a comunicação passa a ser direta com o usuário final.

Para estabelecer e terminar sessões multimídia o SIP possui:

- User Location (localização de usuários)- determina o sistema final que será usado na comunicação.

- User Capabilities (capacidade dos usuários) - determina a mídia que será usada e seus parâmetros.

- User availability (disponibilidade de usuário)- determina a disposição dos usuários finais em participar da comunicação.

- Call setup - estabelece os parâmetros das chamadas de ambos os usuários inicial e final.

- Call handling - inclui transferência e terminação de chamadas.

Endereçamento

Endereça usuários nos hosts identificados com uma SIP URL, que é similar a um endereçamento do tipo user@host, onde o *user* é parte de um nome ou um número de telefone e o *host* é parte do nome do domínio ou um endereço numérico de rede.

O SIP oferece autenticação e controle de acesso a mecanismos e pode disponibilizar para si mecanismos de níveis baixos de segurança, para que o software cliente possa rejeitar chamadas indesejadas e não autorizadas.

Servidor SIP

Quando um cliente deseja enviar uma requisição, ele pode tanto enviá-la para o servidor SIP local quanto para um endereço IP correspondente à URL. Neste último caso o cliente precisa definir o protocolo, a porta e o endereço IP do servidor.

Transação SIP

Uma vez localizado o servidor SIP, o cliente envia uma requisição e espera a resposta. A requisição e a resposta à ela formam a transação SIP. Todas as respostas possuem os mesmos valores no Call-ID, Cseq, To e From existentes nas requisições que a geraram.

O Formato da mensagem SIP e sua operação é independente do protocolo de transporte.

Convite SIP

Um Convite SIP consiste em duas requisições, INVITE seguido de ACK ou de BYE. A requisição INVITE pergunta ao cliente final se este deseja se juntar à conferência ou estabelecer uma conversa. Se o cliente aceita o convite, ele envia ao servidor uma requisição ACK. Quando desejar terminar a participação o cliente envia uma requisição BYE.

Tipicamente uma requisição INVITE possui informações suficientes para que o cliente se junte à sessão. Se o cliente aceitar o convite, ele deve retornar a mensagem com informações sobre as mídias que deseja usar.

É importante ressaltar que a requisição INVITE seguido de um ACK não corresponde a uma transação SIP.

Mensagem

Muita da sintax da mensagem é igual a do protocolo HTTP.

Diferentemente do HTTP, quando enviando dados sobre o TCP ou o UDP, várias transações SIP podem ser carregadas em uma simples conexão TCP.

Métodos

Métodos que não são suportados pelo servidor Proxy ou pelo servidor redirecionador, é tratado como uma opção de método e transmitido de acordo. Os Métodos são os seguintes:

- INVITE - indica que o usuário ou o serviço está sendo convidado a participar de uma sessão. No corpo dessa mensagem estão informações como o tipo de mídia que ele irá receber. O usuário deve responder ao convite identificando na resposta o SIP Call-ID ou um identificador global.

- ACK - A mensagem ACK confirma que o usuário recebeu uma resposta final a requisição INVITE.

- OPTIONS - O servidor é analisado por sua capacidade. Se um servidor acredita que ele pode contactar um usuário que está logado e recentemente ativo, ele deve responder a mensagem setano a capacidade.

- BYE - Indica para o servidor que o usuário deseja terminar a chamada.

- CANCEL - Cancela uma requisição pendente enviando o mesmo Call-ID, To, From e Cseq pertencentes ao cabeçalho da mensagem.

- REGISTER - Um cliente usa esse método para registrar o endereço listado no cabeçalho do SIP server.

3.4.3 *Draft's* para SIP

A [DRAFT5] defende que devido ao fato do SIP vir sendo aceito como um protocolo para sinalização de multimídia e para Internet por telefone, é essencial que se desenvolva um mecanismo para suportar aplicativos para multimídia móvel no ambiente do SIP.

Essa draft descreve os requisitos gerais para o gerenciamento de redes móveis, sem fio. Identifica as funções necessárias para suportar a mobilidade assim como pontos em aberto envolvendo o suporte para usuários em um ambiente SIP.

3.5 RTSP

O *Real Time Streaming Protocol* ou RTSP, definido na [RFC2326], é um protocolo de nível de aplicação, que controla fluxos de tempo-real. Ele é capaz de transmitir através de redes *multicast* e *unicast*. O protocolo, em si, não entrega os dados da mídia apesar disso ser possível através do Stream de controle. O RTSP age como se fosse um "controle remoto de rede" para o servidor multimídia.

Não existe a noção de conexão no RTSP, ao invés disso, o servidor mantém uma sessão aberta através de um identificador. Uma sessão RTSP não está ligada ao nível de transporte, durante uma sessão o RTSP pode abrir e fechar várias sessões com níveis diferentes de conexão de transporte, podendo inclusive atuar sobre um datagrama. O RTSP foi desenvolvido para atuar sobre o RTP, TCP, UDP, entre outros protocolos, o que torna possível para ele aproveitar novas funcionalidades nos protocolos dos níveis inferiores, sem precisar ser adaptado.

A principal característica do RSTP é permitir que dados multimídia sejam entregues eficientemente sobre a Internet. O RSTP oferece os seguintes serviços para as aplicações:

- Permite que as aplicações requisitem entrega de dados de tempo-real.
- Requisite um tipo específico de transporte e de destino para a entrega de dados.
- Requisite informações sobre dados de forma específica.
- Inicie, interrompa e pause as entregas de dados.
- Forneça acesso aleatório a várias porções dos dados.

3.5.1 Mecanismo de funcionamento

Sintaxe

A sintaxe do RTSP é similiar ao do HTTP, isso se dá por vários motivos, entre eles, o fato de qualquer extensão do HTTP poder ser ampliada ao RTSP. Ele pode adotar os esquemas de segurança, caches e proxies do HTTP.

Apesar de serem parecidos o RTSP e o HTTP possuem algumas diferenças, entre ela o fato de tanto o cliente quanto o servidor poder emitir uma requisição.

Operações

O Protocolo suporta as seguintes operações:

- Recuperação das mídias que encontram-se no servidor: o cliente pode solicitar uma descrição de apresentação e pedir para o servidor estabelecer uma seção para enviar dados requisitados.

- Convite para se juntar a uma conferência feito à um servidor: Este pode ser convidado a participar tocando ou gravando uma apresentação.

- Adição de mídias à uma apresentação existente: o servidor, ou o cliente podem avisar um ao outro sobre mídias adicionais que tornaram-se disponíveis.

Métodos

Abaixo seguem os métodos suportados pelo RTSP, é preciso ressaltar que alguns métodos podem ser enviados tanto do servidor para o cliente quanto do cliente para o servidor:

- SETUP - Faz com que o servidor aloque recursos para o *stream* e inicie uma sessão.

- PLAY e RECORD - Inicia a transferência de *stream* alocado no setup.

- PAUSE - Suspende temporariamente o *stream* sem "limpar" os recursos do servidor.

- TEARDOWN - Libera os recursos associados ao *stream*.

CAPÍTULO 4 - QOS E REDES IEEE 802

No Capítulo anterior foram descritos alguns protocolos de suporte a QoS, sem considerar o nível de enlace. Um conjunto de propostas foram feitas para classificar tráfegos e discriminar fluxos multimídia.

Esse conjunto de propostas está relacionado à família IEEE, abaixo segue uma breve descrição dessas propostas.

4.1 802.1p

Esse padrão faz parte do IEEE 802.1D, que cobre a classificação do tráfego e serviços de filtragem dinâmica para *multicast* em pontes LANs. Ele usa o mesmo formato tag proposto pelo 802.1q standard, mas usa três bits adicionais para o controle da informação, que ceta o nível de prioridade. É possível classificar tráfegos IEEE 802.1p específicos utilizando o valor tag, ou definir controles que inserem prioridades em frames transmitidos.

A prioridade possui valores de 0 a 7 e pode ser observada na figura 4.1.a . O Valor 7 é a prioridade máxima e a 1 a menor. O valor 0 tem uma prioridade maior que o 2.

Valor Tag	Tipo de tráfego
1	Background
2	Standart
0 (default)	Best effort
3	Excellent effort (comercio crítico)
4	Controlled Load
5	Video,
6	Voz
7	Controle de Rede

Figura 4.1.a - Tabela de prioridades

O IEEE 802.1p standard endereça separadamente quadros com tempo crítico para reduzir jitter que é causado por flooding multicast. Esse standard define o *Generic Attribut Registration Protocol* (GARP), um mecanismo de transporte do nível 2 que permite que switches e sistemas finais propaguem informações através de domínios.

4.2 802.1q

O IEEE 802.1q é um standard para VLAN que tem por objetivo definir uma arquitetura para uma divisão racional de pontes LANs e prover serviços para grupos de usuários definidos, independentemente da localização física. Além disso, permite a interoperabilidade entre equipamentos de múltiplos fabricantes. O standard também especifica um formato tag que encaixa informação explicita de membros VLAN dentro de cada frame em VLAD ID (VID) de 12-bit, que provem 4094 possíveis VLANs.

4.2.1 VLAN

A virtual LAN é um grupo lógico que permite que usuários finais se comuniquem como se eles estivessem fisicamente conectados, independente de sua configuração física na rede.

VLAN é uma alternativa para obter performance maior e uma implementação fácil para roteadores.

Benefícios

- Pode ajudar a administradores a medir movimentos de gestões sem precisar reconfigurar o IP.
- Reduzir o custo de movimento de equipamentos, upgrades e outras mudanças e simplificar a administração de redes.
- Pode ser usada para isolar tráfegos *unicast* de um simples domínio *broadcast*, fornecendo segurança para rede.
- Criar grupos virtuais onde membros de um mesmo departamento ou sessão aparente ser da mesma LAN, com o tráfego estando no mesmo domínio *broadcast*.
- Ajuda a evitar *floading* e minimizar tráfegos *broadcast* e *multicast*.
- Reduzir a necessidade de roteamento para fornecer high-performance para rede, fácil administração com custo reduzido.
- Controlar comunicação sobre domínios *broadcast*.

4.2.2 Definição do 802.1q Tag

CAPÍTULO 5 - QOS E REDES SEM FIO

5.1 Introdução

O uso de telefones celulares tem aumentado substancialmente nos últimos tempos, no entanto redes sem fio tem se desenvolvido muito lentamente, enquanto computadores portáteis estão crescendo no mundo do PC. A razão para o lento desenvolvimento da rede sem fio está no preço da complexidade, performance fraca e na falta de aplicativos *Killers*. O Suporte para multimídia em redes sem fio pode ser o aplicativo *killer,* necessário para popularizar essa rede.

Existem vários pontos a serem definidos para o suporte de QoS em redes sem fio. A mudança de ambiente na rede decorrente da mobilidade e da interferência gera problemas com a banda de transmissão e taxas de erros que mudam dinamicamente. Tradicionalmente problemas em relação as redes tradicionais, como topologia fixa, disponibilidade de recursos não existem nas redes sem fio.

Várias técnicas foram propostas para suportar multimídia em diferentes níveis da rede. No nível de aplicação, as aplicações de tempo real podem ser feita de forma a se adaptarem para alterar condições da rede. No nível de transporte reservas de recursos podem ser feitas durante a conexão para suportar garantias de QoS fim-a-fim. No nível de redes técnicas para permitir gerenciamento móvel e conectividade fraca podem ser usadas. Mecanismos de roteamento precisa ser voltado para QoS e suportar mobilidade. No nível de enlace é preciso modificar o *Medium Acess Control* para que as reservas sejam respeitadas e assim como as garantias de QoS. Técnicas de controles adaptativos podem ser usados para gerenciar a mobilidade e manter os links ativos. No nível físico várias opções existem abrangendo desde infravermelho até radio.

5.2 - Problemas:

Para garantir QoS em Redes sem fio alguns problemas, decorrentes da mobilidade e da mudança de condição, como disponibilidade de largura de banda e taxas de erro, devem ser superados. Abaixo seguem alguns dessem problemas que precisam ser solucionados.

Meio de Acesso

O maior desafio em redes sem fio está focado no ambiente de comunicação que lida com baixas largura de banda e altas taxas de erros. Durante a possibilidade de desconexão, falha no link ou taxas de erro os nós da rede sem fio perecisam ser capaz de trabalhar sozinhos. Grandes áreas de rede sem fio oferecem banda de 10 Kbps enquanto Wireless LANs oferecem de 1 a 2 Mbps, os nós também precisam se adaptar a essas mudanças.

Mobilidade

Redes sem fio não necessariamente são móveis, mas a grande maioria é. Se um nó pertence a rede sem fio e ao mesmo tempo é móvel, então arquivar a todo o tempo a conectividade é um problema a ser considerado. O poder do sinal recebido varia de acordo com o tempo e a localidade.
O algoritmo de roteamento deve permitir um gerenciamento dinâmico da localidade. Quando um host se movimenta, pacotes pendentes devem ser enviados para o host. O problema piora na presença de aplicativos multicast.

Conexão Ad Hoc Multihop

Arquitetura de redes sem fio assumem que seus nós estão conectados a uma rede fixa via estação base. Um sistema como uma rede de pacotes Multihop necessitam comunicar sem nenhuma infraestrutura de cabo, o que acarreta em nem todos os nós estarem dentro da região e necessitar de pacotes para comunicação. Esse tipo de rede ad hoc é mais complexa que uma simple hop.

Adaptatibilidade

Redes sem fio e móveis precisam se adaptar a várias condições de sistemas. A adaptatibilidade no nível físico á possível através da escolha do canal correto, do mecanismo de controle e de erro, esse último é essencial devido às altas taxas de erros nos links da rede. O nível de redes e algoritmos de roteamento devem ser adaptados a mobilidade. Aplicativos precisam ser adaptados para suportar largura de banda.

5.3 Propostas para suportar Multimídia em redes sem fio

5.3.1 AQuaFWin

O Adaptive QoS Framework foi proposto pelo departamento de ciência da computação da Universidade do estado de Ohio nos EUA, para o suporte de QoS em ambiente dinâmicos, onde QoS deve ser suportado em todos os níveis. Para permitir a adaptatibilidade um mecanismo de feedback fim-a-fim é usado. Ele á arquivado periodicamente por pacotes de controle que pede informações feedback que pode ser usado em níveis diferentes.

As características dos componentes do AquaFWin são:

- Aplicativos adaptativos - São aplicativos de tempo real como voz e vídeo que se tornam adaptativos através da modificação de parâmetros de encapsulamento.

- Arquitetura de Rede - Uma rede ligada de forma hierárquica contendo hosts móveis, cluster de hosts móveis, estações base e redes fixas.

- Adaptatibilidade - Para arquivar informações sobre o estado corrente da rede é usado o feedback.

- QoS flexiveis - Propoem parâmetros flexiveis de QoS que podem ser usados durante a conexão e para suportar ambientes móveis da rede.

- Separação de redes sem fio e redes fixas - As redes fixas devem ser transparentes às mudanças da parte móvel da rede. Essa separação permite o uso de nós supervisores propostos na arquitetura de redes.

5.3.2 - Suporte do RSVP para Ipv6 móveis

Uma extensão do RSVP foi proposta para solucionar questões de IP móveis, possibilitando a comunicação entre hosts móveis e fixo na rede.

Esse método implica em novas mensagens e novas classes de objetos sendo adicionadas ao RSVP, para incrementar a performance, uma delas é a modificação de ambos os hosts móveis e fixos para se adaptar ao endereçamento MIPv6. Outro mecanismos consiste no update da mensagem PATH para que seja possível roteadores intermediários reconhecer conecções e usar recursos mesmo que os endereços mudem. Um mecanismo *flow extension* também é proposto para combinar o fluxo RSVP existente em roteadores IP típicos, com os novos roteadores IP móveis.

CAPÍTULO 6 - ESTADO DA ARTE

Ao longo deste documento foram apresentados algumas das novidades ligadas a Qualidade de Serviço, como as possíveis alterações nos protocolos, onde estes seriam adaptados para proporcionar um melhor desempenho do serviço multimídia, bem como os novos protocolos que surgiram, como por exemplo o COPS para o RSVP (janeiro de 2000).

Adicionalmente também foi visto a nova rede sem fio que está surgindo, seus problemas e algumas possíveis soluções que estão sendo analisadas. Essa rede exige a adaptação dos requisitos de QoS, e possivelmente uma nova parametrização destes para que seja possível suportar as demandas dessa nova rede.

Grandes empresas como a CISCO e a 3COM desenvolvem importantes ferramentas para a monitoração e implementação de QoS nas redes, entre essas ferramentas estão o CoreBuilder 9000 da 3COM e a família CISCO. Essas ferramentas permitem implementar algumas QoS como, por exemplo a CISCO, suporte a classificação de pacotes usando bits no cabeçalho IP, através de um classificador de pacotes chamado CAR (Committed Acess Rate), ou serviços de limitação de fluxos através de protocolos denominads WRED e DWRED, extensões criadas pela CISCO em cima do protocolo de roteamento RED.

Essas empresas trabalham em cima de diversas tecnologias de rede, na tentativa de adequa-las às necessidades de QoS. Na medida em que vão surgindo novos algoritmos e adaptações em cima de protocolos esses vão sendo submetidos aos respectivos órgão para que haja então uma padronização.

CAPÍTULO 7 - CONCLUSÃO

A Qualidade de Serviço permite que se defina pontos importantes para adequar as demandas dos fluxos multimídia. Conforme novas tecnologias vão surgindo é possível observar que seus parâmetros vão se adaptando, para melhor se adequar as necessidades.

Em um mundo onde a demanda por novos sistemas multimídia estão surgindo, há uma necessidade crescente de atualização e diversos projetos e pesquisas em cima deste assunto aparecem.

[RFC]