Modelagem e Implementação de um Protótipo para Sinalização de Qualidade de Serviço na Internet

Philippe Azevedo Q. Santos[3] e Paulo Cesar S. Vidal[1,2]
philippe@citex.eb.mil.br e vidal@ime.eb.br
[1]Instituto Militar de Engenharia (IME) – Rio de Janeiro, RJ, Brasil
[2]2º Centro de Telemática de Área (2º CTA) – Rio de Janeiro, RJ, Brasil
[3]Centro Integrado de Telemática do Exército (CITEx)– Brasília, DF, Brasil

Resumo – O protocolo RSVP (*ReSource reserVation Protocol*) foi definido pela IETF como o padrão para sinalização de qualidade de serviço na *Internet.* Porém, o protocolo RSVP apresenta problemas de escalabilidade e de complexidade e não possui suporte às redes móveis. Para minimizar esses problemas e oferecer um suporte à mobilidade, foi proposto o protocolo RSVPv2. A partir do estudo da especificação do RSVPv2, foi implementado um protótipo do protocolo em redes fixas, utilizando a linguagem de modelagem UML na especificação e documentação do protótipo. Foi criado um ambiente virtual constituído por máquinas virtuais com o sistema operacional Linux e uma topologia de redes virtuais, com o objetivo de testar o protótipo.

Palavras Chave – Protocolo de Sinalização, Qualidade de Serviço, Redes Móveis

Abstract – The RSVP protocol (ReSource reserVation Protocol) was defined by the IETF as the standard for signaling of quality of service in the Internet. However, the RSVP protocol presents scalability problems and of complexity and it don't support to the mobile networks. To minimize these problems and to offer a support to mobility, the RSVPv2 protocol was proposed. From a study of the specification of the RSVPv2, it was implemented a prototype for the protocol in wired networks, using the modeling language UML in the specification and documentation of the prototype. A virtual environment was created, consisting by virtual machines with the operational system Linux and a virtual network topology, with the objective to test the prototype.

Index Terms – Signaling Protocol, Quality of Service, Mobile Networks.

I. INTRODUÇÃO

A evolução dos dispositivos portáteis e de tecnologias de acesso sem fio possibilitou o aumento da potência e da taxa de transmissão dos computadores portáteis. Os usuários de dispositivos móveis na *Internet* desejam utilizar os mesmos serviços oferecidos aos usuários de uma rede fixa.

Contudo, com o esquema de endereçamento tradicional da *Internet*, isso não é possível, pois o endereço é fixo, e assim é incompatível com a mobilidade desejada. Como solução para esse problema foi proposto o protocolo IP móvel, que permite a acessibilidade contínua da estação móvel aos serviços de rede.

Além de garantir acessibilidade contínua é necessário oferecer um tratamento diferenciado às aplicações multimídia sensíveis à largura de banda, ao atraso, à variação de atraso e a perda de pacotes, como VoIP (*Voice over IP*), TV digital e telemedicina. No entanto, o modelo atual de encaminhamento de pacotes da *Internet* é o serviço de melhor esforço que não fornece garantias de qualidade de serviço. Então, para garantir que aplicações recebam um tratamento diferenciado e funcionem satisfatoriamente, é necessário utilizar arquiteturas de qualidade de serviço (*Quality of Service - QoS*). Como exemplo destas arquiteturas, pode-se citar a Arquitetura de Serviços Integrados (arquitetura *Intserv*) e a Arquitetura de Serviços Diferenciados (arquitetura *Diffserv*).

Os principais elementos de uma rede de computadores capaz de oferecer qualidade de serviço, na Internet, são: o escalonador, o controle de admissão e o protocolo de sinalização. Segundo Fu [1], um protocolo de sinalização de qualidade de serviço é necessário para estabelecer, manter e remover estados de reservas nos nós da rede e importante para encaminhar os requisitos de qualidade de serviço fim-a-fim. As aplicações usam o protocolo de sinalização para informar aos nós da rede quais fluxos terão tratamento especial e como serão as suas características e parâmetros.

O protocolo mais utilizado, para esta tarefa é o protocolo RSVP (*ReSource reserVation Protocol*) [2]. Entretanto, o protocolo RSVP apresenta problemas de escalabilidade e complexidade, devido ao armazenamento e processamento dos estados, de cada fluxo, em todos os roteadores que forneçam suporte ao fluxo, por isto seu uso na *Internet*, não obteve o sucesso esperado. Além disto, o protocolo RSVP não oferece suporte à mobilidade.

Diversos trabalhos surgiram para oferecer extensões que possibilitassem o protocolo RSVP a funcionar corretamente em redes móveis, mas mesmo com seu funcionamento adaptado, as extensões ainda possuem problemas relativos à escabilidade e complexidade [3,4].

O grupo de trabalho NSIS (*Next Steps in Signaling*) da IETF tem apresentado diversas propostas nesta área com dois objetivos [5]. Primeiro, é reusar o mecanismo do protocolo RSVP em determinados casos. O segundo objetivo é

padronizar um protocolo de sinalização de qualidade de serviço.

Como parte do grupo, Brunner et. al. [6] propõe o protocolo RSVP versão 2 (*ReSource reserVation Protocol version 2*). Esse protocolo corrige os problemas de complexidade, escalabilidade e de mobilidade. Possui um número reduzido de mensagens para a construção de estados de reserva, suportando, em sua especificação inicial, apenas reserva em comunicação ponto a ponto. Foi realizada a implementação de um protótipo para testes. Esta versão desse protocolo não é uma adaptação ou modificação do RSVPv1, mas sim um novo protocolo.

Ziemer e Vidal [7] apresentam extensões ao protocolo RSVPv2 como a criação de uma mensagem de remoção de reserva enviada pelo roteador de cruzamento, diminuindo o tempo de eliminação da antiga reserva após o *handoff*. Estas extensões foram avaliadas em cenários de mobilidade utilizando o simulador NS-2.

Tschofenig et al. descrevem os principios de projeto, a arquitetura e subprotocolos para sinalização IP, chamado protocolo Cross-Application Signaling Protocol (CASP) [8].

A arquitetura CASP suporta várias aplicações de sinalização, tais como: reserva de largura de banda, distribuição de rótulos em redes MPLS e sinalização de regras de *firewall*.

O trabalho de Hogrefe et al. [9] apresenta o protocolo GIST (*General Internet Signaling Transport*), sua implementação e avaliação de desempenho realizada na Universidade de Gottingen.

Assim, o objetivo desse artigo é apresentar o projeto e a implementação de um protótipo do protocolo RSVPv2 que servirá de base para a realização de estudos e testes para sinalização de qualidade de serviço em redes móveis infra-estruturadas. Como não existe código disponível publicamente desse protocolo, foi realizada uma nova implementação do protótipo.

Esse artigo está organizado da seguinte maneira. A Seção II apresenta as principais características do protocolo RSVPv2. A Seção III mostra o projeto e a implementação do protocolo. A Seção IV apresenta a arquitetura de testes e os resultados de execução. A Seção V é dedicada às considerações finais e trabalhos futuros.

II. VISÃO GERAL DO PROTOCOLO RSVPv2

Os objetivos da nova versão do protocolo RSVPv2 são obter uma maior escalabilidade do protocolo (através da redução da informação armazenada em cada ponto da reserva), diminuição da complexidade (diminuição do número de mensagens) e fornecer suporte à sinalização de qualidade de serviço em redes móveis. Estas características foram determinadas levando em conta os aspectos positivos e negativos do protocolo RSVP quanto ao suporte à mobilidade e a escalabilidade.

O protocolo RSVPv2 possui quatro tipos de mensagens: RESV, ACK/NACK, TEAR e REFRESH.

A mensagem RESV (*Reservation*) é originada pelo transmissor e é destinada à criação das reservas, ao longo do caminho até chegar ao receptor. As mensagens ACK (*ACKnowledgment*) e NACK (*Not ACKnowledgment*), representadas por (ACK/NACK), são usadas como resposta no processo de criação das reservas.

A mensagem ACK informa ao transmissor do pedido da reserva que o processo de criação da reserva foi feito com sucesso. No caso de restabelecimento da reserva, em um novo caminho, esta mensagem é responsável pelo reaproveitamento da reserva. A mensagem NACK indica um erro e sua descrição que ocorreu ao longo da rede.

A mensagem TEAR (remoção) é responsável pela remoção dos estados de reserva de um determinado caminho. Em situações de mobilidade e quando ocorrer uma alteração de caminho esta mensagem é útil na remoção da reserva no caminho antigo.

A mensagem REFRESH é responsável pela atualização da reserva feita anteriormente, antes que a reserva seja removida pelo temporizador, ou seja, expirada.

Uma mensagem RSVPv2 é constituída basicamente por três estruturas: o cabeçalho comum, uma parte específica ao tipo de mensagem e uma parte específica ao serviço. No cabeçalho comum, temos estruturas existentes em todas as mensagens do protocolo. No campo específico ao tipo de mensagem temos campos que existem apenas de acordo com o tipo da mensagem. No campo específico ao serviço, temos os dados relativos ao serviço.

III. PROJETO E IMPLEMENTAÇÃO DO RSVPv2

O protótipo do protocolo RSVPv2 apresenta as seguintes funcionalidades: criar reserva de recursos, finalizar a reserva, mecanismo *soft-state* e mudança de sub-rede.

Criar uma reserva de recursos - permitir ao transmissor iniciar um pedido de reserva, para que se possa iniciar um fluxo de dados ao receptor com garantias mínimas de qualidade de serviço. O pedido é encaminhado pela mensagem RESV, que será analisada e processada pelos roteadores existentes ao longo do caminho, até chegar ao receptor. O receptor irá enviar uma mensagem de confirmação ACK, que será enviada pelos roteadores até chegar ao transmissor, permitindo o início do envio de dados com a garantia do serviço pedido. No caso de erro, ou insucesso na requisição da reserva, uma mensagem de erro é enviada para o transmissor, para que ele possa fazer novamente um novo pedido, ou renegociar os parâmetros de qualidade de serviço.

Finalizar a reserva de recursos - O protótipo deverá permitir ao transmissor finalizar um pedido de reserva de recursos. O transmissor envia a mensagem TEAR em que indica o término da reserva que será analisada e processada pelos roteadores existentes ao longo do caminho, ocasionando assim a liberação de recursos armazenados anteriormente, até chegar ao receptor.

Mecanismo de Soft-State – esse mecanismo armazena as informações de reserva dos fluxos em cada nó na rota do fluxo de uma reserva fim-a-fim. Para que o desempenho do protocolo RSVPv2 seja melhorado, somente os pontos finais enviam as mensagens REFRESH. Esse mecanismo não requer a existência de nenhuma mensagem de remoção explícita de reserva. Esta característica pode se mostrar bastante útil, pois recursos são desalocados no momento em que não são mais utilizados (como por exemplo, na ocorrência de *handoff*).

Foi utilizada a linguagem UML (*Unified Modeling Language*) [10] para a modelagem do protótipo.

O levantamento de requisitos é representado pelo diagrama de casos de uso da Fig. 1.

Os atores do sistema encontrados são: transmissor, receptor e o tempo.

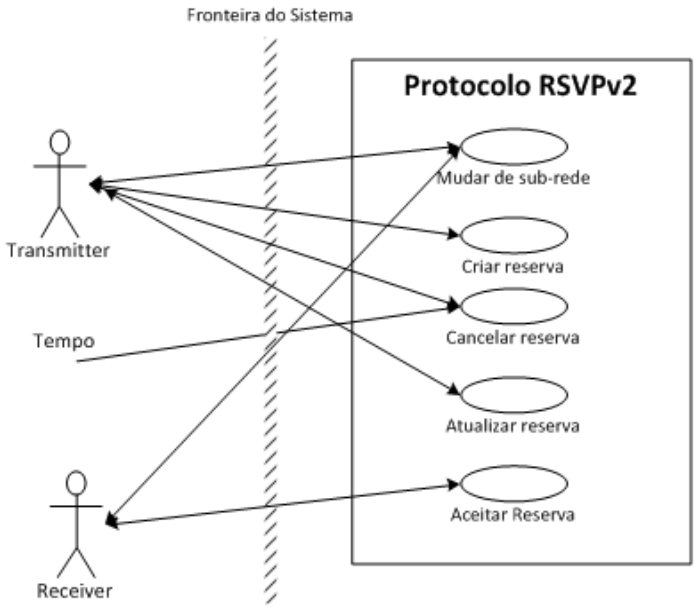

Fig. 1. – Diagrama de casos de uso.

A partir dos casos de uso foi desenvolvido o diagrama de classes para o protótipo que está apresentado na Fig. 2.

As principais classes apresentadas no diagrama são:

- A classe *RSVPMsg* representa qualquer tipo de mensagem do protocolo. Ela contém três objetos relativos as classes *CommonHeader*, *TypeSpecific* e *ServiceSpecific*.

- A classe *Common Header* representa o conjunto de atributos comuns em todas os tipos de mensagens.

- A classe *Service Specific* representa os atributos existentes em relação a cada tipo de serviço. No objetivo desse trabalho foi desenvolvido um único serviço, que é garantia de largura de banda da rede.

- A classe *Type Specific* representa os atributos existentes em cada tipo de mensagem que é definido pelo campo *msgType* da classe *Common Header*. No entanto esta classe é um adaptador, para que possa armazenar um conjunto de atributos específicos a cada mensagem. Assim, temos uma estrutura que pode receber os dados de qualquer tipo de mensagem (RESV,ACK/NACK e REFRESH), exceto a mensagem TEAR, que não possui parâmetros adicionais.

- A classe *Session* representa uma estrutura que contém um RSB (*Reservation State Block*) de uma determinada reserva [2]. Contém, basicamente, informações da mensagem RESV como endereço do receptor, especificação de reserva e estilo de reserva.

- A classe *SessionList* armazena todos os objetos da classe *Session*, sob a forma de uma lista e que são constantemente verificados devido ao mecanismo de *soft-state*. Ou seja, possui todas as reservas efetuadas. No caso de uma reserva expirar, um temporizador irá apagar a reserva da lista. Através desta classe, podem-se calcular os parâmetros de serviços já reservados. E verificar se é possível ou não a criação de uma nova reserva.

- A classe *Router é* responsável pelo envio de uma chamada de sistema que recupera a tabela de roteamento do sistema e assim permite a consulta ao nível de processo do usuário. Nela, temos um método que encontra o endereço IP do próximo salto (*next-hop* ou *gateway*) para o qual a mensagem deve ser retransmitida. Assim, permite o roteamento de mensagens do protocolo.

No diagrama de classes não foram apresentados os métodos implementados para simplificar a figura.

Na implementação do RSVPv2 foram utilizados a linguagem C++ e o sistema operacional Linux Kurumin 6.1. O desenvolvimento foi baseado em redes infraestruturadas. Além das classes citadas acima, foram desenvolvidos três programas executáveis que representam o transmissor, o receptor e o roteador, para representar as entidades presentes nos cenários de teste.

A Tabela I exibe uma correspondência entre os três módulos (programas executáveis) desenvolvidos e as mensagens do protocolo RSVPv2.

TABELA I
RELAÇÕES ENTRE OS MÓDULOS DESENVOLVIDOS, SUAS AÇÕES E AS MENSAGENS ENVOLVIDAS

	Envia	Recebe	Repassa	Processa
Transmissor	RESV REFRESH TEAR	ACK/NACK		ACK/NACK
Receptor	ACK/NACK	RESV REFRESH TEAR		RESV REFRESH TEAR
Roteador			RESV REFRESH TEAR ACK/NACK	RESV REFRESH TEAR ACK/NACK

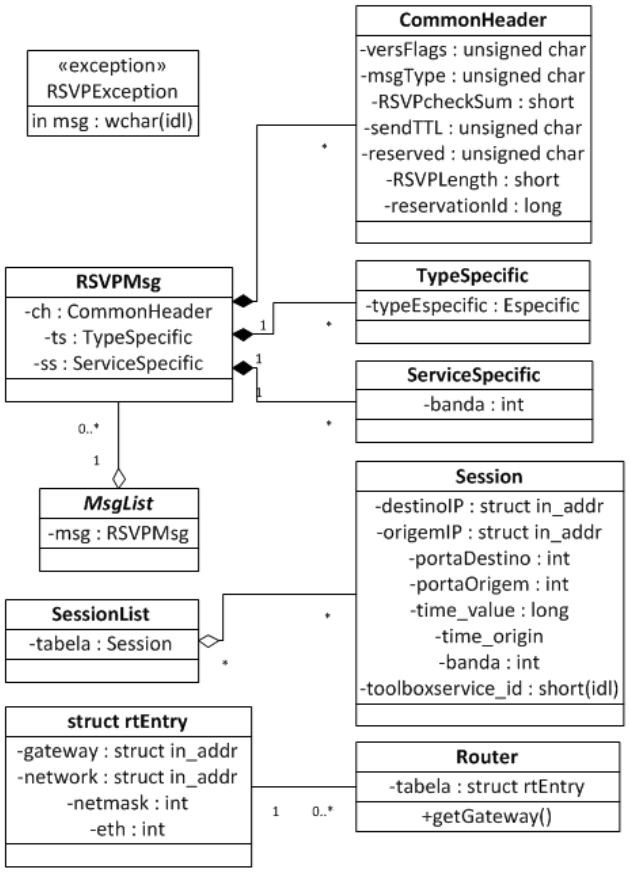

Fig. 2. – Diagrama de classes.

O protótipo do receptor desenvolvido apresenta as seguintes funcionalidades:

- Receber mensagens RESV e criar uma reserva de acordo com os parâmetros de serviço especificados.
- Receber mensagens TEAR indicando o fim da reserva.
- Receber mensagens REFRESH que indica uma atualização da reserva com os novos parâmetros de serviço especificados.
- Enviar mensagens ACK/NACK indicando o sucesso ou a falha na negociação.

O protótipo do transmissor desenvolvido apresenta as seguintes funcionalidades:

- Enviar mensagens RESV indicando a criação de uma reserva com parâmetros de serviço especificados.
- Enviar mensagens TEAR indicando o fim da reserva.
- Enviar periodicamente mensagens REFRESH indicando uma atualização da reserva com os novos parâmetros de serviço especificados aos roteadores.
- Receber mensagens ACK/NACK indicando o sucesso na negociação ou a falha por diversos motivos.

O protótipo do Roteador desenvolvido realiza as seguintes funcionalidades:

- Ao receber uma mensagem RESV, a mensagem é repassada e é criada uma sessão para a mesma, caso os parâmetros de serviço especificados possam ser respeitados.
- A mensagem TEAR é repassada e é finalizada a reserva reservada anteriormente no Roteador.
- Sendo uma mensagem REFRESH recebida, é atualizada a reserva com os novos parâmetros de serviço especificados, além de ser repassada.
- A mensagem ACK/NACK é repassada, indicando o sucesso da reserva de recuros, ou no caso de falha na negociação, são liberados os recursos que já foram alocados.

A funcionalidade de repassar significa que o módulo procura pelo próximo salto (next-hop) a qual a mensagem deve ser transmitida. O endereço IP do destino é encontrado, a partir do objeto Session que possui os principais dados da mensagem RESV. A partir daí, é feito uma consulta na tabela de rotas e é procurado o next-hop. No caso de não haver um next-hop, nesse caso o next-hop é 0.0.0.0, a mensagem deve ser enviada diretamente ao endereço IP de destino.

IV. ARQUITETURA DE TESTE E RESULTADOS

Utilizamos no teste do protótipo uma máquina com a seguinte especificação: processador AMD Athon Xp 2200+, memória Ram de 1 Gb, sistema operacional Windows XP Professional, VMware 5.0 Workstation e Linux Kurumin 6.1 instalado nas máquinas virtuais.

O programa VMware permite criar computadores virtuais (*Virtual Machines* ou *VM*) e criar interfaces de rede virtuais, dentre outras configurações. Antes de instalar o sistema operacional nas máquinas virtuais, é necessário configurar a topologia das redes virtuais inicialmente, conforme apresentado na Fig. 3.

Isso é possível por meio da possibilidade de criar grupos de máquinas virtuais.

Cada grupo pode ter sua topologia de rede virtual que possui segmentos de rede com uma taxa de transmissão de dados e uma taxa de erros específica para cada segmento pré-configurado. Bem como, cada grupo possui o seu próprio conjunto de méquinas virtuais. Foi especificado que em cada grupo tivessem duas máquinas, para representar um quadro típico de gerenciamento de reservas de recursos.

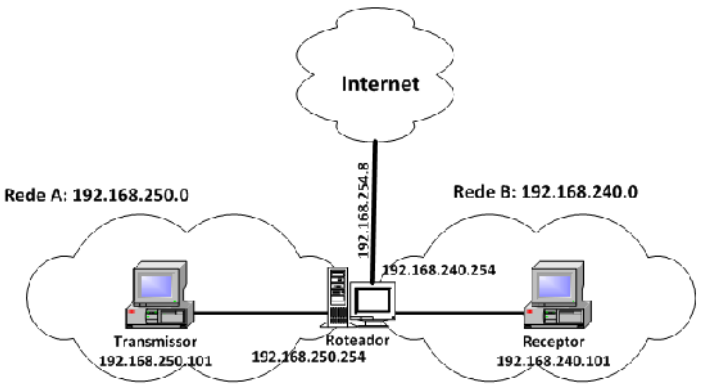

Fig. 3. – Topologia da rede virtual.

Observando a Fig. 3, o roteador foi configurado para prover acesso a *Internet* para as redes A e B. Na rede A, o transmissor foi configurado estaticamente com o IP 192.168.250.101 e ao roteador foi definido o IP 192.168.250.254. Na rede B, o receptor foi configurado com o IP 10.68.240.101 e ao roteador o IP 192.168.240.254.

Criação da Reserva

Abaixo, é representada a criação de uma reserva de banda no valor de 100 kbytes/s com um intervalo de tempo no valor de 10 segundos. O destino da reserva é representado pelo receptor no endereço IP 192.168.240.101.

Criando pacote
Tempo de reserva (em mili segundos)
10000
Mensagem RSVP: Resv Id: 1804289383
Header: CheckSum: 0xfad6
Header: TTL: 0x3
RESV: IP Local: 192.168.250.101
RESV: IP Destino: 192.168.240.101
RESV: TimeValue: 10000
SRV: ESPEC Banda:100
Enviado!
Esperando por resposta...
.....
0x80512fcReserva confirmada!
Mensagem RSVP: Resv Id: 1804289383
Header: CheckSum: 0x1258
Header: TTL: 0x2
ACK: Code Desc: ACK
ACK: Service Desc: RSVP carga controlada e serviço garantido
SRV: ESPEC Banda: 100 kbytes/s

O fluxo das mensagens RESV, recebida do transmissor, e ACK, proveniente do receptor, é registrado no roteador. No receptor a mensagem RESV é recebida e é enviada uma mensagem ACK de confirmação de reserva:

Mensagem RSVP: Resv Id: 1804289383
Header: Version: 0x2
Header: CheckSum: 0xfad7
Header: TTL: 0x2
RESV: IP Local: 192.168.250.101
RESV: IP Destino: 192.168.240.101
RESV: TimeValue: 10000
RESV: Service Desc: RSVP carga controlada e serviço garantido
SRV: ESPEC Banda:100
Enviado!
.....
0x80512ecPedido confirmado!
Mensagem RSVP: Resv Id: 1804289383
Header: Version: 0x2
Header: CheckSum: 0x1257
Header: TTL: 0x3
ACK: Code Desc: ACK
ACK: Service Desc: RSVP carga controlada e serviço garantido
ACK: Service Id: 1
SRV: ESPEC Banda:100 kbytes/s

Remoção da Reserva

No caso do transmissor, após uma mensagem exceder o tempo pedido de reserva será exibida a seguinte mensagem:

0x80512fcReserva : 846930886 expirada.

No receptor e no transmissor não é exibida nenhuma mensagem, mas o pedido de reserva é removido e os recursos, antes alocados, passam a ficar disponível para futuras reservas.

Também é possível remover uma reserva com a iniciativa do transmissor. Nesse caso, a reserva de identificador 1804289383 é cancelada com um envio de uma mensagem TEAR ao transmissor.

Erros na Reserva

Nos roteadores, a mensagem RESV é recebida, processada, e verificada por meio da soma de verificação se ocorreu um erro ou não. Também é verificado se é possível reservar a largura de banda ***L*** pedida conforme a fórmula abaixo:

$$L_{Disponível} + L_{Pedida} \leq L_{Total} \quad (1)$$

Caso não seja possível, é enviada uma mensagem NACK para a fonte descrevendo um erro de indisponibilidade de recursos. O transmissor irá receber uma mensagem ACK/NACK indicando um erro, exemplificada abaixo:

0x80512fcErro na mensagem
Mensagem RSVP: Resv Id: 1714636915
Header: Version: 0x2
Header: Flags: 0x5
Header: Types: 0xd
Header: CheckSum: 0x4f7e
Header: TTL: 0x2
Header: Reserved: 0x0
Header: Length: 40bytes
ACK: Code Desc: Error
ACK: Code:
ACK: Reason Desc: Recursos indisponiveis
ACK: Reason:
ACK: ToolBox Desc: Sem funcionalidade
ACK: ToolBox: 0
ACK: Service Desc: RSVP carga controlada e serviço garantido
ACK: Service Id: 1
SRV: ESPEC Banda:50 kbytes/s

V. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Um dos principais componentes em um sistema de comunicação com qualidade de serviço é o protocolo de sinalização. Ele realiza a interação das aplicações com a rede para reservar os recursos necessários para as aplicações tanto de usuários de uma rede fixa quanto aos usuários de uma rede móvel.

Esse artigo apresentou a modelagem, utilizando a linguagem de modelagem UML, e a implementação de um protótipo do protocolo RSVPv2 para uso em uma plataforma de rede real, a partir da sua especificação. O protótipo foi testado em um ambiente virtual que simula computadores e uma rede real.

A partir do trabalho apresentado podemos tomar diversas direções, com o objetivo de adicionar novos serviços fornecidos pelo protótipo além da garantia de largura de banda, como estabelecer um atraso máximo entre a fonte e o receptor e um nível mínimo de perda de pacotes de acordo com a aplicação.

Outras possibilidades têm como base os eventos de mudança de sub-rede e a ocorrência de *handoff*. Nessas situações é sugerida a criação de uma rotina de sistema (*system call*) para quando a máquina obter um novo endereço IP, seja sinalizado de alguma forma, como por exemplo, utilizar memória compartilhada para informar que o endereço IP e consequentemente a rede mudou.

Desta forma, bastaria enviar uma nova mensagem com o mesmo identificador para o destino, para os roteadores garantindo a reserva de recursos. No caso de um roteador não pertencer mais ao caminho de roteamento do fluxo de dados, os recursos seriam liberados com a ocorrência de *timeout.*

REFERÊNCIAS

[1] X. Fu, “Development of QoS Signaling Protocols in Internet”, IEEE Conference on Local Computer Networks (LCN), Workshop on High-Speed Local Networks, p. 636-637, 2003.

[2] R. Braden, L. Zhang, S. Berson; S. Herzog, S. Jamin, “Resource reSerVation Protocol (RSVP)” - versão 1 - especificação funcional, RFC 2205, 1997.

[3] P. C. S. Vidal, P. G. Ziemer, "Protocolos de Sinalização de Qualidade de Serviço em Redes Móveis na *Internet*", Mini-curso, XXII Simpósio Brasileiro de Telecomunicações (SBrT), 2005.

[4] J. Manner, X. Fu, “Analysis of Existing Quality-of-Service Signaling Protocols”. IETF, RFC 4094 – Informational, 2005.

[5] NSIS. “Next Steps in Signaling – NSIS”. Working Group, IETF.

[6] M. Brunner, R. Greco, L. Delgrossi, “Towards RSVP Version 2”, Lecture Notes In Computer Science archive Proceedings of the Second International Workshop on Quality of Service in Multiservice IP Networks, p. 704-716, 2003.

[7] P. C. S. Vidal, P. G. Ziemer, **“**Extension and Evaluation of the RSVP version 2 Protocol in Infra-Structured Mobile Networks”. 4th Latin American Network Operations and Management Symposium (LANOMS), 2005.

[8] H. Tschofenig, et al. “Beyond QoS Signaling: a Generic IP Signaling Framework”. Computer networks, 50:3416 - 3433, dezembro de 2006.

[9] D. Hogrefe et al. “Overhead and Performance Study of the General Internet Signaling Transport (GIST) Protocol”. ACM/IEEE Transactions on Networking, 17(1): 158-171, fevereiro de 2009.

[10] G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, 1999.