



Estudo de QoS IP sobre redes ATM

Gustavo Bittencourt Figueiredo <guto@ufba.br>

Daniel Macêdo Batista <danielmb@ufba.br>

Mercia Eliane Bittencourt Figueiredo <mercia@ufba.br>

Projeto REMAV-Salvador

Universidade Federal da Bahia

1. Introdução
 2. IntServ X DiffServ
 3. Experimentos realizados
 4. Conclusão
- Referências bibliográficas

1. Introdução

A enorme quantidade de informações, em forma de voz, vídeo, imagem e dados, que diariamente trafegam pela Internet, vem fazendo com que o suporte à qualidade de serviço torne-se indispensável na mesma. Neste sentido, estudos estão sendo dirigidos com o objetivo de fazer com que as informações enviadas pela rede sejam entregues com uma qualidade igual ou superior às exigidas pelas aplicações.

A tecnologia ATM provê qualidade de serviço nativamente, porém esse benefício é subutilizado, pois a grande maioria das aplicações, atualmente, é desenvolvida para redes TCP/IP. Por conta disso, o objetivo deste estudo é verificar a viabilidade do uso de QoS IP na infra-estrutura do Projeto Rema.

Existem dois enfoques principais em que os estudos se concentram: os serviços integrados e os serviços diferenciados. Na verdade, há um embate entre estas duas opções que, em última análise, é a retomada da velha discussão entre protocolos orientados ou não orientados a conexão. A propostas dos serviços integrados é baseada no paradigma orientado a conexão, utilizando a reserva de recursos como principal mecanismo para a provisão de qualidade de serviço.

Contudo, teme-se que a demanda crescente por reservas de *buffers* e largura de banda provoque escassez de recursos, além de exigir memorização de uma quantidade enorme de estados de sinalização o que culminaria na chamada explosão dos estados de sinalização, onde os equipamentos não teriam mais como armazenar todos os estados de sinalização, provocando assim um "crash" na rede.

Como alternativa aos serviços integrados, e provendo solução para o problema de explosão dos estados de sinalização, está a abordagem dos serviços diferenciados. Neste paradigma, não há nenhuma espécie de reserva de recursos, sendo a indicação de prioridade feita no campo ToS dos pacotes. Na prática, apenas 6 bits dos 8 bits presentes no campo ToS são utilizados. Isto corresponde a um número de 64 diferentes possibilidades de classes de serviço.

Ao ingressar em um domínio que implementa o diffserv, os pacotes terão o campo ToS, neste contexto chamado de DS (Differentiated Service) de acordo com o nível de serviço pretendido. Em cada nó do domínio os pacotes são

Nesta edição:

Netplus - um ambiente para gerência de QoS baseado na web

Uma proposta de infra-estrutura para a medição passiva de tráfego na rede RNP2

Experiência de transmissão multicast inter-domínio no XIX SBRC

Estudo de QoS IP sobre redes ATM

Especificação e implementação do DynaVideo VoD

HealthNet: um sistema integrado de telediagnóstico e segunda opinião médica

O desenvolvimento de um centro cultural virtual na Internet

Co-autoria de aulas na Rede Metropolitana de Alta Velocidade de Recife

Implementação de ambiente heterogêneo de telefonia IP

Bookmarks

Cartas

NewsGeneration:

Artigos publicados

Autores

FAQ

Assine

Página inicial

analisados um a um, e tratados de acordo com a indicação do campo DS, caracterizando assim um comportamento por nó (*Per Hop Behavior*- PHB).

^

2. IntServ X DiffServ

O RSVP é um protocolo de sinalização que gerencia as informações na arquitetura IntServ e que as aplicações usam para requisitar os recursos de rede e receber a resposta de aceitação ou negação dessas requisições. Uma característica importante do RSVP é que ele usa requisições de reserva orientadas ao receptor, o que o torna ideal para grupos *multicast* muito grandes, pois cada *host* receptor teria a qualidade de acordo com as suas necessidades e condições [1].

O RSVP irá sinalizar os requerimentos de recursos para cada fluxo nos elementos de rede, usando parâmetros do IntServ. Ele atua nos *hosts* gerando a requisição de QoS específica para cada fluxo de dados e, nos roteadores, tem a função de transmitir essa requisição para todos os nós do caminho percorrido pelo fluxo, além de estabelecer e manter o estado, provendo, dessa forma, a QoS requisitada[2].

A reserva em cada nó é feita através da comunicação de um *daemon* com dois módulos locais de decisão: o controle de admissão e o controle de política. O controle de admissão determina se o nó tem recursos suficientes disponíveis para a QoS requisitada e o controle de política determina se o usuário tem permissões administrativas para obter a reserva (confrontando a requisição contra as políticas armazenadas em uma base de dados de políticas). Se alguma dessas duas verificações falhar, o *daemon* retorna um erro para a aplicação no *host* que solicitou a requisição; já se as duas verificações ocorrerem sem problemas, o *daemon* configura os parâmetros requisitados em um classificador de pacotes e em um escalonador de pacotes. O classificador tem a função de determinar a classe de QoS para cada pacote enquanto que o organizador ordena os pacotes para a transmissão, conseguindo dessa forma a QoS solicitada para cada fluxo em particular.

Ao contrário da orientação por fluxo utilizada no RSVP, as redes baseadas em DiffServ classificam os pacotes dentro de um pequeno número de fluxos agregados ou classes, baseando-se no DSCP (*Differentiated Service Code Point*), indicado no campo do cabeçalho ToS (*Type of Service*) do pacote IP [7]. Esse método de classificação é chamado classificação BA (*Behaviour aggregate*)[6].

Em cada roteador DiffServ, os pacotes são sujeitos a um PHB, o qual é invocado pelo DSCP. Ele pode ser implementado determinando o percentual de utilização da largura de banda em um enlace de saída. Quando o roteador não consegue atender a classe desejada, passa para a classe seguinte. O primeiro benefício com isso tudo é a escalabilidade, pois o DiffServ elimina a necessidade do processamento por fluxo e dessa forma se encaixa perfeitamente em redes de grande porte.

Mas para que isso funcione corretamente, deve-se efetuar a marcação do DSCP nos cabeçalhos dos pacotes e há duas formas para isso ser feito: *host marking* e *router marking*. No primeiro caso, o sistema operacional deve marcar o DSCP nos pacotes transmitidos. A vantagem é que o sistema operacional do *host*, sem dúvida, é melhor em determinar a importância dos pacotes do que os demais elementos da rede. No segundo caso, os roteadores são configurados para identificar cada tráfego específico e para marcar o DSCP nos pacotes que os atravessam. Isso pode ser feito de forma dinâmica ou estática, através de uma configuração manual ou através de scripts.

^

3. Experimentos realizados

Para os testes foram utilizadas duas máquinas (zeus e amazonia), rodando linux Slackware 7.1, na mesma rede enviando pacotes para uma máquina (cpd1), rodando linux Redhat 7.0, em outra rede. O *gateway default* para zeus e amazonia foi a máquina gandalf enquanto que para cpd1 foi a máquina poseidon. Ambos os *gateways* rodavam linux Redhat 7.0, o kernel para todas as máquinas foi o 2.4.1-ac10. A placa de rede de todas as máquinas foi a IBM Turboways 25, utilizando o driver it25 versão 0.78a [8]. Poseidon e gandalf possuem processador celeron de 466 Mhz e 64MB de memória RAM enquanto que as demais (amazonia, zeus e cpd1) têm processador Pentium III de 500MHz e 128MB de memória RAM.

Para verificar o funcionamento do RSVP e do DiffServ, fizemos testes utilizando as mesmas taxas de envio de pacotes amazonia mandando a 4Mbps e zeus mandando a 16Mbps. Em um primeiro momento, os fluxos foram transmitidos sem nenhuma qualidade (melhor esforço). No segundo momento, amazonia passou a mandar o fluxo utilizando RSVP e, por último, o teste foi realizado com DiffServ nos roteadores.

Nos três testes, amazonia gerava o tráfego para o qual queríamos QoS e zeus funcionou como a máquina que gerava o tráfego concorrente. Os programas utilizados para geração de fluxo e coleta dos dados foram o mgen, drec e mcalc, presentes no pacote MGEN versão 3.2a1[11]. No teste com RSVP, a implementação utilizada nos *hosts* foi o RSVP da ISI versão 4.2a3[9] e, nos roteadores, o RSVP da KOM versão 2.1.1 [10].

Para garantir a largura de banda nos testes com DiffServ, utilizamos o programa tc, presente no pacote iproute versão 2.2.4[12]. Gandalf foi configurado como roteador de ingresso, responsável pela marcação do campo DS, e poseidon como roteador de saída. Em ambos roteadores, foram utilizadas duas classes de serviço. Uma classe (AF) para o tráfego originado em amazonia e com destino para porta 5008, policiamento de taxa de 5Mbps e a outra classe melhor esforço (BE) para o restante do tráfego, policiada em 10Mbps. Em todos os testes o tamanho do pacote ficou fixo em 512 bytes.

Outros testes foram feitos para verificar a taxa de perda de pacotes nos roteadores e máquinas-fim. O fluxo em amazonia foi mantido constante a 4Mbps e variamos o fluxo de zeus de 8Mbps até 16Mbps (de 2 em 2 Mbps), ambos em melhor esforço para descobrirmos qual a maior taxa de interferência com o percentual menor de perda.

Apesar de termos instalado o *daemon* do RSVP nos roteadores, verificamos que o mesmo não tinha suporte para controle de tráfego implementado para o linux, por isso todos os testes feitos com RSVP se mostraram muito próximos dos testes em melhor esforço e por isso eles foram excluídos dos gráficos abaixo:

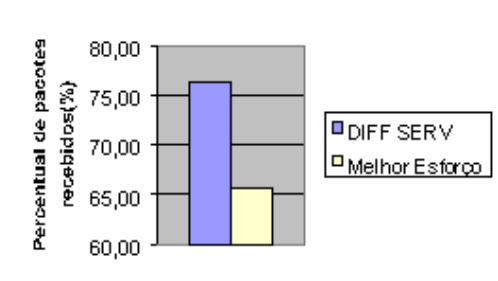


Figura 1 - Comparação do percentual de pacotes recebidos entre os testes com diffServ e "melhor esforço"

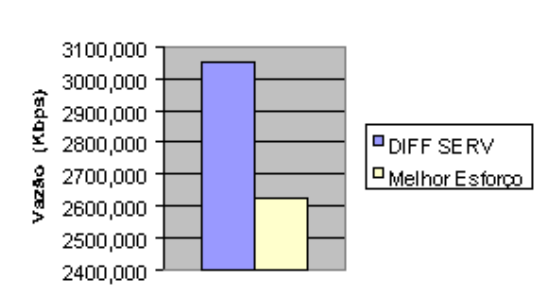


Figura 2 - Comparação da vazão entre os testes com diffServ e "melhor esforço"

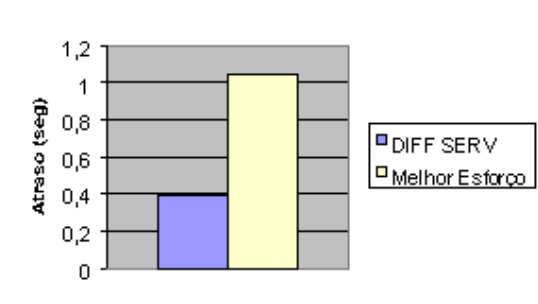


Figura 3 - Comparação do atraso entre os testes de diffserv e "melhor esforço"

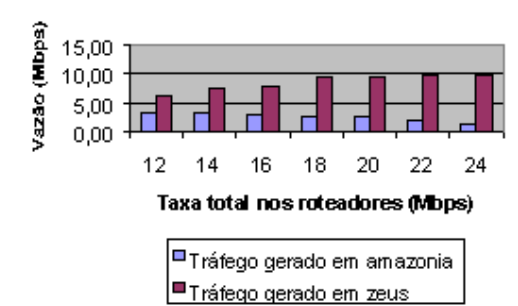


Figura 4 - Comparação da vazão em testes realizados com "melhor esforço", variando-se a taxa de envio do tráfego de concorrência

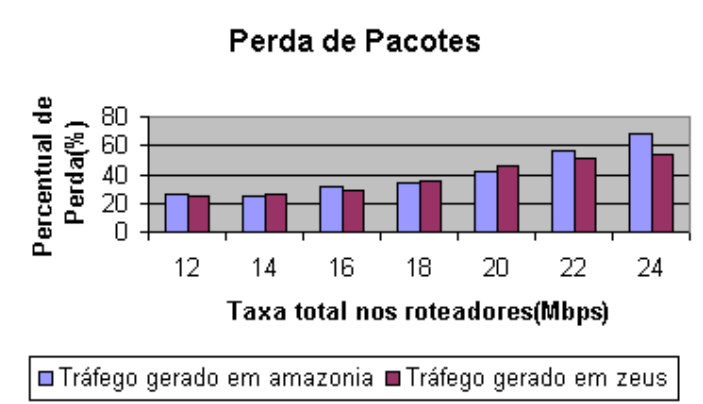


Figura 5 - Comparação da perda de pacotes em testes realizados com "melhor esforço", variando-se a taxa de envio do tráfego de concorrência

^

4. Conclusão

Pelos gráficos, podemos notar que houve uma melhora significativa no retardo e vazão dos pacotes, comparando-se os testes com DiffServ e com melhor esforço. Houve, também, uma diminuição das perdas, o que justifica a utilização dessa arquitetura na rede com o objetivo de conseguir a QoS desejada.

Não podemos concluir nada a respeito do RSVP, já que o *daemon* que rodamos no roteador não tinha suporte a controle de tráfego. Pretendemos dar continuidade a estes assim que obtivermos uma implementação de RSVP adequada.

^

Referências bibliográficas

- [1] RSVP Protocol Overview - <http://www.isi.edu/div7/rsvp/overview.html>
- [2] Zhang,L., et al., RFC 2205 - Resource ReSerVation Protocol (RSVP) -Version 1, Functional Specification
- [3] Mankin, A., et al., RFC 2208 - Resource ReSerVation Protocol (RSVP) -Version 1, Applicability Statement Some Guidelines on Deployment
- [4] Braden, R., Zhang,L. RFC 2209 - Resource ReSerVation Protocol (RSVP) - Version 1, Message Processing Rules
- [5] Bernet,Y. Et al., RFC 2998 - A Framework for Integrated Services Operation over Diffserv Networks
- [6] Bernet,Y., et al., A Conceptual Model for Diffserv Routers
<http://www.globecom.net/ietf/draft/draft-ietf-diffserv-model-00.html>
- [7] Barbosa, Hilza Miranda, Loureiro, Antônio Alfredo Ferreira - Mapeando Classes de Serviços Diferenciados para Categorias de Serviços ATM, Hilza Miranda Barbosa, Loureiro <http://www.dcc.ufmg.br/~hilza/hilzaposfinal.htm>
- [8] Westall, Mike - ATM Networking in Linux -
<http://www.cs.clemson.edu/~westall/atm/atm.htm>
- [9] RSVP Software - <http://www.isi.edu/div7/rsvp/release.html>
- [10] KOM RSVP Engine - <http://www.kom.e-technik.tu-darmstadt.de/rsvp/>
- [11] ManiMac Experimental Internet Application site -
<http://manimac.itd.nrl.navy.mil/>
- [12] < <ftp://ftp.inr.ac.ru/ip-routing/iproute2-current.tar.gz> >

^

**NewsGeneration, um serviço oferecido pela
RNP – Rede Nacional de Ensino e Pesquisa**

Copyright © RNP, 1997 – 2004