

QoS Fim a Fim Através da Combinação entre Serviços Integrados e Serviços Diferenciados

Antônio Jorge G. Abelém^{1, ‡}

Michael A. Stanton[†]

Noemi Rodriguez[‡]

‡- Depto. de Informática – DI
Pontifícia Universidade Católica - PUC-Rio
Rua Marquês de São Vicente, 225
Cep: 22453-900 Gávea – Rio de Janeiro – RJ
e-mail : abelem.noemi@inf.puc-rio.br

†- Instituto de Computação – IC
Universidade Federal Fluminense - UFF
Rua Passo da Pátria, 156, bloco E, sala 350
Cep: 24210-240 Niterói – RJ
e-mail : michael@ic.uff.br

PUC-RioInf.MCC06/01 Março, 2001.

Abstract

Several alternatives for offering end-to-end QoS in IP internetworks have been proposed over the last few years. Recently, researchers of the IETF (“Internet Engineering Task Force”) have proposed a model which combines the integrated services/RSVP (intserv) and differentiated services (diffserv) architectures. In this scheme, the border routers between one region and another play a fundamental role, as they are responsible for mapping QoS functionality between the different regions. In this article, an agent for NS simulator (Network Simulator) is presented, which was developed to perform tasks in support of this mapping, which include the choice of a suitable Per-Hop Behavior (PHB), appropriate policing at the border of the diffserv region, and admission control in accordance with the resources available in the diffserv regions.

Keywords: IP internetworks, end-to-end QoS, integrated services, differentiated services, border routers.

Resumo

Diversas propostas para oferecer QoS fim a fim nas inter-redes IP têm sido apresentadas nos últimos anos. Recentemente, um grupo de pesquisa da IETF (“Internet Engineering Task Force”) propôs um modelo que combina a arquitetura de serviços integrados/RSVP com a arquitetura de serviços diferenciados. Nesse esquema os roteadores de fronteira entre uma região e outra desempenham papel fundamental, pois são eles os responsáveis pelo mapeamento das capacidades entre as diferentes regiões. Este artigo apresenta um agente, desenvolvido para o simulador de redes NS (“Network Simulator”), que realiza as funções envolvidas nesse mapeamento entre as regiões de serviços integrados e de serviços diferenciados, que, entre outras tarefas, inclui a escolha de um comportamento agregado por nó (“PHB-Per-Hop Behavior”) adequado, um policiamento apropriado nas extremidades das redes diffserv e a realização de um controle de admissão de acordo com os recursos disponíveis nas regiões de serviços diferenciados.

Palavras-Chave: inter-redes IP, QoS fim a fim, serviços integrados, serviços diferenciados, roteadores de fronteira.

¹ - Professor do Departamento de Informática da Universidade Federal do Pará (UFPA), atualmente licenciado para cursar Doutorado no Depto. de Informática da PUC-Rio, através do Programa Institucional de Capacitação Docente e Técnica (PICDT) da CAPES.

1. Introdução

Tradicionalmente, as inter-redes IP têm oferecido serviços chamados de melhor esforço (“best-effort”). Neste esquema, todos os pacotes dos usuários competem igualmente pelos recursos da rede e nenhuma garantia é dada sobre quando ou se os pacotes vão chegar ao destino. Entretanto, com o crescimento acentuado da Internet nos últimos anos e o surgimento de novas classes de aplicações distribuídas, com grande necessidade de largura de banda e restrições de retardo, como por exemplo, videoconferência, distribuição de vídeo e voz sobre IP, outras classes de serviços com diferentes prioridades e certas garantias de disponibilidade de recursos da rede passaram a ser demandas pelos usuários da grande rede.

Para atender as necessidades destas novas aplicações, o tipo de serviço oferecido pelas inter-redes IP precisa ser refinado, essencialmente, em dois importantes aspectos: primeiro ele deve ser capaz de controlar o desempenho do perfil da rede de modo a atender as necessidades das aplicações; segundo, ele deve oferecer um razoável espectro de perfis para atender os diferentes tipos de aplicações existentes. A estes requisitos tem se dado o nome de qualidade de serviço (QoS), apesar de ser melhor descrito como uma engenharia de desempenho, onde a rede pode ser ajustada para oferecer desempenhos específicos para diferentes classes de serviços [HUS00].

Diversas propostas para oferecer QoS fim a fim nas inter-redes IP têm sido apresentadas nos últimos anos [BRA94], [BLA98], [CRA98], [RAJ99], [APO99]. Recentemente, Bernet et al. [BER00] propuseram um modelo que combina a arquitetura de serviços integrados/RSVP [BRA94] com a arquitetura de serviços diferenciados [BLA98]. A proposta, que tem recebido especial atenção da IETF (“Internet Engineering Task Force”) em função de aliar escalabilidade à robustez, baseia-se em uma inter-rede referência consistindo de uma ou mais regiões que oferecem serviços diferenciados (diffserv) no meio de uma grande rede que suporta serviços integrados/RSVP (intserv) fim a fim. De modo simplificado, procura-se utilizar a arquitetura diffserv nas redes centrais, chamadas redes de trânsito, enquanto emprega-se a arquitetura intserv nas redes das extremidades, dos consumidores (ver Figura 1).

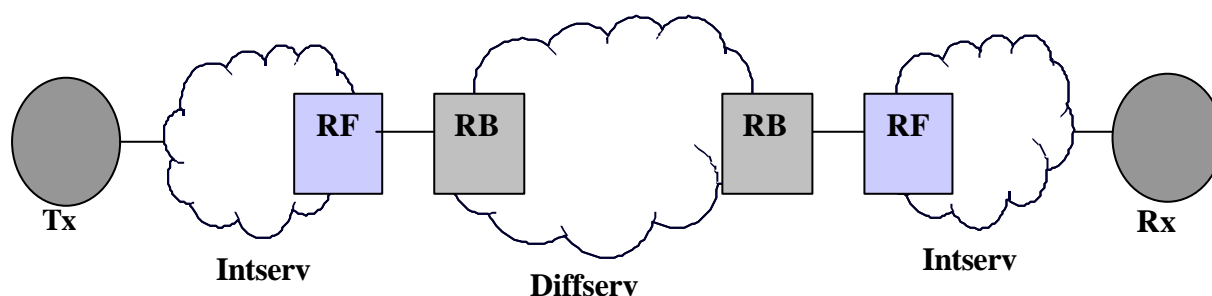


Figura 1 – Modelo combinando as arquiteturas intserv e diffserv.

Nesse referido esquema os roteadores de fronteira (RF e RB) entre uma região e outra desempenham papel fundamental, pois são eles os responsáveis pelo mapeamento das capacidades entre as diferentes regiões. Entre os vários aspectos desse mapeamento, pode-se destacar: a escolha de um comportamento agregado por nó (PHB) adequado, um policiamento apropriado nas extremidades das redes diffserv, e a realização de um controle de admissão de acordo com os recursos disponíveis nas regiões de serviços diferenciados.

Algumas propostas teóricas, como as de Xiao [XIA99] e a de Rajan [RAJ99], têm sido apresentadas sugerindo aperfeiçoamentos ou mesmo variações para o modelo apresentado por

Bernet et al. Este artigo apresenta um agente², desenvolvido para o ambiente de simulação NS (“Network Simulator”) [FAL00], capaz de realizar a integração entre os serviços integrados e diferenciados, desempenhando as funções de mapeamento executadas pelos roteadores de fronteira entre as duas regiões. Desta forma, procura-se neste trabalho explorar os aspectos de implementação do modelo, através da simulação de diversas situações onde a atuação desses roteadores de fronteira é determinante.

Além desta seção introdutória, este artigo é composto de mais quatro seções. Na seção 2 apresenta-se, resumidamente, os fundamentos teóricos das arquiteturas que serviram de base para o trabalho. Na seção 3 detalha-se o comportamento do agente implementado. Na seção 4 apresenta-se alguns resultados experimentais. Na seção 5 apresenta-se as conclusões e sugere-se possíveis trabalhos futuros.

2. Arquiteturas para Qualidade de Serviço

Nos últimos anos, intensas atividades de pesquisa têm sido dedicadas ao desenvolvimento de novas arquiteturas de redes e novos modelos de serviços que incrementem os serviços oferecidos pelas inter-redes IP [AWD98], [XIA99], [RAJ99], [XIA00]. A IETF (“Internet Engineering Task Force”) tem proposto diversos modelos e mecanismos procurando oferecer serviços com QoS para as inter-redes IP, entre os quais pode-se destacar: a arquitetura de serviços integrados/RSVP [BRA94], [BRA97], a arquitetura de serviços diferenciados [BLA98], [JAC99], o MPLS (“Multiprotocol Label Switching”) [ROS01], [VAN98], a engenharia de tráfego [AWD98], [AWD98B] e o roteamento baseado em restrições [CRA98], [APO99]. Além disso, como cada uma das propostas acima possui virtudes e problemas, várias abordagens têm sido apresentadas tentando combiná-las de forma a otimizar, ou pelo menos melhorar, o desempenho que estas teriam individualmente [GUE97], [AWD98B], [LI98], [BER00]. Nesta seção serão apresentadas, resumidamente, as propostas da IETF que foram usadas no desenvolvimento deste trabalho.

2.1. Fornecendo QoS através de Serviços Integrados/RSVP

A arquitetura de serviços integrados (intserv) propõe um conjunto de extensões ao tradicional modelo de melhor esforço das inter-redes IP com o objetivo de oferecer qualidade de serviço (QoS) fim a fim para as aplicações, tanto através da comunicação um para um (“unicast”) como da comunicação multiponto (“multicast”). Em adição ao serviço de melhor esforço, o modelo sugere mais duas classes de serviços:

- Garantido – Para aplicações que necessitam de limites de retardos fixos [SCH97].
- Carga controlada – Para aplicações que requerem serviços semelhantes aos oferecidos pela classe de melhor esforço em condições de tráfego sem congestionamento [WRO97].

A filosofia deste modelo é baseada em fluxos e supõe que seja necessário que os roteadores reservem recursos de modo a fornecer alguma garantia de QoS para os fluxos dos usuários. As reservas de recursos são feitas de acordo com as requisições dos receptores. Um fluxo de dados identifica um conjunto de pacotes que receberá tratamento especial. Esses fluxos são definidos por sessão, as quais, por sua vez, são identificadas por uma especificação genérica, contendo o endereço IP, o protocolo de transporte utilizado e o número da porta do destino, juntamente com uma lista de emissores para aquela sessão. Cada

² - O termo agente é empregado neste trabalho com o mesmo sentido usado no ambiente de simulação NS (“Network Simulator”), onde agentes são “endpoints” nos quais os pacotes de rede são construídos e consumidos, como também são usados para implementação de protocolos em diversas camadas.

emissor é identificado pelo seu endereço IP e o número da porta, enquanto o protocolo de transporte que ele utiliza deve ser o mesmo definido na identificação da sessão [BRA94].

A arquitetura supõe também que, para realizar essas reservas, algum mecanismo explícito de sinalização que leve as informações aos roteadores seja utilizado. O protocolo mais conhecido para este propósito, apesar da arquitetura poder acomodar outros, é o RSVP (“Resource Reservation Protocol”) [BRA97], o qual é independente da arquitetura intserv. O RSVP é um protocolo de sinalização, que pode transportar informações intserv [WRO97b], enquanto a arquitetura de serviços integrados define os modelos para expressar tipos de serviços, quantificar as necessidades de recursos, e determinar a disponibilidade dos recursos requisitados nos elementos relevantes da rede (controle de admissão).

O RSVP, como dito anteriormente, é um protocolo de sinalização que as aplicações podem utilizar para requisitar recursos da rede. A abordagem do protocolo é baseada no receptor, de modo que são os receptores que escolhem o nível de recursos a serem reservados. A rede responde explicitamente, admitindo ou rejeitando essas requisições. As aplicações que possuem necessidades quantificadas de recursos podem expressar tais necessidades através de parâmetros, como definido nas especificações dos respectivos serviços oferecidos pela arquitetura intserv [SCH97], [WRO97].

No processo de sinalização, ilustrado na Figura 2, o emissor (S1) envia uma mensagem PATH para um ou mais potenciais receptores (RCV1, RCV2 e RCV3), especificando as características do tráfego. Após receber a mensagem PATH, os receptores interessados respondem com uma mensagem RESV, requisitando os recursos desejados. Cada nó intermediário ao longo do caminho pode rejeitar ou aceitar a requisição. Se a requisição for rejeitada, o roteador irá enviar uma mensagem de erro, RESVERR, para o receptor e o processo de sinalização é finalizado. Se a requisição é aceita, a largura de banda e espaço em buffer necessários são alocados para o fluxo e as informações relacionadas ao fluxo são instaladas no roteador.

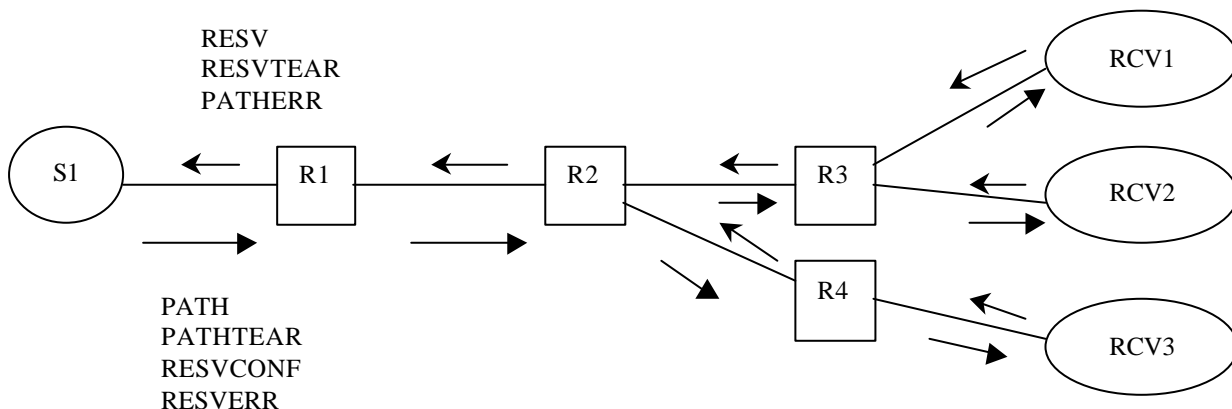


Figura 2 – Processo de sinalização RSVP

A arquitetura de serviços integrados/RSVP representa uma mudança fundamental na modelo da Internet, o qual é baseado na premissa de que todas as informações de estado relacionadas ao fluxo devem ficar nas extremidades, nos sistemas dos usuários finais [CLA88]. No entanto, a arquitetura de serviços integrados/RSVP é considerada excessivamente complexa e pouco escalável [HUS00]. Os principais problemas existentes com ela são a quantidade de informações de estado em cada roteador, que cresce proporcionalmente ao número de fluxos e a carga excessiva sobre os roteadores. No primeiro ponto, para cada novo fluxo que solicita reservas, informações de estado PATH e RESV devem ser criadas em cada roteador ao longo do caminho, o que exige capacidade extra de

armazenamento e processamento nos roteadores. No segundo ponto, como cada roteador ao longo do caminho pode aceitar ou negar a solicitação de reservas, todos eles devem ser aptos a processar RSVP, realizar controle de admissão, classificar microfluxos e escalonar pacotes, o que acaba por sobrecarregar os mesmos.

2.2. Fornecendo QoS através de Serviços Diferenciados

Em função dos problemas existentes com a arquitetura de serviços integrados/RSVP, principalmente no que diz respeito à escalabilidade e à complexidade de implementação, uma outra proposta ganhou força, a arquitetura de serviços diferenciados [BLA98], [JAC99]. Em contraste com a orientação por fluxos do RSVP, as redes de serviços diferenciados (diffserv) foram projetadas para oferecer classes de serviços agregados, onde os fluxos individuais são agrupados e tratados pela rede de acordo com o classe de serviços em que se encontram.

Para poder identificar a classe de um pacote, as redes diffserv utilizam o campo chamado DiffServ (DS), conhecido anteriormente como tipo de serviço (ToS-“Type of Service”), e situado dentro do cabeçalho dos pacotes IP [NIC98]. Desta forma, um novo tráfego que chega a uma rede diffserv é primeiro classificado, em seguida passa por um tipo de “filtro de admissão”, com o intuito de moldá-lo de acordo com a política de controle associada com aquela classificação. Este fluxo de tráfego é então atribuído a um comportamento agregado específico, através da (e)marcação apropriada do campo DS. A partir disso, em cada roteador diffserv intermediário (ver Figura 3) os pacotes são sujeitos a um tratamento específico, chamado comportamento agregado por nó (PHB-“Per Hop Behavior”), o qual dependerá do valor do campo DS dos pacotes [JAC99b], [HEI99].

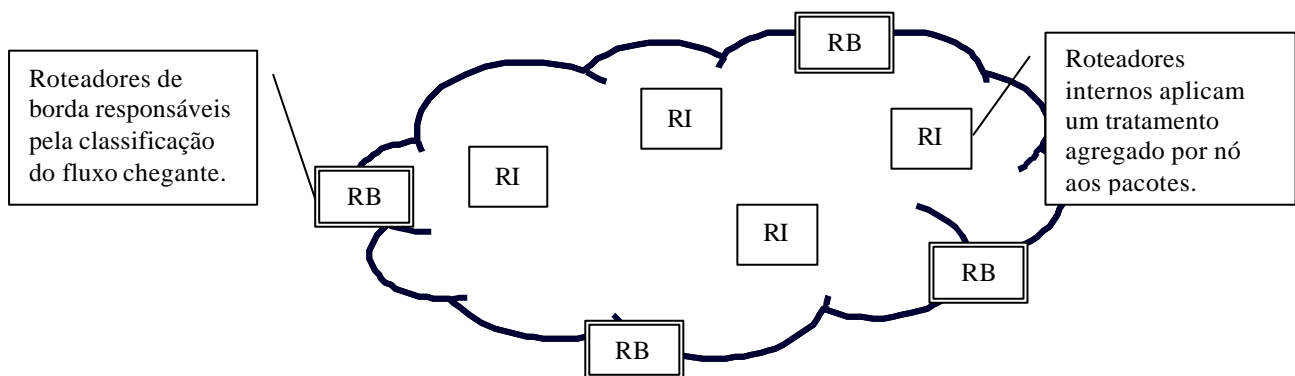


Figura 3 – A arquitetura de serviços diferenciados.

Para um consumidor poder usufruir dos serviços diferenciados, a idéia é que ele deve definir um contrato com o seu provedor de serviços Internet (ISP), chamado de acordo do nível de serviço (ou “SLA- Service Level Agreement”). Os SLAs especificam basicamente as classes de serviço suportadas e a quantidade de tráfego permitida por classe. Além disso, os SLAs podem ser estáticos ou dinâmicos. Os estáticos são negociados de forma regular (por exemplo, mensal, anual), enquanto os dinâmicos devem usar um protocolo de sinalização (por exemplo, RSVP) para requisitar os serviços sob demanda. Definido o SLA entre o consumidor e o ISP, os pacotes oriundos da rede do consumidor devem vir com o campo DS marcado de acordo com a classe desejada, podendo esta marcação ser feita tanto pela aplicação do consumidor como pelo roteador de borda da rede [BLA98]. Ao ingressar na rede de serviços diferenciados os pacotes, como dito antes, são classificados, policiados e

possivelmente moldados de acordo com o SLA. Quando um pacote passa de um domínio para outro, seu campo DS pode ser remarcado dependendo do que foi determinado pelo SLA entre os domínios.

A arquitetura de serviços diferenciados é significativamente diferente da de serviços integrados. Primeiro, porque existe um número limitado de classes de serviço, indicadas pelo campo DS. Como os serviços são alocados de acordo com as classes, a quantidade de informações de estado é proporcional ao número de classes ao invés do número de fluxos, fazendo com que a arquitetura de serviços diferenciados seja mais escalável. Segundo porque nela a classificação, a marcação, o policiamento e a moldagem são tarefas necessárias apenas nos elementos de borda das redes, tornando a arquitetura mais simples e fácil de desenvolver. No entanto, apesar de mais escalável em função de sua reduzida complexidade, a arquitetura de serviços diferenciados também possui alguns problemas, os quais são atribuídos justamente ao excesso de simplicidade desta [HUS00]. Entre as principais críticas à arquitetura diffserv, pode-se destacar: a dificuldade de uma rede ficar sabendo das políticas de ingresso da rede vizinha quando o tráfego precisa ser passado por esta última, a dificuldade dos elementos de borda em aplicar técnicas de controle de admissão sem informações suficientes e atualizadas sobre a utilização dos recursos no seu respectivo domínio e a dificuldade de redes diffserv em atender as características heterogêneas dos grupos multiponto, entre outras.

2.3. Fornecendo QoS Combinando Serviços Integrados e Diferenciados

O modelo de serviços integrados, apesar de fornecer um alto nível de garantia de reserva de recursos por fluxo, tem sérios problemas de escalabilidade e complexidade. O modelo de serviços diferenciados, por sua vez, é simples e escalável, mas apresenta deficiências no gerenciamento de recursos. A observação de que nenhum dos dois modelos atende todos os requisitos para se fornecer uma garantia de QoS fim a fim tem propiciado o surgimento de propostas que combinam os dois modelos, usando os mecanismos de controle intserv nas redes de ponta, enquanto os serviços diferenciados são empregados nas redes centrais, também chamadas de redes de trânsito [XIA99], [RAJ99]. Estas propostas têm sido motivadas pela observação que alguma forma de negociação de QoS fim a fim é necessária para atender as necessidades das aplicações interativas e de grupo. O modelo adotado neste trabalho é proposto em [BER00], e baseia-se em uma rede de referência consistindo de uma ou mais regiões que oferecem serviços diferenciados (diffserv) no meio de uma grande rede que suporta serviços integrados/RSVP (intserv) fim a fim (ver Figura 4).

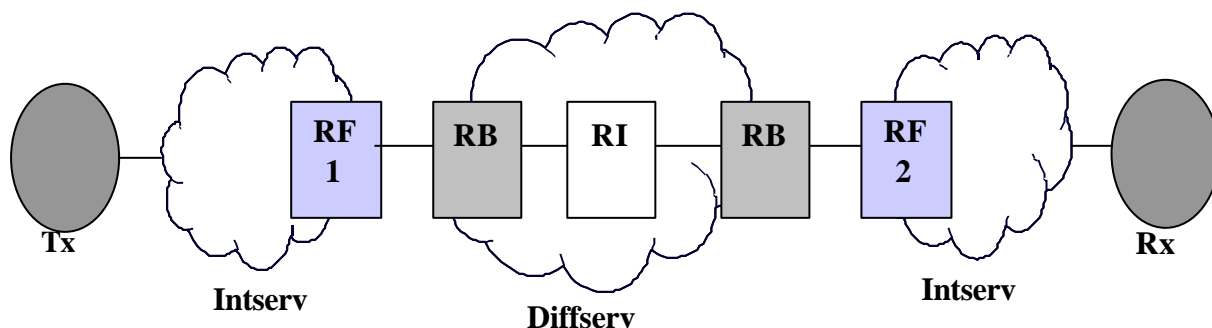


Figura 4 – Modelo combinando as arquiteturas intserv e diffserv.

A idéia é que as aplicações usem o protocolo RSVP para requisitar a admissão de seus fluxos na rede. Se a requisição for aceita, significa que existem recursos disponíveis nas redes intserv, bem como que os serviços requisitados foram mapeados em uma classe de serviços compatível dentro da rede diffserv. Os elementos da rede diffserv devem ser capazes de transportar as mensagens RSVP até a rede intserv na outra extremidade, de modo que a reserva de recursos possa ser feita dentro desta última. As redes diffserv podem, mas não são obrigadas, a participar do processo de sinalização fim a fim, baseado no RSVP.

Do ponto de vista de serviços integrados, as redes diffserv são tratadas como enlaces virtuais conectando elementos intserv [SEA00]. Dentro das redes diffserv, os elementos implementam controle de tráfego agregado por classe. A quantidade de tráfego que será admitida nas redes diffserv poderá ser controlada através de policiamento nos elementos de entrada da rede.

O modelo proposto por Bernet et al. considera dois casos específicos: no primeiro, os recursos dentro da região diffserv são distribuídos estatisticamente e estas regiões não possuem dispositivos aptos a processar RSVP. No segundo, os recursos dentro das regiões diffserv são alocados dinamicamente e os dispositivos dentro dessas regiões participam da sinalização RSVP.

Para que o esquema de sinalização intserv/RSVP seja suportado, as mensagens de sinalização RSVP devem viajar de uma extremidade a outra das redes, entre os emissores e os receptores. Desta forma, as mensagens RSVP, no mínimo, devem ser transportadas através das regiões diffserv. Dependendo do tipo de modelo adotado, as mensagens RSVP podem ou não ser processadas nessas regiões.

Os roteadores de fronteira das redes intserv, denominados RF (ver Figura 4), têm funcionalidades que variam dependendo do tipo de esquema adotado. No caso onde as regiões diffserv não processam RSVP, esses roteadores RF funcionam como agentes de controle de admissão das redes diffserv. Eles processam as mensagens de sinalização tanto dos emissores quanto dos receptores e aplicam o controle de admissão baseado na disponibilidade de recursos dentro da rede diffserv e nas políticas de alocação definidas com os consumidores. No caso em que as regiões diffserv processam RSVP, os roteadores RF aplicam o controle de admissão baseado na disponibilidade de recursos locais e na política de alocação definida com os consumidores. Neste último caso, são os roteadores de borda das redes diffserv, chamados RB, que executam o controle de admissão (ver Figura 4).

Os roteadores RB, situados nas fronteiras das regiões diffserv, também possuem funcionalidades que variam de acordo com o modelo adotado para a rede. Se as regiões diffserv não são aptas para processar RSVP, então esses roteadores RB agem como roteadores diffserv puros, ou seja, eles apenas policiam o tráfego baseado no nível de serviço especificado no campo DS e no contrato negociado com o consumidor. No entanto, se as regiões diffserv são aptas para processar RSVP, os roteadores RB participam da sinalização RSVP e atuam como agentes de controle de admissão para as regiões diffserv.

De uma forma geral, se observa que as diversas propostas existentes combinando serviços integrados e serviços diferenciados consideram estas duas arquiteturas como sendo complementares no propósito de oferecer QoS fim a fim [HUS00]. Juntas, acredita-se que elas podem facilitar o desenvolvimento das novas aplicações com grandes necessidades de largura de banda e restrições de retardo. Isso ocorre porque a arquitetura intserv habilita os computadores a requisitarem recursos quantificados por fluxo ao longo de caminhos fim a fim, com respostas de aceitação ou não das requisições, enquanto a arquitetura diffserv oferece a escalabilidade para as redes de backbone.

3. Agente para Integração entre Serviços

Como mencionado anteriormente (seção 1), o agente desenvolvido neste trabalho foi implementado para o ambiente de simulação “NS-2” (“Network Simulator”) [FAL00]. O NS é um simulador de redes orientado a eventos desenvolvido dentro do projeto VINT [VIN97]. Ele segue uma estruturação orientada a objetos, tendo sido escrito em C++ e utilizando o interpretador OTcl³ [WET95] como “frontend”.

A versão do NS utilizada neste trabalho foi a 2.1b6. Utilizou-se também os “*patches*” desenvolvidos por Marc Greis [GRE99] para arquitetura de serviços integrados/RSVP e por Sean Murphy [MUR00] para arquitetura de serviços diferenciados, já que a versão do simulador utilizada não possuía tais funcionalidades incorporadas a ele. Além disso, como esses “*patches*” foram desenvolvidos separadamente, houve a necessidade de algumas adequações em seus respectivos códigos, como também nos arquivos de construção do simulador para que eles pudessem ficar operacionais conjuntamente no simulador.

No NS, agentes são “endpoints” onde os pacotes da camada de rede são criados ou consumidos, como também são usados para a implementação de protocolos de várias camadas. Para adicionar novos módulos, como por exemplo novos agentes, ou mesmo para alterar componentes já existentes no simulador, deve-se, antes de mais nada, ter conhecimento da hierarquia de classes do NS e saber onde cada parte do código está situada dentro da árvore de diretórios do simulador. Adquirido esses conhecimentos, os passos seguintes no processo de adição de novos módulos ao simulador pode ser resumidamente dividido em três etapas principais: determinar as necessidades para as novas funcionalidades, determinar a posição dos novos componentes dentro da hierarquia de classe do NS e, por fim, escrever o novo código e associá-lo com o simulador, o que envolve escrever códigos em C++ e em OTcl.

No caso do agente apresentado neste artigo, além de executar as funções de um agente RSVP/intserv padrão, ele deve realizar o mapeamento das capacidades da região intserv para a região diffserv. Em função disso, a estratégia adotada para a criação do nosso agente foi criar uma nova classe posicionando-a abaixo da classe do agente intserv/RSVP, desenvolvido por Marc Greis, dentro da hierarquia de classes do NS, pois desta forma o novo agente herdaria todas as funcionalidades de sua classe pai. A esta nova classe adicionou-se uma estrutura que armazena as quantidades de banda disponíveis para cada classe de serviço diffserv determinadas pelo acordo SLA. Além disso, acrescentou-se as funções necessárias tanto para manipular a nova estrutura criada como também para tornar o novo agente apto a executar o mapeamento das capacidades entre as duas regiões. Entre as principais funcionalidades envolvidas nesse mapeamento pode-se destacar:

- Selecionar um comportamento agregado por nó (PHB) apropriado;
- Realizar o policiamento do tráfego apropriado nas bordas das redes diffserv;
- Realizar o controle de admissão nas requisições intserv, de acordo com os recursos disponíveis nas redes diffserv.

Exatamente como essas funções serão realizadas depende da forma como os recursos são gerenciados dentro das redes diffserv. A RFC2998, na qual o agente implementado aqui se baseia, sugere três formas de se fazer este gerenciamento:

- Estaticamente;
- Dinamicamente, utilizando o RSVP;
- Dinamicamente utilizando outros meios, como um gerenciador de banda.

O agente implementado por nós oferecerá as duas primeiras formas de gerenciamento. Contudo, até o momento, apenas o gerenciamento estático foi desenvolvido. A descrição detalhada do desenvolvimento do agente pode ser encontrada em [ABE01]. Este trabalho

³ - Extensão orientada a objetos para a linguagem Tcl [WEL00] desenvolvida pelo MIT.

apresentará e analisará os resultados obtidos nesta primeira fase de desenvolvimento onde o gerenciamento dos recursos da rede diffserv é feito estaticamente.

3.1. Gerenciamento Estático dos Recursos das Redes Diffserv

Neste caso, nenhum dispositivo das redes diffserv está apto a processar RSVP. Os recursos das redes diffserv são estaticamente alocados. Para tanto, o(s) consumidor(es) e o administrador(es) da(s) rede(s) diffserv negociam um contrato estático (SLA), que determina quais os recursos a serem disponibilizados para o(s) consumidor(es) em cada nível de serviço. Esses recursos podem ser simplesmente uma determinada quantidade de banda, ou mesmo um conjunto mais complexo envolvendo outros fatores como taxa de pico, tamanho da rajada, hora do dia, etc.

Cada roteador de fronteira na rede intserv do consumidor deve ser composto de duas partes: uma parte intserv, que interage com a rede do consumidor, e uma parte diffserv, que, por sua vez, interage com a rede diffserv. A parte intserv está apta a identificar e processar o tráfego de acordo com forma padrão intserv, através de uma granularidade por fluxo.

Já a parte diffserv do roteador de fronteira pode ser considerada como consistindo de um determinado número de enlaces virtuais, um para cada classe de serviço negociada no contrato SLA. O roteador deve manter uma tabela que indica os recursos a serem disponibilizados para cada classe de serviço, de acordo com o contrato SLA firmado. Esta tabela em conjunção com o campo DS é usada para realizar o controle de admissão no fluxo que irá atravessar a rede diffserv.

Existem várias possibilidades de se agregar (mapear) fluxos intserv para classes de serviços diffserv. A adotada aqui é a sugerida em [FOR98] e [WRO00] como sendo a mais razoável:

Serviços Intserv	Serviços Diffserv
Melhor Esforço	Melhor esforço, “default” (BE)
Carga controlada	“Assured” (AF)
Garantido	“Expedited”, “premium” (EF)

Tabela 1 – Tabela de mapeamento das classes de serviço intserv para diffserv

Desta forma, todo o tráfego que usa o serviço de carga controlada é mapeado para uma única classe de serviço “assured” (neste artigo, especificamente, mapeia-se para classe AF11) e o serviço garantido é posto na classe de serviço “premium”.

Além disso, para que a parte diffserv do roteador de fronteira possa determinar corretamente o nível de serviço dentro da região diffserv, é necessário que o campo DS esteja (ou seja) marcado corretamente. Segundo Bernet et al. [BER00], existem duas possibilidades para realizar essa marcação: na primeira a marcação é feita no próprio emissor, na segunda a marcação é feita pelos roteadores. Quando a marcação é feita pelo computador emissor, o sistema operacional deste marca adequadamente o campo DS em cada pacote transmitido. Quando a marcação é feita pelo roteador, estes devem ser configurados para identificar o tráfego e marcar o campo DS apropriadamente. Isto pode ser feito dinamicamente ou estaticamente [BER00]. Neste trabalho, seguindo a sugestão da RFC 2998 para o caso do gerenciamento estático dos recursos da rede diffserv, a marcação será feita pela emissor. O campo DS será marcado de acordo com a tabela 1, em função do tipo de serviço solicitado na requisição de recursos admitida.

3.2. Seqüência de Eventos na Obtenção de QoS fim a fim

Para ilustrar melhor o processo pelo qual uma aplicação tenta obter QoS fim a fim de acordo com o modelo adotado neste trabalho, descreve-se a seguir a seqüência de eventos gerados durante o processo de sinalização (ver Figura 4):

1. O emissor Tx gera uma mensagem RSVP PATH descrevendo o tráfego que será oferecido pela aplicação.
2. A mensagem PATH é transportada até o receptor Rx. Na rede a qual o emissor pertence, a mensagem PATH sofre o processamento padrão intserv/RSVP.
3. No roteador de fronteira RF1, a mensagem PATH também sofre o processamento padrão intserv/RSVP, com os respectivos estados PATH sendo criados no roteador. Em seguida, a mensagem PATH é enviada para a rede diffserv.
4. Na rede diffserv a mensagem PATH é ignorada pelos roteadores, sendo apenas passada em frente até o roteador RF2 onde a mensagem volta a sofrer o processamento padrão intserv/RSVP.
5. Quando a mensagem PATH alcança um potencial receptor Rx, este gera uma mensagem RESV, indicando o interesse em receber o tráfego oferecido.
6. A mensagem RESV é transportada de volta para o emissor, passando pela rede diffserv. De acordo com o processamento padrão intserv/RSVP, ela pode ser rejeitada por qualquer nó apto a processar RSVP ao longo do caminho, caso os recursos demandados não estejam disponíveis.
7. No roteador RF2, a mensagem RESV é sujeita ao processamento padrão intserv/RSVP, de modo que ela pode ser rejeitada caso os recursos disponíveis na interface do caminho de volta de RF2 sejam insuficientes para atender as requisições do receptor. Caso não seja rejeitada, a mensagem será transportada através da rede diffserv até o roteador RF1, sem sofrer o processamento padrão intserv/RSVP.
8. No roteador RF1, a mensagem RESV dispara o processo de controle de admissão para a rede diffserv. RF1 então compara os recursos requisitados pela mensagem RESV com os recursos disponíveis na região diffserv, para o nível de serviço diffserv correspondente. O nível de serviço é definido através do mapeamento descrito na Tabela 1 (seção 3.1). A disponibilidade de recursos para aquele nível de serviço diffserv é determinada pela capacidade estabelecida no contrato SLA. Desta forma, RF1 toma a decisão compatível, podendo rejeitar a reserva de recursos ou aceitá-la de acordo as regras definidas no contrato. De qualquer modo, os recursos agregados são determinados pelos contratos SLAs.
9. Se RF1 aprovar a requisição, a mensagem RESV é admitida e continua seu caminho até o emissor. Além disso, RF1 atualiza suas tabelas internas registrando a redução na capacidade de recursos disponíveis no nível de serviço admitido. Caso ela seja rejeitada, a mensagem RESV não é encaminhada adiante e uma mensagem de erro é enviada para o receptor.
10. Dentro da rede do emissor a mensagem RESV volta a sofrer o processamento padrão intserv/RSVP, podendo ser rejeitada em qualquer nó, por indisponibilidade de recursos. Caso, a mensagem não seja rejeitada e chegue ao emissor Tx, este irá interpretar a mensagem recebida como indicação de que aquele tráfego específico foi admitido para o tipo de serviço intserv especificado.

Desta maneira obtém-se QoS fim a fim quando usa-se o gerenciamento estático no modelo combinando a arquitetura intserv/RSVP e a arquitetura diffserv adotado neste trabalho.

4. Análise dos Resultados

De acordo com [JAI91], para validar um modelo de simulação existem três alternativas possíveis:

- Através da intuição de especialistas
- Através de medidas em um sistema real
- Através de resultados teóricos

Neste trabalho optou-se pela intuição de especialistas, uma vez que não existe nenhum “test-bed” razoavelmente completo que possa ser usado para avaliar o comportamento do modelo combinado intserv e difserv, da mesma forma que o desenvolvimento de um modelo teórico correto e completo levaria muito mais tempo que o desenvolvimento do agente e desses exemplos. Desta forma, diversos casos foram simulados para testar o comportamento do agente em diferentes situações consideradas críticas. Neste artigo serão apresentados exemplos que ilustram o agente desenvolvido desempenhando suas principais funções e tendo papel decisivo na aceitação ou rejeição de solicitações de reservas de acordo com os recursos disponíveis nas redes de serviços diferenciados.

A topologia escolhida para realizar as simulações e testar as diversas funcionalidades do agente é apresentada na Figura 5:

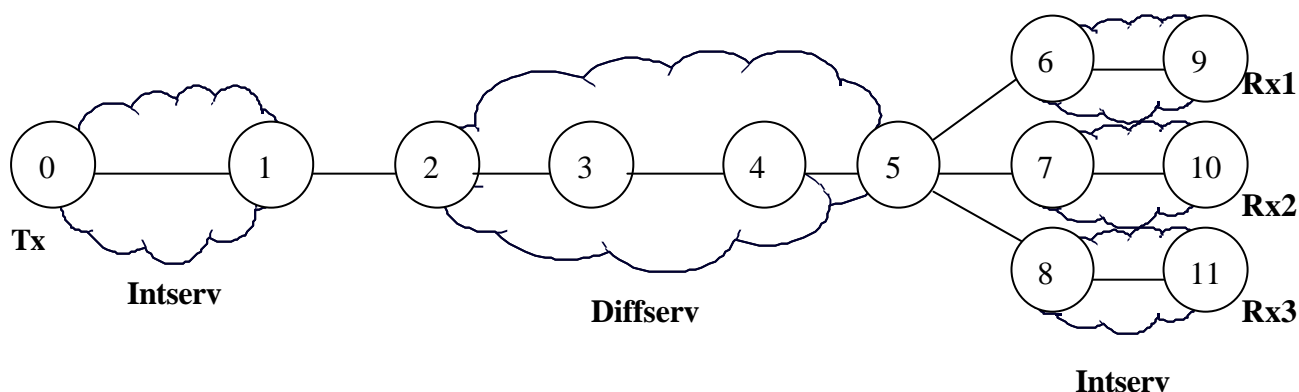


Figura 5 –Topologia da rede simulada

A rede, composta por 12 nós, foi gerada procurando representar a rede de referência do modelo proposto por Bernet et al. Os nós de 0 e 1 fazem parte da rede intserv 1, enquanto os nós de 2 a 5 fazem parte da rede diffserv 1. Os nós de 6 a 11 pertencem também a redes intserv, agrupados dois a dois, sendo que os nós 6 e 9 pertencem a rede intserv 2, os nós 7 e 10 são da rede intserv 3 e, por fim, os nós 8 e 11 fazem parte da rede intserv 4. Todos os enlaces foram configurados com capacidade de 1Mbit/s, com retardo de 0,002 segundos cada um. Propositamente, tanto os enlaces das redes intserv como os da rede diffserv possuem capacidades relativamente baixas para que estes possam saturar com os fluxos gerados pelo nó 0 da rede intserv 1. Os nós de 0 e de 6 a 11 estão associados a agentes intserv aptos a processar mensagens RSVP. Os nós de 2 a 5 e os respectivos enlaces que interligam esses nós foram configurados para se comportarem como uma rede diffserv. Ao nó 1 foi associado o agente desenvolvido neste trabalho, denominado daqui por diante de agente isds (como referência a integração de serviços integrados e diferenciados), e com o comportamento descrito na seção 3.

O escalonador associado aos nós da rede diffserv possui três filas, uma para cada tipo de classe de serviço diffserv (EF, AF e BE). Essas filas são servidas usando-se um esquema de escalonamento WRR (“Weighted Round Robin”) simplificado [BRA98], [MUR00], sendo que a fila para o nível de serviço EF é uma simples “drop tail”, para o nível AF a fila usada é

a RIO (“Random Early Detection with In and Out”) [CLA97], enquanto para o classe BE utiliza-se a fila RED (“Random Early Detection”) [FLO93]. Na parte intserv, atualmente apenas o serviço de carga controlada é suportado e a garantia de banda é fornecida através de escalonadores WFQ (“Weighted Fair Queuing”) [DEM89]. Como o esquema de mapeamento adotado neste trabalho (vide seção 3.1) mapeia para classe de serviço EF apenas os pacotes das requisições de serviço da classe garantida intserv, o comportamento do agente isds será analisado apenas para as classes de serviços AF e BE.

Inicialmente, no exemplo 1, simulou-se uma rede saturada e sem nenhum tipo de reserva de recursos, baseada apenas no serviço de melhor esforço e sem a atuação de qualquer tipo de agente. O nó 0 (Tx) atuou como emissor, gerando três fluxos distintos com taxas constantes (CBR), denominados CBR1, CBR2 e CBR3, através de transmissões um para um, enviando para os nós 9 (Rx1), 10 (Rx2), e 11 (Rx3), respectivamente. Cada fluxo foi gerado com taxa de 500Kbits/s (Kbps) com pacotes de tamanho de 100 bytes (B). Em cada receptor foi instalado um monitor para medir a taxa de pacotes recebidos. Como se esperava, com o enlace principal saturado, os pacotes de todos os fluxos foram descartados aleatoriamente. Os resultados são mostrados na Figura 6.

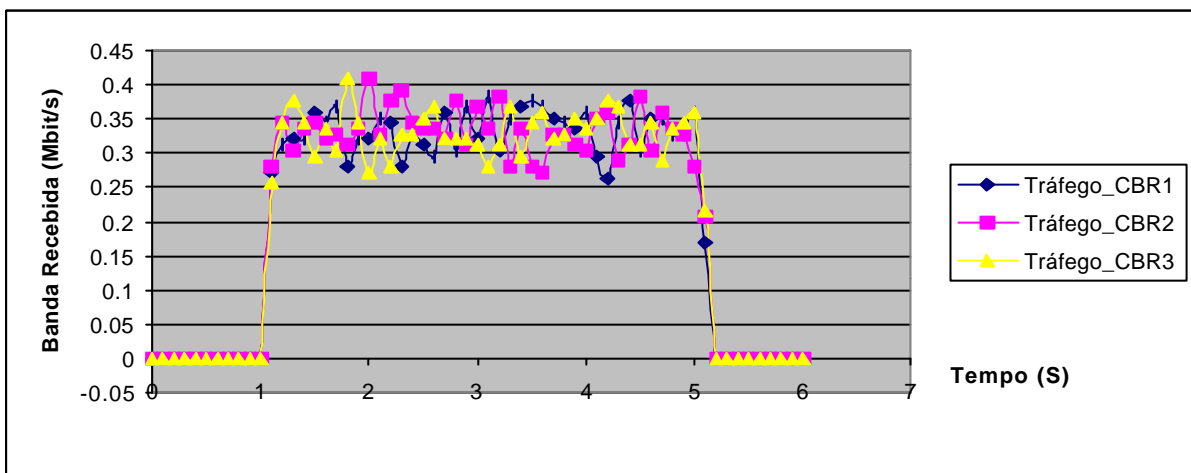


Figura 6 – Resultados do comportamento da rede saturada sem nenhum tipo de reserva.

Na segunda fase de simulações adicionou-se os agentes intserv/RSVP, diffserv e isds aos respectivos nós e estabeleceu-se um acordo SLA estático para controle apenas de banda onde a classe de serviço “Assured” (AF) tem direito a 550Kbps. Os pacotes quando são classificados para o nível de serviço “Assured” têm o campo DS marcado inicialmente como AF11. Aqueles pacotes que forem classificados como fora dos limites (“out-of-profile”) serão apenas remarcados para a classe AF12 e não serão descartados. Os pacotes de pedidos de reserva não atendidos são marcados como pertencentes a classe de melhor esforço (BE) e terão direito a usar toda capacidade ociosa do enlace. Os pacotes que excedem a capacidade disponível do enlace na classe de melhor esforço são sumariamente descartados.

No exemplo 2, com o intuito de mostrar o comportamento do agente na presença de indisponibilidade de recursos para um determinado nível de serviço diffserv, manteve-se a configuração saturada do exemplo 1 com as três fontes de tráfego CBR a partir do nó 0, todas com taxa de 500Kbps e com pacotes de tamanhos de 100B. Houve solicitação de reservas de recursos, pelos receptores, para os três fluxos, sendo as reservas feitas na seguinte ordem: primeiro Rx1, depois Rx2 e, por fim, Rx3. Todas as solicitações de reservas de recursos foram de 500Kbps e foram feitas antes de iniciar as transmissões dos respectivos fluxos. Os resultados da simulação, mostrando a banda recebida pelos monitores de tráfego instalados nos receptores Rx1, Rx2 e Rx3, são apresentados nas Figura 7a e 7b.

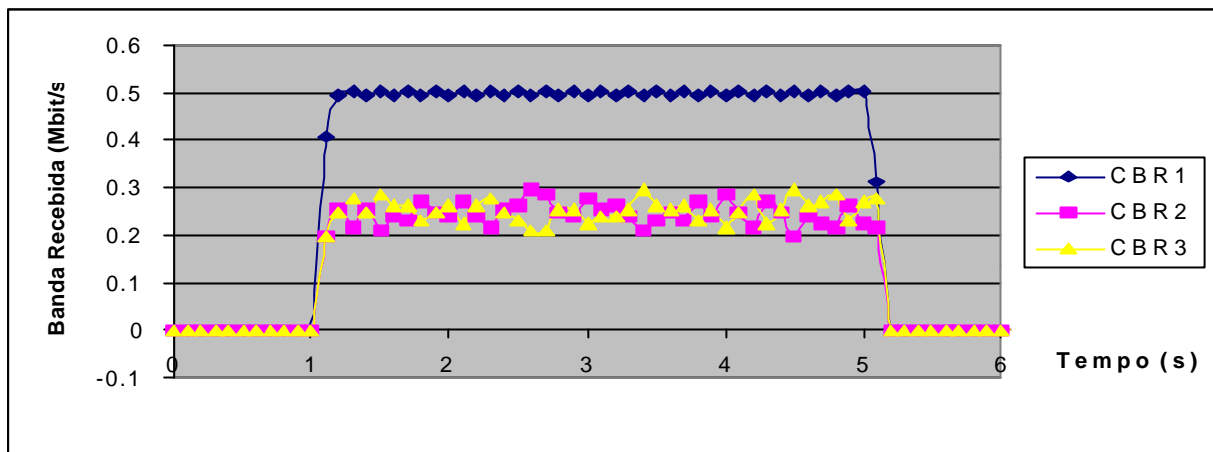


Figura 7a – Banda recebida pelos monitores de tráfego instalados nos receptores após apenas uma das reservas solicitadas, no caso para CBR1, ter sido aceita.

```

Iniciando simulacao...
1 PATH EVENT at 0.103 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
6 PATH EVENT at 0.117 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
9 PATH EVENT at 0.120 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
1 PATH EVENT at 0.303 : SID: 1 RATE: 500000 BUCKET: 100000 SENDER: 0
7 PATH EVENT at 0.317 : SID: 1 RATE: 500000 BUCKET: 100000 SENDER: 0
10 PATH EVENT at 0.320 : SID: 1 RATE: 500000 BUCKET: 100000 SENDER: 0
1 PATH EVENT at 0.503 : SID: 2 RATE: 500000 BUCKET: 100000 SENDER: 0
8 PATH EVENT at 0.517 : SID: 2 RATE: 500000 BUCKET: 100000 SENDER: 0
11 PATH EVENT at 0.520 : SID: 2 RATE: 500000 BUCKET: 100000 SENDER: 0
6 RESV EVENT at 0.603 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
1 RESV EVENT at 0.618 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
0 RESV EVENT at 0.621 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
7 RESV EVENT at 0.703 : SID: 1 RATE: 500000 BUCKET: 100000 SENDER: 0
7 RESVERROR EVENT at 0.731 : SID: 1 CODE: 1 VALUE: 0 NODE: 1
10 RESVERROR EVENT at 0.734 : SID: 1 CODE: 1 VALUE: 0 NODE: 1
8 RESV EVENT at 0.903 : SID: 2 RATE: 500000 BUCKET: 100000 SENDER: 0
8 RESVERROR EVENT at 0.931 : SID: 2 CODE: 1 VALUE: 0 NODE: 1
11 RESVERROR EVENT at 0.934 : SID: 2 CODE: 1 VALUE: 0 NODE: 1
running nam...
running xgraph...

```

Figura 7b – Pedidos de reservas feitos durante a simulação e sinalizados pelos agentes vinculados aos respectivos nós para o exemplo 2.

Como esperado, apenas a reserva para o primeiro fluxo (CBR1) foi aceita pelo agente isds (nó 1), já que a capacidade de recursos disponível para a classe AF era de 550Kbps, de acordo com o contrato SLA firmado. Desta forma, os pacotes de CBR1 passaram a ter seu campo DS marcado com AF11, recebendo tratamento preferencial na rede, enquanto os campos DS dos pacotes de CBR2 e CBR3 passaram a ser marcados com BE, sendo então tratados pelo serviço de melhor esforço e, conseqüentemente, tendo os pacotes excedentes à capacidade do enlace descartados. Os resultados acima mostram o correto comportamento do agente isds, não aceitando reserva de recursos quando a rede diffserv não possui recursos disponíveis, mesmo com as redes intserv estando com recursos disponíveis para aceitar mais de uma reserva.

No exemplo 3 uma nova simulação é feita procurando-se mostrar o comportamento do agente quando mais de um tráfego, no caso dois, compartilham os recursos reservados em uma mesma classe de serviço. Outro ponto que deseja-se destacar aqui é a falta de flexibilidade do agente quando o gerenciamento de recursos na rede diffserv é feito de forma estática. As taxas das três fontes de tráfego foram alteradas para: CBR1=200Kbps, CBR2=400Kbps e CBR3=300Kbps. Para melhor avaliação dos resultados adicionou-se uma nova fonte de tráfego, só que agora gerando um tráfego em rajada, simulando uma aplicação FTP, originada em Tx e enviada para o receptor Rx1. Adicionou-se também outro monitor em Rx1 para este novo fluxo. Novamente, antes do início das transmissões dos fluxos, foram feitas solicitação de reservas de recursos para os três tráfegos CBR, só que agora na seguinte ordem: primeiro Rx3 solicitou 300Kbps, depois Rx2 pediu 400Kbps e, por último, Rx1 requisitou 200Kbps. Os resultados são mostrados nas Figuras 8a e 8b:

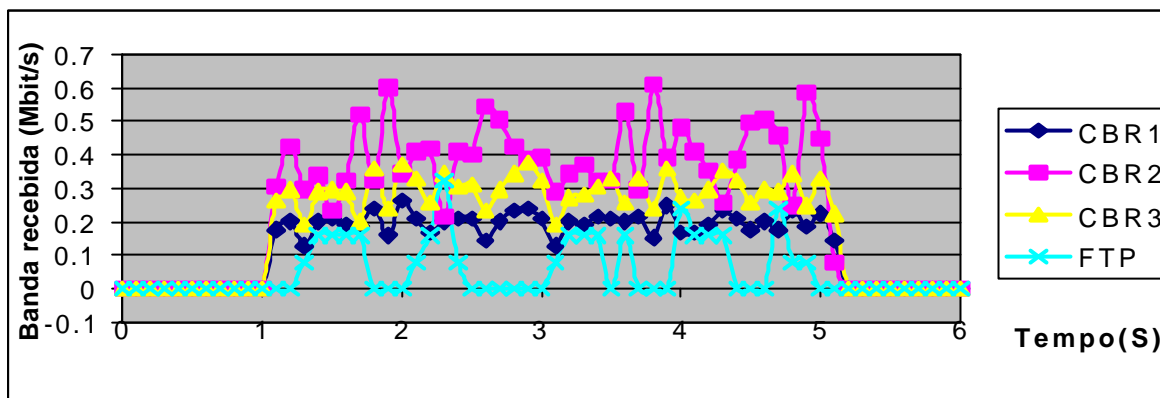


Figura 8a – Banda recebida pelos monitores de tráfego instalados nos receptores para o ex. 3.

```

Iniciando simulacao...
1 PATH EVENT at 0.103 : SID: 0 RATE: 200000 BUCKET: 100000 SENDER: 0
6 PATH EVENT at 0.117 : SID: 0 RATE: 200000 BUCKET: 100000 SENDER: 0
9 PATH EVENT at 0.120 : SID: 0 RATE: 200000 BUCKET: 100000 SENDER: 0
1 PATH EVENT at 0.303 : SID: 1 RATE: 400000 BUCKET: 100000 SENDER: 0
7 PATH EVENT at 0.317 : SID: 1 RATE: 400000 BUCKET: 100000 SENDER: 0
10 PATH EVENT at 0.320 : SID: 1 RATE: 400000 BUCKET: 100000 SENDER: 0
1 PATH EVENT at 0.503 : SID: 2 RATE: 300000 BUCKET: 100000 SENDER: 0
8 PATH EVENT at 0.517 : SID: 2 RATE: 300000 BUCKET: 100000 SENDER: 0
11 PATH EVENT at 0.520 : SID: 2 RATE: 300000 BUCKET: 100000 SENDER: 0
8 RESV EVENT at 0.603 : SID: 2 RATE: 300000 BUCKET: 100000 SENDER: 0
1 RESV EVENT at 0.618 : SID: 2 RATE: 300000 BUCKET: 100000 SENDER: 0
0 RESV EVENT at 0.621 : SID: 2 RATE: 300000 BUCKET: 100000 SENDER: 0
7 RESV EVENT at 0.703 : SID: 1 RATE: 400000 BUCKET: 100000 SENDER: 0
7 RESVERROR EVENT at 0.731 : SID: 1 CODE: 1 VALUE: 0 NODE: 1
10 RESVERROR EVENT at 0.734 : SID: 1 CODE: 1 VALUE: 0 NODE: 1
6 RESV EVENT at 0.903 : SID: 0 RATE: 200000 BUCKET: 100000 SENDER: 0
1 RESV EVENT at 0.918 : SID: 0 RATE: 200000 BUCKET: 100000 SENDER: 0
0 RESV EVENT at 0.921 : SID: 0 RATE: 200000 BUCKET: 100000 SENDER: 0
running nam...
running xgraph..

```

Figura 8b – Pedidos de reservas feitos durante a simulação e sinalizados pelos agentes vinculados aos respectivos nós para o exemplo 3.

Como pode ser observado, mais uma vez o agente isds comportou-se como esperado aceitando apenas as reservas de Rx1 e Rx3, que estavam dentro da capacidade de recursos diffserv disponíveis para a classe de serviço “Assured”. O pedido de reserva para o tráfego CBR2 feito por Rx2 foi recusado, mesmo com os enlaces intserv e diffserv tendo recursos disponíveis para aceitá-lo. Com isso o fluxo CBR2 teve que compartilhar com o fluxo FTP os recursos disponíveis para a classe de melhor esforço, enquanto os fluxos CBR1 e CBR3 receberam tratamento preferencial da rede. É importante ressaltar neste momento, que não se quer aqui sugerir esse esquema de gerência de recurso de forma estática como o melhor a ser utilizado, muito pelo contrário, pretende-se além de atestar o correto funcionamento do agente isds de acordo com o que foi proposto, mostrar o excesso de rigidez desse esquema de gerenciamento de recursos.

Uma última simulação, exemplo 4, para ilustrar o funcionamento do agente isds foi feita para o caso de uma união (“merge”) de reserva em uma comunicação multiponto. Para tanto manteve-se a topologia da rede e produziu-se novamente três fontes de tráfego CBR1, CBR2 e CBR3, todas com taxa de 500Kbps, transmitindo, via comunicação multiponto, para os três receptores Rx1, Rx2 e Rx3. Instalou-se três monitores de tráfego em cada receptor, um para cada fluxo. Nas solicitações de reserva, apenas Rx3 as fez e somente para o fluxo CBR1, na seguinte ordem: primeiro aos 0.6 segundos Rx3 solicitou uma reserva de 300Kbps, depois, no instante de tempo igual a 1,7 segundos, fez um pedido de 400Kbps e, por último, aos 2,8 segundos solicitou 500Kbps. Aos 4 segundos, Rx3 liberou as reservas. A Figura 9a apresenta a banda recebida pelos monitores dos três tráfegos instalados em Rx3 e a Figura 9b mostra os pedidos de reserva para o exemplo 4.

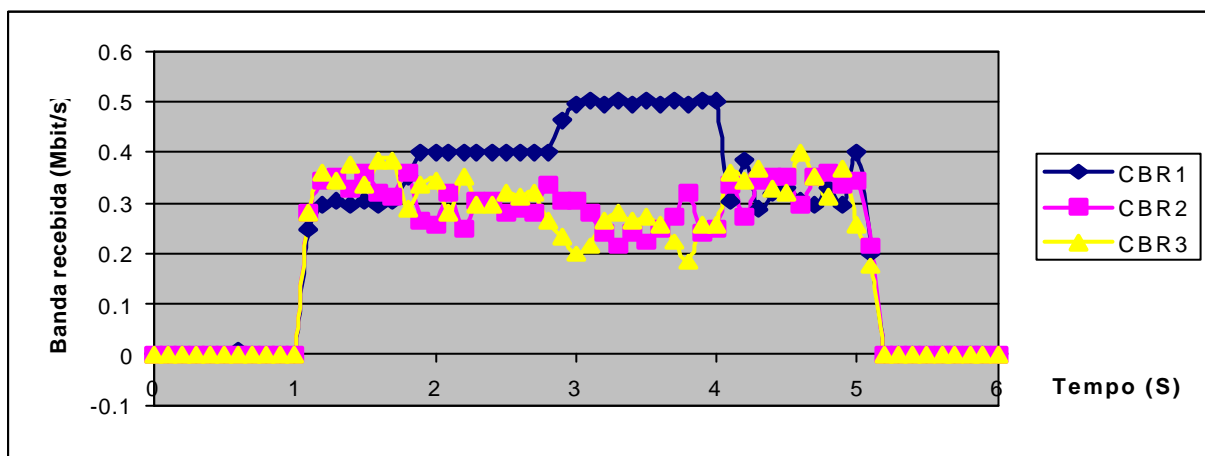


Figura 9a – Banda recebida pelos monitores de tráfego instalados em Rx3 para o ex. 4.

```

Iniciando simulacao...
1 PATH EVENT at 0.503 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
6 PATH EVENT at 0.517 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
7 PATH EVENT at 0.517 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
8 PATH EVENT at 0.517 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
9 PATH EVENT at 0.520 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
10 PATH EVENT at 0.520 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
11 PATH EVENT at 0.520 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
8 RESV EVENT at 0.603 : SID: 0 RATE: 300000 BUCKET: 100000 SENDER: 0
1 RESV EVENT at 0.618 : SID: 0 RATE: 300000 BUCKET: 100000 SENDER: 0
0 RESV EVENT at 0.621 : SID: 0 RATE: 300000 BUCKET: 100000 SENDER: 0
8 RESV EVENT at 1.703 : SID: 0 RATE: 400000 BUCKET: 100000 SENDER: 0
1 RESV EVENT at 1.718 : SID: 0 RATE: 400000 BUCKET: 100000 SENDER: 0
0 RESV EVENT at 1.721 : SID: 0 RATE: 400000 BUCKET: 100000 SENDER: 0
8 RESV EVENT at 2.803 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
1 RESV EVENT at 2.818 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
0 RESV EVENT at 2.821 : SID: 0 RATE: 500000 BUCKET: 100000 SENDER: 0
8 RESVTEAR EVENT at 4.003 : SID: 0 SENDER: 0
1 RESVTEAR EVENT at 4.016 : SID: 0 SENDER: 0
0 RESVTEAR EVENT at 4.018 : SID: 0 SENDER: 0
running nam...
running xgraph...

```

Figura 9b – Pedidos de reservas feitos durante a simulação e sinalizados pelos agentes vinculados aos respectivos nós para o exemplo 4.

Observa-se novamente o correto funcionamento do agente isds. Haja vista, que o “merge” dos pedidos (500Kbps) não ultrapassou a capacidade de recursos disponível para a classe em questão de acordo com o que foi firmado no contrato SLA (550Kbps). Desta forma, os três pedidos de reserva de Rx3 foram aceitos e as reservas do recursos foram corretamente efetuadas.

5. Conclusões Gerais e Trabalhos Futuros

Este trabalho apresentou um agente para integrar a arquitetura de serviços integrados com a arquitetura de serviços diferenciados, realizando os mapeamentos necessários entre as capacidades das duas arquiteturas. O agente foi desenvolvido para o ambiente de simulação NS (Network Simulator) e, neste trabalho, apenas os resultados obtidos com o gerenciamento estático de recursos foram analisados.

Os resultados das simulações apresentados aqui serviram tanto para atestar o correto funcionamento do agente nas diferentes situações propostas, como também para corroborar a eficiência do uso de ferramentas de simulação, como o NS, para desenvolver e avaliar protocolos de redes em geral. Além disso, os resultados serviram também para comprovar que o gerenciamento estático dos recursos da região diffserv apresenta sérios problemas de flexibilidade. Isso porque, em muitos casos, como no exemplo 3 (ver Figura 8), constata-se que é difícil fazer uso eficiente dos recursos na rede diffserv, já que o roteador de fronteira responsável pelo controle de admissão desta rede teria que ser reconfigurado a cada mudança dinâmica no contrato. Outro ponto que desabona o uso de gerenciamento estático dos recursos é a incapacidade dos roteadores de fronteira em garantir que os recursos da rede diffserv não serão duplamente alocados para diferentes consumidores quando houver dois ou mais desses roteadores. Na verdade, nestes casos a única alternativa é deixar esta responsabilidade a cargo do responsável pela rede diffserv, já que este assume o compromisso de que haverá determinada disponibilidade de recursos através de contratos.

Uma alternativa para resolver esse problema é utilizar um esquema de gerenciamento dinâmico, utilizando RSVP ou outro meio como um gerenciador de banda (“bandwidth broker”). Possíveis trabalhos futuros envolvem a simulação e análise do comportamento do agente desenvolvido aqui, utilizando esses esquemas de gerenciamento de recursos de forma dinâmica. Fora isso, outros trabalhos relacionados com o tema e com o ambiente de simulação NS incluem o desenvolvimento de outros tipos de perfis (“profiles”) para o módulo diffserv desenvolvido por Sean Murphy, o desenvolvimento da classe de serviço garantido para o programa intserv/RSVP desenvolvido por Marc Greis, bem como a simulação do modelo proposto por Bernet et al. combinado com outros mecanismos, como o MPLS, ou o roteamento com restrições.

6. Referências Bibliográficas

- [ABE01]-ABELÉM, A. “Implementação de um Agente para Oferecer QoS Fim a Fim Através da Combinação entre Serviços Integrados e Serviços Diferenciados”. *Trabalho de Proj. Final de Programação*, Departamento de Informática, PUC-Rio, Fevereiro, 2001.
- [APO99]-APOSTOLOPOULOS, G. et al. “QoS Routing Mechanisms and OSPF Extensions”. *RFC2676*. Agosto, 1999.
- [AWD98]-AWDUCHE, D. et al. “Requirements for Traffic Engineering over MPLS”. *Internet draft*, draft-ietf-mpls-traffic-eng.00.txt. Outubro, 1998.
- [AWD98B]-AWDUCHE, D. et al. “Extension to RSVP for Traffic Engineering”. *Internet draft*, draft-swallow-mpls-RSVP-trafeng-00.txt. Agosto, 1998.
- [BAJ99]-BAJAJ, S. et al. “Improving Simulation for Network Research”. *Technical Report*, Computer Science Department, USC. Setembro, 1999.
- [BER00]-BERNET, Y. et al. “A Framework for Integrated Services Operation over Diffserv Networks”. *RFC2998*. Novembro, 2000.
- [BLA98]-BLAKE, S. et al. “An Architecture for Differentiated Services”. *RFC2475*. Dezembro, 1998.
- [BRA94]-BRADEN, R. et al. “Integrated Service in the Internet Architecture: an Overview”. *RFC1633*. Junho, 1994.
- [BRA97]-BRADEN, R. et al. “Resource Reservation Protocol (RSVP) Version 1 Functional Specification”. *RFC2205*. Setembro, 1997.
- [BRA98]-BRADEN, B. et al. “Recommendation on Queue Management and Congestion Avoidance in the Internet”. *RFC2309*. Abril, 1998.
- [CLA88]-CLARK, D. “The Design Philosophy of the DARPA Internet Protocol”. In: *Anais ACM SIGCOMM’88*, Agosto, 1988.
- [CLA97]-CLARK, D., et al. “An Approach to Service Allocation in the Internet”. *Internet draft*, draft-clark-different-svc-alloc-00.txt. Julho, 1997.
- [CRA98]-CRAWLEY, E. et al. “A Framework for QoS-based Routing in the Internet”. *RFC2386*. Agosto, 1998.
- [FLO93]-FLOYD, S. e JACOBSON, V. “Random Early Detection Gateways for Congestion Avoidance”. *IEEE/ACM Transactions on Networking*. Agosto, 1993.
- [FAL00]-FALL, K e VARADHAN, K. (editores) “*NS Notes and Documentation*”. Fevereiro, 2000.
- [FOR98]-FORD, P. e BERNET, Y. “Integrated Services Over Differentiated Services”. *Internet draft*, draft-ford-issll-diff-svc-00.txt, Março 1998.
- [GRE99]-GREIS, M. “RSVP/NS: An Implementation of RSVP for the Network Simulator ns-2”. *Technical Report*, Computer Science Department IV, University of Bonn, 1999.

- [GUE97]-GUERIN, R. et al. "Aggregating RSVP based QoS Requests". *Internet draft*, draft-guerin-aggreg-RSVP-00.txt. Novembro, 1997.
- [HEI99]- HEINANEN, J. et al. "Assured Forwarding PHB Group". *RFC2597*. Junho, 1999.
- [HUS00]-HUSTON G. "*Internet Performance Survival Guide: QoS Strategies for Multiservice Networks*". John Wiley & Sons, EUA, 2000.
- [JAC99]-JACOBSON, V., et al. "A Two-bit Differentiated Services Architecture for the Internet", *RFC2638*. Junho, 1999.
- [JAC99b]-JACOBSON, V., et al. "An Expedited Forwarding PHB". *RFC2598*. Junho, 1999.
- [JAI99]-JAIN, R. "*The Art of Computer System Performance Analysis*". John Wiley & Sons, EUA, 1991.
- [LI98]-LI, T. e REKHTER, Y. "Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)". *RFC2430*. Outubro, 1998.
- [MUR00]-MURPHY, S. "Diffserv additions to ns-2". *Url: <http://www.teltec.dcu.ie/%7Emurphys/ns-work/diffserv/index.html>*, Julho, 2000.
- [NIC98]- NICHOLS, K. et al. "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers". *RFC2474*. Dezembro, 1998.
- [RAJ99]-RAJAN, R. et al. "A Policy Framework for Integrated and Differentiated Services in the Internet". *IEEE Network*, Setembro, 1999.
- [ROS01]-ROSEN, E. et al. "Multiprotocol Label Switching Architecture". *RFC3031*. Janeiro, 2001.
- [SCH97]-SCHENKER, S. et al. "Specification of Guaranteed Quality of Service". *RFC2212*. Setembro, 1997.
- [SEA00]-SEAMAN, M., et al. "Integrated Service Mappings on IEEE 802 Network". *RFC2815*. Maio, 2000.
- [VAN98]-VAANANNEN, P. e RAVIKANTH, R. "Framework for Traffic Management in MPLS Networks". *Internet Draft*, draft-vaanannen-mpls-tm-framework-00.txt. Março, 1998.
- [VIN97]-VINT Project- Virtual InterNetwork Testbed. *Url: <http://www.isi.edu/nsnam/vint>*, outubro, 1997.
- [XIA99]-XIAO, X. et al. "Internet QoS: A Big Picture". *IEEE Network*. Março, 1999.
- [XIA00]-XIAO, X. et al. "Traffic Engineering with MPLS in the Internet". *IEEE Network*. Março, 2000.
- [WEL00]-WELCH, B. "*Practical Programming in Tcl and Tk*". EUA, Prentice-Hall, 3ª edição, 2000.
- [WET95]- WETHERALL, D. e LINDBLAD, C. "Extending Tcl for Dynamic Object-Oriented Programming". In: *Proceeding of Tcl/Tk Workshop 95*. Toronto, Canadá, julho, 1995.
- [WRO97]-WROCLAWSKI, J. "Specification of the Controlled-Load Network Element Service". *RFC2211*. Setembro, 1997.
- [WRO97b]-WROCLAWSKI, J. "The use of RSVP with IETF Integrated Services". *RFC2210*. Setembro, 1997.
- [WRO00]-WROCLAWSKI, J. e CHARNY, A. "Integrated Service Mappings for Differentiated Services Networks". *Internet draft: draft-ietf-issll-ds-map-00.txt*, Setembro, 2000.