

---

# Desafios Tecnológicos na Internet de Próxima Geração

Technological Challenges in Next Generation Internet

Alencar Melo Júnior<sup>1</sup>, PUCCAMP, UNISAL, Metrocamp

## Resumo

A convergência de redes específicas, como as de telefonia e rádio, para Internet, com a consequente proliferação de aplicações multimídia, ocasionaram um grande aumento de tráfego, e, conseqüentemente, exibindo assim as deficiências do modelo comum de desenvolvimento. Este trabalho tem como objetivo apresentar os novos serviços de rede que estão sendo propostos para a Internet de Próxima Geração, bem como discutir os diversos desafios tecnológicos que são enfrentados para viabilização destes avanços, com destaque para a implementação de escalonadores de pacotes, componentes vitais para qualquer arquitetura Internet orientada à qualidade de serviços.

**Palavras-chave.** convergência; qualidade de serviço (QoS); internet de próxima geração; escalonador de pacotes; controlador fuzzy.

## Abstract

The convergence of specific networks such as telephony and radio, Internet, and the consequent proliferation of multimedia applications, caused a large increase in traffic, and consequently, thus exhibiting the shortcomings of the common development. This paper aims to present the new network services that are being proposed for the Next Generation Internet, and discuss the various technological challenges that are faced in enabling these advances, with emphasis on the implementation of packet schedulers, vital components for any Internet-oriented architecture to the quality of services.

**Keywords.** convergency; quality of service (QoS); next generation internet; packet scheduler; fuzzy controller.

## 1.Introdução

Tem se observado nos últimos anos uma tendência das redes isoladas e especializadas, tais como as redes de telefonia, televisão e rádio, convergirem para a Internet, ocasionando um grande aumento de tráfego. Com o surgimento das aplicações multimídia, os computadores passaram a processar áudio e vídeo (mídias contínuas); implementações de aplicações multimídia quando desenvolvidas sobre a Internet melhor-esforço atual não funcionam de modo adequado, devido principalmente aos grandes atrasos.

Os requisitos dos serviços de comunicação para aplicações multimídia excedem os serviços de comunicação atuais, orientados para dados, principalmente quanto ao suporte à Qualidade de Serviço (QoS). O termo QoS é empregado neste trabalho para caracterizar a habilidade de qualquer elemento da rede em fornecer aos seus usuários serviços com desempenho superior ao serviço melhor-esforço disponível atualmente.

O objetivo principal deste trabalho é apresentar os novos serviços de rede que estão sendo propostos para a Internet de Próxima Geração (NGI), bem como discutir diversos desafios tecnológicos que estão sendo enfrentados para a viabilização dos mesmos, com destaque para a implementação de escalonadores de pacotes. O escalonador de pacotes é um componente vital para qualquer arquitetura Internet orientada à QoS, pois é fundamental para assegurar requisitos de QoS aos diversos fluxos de pacotes servidos pelos roteadores.

Na Seção 2 discute-se as características da Internet atual. Conceitos relacionados à Qualidade de Serviço (QoS) são abordados na Seção 3; a natureza das aplicações multimídia é apresentada na

---

<sup>1</sup>Doutor. Docente da Pontifícia Universidade Católica de Campinas-PUCAMP (Campinas-SP), Centro Universitário Salesiano de São Paulo-UNISAL (Campinas-SP) e Faculdades Integradas Metropolitanas de Campinas-Metrocamp (Campinas-SP).

Seção 4. As principais arquiteturas Internet orientadas à QoS, as Arquiteturas de Serviços Integrados e de Serviços Diferenciados, bem com os diversos modelos de serviços são descritos na Seção 5. A Seção 6 discute aspectos fundamentais relacionados ao escalonamento de pacotes e a Seção 7 apresenta a principal contribuição deste trabalho, uma metodologia de desenvolvimento de escalonadores de pacotes utilizando controle *fuzzy* e aprendizado de máquina. Finalmente, as conclusões são apresentadas.

## 2.A Internet atual

A arquitetura e os principais protocolos da Internet tomaram sua forma corrente há mais de vinte anos. Os seus aplicativos iniciais, tais como terminal remoto, transferência de arquivos e correio eletrônico manipulavam essencialmente dados discretos, não requerendo da rede a garantia de rígidas restrições temporais. Os protocolos asseguravam apenas a fidelidade nas transmissões de mensagens, basicamente fazendo uso de mecanismos de retransmissão. As principais características dos protocolos atuais da família TCP/IP incluem (CORMER, 2000):

- independência da tecnologia de subrede (níveis 1 e 2 do modelo OSI/ISO), o que propicia máxima interoperabilidade;
- *ack's* (mensagens de reconhecimento positivo) fim-a-fim, dispensando *ack's* nos sistemas intermediários;
- fornecimento de uma única classe de serviços melhor-esforço;
- inexistência de mecanismos de controle de admissão - as aplicações não necessitam obter permissão prévia para iniciar a transmissão de seus pacotes;
- ausência de garantias de atrasos na entrega dos pacotes.

O enorme sucesso do protocolo IP é devido principalmente à sua simplicidade, o que o torna virtualmente capaz de operar sobre qualquer subrede de comunicação, desde tecnologias não orientadas à conexão, como Ethernet, até tecnologias orientadas à conexão, como o ATM. A filosofia sem-conexão caracteriza o protocolo IP, pois não é estabelecida uma conexão prévia entre emissor e receptor; os roteadores IP simplesmente fazem uma busca em sua tabela de retransmissão, fornecendo como entrada o endereço IP destino e obtendo o próximo *hop* para o qual o datagrama IP deverá ser transmitido. Se a fila para o próximo *hop* for longa, o datagrama sofrerá grandes atrasos. Caso a fila esteja cheia, o datagrama será descartado pelo roteador. As principais funcionalidades do protocolo IP são concentradas nos sistemas finais, ou seja, nos *hosts* origem e destino, deixando os sistemas intermediários com poucas atribuições.

A filosofia de projeto do IP faz com que o estado mantido no núcleo da rede para cada fluxo seja minimizado, o que facilita a recuperação em casos de falhas de enlaces ou roteadores, além de simplificar o mapeamento do protocolo IP para as mais diversas infra-estruturas de comunicação. Contudo, como consequência principalmente das opções iniciais de projeto, o protocolo IP atual fornece uma única classe de serviço para entrega dos datagramas IP, chamada melhor-esforço. Uma aplicação que faz uso do serviço melhor-esforço está sujeita a atrasos imprevisíveis. De modo geral, pode-se dizer que serviços do tipo melhor-esforço não oferecem para os usuários nenhum tipo de garantia para os parâmetros típicos de QoS. A seguir, são discutidos diversos aspectos relacionados à Qualidade de Serviço.

## 3.Qualidade de Serviço

Devido ao enorme crescimento que a Internet vem experimentando nos últimos anos e ao surgimento de diversas novas aplicações, o tráfego da Internet aumentou muito em volume, mas também sofreu grandes alterações em sua natureza. As principais diferenças entre dados discretos e mídia contínua (áudio o vídeo), produzida pelas novas aplicações, são exibidas na Tabela 1.

**Tabela 1. Dados Discretos versus Média Contínua.**

	<b>Dados Discretos</b>	<b>Média Contínua</b>
Largura de Banda	baixa	alta
Sensibilidade aos Atrasos	pouco sensível (aplicações assíncronas)	muito sensível (aplicações isócronas)
Sensibilidade ao Descarte	nenhuma perda (detecção e recuperação)	admite alguma perda (descarte de pacotes com erro)
Padrão de Tráfego	rajadas (sem limite superior de utilização de banda)	fluxo contínuo (com limite superior de utilização de banda)
Modo de Comunicação	ponto-a-ponto	multiponto
Relacionamento Temporal	nenhum	transmissão sincronizada (fluxos de áudio e vídeo)

Os requisitos dos serviços de comunicação para aplicações multimídia excedem os serviços de comunicação correntes, orientados para dados, quanto ao suporte à QoS e ao suporte à comunicação de grupos. O termo QoS em comunicação é um pouco ambíguo, podendo apresentar múltiplas interpretações, dependendo dos problemas e do ambiente em que este está sendo considerado. De modo geral, podemos usar o termo QoS para caracterizar a habilidade de um elemento qualquer da rede (aplicações, *hosts* ou roteadores) de propiciar aos usuários um serviço que apresente desempenho superior ao serviço melhor-esforço atual, levando-se em consideração os parâmetros típicos de QoS, tais como largura de banda, atraso na entrega dos pacotes, *jitter* (variação dos atrasos) e taxa de descarte de pacotes.

Nas aplicações multimídia em rede freqüentemente existe interação tempo-real entre os dois extremos da aplicação; logo, baixos atrasos são fundamentais. O atraso possui vários componentes, como o atraso de propagação no meio físico (constante, relacionado à camada física), o atraso de transmissão (variável em LANs, depende da tecnologia empregada na camada de enlace) e o atraso nas filas internas dos roteadores. Normalmente, o atraso nas filas dos roteadores constitui a principal causa do *jitter*, principalmente quando os roteadores implementam internamente uma única fila de pacotes com disciplina de acesso do tipo FIFO. Quando a rede apresenta *jitter* com valores elevados, o dimensionamento dos recursos, tipicamente largura de banda e tamanho dos *buffers* nos roteadores, fica bastante prejudicada, pois *jitter* alto implica em comportamento da rede pouco previsível. Com isso, a capacidade de adaptação das aplicações nos sistemas finais fica bastante comprometida.

Uma idéia comum a quase todas as definições de QoS é a capacidade da rede de diferenciar entre tipos de tráfego ou serviço, de tal modo que os usuários possam ter a disposição mais de uma classe de serviço (CoS) diferente, possibilitando a escolha dos serviços mais adequados para as suas aplicações (FERGUSON & HUSTON, 1998). O termo CoS implica na distinção de tipos de tráfego, os quais podem ser classificados e administrados diferentemente através da rede. Nem sempre é desejável que um tráfego receba tratamento prioritário sobre outros tipos de tráfego; muitas vezes prefere-se que algumas características comportamentais, tais como os atrasos nas filas dos roteadores e a banda disponível, permaneçam previsíveis. O termo QoS, contudo, tem uma conotação mais abrangente, englobando CoS, controle de admissão, policiamento e reformatação de tráfego, escalonamento de pacotes, controle de congestionamento, roteamento etc.

Os usuários enxergam QoS como a habilidade da rede em fornecer algum nível de garantia de que os requisitos de serviço de seu tráfego possam ser satisfeitos. Garantir ou assegurar QoS em um sistema distribuído é um problema fim-a-fim, envolvendo a cooperação de todos os elementos da rede fim-a-fim, bem como de todas as camadas de *software* e *hardware* relacionadas em cada elemento da rede ao longo do caminho. As garantias serão sempre limitadas pelo elo mais fraco da cadeia entre emissor e receptor. Basicamente, serviços de comunicação orientados à QoS podem oferecer garantias quantitativas ou qualitativas. A possibilidade de oferecer garantias quantitativas de QoS tem despertado muitas polêmicas; muitos advogam que é praticamente impossível garantir um desempenho consistente, previsível e sem perdas em redes de comutação de pacotes.

Para atender aos requisitos de comunicação de aplicações multimídia com a infra-estrutura melhor-esforço atual temos basicamente duas opções:

- manter o nível de utilização da rede extremamente baixo, o que implica em superdimensionar toda a rede; esta alternativa atualmente não é viável economicamente, apesar de alguns autores considerarem que no futuro a banda de rede possa vir a ser gratuita (NEGROPONTE, 1995);
- tornar as aplicações adaptativas à carga atual da rede, ou seja, alterar as aplicações ao invés de modificar a infra-estrutura atual. Técnicas adaptativas tem sido usadas com sucesso em várias aplicações para a Internet, mas para aplicações que necessitam de maiores garantias, ou mesmo para um melhor desempenho de aplicações menos exigentes, elas não são suficientes.

Portanto, fazem-se necessárias novas arquiteturas de rede, que possam diferenciar aplicações e atendê-las segundo suas necessidades. A natureza das aplicações multimídia, responsável por direcionar o desenvolvimento das novas arquiteturas de rede, é abordada na próxima seção.

## 4. Natureza das aplicações multimídia

A grande maioria das aplicações multimídia distribuídas são do tipo tempo-real Tabela 1 (CLARK, SHENKER & ZHANG, 1992). Nestas aplicações, uma fonte de tráfego gera algum sinal, empacota-o e transmite os pacotes através da rede. A rede inevitavelmente introduz alguma variação no atraso dos pacotes entregues, provocando *jitter*. O receptor desempacota os dados e tenta reproduzir o sinal de modo fiel, apresentando o sinal com um *offset* de atraso fixo a partir do tempo de partida original do sinal, como mostra a Equação 1.

### Equação 1. Ponto de *playback*

$$\text{ponto de playback} = \text{tempo de partida} + \text{offset de atraso}$$

O desempenho de uma aplicação *playback* pode ser medido ao longo de duas dimensões: *latência* e *fidelidade*; a fidelidade pode ser classificada como apresentação fiel, distorcida ou incompleta. Aplicações que envolvem interação entre os dois extremos da conexão, como telefonia, são mais sensíveis à latência; outras aplicações, como a transmissão de um filme, são menos sensíveis a latência, mas requerem maior largura de banda. O *jitter* é de grande importância para o desempenho de aplicações *playback*, pois o tamanho de *buffer* requerido para uma conexão é diretamente proporcional ao *jitter*.

De acordo com a sensibilidade à perda de fidelidade, as aplicações tempo-real podem ser classificadas em intolerantes ou tolerantes. As aplicações tempo-real intolerantes não admitem perda de fidelidade e devem usar um *offset* de atraso fixo maior do que o atraso máximo, para evitar pacotes atrasados. O desempenho é independente de quando os pacotes chegam, desde que estes cheguem dentro do limite de atraso. Estas aplicações requerem um serviço de entrega de pacotes com um limite superior de atraso confiável.

As aplicações tempo-real tolerantes admitem alguns pacotes atrasados, não precisando usar um *offset* de atraso maior do que o atraso máximo. Elas podem tentar reduzir a latência através de um ajuste dinâmico do *offset* de atraso, observando os atrasos reais sofridos pelos pacotes em um passado recente, usando técnicas adaptativas à carga corrente da rede. Estas aplicações requerem um serviço de entrega de pacotes que procure melhorar a distribuição de atrasos como um todo e não garantir o atraso de pacotes individuais. Segundo Clark, Shenker & Zhang (1992), a grande maioria das aplicações *playback* é do tipo tolerante adaptativo. Muitas aplicações que se popularizaram recentemente, tais como as que possibilitam a telefonia sobre redes IP, comprovam esta afirmação.

Aplicações que envolvem monitoramento de processos, bolsas de valores etc. também requerem um limite superior de atraso confiável, mas não são necessariamente aplicações multimídia.

As aplicações não tempo-real ou elásticas normalmente usam os dados imediatamente (não *bufferizam*) e sempre optam por esperar um pacote atrasado do que prosseguir sem ele. O desempenho depende

mais da média dos atrasos do que dos extremos da distribuição de atrasos. Aplicações elásticas podem ser divididas em várias categorias, tais como as rajadas interativas, que são mais exigentes quanto ao atraso dos pacotes (por exemplo: telnet, nfs, etc.); as transferências interativas, que apresentam uma exigência intermediária quanto aos atrasos (como ftp) e as transferências assíncronas, que apresentam uma baixa exigência quanto aos atrasos dos pacotes (como correio eletrônico).

Continuar a oferecer um serviço uniforme para todas as aplicações não é satisfatório. Fazem-se necessárias novas arquiteturas de rede, que possam diferenciar as aplicações e atendê-las segundo suas necessidades, com diferentes classes de serviço, de modo eficiente e buscando evitar congestionamentos. As duas principais propostas para o oferecimento de QoS aos usuários da Internet são apresentadas a seguir.

## 5. Arquiteturas Internet orientadas à QoS

A QoS deve ser fornecida por fluxo, onde um fluxo é uma abstração de uma seqüência de pacotes relacionados que resultam de uma atividade e requerem a mesma QoS, podendo ter várias fontes e vários receptores. De modo geral, para que um fluxo receba garantias de QoS fim-a-fim devem ser observadas as seguintes condições: não ocorrer alterações de caminho ou falhas de roteadores, o padrão de tráfego deve respeitar os descritores previamente acordados e os pacotes não devem sofrer fragmentação.

Questões fundamentais relacionadas às Arquiteturas Internet de Próxima Geração são discutidas por Shenker (1995). As principais conclusões são:

- deve ser incluído outros serviços de entrega de pacotes além do serviço melhor-esforço atual, com o objetivo de melhor atender aos diferentes requisitos das aplicações;
- a aplicação deve escolher o serviço apropriado, ao invés da rede escolher o serviço para o fluxo, de modo a manter a independência entre as aplicações e a rede;
- deve ser feito o controle de admissão de novos fluxos, o qual pode ser realizado de modo explícito, fazendo uso de um protocolo de sinalização como o RSVP (ZHANG *et. al.*, 1993), ou de modo implícito, através do condicionamento e policiamento de tráfego nas bordas de um domínio.

As Arquiteturas Internet de Próxima Geração orientadas à Qualidade de Serviços podem ser classificadas em dois tipos:

- *QoS quantitativa*: são as arquiteturas que fazem reserva explícita de recursos para cada um dos fluxos produzidos pelas aplicações, também chamados de micro-fluxos. O estado dos fluxos no interior da rede é mantido durante todo o seu ciclo de vida, fazendo uso de protocolos de sinalização; como exemplo temos a Arquitetura de Serviços Integrados (BRANDEN, CLARK & SHENKER, 1994);
- *QoS qualitativa*: são as arquiteturas que simplesmente dão tratamento preferencial aos pacotes de aplicações que possuem requisitos mais estritos e não fazem reservas para micro-fluxos individuais. Na Arquitetura de Serviços Diferenciados (BLAKE *et. el.* 1998), os pacotes dos micro-fluxos individuais são classificados na borda da rede em agregados de fluxos, os quais podem receber diferentes níveis de serviços.

Os pacotes de um micro-fluxo possuem endereços origem, destino e número de porta comuns, sendo que na Arquitetura de Serviços Diferenciados vários micro-fluxos podem ser agregados em uma única classe de serviço, compartilhando os recursos associados à classe. A Arquitetura de Serviços Integrados mantém os micro-fluxos isolados, com reservas de recursos específicas para cada um, possibilitando oferecer serviços que apresentam um limite superior de atraso confiável (Serviço Garantido), mas sua escalabilidade é limitada, principalmente em redes de maior abrangência geográfica. A Arquitetura de Serviços Diferenciados, por fazer agregação de fluxos, não pode oferecer serviços que ofereçam garantias quantitativas de desempenho. Outros problemas relacionados ao isolamento e compartilhamento de reservas de recursos por diferentes fluxos são discutidos por Clark, Shenker & Zhang (1992).

## 5.1. Arquitetura de Serviços Integrados

A Arquitetura de Serviços Integrados (IntServ) (BRANDEN, CLARK & SHENKER, 1994) se propõe a estender a funcionalidade atual do modelo arquitetural IP, fornecendo outros serviços além do melhor-esforço, de modo a atender os requisitos de QoS das aplicações de tempo-real. Os novos serviços propostos são o Serviço Garantido (SHENKER, PARTRIDGE & GUERIN, 1997), voltado para as aplicações tempo-real intolerantes, e o Serviço Carga-Controlada (WROCLAWSKI, 1997), destinado a atender aos requisitos das aplicações tempo-real tolerantes e adaptativas.

O Serviço Garantido fornece garantias de limite superior de atraso fim-a-fim, de largura de banda, e de que não ocorrerá perdas de pacotes devido ao estouro de *buffers* para os pacotes em conformidade com o descritor de tráfego *token-bucket* (CLARK, SHENKER & ZHANG, 1992) do seu fluxo. A aplicação informa os roteadores sobre as características de tráfego de seus fluxos e da reserva de recursos requerida através do protocolo de sinalização RSVP (ZHANG *et. al.*, 1993). Cada roteador ao longo do caminho aloca, para cada fluxo aceito, uma determinada largura de banda e espaço em *buffer*.

O Serviço Carga-Controlada fornece um compromisso ao fluxo de que ele irá receber um serviço equivalente ao de um fluxo melhor-esforço em uma rede com carga baixa, usando controle de admissão para garantir que a QoS não será degradada com o aumento da carga de utilização da rede.

Além dos novos serviços e do protocolo de sinalização, os outros componentes básicos da arquitetura IntServ são o classificador de pacotes, que mapeia cada pacote recebido em uma classe (todos os pacotes de uma mesma classe recebem a mesma QoS); o escalonador de pacotes, que determina a seqüência com que os pacotes pertencentes às diversas classes serão transmitidos, de forma que as exigências de QoS possam ser satisfeitas, e o controle de admissão, que implementa um algoritmo de decisão que os roteadores usam para determinar se um novo fluxo pode ter seus requisitos de QoS garantidos, sem violar as garantias dos fluxos previamente aceitos. O protocolo de sinalização RSVP possibilita que emissores, receptores e roteadores de uma sessão *unicast* ou *multicast* comuniquem entre si com o objetivo de estabelecer o estado necessário nos roteadores para suportar os Serviços Garantido e Carga-Controlada.

Na Arquitetura de Serviços Integrados, o tráfego de um fluxo é caracterizado principalmente por um filtro *token-bucket* ( $r, b$ ), onde  $b$  é a capacidade do balde, o qual é abastecido com *tokens* a uma taxa de  $r$  *tokens* por segundo. Para que um pacote possa ser transmitido, um certo número de *tokens* deve ser retirado do balde, de acordo com o tamanho do pacote. O filtro *token-bucket* possibilita rajadas controladas, limitadas pela quantidade de *tokens* disponíveis no balde. Nas bordas da rede, realiza-se o policiamento dos fluxos, comparando-se o tráfego de um fluxo com o seu descritor para verificar a conformidade; pacotes que não estão em conformidade podem ser transmitidos como pacotes melhor-esforço, por exemplo. Pode-se fazer necessário a reformatação de tráfego nos roteadores internos ao domínio IntServ de modo a suavizar as distorções introduzidas ao longo do caminho, readequando os fluxos aos seus filtros *token-bucket*.

A aplicabilidade da Arquitetura de Serviços Integrados e do RSVP em redes de longa distância como a Internet enfrenta limitações. Devido à necessidade do estado para cada um dos fluxos no interior da rede ser mantido, a solução torna-se pouco escalável. O desenvolvimento incremental da arquitetura também torna-se muito difícil, pois os roteadores precisam sofrer grandes alterações e as aplicações precisam gerar sinalização RSVP. Diante disso, a Arquitetura de Serviços Diferenciados tornou-se a proposta mais promissora para a Internet de Próxima Geração, principalmente por ser mais escalável e apresentar um menor custo computacional.

## 5.2. Arquitetura de Serviços Diferenciados

Na Arquitetura de Serviços Diferenciados (DiffServ) (BLAKE *et. al.*, 1998), os pacotes dos micro-fluxos individuais são classificados na borda da rede em agregados de fluxos, marcando ou remarcando os cabeçalhos dos pacotes de acordo com critérios previamente estabelecidos, definidos por meio de um Acordo de Nível de Serviços (SLA), estabelecido entre redes que possuem interconexão. Os pacotes podem também ser marcados previamente à sua entrada em um domínio DiffServ, por meio da aplicação ou do *host* emissor. Cada agregado de fluxos recebe nos roteadores internos ao domínio DiffServ uma das classes de serviço existentes no domínio, as quais possuem diferentes prioridades.

O tráfego em desacordo com o perfil contratado pode não obter a QoS desejada ou sofrer tarifação adicional. Os perfis de tráfego são derivados a partir do SLA.

A escalabilidade da arquitetura DiffServ é obtida graças à implementação de funções complexas como classificação e condicionamento de tráfego apenas nos nós da borda da domínio. No interior do domínio, os pacotes são simplesmente retransmitidos de acordo com a classe de serviço indicada em seus cabeçalhos. Devido às suas opções de projeto, a arquitetura DiffServ pode ser empregada em *backbones* WAN e as aplicações já existentes são beneficiadas, visto que não se faz necessário um protocolo de sinalização fim-a-fim como o RSVP.

No cabeçalho de cada pacote IP, o campo DS denota a classe de serviço (PHB) que o pacote deve receber em cada um dos *hops* do domínio DiffServ (NICHOLS, 1998). O campo DS é definido pelo octeto Tipo-de-Serviço (TOS) no protocolo IPv4 ou pelo octeto Classe de Tráfico (*Traffic Class*) no IPv6. O PHB (*Per-Hop Behavior*) define o serviço oferecido por um roteador para um agregado de fluxo BA (*Behavior Aggregate*), isto é, um agregado de fluxos cujos pacotes possuem as mesmas marcações no campo DS. Os PHBs constituem um modo granular para alocar recursos como largura da banda e *buffer* em cada um dos roteadores do domínio DiffServ, e são implementados por meio de algoritmos de escalonamento de pacotes e de gerenciamento de *buffers*.

Entre os primeiros serviços a serem propostos para a arquitetura DiffServ podem ser destacados o *Expedited Forward* (EF) (JACOBSON, NICHOLS & PODURI, 1999) e o *Assured Forward* (AF) (HEINANEN *et. al.*, 1999). O serviço EF garante aos usuários uma largura de banda nominal ao longo de um certo caminho na rede, similar a uma linha privada, buscando minimizar o atraso, o *jitter* e o descarte de pacotes. As garantias de QoS oferecidas pelo serviço EF vão na linha da diferenciação de serviços do tipo absoluta, requerendo para isso algum tipo de reserva de banda e fixação de rotas; logo, o serviço EF pode ser pensado como uma extensão para o Serviço Garantido da arquitetura IntServ.

O serviço AF define quatro classes de serviço, cada uma com três níveis de precedência para descarte de pacotes. A diferenciação de serviço proposta pelo serviço AF pode ser vista como uma diferenciação relativa, onde os usuários que subscrevem uma classe de serviço mais alta obtêm níveis de QoS superiores aos usuários que subscrevem classes mais baixas. A reserva de recursos para o serviço AF não é requerida, contudo alguma forma de provisionamento ou garantias estatísticas para banda e *buffer* fazem-se necessário (HEINANEN *et. al.*, 1999). O entendimento e a especificação do serviço AF tem se mostrado até certo ponto vaga, o que certamente dificultou a oferta do serviço; Kilkki (1999) afirma que o modo como se dá o relacionamento entre as quatro classes AF é um assunto ainda aberto e o uso das mesmas é até certo ponto confuso. As colocações de Kilkki podem ser comprovadas em diversos trabalhos; por exemplo, em Leung, Lui & Yau (2000), o serviço AF é classificado como sendo do tipo diferenciação absoluta e em Dovrolis & Ramanathan (1997), por sua vez, afirma-se que o serviço AF pode ser visto como um esquema de diferenciação de serviços relativa.

Um novo modelo de serviço para a arquitetura DiffServ, denominado Modelo de Diferenciação Proporcional (MDP), é proposto em Dovrolis & Ramanathan (1997). O MDP tem recebido bastante atenção devido principalmente à clareza de sua especificação e à sua viabilidade de implementação, por meio de mecanismos escaláveis. O MDP tem por objetivo fornecer um pequeno número de classes de serviço com garantias apenas para o ordenamento relativo do desempenho das classes, considerando parâmetros de QoS como atraso de fila e descarte de pacotes. O MDP não requer provisionamento de recursos e a fixação de rotas não é um aspecto importante. Uma análise mais aprofundada do MDP é realizada em Dovrolis, Stiliadis & Ramanathan (2002). Um novo escalonador de pacotes que possibilita atender aos objetivos do MDP com maior precisão é apresentado em Melo Jr., Magalhães & Adán Coello (2005). As principais características do MDP são:

- controlabilidade: do ponto de vista do operador da rede; o operador pode ajustar o espaçamento de QoS entre as classes de serviço, a partir de um conjunto de parâmetros;
- previsibilidade: do ponto de vista do usuário; deve-se procurar manter uma ordenação consistente entre as classes da QoS, conforme os parâmetros especificados pelo operador da rede.

O escalonador de pacotes é um componente vital em qualquer Arquitetura Internet de Próxima Geração, sendo fundamental para assegurar os requisitos de QoS dos fluxos. A próxima Seção discute diversos aspectos relacionados ao escalonamento de pacotes.

## 6. Escalonamento de pacotes

Para atender aos requisitos de QoS das novas aplicações para a Internet faz-se necessário empregar soluções mais sofisticadas do que os tradicionais escalonadores FIFO, principalmente quando as flutuações estatísticas das fontes de tráfego causam filas de tamanho considerável nos roteadores. O escalonamento de pacotes desempenha uma função fundamental em qualquer arquitetura Internet orientada à QoS, podendo ser classificado em duas famílias:

- escalonamento nas arquiteturas Internet orientadas à QoS quantitativa: o escalonador deve ser capaz de suportar aplicações do tipo tempo-real, oferecendo garantias determinísticas ou estatísticas de limite de desempenho para os micro-fluxos (atraso máximo etc.), requerendo reservas de recursos individuais para os mesmos;
- escalonamento nas arquiteturas Internet orientadas à QoS qualitativa: o escalonador normalmente deve suportar aplicações menos exigentes e adaptativas, procurando oferecer garantias relativas de QoS aos agregados de fluxos e também evitando a ocorrência de inanição de serviços em algum agregado.

Um escalonador de pacotes desempenha duas tarefas ortogonais: decidir a ordem na qual os pacotes serão servidos e gerenciar a fila de requisições pendentes. Logo, o escalonador pode influenciar no atraso (e conseqüentemente no *jitter*), na largura de banda fornecida e na taxa de descarte de pacotes.

Ao se projetar um escalonador de pacotes, deve-se considerar os seguintes graus de liberdade (escolhas fundamentais) (KESHAV, 1997):

- classes de serviço: o número de classes de serviço e o relacionamento entre elas deve ser determinado;
- classes de serviço conservativas versus não-conservativas: para cada uma das classes de serviço, deve-se decidir entre serviço conservativo, que nunca deixa o enlace ocioso quando existem pacotes para serem transmitidos e serviço não-conservativo, que pode deixar o enlace ocioso mesmo quando existem pacotes aguardando transmissão, visando reformatar o tráfego;
- grau de agregação de fluxos: o escalonador pode possuir uma fila por fluxo, uma fila para todas os fluxos (mesma QoS para todos) ou uma fila para cada classe diferente de QoS;
- ordem de serviços: é a ordem na qual o escalonador serve os pacotes de um fluxo ou agregado de fluxos. O escalonador pode empregar uma estratégia sem reordenação (algoritmo FIFO) ou com reordenação, o que pode acarretar um grande custo computacional para computar os rótulos dos pacotes.

Os diversos fluxos de pacotes pertencentes a uma mesma classe de serviço não são protegidos uns dos outros; logo, o comportamento inadequado de um fluxo, por exemplo gerando tráfego acima dos SLA contratado, pode comprometer o desempenho dos demais fluxos pertencentes a mesma classe.

De modo ideal, um escalonador deve ser escalável e possuir uma baixa complexidade, de modo a suportar um grande número de fluxos em enlaces de alto desempenho; também deve prover proteção, de forma que uma largura de banda mínima esteja garantida para cada fluxo, independentemente do comportamento dos outros fluxos. Deve ainda fornecer justiça, por meio de uma alocação de recursos justa para cada fluxo, de acordo com suas necessidades (KESHAV, 1997). A seguir, apresenta-se o arcabouço de uma proposta para o desenvolvimento de escalonadores de pacotes empregando técnicas baseadas em aprendizado de máquina.

## 7. Escalonamento de pacotes utilizando aprendizado de máquina

Nos últimos tempos pode-se observar o aparecimento de algumas propostas para o problema de escalonamento de pacotes em arquiteturas DiffServ empregando inteligência computacional; a seguir, algumas dessas propostas são discutidas.

Em Fernandez, Pedrosa & Rezende (2001) é proposto um controlador *fuzzy* para ajustar dinamicamente os pesos das filas de um escalonador *Weighted Round-Robin* tradicional, procurando atender aos requisitos de QoS das classes de serviço Melhor-Esforço e *Expedited Forward*; entretanto, o trabalho não discute como as regras do controlador (um total de 45) foram obtidas. Uma melhoria para o controlador *fuzzy* proposto em Fernandez, Pedrosa & Rezende (2001) é apresentada pelos mesmos autores em 2002, onde são empregados algoritmos genéticos para otimizar as funções de pertinência e o método de Wang-Mendel (WANG & MENDEL, 1992) para eliminar as inconsistências da base de regras. Nesse trabalho os autores afirmam que utilizar algoritmos genéticos e o método Wang-Mendel propicia a obtenção de um controlador ótimo para realizar o escalonamento de pacotes, sem depender de critérios de escolha dos projetistas. É certo que o desempenho do controlador foi melhorado com a utilização das duas técnicas, contudo a base de regras inicial do controlador foi fornecida pelos projetistas; logo, o controlador depende da escolha das regras iniciais, cuja metodologia não é abordada no trabalho.

Em Zhang & Ma (2000), o escalonador de pacotes é implementado por meio de um controlador *fuzzy*, cuja base de regras é obtida de modo intuitivo. Para um escalonador que leve em consideração um número maior de entradas, a tarefa de encontrar a base de regras é bastante complexa, requerendo uma abordagem mais elaborada.

Um dos principais objetivos deste trabalho é delinear uma proposta para a obtenção de escalonadores de pacotes utilizando aprendizado de máquina e controle *fuzzy* no contexto das arquiteturas Internet orientadas à QoS qualitativa, tais como DiffServ. O aprendizado de máquina está relacionado com a construção de programas de computador que melhoram automaticamente seu desempenho em alguma tarefa através da experiência, utilizando diversas técnicas, tais como redes neurais, árvores de decisão etc. (MITCHELL, 1997). O controle *fuzzy* é baseado na idéia de conjunto *fuzzy*, que é uma generalização dos conjuntos ordinários, através de um grau de pertinência para cada elemento (MUNAKATA & JANI, 1994; GOMIDE & PEDRYCZ, 1998). O grau de pertinência de um elemento é um número real situado no intervalo  $[0, 1]$ .

O emprego de controle *fuzzy* (GOMIDE & PEDRYCZ, 1998) na construção de escalonadores de pacotes mostra-se bastante apropriado, devido principalmente aos seguintes motivos:

- o controle *fuzzy* possibilita tomar decisões em situações na quais as informações disponíveis estão incompletas ou imprecisas - quando se usa protocolos sem conexão, como o IP, esta característica pode ser bastante útil;
- diversas variáveis do problema possuem fronteiras não claramente delineadas, tais como grau de utilização de recursos (banda/*buffer*) no roteador, que pode ser caracterizada como muito alta, alta, média e baixa, por exemplo;
- as regras de escalonamento podem ser bem descritas por regras heurísticas, tais como:
 

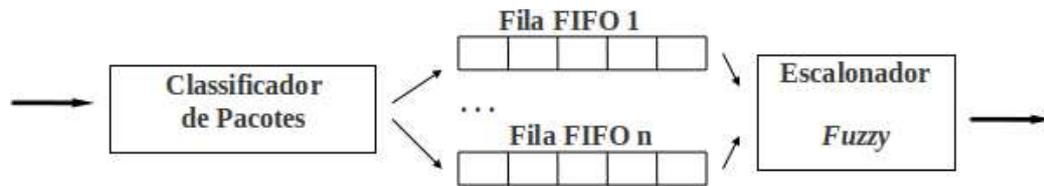
Se (utilização de *buffer* da classe de serviço é alta)  
     e (justiça na alocação de banda é baixa)  
 então  
     (priority de transmissão da classe de serviço é alta)
- o modelamento matemático para o comportamento do sistema é muito complicado de ser obtido;
- as regras podem considerar diversos parâmetros de QoS;
- as regras de controle *fuzzy* são normalmente mais simples e freqüentemente requer-se um número menor de regras, o que propicia eficiência e robustez.

A principal deficiência de sistemas *fuzzy* puros é a incapacidade para aprendizado e reconhecimento de padrões, por não possuir memória. Isto pode ser contornado associando *fuzzy* com técnicas que possibilitam o reconhecimento de padrões como a indução de regras (MITCHELL, 1997). O classificador de pacotes nas arquiteturas orientadas à QoS realiza basicamente uma tarefa de mapeamento, recebendo pacotes pertencentes aos vários fluxos e agrupando-os em classes, segundo os requisitos de desempenho e de utilização de recursos dos fluxos. A identificação das classes é

fundamentalmente uma tarefa de *clustering* e pode ser feita através de métodos de aprendizado não supervisionado, tal como os mapas de Kohonen (KOHONEN, 1982). Uma vez conhecidas as regras, um algoritmo de indução de regras, como o C 4.5 (MITCHELL, 1997) pode ser usado para induzir um classificador de pacotes estruturado na forma de regras.

Após os pacotes terem sido classificados em diferentes filas FIFO, o problema de escalonamento pode ser reduzido a escolher uma das filas não vazias e transmitir o pacote que se encontra em sua cabeça. Este procedimento simplifica bastante o problema, pois não se faz necessário reordenação de pacotes dentro de cada classe. O escalonador de pacotes pode ser implementado como um controlador *fuzzy*, utilizando entradas como descritores de tráfego dos fluxos, o nível de utilização de recursos, o grau de justiça e outras características do ambiente. A saída do controlador irá definir a fila a partir da qual o próximo pacote será transmitido. A Figura 1 apresenta um esboço da proposta.

**Figura 1. Escalonador *fuzzy* baseado em aprendizado.**



A grande inovação desta proposta é o emprego de algoritmos de indução de regras como o C 4.5 e C 5.0 (MITCHELL, 1997) para inferir as principais regras do controlador *fuzzy*, segundo as etapas abaixo:

1. um modelo de rede apropriado é escolhido, definindo-se os roteadores, hosts e características dos enlaces;
2. pacotes IP são injetados em um escalonador de pacotes randômico. Após o pacote ter sido escalonado (ou não, no caso de descarte), eles podem ser rotulados com um ou mais dos seguintes atributos: atraso de fila, pacote descartado ou não etc. As informações sobre o pacote escalonado ou não, juntamente com os demais atributos do ambiente no momento, são armazenadas;
3. usando a informação coletada na etapa 2, emprega-se o algoritmo C 4.5 ou C 5.0 para induzir regras de decisão a partir das informações obtidas;
4. a partir das regras obtidas, escolhe-se as mais efetivas para serem usadas no controlador *fuzzy*;
5. as regras iniciais podem refinadas e novas regras podem ser adicionadas ao controlador, usando métodos que levem em consideração o conhecimento humano.

As principais regras do escalonador *fuzzy* são obtidas a partir das decisões tomadas por um escalonador randômico durante um intervalo de tempo apropriado, usando um algoritmo de indução de regras. Por traz de cada uma das decisões tomadas de modo aleatório pelo escalonador randômico encontram-se regras mais ou menos efetivas para o problema. Cabe ao algoritmo de indução de regras descobrir quais são as regras mais úteis. As regras obtidas são de fácil compreensão e podem ser adaptadas pelo projetista usando conhecimento humano. Este procedimento não pode ser facilmente realizado quando são utilizados métodos de aprendizado do tipo caixa-preta, tais como as redes neurais (HAYKIN, 2001). Esta proposta para obtenção de escalonadores de pacotes *fuzzy* baseados em aprendizado de máquina foi apresentada originalmente em Melo Jr. & Adán Coello (2000).

Uma vez obtido um escalonador de pacotes, o mesmo pode ser avaliado escalonando-se um novo conjunto de pacotes com o escalonador *fuzzy* obtido e com o escalonador randômico. O desempenho dos dois escalonadores pode ser comparado de acordo com os parâmetros de QoS considerados. Cabe ressaltar que a metodologia proposta possui uma menor dependência de definições subjetivas do comportamento do escalonador por parte de projetistas.

Outras técnicas de aprendizado de máquina, além das árvores de decisão, podem ser empregadas como um mecanismo de predição para o estado da rede em um futuro próximo, possibilitando tomar decisões de modo antecipado, de modo a melhor atender aos requisitos de QoS das aplicações. Uma

possibilidade é utilizar as redes neurais, que constituem um método prático e geral para aprender funções dos mais variados tipos a partir de exemplos (HAYKIN, 2001). As redes neurais são robustas em relação à presença de erros nos dados de treinamento e são aplicadas nas mais diversas áreas: reconhecimento de padrões, controle, finanças etc. Qualquer função pode ser aproximada com uma precisão arbitrária por uma rede *feedforward* (forma direcionada acíclica) com 3 camadas de unidades. A aproximação das funções é realizada fazendo-se uma busca em um espaço de hipóteses contínuo, ao passo que outros métodos de aprendizado, tais como árvores de decisão, se baseiam em representações discretas, o que diminui a capacidade de generalização dos mesmos. Com uma rede neural, a função a ser aprendida é definida sobre instâncias do problema, as quais são descritas através de um vetor de características predefinidas. Os atributos de entrada podem ser independentes ou altamente correlacionados, e podem possuir qualquer valor real. Uma vez aprendida uma função, a sua avaliação pela rede neural é em geral muito rápida.

Quando as redes neurais são aplicadas a séries de dados temporais, onde as saídas da rede no tempo  $t$  são usadas como entradas no tempo  $t+1$ , faz-se necessário empregar um tipo especial de rede neural, as chamadas redes recorrentes. As redes recorrentes possuem uma forma direcionada cíclica e de modo geral são mais difíceis de treinar e não generalizam de modo tão confiável quanto as redes *feedforward*; contudo, são muito importantes pois o seu poder de representação considera as entradas ao longo de uma janela de tempo arbitrária no passado para predizer as saídas no tempo  $t+1$ .

## 8. Conclusões

Este trabalho apresentou inicialmente as características da Internet atual e dos novos serviços de rede que estão sendo propostos no âmbito de duas das principais arquiteturas Internet orientadas à QoS. A seguir, discutiu-se diversos desafios encontrados para viabilizar a Internet de Próxima Geração, com destaque para a implementação de escalonadores de pacotes, um elemento de importância fundamental para os novos serviços. Uma proposta para o desenvolvimento de escalonadores de pacotes em arquiteturas Internet orientadas à QoS, usando técnicas baseadas em aprendizado de máquina e controle *fuzzy* foi apresentada. A principal inovação desta proposta é a utilização de algoritmos de indução de regras para obtenção das principais regras do controlador *fuzzy*.

Uma vez obtido um escalonador de pacotes *fuzzy*, o mesmo pode ser usado em ambientes diversos, com padrões de tráfego diferentes, requerendo para isso basicamente uma calibração das funções de pertinência. Ferramentas de Mineração de Dados (MITCHELL, 1997) podem ser empregadas para explorar os dados disponíveis, encontrando os pontos críticos no espaço de entrada e de saída do controlador *fuzzy*. O grau de pertinência nos pontos críticos pode ser considerado 1.0 e funções de pertinência apropriadas podem ser obtidas decaindo o valor para 0.0 de modo adequado.

O simulador de código aberto NS (ISI, 2005) é atualmente uma ferramenta amplamente disseminada na comunidade de redes de computadores, e pode ser empregado para desenvolver e validar as idéias originadas a partir do presente trabalho.

## 9. Referências bibliográficas

BLAKE, S.; et al. *An Architecture for Differentiated Services*. I RFC 2475, IETF DiffServ Working Group, December 1998.

BRADEN, R.; CLARK, D. & SHENKER, S. *Integrated Services in the Internet Architecture: an Overview*. In RFC 1633, IETF Network Working Group, June 1994.

CLARK, D.D.; SHENKER, S. & ZHANG L.. *Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism*. In Proceedings of SIGCOMM'92. pp. 14-26, 1992.

COMER, D.E. *Internetworking with TCP/IP*. Vol. 1. 4th. ed. New Jersey: Prentice Hall, 2000.

DOVROLIS, C. & RAMANATHAN, P. *A Case for Relative Differentiated Services and the Proportional Differentiation Model*. In IEEE Network Magazine. Vol.(13):26-34, Sept./Oct. 1999.

DOVROLIS, C.; STILIADIS, D. & RAMANATHAN, P. *Proportional Differentiated Services: Delay Differentiation and Packet Scheduling*. In IEEE/ACM Transactions on Networking. Vol.(10):12-26, nº 1, February 2002.

FERGUSON, P. & HUSTON, G. *Quality of Service*. New York: John Wiley and Sons, 1998.

FERNANDEZ, M.P.; PEDROZA, A.C.P. & REZENDE, J.F. *Qualidade de Serviço em um Domínio DiffServ através de Gerenciamento Baseado em Políticas*. In Anais do 19º Simpósio Brasileiro de Redes de Computadores. Maio 2001.

\_\_\_\_\_. *Otimização de Controlador Fuzzy para Provisionamento de Recursos em Ambiente DiffServ através de Algoritmo Genético*. In Anais do 20º Simpósio Brasileiro de Redes de Computadores. Maio 2002.

GOMIDE, F. & PEDRYCZ, W. *An Introduction to Fuzzy Sets - Analysis and Design*. Cambridge: MIT Press, 1998.

HAYKIN, S. *Redes Neurais: Princípios e Prática*. 2a. ed. Porto Alegre: Bookman, 2001.

HEINANEN, J.; BAKER, F.; WEISS, W. & WROCLAWSKI, J. *Assured Forwarding PHB Group*. In RFC 2597, IETF Network Working Group, June 1999.

ISI (Information Sciences Institute). *The Network Simulator: ns-2*. Disponível em: <http://www.isi.edu/nsnam/ns>. Acesso em 01/junho/2005.

JACOBSON, V.; NICHOLS, K. & PODURI, K. *An Expedited Forwarding PHB*. In RFC 2598, IETF Network Working Group, June 1999.

KESHAV, S. *An Engineering Approach to Computer networking: ATM Networks, the Internet, and the Telephone Network*. Boston: Addison-Wesley, 1997.

KILKKI, K. *Differentiated Services for the Internet*. Indianapolis: Macmillan Publishing Co., 1999.

KOHONEN, T. *Self-Organized Formation of Topologically Correct Feature Maps*. In Biological Cybernetics, pp. 59-69, nº 43, 1982.

LEUNG, M.K.H.; LUI, J.C.S. & YAU, D.K.Y. *Characterization and Performance Evaluation for Proportional Delay Differentiated Services*. In Proceedings of the 2000 International Conference on Network Protocols. pp. 295-, 2000.

MELO JR., A. & ADÁN COELLO, J.M. *Packet Scheduling Based on Learning in the Next Generation Internet Architectures*. In Proceedings of The Fifth IEEE Symposium on Computers and Communication, pp. 673-678, July 2000.

MELO JR., A.; MAGALHÃES, M.F. & ADÁN COELLO, J.M. *WWTP: Um Escalonador de Pacotes para o Modelo de Diferenciação Proporcional Baseado em Janelas de Medidas*. In Anais do 23º Simpósio Brasileiro de Redes de Computadores. Vol.(2):773-782, Maio 2005.

MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997.

MUNAKATA, T. & JANI, Y. *Fuzzy Systems: An Overview*. In Communications of the ACM. Vol. (37):69-76, nº 3, March 1994.

NEGROPONTE, N. *Being Digital*. New York: Knopf Publishing, 1995.

NICHOLS, K. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. In RFC 2474, IETF Network Working Group, December 1998.

SHENKER, S. *Fundamental Design Issues for the Future Internet*. In IEEE Journal on Selected Areas in Communications. Vol.(13):1176-1188, nº 7, September 1995.

SHENKER, S.; PARTRIDGE, C. & GUERIN, R. *Specification of Guaranteed Quality of Service*. In RFC 2212, IETF Network Working Group, September 1997.

WANG, L.X. & MENDEL, J. *Generating Fuzzy Rules by Learning From Examples*. In IEEE Transactions on Systems, Man and Cybernetics. Vol.(22):1414-1427, July 1992.

WROCLAWSKI, J. *Specification of the Controlled-Load Network Element Service*. In RFC 2211, IETF Network Working Group, September 1997.

ZHANG, L.; et al. *RSVP: A New Resource Reservation Protocol*. In IEEE Network. pp. 8-18, September 1993.

ZHANG, R. & MA, J. *Fuzzy QoS Management in Diff-Serv Networks*. In Proceedings of The IEEE International Conference on Systems, Man and Cybernetics. pp. 3752-3757, August 2000.