Universidade Federal de Santa Catarina – UFSC
Programa de Pós-Graduação em Ciências da Computação – PPGCC
Disciplina: Sistemas Operaciaonais
Aluno: Luiz Henrique Vicente

Serviços Diferenciados em Sistemas Operacionais Linux

Introdução

O uso de tecnologias tem avançado muito e cada vez mais está incorporando à vida das pessoas. Dentro destas tecnologias, temos as ferramentas de aplicações multimídia como o grande pivô para popularização dos meios de comunicação eletrônica.
O grande problema é que os recursos de rede não avançam na mesma proporção que as ferramentas que exigem cada vez mais destes para poderem oferecer tais serviços com qualidade. O uso do padrão ethernet para transmissão de voz e vídeo passa pela barreira de competir com outras aplicação de comunicação de dados existentes hoje pela Internet.
A fim de impor uma certa qualidade aceitável para que as aplicações multimídia possam efetuar transmissões decentes de conteúdo, o uso de técnicas que estabeleçam um mínimo de qualidade de tráfego na rede sse faz necessário. Com isso, surgiu o termo QoS (quality of service) que dispõem padrões de qualidade para comunicações multimídia através de reserva de rede para tais serviços.
Dentre os padrões existentes hoje, temos o InteServ e o DiffServ como os principais meios de buscar uma garantia na qualidade de serviço ofertado na rede. O primeiro é baseado na reserva de recursos de rede na comunicação entre dois pontos utlizando RSVP como mecanismo de difusão dos dados QoS. O segundo opera no fluxo de pacotes que trafegam na rede, através de uma sinalização contida em cada um deles e baseado em um acordo de nível de serviço estabelecido entre o cliente e o provedor de serviços. O DiffServ é o padrão mais usado e por isto foi escolhido para os estudos neste trabalho.

O padrão DiffServ

O modelo de arquitetura do DiffServ é aquele no qual o tráfego que entra na rede é classificado, possivelmente condicionado nos limites da rede e destinado a um tipo de comportamento específico BA (Behavior Aggregaté). Cada BA é identificado através de um código DSCP. No núcleo o da rede, os pacotes são roteados de acordo com o comportamento por salto (PHB – Per Hop Behavior) associado a eles via DSCP. O funcionamento da arquitetura Diffserv começa pela entrada de fluxos de dados que passa primeiramente no roteador de borda. Inicialmente ele passa por um processo de classificação de acordo com o campo de DSCP (TOS ou TCF), que dependo do resultado poderá ser marcado ou remarcado devido a sua taxa de envio para ser transmitido. A etapa seguinte é a de modelagem do tráfego que irá descartar aquele determinado dado que exceder a taxa limite configurada para envio, segundo o mesmo autor. O classificador seleciona os pacotes baseado nas

informações de seus cabeçalhos. São definidos dois tipos de classificadores, o primeiro classifica os pacotes pelo byte DS ou qualquer outra informação contida no cabeçalho do pacote já o segundo classifica os pacotes apenas pelo comportamento agregado associado ao DSCP. Uma vantagem, além da agregação dos fluxos em classes, que torna esta arquitetura bastante escalável é o fato de que toda a classificação complexa é feita nas bordas da rede. O medidor de tráfego confere se os pacotes classificados estão de acordo com o perfil de cada tráfego, especificado no acordo de nível de serviço (SLA) entre provedor e cliente. E ainda o medidor tem a função de repassar informações dos pacotes para os elementos funcionais, a fim de que seja realizada uma ação sobre o pacote que está ou não dentro do perfil. O marcador marca o pacote com o DSCP adequado de acordo com o resultado da classificação do mesmo. O condicionador faz a moldagem do tráfego, baseado em alguma propriedade configurada e especificada no SLA. Por exemplo, o tráfego pode ser moldado baseado em uma taxa de pico acordada entre cliente e provedor, sendo que os pacotes que não estão dentro do perfil são descartados.

A arquitetura de serviços diferenciados (DiffServ) provê uma estrutura de diferenciação dos serviços que trafegam na rede. Os serviços podem ser diferenciados por várias maneiras, tais como atraso na rede ou perda de pacotes. Nele os pacotes antes de entrarem são previamente classificados pelo emissor, onde ele preenche um campo chamado TOS (Type of Service), que hoje é chamado de DSCP (DiffServ Code Point), encontrado no cabeçalho IP junto a classe de dados

Quando um pacote entra numa rede DiffServ, as políticas são aplicadas pelo receptor. Os pacotes que necessitam de mais prioridade possuem maior valor no campo DSCP. No caso de excesso de tráfego, o cliente poderá ser cobrado por este excesso e o roteador tem estabelecido uma política de descarte com base na freqüência de *drop*, se não houver espaço em *buffer*.

Cada roteador, aqui também chamado de Per-Hop-Behavior (PHB), pode definir seu próprio valor a cada serviço diferenciado ofertado, mas existem valores padrões que podem ser utilizados nos PHBs.

A estrutura geral do DiffServ está definida como segue na figura1.

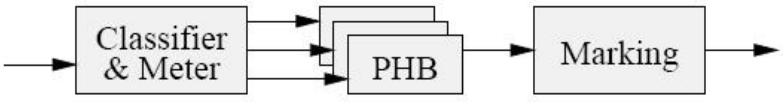


Figura 1: Encaminhamento Geral do DiffServ

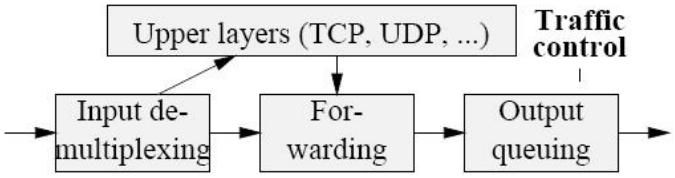
Nesse porcesso, o *Marking* pode acontecer em outros estágios, depende da implementação.

O DiffServ distingue dois tipos de classificação, um baseado somente no campo DS e outro que pode analisar o campo por inteiro.

A marcação é o processo de ajuste e modificação do campo DS. Ela se faz necessário em vários casos, como por exemplo quando um pacote vem de uma rede não DiffServ para uma rede DiffServ, seu campo DS deve ser ajustado para o DSCP apropriado.

Controle de Tráfego no Linux

A figura 2 mostra como o kernel processa os dados recebidos e gera novos dados para serme enviados a rede.



Na seção *forwarding* é feita a seleção da interface de saída. Feito isto, os pacotes são enfileirados na sua respectiva interface de saída. O controle de tráfego pode, entre outras coisa, definir se o pacote será colocado na fila ou descartado, em qual ordem eles serão enviados, pode atrasar o envio dos pacotes.
Cada dispositivo de rede possui sua disciplina de enfileiramento que controla como os apcotes enfileirados no dispositivo são tratados. Uma forma simples de enfileirar pode ser o de uma fila simples, onde os pacotes são ordenados na fila e enviados tão rápido quanto puder o fazer sua respectiva placa de rede. Pode-se ainda ser usado um filtro com diferenciação entre classes de pacotes para cada processo.

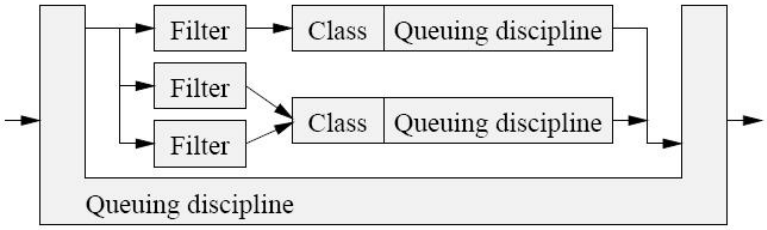


Figura 3: Enfileiramento com Múltiplas Classes

Os pacotes inciam o processo através de uma função de chamada de uma disciplina de enfileiramento. Com isso é feito uma procura nos identificadores de filtros aquele que se iguala ao pedido. Se um pacote não possuir um filtro de enfileiramento que se iguale aos existentes na lista é atribuído um filtro padrão a ele. Em geral, cada classe detém uma fila, mas é possível que exista várias classes na mesma fila.

DiffServ no Linux

Uma boa base de implementação de um roteador com o campo DS é o sistema operacional linux e suas distribuições. Ele roda em um PC padrão seu código é aberto e disponível. Além disso, desde a versão 2.1 do seu kernel ele já suporta uma variedade de disciplinas de enfileiramento para os dispositivos de rede, além das funções de roteamento.
Para entender como o DiffServ é realizado, é preciso compreender as funções básicas de rede. Para tanto, segue uma explanação da implementação de rede IP em Linux com base na figura 4, que segue:

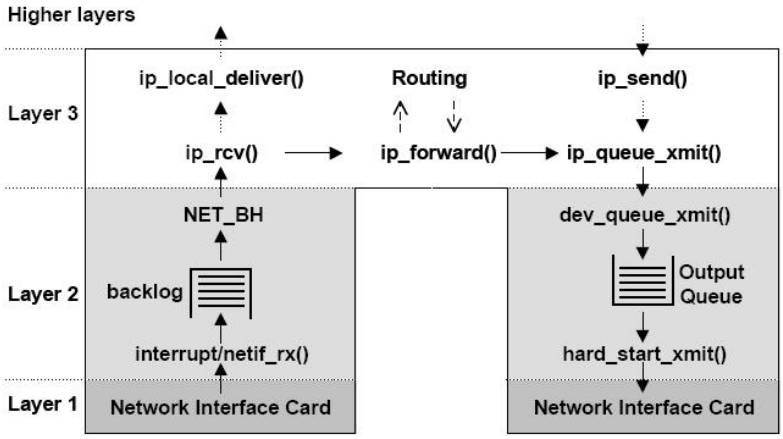


Figura 4: O curso do pacote pelo sistema

O pacote recebido na placa de rede gera uma interrupção de hardware. Co isto uma rotina de manipulação de interrupção é chamada e é determinada o tipo de interrupção. Para interrupções geradas por entrada de pacotes uma adicional rotina de manipulação é chamada e os pacotes são simplesmente copiados da placa para uma estrutura de *socket* interno e um procedimento é chamado para tratá-los. Os pacotes de todos os adaptadores são concentrados em uma filacentral (*backlog*).
Numa segunda parte é manipulada pelo *NET_BH* que normalmente é chamado pelo kernel. Primeiro, ele checa há pacotes aguardando a transmissão em uma fila de saída de qualquer adaptador. Se houver, ele será processado por um dado limite de tempo. Na sequencia, o próximo pacote da fila *backlog* é chamado e determinado o protocolo de manipulação, que neste caso é o IP. Ele verfifica a exatidão do pacote e quaisquer opções de processos existentes. Ele ainda remonta o pacote IP original através de seus fragmentos e verifica se chegou a seu destino final. Em último caso, o pacote é entregue lcalmente, caso contrário ele é encaminhado e enviado ao seu destino. Em seguida, é identificada a placa de rede correta e o pacote é encaminhado. Se houver entradas válidas na tabela de rotas, algumas operações finais são tratadas, como decremento do valor do TTL e recálculo do *checksum* no cabeçalho IP. Feito isto, os pacotes são enfileirados para saída. Neste ponto, uma disciplina de enfileiramento especial pode ser chamada. Assim, a transferência é feita.
A figura 5 mostra como pode ser feito o enfileiramento de saída dos pacotes utilizando diferentes disciplinas de tratamento dos pacotes.

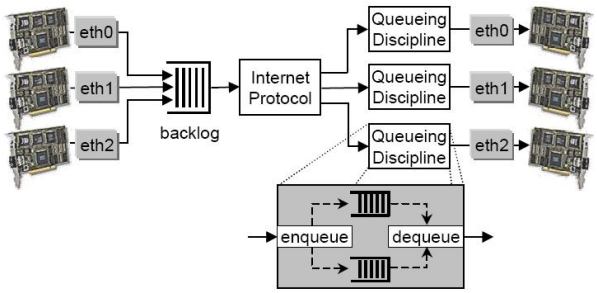


Figura 5: Enfileiramento de sáida com diferentes disciplinas

O kernel do Linux já possui várias disciplinas de enfileiramento tais como o CBQ (Class Based Queueing), WFQ (Weighted Fair Queueing) ou o RED (Random Early Detection).
Uma outra forma de se fazer as decisões está no uso de módulos de kernel para implementação de funcionalidade DS. Módulos de kernle não precisam estar presente em todo o tempo no kernel,

então o kernel pode rodar sem eles se estes não estiverem sendo usados. O kernle recompilado com esta funcionalidade quando reiniciado já carrega parte do módulo, o que simplifica uma nova carga do módulo e o tempo de desenvolvimento.

A implementação deste método compreende dois módulos: Clssificação de Serviços Diferenciados (DSC) e Enfileiramento de Disciplinas de Serviços Diferenciados (DSQD). O DSC classifica os pacotes baseado em uma combinação de um ou mais campos do cabeçalho e associa seu respectivo perfil. O BSQD inclui um medidor de tráfego realizado por um *token bucket* e um *leaky bucket,* um marcador, moderador de tráfego e um descartador.

O módulo de classificação de serviços diferenciados é responsável pela classificação de qualquer pacote entrando e encontrar o correspondente perfil de tráfego para ele. A classificação é feita identificando a entrada no adaptador e o campo DS para permitir uma política de tráfego de acordo com o nível de serviço estabelecido com o provedor. O perfil de dados compreende o parâmetro de tráfego para um serviço específico. Ele é identificado por um número único.

A figura 6 a seguir mostra o funcionamento destes módulos na classificação dos pacotes.

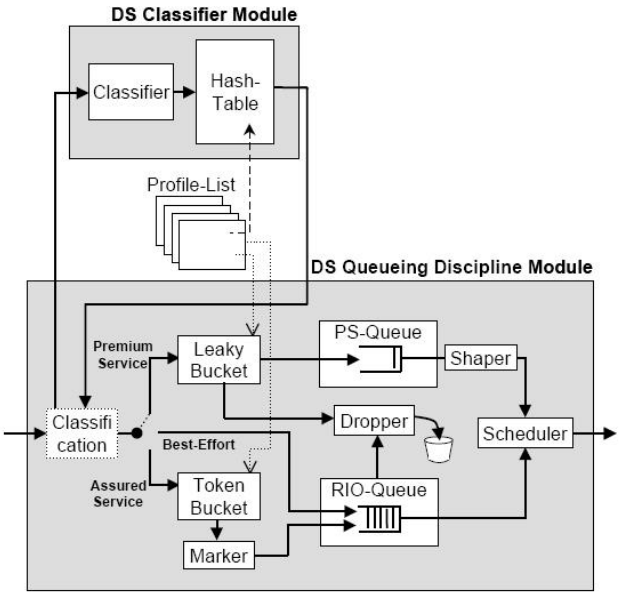


Figura 6: Arquitetura de Implementação dos Módulos de Classificação

O Linux é um sistema operacional que possui suporte à comunicação de dados, inclusive à implementação da rede TCP/IP. A partir da versão 2.4 do Kernel, o LINUX disponibiliza a implementação de arquiteturas de qualidade de serviço e de um mecanismo para controle de tráfego em redes de computadores, o IPROUTE2, o qual utiliza a linguagem Traffic Control - TC , que permite configurar classes de serviços, hierarquia de classes, filtros, marcadores e escalonadores de pacotes, largura de banda, dentre outros. O processo de controle de tráfego tem como principais componentes os descritos a seguir: a disciplina de enfileiramento (Queuing Disciplines – QDISC), controla como os pacotes devem ser enfileirados de acordo com o plano de recursos da rede, utilizando algoritmos como DSMARK, CBQ, FIFO, etc.; as classes – identifica um conjunto de características que definem o comportamento dos recursos de uma rede, e pode ou não ser definida, dependendo da QDISC a ser utilizada; o filtro classifica os pacotes utilizando as suas propriedades, como o TOS, o endereço IP, o número da porta, dentre outras. Os filtros são utilizados pelas classes de serviço e/ou pelas QDISCS; e, por fim, o policiamento (police), certifica-se de que o tráfego utilizado não ultrapasse a largura de banda reservada e permite acionar mecanismo que consiga

regular o tráfego.
Na busca da implementação de DiffServ em Linux, tem-se o tratamento dos elementos tradicionais de controle de tráfego. O *Shaping* é o mecanismo pelo qual os pacotes são retardados antes que uma fila de transmissão encontre uma taxa (raté) de velocidade desejada para envio. Os escalonadores (*Scheduling*), são mecanismos pelos quais os pacotes são organizados (ou reorganizados) em uma fila de entrada ou saída. O agendador (*scheduler*) mais utilizado é o FIFO. Os classificadores (Classifying) são mecanismos pelos quais os pacotes são separados para um tratamento diferenciado. Durante o processo de aceitação, roteamento e transmissão de um pacote, o dispositivo de rede pode classificá-lo de várias maneiras. O policiamento (*Policing*) funciona como um elemento do controle de tráfego, é simplesmente um mecanismo pelo qual o tráfego pode ser limitado. Este conceito é geralmente usado nas bordas das redes (em um firewall por exemplo) para garantir que uma estação não consuma mais do que a banda destinada à ela.
O *qdisc* é o agendador do Linux. Toda e qualquer interface precisa de um agendador de algum tipo; o padrão utilizado é o FIFO. Outros *qdiscs* disponíveis no Linux redistribuirão os pacotes nas filas de acordo com suas regras de agendamento. *Qdisc* é o elemento principal no Controle de Tráfego do Linux, também pode ser chamado de disciplina de enfileiramento.

Ferramentas para Implementação do Diffserv no Linux

As ferramentas utilizadas para a implementação do diffServ no Sistema Operacional Linux já o acompanha desde a versão que utiliza a versão do Kernel 1.1. Nele temos a ferramenta *IPTABLES* que ela insere regras e políticas de entrada e saída de acesso a pacotes dentro de um domínio IP, para que essas regras não se percam na hora de reiniciar a maquina ela utiliza de scripts que são lidos a cada inicialização. No Linux, a filtragem de pacotes é feita no kernel, existem algumas funções de manipulação de pacotes, porém, essencialmente esse módulo lê os cabeçalhos dos pacotes e decide o que fazer com ele. O IPROUTE2 é o principal elemento para a implantação de uma rede DiffServ. Ele é uma ferramenta para configuração das rotas para os tráfegos dentro de um domínio. Para isso, é preciso a configuração dos endereços dos hosts, as tabelas de roteamento nas máquinas que servirão como roteadores, e das classes de serviço, policiamento e filtros. O pacote IPROUTE2 vem com dois comandos principais: ip, utilizado para o hosts, tabelas de roteamento nas máquinas e comando tc classes de serviço, policiamento e filtros. Para a configuração das classes de serviço, policiamento e filtros, exige uma série de passos como definição e remoção de classes, filtros e policiamento que normalmente são criados scripts para facilitar a configuração de ambientes específicos dos roteador de borda, núcleo e manutenção/atualização dos mesmos.

Conclusão

Os conceitos aqui apresentados são baseados em pesquisas implementadas anteriormente. Pode-se notar que há uma grande preocupação em oferecer melhor qualidade de serviço tornando alguns serviços possíveis de serem aplicados nas estruturas de redes hoje oferecidas. Mais do que

implementar uma boa política de acesso ao meio, uma ferramenta de QoS deve oferecer formas de controle de banda e formas mais ágeis de se estabelecer qualidade sem perda de capacidade.
Assim, os sistemas operacionais, aqui em especial os sistemas da plataforma Linux, são peças fundamentais na prática de políticas de QoS. Toda e qualquer aplicação faz requisições ao sistema que o coloca em contato com a máquina e esta como os meios de acesso a rede. Portanto, é vital importância que os sistemas operacionais desenvolvam ferramentas para estabelecer uma melhor eficiência no acesso aos recursos de rede e assim agilizar as comunicações, oferecendo qualidade nos serviços multimídia providos através de redes baseadas em melhor esforço.

Bibliografia

Wikipédia, a enciclopédia livre. DiffServ. Disponível em: http://pt.wikipedia.org/wiki/Diffserv

Almesberger, W.; Salim, J.H.; Kuznetsov, A. **Differentiated Services on Linux**. http://ieeexplore.ieee.org/iel5/6675/17972/00830189.pdf?tp=&arnumber=830189&isnumber=17972

BLESS, R.; WEHRLE, K. **Evaluation of Differentiated Services using an Implementation under Linux**. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=1656705EB13307F7EE4575750C808BA8?doi=10.1.1.20.1235&rep=rep1&type=pdf

MACHADO , L. I. **Implementação de QoS em redes IP com Diffserv em um ambiente Linux.** http://net2.santoagostinho.edu.br/biblioteca/monografias/30.pdf