

phpTCAdmin: Uma Solução Para o Planejamento e Implementação de Qualidade de Serviço em Redes de Computadores

***Abstract.** Quality of Service policy management under GNU/Linux systems is extremely complex from the user's point of view, since available software for traffic control rules management are command line based, thus not intuitive. This paper presents phpTCAdmin, a proposal for Computer Network's Quality of Service policy management.*

***Resumo.** A gerência de políticas de Qualidade de Serviço nos sistemas GNU/Linux é extremamente complexa do ponto de vista do usuário, uma vez que os softwares disponíveis para a manipulação das regras de controle de tráfego são baseados em linhas de comando, portanto pouco intuitivos. Este artigo apresenta o phpTCAdmin, um software para a gerência de políticas de Qualidade de Serviço em redes de computadores.*

1. Introdução

O uso de software livre na gerência e administração de redes de computadores é uma constante no atual contexto da maioria das redes de computadores institucionais e acadêmicas. As empresas investem na contratação e no aperfeiçoamento de recursos humanos especializados nas mais novas tendências em software livre, e na migração de plataformas proprietárias para plataformas livres, com o principal objetivo de redução de custos, visando à obtenção de software que melhor se adapte a requisitos de eficiência e confiabilidade, entre outros [CORDEIRO 2006].

Este artigo tem como objetivo endereçar um dos problemas enfrentados pelos administradores de redes – a definição de políticas de Qualidade de Serviço (do inglês *Quality of Service*, *QoS*) em redes de computadores – ao propor uma tecnologia baseada nas premissas de software livre para auxiliar de forma eficiente a gerência de políticas de QoS em redes de computadores.

O restante do artigo está dividido em 4 seções. A Seção 2 discute alguns dos aspectos relacionados à políticas de QoS em redes de computadores. A Seção 3 discute os principais trabalhos relacionados. A Seção 4 apresenta o phpTCAdmin, um software para o auxílio na gerência de políticas de QoS. A Seção 5 apresenta um estudo de caso do software proposto, enquanto que a Seção 6 conclui o artigo, apresentando possíveis trabalhos futuros.

2. Provisionamento de Qualidade de Serviço em Redes de Computadores

Diversas aplicações, por exemplo VoIP e videoconferência, requerem que as redes de computadores forneçam garantias de QoS específicas, como por exemplo disponibilidade de banda, e *jitter* específicos. Por outro lado, existe por parte do administrador de rede a necessidade de garantir que estações de trabalho tenham um acesso justo a um enlace de comunicação contratado, de

modo que nenhuma estação venha a monopolizar o enlace para uso próprio, e definindo prioridades de acesso sempre que necessário.

Em redes que não fornecem garantias fim a fim, os escalonadores de pacotes devem considerar que (i) a capacidade computacional do servidor deve ser suficiente para o processamento dos dados, não ocasionando atrasos (apesar de o kernel considerar o tráfego de pacotes como prioridade de tempo real, não há garantias explícitas de disponibilidade de recursos); e (ii) o enlace de comunicação deve ter uma taxa de transmissão máxima fixa, e esta capacidade deve ser conhecida e alcançada sempre que necessário, ou seja, deve-se observar a máxima taxa de transmissão alcançada.

A utilização de algoritmos de controle de tráfego permite solucionar problemas cotidianos da gerência de redes, por exemplo limitar as demandas de tráfego de uma estação ou otimizar o uso de um *link* entre as estações de uma rede, assim como prover uso dos mecanismos de QoS *diffserv* [BLAKE 1998] através do algoritmo *DSMARK*, e *intserv* [BRADEN 1994], através do suporte ao protocolo *RSVP*.

O provisionamento de QoS em redes de computadores pode ser implementado utilizando principalmente algoritmos de enfileiramento de pacotes, entre outros mecanismos. O *kernel* (núcleo) do sistema operacional *Linux* provê esta capacidade através do utilitário chamado *tc* (*traffic control*) [OLSHEFSKI 2001], o qual realiza o controle do tráfego de rede [ALMESBERGER 1999] através do uso de algoritmos de controle de tráfego.

3. Trabalhos Relacionados

Entre os softwares que interagem com a *TCAPI* [OLSHEFSKI 2001] destaca-se o *CBQ.init traffic management script* [CBQINIT 2007], cujo objetivo é gerar regras para o utilitário *tc* a partir de arquivos de configuração. Este projeto limita-se a manipular regras com algoritmo *Class Based Queuing* (CBQ) [FLOYD 1995], não apresentando mecanismos de fomentem a compreensão deste algoritmo. Outra deficiência é que ele não permite a criação de classes em tempo de execução, sendo necessário que todas as regras sejam reaplicadas.

O projeto *HTB.init setup script* [HTBINIT 2007] é semelhante ao anteriormente citado, sendo que ele manipula regras para o algoritmo *Hierarchical Token Bucket* (HTB). Este software, no entanto, não apresenta uma interface que facilite a implementação de regras de QoS utilizando a *TCAPI*.

4. O Software phpTCadmin

O phpTCadmin é uma solução baseada nas premissas de software livre, distribuído sob licença GNU GPL [GPL 2006], e que permite a administração remota das políticas de QoS implementadas na rede através de uma interface *web* seguindo padrões W3C (XHTML/CSS) [W3C 2007]. Ele facilita a criação de disciplinas de enfileiramento, classes e classificadores em complexas estruturas; permite exportar e importar regras de QoS definidas previamente; e manipula a *TCAPI* em tempo de execução, indicando os parâmetros para cada algoritmo, sem a necessidade de reaplicação de todas as regras de QoS.

O software phpTCadmin possui dois componentes principais: um que interage com a *TCAPI* através do utilitário *tc*, e outro que exhibe de forma

intuitiva para o usuário as disciplinas de enfileiramento, classes e classificadores. A arquitetura do phpTCAdmin é mostrada na Figura 1.

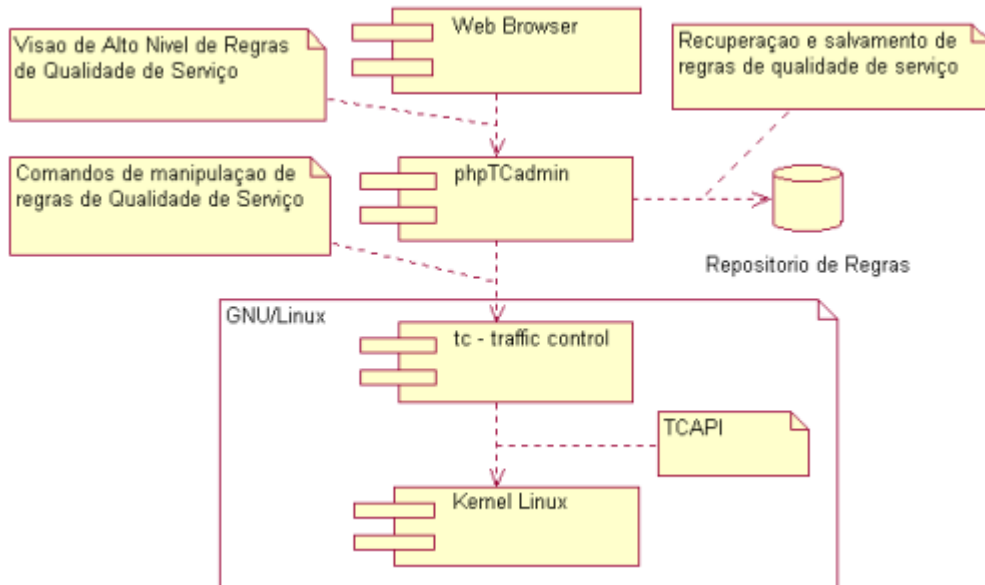


Figura 1. Arquitetura do software phpTCAdmin.

O phpTCAdmin apresenta quatro modos de visualização dos componentes do TCAPI: (i) *normal*, no qual a exibição ocorre de forma semelhante ao retorno da opção *show* do comando *tc*; (ii) *tree*, no qual a visualização dos componentes reflete a forma como as regras estão acopladas para realizar o controle do tráfego; (iii) *natural*, semelhante à opção *tree*, no entanto apresentando os classificadores em uma tabela separada da árvore; e (iv) *debug*, no qual várias informações adicionais são exibidas, por exemplo a taxa de utilização de cada classe.

O modo de exibição *tree* permite a visualização desde a disciplina de enfileiramento principal da interface, suas classes, assim como as disciplinas de enfileiramento que controlam cada classe, conforme ilustrado na Figura 2. O modo de visualização *natural* permite uma perspectiva de visualização das regras de QoS conforme o funcionamento natural da TCAPI, na qual cada pacote é verificado com os classificadores e então enfileirado nas classes que estão interligadas em uma árvore.

```

Tree View
└─ qdisc htb 1: r2q 10 default 1000 direct_packets_stat 0
   └─ class htb 1:1000 root rate 30000Kbit ceil 30000Kbit burst 16593b cburst 16593b
      └─ class htb 1:1001 parent 1:1000 rate 5000Kbit ceil 10000Kbit burst 4Kb cburst 6598b
         └─ class htb 1:1002 parent 1:1001 leaf 1002: prio 0 rate 2000Kbit ceil 5000Kbit burst 2599b cburst 4Kb
            ├── filter protocol ip pref 10 flowid 1:1002 u32 match dst 10.20.0.0/24
            ├── filter protocol ip pref 10 flowid 1:1002 u32 match dst 10.30.0.0/24
            └─ qdisc sfq 1002: parent 1:1002 limit 128p quantum 1514b perturb 10sec
               └─ class htb 1:1003 parent 1:1001 leaf 1003: prio 0 rate 2000Kbit ceil 2000Kbit burst 2599b cburst 2599b
                  └─ class htb 1:1004 parent 1:1001 leaf 1004: prio 0 rate 1000Kbit ceil 3000Kbit burst 2099b cburst 3099b
                     └─ class htb 1:1005 parent 1:1000 leaf 1005: prio 1 rate 10000Kbit ceil 30000Kbit burst 6598b cburst 16593b
                        ├── filter protocol ip pref 11 flowid 1:1005 u32 match dst 10.0.0.0/8
                        └─ qdisc red 1005: parent 1:1005 limit 60Kb min 15Kb max 45Kb

```

Figura 2. Modo de visualização *tree* do phpTCAdmin.

Na interface do phpTCAdmin as informações relevantes – por exemplo o

identificador de classe (*classid*), a taxa de transmissão garantida e máxima, e os dados dos classificadores – são destacadas. A interface também permite interação com cada componente através do mouse, permitindo assim a visualização ou ocultação de informações relativas a cada componente.

O phpTCadmin permite a definição de disciplinas de enfileiramento, classes e classificadores em tempo de execução e sem a necessidade de reaplicação das regras existentes. O phpTCadmin manipula todos os algoritmos disponíveis no *TCAPI* do *kernel Linux*: *Classe Based Queuing (CBQ)*, *Hierarchical Token Bucket (HTB)*, *Hierarchical Fair Service Curve (HFSC)*, *Multi Band Priority Queueing (PRIO)*, *Stochastic Fairness Queueing (SFQ)*, *Token Bucket Filter (TBF)*, *Random Early Detection (RED)*, assim como suporte aos classificadores *Universal 32bit comparisons w/ hashing*, e *Netfilter Mark*, possuindo suporte inicial aos algoritmos *Generic Random Early Detection* e *Differentiated Services marker* e classificadores de suporte a *RSVP*. O phpTCadmin é, portanto, uma suíte completa para a gerência das políticas de QoS no sistema GNU/Linux, ao fornecer todas as funcionalidades que podem ser acessadas via linha de comando, porém de uma forma simples e intuitiva.

5. Estudo de Caso

A evolução dos meios de transmissão para *intranets*, por exemplo as redes *Gigabit Ethernet*, tornou a comunicação em LANs extremamente eficiente. Entretanto, essa evolução não ocorreu na mesma proporção dos *links* de acesso à *Internet*. Dessa forma, é indispensável o controle do tráfego gerado pela *intranets* com destino à *Internet*, seja para definir prioridades de acesso, seja para dividir o uso do *link* de *Internet* igualmente.

O phpTCadmin está em uso em uma empresa de processamento de dados de uma determinada cidade, a qual possui uma rede interna *Gigabit Ethernet* e interliga cerca de 400 *sites* através de *links* privados, além de 20 novos pontos em implantação através da rede metropolitana dessa mesma cidade. A justificativa de uso do phpTCadmin nesse contexto é a necessidade de controlar de forma diferenciada o acesso ao *link* de *Internet* de 30 Mbit entre todos os usuários da rede e o alto número e a complexidade das regras de QoS específicas para cada usuário.

Utilizando o phpTCadmin, além de obter uma melhor compreensão das políticas de QoS previamente definidas, foi possível definir de maneira prática regras específicas para cada uma das interfaces de rede presentes no roteador principal da rede, o que resultou em aumento da produtividade. A Figura 3 ilustra o primeiro procedimento realizado, a aplicação do algoritmo *Hierarquical Tocken Bucket*, com uma classe principal de 30 Mbit.

Queue Discipline eth0 root

Algorithm: Hierarchical Token Bucket (HTB) ▼

Interface data rate: 30 Mbits/s ▼

Finish

Figura 3. Definição do algoritmo HTB como disciplina raiz da interface principal.

O segundo procedimento realizado para a estruturação das políticas de QoS da empresa foi a criação de classes para cada usuário, com objetivo de garantir uma determinada taxa de transmissão, conforme ilustra a Figura 4.

Class eth0

Parent:

Assured Rate:

Max Rate: Priority:

Figura 4. Criação de classes específicas para cada usuário.

O terceiro procedimento foi a criação dos classificadores, associando um fluxo de dados de um usuário a uma determinada classe, conforme ilustrado na Figura 5.

Classifier eth0

Algorithm:

Src Address: Src Port:

Dst Address: Dst Port:

Classid: Priority match:

Figura 5. Associação do tráfego com destino a rede 10.0.0.0/24 à classe 1:1002.

A utilização do phpTCadmin permitiu, portanto, um melhor controle das políticas de QoS por parte do administrador da rede e, conseqüentemente, um aumento na produtividade da empresa, bem como um eficiente atendimento às demandas de rede dos clientes institucionais e corporativos da empresa.

6. Considerações Finais e Trabalhos Futuros

O uso de um software para a compreensão, planejamento, definição e gerência de políticas de QoS em redes de computadores pode tornar muito mais simples para o administrador de uma rede a tarefa de definição e gerência de políticas de QoS em redes, independentemente da complexidade das políticas a serem aplicadas ou da rede de computadores em si. Além disso, a disponibilidade do software utilizando licenças de software livre permite tanto a redução dos custos de implantação de uma solução de gerência de infraestrutura de redes de computadores, bem como a possibilidade de estudo dos mecanismos envolvidos na definição de políticas de QoS por parte de estudantes e profissionais em geral.

Este trabalho apresentou o software phpTCadmin, um software que visa facilitar a tarefa do administrador de redes em definir e gerenciar a aplicação de políticas de QoS. O uso do software em um ambiente de produção permitiu

validar o mecanismo proposto bem como avaliar o impacto da adoção do software na produtividade dos administradores de redes.

Uma proposta de trabalho futuro é a melhoria da interface com o usuário, de modo que gráficos possam ser gerados a respeito das regras atualmente definidas, regras aplicadas e regras atualmente utilizadas. Outra proposta de trabalho futuro é permitir que o software phpTCadmin possa realizar a gerência de políticas de QoS de forma distribuída, fornecendo uma interface centralizada de gerência e o uso de Web Services [W3C 2007], possivelmente utilizando a integração com SNMP [VIANNA 2006], para a comunicação e configuração de cada um dos roteadores (utilizando GNU/Linux ou não) na rede gerenciada.

Referências

- ALMESBERGER, W. (1999) "Linux network traffic control: Implementation overview". Disponível em: <ftp://icaftp.epfl.ch/pub/people/almesber/pub/tcio-current.ps.gz>.
- BLAKE, S. *et al.* (1998) "An architecture for differentiated services". IETF Request for Comments: 2475.
- BRADEN, R.; Clark, D.; Shenker, S. (1994) "Integrated services in the internet architecture: an overview". IETF Request for Comments: 1633.
- CBQINIT. CBQ.init traffic management script. Disponível em: <http://sourceforge.net/projects/cbqinit>. Acesso em: Janeiro, 2007.
- CORDEIRO, W. *et al.* (2006) "Log Analyzer: Uma Proposta de Software Livre Para o Correlacionamento de Eventos em Arquivos de Log". In: Workshop de Software Livre 2006. Porto Alegre, RS.
- FLOYD, S.; JACOBSON, V. (1995) Link-sharing and Resource Management Models for Packet Networks. IEEE/ACM Transactions on Networking, Vol. 3 No. 4, pp. 365- 386, Agosto.
- GPL. GNU GENERAL PUBLIC LICENSE. Disponível em: <http://www.gnu.org/licenses/gpl.html>. Acesso em: Fevereiro, 2006.
- HTBINIT. HTB setup script. Disponível em: <http://sourceforge.net/projects/htbinit>. Acesso em: Janeiro, 2007.
- OLSHEFSKI, D. (2001) "Notes on linux network QoS- TCAPi version 1.0". Disponível em: <ftp://www-126.ibm.com/pub/tcapi/tcapi.tar.gz>.
- VIANNA, R. *et al.* (2006) Comparando Aspectos de Desempenho do Protocolo SNMP com Diferentes Estratégias de Gateways Web Services. In: 24o Simpósio Brasileiro de Redes de Computadores, Curitiba, Brasil.
- W3C. WORLD WIDE WEB CONSORTIUM. Disponível em: <http://www.w3c.org>. Acesso em: Janeiro, 2007.